UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

T-Explainer: uma abordagem para a explicabilidade em Aprendizado de Máquina baseada em gradientes

Evandro Scudeleti Ortigossa

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Evandro Scudeleti Ortigossa

T-Explainer: uma abordagem para a explicabilidade em Aprendizado de Máquina baseada em gradientes

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de concentração: Ciências de Computação e Matemática Computacional.

Orientador: Prof. Dr. Luis Gustavo Nonato.

USP – São Carlos
Junho de 2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

O77t

Ortigossa, Evandro Scudeleti
T-Explainer: uma abordagem para a explicabilidade em Aprendizado de Máquina baseada em gradientes / Evandro Scudeleti Ortigossa; orientador Luis Gustavo Nonato. -- São Carlos, 2024.
202 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Inteligência Artificial Explicável. 2. Interpretabilidade. 3. Modelos Caixa-Preta. 4. Atribuição de Importância. 5. Inteligência Artificial Robusta e Responsável. I. Nonato, Luis Gustavo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Evandro Scudeleti Ortigossa

T-Explainer: an explainability framework for Machine Learning based on gradients

Thesis submitted to the Institute of Mathematics and Computer Science – ICMC-USP, in accordance with the requirements of the Computer Science and Computational Mathematics Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics.

Advisor: Prof. Dr. Luis Gustavo Nonato.

USP – São Carlos
June 2024

Dedicado a todos os cientistas brasileiros que, mesmo com o pouco reconhecimento que recebem, fazem mais do que o possível para contribuir com o desenvolvimento deste país.

Agradecimentos

À Universidade de São Paulo e ao Instituto de Ciências Matemáticas e de Computação, por oferecerem ensino superior do mais alto nível e por proporcionarem um ambiente que valoriza o conhecimento científico.

Aos professores que me acompanharam desde a graduação até o doutorado, em especial ao Prof. Dr. Luis Gustavo Nonato, pela valiosa orientação, atenção, por todas as ideias inspiradoras, pelas sugestões e confiança na elaboração desta pesquisa.

Ao Prof. Dr. Rodrigo Fernandes de Mello e ao Prof. Dr. Francisco Rodrigues, por compartilharem conhecimento de alto nível em Ciência de Dados e Aprendizado de Máquina, em seus respectivos canais no *YouTube*.

Aos integrantes dos grupos de pesquisas *Explainable AI Collaboration* e *Graphics, Imaging, Visualization, and Analytics* (GIVA), pelas proveitosas discussões que reuniam pesquisadores em diferentes fusos horários, em busca de soluções para os desafios da ciência de dados. Fica também registrada a minha lembrança aos colegas do eterno *Visualization, Imaging, and Computer Graphics* (VICG), pelos momentos de pesquisa e descontração.

Aos meus queridos pais, que sempre deram todo o suporte às minhas escolhas e não mediram esforços para eu chegar até esta etapa da minha vida. Vocês são fundamentais.

Aos tios, tias, primos, primas e todos os familiares que me incentivaram, especialmente aos tios Marco e Mali que fizeram parte desta trajetória desde o primeiro dia de universidade.

A todos os amigos e amigas que de algum modo contribuíram nos estudos e trabalhos para que conseguíssemos entrar, sobreviver e passar pela universidade. Entre eles: Tamires, Carol, Isabela, Bruna, Diego, Jeyse, Albert, Mariana, Thales, Alexandre, Gabriel, Matheus, Lucas, Cesar, Fábio, Rodney, Jean, Paulo, dentre outros cujos nomes me falham a memória agora, mas sintam-se igualmente homenageados.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro nesses anos de doutorado.

Por fim, um agradecimento mais do que especial à Marina, que sempre me incentivou a evoluir e seguir em frente no trabalho, além de me motivar todos os dias a fazer o melhor para crescermos juntos.

"O que nos causa problemas não é o que não sabemos. É o que temos certeza que sabemos e que, no final, não é verdade."

Mark Twain

"É assustador imaginar que não sabemos algo, mas mais assustador ainda é imaginar que, em geral, o mundo é dirigido por pessoas que acreditam saber exatamente o que está acontecendo."

Amos Tversky

"Hoje em dia quase todas as pessoas de talento morrem de medo de serem ridículas e por isso são infelizes (...) Você é como todos os outros, isto é, como muitos, só que não precisa ser tal qual todos os outros, essa é a questão."

Dostoiévski

Resumo

ORTIGOSSA, E. S. **T-Explainer: uma abordagem para a explicabilidade em Aprendizado de Máquina baseada em gradientes**. 2024. 202 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos–SP, 2024.

Sistemas baseados em Aprendizado de Máquina têm alcançado desempenhos notáveis em muitas tarefas dentro dos mais variados domínios de aplicação. Quando aplicados em áreas sensíveis, ou seja, aquelas que podem impactar na vida e bem-estar de seus usuários, os modelos de aprendizado encontram barreiras. Isso ocorre devido à natureza complexa dos mecanismos de decisão desses modelos, que leva à falta de transparência em seus resultados e os transforma em "caixas-pretas", de onde não é possível compreender a lógica do processo de tomada de decisão. Compreender o porquê um modelo fez uma certa predição é tão e muitas vezes até mais importante do que a sua precisão preditiva, revelando a necessidade de equilíbrio entre a interpretabilidade e a acurácia. Embora não seja possível interpretar modelos complexos diretamente, é possível explicá-los. Prover explicações para decisões tomadas por sistemas computacionais pode ser visto como um meio de justificar sua confiabilidade, além de proporcionar um modo efetivo de verificar e corrigir erros antes escondidos dentro da complexidade estrutural do Aprendizado de Máquina. A pesquisa em *eXplainable Artificial Intelligence* (XAI) vem propondo diversas abordagens neste sentido, sendo os métodos de atribuição de importância de particular interesse devido à sua capacidade de explicar a importância de atributos individuais para a decisão do modelo. No entanto, as principais soluções em atribuição de importância sofrem de instabilidade, com explicações distintas podendo ser geradas a cada aplicação do método em uma mesma instância de dado. Neste cenário, esta pesquisa de doutorado vem para contribuir com o desenvolvimento da explicabilidade ao propor e desenvolver um novo método de atribuição de importâncias denominado T-Explainer. Baseado na expansão em série de Taylor, T-Explainer possui um conjunto de propriedades desejáveis, como precisão local e consistência, enquanto mantém a estabilidade das suas explicações. Finalmente, a eficácia do T-Explainer é demonstrada por meio de um conjunto de experimentos comparativos com outros métodos de atribuição de importâncias do atual estado da arte.

Palavras chave: inteligência artificial explicável; interpretabilidade; modelos caixa-preta, atribuição de importância; inteligência artificial robusta e responsável.

Abstract

ORTIGOSSA, E. S. **T-Explainer: an explainability framework for Machine Learning based on gradients**. 2024. 202 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos–SP, 2024.

Intelligent systems based on Machine Learning have achieved remarkable performance rates in a wide range of tasks from many domains. However, understanding why an algorithm makes a particular decision is still an open question in the research area. When applied in sensible environments, which may cause an impact on the life and welfare of its users, the learning-based models raise some problems. The complex nature of these methods' decision mechanisms makes them the so-called "black boxes," from which it is not trivial for humans to understand the logic behind the decision-making process of the models. Furthermore, in some contexts, the reasoning that led a model to provide some specific prediction is more important than the accuracy, thus introducing a trade-off between interpretability and model accuracy. Providing explanations to computer-aided systems decisions can be seen as a way to justify their reliability, besides providing an effective tool for checking and correcting errors previously hidden within learning models, opening up an avenue of possibilities for responsible applications. The eXplainable Artificial Intelligence (XAI) research has proposed a number of approaches in this context, being feature attribution techniques of particular interest due to their ability to characterize the importance of particular features on the model decision. However, the lead solutions in such domain suffer from instability, i.e., distinct explanations may result from each application of the explanation method on the same data instance. In this scenario, this Ph.D. research aims to contribute to Machine Learning explainability by proposing a new XAI method called T-Explainer, a Taylor expansion-based technique that holds a set of desirable properties, such as local accuracy and consistency, while still being stable in its explanations. Finally, the results demonstrate T-Explainer's effectiveness through benchmarking experiments and comparisons against state-of-the-art references in feature attribution.

Keywords: explainable artificial intelligence*;* interpretability; black-box models; feature importance attribution; robust and responsible artificial intelligence.

Lista de Figuras

Figura 2.1 – Fluxograma do processo de desenvolvimento e aplicação de modelagens de Aprendizado de Máquina. 40

Figura 2.2 – Representação de uma SVM. As amostras são classificadas e separadas em espaços de características. 42

Figura 2.3 – Representação do Perceptron, o neurônio aritmético baseado no modelo de McCulloch e Pitts (1943). 43

Figura 2.4 – Rede de neurônios ilustrando uma MLP com três camadas. 44

Figura 2.5 – Protótipo de uma rede neural profunda com n_x entradas, m camadas e n_m saídas. 47

Figura 2.6 – Representação de uma Árvore de Decisão aplicada na classificação do comportamento de motoristas baseado em limiares de velocidade. . . 48

Figura 2.7 – Construção de um classificador *Random Forest*. 49

Figura 2.8 – Treinamento de um preditor $\hat{f}$ com *Boosting* em k modelos intermediários. 50

Figura 2.9 – A contribuição do viés e da variância para o erro em função da complexidade. 53

Figura 2.10 – Diferença entre interpretabilidade e explicabilidade dentro do XAI. . 58

Figura 2.11 – O desenvolvimento tradicional do Aprendizado de Máquina, centrado no algoritmo (contém humor). 60

Figura 2.12 – A explicabilidade se posiciona como um complemento do Aprendizado de Máquina, ao construir ferramentas capazes de facilitar a interpretação de modelos caixa-preta. 61

Figura 2.13 – Explicando a predição de um diagnóstico de gripe. Os sintomas que contribuem com o resultado estão destacados em verde, os sintomas que não contribuem estão em vermelho. 62

Figura 2.14 – Relação entre a interpretabilidade e o desempenho preditivo dos principais modelos de aprendizado. 64

Figura 2.15 – Taxonomia das metodologias XAI em relação aos diferentes objetivos de compreensão dos problemas caixa-preta. 65

Figura 2.16 – Diagrama comparativo entre as metodologias de explicação globais e locais. 68
Figura 2.17 – O propósito de uma técnica de explicação também depende do público a quem ela é destinada. 70
Figura 2.18 – Desafios no desenvolvimento responsável de explicações para sistemas baseados em Aprendizado de Máquina. 71

Figura 3.1 – Diferentes abordagens *post-hoc*. Os métodos podem ser classificados de acordo com a apresentação das explicações. 80
Figura 3.2 – Explicação gerada pelo LIME sobre uma amostra do conjunto Iris classificada como pertencente à classe *virginica*. 89
Figura 3.3 – Diagrama com as possíveis combinações para um conjunto contendo três atributos. 92
Figura 3.4 – Ferramentas gráficas do SHAP aplicadas em todo o conjunto Iris (global) e também sobre uma amostra classificada como pertencente à classe *virginica* (local). 96

Figura 4.1 – Fluxograma com os módulos básicos do T-Explainer. 113
Figura 4.2 – Diagrama esquemático da aplicação do T-Explainer para gerar explicações por meio da descoberta dos atributos mais importantes. 114
Figura 4.3 – A importância ϕ_i é obtida pela projeção do gradiente de f sobre o i-ésimo eixo do espaço de atributos de $\mathbf{x}$. 118
Figura 4.4 – *Pipeline* detalhado do T-Explainer. 121
Figura 4.5 – Aproximação de $f'(\mathbf{x})$ utilizando a *centered finite difference* é interpretada como uma média das inclinações das linhas secantes laterais a $f(\mathbf{x})$. 122
Figura 4.6 – Tratamento de dados categóricos utilizando *one-hot-encoding*. 126
Figura 4.7 – Indução de continuidade em uma variável categórica transformada por *one-hot-encoding* por meio de perturbação uniforme. 127
Figura 4.8 – Indução de continuidade sobre atributos com valores discretos binários. 128

Figura 5.1 – Distribuição dos atributos preditivos do conjunto de dados 4-FT. . . 137
Figura 5.2 – Importâncias atribuídas por TreeSHAP e T-Explainer para a mesma instância classificada com o modelo XRFC sobre os dados 4-FT. . . . 140
Figura 5.3 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas a partir do conjunto de dados 4-FT. 142
Figura 5.4 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre o conjunto 20-FT. 145

Figura 5.5 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre o conjunto de dados 4-FT-2CAT. 148
Figura 5.6 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre os dados 6-FT-2CAT. . . 151
Figura 5.7 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre o conjunto 25-FT-5CAT. 154
Figura 5.8 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas sobre o conjunto *Breast Cancer Wisconsin*. 157
Figura 5.9 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas sobre o conjunto *Banknote Authentication*. 159
Figura 5.10 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas a partir do conjunto HIGGS. . . 162
Figura 5.11 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas sobre o conjunto *Titanic Disaster*.165
Figura 5.12 – *Summary plots* com as explicações feitas pelos métodos SHAP e T-Explainer para as predições geradas sobre o conjunto *German Credit*. 168
Figura 5.13 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para predições feitas sobre o conjunto HELOC. 171
Figura 5.14 – Tempos de execução dos métodos SHAP e T-Explainer explicando as predições do modelo 3H-NN treinado sobre o conjunto 16-FT. 174
Figura 5.15 – Demanda por memória RAM na execução dos métodos SHAP e T-Explainer explicando as predições do modelo 3H-NN sobre os dados 16-FT. 175

Lista de Tabelas

Tabela 3.1 – Sumário das metodologias e pesquisas em *Explainable Artificial Intelligence* revisadas nesta pesquisa. 107

Tabela 5.1 – Configuração dos modelos classificadores caixa-preta utilizados. . . . 135
Tabela 5.2 – T-Explainer, SHAP, e TreeSHAP *Local Accuracy* explicando as predições do modelo XRFC treinado no conjunto de dados sintéticos 4-FT. 140
Tabela 5.3 – T-Explainer, SHAP, e TreeSHAP *Local Accuracy* explicando as predições do modelo XRFC treinado sobre o conjunto de dados 20-FT. . . 141
Tabela 5.4 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 4-FT. 141
Tabela 5.5 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 4-FT. 142
Tabela 5.6 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 20-FT. 143
Tabela 5.7 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 20-FT. 144
Tabela 5.8 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 4-FT-2CAT. 147
Tabela 5.9 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 4-FT-2CAT. 147
Tabela 5.10 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 6-FT-2CAT. 149
Tabela 5.11 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 6-FT-2CAT. 150
Tabela 5.12 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto sintético com atributos categóricos 25-FT-5CAT. 152
Tabela 5.13 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre os dados sintéticos com atributos categóricos 25-FT-5CAT. 153

Tabela 5.14 – Principais propriedades dos conjuntos de dados reais desta seção. *Cancer* e *German* se referem aos conjuntos *Breast Cancer Wisconsin* e *German Credit*, respectivamente. 156
Tabela 5.15 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto *Breast Cancer Wisconsin*. 156
Tabela 5.16 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto *Banknote Authentication*. 158
Tabela 5.17 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto HIGGS. 160
Tabela 5.18 – Estabilidade dos métodos XAI sobre as predições do modelo 5H-64-NN treinado no conjunto de dados HIGGS. 160
Tabela 5.19 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto *Titanic Disaster*. 164
Tabela 5.20 – Estabilidade dos métodos XAI sobre as predições do modelo 3H-NN treinado no conjunto *German Credit*. 167
Tabela 5.21 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto HELOC. 170
Tabela 5.22 – Estabilidade dos métodos XAI sobre as predições do modelo 5H-128-NN treinado no conjunto de dados HELOC. 170

Lista de Abreviaturas e Siglas

AI . . . . . *Artificial Intelligence*
ANN . . . . . *Artificial Neural Network*
Bagging . . . . . *Bootstrap AGGregatING*
CCPA . . . . . *California Consumer Privacy Act*
CNN . . . . . *Convolutional Neural Network*
DeepSHAP . . . . . *Deep Learning SHAP*
DeepLIFT . . . . . *Deep Learning Important FeaTures*
DNN . . . . . *Deep Neural Network*
GAM . . . . . *Generalized Additive Model*
GDPR . . . . . *General Data Protection Regulation*
GPU . . . . . *Graphics Processing Unit*
k-means . . . . . *k-means clustering*
k-NN . . . . . *k-Nearest Neighbors*
KernelSHAP . . . . *Kernel-based SHAP*
LIME . . . . . *Local Interpretable Model-Agnostic Explanations*
LRP . . . . . *Layer-wise Relevance Propagation*
LSTM . . . . . *Long Short-Term Memory*
MLP . . . . . *Multilayer Perceptron*
RNA . . . . . Rede Neural Artificial
RNN . . . . . *Recurrent Neural Network*
ROAR . . . . . *RemOve And Retrain*
SHAP . . . . . *SHapley Additive exPlanations*
SVM . . . . . *Support-Vector Machine*
t-SNE . . . . . *t-Distributed Stochastic Neighbor Embedding*
TreeSHAP . . . . . *Tree Ensemble-based SHAP*
XAI . . . . . *EXplainable Artificial Intelligence*

Sumário

1 Introdução 27
1.1 Contextualização 27
1.2 Objetivo e Contribuições da Pesquisa 30
1.2.1 Produção Bibliográfica 33
1.3 Organização do Texto 33

2 Conceitos e Fundamentos 35
2.1 Considerações Iniciais 35
2.2 Terminologia Básica 35
2.3 Um Breve Histórico Sobre Aprendizado de Máquina 37
2.3.1 Regressão Logística 40
2.3.2 *Support Vector Machine* 41
2.3.3 Redes Neurais Artificiais 42
2.3.4 *Random Forests* 47
2.4 O Que Define Algo Complexo? 50
2.4.1 Viés e Variância 51
2.4.2 Não-linearidade 54
2.5 *EXplainable Artificial Intelligence* (XAI) 56
2.5.1 Definições Conceituais 56
2.5.2 Necessidades e Desafios da Explicabilidade 59
2.5.3 Objetivos da Explicabilidade 63
2.5.4 Categorização das Abordagens XAI 64
2.5.5 Responsabilidade em Inteligência Artificial 69
2.6 Tecnologias e Ferramentas Utilizadas 71
2.7 Considerações Finais 72

3 Trabalhos Relacionados 75
3.1 Considerações Iniciais 75
3.2 Estado da Arte em *EXplainable Artificial Intelligence* 75
3.2.1 XAI Baseada em Aproximações 79
3.2.2 XAI Baseada em Visualização de Informações 81

3.2.3 XAI Baseada em *Decision Boundaries* . . . 83
3.2.4 XAI Baseada em Exemplos Contrastivos e Contrafactuais . . . 84
3.2.5 XAI Baseada em Decomposição de Sinais e Gradientes . . . 86
3.2.6 XAI Baseada em Simplificação . . . 87
3.2.7 XAI Baseada em *Shapley Values* . . . 90
3.2.8 Questões em XAI relacionadas aos *Shapley Values* . . . 99
3.3 Métodos e Métricas de Avaliação . . . 101
3.4 Limitações em XAI . . . 104
3.5 Resumo e Caracterização dos Métodos XAI . . . 106
3.6 Considerações Finais . . . 106

4 Desenvolvimento da Proposta . . . 109
4.1 Considerações Iniciais . . . 109
4.2 Visão Geral . . . 109
4.3 T-Explainer . . . 113
4.4 T-Explainer – Propriedades . . . 118
4.4.1 *Local Accuracy* . . . 118
4.4.2 *Missingness* . . . 119
4.4.3 *Consistency* . . . 119
4.5 T-Explainer – Aspectos Computacionais . . . 120
4.5.1 Otimizador para Aproximação de Derivadas . . . 122
4.5.2 Tratamento de Atributos Categóricos . . . 125
4.6 Métricas para Avaliação Quantitativa . . . 128
4.6.1 *Relative Input/Output Stability* . . . 128
4.6.2 *Run Explanation Stability* . . . 130
4.6.3 *Faithfulness of an Additive Explanator* . . . 131
4.7 Considerações Finais . . . 131

5 Resultados Experimentais . . . 133
5.1 Considerações Iniciais . . . 133
5.2 Configuração dos Experimentos . . . 133
5.3 Análise Comparativa – Dados Sintéticos . . . 136
5.3.1 Atributos Sintéticos Numéricos . . . 137
5.3.2 Atributos Sintéticos Categóricos . . . 146
5.4 Análise Comparativa – Dados Reais . . . 154
5.4.1 Atributos Reais Numéricos . . . 156
5.4.2 Atributos Reais Categóricos . . . 162
5.5 Desempenho Computacional . . . 172

5.6 Considerações Finais 176

6 Conclusão 179

6.1 Limitações e Trabalhos Futuros 179

6.2 Considerações Finais 183

Referências Bibliográficas 185

Capítulo 1

Introdução

1.1 Contextualização

Nos últimos anos, sistemas baseados em inteligência artificial têm alcançando níveis de desempenho sem precedentes, aprendendo a solucionar tarefas cada vez mais complexas, o que coloca a Inteligência Artificial no centro de muitas áreas e atividades em que a tecnologia se posiciona como um agente transformador (RUSSELL; NORVIG, 2010). O conceito de Inteligência Artificial não é uma novidade, com suas raízes remontando a várias décadas no passado, entre os primeiros passos de desenvolvimento da computação, acompanhando o sonho de se construir máquinas "inteligentes" que fossem capazes de tomar decisões à semelhança do pensamento humano. Atualmente, a adoção de sistemas com capacidade de aprendizado e adaptação está num momento notável de ubiquidade que, se aproveitado adequadamente, pode entregar grandes resultados dentro dos muitos setores em que estes sistemas se inserem (ARRIETA *et al.*, 2020). A Inteligência Artificial é considerada de fundamental importância para acelerar os avanços futuros no desenvolvimento de uma sociedade mais algorítmica e digital (WEST, 2018).

O Aprendizado de Máquina (*Machine Learning*) é um ramo da Inteligência Artificial que faz uso ativo do conhecimento de áreas das ciências como a matemática, física, biologia, estatística, linguística, psicologia, entre outras, para simular computacionalmente as habilidades cognitivas da inteligência humana. O Aprendizado de Máquina tem recebido considerável atenção da comunidade de pesquisas devido à sua capacidade e precisão em realizar predições sobre uma grande variedade de fenômenos complexos (MURDOCH *et al.*, 2019). Muitos algoritmos baseados em aprendizado surgiram na última década, especialmente após 2012, impulsionados pela expressiva redução nos custos de armazenamento de dados, que possibilitou o aumento das informações disponíveis em grandes conjuntos de dados, além da significativa melhoria no *hardware*, com destaque para as GPUs (*Graphics Processing Units*) com alto poder computacional, combinados com o desenvolvimento de

novos algoritmos de otimização. Também, é possível atribuir esses avanços às novas linguagens de programação e às bibliotecas de código aberto de alta qualidade, que permitem a programadores em todo o mundo criarem ou executarem protótipos, testando modelos e algoritmos mais rapidamente (DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018).

A sofisticação dos modelos de aprendizado aumentou a tal ponto que estes têm alcançado desempenhos (sobre)humanos em muitas tarefas que antes eram consideradas inatingíveis computacionalmente (LECUN; BENGIO; HINTON, 2015), com alguns sistemas inteligentes demandando quase nenhuma intervenção humana para seu ajuste ou treinamento (ARRIETA *et al.*, 2020). De acordo com Zhang e Zhu (2018) e Chakraborty *et al.* (2017), alguns modelos modernos já superaram as capacidades humanas em várias tarefas e, por isso, têm sido frequentemente utilizados nos mais diversos processos, desde a recomendação de produtos, músicas, filmes e amigos em redes sociais, até a tomada de decisões em áreas críticas, como medicina, mercados financeiros, carros autônomos, planejamento estratégico de governos, bioinformática e sistemas criminais. Entretanto, esses sistemas também chamam a atenção para problemas relacionados à confiança, especialmente quando as decisões envolvem contextos sensíveis, como diagnóstico médico, em que é importante conhecer as razões por trás das decisões (ADADI; BERRADA, 2018).

Embora tais modelos de aprendizado tenham alcançado altos níveis de precisão preditiva, a natureza complexa na qual operam acaba reduzindo a "transparência" dos seus processos internos de decisão, tornando-os verdadeiras "caixas-pretas" (BURRELL, 2016). Algoritmos de aprendizado sofrem com a opacidade, isto é, descrever qual o grau de impacto que cada parte da informação de entrada tem na decisão tomada, é um problema fundamental neste contexto (SAMEK; WIEGAND; MÜLLER, 2017). Confiar decisões importantes a um sistema que não pode ser interpretado ou que não fornece explicações sobre seus resultados, pode ser perigoso e levanta uma problemática bastante contundente quanto à pouca habilidade que os humanos têm em compreender os mecanismos de funcionamento desses algoritmos (CARUANA *et al.*, 2015; ADADI; BERRADA, 2018).

Como explicar uma predição quando um modelo comete um erro, ou mesmo como detectar uma predição errônea? Infelizmente, é difícil dizer o que está errado porque as múltiplas estruturas dos modelos de aprendizado modernos os tornam virtualmente impossíveis de se interpretar (RIBEIRO; SINGH; GUESTRIN, 2016c; ARRIETA *et al.*, 2020). Consequentemente, diversos trabalhos na literatura têm explorado as desvantagens de modelos de alto desempenho quando expostos a ataques ou pequenas modificações nos dados de entrada (GOODFELLOW; SHLENS; SZEGEDY, 2014; KARIMI; DERR; TANG, 2019; SU; VARGAS; SAKURAI, 2019). Se não é possível confiar em algo incompreensível, isso torna poderosos modelos inteligentes em algoritmos inúteis para tomar decisões críticas (TJOA; GUAN, 2020). Neste cenário, "interpretabilidade" e "explicabilidade"

emergem como novos conceitos trazidos à luz pela comunidade de pesquisas em Aprendizado de Máquina, surgindo a área do *eXplainable Artificial Intelligence* (XAI).

O termo "interpretar" é definido como a habilidade de apresentar algo de modo compreensível (DOSHI-VELEZ; KIM, 2017). Os humanos podem justificar suas ações e previsões por meio de uma série de escolhas logicamente consistentes, descritíveis e compreensíveis produzidas por sua habilidade de "pensar" (CHAKRABORTY *et al.*, 2017). Exceto pelo resultado final, é difícil compreender o processo lógico de um modelo de aprendizado, o que acaba impedindo a interpretação dos seus resultados (ZHANG; ZHU, 2018). Porém, quando não é possível interpretar diretamente uma decisão, é possível buscar por elementos explicativos que são capazes de clarear os opacos processos de tomada de decisão de modelos complexos. A explicabilidade pode levar a um avanço no sentido de obter mais transparência desses modelos complexos, por meio de explicações das características de processamento e/ou das escolhas lógicas feitas pelos algoritmos, depurando a mera representação de resultados. Assim, dentro do contexto de Aprendizado de Máquina, a explicabilidade pode ser vista como a contrapartida do método de racionalização da tomada de decisões no pensamento humano (CHAKRABORTY *et al.*, 2017).

Em linhas gerais, todas as iniciativas e esforços feitos no sentido de reduzir a complexidade de uma modelagem baseada em aprendizado, com a finalidade de melhorar a transparência e a compreensão de seu funcionamento, podem ser consideradas abordagens de XAI (MILLER, 2019). Especificamente, XAI é uma área de pesquisas que se inspira em ideias das ciências sociais, além de levar em conta a psicologia da explicação, para criar técnicas que tornem os resultados dos sistemas de aprendizado de máquina mais explicáveis, mantendo o alto nível de desempenho preditivo e capacitando os humanos a compreender, confiar apropriadamente e gerenciar as próximas gerações expostas à Inteligência Artificial (GUNNING; AHA, 2019). No entanto, interpretabilidade e explicabilidade não são características monolíticas, com as definições formais do que é a interpretabilidade e a explicabilidade permanecendo ainda como algo de compreensão subjetiva na literatura especializada, sem especificações consensuais do que seria um algoritmo de aprendizado interpretável ou um meio adequado de explicação e de como estes devem ser avaliados (LIPTON, 2018).

A classe de métodos XAI baseados em atribuição de importância é de particular interesse dentro da explicabilidade. Segundo Tan *et al.* (2023), muitos métodos XAI, tanto locais quanto globais, explícita ou implicitamente, realizam explicações caracterizando a importância de atributos no modelo preditor. Um método de atribuição de importância quantifica individualmente a contribuição das variáveis (atributos) de entrada para as predições de um modelo de caixa-preta (ADADI; BERRADA, 2018). No entanto, diferentes métodos de atribuição de importância podem gerar explicações discordantes uns dos

outros (TAN *et al.*, 2023). Além disso, métodos populares como SHAP (LUNDBERG; LEE, 2017) e LIME (RIBEIRO; SINGH; GUESTRIN, 2016c) enfrentam sérias desvantagens, incluindo problemas com instabilidade. Outros métodos, como os baseados em gradiente, oferecem maior estabilidade, mas têm aplicabilidade limitada, pois requerem modelos de aprendizado com parâmetros diferenciáveis (SMILKOV *et al.*, 2017).

Vale destacar que a necessidade de se interpretar ou explicar o comportamento de modelos de aprendizado que afetam diretamente a vida das pessoas, não é apenas uma característica desejável, mas também legal. A União Europeia introduziu o direito à explicação em sua *General Data Protection Regulation* (GDPR), como tentativa de mitigar os impactos sociais dos sistemas computacionais, incluindo diretrizes sobre a tomada de decisão algorítmica (EU Regulation, 2016). Para se ter uma ideia, entre outras exigências que serão apresentadas ao longo deste texto, o GDPR declara que o controlador deve garantir o direito dos indivíduos em obter informações sobre as decisões de qualquer sistema automatizado (AMPARORE; PEROTTI; BAJARDI, 2021). De modo similar, o recente *California Consumer Privacy Act* (CCPA) (CALIFORNIA, 2021) define direitos com relação ao uso e proteção de dados, que estão influenciando a legislação sobre privacidade nos Estados Unidos. Então, explicar as decisões de modelos caixa-preta deixa de ser algo meramente desejável para se tornar mandatório, motivando a recente explosão no desenvolvimento de técnicas XAI (GILPIN *et al.*, 2018).

Explicar a racionalização por trás das predições pode ser tão ou ainda mais importante do que o desempenho preditivo em si (LUNDBERG *et al.*, 2020). O desenvolvimento da explicabilidade vem para conferir uma camada a mais no sucesso inegável visto na área do Aprendizado de Máquina, ao ir além das métricas de *performance* que são tradicionalmente utilizadas, buscando por meios de prover entendimento direto sobre o comportamento dos algoritmos de aprendizado. Com isso, argumenta-se que a explicabilidade de modelos de aprendizado de máquina complexos é uma necessidade que segue na direção das recentes demandas e regulamentações, que exigem mais transparência por parte dos sistemas de tomada de decisão que afetam a vida dos seus usuários. Motivada pelas observações e preocupações elencadas, esta pesquisa de doutorado vem para contribuir no contexto entre o Aprendizado de Máquina e a explicabilidade, provendo uma sólida base à luz da literatura dos conceitos e do atual estado da arte em XAI, para fundamentar o desenvolvimento e avaliação de novos mecanismos para a explicabilidade de predições.

1.2 Objetivo e Contribuições da Pesquisa

Apesar dos avanços significativos na definição e construção de algoritmos de aprendizado, a explicabilidade é uma linha de pesquisas relativamente nova. Mesmo assim, a comunidade

de *Explainable Artificial Intelligence* tem se mostrado bastante ativa, fomentando muitos eventos científicos dedicados ao tema (ADADI; BERRADA, 2018), com artigos e pesquisas de alta qualidade sendo publicados em importantes veículos científicos, como a renomada revista Nature, por exemplo (LUNDBERG *et al.*, 2018; LAPUSCHKIN *et al.*, 2019). Isso demonstra o reconhecimento da importância em se explicar as predições de modelos, não somente para satisfazer requisitos de transparência, mas também para facilitar a interação entre humanos e a Inteligência Artificial, algo que vem para ajudar no desenvolvimento, manutenção e monitoramento de sistemas de aprendizado (LUNDBERG *et al.*, 2020).

Em síntese, a problemática a ser abordada nesta pesquisa de doutorado pode ser descrita do seguinte modo:

- **Problema** – Os modelos de aprendizado não analisam dados do mesmo modo que os humanos. Estas aplicações utilizam mecanismos matemáticos complexos para encontrar padrões que um analista humano pode não conhecer ou mesmo entender completamente (PASSI; JACKSON, 2018). Embora os modelos atuais apresentem alto poder discriminante (ao custo da alta complexidade e consequente baixa interpretabilidade das suas representações), essa alta precisão não garante que as suas decisões sejam de fato justas e não contenham algum tipo de viés indevido. A falta de poder explicativo levanta questões de confiabilidade que acabam transformando os algoritmos de aprendizado em sistemas de apoio à decisão pouco confiáveis, dificultando a sua implantação em diversos cenários de mundo real (DORAN; SCHULZ; BESOLD, 2017). Assim surge o direito à explicação, ou seja, existe a necessidade de tornar mais transparentes as decisões tomadas por modelos automatizados que podem afetar a vida de seus usuários. Mas ainda que existam explicações, estas podem não ser significativas ou mesmo úteis. Muitos métodos XAI têm se provado ferramentas valiosas, mas eles vêm com algumas limitações que reduzem a confiança depositada em suas explicações, entre elas está a instabilidade. Um explicador alcança a estabilidade quando este gera explicações consistentes entre (*i*) múltiplas execuções sobre a mesma instância e (*ii*) ao explicar instâncias similares. Segundo Amparore, Perotti e Bajardi (2021), para que um método explicador seja confiável, este deve ser, no mínimo, estável. Logo, a explicabilidade deve atender a certos critérios, entre eles, a precisão e a confiabilidade. Se as explicações não são consistentes, elas não são confiáveis e, por consequência, são inúteis.
- **Objetivo** – O sucesso empírico do Aprendizado de Máquina deriva de algoritmos computacionalmente eficientes e de seu grande espaço paramétrico, com centenas (milhares ou milhões) de parâmetros (ARRIETA *et al.*, 2020). Extrair conhecimento sobre o modo de trabalho destes algoritmos, tornando-os mais transparentes e

verificáveis motiva as pesquisas em XAI. No entanto, foram identificados problemas na aplicação da explicabilidade, principalmente quanto à falta de confiabilidade derivada da instabilidade apresentada por muitos dos métodos mais amplamente conhecidos. A estabilidade é uma propriedade fundamental na explicabilidade pois, se um método expicador gerar explicações inconsistentes, será difícil confiar em tais explicações. Neste contexto, esta pesquisa tem como objetivo promover avanços no sentido de elucidar as predições de modelos de aprendizado de máquina, buscando compreender o porquê tais decisões foram alcançadas, de modo confiável e consistente. Isso foi feito com o desenvolvimento do T-Explainer, um novo método XAI baseado nos sólidos fundamentos matemáticos da expansão de Taylor para realizar explicações de predições, que possui propriedades desejáveis, como precisão, consistência e estabilidade entre múltiplas execuções ao explicar a mesma instância ou instâncias similares;

- **Contribuições** – O T-Explainer é uma técnica para explicação de predições por atribuição de importância aditiva que possui propriedades desejáveis, como precisão local e consistência (LUNDBERG; LEE, 2017), ao aproximar o comportamento local de modelos de caixa-preta, permitindo interpretações estáveis e confiáveis. Além disso, o T-Explainer foi integrado com ferramentas de visualização de informações disponíveis na biblioteca SHAP, ampliando a usabilidade do método. Para atestar a qualidade e utilidade da metodologia desenvolvida, foram realizados testes com diversas métricas quantitativas para avaliação de métodos XAI, em termos de estabilidade local e preservação da aditividade, comparando o desempenho do T-Explainer em um conjunto abrangente de comparações com métodos do estado da arte em atribuição de importâncias a nível local. Todos os desenvolvimentos desta pesquisa, incluindo o T-Explainer, métricas quantitativas de avaliação de estabilidade, métodos para geração de dados sintéticos e de pré-processamento, estão disponíveis *online* em uma biblioteca *Python* que integra esta pesquisa, transformando o T-Explainer em um *framework* XAI abrangente e de fácil utilização em diferentes domínios de aplicação. O T-Explainer vem para se posicionar como uma alternativa XAI estável e que não utiliza componentes de natureza aleatória.

Existem propriedades consideradas desejáveis para métodos de explicabilidade pois, estas propriedades, se atendidas, fornecem base teórica para as explicações no sentido de uma distribuição justa dos valores entre os atributos. O T-Explainer difere de outros métodos XAI da atual literatura por apresentar garantias teóricas sólidas que lhe rendem explicações consistentes. Mais especificamente, o T-Explainer é um método baseado em gradientes que define um procedimento de otimização para estimar derivadas parciais, sob

uma modelagem de atribuição de importâncias aditiva que trabalha com dados numéricos e categóricos. Embora a versão atual ainda não esteja totalmente desenvolvida para operar com modelos baseados em árvores, o T-Explainer é projetado para ser independente de modelos, capaz de gerar explicações para uma ampla gama de arquiteturas de aprendizado de máquina, pois não depende de informações das estruturas internas, resultados parciais, instâncias de referência complexas ou quaisquer parâmetros relacionados à arquitetura de modelo, como é o caso dos demais métodos baseados em gradientes.

Para avaliar a qualidade e utilidade do T-Explainer, foram realizados diversos testes com comparações quantitativas contra métodos XAI de atribuição de importâncias bem-estabelecidos. Os experimentos se baseiam em dados tabulares e modelos de aprendizado complexos de duas arquiteturas distintas – Redes Neurais e *Gradient-boosted Tree Ensembles*. Esta é uma configuração típica que segue as práticas da literatura recente, focando em Redes Neurais como caixas-pretas.

1.2.1 Produção Bibliográfica

Foram elaborados dois artigos a partir dos resultados desta pesquisa, um deles submetido no periódico *IEEE Intelligent Systems* e o outro no periódico *IEEE Access*:

- ORTIGOSSA, E. S.; DIAS, F. F.; BARR, B.; SILVA, C. T.; NONATO, L. G. T-Explainer: A Model-Agnostic Explainability Framework Based on Gradients, *IEEE Intelligent Systems*, 2024. *Preprint arXiv:2404.16495*.
- ORTIGOSSA, E. S.; GONÇALVES, T.; NONATO, L. G. EXplainable Artificial Intelligence (XAI)–From Theory to Methods and Applications, *IEEE Access*, vol. 12, pp. 80799-80846, 2024. DOI: 10.1109/ACCESS.2024.3409843.

O primeiro artigo descreve o T-Explainer (atualmente em processo de revisão, mas com preprint já disponível no *arXiv*), enquanto o segundo artigo (publicado) aborda aspectos teóricos e práticos sobre a aplicação de XAI em diferentes contextos.

1.3 Organização do Texto

Para documentar e demonstrar o que foi produzido, este documento está estruturado da seguinte maneira:

- No Capítulo 2 são descritos os conceitos teóricos que fundamentam o desenvolvimento desta pesquisa, desde a evolução do Aprendizado de Máquina até a teoria por trás da explicação de suas predições. O capítulo prepara o leitor com um esclarecimento sobre os conceitos e a terminologia abordada neste documento;

- No Capítulo 3 é apresentada uma revisão bibliográfica sobre alguns dos trabalhos considerados estado da arte dentro do escopo desta pesquisa. As características dos métodos elencados foram identificadas por meio de um estudo crítico e comparativo, com destaque para as suas vantagens e limitações;
- No Capítulo 4 são especificadas as definições, as propriedades e os aspectos computacionais da metodologia XAI desenvolvida, motivando as técnicas e abordagens escolhidas para gerar explicações sobre predições de modelos complexos, além de apresentar as métricas quantitativas utilizadas;
- No Capítulo 5 estão descritos os resultados obtidos, revelando as capacidades e funcionalidades do T-Explainer e as configurações dos experimentos realizados para validar adequadamente estes resultados, comparando o T-Explainer com outros métodos XAI bem conhecidos, sob diferentes contextos de modelos e dados;
- Finalizando, no Capítulo 6 há uma discussão que leva em consideração não apenas as vantagens e as principais contribuições alcançadas para a explicabilidade, mas também as limitações identificadas na estratégia apresentada, além de um resumo dos trabalhos futuros que podem ser derivados deste.

Capítulo 2

Conceitos e Fundamentos

2.1 Considerações Iniciais

Antes de mergulhar no processo de desenvolvimento, deve ser estabelecida a base conceitual necessária para melhor compreender os detalhes técnicos desta pesquisa, o que inclui uma visão geral sobre as características dos algoritmos de aprendizado mais utilizados e o porquê estes são considerados "caixas-pretas", bem como as definições sobre as abordagens de explicabilidade. Neste capítulo são apresentados os fundamentos teóricos necessários para atingir os objetivos e contribuições esperadas desta pesquisa.

Na Seção 2.2 são definidas as terminologias básicas utilizadas no ambiente desta pesquisa. A Seção 2.3 apresenta um histórico sobre a evolução dos modelos de aprendizado de máquina, desde os paradigmas de aprendizado, as características das principais técnicas, até as suas vantagens e limitações. Na Seção 2.4, há uma discussão sobre complexidade. Já na Seção 2.5, é feita uma análise sobre os conceitos que envolvem a área de XAI, trazendo à luz suas definições e desafios. As tecnologias computacionais utilizadas para desenvolver a metodologia proposta nesta pesquisa estão descritas na Seção 2.6. Por fim, as considerações finais deste capítulo estão na Seção 2.7.

2.2 Terminologia Básica

Como o leitor já deve ter notado, foram utilizadas várias palavras em inglês ao longo deste texto. Embora existam traduções em português para alguns termos ou nomes de métodos citados aqui, é notório que a literatura especializada na área de *Machine Learning* esteja majoritariamente escrita em língua inglesa. Destaca-se que praticamente todo o material considerado referência e utilizado na elaboração deste trabalho, vem de veículos científicos reconhecidos, como jornais e revistas, que estão todos publicados em inglês, salvo algumas poucas exceções encontradas em páginas da *internet* com curiosidades e

pequenas informações, em português.

O próprio nome "*Machine Learning*" possui contraparte traduzida para o português (Aprendizado de Máquina), entretanto, é bastante frequente encontrar o termo original em inglês dentro de conteúdos escritos em língua portuguesa. Além disso, existe uma certa falta de consenso quanto à exatidão das traduções, por exemplo: Aprendizagem de Máquina também pode ser encontrado como tradução alternativa de *Machine Learning*.

Para ser fiel à correta nomenclatura, o autor deste texto se compromete em utilizar a terminologia adequada, preferencialmente em português. No entanto, quando não houver uma tradução adequada, quando não houver consistência ou precisão na tradução ou mesmo quando a versão original for de amplo uso, será mantida a terminologia em inglês, mas sempre destacada em itálico e acompanhada de uma justificativa ou breve explicação de seu significado. Este é o caso da área tema desta pesquisa: *Explainable Artificial Intelligence*, que poderia ser traduzida como "Inteligência Artificial Explicável". Porém e talvez por ser uma área recente, ao se pesquisar pela versão em português, encontram-se páginas com o acrônimo em inglês, XAI, e raras publicações em veículos expressivos, mas sem uma concordância quanto à tradução (o nome em inglês da área não é unanimidade, com *Explainable Machine Learning* também sendo utilizado, embora menos frequente).

A literatura XAI é vasta em desenvolver soluções para problemas do tipo *black box*, ou seja, modelos não interpretáveis ou caixas-preta. Especificamente, o T-Explainer é um método baseado em gradientes (*gradient-based*) que gera explicações por meio da importância de atributos. É comum encontrar na literatura, as nomenclaturas *feature attribution* ou *feature importance* para denominar a classe de métodos XAI que atribuem importância como meio de explicação.

Outro termo que será utilizado com frequência ao longo desta tese é "modelo". A palavra modelo em si pode denotar os mais diversos significados dentro das diferentes áreas em que esta se insere e, mesmo restringindo o escopo apenas para o Aprendizado de Máquina, ainda assim é possível haver algum mal-entendimento. Por isso e seguindo a terminologia comumente adotada dentro da literatura XAI, é necessário fixar que a palavra modelo, sempre que utilizada neste texto, será empregada como um modo simplificado de alusão a alguma metodologia ou algoritmo de aprendizado de máquina.

Os algoritmos de aprendizado abordados aqui, são usualmente treinados sobre bases de dados multidimensionais, que são conjuntos de dados cujas m instâncias individuais são compostas por uma coleção de características, formalmente definido por:

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\} \tag{2.1}$$

em que cada vetor $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$ é uma instância de dado em $\mathbb{R}^n$. Por convenção, os elementos x_i que caracterizam as instâncias nesse tipo de conjunto de dados, serão

nomeados aqui como atributos ou variáveis dos dados. Já os múltiplos elementos que compõem os modelos de aprendizado e que são atualizados durante os procedimentos de aprendizado, ou seja, que variam conforme o modelo é treinado, serão chamados de parâmetros do modelo ou simplesmente de parâmetros.

2.3 Um Breve Histórico Sobre Aprendizado de Máquina

Inteligência Artificial e Aprendizado de Máquina são duas áreas de pesquisa em estreita correlação e que atualmente são associadas, com frequência, ao desenvolvimento de sistemas computacionais inteligentes. Porém, ainda que de fato exista uma grande simbiose entre as tecnologias e modelos de ambos os domínios, ao ponto de serem utilizadas as nomenclaturas, por vezes, como se fossem sinônimos, há que se destacar que existem diferenças conceituais significativas entre elas.

A Inteligência Artificial é uma área multidisciplinar da ciência, com aplicações em diversos campos teóricos e práticos, estando voltada principalmente ao desenvolvimento de sistemas com a capacidade de processar dados ou informações do meio em que se encontram e, baseados nesta percepção ambiental, executar um conjunto de ações que melhor se adéquam no sentido de atingir um conjunto de resultados previamente esperados (POOLE; MACKWORTH; GOEBEL, 1998). Então, os chamados "sistemas inteligentes" são aqueles capazes de tomar decisões com base no seu próprio discernimento, algo à semelhança do processo de racionalização para tomada de decisões encontrado no pensamento humano.

As técnicas de Inteligência Artificial são tradicionalmente divididas em duas categorias: as simbólicas e as conexionistas. O paradigma simbólico (ou clássico), muito popular até os anos 1980, funciona incorporando lógica de predicados baseada em símbolos e regras que representam o conhecimento humano sobre um determinado problema, permitindo que o algoritmo estabeleça uma série de raciocínios lógicos semelhantes à linguagem. As representações simbólicas são de caráter proposicional e definem a existência de relações entre os objetos, enquanto o "raciocínio" desenvolve novas relações lógicas apoiadas por um conjunto de regras de inferência (GARNELO; SHANAHAN, 2019). Note a semelhança com o processo de formação do raciocínio humano, que relaciona objetos e conceitos abstratos e, a partir desse conhecimento adquirido, cria regras de associação para generalizar quando exposto a novos cenários.

Uma vantagem da Inteligência Artificial simbólica está no fato de ser auto-explicativa (interpretável), ou seja, é possível extrair elementos explicativos sobre o processo racional que levou às decisões do modelo (ADADI; BERRADA, 2018). Porém, uma importante limitação deste paradigma reside na necessidade de definir explicitamente todo o conhecimento necessário, com os elementos representacionais sendo formalizados manualmente

em vez de adquiridos a partir de dados, por exemplo (HARNAD, 1990). Isso torna o desenvolvimento de modelos simbólicos um processo custoso, resultando em sistemas, em geral, específicos ao domínio (baixa capacidade de generalização). Essa limitação faz com que a Inteligência Artificial simbólica seja vista, atualmente, como obsoleta.

Já o paradigma conexionista surgiu em 1959 com o conceito de Aprendizado de Máquina, quando Arthur Samuel o definiu como "um campo de estudo que dá aos computadores a habilidade de aprender sem terem sido programados para tal" (SAMUEL, 1959). O Aprendizado de Máquina deriva da Inteligência Artificial, voltando-se à pesquisa de modelagens computacionais capazes de adquirir novos conhecimentos, novas habilidades e novos modos de organizar o conhecimento existente (CARBONELL; MICHALSKI; MITCHELL, 1983). Formalmente, o Aprendizado de Máquina pode ser definido como uma coleção de técnicas que permitem aos computadores automatizarem a construção e a programação de modelos por meio da descoberta e generalização sobre padrões estatisticamente significativos nos dados disponíveis (BHAVSAR *et al.*, 2017). Em outras palavras, o Aprendizado de Máquina desenvolve sistemas com a capacidade de aprender tarefas com base em modelos de treinamento gerados por meio de dados ou experiências anteriores, se adaptando a novas entradas para fazer predições de modo similar ao humano.

Baseado nas definições acima, é possível afirmar que todo modelo de aprendizado de máquina é uma Inteligência Artificial, mas esta abrange um escopo mais amplo de técnicas, isto é, nem toda aplicação de Inteligência Artificial pode ser classificado como Aprendizado de Máquina. Mesmo que o estudo de algoritmos de aprendizado de máquina tenha iniciado décadas atrás, muitas de suas contribuições impactantes são relativamente recentes, com um intenso desenvolvimento de novos algoritmos ocorrendo especialmente após meados de 2012. O interesse em desenvolver modelos mais sofisticados esteve em segundo plano durante algum tempo, devido ao fato dessa classe de modelos exigir recursos de *hardware* capazes de suportar operações intensas de processamento sobre grandes volumes de dados, dentro de tempos factíveis, algo que não estava disponível antes do advento das atuais tecnologias computacionais de alto desempenho, como as GPUs. Contornadas essas limitações, os modelos de classificação e predição baseados em Aprendizado de Máquina têm rapidamente evoluído e se tornado parte integral de várias aplicações cotidianas.

Um dos pontos fortes do Aprendizado de Máquina está na sua grande capacidade de generalização, com alguns modelos sendo capazes de identificar padrões em dados de alta dimensionalidade com pouca ou mesmo nenhuma intervenção humana (GARNELO; SHANAHAN, 2019). As abordagens de Aprendizado de Máquina podem ser categorizadas de acordo com o tipo de aprendizado que utilizam, existindo vários paradigmas na literatura. Entretanto, uma análise detalhada sobre as características de cada um dos paradigmas de treinamento foge ao escopo deste trabalho, uma vez que esta pesquisa se concentra

em construir uma abordagem XAI dedicada aos modelos de aprendizado supervisionado que, pode-se dizer, é o paradigma que atualmente concentra a maior parte dos avanços em Aprendizado de Máquina (DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018).

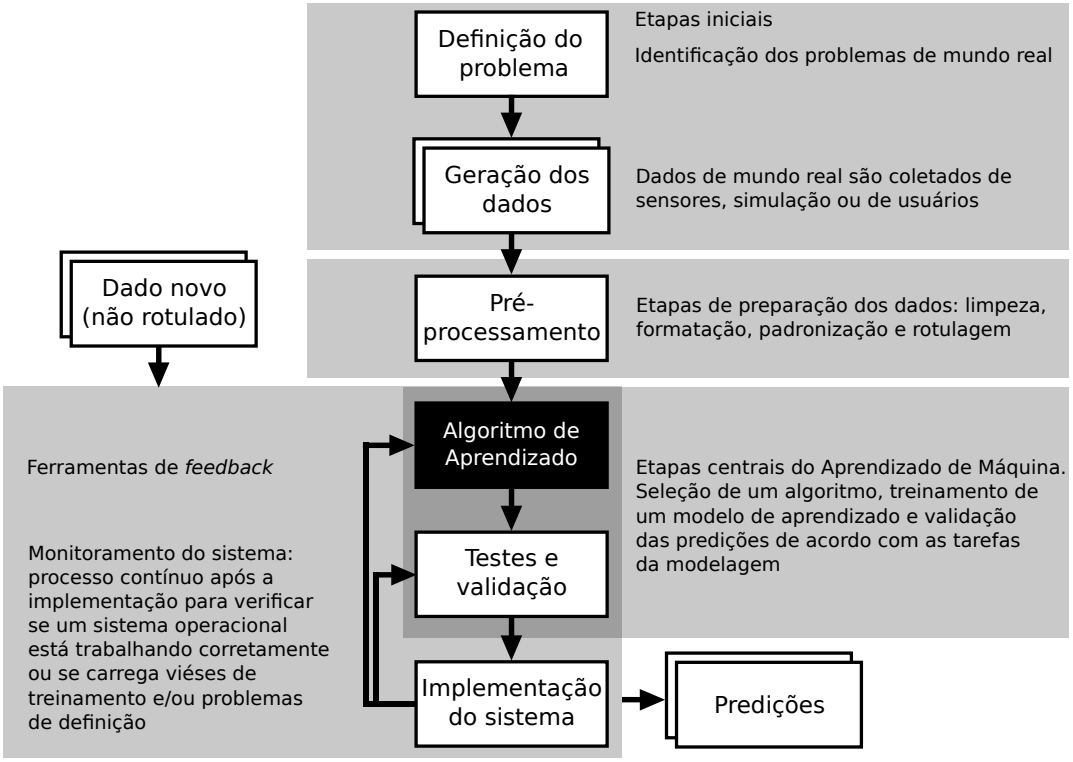
Entende-se por aprendizado supervisionado, o treinamento de um mapeamento generalizado baseado em dados nos quais cada uma das instâncias de entrada foi previamente analisada e rotulada em uma saída desejada. Mais especificamente, o conjunto de dados utilizado em procedimentos de aprendizado supervisionado é um par $(\mathbf{X}, \mathbf{Y})$, com $\mathbf{X}$ seguindo a Equação 2.1, ao passo que $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m\}$ representa o conjunto que define os respectivos mapeamentos de cada entrada $\mathbf{x}_k \in \mathbf{X}$, em que $\mathbf{y}_k \in \mathbb{R}^c$ para cada k. Com isso, o objetivo do processo de aprendizado é encontrar um modelo matemático que minimize uma função de perda, aplicada sobre a diferença (ou divergência) entre todos os valores preditos pelo modelo e os valores reais (BHAVSAR *et al.*, 2017). A função de perda quantifica o quão distante a predição está do valor real para uma dada instância, ou seja, quanto mais precisas as predições, menores serão as saídas da função de perda.

A Figura 2.1 ilustra as etapas de uma modelagem de aprendizado, desde a definição do problema a ser tratado até a validação e implantação do modelo em si. Como pode ser observado, o aprendizado supervisionado depende da disponibilidade de dados históricos conhecidos (previamente rotulados), tanto em termos de quantidade como de qualidade. Na verdade, qualquer aplicação de aprendizado de máquina é dependente do modo como os problemas de mundo real foram definidos, com a coleta e o pré-processamento dos dados representando fatores que afetam significativamente na precisão e eficiência dos modelos em capturar padrões nos dados. Porém, é importante destacar que o modelo gerado não representa o mundo real, mas sim a realidade presente nos dados (BHAVSAR *et al.*, 2017).

Geralmente, não são utilizados os mesmos dados sobre os quais o modelo foi treinado durante o procedimento de testes e validação porque as mesmas regras do conjunto de treinamento podem não se aplicar a dados novos. Além do mais, o modelo pode memorizar todo o conjunto de treinamento, o que levaria a taxas de *performance* injustas. Como se espera bons desempenhos preditivos sobre dados novos, ou seja, a capacidade de generalizar, usualmente se exclui uma porção dos dados disponíveis exclusivamente para ser utilizada na avaliação do desempenho do modelo (o chamado conjunto de teste).

Com relação à natureza da rotulagem em cada entrada do conjunto de dados, uma tarefa de Aprendizado de Máquina pode ser categorizada como uma Regressão ou Classificação. Regressão é a tarefa de treinar uma função em que a saída pode ser um valor real ($\mathbb{R}$). Em outras palavras, a saída é definida como um vetor de valores reais. Já em problemas de classificação, a tarefa do modelo é aprender uma função a partir dos dados de entrada que leve a um conjunto finito de saídas, isto é, cada entrada é mapeada dentro de um conjunto finito de possibilidades.

Figura 2.1 – Fluxograma do processo de desenvolvimento e aplicação de modelagens de Aprendizado de Máquina.



Fonte: Adaptada de Ortigossa, Gonçalves e Nonato (2024).

Atualmente, existem diversos algoritmos de aprendizado, dos mais simples e interpretáveis aos mais complexos e menos interpretáveis (caixas-pretas), com cada um deles apresentando suas especificidades e capacidades de atuação. Nas próximas subseções, serão apresentados quatro das principais classes de modelos de aprendizado supervisionado (NASCIMENTO *et al.*, 2019). Observe que a comunidade de pesquisas em Aprendizado de Máquina é bastante produtiva e, por isso, os modelos discutidos estão em constante evolução, com novas versões, extensões e aprimoramentos sendo publicados a cada ano.

2.3.1 Regressão Logística

Considere um conjunto de dados arbitrário contendo uma variável objetivo, a qual está sujeita a erros de aferição e pode depender de uma ou mais variáveis dos dados. O processo de regressão descreve a natureza da dependência entre a variável objetivo e as variáveis de entrada, além de quantificar o erro ao determinar uma função, a princípio desconhecida, que mapeia as entradas ao objetivo (BHAVSAR *et al.*, 2017). A Regressão Logística é uma modelagem estatística que deriva da família dos Modelos Lineares Generalizados, sendo utilizada como classificador linear para predizer variáveis objetivo categóricas de natureza binária (embora seja possível trabalhar com variáveis mais complexas).

A Regressão Logística está entre os modelos de classificação supervisionados mais

simples e amplamente utilizados, nas mais diversas aplicações, desde as ciências sociais (ARRIETA *et al.*, 2020), medicina (LUNDBERG *et al.*, 2018) e finanças (LIPTON, 2018). Um dos motivos que explica o sucesso do modelo, além da relativa simplicidade de implantação, é o fato de possuir a desejável característica de ser passível de decomposição e simulação (ARRIETA *et al.*, 2020). Então, é possível explicar os resultados da Regressão Logística. De fato, o modelo é considerado "transparente", sendo utilizado para gerar explicações em conjunto com modelos mais complexos e precisos, além de também ser aplicada como técnica de *benchmark* (avaliação) de modelos (LUNDBERG *et al.*, 2020).

Entretanto, a Regressão Logística possui algumas limitações significativas. O modelo assume que os dados de entrada são linearmente separáveis, algo que dificilmente ocorre em situações de mundo real. Além disso, a alegada transparência do modelo não é direta e pode requerer técnicas de explicabilidade adicionais. Quando as variáveis de entrada são altamente correlacionadas e com relacionamentos complexos de se compreender, o modelo final pode se tornar longe de ser interpretável (ARRIETA *et al.*, 2020).

2.3.2 *Support Vector Machine*

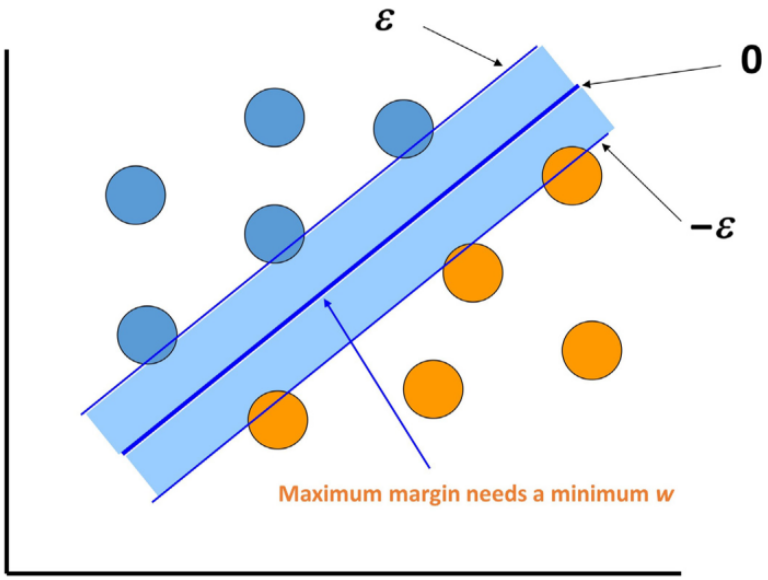
Support Vector Machines (SVMs) são modelos de aprendizado supervisionados de considerável presença na literatura, despontando entre os modelos mais utilizados devido às suas excelentes capacidades de predição e generalização (ARRIETA *et al.*, 2020; DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018). Formalmente, uma SVM classifica padrões de dados ao construir um hiperplano separador com máxima distância entre as instâncias de duas classes diferentes (VAPNIK, 1999). Em outras palavras, o modelo encontra o hiperplano que maximiza a margem de separação entre duas classes, separando-as em hiperespaços distintos, chamados de espaços de características. O hiperplano com a maior margem de separação será o que apresenta o menor erro de generalização do classificador (ARRIETA *et al.*, 2020). A Figura 2.2 apresenta o conceito de uma SVM simplificada.

Como frequentemente ocorre em cenários de mundo real, os dados podem não ser separáveis em duas classes por um hiperplano. No entanto, a SVM é conceitualmente fundamentado pela Teoria do Aprendizado Estatístico, que confere garantias teóricas de generalização para o modelo. Neste cenário, uma abordagem comum aplicada para contornar o problema citado é a transformação dos dados não separáveis para um espaço de maior dimensionalidade em que exista um hiperplano separador, utilizando o chamado *kernel trick* (COVER, 1965).

Além da não-trivialidade matemática em determinar o mapeamento correto dos dados para um espaço de maior dimensão, algo que pode restringir o uso da abordagem, SVMs apresentam limitações. SVMs têm dificuldades em lidar com dados contendo classes desbalanceadas. O algoritmo é de ordem quadrática (cúbica, no pior caso), logo, sofre com

a escalabilidade (tempo de processamento) ao treinar grandes conjuntos de dados, seja em termos de quantidade de instâncias ou dimensionalidade (BHAVSAR *et al.*, 2017).

Figura 2.2 – Representação de uma SVM. As amostras são classificadas e separadas em espaços de características.



Fonte: Bhavsar *et al.* (2017).

2.3.3 Redes Neurais Artificiais

A história das Redes Neurais Artificiais (RNAs) ou, do inglês, *Artificial Neural Networks* (ANNs), teve seu início na década de 1940, antes mesmo do surgimento do conceito de Aprendizado de Máquina, a partir do trabalho pioneiro de McCulloch e Pitts (1943). A ideia por trás de uma RNA é desenvolver uma rede de unidades de processamento que funcionem de modo similar ao sistema nervoso humano, numa composição com grupos de células simplificadas interligadas, os neurônios lógico-matemáticos.

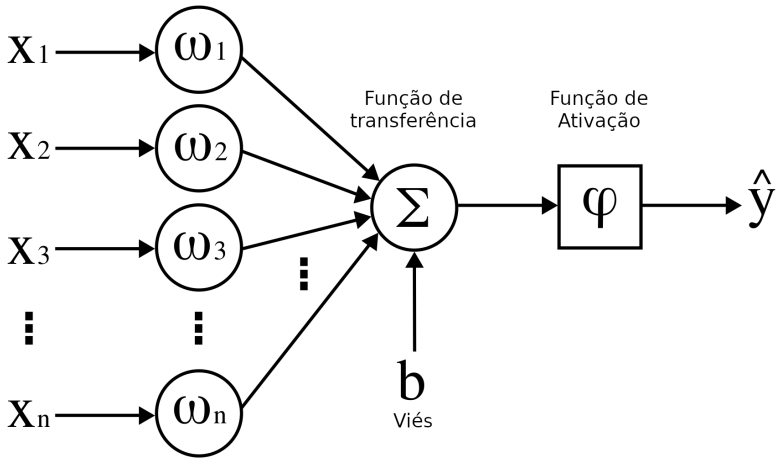
Por simplificação, imagine uma RNA com um único neurônio matemático, que também é chamado de Perceptron (ROSENBLATT, 1958). Essa unidade de processamento recebe dados por meio de um vetor de variáveis de entrada, que são ponderadas de acordo com as suas respectivas importâncias e enviadas até uma função de transferência, onde são acrescidas de um termo de viés. Por fim, o resultado é submetido a uma função de ativação de sinal do tipo *threshold* (limiar). A Figura 2.3 ilustra as estruturas básicas de uma rede contendo apenas um neurônio matemático do tipo Perceptron.

Formalmente, o neurônio matemático recebe um vetor de entrada, $\mathbf{x} = (x_1, \ldots, x_n)$, com cada uma das variáveis associadas a um vetor de pesos, $\omega = (\omega_1, \ldots, \omega_n)$. Pesos maiores conferem maior importância para a variável associada, isto é, mais expressivo será o sinal enviado ao neurônio a partir desta variável. Então, o conjunto de variáveis e pesos

é combinado e acrescentado a um termo de liberdade, o viés. Finalmente, o resultado será submetido à função de ativação φ, que pode ser vista como um formatador pré-definido, assumindo diversas formas a depender da natureza da variável objetivo, $\hat{\mathbf{y}}$. Com isso, é possível traduzir matematicamente a ilustração da Figura 2.3 pela equação:

$$\hat{\mathbf{y}} = \varphi\left(\sum_{i=1}^{n} \omega_i x_i + b\right). \tag{2.2}$$

Figura 2.3 – Representação do Perceptron, o neurônio aritmético baseado no modelo de McCulloch e Pitts (1943).



Fonte: Elaborada pelo autor.

O Perceptron nada mais é do que um hiperplano separador e, por isso, está limitado à classificação de dados linearmente separáveis, o que reduz consideravelmente a sua utilidade. Essa limitação pode ser superada ao se estruturar uma rede com múltiplos neurônios distribuídos em camadas (*layers*). Com isso, aumenta-se a capacidade de representação da rede por meio do aumento na complexidade dos hiperplanos separadores que dividem o espaço de aprendizado, que agora pode ser arranjado em diferentes níveis de abstração. A partir deste conceito, surgiram as redes *Multilayer Perceptron* (MLP). A Figura 2.4 apresenta a rede de neurônios de uma MLP simples com três camadas e três neurônios por camada. Porém, aqui é preciso esclarecer que, em geral, as camadas de entrada e de saída de uma MLP não são propriamente consideradas *layers*. Então, para ser preciso com a nomenclatura, a MLP da Figura 2.4 contém uma camada de entrada, uma *layer* oculta e uma camada de saída.

Observe que cada uma das unidades da *layer* oculta é uma estrutura neuronal completa, semelhante ao ilustrado na Figura 2.3, com entradas associadas a pesos enviando informações para serem processadas e encaminhadas para a *layer* seguinte (ou para a camada de saída). O diagrama de rede de neurônios (Figura 2.4) é uma representação tão comum que, por vezes, até se esquece que uma rede neural multicamadas é, na verdade, uma

função matemática. Formalmente, uma rede neural multicamadas f é uma composição de k funções multivariadas, $f : \mathbb{R}^n \to \mathbb{R}^p$, com n sendo a dimensionalidade da entrada, $\mathbf{x}$, e p a dimensão da camada de saída, $\hat{\mathbf{y}} = f(\mathbf{x})$, definida como:

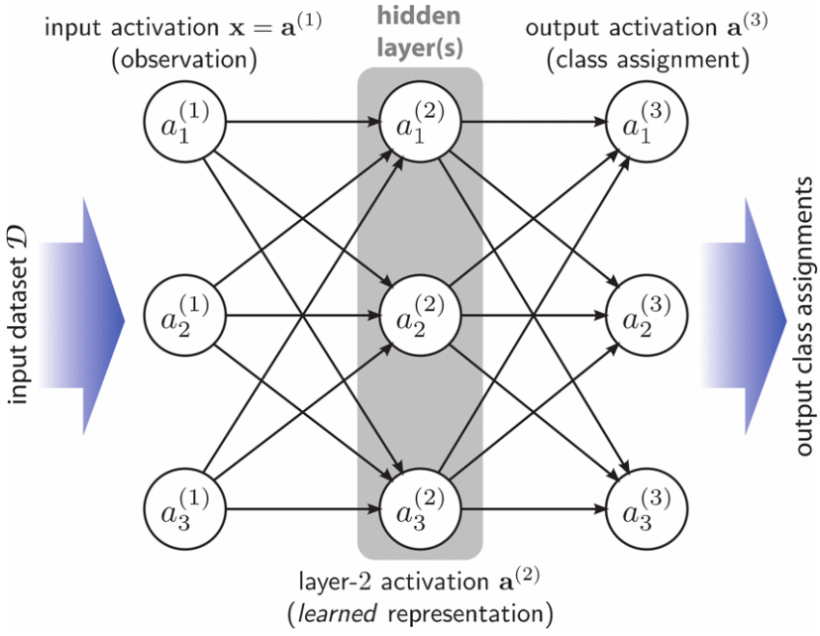
$$f(\mathbf{x}) = g \circ f_k \circ \ldots f_2 \circ f_1(\mathbf{x}) \tag{2.3}$$

em que cada função multivariada composta $f_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i-1}}$,

$$f_i(\mathbf{x}_i) = \varphi\left(\omega_i \mathbf{x}_{i-1} + b_i\right) \tag{2.4}$$

representa uma das k camadas intermediárias da rede neural (cf. Equação 2.2), com g a função de transformação de saída. A camada de saída retorna o resultado do modelo, logo, o número de neurônios e a transformação de saída dependerão do objetivo da modelagem. Em problemas de classificação binária há apenas um neurônio de saída com a função sigmoide ou tangente hiperbólica. Já em problemas multi-classe, o número de neurônios de saída é igual ao número de classes p, com a *softmax* sendo a função de transformação comumente aplicada (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Figura 2.4 – Rede de neurônios ilustrando uma MLP com três camadas.



Fonte: Rauber *et al.* (2016).

A função completa de uma rede neural costuma ser muito difícil de se ler, com a representação em rede de neurônios melhorando a legibilidade do modelo. Lembre-se que as ligações entre os neurônios configuram as entradas ponderadas de dados que cada neurônio receberá. Neste contexto, é uma tarefa consideravelmente complexa determinar valores adequados para que cada vetor de pesos associado a cada neurônio de uma MLP atribua a devida importância às variáveis de entrada que são de fato importantes para

a solução do problema. Este entrave fez com que o interesse em pesquisas na área das RNAs permanecesse em segundo plano por algum tempo, até o surgimento do algoritmo de aprendizado *Backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986).

No algoritmo de *Backpropagation*, o erro de aproximação entre o valor predito pelo modelo e o valor esperado é quantificado por uma função de perda (*Loss Function*). A escolha específica desta função depende da natureza da tarefa de aprendizado. O *Backpropagation* computa o gradiente da perda em relação aos pesos e vieses da rede, atualizando esses parâmetros na direção oposta do gradiente, buscando convergir para um erro mínimo, num processo iterativo conhecido por Gradiente Descendente, um dos algoritmos de otimização mais utilizados na minimização de funções de perda (RUDER, 2016). Aqui vale uma anotação. O termo "Gradiente Descendente" se trata, na verdade, de uma tradução abrasileirada um tanto imprecisa do original, *Gradient Descent*, pois, o gradiente em si não desce em coisa alguma, mas sim alguém desce pelo gradiente. Por isso, uma tradução mais adequada seria "Descida pelo Gradiente".

Mesmo com essa ressalva, neste documento será utilizada a nomenclatura convencionalmente aceita para o método, Gradiente Descendente. Esse processo iterativo adapta os vetores de pesos dos neurônios, resultando em um valor de erro recalculado. A partir dessa informação, o *Backpropagation* desencadeia uma série de iterações de otimização, ajustando os parâmetros da rede para minimizar a diferença entre as predições da rede e os valores esperados, de modo que o erro final seja próximo de zero ou de um limiar preestabelecido (WERBOS, 1990). A Equação 2.5 apresenta a formulação do *Backpropagation* com o erro quadrático como função de perda, para uma rede neural multicamadas:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\delta E^2}{\delta \omega_{ij}} \tag{2.5}$$

sendo t o instante de iteração, ou o passo de atualização do algoritmo, do i-ésimo peso do vetor associado ao neurônio j, $E^2 = (\mathbf{y} - \hat{\mathbf{y}})^2$ a divergência entre o valor esperado $\mathbf{y}$ e o valor calculado pela rede $\hat{\mathbf{y}}$, e $\eta > 0$ um fator de otimização.

Em resumo, o *Backpropagation* é responsável (durante o processo de treinamento) por compartilhar o erro calculado na saída da rede entre os neurônios das camadas intermediárias, partindo da camada de saída até alcançar a entrada do modelo, com cada neurônio "herdando" a contribuição entre a suas saídas na proporção de quanto cada neurônio contribuiu com a saída esperada (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017). Em outras palavras, o *Backpropagation* ajusta os parâmetros das Redes Neurais de modo que cada parâmetro é ajustado na proporção da sensitividade do erro geral da rede quanto a alterações nesses parâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016).

As redes de múltiplas camadas aliadas ao algoritmo de aprendizado *Backpropagation*

foram responsáveis pelo forte ressurgimento do interesse de desenvolvimento em RNAs, com pesquisas indicando que MLPs de duas camadas são, na verdade, modelos aproximadores universais (CSÁJI *et al.*, 2001). Com a expressiva melhoria na capacidade de processamento dos computadores atuais e a disponibilidade de grandes bases de dados, tem ocorrido uma verdadeira disputa entre os desenvolvedores de RNAs, surgindo novas arquiteturas cada vez mais profundas (quantidade de camadas) compostas por grupos de células com diferentes graus de complexidade (CANTAREIRA; ETEMAD; PAULOVICH, 2020).

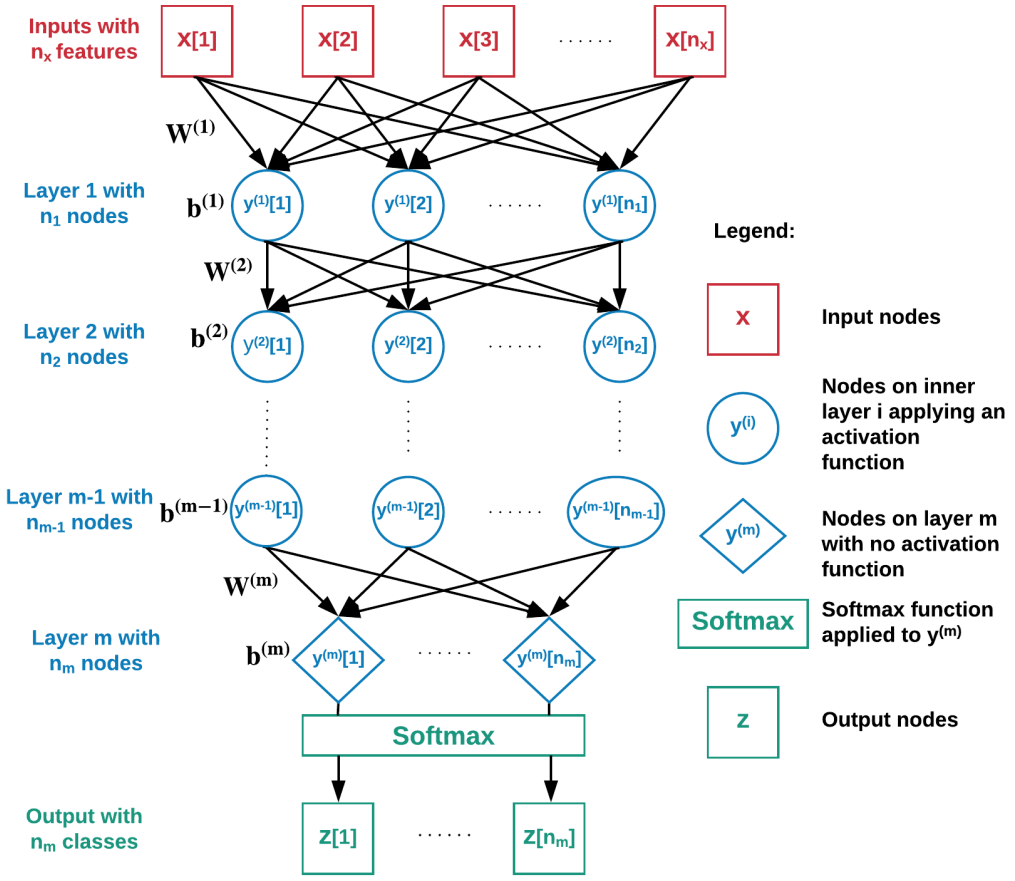
Essas novas redes com grandes quantidades de camadas e, por vezes, milhares de parâmetros, criaram um ponto de disrupção no *Machine Learning*, sendo natural o surgimento de uma subárea específica para as pesquisas de modelos de aprendizado profundo, o *Deep Learning* (LECUN; BENGIO; HINTON, 2015). As redes DNN (*Deep Neural Networks*), por exemplo, possuem estruturas de camadas hierarquizadas com grupos neuronais simples, destinados a extração de características básicas, e neurônios mais complexos, especializados na detecção de representações abstratas dos dados (CANTAREIRA; ETEMAD; PAULOVICH, 2020).

Redes profundas convolucionais do tipo CNN (*Convolutional Neural Network*) são consideradas o estado da arte em aplicações de visão computacional e reconhecimento de padrões em imagens (ARRIETA *et al.*, 2020; LECUN; BENGIO; HINTON, 2015). De fato, as redes CNN têm produzido significativos avanços em relação às abordagens anteriores dentro do Aprendizado de Máquina, devido às suas capacidades de tratar propriedades geométricas dos dados, ao incorporar análise multiescala e invariância (ou equivariância) quanto a desvios de alinhamento nas entradas do modelo (GEIFMAN *et al.*, 2022). Já as redes do tipo RNN (*Recurrent Neural Network*) e LSTM (*Long Short-Term Memory*), contornam o problema da "falta de memória" do gradiente descendente e por isso têm sido utilizadas com sucesso em aplicações que demandam contexto semântico, como processamento de linguagens e análise de séries temporais (ARRIETA *et al.*, 2020). A Figura 2.5 ilustra os componentes estruturais de uma DNN.

Entretanto, existem limitações. A espessa formulação matemática desses modelos lhes confere altas taxas de acurácia, mas acaba ironicamente minando a interpretabilidade e, por consequência, a sua confiabilidade (TJOA; GUAN, 2020). A segurança também é um ponto sensível, com pesquisas demonstrando a fragilidade das Redes Neurais quanto a ataques adversariais (GOODFELLOW; SHLENS; SZEGEDY, 2014). Além do mais, embora as RNAs com aprendizado baseado em *Backpropagation* se inspirem no cérebro humano, elas não operam no mesmo nível fundamental de similaridade, sendo que nem ao menos existem garantias de que tal modelo exista (TJOA; GUAN, 2020).

As RNAs se transformaram em modelos profundos, com redes verdadeiramente complexas e sofisticadas. Entender o que acontece com os dados dentro desses sistemas intrincados

Figura 2.5 – Protótipo de uma rede neural profunda com n_x entradas, m camadas e n_m saídas.



Fonte: Yousefzadeh e O'Leary (2019a).

é de fundamental importância para melhorar sua eficiência e confiabilidade, promovendo, inclusive, a correção de falhas e deficiências (CANTAREIRA; ETEMAD; PAULOVICH, 2020; ARRIETA *et al.*, 2020). Note que o universo das RNAs é vasto, compreendendo desde simples redes MLPs até redes neuromórficas com milhões de neurônios (INTEL, 2020). O leitor interessado em maiores detalhes sobre os diversos modelos de Redes Neurais pode consultar Asimov (2016) e LeCun, Bengio e Hinton (2015).

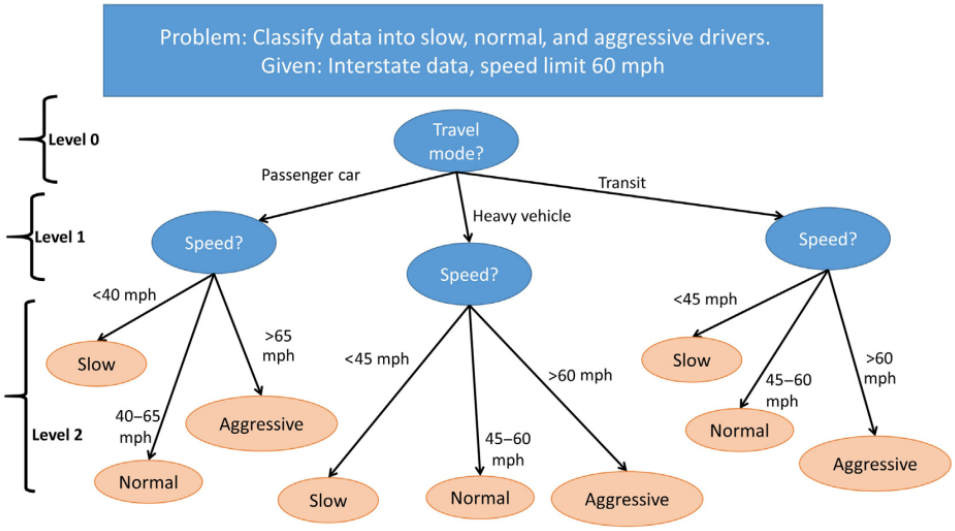
2.3.4 *Random Forests*

O *ensemble learning* (aprendizado em conjunto) é um conceito de Aprendizado de Máquina supervisionado que combina as decisões de múltiplos modelos base, independentes entre si, para gerar um novo modelo preditivo otimizado em que o resultado final é determinado por sistemas de votação. Atualmente, os algoritmos baseados em conjuntos de árvores (*tree ensembles*) são considerados modelos de elevada precisão que podem ser aplicados dentro dos mais variados contextos (ARRIETA *et al.*, 2020). Seu desenvolvimento deriva de eficientes melhorias nas Árvores de Decisão, que são modelos de classificação não-paramétricos com estrutura topológica similar às ramificações de uma árvore.

Árvores de decisão implementam algoritmos que operam apenas instruções condicionais. O modelo começa com um nó raiz contendo uma questão inicial sobre o problema a ser resolvido. Caso essa questão se divida em múltiplas opções de resposta, novos nós são criados ligando a raiz às possíveis soluções. Cada solução é examinada e, caso necessário, novos níveis de decisão são criados. Esse processo de ramificação continua até que todas as possíveis soluções tenham sido verificadas, criando uma estrutura que leva a um fluxo lógico de decisões (BHAVSAR *et al.*, 2017).

Entre os benefícios das árvores de decisão, é possível citar a relativa simplicidade de construção dos modelos, além do fato de serem considerados modelos interpretáveis (BHAVSAR *et al.*, 2017; DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018; DOSHI-VELEZ; KIM, 2017), como pode ser observado na Figura 2.6, que ilustra o processo de classificação de motoristas utilizando uma árvore de decisão. Porém, árvores de decisão únicas costumam apresentar baixa capacidade de generalização, além de serem propensas ao *overfitting* (ARRIETA *et al.*, 2020). O *overfitting* (sobreajuste em relação aos padrões de um recorte dos dados) é um sério problema em Aprendizado de Máquina, ocorrendo quando o modelo aprende bem demais as características dos dados de treinamento, incluindo o ruído, o que tende a gerar preditores menos efetivos (ou muito errados) quando aplicados em dados desconhecidos.

Figura 2.6 – Representação de uma Árvore de Decisão aplicada na classificação do comportamento de motoristas baseado em limiares de velocidade.

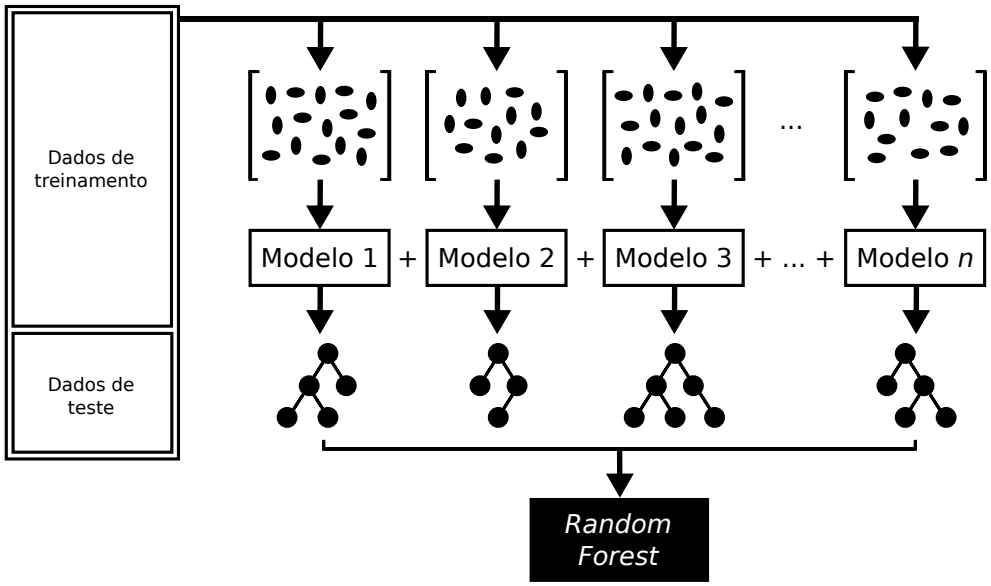


Fonte: Bhavsar *et al.* (2017).

Para contornar essas limitações, os *tree ensembles* combinam diferentes árvores e, em seguida, obtém uma predição agregada de todas as árvores. Cada árvore é gerada utilizando uma amostra aleatória dos dados de treinamento, com repetição de atributos (HO, 1995). A quantidade de instâncias ou atributos em uma amostragem pode ser grande ou pequena. Amostras semelhantes podem ser extraídas repetidamente em momentos diferentes. Todo

esse processo de amostragem e construção de modelos, conhecido por Bagging (*Bootstrap AGGregatING*) (BREIMAN, 1996; GHOJOGH; CROWLEY, 2019), é feito de modo independente e simultâneo, como ilustrado na Figura 2.7. Tem-se então um conjunto de árvores de decisão geradas aleatoriamente, ou seja, uma floresta aleatória, que dá origem ao nome: *Random Forest* (BREIMAN, 2001). A topologia final de classificação do modelo é construída a partir da média das classificações de todas as árvores ou por meio de um sistema de votação, algo que equilibra a tendência ao *overfitting* de árvores de decisão únicas. Esta abordagem também reduz a variância do modelo sem aumentar o viés de incerteza, o que aprimora o desempenho preditivo (MURPHY, 2012).

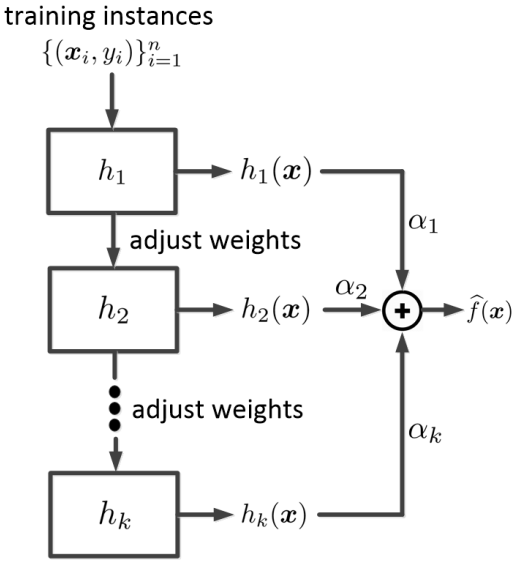
Figura 2.7 – Construção de um classificador *Random Forest*.



Fonte: Elaborada pelo autor.

Novas abordagens têm surgido e apresentado significativas melhorias no desempenho preditivo do *Random Forest*, graças ao desenvolvimento de otimizações que incluem a aplicação do *Gradient Boosting* (CHEN; GUESTRIN, 2016). Em resumo, o *Boosting* é um meta-algoritmo que se baseia no conceito de *ensemble learning* sequencial, podendo ser utilizado com praticamente qualquer modelo de aprendizado (GHOJOGH; CROWLEY, 2019). Em vez de treinar múltiplos modelos independentes, como é tradicionalmente feito no *ensemble learning*, o *Boosting* gera uma sequência hierárquica de modelos, com cada novo modelo sendo treinado para corrigir os erros residuais do anterior. De modo iterativo, o resultado do modelo anterior é utilizado como parte da entrada do próximo, com o resultado final sendo gerado a partir de uma agregação (média ponderada) entre as predições corretas e incorretas. Logo, o *Gradient Boosting* é uma técnica baseada no *Boosting* em que os elementos ponderadores são ajustados a cada iteração utilizando o algoritmo de Gradiente Descendente. A Figura 2.8 descreve o procedimento de *Boosting*.

Figura 2.8 – Treinamento de um preditor $\hat{f}$ com *Boosting* em k modelos intermediários.



Fonte: Ghojogh e Crowley (2019).

Devido às suas características positivas, o *Random Forest* tem sido aplicado nos mais diversos contextos, como medicina, finanças, gerenciamento de cadeias de suprimentos, entre outros (LUNDBERG *et al.*, 2020; MAIER; HANDELS, 2016). Porém, a combinação de modelos acaba reduzindo a interpretabilidade das abordagens do tipo *ensemble*, com o *Random Forest* sendo considerado "opaco" ao ponto de sua complexidade impedir o entendimento da lógica por trás de suas predições (DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018).

2.4 O Que Define Algo Complexo?

Em Ciências de Computação, existem diversas métricas utilizadas para definir complexidade. A Teoria da Complexidade Computacional é uma área da computação teórica que possui vasta literatura em definição e classificação de problemas computacionais, desde os mais simples e tratáveis, que podem ser otimizados e resolvidos de modo mais eficiente em relação às soluções já existentes, até aqueles mais complexos, que demandam estudos aprofundados no domínio da aplicação para o desenvolvimento de novas ferramentas que sejam capazes de resolver (ou aproximar) tarefas que exigem recursos significativos. Existem problemas cuja solução ainda não foi encontrada e também aqueles considerados tão complexos que, ainda que possam ser teoricamente solucionados, na prática, sua resolução é inviável com os atuais recursos, isto é, são intratáveis.

Neste contexto, a complexidade algorítmica é uma linha de pesquisas bem conhecida que se concentra em definir o quão rápido ou custoso um algoritmo em particular executa ao solucionar uma determinada tarefa. Entretanto, esta pesquisa não se concentra no tempo como uma medida de complexidade. Além disso, uma análise detalhada sobre os

fundamentos da complexidade está fora do escopo deste trabalho, mas pode ser encontrada em Goldreich (2008) e Arora e Barak (2009).

Porém, a noção de complexidade já foi utilizada diversas vezes e ainda será visitada outras mais ao longo desta leitura, o que levanta a necessidade de algumas definições. Específico ao contexto dos algoritmos de aprendizado de máquina, quais são os critérios que definem o que é simples e o que é complexo? A resposta para esta pergunta claramente depende de qual aspecto da modelagem está sob análise.

2.4.1 Viés e Variância

Viés e variância são dois temas correlacionados de extenso debate em Aprendizado de Máquina, e não seria possível passar por esta seção sem uma discussão sobre o assunto. Considere uma típica tarefa de aprendizado supervisionado de predizer (estimar) uma variável $\mathbf{y} \in \mathbf{Y}$ a partir de uma entrada n-dimensional $\mathbf{x} \in \mathbf{X}$. Existe uma função f que captura os verdadeiros relacionamentos entre ambas as variáveis, ou seja, $\mathbf{y} = f(\mathbf{x}) + \epsilon$, com ϵ sendo uma parte de $\mathbf{y}$ que não pode ser estimada a partir de $\mathbf{x}$. Neste contexto, o objetivo da tarefa de aprendizado é determinar um modelo estimador $\hat{f}$ que aproxime o comportamento de f, tal que este modelo descreva o relacionamento entre a entrada (variáveis explicativas ou preditivas) e a saída (variáveis dependentes ou objetivo). Um bom estimador é aquele cujo resultado é o mais próximo possível do verdadeiro processo que gerou os dados, a princípio desconhecido. No cenário ideal, o modelo estimador seria treinado sobre dados ilimitados até que fosse possível aprender os seus padrões preditivos tão bem que o erro de estimativa tenderia a zero.

Porém, no mundo real, trabalha-se com conjuntos de treinamento de tamanho limitado e, além disso, toda fonte geradora de dados envolve uma mistura de componentes regulares (repetíveis) e estocásticos (BRISCOE; FELDMAN, 2011). Mais especificamente, o objetivo de uma modelagem de aprendizado é treinar um estimador com os dados disponíveis, tal que este desenvolva a habilidade de generalização, buscando maximizar a acurácia para predições futuras quando o modelo for exposto a dados novos e desconhecidos. Tecnicamente, "dados novos" são os dados que não foram utilizados na etapa de treinamento do modelo (FORTMANN-ROE, 2012a). Mas essa acurácia não é maximizada simplesmente aprendendo as características dos dados de treinamento o mais precisamente possível (BRISCOE; FELDMAN, 2011).

Como já introduzido, um ajuste excessivo (*overfitting*) sobre os dados de treinamento tende a capturar aspectos aleatórios da amostragem (que não serão repetidos) como se fossem elementos regulares e, com isso, perdendo padrões mais amplos. Por outro lado, quando o modelo é treinado de modo demasiado generalista (*underfitting*), o ajuste a dados novos tende a considerar menos os efeitos aleatórios, mas ao custo de também desconsiderar

componentes regulares (GHOJOGH; CROWLEY, 2019; BRISCOE; FELDMAN, 2011). Logo, deve haver um equilíbrio no ajuste (*fitting*) de treinamento para que seja possível aprender os verdadeiros padrões e desconsiderar o ruído, minimizando o erro de estimativa.

Segundo Neal *et al.* (2018), um dos possíveis modos de se medir a qualidade de um modelo preditor está em quantificar o seu erro total esperado, o que pode ser verificado pela seguinte expressão:

$$Err(\mathbf{x}) = \mathbb{E}\left[(\mathbf{Y} - \hat{f}(\mathbf{x}))^2\right] \tag{2.6}$$

em que se eleva a diferença ao quadrado, por questão de simetria, para mensurar o erro quadrático médio. O erro total esperado pode então ser decomposto em três componentes:

$$Err(\mathbf{x}) = \mathcal{E}_{\text{viés}} + \mathcal{E}_{\text{variância}} + \mathcal{E}_{\text{ruído}} \tag{2.7}$$

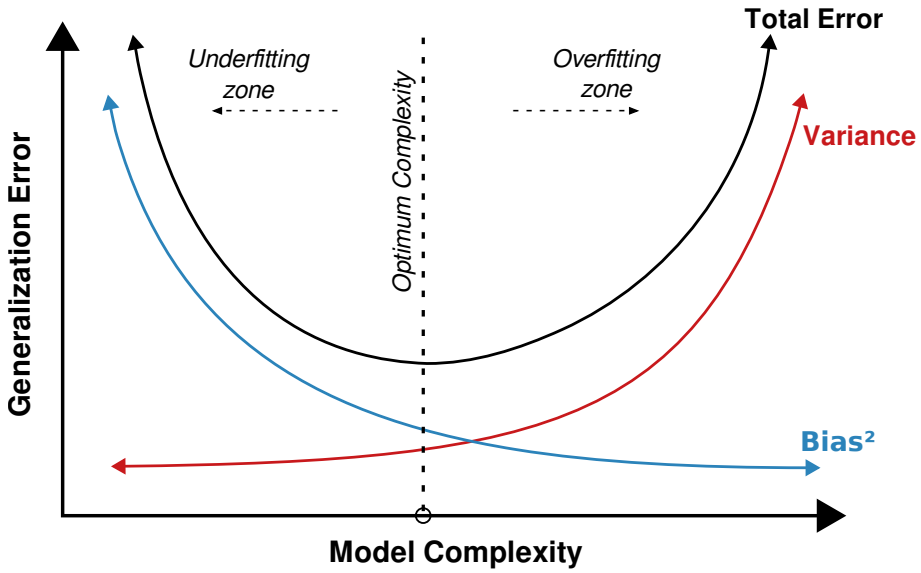
A demonstração completa desta decomposição é amplamente conhecida e está detalhada em Hastie, Tibshirani e Friedman (2009), Goodfellow, Bengio e Courville (2016) e Ghojogh e Crowley (2019). Observe que o erro total de aproximação de um preditor está em função do viés e da variância, acrescidos de um componente intangível, isto é, o ruído do verdadeiro relacionamento entre as variáveis preditivas (ϵ) que não pode ser fundamentalmente reduzido por qualquer modelo de aprendizado (FORTMANN-ROE, 2012b). Logo, para que o erro total seja o menor possível, é desejável que o estimador tenha baixo viés e também baixa variância.

A cada iteração do processo de treino, um novo modelo é gerado e, pela aleatoriedade dos dados, tem-se uma variedade de predições. Viés é o erro inerente ao modelo e reflete o quão distante as predições estão em relação à classe objetivo. O erro devido ao viés vem da diferença entre a predição esperada (ou média) do modelo estimador e o valor correto da variável sendo predita. Já a variância captura o quanto as predições desviam umas das outras. O erro devido à variância é visto como a sensibilidade do estimador a pequenas flutuações em função de uma amostra independente de dados (FORTMANN-ROE, 2012b).

Geman, Bienenstock e Doursat (1992) verificaram a inconsistência de convergência entre viés e a variância, afirmando que o custo em se reduzir o viés de um estimador é o aumento da variância. Assim, um modelo preditor deve, por meio do aprendizado, assumir um ponto no contínuo entre viés e variância. Segundo Briscoe e Feldman (2011), o parâmetro crítico que modula viés e variância é a complexidade das hipóteses assumidas pelo modelo. A medida mais comum para mensurar a complexidade de estimadores é a sua quantidade de parâmetros, pois, em geral, o número de parâmetros estabelece os graus de liberdade do modelo em relação aos dados de treinamento. Em outras palavras, hipóteses mais complexas (modelos com mais parâmetros) podem se ajustar melhor aos dados de treino (maior variância), enquanto as hipóteses menos complexas (menos parâmetros)

impõem uma forte expectativa (alto viés) nos dados, sacrificando o ajuste (BRISCOE; FELDMAN, 2011). A Figura 2.9 contrasta a relação entre viés e variância com o erro de generalização em função da complexidade dos modelos de aprendizado.

Figura 2.9 – A contribuição do viés e da variância para o erro em função da complexidade.



Fonte: Ortigossa, Gonçalves e Nonato (2024).

Ainda segundo Briscoe e Feldman (2011), conforme o modelo se torna mais complexo, a generalização melhora e o erro diminui até um ponto mínimo e, então, passa a aumentar. Em altas complexidades, o modelo entra em *overfitting* sobre os dados de treinamento e o desempenho preditivo para dados novos tende a sofrer. Acredita-se que a generalização para dados não vistos diminui em altas complexidades devido ao fenômeno da memorização, que acontece quando o modelo de aprendizado simplesmente "memoriza" os dados de treinamento. O fenômeno da memorização é um grande desafio para aplicações teóricas e práticas de *Machine Learning*, pois tem implicações sobre a compreensão da generalização, além de estar potencialmente ligado a aspectos negativos de privacidade, pois pode ser explorado em ataques para revelar dados de treinamento confidenciais (HAIM *et al.*, 2022).

O ponto ótimo de complexidade depende da natureza dos padrões a serem aprendidos, pois o perfil dos processos regulares e estocásticos varia dependendo da natureza da fonte geradora dos dados. Por isso, o equilíbrio (*tradeoff*) entre viés e variância é tido como uma compensação entre complexidade e ajuste de dados, ou seja, é uma medida da habilidade de generalização do modelo.

Para Geman, Bienenstock e Doursat (1992), o *tradeoff* entre viés e variância é universal, com esta perspectiva sendo um dos grandes dilemas do aprendizado. Entretanto, trabalhos recentes têm mostrado que é possível aumentar a complexidade dos modelos de aprendizado e reduzir o viés, mas sem aumentar o erro total, graças aos avanços no desenvolvimento de técnicas de regularização e otimização. Neal *et al.* (2018) apresentaram evidências de

que ambos, viés e variância, diminuem ao mesmo tempo que a complexidade aumenta em Redes Neurais modernas, contradizendo a intuição de equilíbrio estrito de Geman, Bienenstock e Doursat (1992). Os resultados de Neal *et al.* (2018) demonstram que a variância diminui em Redes Neurais grandes devido às otimizações, enquanto a variância de amostragem aumenta lentamente desde que a rede seja suficientemente parametrizada.

Zhang *et al.* (2021) indicaram que os modelos de aprendizado modernos tendem a ajustar os dados de treinamento perfeitamente enquanto também apresentam bons desempenhos sobre os dados de teste. Além disso, trabalhos recentes têm estudado o interessante fenômeno do "*benign overfitting*" (*overfitting* benigno), não restrito apenas aos modelos mais complexos (BARTLETT *et al.*, 2020; WANG; MUTHUKUMAR; THRAMPOULIDIS, 2021). Para um aprofundamento teórico sobre os mecanismos aplicados para lidar com o *tradeoff* entre viés e variância, Ghojogh e Crowley (2019) publicaram uma extensa pesquisa sobre métodos de regularização e otimização, além de definições e formulação de conceitos.

2.4.2 Não-linearidade

Especificamente dentro do contexto desta pesquisa, a complexidade é tomada como meio de refletir a quantidade e também os níveis de interação entre os parâmetros de um modelo de aprendizado. Neste sentido, a complexidade pode ser utilizada como uma indicação da transparência dos algoritmos de aprendizado, pois espera-se que um modelo simples seja transparente e, por consequência, interpretável, visto que este normalmente possui uma quantidade reduzida de parâmetros com poucos (ou nenhum) relacionamento(s) entre si. Então, estes parâmetros simplificados podem ser diretamente checados para avaliar os efeitos exercidos sobre cada uma das variáveis de entrada. Quando é possível obter essa informação facilmente, o modelo é dito transparente, não havendo a necessidade de aplicar uma técnica para gerar maiores explicações.

Por outro lado, quando o modelo possui grande quantidade de parâmetros e, além disso, estes se relacionam de modos sofisticados (não-lineares), é difícil obter essa visão direta a respeito dos efeitos causados. Logo, quanto mais parâmetros e quanto maiores os níveis de não-linearidade entre eles, menor a transparência, isto é, maior é a opacidade do modelo. Modelos em que a quantidade e o nível de não-linearidade é alto, dificilmente podem ser interpretados de modo direto.

Existem alguns algoritmos tradicionalmente considerados transparentes e interpretáveis, como os modelos lineares, as árvores de decisão e as listas de regras (*rule lists*). Os algoritmos derivados do paradigma simbólico, por definição, são (em tese) transparentes e dispensam a explicabilidade. Porém, ainda que fossem de fato totalmente transparentes, os algoritmos simbólicos têm alcance e capacidade de generalização limitados. Já os modelos lineares são vistos como abordagens simples, eficientes (em certas aplicações)

e interpretáveis, por não possuírem relacionamentos de não-linearidade entre os seus parâmetros, muito embora essa seja uma visão um tanto simplista e questionável, porque há casos em que mesmo um modelo linear pode ser difícil de interpretar. Observe:

$$\begin{aligned}\mathbf{y} = {} & 0.33x_1 + 2.5x_2 + 8.2x_3 - 4.81x_4 + 7.6x_5 + 1.84x_6 - 43.2x_7 + 9.1x_8 + \\ & 0.5x_9 - 0.1x_{10} + 6.4x_{11} - 3.6x_{12} + 2.4x_{13} + 2.6x_{14} - 6.3x_{15} + 1.9x_{16} + \\ & 4.8x_{17} + 0.25x_{18} - 16.7x_{19} + 4.28x_{20} - 12.1x_{21} + 5.13x_{22} + 9.01x_{23} + \\ & 1.8x_{24} - 8.5x_{25} + 4.33x_{26} - 7.6x_{27} + 11.4x_{28} + 0.99x_{29} - 7.8x_{30}.\end{aligned} \tag{2.8}$$

Esta é uma modelagem linear com trinta parâmetros. Pode ser considerado simples do ponto de vista matemático, mas é facilmente interpretável? A complexidade de inspeção aumenta conforme a quantidade de parâmetros aumenta, mas ainda assim é possível extrair o efeito de cada variável com apoio de métodos estatísticos e gráficos, como *Friedman H-statistic* (FRIEDMAN; POPESCU, 2008), *Dependence Plots* (FRIEDMAN, 2001) e *Individual Conditional Expectation plots* (GOLDSTEIN *et al.*, 2015). E mesmo com muitos parâmetros, pode haver poucos realmente importantes para influenciar na predição, o que facilitaria a tarefa de inspeção. Então, se é possível gerar predições e extrair informações de modelos simplificados, como os lineares, qual seria o motivo para empregar modelos tão complexos (não-lineares) como as Redes Neurais ou *ensembles*? A resposta para essa questão não vem meramente do capricho dos desenvolvedores, mas sim dos dados.

As fontes de informações atuais são grandes conjuntos de dados multidimensionais, que podem conter quantidades arbitrárias de variáveis. Muitas dessas variáveis podem apresentar relacionamentos de dependência entre si, seguindo as mais diferentes razões não-lineares (relações quadráticas, exponenciais, entre outras). Esse tipo de interação não-linear não pode ser capturado com precisão por um modelo linear. Isso significa que, quando operando em dados com relacionamentos de interdependência complexos não-lineares e, a princípio, desconhecidos pelos analistas, modelos mais simples, como a regressão linear, apresentam deficiências quanto às suas capacidades de generalização para "desdobrar" as intrincadas correlações entre as variáveis.

Algoritmos não-lineares como as Redes Neurais ou o *Random Forest*, por exemplo, podem modelar relacionamentos não-lineares com relativa facilidade. É natural esperar que dados permeados com interações complexas demandem soluções sofisticadas para a tarefa de descoberta dos seus padrões. Os modelos aqui chamados de "complexos" foram desenvolvidos para trabalhar com esse tipo de modelagem de relacionamentos sofisticada, buscando soluções dentro de contextos que dificilmente seriam enfrentados com ferramentas menos complexas. Mas toda essa sofisticação adicional vem ao custo da opacidade. Enquanto uma transformação linear pode ser interpretada por meio da verificação dos

pesos associados às variáveis de entrada, múltiplas camadas com interações não-lineares em cada camada implicam em estruturas complicadas de compreender, demandando ferramentas para obter explicações acerca dos seus resultados (ADADI; BERRADA, 2018).

Ainda que os modelos lineares fossem totalmente transparentes, eles sofrem com a baixa acurácia geral quando comparados com técnicas não-lineares mais sofisticadas e precisas. Como bem pontuado por Arrieta *et al.* (2020), existem exceções quando os dados são "bem-comportados" e, nestes cenários, é possível ter modelos simples e precisos. Porém, são poucas as aplicações reais que trabalham com dados controlados, com os modelos complexos sendo mais vantajosos ao oferecerem maior flexibilidade de aproximação.

2.5 *EXplainable Artificial Intelligence* (XAI)

Segundo o grande teórico comportamental e Prêmio Nobel de Economia, Daniel Kahneman, onde quer que exista julgamento humano, existe ruído (KAHNEMAN; SIBONY; SUNSTEIN, 2021). Em outras palavras, os seres humanos são suscetíveis ao ruído e aos mais diversos vieses ao fazerem suas escolhas. Analistas financeiros profissionais podem elaborar previsões de mercado contrárias, juízes podem proferir sentenças diferentes para o mesmo crime, médicos podem fazer diagnósticos distintos para pacientes com o mesmo problema. Agora, quais foram os fatores que influenciaram essas decisões, talvez o dia da semana ou o horário em que estas foram tomadas? Ainda segundo Kahneman, Sibony e Sunstein (2021), esses são exemplos de ruído que podem levar à variabilidade em julgamentos que deveriam ser idênticos. Entretanto, é possível confrontar as ações humanas em busca do processo racional que levou uma pessoa a tomar uma determinada decisão e, com isso, identificar dentro do conjunto de variáveis consideradas, aquilo que é de fato importante e o que é ruído. Agora imagine que a decisão tenha sido executada com base em um modelo de aprendizado complexo. Em muitos cenários, o ruído pode ter impactos prejudiciais que não devem ser ignorados. Mas como interpretar os processos deste modelo "caixa-preta"? Como explicar as suas principais influências?

2.5.1 Definições Conceituais

Um dos primeiros desafios encontrados ao se aprofundar na literatura XAI está na falta de uma terminologia comumente aceita. Existem muitos termos diferentes que são frequentemente utilizados intercambiavelmente e sem uma definição clara. Amann *et al.* (2022) discutem a necessidade de harmonizar a terminologia em XAI, pois a consequência direta dessa falta de definição é que toda nova publicação na área precisa definir em detalhes qual o significado dos termos que serão utilizados, o que acaba causando confusão e também uma inflação de definições em uso. Note que o XAI é uma área recente e com

influência de várias outras áreas do conhecimento, incluindo as ciências humanas, e alguns dos termos mais utilizados dentro do escopo XAI são abrangentes e com diferenças sutis.

Antes de prosseguir e para evitar enganos futuros, é necessário esclarecer alguns termos recorrentes até aqui, e que ainda serão utilizados com frequência. Após uma longa revisão na literatura, espera-se sintetizar definições esclarecedoras. Embora sejam, por vezes, utilizados de modo equivalente, *interpretability* (interpretabilidade) e *explainability* (explicabilidade) são conceitos que guardam algumas distinções. Interpretabilidade é um conceito geral um tanto elusivo que pode ter caracterizações diferentes (LIPTON, 2018). A palavra "interpretar" indica aquilo que pode ser explicado de modo compreensível (DOSHI-VELEZ; KIM, 2017). Um sistema interpretável é aquele em que os seus usuários conseguem observar, estudar e compreender como os dados de entrada são matematicamente mapeados (ADADI; BERRADA, 2018). Neste sentido, a interpretabilidade pode ser vista como uma característica passiva, indicando o quanto de sentido é extraído de um domínio com informações abstratas (ARRIETA *et al.*, 2020; DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018).

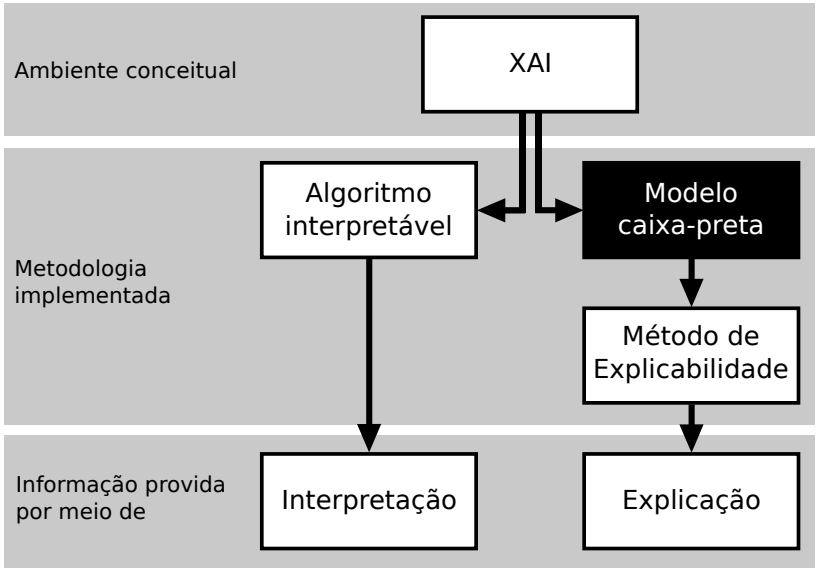
Já a explicação é, em termos psicológicos, "a moeda em que se trocam as crenças" (DOSHI-VELEZ; KIM, 2017; LOMBROZO, 2006), isto é, trata-se da comunicação daquilo que se compreendeu. A explicabilidade também é um conceito abrangente, mas é possível utilizar explicabilidade em XAI para se referir às informações adicionais que são geradas para verificar como um modelo de aprendizado chegou a um certo resultado (AMANN *et al.*, 2022). Logo, a explicabilidade é estabelecida como uma característica ativa, indicando a coleção de ações ou procedimentos realizados no sentido de esclarecer ou detalhar uma decisão de modelo (ARRIETA *et al.*, 2020; DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018). O que torna uma explicação melhor do que outra depende do contexto de trabalho e de quais perguntas se buscam respostas.

A Figura 2.10 ilustra a sutil, porém expressiva, diferença entre interpretabilidade e explicabilidade dentro do ambiente XAI. Algoritmos interpretáveis e explicativos têm o mesmo objetivo em comum, diferindo quanto às tecnologias aplicadas. Em resumo:

- **Algoritmos interpretáveis** são aqueles em que a sua lógica de trabalho pode ser inerente e intuitivamente compreendida por um humano;
- **Algoritmos de explicabilidade** são aqueles que tentam abrir as caixas-pretas *a posteriori*, gerando informações úteis sobre o comportamento de modelos não interpretáveis (AMANN *et al.*, 2022; ADADI; BERRADA, 2018).

Embora ambos os termos não sejam específicos ou restritos o suficiente para que se faça uma formalização em definitivo, assume-se neste trabalho que quando um modelo de aprendizado não for interpretável diretamente, então ele é passível de explicações que sejam, estas sim, compreensíveis.

Figura 2.10 – Diferença entre interpretabilidade e explicabilidade dentro do XAI.



Fonte: Adaptada de Ortigossa, Gonçalves e Nonato (2024).

Outros termos encontrados com frequência quando se discute XAI em Aprendizado de Máquina são: *comprehensibility* (compreensibilidade), *model transparency* (transparência do modelo) e *trust* (confiança). Compreensibilidade é descrito na literatura como um sinônimo de interpretabilidade, uma vez que se trata da capacidade do próprio modelo em expressar informações (ARRIETA *et al.*, 2020; GLEICHER, 2016). De modo semelhante, um modelo transparente é aquele passível de ser interpretado, ou seja, quando o modelo apresenta algum grau de interpretabilidade (LIPTON, 2018; ARRIETA *et al.*, 2020). Por fim, a confiança é também um termo com definição subjetiva, geralmente associado a um estado psicológico de segurança que, no contexto do Aprendizado de Máquina, tem sido expressa por meio de bons desempenhos preditivos dos modelos (avaliadas por métricas de *performance*). No entanto, este trabalho está aqui para mostrar que essa é uma visão simplista e que existem mais critérios de confiança para serem levados em conta, com as métricas de *performance* tradicionais, quando tomadas isoladamente, podendo levar a avaliações enganosas em certos contextos.

As principais métricas de *performance* para problemas de classificação são: *acurácia*, que mede a porcentagem de predições classificadas corretamente (positivas) em relação a todos os resultados preditos; *recall*, avalia o total de resultados de fato positivos entre os verdadeiros positivos e os falsos positivos; *precisão*, mede a taxa de predições positivas em função dos resultados positivos obtidos (corretos e incorretos); *F1 Score*, combina o *recall* e a precisão em uma média harmônica para verificar a tensão entre falsos positivos e falsos negativos; ROC (*Receiver Operating Characteristics*), contrasta a taxa de verdadeiros positivos com a taxa de falsos positivos e, em modo gráfico, indica melhor desempenho do modelo quanto mais próxima a curva ROC estiver do eixo dos valores positivos. O leitor

interessado pode encontrar descrições detalhadas sobre as métricas aplicadas para avaliar algoritmos de aprendizado em Gareth *et al.* (2017).

É de fundamental importância medir precisamente o erro de predição de um modelo para avaliar a sua qualidade. O principal objetivo de uma modelagem baseada em Aprendizado de Máquina é construir modelos que façam predições precisas sobre dados novos. Tecnicamente, "dado novo" é o dado não utilizado no treinamento. Logo, as métricas de *performance* utilizadas deveriam refletir o objetivo de modelagem, mas, na prática, em vez de reportar o erro do modelo sobre dados novos, as métricas tradicionais são aplicadas sobre os chamados conjuntos de teste, que usualmente vêm de uma porção do conjunto de dados original que foi separada dos dados aplicados no treinamento. Ou seja, os resultados das métricas refletem o passado (embora existam mecanismos de avaliação que consideram dados novos para monitorar a qualidade das predições e ajustar o modelo em tempo de execução). Note que qualquer modelo de aprendizado é naturalmente otimizado para descrever os padrões dos dados em que foi treinado. Por isso, as informações geradas pela metodologia comumente aplicada para medição de erros em modelagens de aprendizado podem ser enganosas e levar à seleção de modelos imprecisos e inferiores (FORTMANN-ROE, 2012a).

De acordo com Amann *et al.* (2022), a transparência é um dos principais requisitos para se estabelecer confiança em sistemas inteligentes. Em aplicações baseadas em modelos não interpretáveis (caixas-pretas), esforços devem ser feitos para incluir transparência, com a explicabilidade sendo uma ferramenta para alcançar a transparência.

2.5.2 Necessidades e Desafios da Explicabilidade

Normalmente, a literatura sobre Aprendizado de Máquina esteve "centrada no algoritmo", assumindo que os modelos desenvolvidos são obviamente interpretáveis, sem de fato verificar a sua interpretabilidade (TJOA; GUAN, 2020). O quadrinho da Figura 2.11 aborda essa questão de modo bem-humorado.

Por todo o exposto até aqui, é possível dizer que os sistemas de aprendizado de máquina, grosso modo, são construídos a partir de "pilhas" de álgebra linear e cálculo, como ferramentas matemáticas de modelagem, em que os dados coletados entram de um lado e as respostas saem do outro lado. E, no caso de resultados incorretos (que não atendam a critérios ou métricas de *performance*), basta "guiar" toda a pilha matemática até que as respostas comecem a "parecer" corretas. Embora essa seja uma visão cômica e um tanto satírica do processo de aprendizado, ela sutilmente levanta uma questão de grande importância para o estabelecimento de confiança na área e que esteve em segundo plano até então: sobre o resultado "parecer correto".

Modelos de *deep learning* ou *ensemble learning*, por exemplo, apresentam mecanismos

Figura 2.11 – O desenvolvimento tradicional do Aprendizado de Máquina, centrado no algoritmo (contém humor).

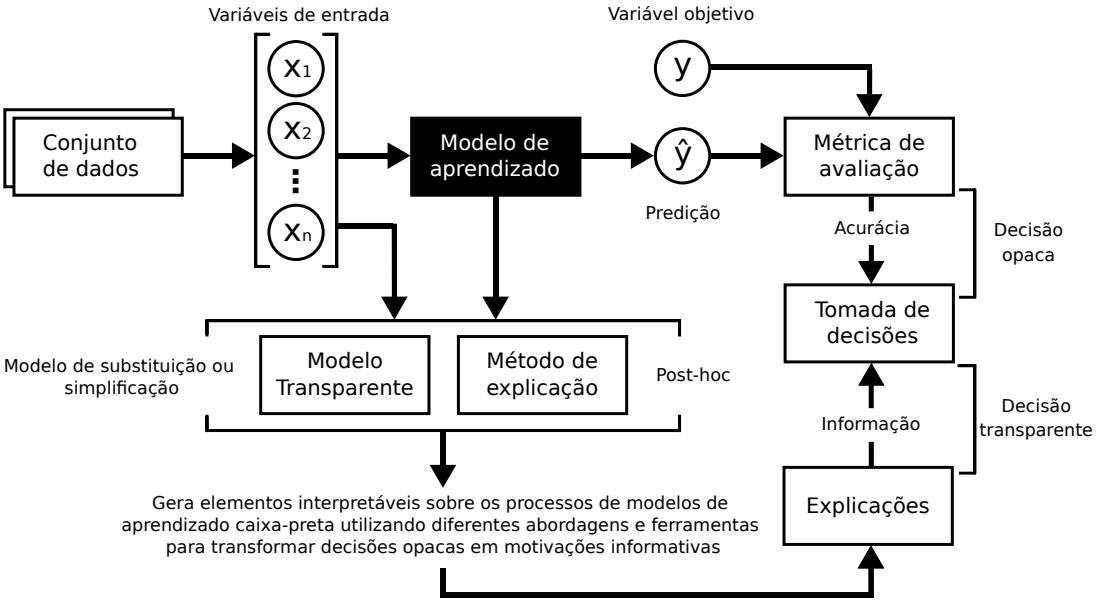
Fonte: Munroe (2010).

internos intrincados ou tão complexos ao ponto de ser impossível interpretar os motivos que levaram às suas decisões, o que acaba trazendo opacidade para o processo de verificação da lógica por trás das predições (RIBEIRO; SINGH; GUESTRIN, 2016c). Modelos opacos são "*black box*es" (caixas-pretas), isto é, quando se tem um contexto em que os dados entram de um lado e as predições simplesmente saem do outro lado, com os detalhes do processamento permanecendo obscuros ou desconhecidos. Tecnicamente, componentes caixa-preta são aqueles que não esclarecem sua lógica interna, dificultando a compreensão adequada de como chegaram em um dado resultado (ADADI; BERRADA, 2018). O entendimento do processo racional que leva esses modelos a tomar suas decisões têm sido primariamente ignorado (KARIMI; DERR; TANG, 2019).

É neste ponto que entra o *Explainable Artificial Intelligence*, ao levantar uma discussão sobre a necessidade em se compreender o funcionamento de modelos de aprendizado complexos, fomentando o desenvolvimento de ferramentas que sejam capazes de explicar aquilo que não pode ser facilmente interpretado. Quando sistemas baseados em aprendizado são aplicados na tomada de decisões sensíveis, não basta que os resultados pareçam corretos, eles devem estar corretos e, para que existam garantias de correção, é necessário haver mecanismos de checagem. A Figura 2.12 ilustra a temática.

Note que assumir que um algoritmo é obviamente interpretável nem sempre está errado ou é um problema. Em certos casos, a interpretabilidade pode não ser necessária, como em cenários em que não há consequências significativas que possam afetar a segurança dos usuários, quando não há a possibilidade de gerar injustiças a partir das decisões de um algoritmo, ou se o problema sob investigação já é de aceitação geral e foi suficientemente

Figura 2.12 – A explicabilidade se posiciona como um complemento do Aprendizado de Máquina, ao construir ferramentas capazes de facilitar a interpretação de modelos caixa-preta.



Fonte: Adaptada de Ortigossa, Gonçalves e Nonato (2024).

testado (TJOA; GUAN, 2020; DOSHI-VELEZ; KIM, 2017). Entretanto, o alcance das decisões tomadas por sistemas inteligentes baseados em aprendizado de máquina cresce a cada dia, não estando mais restritos aos ambientes teóricos e de pesquisas.

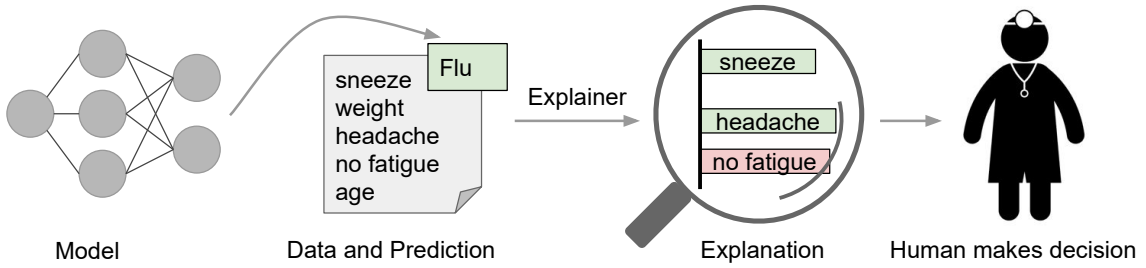
As novas arquiteturas de aprendizado têm alcançado desempenhos cada vez melhores nos mais variados domínios de aplicação (LECUN; BENGIO; HINTON, 2015; DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018; VASWANI *et al.*, 2017; WIEGREFFE; PINTER, 2019). Mas nem tudo é perfeito mesmo quando altas taxas de acurácia são alcançadas, com pesquisas demonstrando as fragilidades desses modelos. Recentemente, foi apontado viés étnico em um modelo aplicado na predição de reincidência criminal, e a exclusão de minorias em um *software* utilizado pela Amazon para determinar áreas nos Estados Unidos que receberiam ofertas de desconto (GUIDOTTI *et al.*, 2018b). Goodfellow, Shlens e Szegedy (2014) apontaram a suscetibilidade das RNAs a um tipo de ataque que tenta descobrir as mínimas mudanças que devem ser feitas nos dados de entrada para "enganar" a rede e causar classificações erradas, os chamados ataques adversariais.

Outro ponto importante que pode passar despercebido pelas métricas de avaliação comumente utilizadas, diz respeito a capacidade de generalização dos modelos de aprendizado. Por exemplo, Lapuschkin *et al.* (2019) apresentaram uma interessante pesquisa com casos em que predições feitas por modelos de aprendizado estavam baseadas em correlações espúrias que nada tinham a ver com o objetivo do aprendizado, conhecido como fenômeno de *Clever Hans*. Assim, quando decisões são tomadas por modelos complexos e difíceis de se interpretar, dentro de contextos sensíveis e que afetam a vida de pessoas, como a avaliação de *scores* de crédito bancário, administração pública e medicina, é importante

saber as razões por trás de tais decisões, verificando se os resultados do modelo estão de fato corretos e não apresentam algum tipo de viés espúrio ou ruído.

Considere um sistema de apoio ao diagnóstico médico em que, baseado em um modelo de classificação, um determinado paciente recebe a notícia de que está com gripe a partir de um histórico de sintomas descritos pelo próprio paciente e/ou verificados pela equipe de atendimento hospitalar. Nesta situação, seria interessante e bastante informativo para o médico se, além da predição, o sistema apontasse quais os sintomas que levaram àquela decisão. Com isso, o médico teria maior embasamento em seu diagnóstico em vez de simplesmente tomar uma decisão confiando em um resultado automático (RIBEIRO; SINGH; GUESTRIN, 2016c) A Figura 2.13 é bastante assertiva em ilustrar essa situação e também a importância em haver meios adequados para a explicabilidade.

Figura 2.13 – Explicando a predição de um diagnóstico de gripe. Os sintomas que contribuem com o resultado estão destacados em verde, os sintomas que não contribuem estão em vermelho.



Fonte: Adaptada de Ribeiro, Singh e Guestrin (2016c).

Além da necessidade de apoio a tomada de decisões melhor fundamentadas, existem as questões éticas e legais. Neste contexto, a União Europeia efetivou em 2018 a *General Data Protection Regulation* (GDPR), uma legislação para regular as aplicações que fazem uso de inteligência computacional (EU Regulation, 2016). Entre as suas diretivas, a GDPR define que os sistemas "*(...) que tomam decisões de cunho jurídico (sobre um cidadão) ou de importância semelhante, não devem se basear em dados que contenham informações sensíveis como, por exemplo, origens étnicas, opiniões políticas, orientação sexual*"; e o direito à informação, ou seja, "*o controlador deve garantir o direito dos indivíduos em obter mais informações sobre a decisão de qualquer sistema automatizado*" (AMPARORE; PEROTTI; BAJARDI, 2021). Entretanto, a lei é vaga e não define os meios que devem ser desenvolvidos ou disponibilizados para que tal direito seja observado.

Como foi exemplificado logo acima, alguns sistemas inteligentes foram treinados sobre dados que continham vieses sistemáticos e acabaram tomando decisões discriminatórias e/ou prejudiciais. A explicabilidade pode ser utilizada para verificar se o sistema é justo em suas decisões, particularmente quando os dados de treino incluem recortes enviesados ou incompletos (POURSABZI-SANGDEH *et al.*, 2021). A existência de correlação não implica automaticamente em causalidade. Porém, como a causalidade envolve correlação, a

explicabilidade também pode validar os resultados preditos ao revelar possíveis correlações entre os atributos que levaram a um determinado resultado (ARRIETA *et al.*, 2020). Explicar sistemas de aprendizado pode facilitar a descoberta de falhas potenciais, ajudando a identificar as causas desses erros de modo mais eficiente, além de indicar o que o sistema realmente aprendeu dos dados (BHATT *et al.*, 2020).

Entretanto, não é uma tarefa trivial "abrir" as intrincadas caixas-pretas que são os sistemas de aprendizado de máquina modernos. Os modelos considerados atualmente estado da arte são caixas-pretas difíceis de compreender (TJOA; GUAN, 2020). Mesmo os modelos lineares, vistos como mais simples e transparentes, apenas são de fato transparentes em contextos limitados, sendo também uma tarefa difícil prover explicações para modelos lineares de alta dimensionalidade (GUIDOTTI *et al.*, 2018b). Neste caso e contrariando as expectativas, modelos considerados transparentes e fáceis de compreender podem, na verdade, diminuir as chances dos usuários em detectar erros devido à alta quantidade de informações (POURSABZI-SANGDEH *et al.*, 2021).

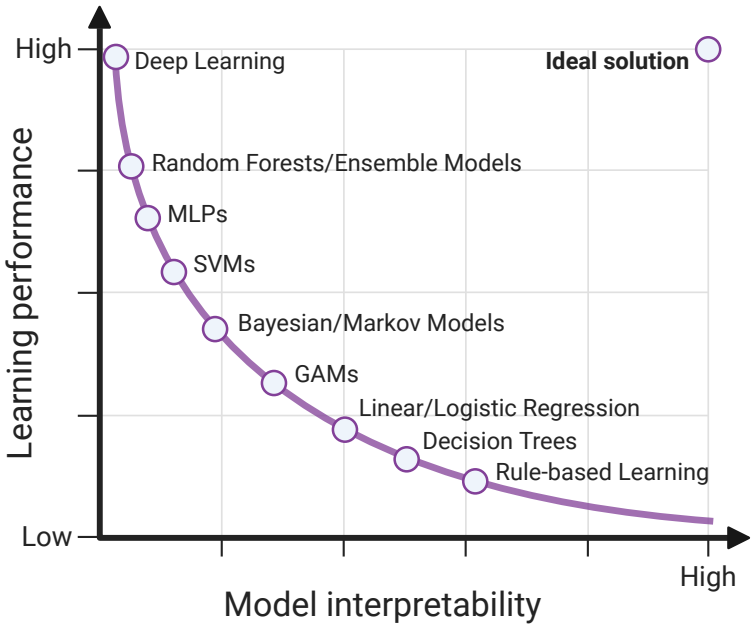
Omitir a explicabilidade leva a desafios quanto à confiança no modelo. Porém, para adicionar explicabilidade adequadamente, é preciso que estas sejam úteis, compreensíveis e válidas. Só porque uma explicação faz sentido não quer dizer que ela está correta (YANG; KIM, 2019). Dentro do ambiente XAI, um dos grandes desafios hoje existentes é o de prover explicações suportadas por validações robustas (AMANN *et al.*, 2022).

2.5.3 Objetivos da Explicabilidade

Autores como Došilović, Brčić e Hlupić (2018), Gunning e Aha (2019) e Arrieta *et al.* (2020), citam a existência de um *tradeoff* (compensação) entre a interpretabilidade dos modelos e seu desempenho. A Figura 2.14 ilustra o conflito de objetivos entre o desempenho preditivo e a transparência de alguns dos algoritmos mais comuns, desde a regressão até o *Deep Learning*. Vale lembrar que cada modelo tem o seu contexto de aplicação e as suas capacidades de atuação dentro de cada domínio, conforme destacado na Seção 2.3, e a Figura 2.14 faz uma comparação um tanto simplista, mas mesmo assim apresenta uma boa visão sobre o contraste entre interpretabilidade e precisão.

A falta de interpretabilidade limita o processo de tomada de decisões apenas a escolha de executar ou não uma recomendação automática, mas sem maiores informações que permitam compreender essa recomendação e justificar a escolha tomada (AMANN *et al.*, 2022). Neste contexto, um dos principais objetivos em XAI é gerar explicações esclarecedoras a respeito da cadeia racional que levou às predições ou recomendações por sistemas de aprendizado de máquina, de modo que os usuários possam compreender melhor os impactos dos modelos computacionais. Além disso, XAI vem para dar fundamentação teórica para os conceitos de interpretabilidade e explicabilidade, algo particularmente

Figura 2.14 – Relação entre a interpretabilidade e o desempenho preditivo dos principais modelos de aprendizado.



Fonte: Ortigossa, Gonçalves e Nonato (2024).

importante em aplicações em que modelos precisos, porém opacos, fazem predições críticas a respeito de pessoas (como visto anteriormente, no exemplo sobre predição de reincidência criminal) (KUMAR *et al.*, 2021).

Entretanto, XAI não vem para impor limitações, invalidar ou mesmo inviabilizar o Aprendizado de Máquina. As pessoas afetadas pelos resultados de aplicações baseadas em aprendizado têm o direito de conhecer e compreender apropriadamente quais foram os fatores importantes que levaram àquelas decisões, ou seja, o direito à informação, mas essas decisões de modelos devem ser explicadas mantendo os altos níveis de desempenho de aprendizado (acurácia de predição) (GUNNING; AHA, 2019).

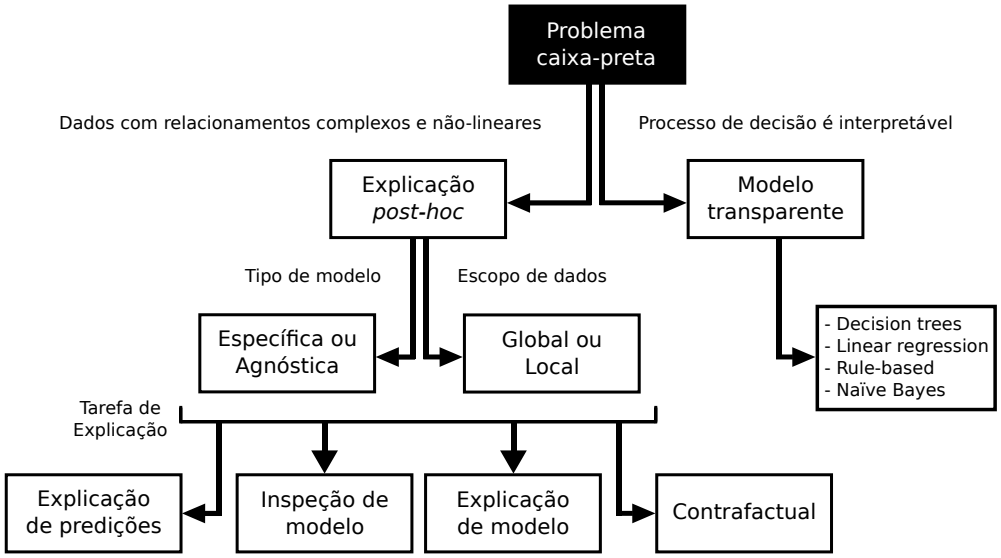
2.5.4 Categorização das Abordagens XAI

As primeiras iniciativas em gerar explicações para modelos de inteligência artificial datam dos anos 1980, quando, curiosamente, questionava-se a negligência dos sistemas do paradigma simbólico quanto às explicações (KASS; FININ, 1988). No entanto, o conceito de XAI se consolidou de fato após 2012, com a crescente necessidade de explicar os resultados de modelos de aprendizado complexos (ARRIETA *et al.*, 2020). Mesmo sendo relativamente recente, a comunidade de pesquisas em XAI tem demonstrado intensa produtividade, lançando diversas abordagens dedicadas à explicação em modelos específicos ou mesmo independentes de qualquer modelo.

Essas técnicas podem ser alocadas em diferentes categorias, de acordo com as suas características de atuação e objetivos de explicação. Basicamente, é possível "abrir" as

caixas-pretas de duas maneiras: (*i*) construindo sistemas transparentes que sejam precisos o suficiente para aproximar e substituir modelos não interpretáveis ou para atuar em conjunto com estes, em modo de apoio ou redundância (CARUANA *et al.*, 2015; LOU *et al.*, 2013); ou (*ii*) por meio da explicabilidade *a posteriori*. A Figura 2.15 apresenta um diagrama classificando as abordagens XAI.

Figura 2.15 – Taxonomia das metodologias XAI em relação aos diferentes objetivos de compreensão dos problemas caixa-preta.



Fonte: Adaptada de Ortigossa, Gonçalves e Nonato (2024).

Quando não é possível introduzir um modelo transparente, então há a necessidade de explicabilidade por meio de métodos *post-hoc*, ou seja, métodos destinados a explicar modelos de aprendizado previamente treinados que não são facilmente interpretáveis e que não podem ser eficientemente aproximados por um modelo transparente (ARRIETA *et al.*, 2020). As técnicas de explicação *post-hoc* fazem uso ativo de diversas ferramentas para aprimorar o entendimento sobre aplicações caixa-preta e, normalmente, não tem o objetivo de desvendar com precisão o modo como um modelo de aprendizado funciona internamente, mas sim apresentar informações úteis como, por exemplo, a importância de certos parâmetros da modelagem para os usuários do sistema (GUIDOTTI *et al.*, 2018b).

Quanto ao Tipo de Modelo

As técnicas *post-hoc* diferem entre si de acordo com os tipos de modelo para os quais elas são projetadas para gerar explicações. Os métodos XAI para modelos específicos são aqueles desenvolvidos para serem aplicados dentro de uma classe específica de modelos de aprendizado, e são mais indicados quando o objetivo de explicabilidade é desvendar a lógica do classificador. Já os métodos independentes de modelo, ou *model-agnostic*, são (teoricamente) capazes de serem aplicados sobre qualquer modelo de aprendizado, pois não consideram as características de algum tipo específico de modelo.

Métodos específicos podem ter desempenho superior devido às suas especificações de *design* levarem em consideração os requisitos de funcionamento da classe de modelos sob explicação. No entanto, esta categoria de métodos é limitada em termos das suas capacidades de atuação, não sendo flexível o suficiente para trabalhar com algum tipo particular de modelo fora da classe para a qual foi projetada (ADADI; BERRADA, 2018). Por outro lado, os métodos *model-agnostic* visam a compreensão do raciocínio por trás de uma predição, utilizando, para isso, simplificações, estimativas de relevância ou visualização (ARRIETA *et al.*, 2020). Os métodos *model-agnostic* normalmente separam a predição da explicação, buscando elementos explicativos sem entrar na lógica de trabalho do modelo classificador (ADADI; BERRADA, 2018).

Ribeiro, Singh e Guestrin (2016c) defendem que as técnicas *model-agnostic* são particularmente mais úteis, pois, ao produzirem explicações para qualquer algoritmo, permitem que modelos diferentes possam ser comparados utilizando a mesma técnica de explicabilidade. Já Chen, Lundberg e Lee (2021), argumentam que os métodos *model-agnostic* dependem da modelagem *a posteriori* de funções de aprendizado arbitrárias e, assim, podem sofrer com a variabilidade de amostragem quando aplicadas a modelos com muitas variáveis de entrada, o que pode dificultar a convergência entre resultados.

Neste sentido, a escolha adequada entre métodos XAI específicos ou *model-agnostic*, depende da tarefa de explicação. Se não houver a necessidade de estabelecer comparações entre modelos de classes diferentes, um método específico para o modelo sob explicação pode ser mais interessante. Caso contrário, métodos *model-agnostic* oferecerão vantagens, pois muitos classificadores do estado da arte são caixas-pretas, e um método XAI independente de modelo fornece flexibilidade para explicar os classificadores atuais e futuros (RIBEIRO; SINGH; GUESTRIN, 2016c).

Por outro lado, os modelos em série têm imposto um desafio a mais à transparência. Quando as saídas de um modelo preditivo são utilizadas como as entradas de outros modelos, tem-se uma arquitetura de modelos de aprendizado em série (CHEN; LUNDBERG; LEE, 2022). Modelos em série são *pipelines* complexos compostos pelos mais diversos tipos de caixas-pretas, como modelos lineares, em árvore e redes profundas (WOLPERT, 1992; DOUMPOS; ZOPOUNIDIS, 2007; HEALEY *et al.*, 2018). Por exemplo, dentro de uma aplicação de pontuação de consumidores, diferentes modelos estão distribuídos em instituições distintas. Cada ramificação do *pipeline* possui segmentos de dados para simular elementos diferentes sobre a pontuação de cada consumidor (sobre fraude, crédito, risco à saúde, entre outros). Note que essa composição torna os modelos em série mais complexos de explicar, se comparado a um modelo único.

Os modelos em série têm se destacado em aplicações sensíveis, como no exemplo acima, exigindo abordagens dedicadas a explicar seus resultados para depuração e geração de

confiança, visando contornar a considerável falta de transparência deste tipo de estrutura complexa (CHEN; LUNDBERG; LEE, 2022). Uma solução natural poderia ser a aplicação de métodos XAI independentes de modelo para explicar toda a série de modelos de uma só vez. Embora as técnicas *model-agnostic* padrão possam, em tese, explicar modelos em série, elas não funcionariam adequadamente porque os métodos *model-agnostic* possuem algumas deficiências neste contexto: exigem acesso a todos os modelos da série, mas as instituições proprietárias podem não compartilhar os seus dados ou modelos. Além disso, métodos *model-agnostic* costumam ter um custo computacional mais elevado, o que pode resultar em intratabilidade para grandes séries (CHEN; LUNDBERG; LEE, 2022).

Já os métodos específicos padrão são geralmente mais rápidos do que as alternativas independentes de modelo. No entanto, eles não podem ser utilizados diretamente para explicar modelos em série, pois são projetados especificamente para operar considerando um tipo de caixa-preta, e um modelo em série pode conter diversos tipos de modelos preditivos em sua estrutura. Os modelos em série ainda são pouco explorados em XAI, exigindo métodos específicos híbridos, ou seja, capazes de lidar com a característica distribuída, mas, ao mesmo tempo, generalistas o suficiente para gerenciar a diversidade de modelos.

Quanto ao Escopo de Dados

Outra categoria relevante dentro do XAI é quanto a granularidade das explicações. Para tornar o Aprendizado de Máquina interpretável é preciso estabelecer confiança no modelo e nas predições. Logo, confiança emerge como um conceito vital porque se os usuários não confiarem no modelo e/ou nas predições, eles não utilizarão esses sistemas. Ribeiro, Singh e Guestrin (2016c) especificam dois tipos de confiança de acordo com a granularidade:

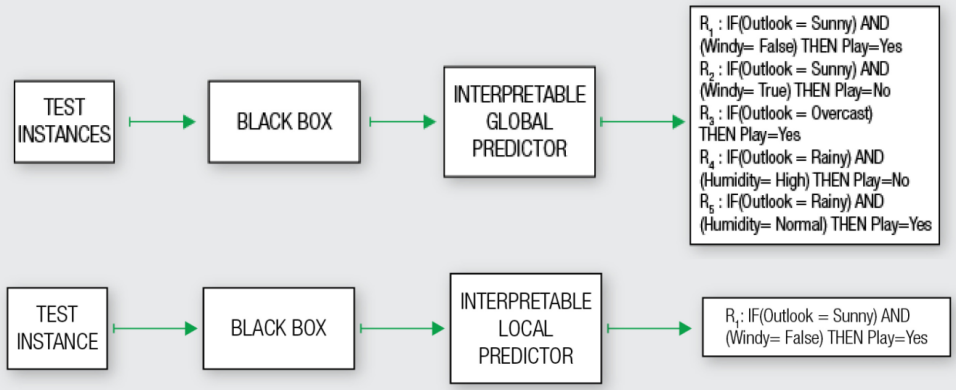
- **No modelo como um todo** – O usuário tem confiança o suficiente que o modelo se comportará de modo razoável quando implantado. Na etapa de modelagem são aplicadas métricas de avaliação sobre o modelo gerado a partir dos dados (o processo de validação), objetivando mimetizar o comportamento de mundo real. Mas existem diferenças significativas entre o conteúdo dos dados e o mundo real;
- **Em predições individuais** – O usuário confia o suficiente na predição para tomar uma decisão baseada nela. Apesar da aplicação das métricas de avaliação, é necessário testar as predições individualmente. Não é razoável aceitar uma predição não fundamentada, especialmente quando as consequências dos atos podem ser catastróficas, como no diagnóstico médico ou no combate ao terrorismo, por exemplo.

Neste contexto, quando a tarefa de explicação envolve obter uma visão geral do comportamento do modelo, em termos dos elementos que mais influenciaram o modelo

como um todo, tem-se um método global (AAS; JULLUM; LØLAND, 2021). Métodos globais oferecem uma descrição sumarizada e compacta do comportamento do modelo quando submetido a todo o conjunto de dados ou a um recorte deste (CHAN *et al.*, 2020a). A abordagem é comumente utilizada para comparar a relevância global de uma variável e compreender quais outras variáveis são as mais relevantes em decisões a nível populacional, como mudanças climáticas ou tendências de consumo de drogas, por exemplo, em que a estimativa do comportamento global é mais útil do que explicações para todas as possíveis características da modelagem (KUMAR; CHANDRAN, 2021). Porém, nos demais casos, pode ser difícil ou pouco informativo capturar uma visão geral de todo o mapeamento gerado, especialmente em modelos com muitas variáveis.

Os métodos dedicados à explicabilidade local são indicados quando o objetivo é reter uma descrição mais acurada dos detalhes que envolvem a predição de uma única instância (CHAN *et al.*, 2020b). Explicações locais são úteis para modelos complexos que se comportam de maneiras diferentes quando estão sob diferentes combinações de variáveis (AAS; JULLUM; LØLAND, 2021). Gerar explicações individuais justifica o motivo do modelo ter tomado uma decisão específica para uma instância de dado, quando uma visão global do comportamento não seria descritiva o suficiente (ADADI; BERRADA, 2018). A Figura 2.16 apresenta um comparativo entre as abordagens globais e locais.

Figura 2.16 – Diagrama comparativo entre as metodologias de explicação globais e locais.



Fonte: Guidotti *et al.* (2018b).

Conforme Ribeiro, Singh e Guestrin (2016c), para que uma explicação seja significativa, ela deve manter a fidelidade local, isto é, deve corresponder ao modo como o modelo se comporta na vizinhança em que a instância foi predita. Os autores ainda pontuam que a fidelidade local não implica simultaneamente em fidelidade global, pois, características globalmente importantes podem não ser localmente importantes e vice-versa. Obviamente, não é possível ter uma explicação global completamente fiel sem que esta seja uma descrição completa do próprio modelo. A simples coleção de explicações de múltiplas instâncias pode não funcionar bem na caracterização a nível populacional porque as explicações locais são específicas a nível de instância, o que muitas vezes é inconsistente com demandas

globais (WOJTAS; CHEN, 2020). Assim, identificar explicações globalmente fiéis e interpretáveis permanece um desafio (RIBEIRO; SINGH; GUESTRIN, 2016c).

Quanto ao Tipo de Explicação

Existem métodos dedicados à compreensão das estruturas e mecanismos dos modelos de aprendizado. Esta categoria de técnicas é normalmente encontrada em aplicações de Redes Neurais, em que a visualização é aplicada para gerar representações dos padrões internos das unidades neurais. Porém e ao contrário da intuição, Poursabzi-Sangdeh *et al.* (2021) mostraram que expor os mecanismos internos de um modelo de aprendizado, na verdade reduz a habilidade dos usuários em detectar comportamentos errôneos para instâncias não usuais. Amann *et al.* (2022) indicam que essa redução na interpretabilidade pode ser motivada pela sobrecarga gerada pela grande quantidade de informações que os usuários são expostos durante o processo de compreensão, mesmo de modelos transparentes. Isso não invalida os métodos de explicação de modelos, mas alerta os desenvolvedores para que projetem cuidadosamente ferramentas capazes de sintetizar informações.

A inspeção é utilizada quando o objetivo é verificar a sensibilidade do modelo, ou seja, o modo como o algoritmo de aprendizado ou as suas predições se comportam ao sofrer perturbações que variem os dados de entrada. Já a explicação de predições consiste em gerar esclarecimentos a respeito dos fatores que influenciaram na decisão final do modelo, apresentando artefatos textuais ou visuais que forneçam compreensão qualitativa do relacionamento entre as variáveis de entrada e a predição.

De acordo com Ribeiro, Singh e Guestrin (2016c), explicar predições de modo fiel e inteligível favorece o estabelecimento de confiança entre os usuários e os sistemas de aprendizado. A explicação local de predições não requer que seja desvendada toda a lógica interna de um classificador e, atualmente, é uma das principais vertentes em XAI, com diversas técnicas dedicadas a quantificar a contribuição dos elementos envolvidos nas predições de modelos complexos (ADADI; BERRADA, 2018; TAN *et al.*, 2023). Além disso, os resultados dos modelos também podem ser interpretados por meio de explicações baseadas em evidências (factuais ou contrafactuais). As explicações contrastivas e contrafactuais buscam justificativas para o porquê uma decisão não foi algo diferente do predito e como ela pode ser modificada, respectivamente (STEPIN *et al.*, 2021).

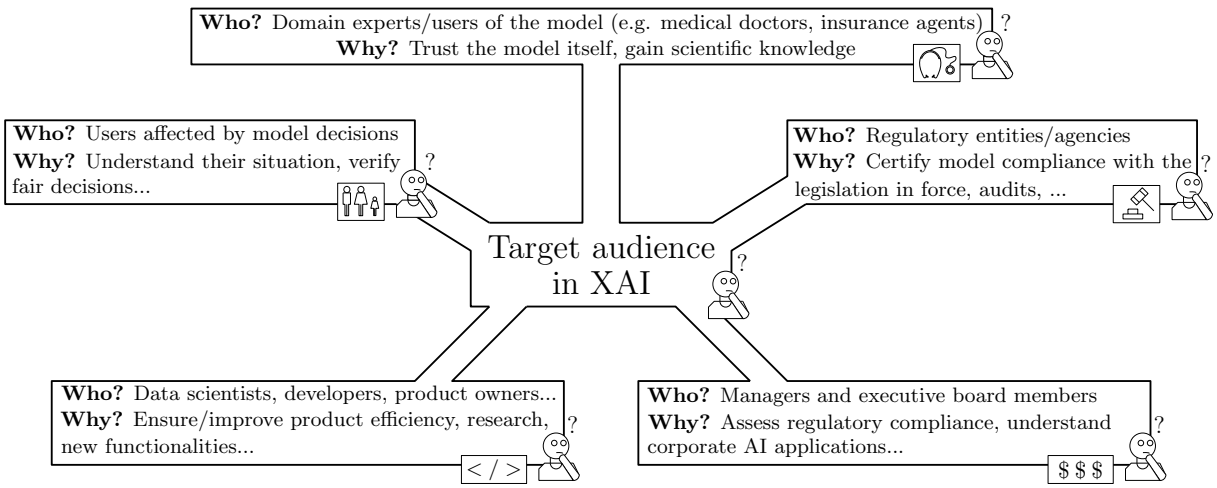
2.5.5 Responsabilidade em Inteligência Artificial

Estabelecer confiança é um dos principais fundamentos do XAI. Logo, não é razoável utilizar um esclarecedor de caixas-preta como uma caixa-preta em si. Segundo a GDPR, no que se refere ao direito à explicação: "*caso uma aplicação forneça explicações inconsistentes para instâncias similares (ou para a mesma instância), essas explicações não podem ser*

consideradas confiáveis". Isso quer dizer que um bom método de explicabilidade não deve prover explicações (completamente) contrastantes quando for executado múltiplas vezes para gerar explicações sobre a mesma instância ou o mesmo grupo de instâncias. Além disso, também deve haver preservação da consistência ao explicar dados similares. Logo, a estabilidade é um dos requisitos mínimos a ser observado pelos métodos XAI que intencionam ser ferramentas confiáveis (AMPARORE; PEROTTI; BAJARDI, 2021). Caso contrário, sem consistência seria possível levantar questionamentos sobre a confiabilidade geral da explicação, algo que colocaria em cheque todo o seu propósito esclarecedor.

É importante destacar que explicações são dependentes de contexto, ou seja, para quais perguntas se busca resposta? Não é possível responder a esta questão sem ter em mente a audiência ou o público-alvo a quem as explicações são endereçadas (RIBEIRO; SINGH; GUESTRIN, 2016c). Especialistas de domínio e desenvolvedores podem se interessar por descobrir erros ou vulnerabilidades ocultas pela complexidade dos modelos, e melhorar o seu entendimento levaria à correção de suas deficiências (ARRIETA *et al.*, 2020). Usuários e agências reguladoras podem demandar por respostas lógicas e verificáveis dos sistemas de tomada de decisão que os afetam, seja para esclarecer dúvidas ou para garantir que critérios de *compliance* (conformidade) e/ou isonomia estejam sendo observados. Já os cientistas de dados e gerentes corporativos demandam por meios que possibilitem verificar se os seus dados estão sendo transformados em informações realmente úteis e pelos motivos corretos. Liao, Gruen e Miller (2020) definiram o "banco de questões XAI", um conjunto de questões baseadas em "como", "por que" e "o que" em que os usuários de Aprendizado de Máquina podem se interessar em obter respostas. Essas questões visam guiar boas práticas de *design* para os desenvolvedores XAI. A Figura 2.17 ilustra os diferentes contextos em que as técnicas XAI se inserem e suas respectivas demandas.

Figura 2.17 – O propósito de uma técnica de explicação também depende do público a quem ela é destinada.



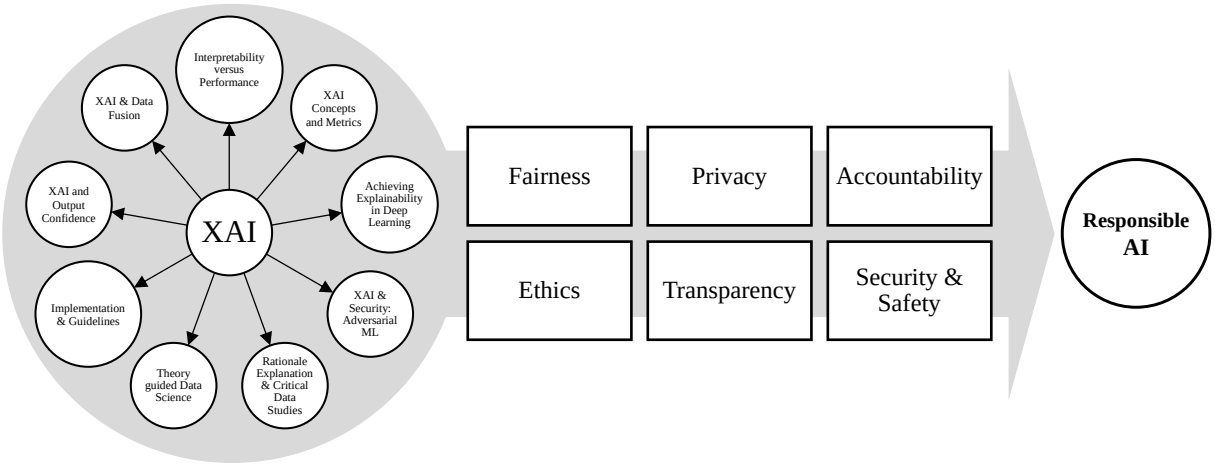
Fonte: Arrieta *et al.* (2020).

Antes de finalizar esta seção, vale ressaltar mais alguns dos principais desafios na área de *Explainable Artificial Intelligence*. Esses desafios vão além das já expostas dificuldades técnicas em gerar explicações para modelos complexos como as DNNs, por exemplo. Princípios importantes que devem sempre permear o desenvolvimento e implantação de qualquer sistema de Inteligência Artificial, como a segurança, a privacidade e a garantia de proteção dos dados, também devem constar nos métodos de explicabilidade.

Afinal, explicar uma predição não significa expor dados sensíveis e que não deveriam ser publicados. Entretanto, o que é ou não é considerado um dado sensível deve obedecer a critérios como ética e justiça. É imperativo que os sistemas computacionais inteligentes sejam imparciais com relação a aspectos sociais como religião, opiniões políticas, origens étnicas, dentre outros, e os métodos XAI, além de também respeitar esse critério, devem trabalhar no sentido de aprimorar os modelos de aprendizado para garantir que estes cumpram suas tarefas com precisão e responsabilidade.

A Figura 2.17 destaca os desafios a serem superados ao promover avanços na direção da Inteligência Artificial responsável. Arrieta *et al.* (2020) e Tjoa e Guan (2020) apresentam extensas discussões sobre os cenários desafiadores em questão.

Figura 2.18 – Desafios no desenvolvimento responsável de explicações para sistemas baseados em Aprendizado de Máquina.



Fonte: Arrieta *et al.* (2020).

2.6 Tecnologias e Ferramentas Utilizadas

Foi utilizado o *Python* como linguagem de programação para desenvolver todos os *scripts* relacionados à metodologia XAI projetada neste trabalho. *Python* é uma linguagem de propósito geral que tem se tornado uma das alternativas mais populares para a computação científica (PEDREGOSA *et al.*, 2011). *Python* oferece ao desenvolvedor um ambiente rico em bibliotecas de código aberto, dedicadas aos processos de aprendizado de máquina,

mineração de dados, transformações matemáticas algébricas, métodos otimizados voltados para a execução de tarefas analíticas dependentes de intenso processamento de dados, além da construção de gráficos de alta qualidade (MCKINNEY, 2012).

Em especial, as bibliotecas *Scikit-learn* (https://scikit-learn.org/), *XGBoost gradient boosting* (CHEN; GUESTRIN, 2016), *PyTorch* (https://pytorch.org/) e *TensorFlow* (https://www.tensorflow.org/) oferecem implementações otimizadas e portáteis para os principais algoritmos de aprendizado, por meio de interfaces simples de se compreender e trabalhar (PEDREGOSA *et al.*, 2011). Outras bibliotecas serão utilizadas, entre elas: *numpy* (https://numpy.org/) e *SciPy* (https://scipy.org/), para estruturas de dados e métodos matemáticos; *pandas* (https://pandas.pydata.org/), para operações de análise e manipulação de dados; SHAP (https://shap.readthedocs.io/), LIME (https://lime-ml.readthedocs.io/) e Captum (https://captum.ai/), com os principais métodos XAI da atualidade; e a OpenXAI (https://open-xai.github.io/) para geração de dados sintéticos.

Vale destacar a estabilidade e a boa documentação que normalmente acompanha as bibliotecas *Python*, além da extensa comunidade de desenvolvedores ativos nos fóruns de dúvidas e a grande quantidade de exemplos disponíveis em repositórios na *Internet*, com trechos de código e aplicações que auxiliam no desenvolvimento, o que reforça a escolha do *Python* como a linguagem de programação utilizada nesta pesquisa.

O ambiente de programação escolhido foi o *Jupyter Notebook* (https://jupyter.org/), que é uma interface *web* interativa para codificação em *Python*. O *Jupyter Notebook* é de fácil configuração e possui um *layout* modular que permite organizar o fluxo de trabalho em blocos de programação que podem ser testados de modo rápido e independente, além de oferecer meios que facilitam a instalação de bibliotecas externas e a inclusão de comentários para a geração de relatórios.

Os códigos-fonte desenvolvidos nesta pesquisa estão disponíveis em um repositório *online* que pode ser acessado em **https://github.com/evortigosa/EXplainable_AI**. Este diretório contém todos os códigos desenvolvidos ao longo deste doutorado, incluindo versões descontinuadas e preliminares de métodos que, de algum modo, contribuíram com a solução implementada e descrita aqui.

Todos os experimentos realizados foram executados no mesmo computador com 16GB de RAM e processador Intel® Core™ i7-3520M de 2.90GHz, em ambiente Linux 64-bits, utilizando *Python* 3.8 e *Jupyter Notebook* sobre o navegador *Google Chrome*.

2.7 Considerações Finais

Neste capítulo foram discutidos os conceitos que envolvem o amplo universo dos algoritmos baseados em Aprendizado de Máquina, desde as suas características, vantagens

e limitações até o que os tornam problemas do tipo "caixa-preta". Conforme exposto, são muitas as questões que devem ser consideradas e que podem influenciar na capacidade das ferramentas XAI em revelar a lógica e o modo de funcionamento de modelos de aprendizado sofisticados com múltiplos parâmetros com relacionamentos complexos, que operam sobre dados de alta dimensionalidade.

Lembrando que nem todos os sistemas baseados em aprendizado exigem interpretabilidade. Isto é, não há a necessidade de maiores explicações quando não há consequências significativas para os resultados de um algoritmo, ou se o problema sob investigação já foi suficientemente testado em situações reais (DOSHI-VELEZ; KIM, 2017). Entretanto, como modelos de aprendizado complexos têm sido cada vez mais empregados para a tomada de decisões importantes dentro de contextos críticos, como a medicina de precisão, por exemplo, a demanda por um aumento na transparência desses algoritmos também tem aumentado. Isso basicamente se deve pelo fato dos humanos se mostram reticentes em empregar técnicas que não sejam interpretáveis, tratáveis ou confiáveis, algo compreensível e que acaba limitando o alcance do Aprendizado de Máquina (THEODOROU; WORTHAM; BRYSON, 2017); além da atual compreensão de que a utilização de decisões não justificáveis ou que simplesmente não permitam explicações detalhadas sobre o seu processo lógico, pode acarretar em situações de risco ou mesmo causar impactos profundos na dinâmica social (KUCHARSKI, 2016; PREECE *et al.*, 2018; ARRIETA *et al.*, 2020).

O "direito à informação", definido pela GDPR europeia, e também o recentemente introduzido *California Consumer Privacy Act* (CCPA) (CALIFORNIA, 2021), ilustram a sensação de inerente falta de ética que muitos têm sobre o processo de tomada de decisões envolvendo indivíduos e sistemas automatizados, sem que ao menos se tenha uma explicação razoável (KUMAR *et al.*, 2020). Por isso, a pesquisa e o desenvolvimento de técnicas XAI compreensivas é essencial para se obter entendimento sobre as decisões tomadas pelos sistemas de *Machine Learning*, oferecendo aos analistas e usuários visões sumarizadas sobre as informações ocultadas pelos complexos e intrincados espaços paramétricos que os atuais modelos de aprendizado costumam apresentar.

Capítulo 3

Trabalhos Relacionados

3.1 Considerações Iniciais

Segundo Lundberg e Lee (2017), a melhor explicação para um modelo simples é o próprio modelo, uma vez que este representa a si próprio perfeitamente e é fácil de compreender. Entretanto, quando se trata de abordagens complexas como o *Random Forest* e as *Deep Neural Networks*, o modelo original não pode ser utilizado porque é difícil compreender as suas estruturas de decisão. A interpretabilidade vem do próprio *design* do modelo, e quando não é possível interpretar diretamente, deve-se aplicar métodos de explicação que transformem elementos obscuros em informações interpretáveis.

Este capítulo apresenta uma revisão da literatura, por meio de uma análise sobre as características e funcionalidades de algumas das pesquisas mais relevantes em XAI. As ferramentas construídas para prover explicabilidade em modelos de aprendizado sofisticados e pouco transparentes devem ser desenvolvidas a partir de um profundo estudo do domínio de aplicação, das necessidades analíticas dos usuários, e das características a serem desempenhadas dentro do contexto em que atuarão.

Na Seção 3.2 são destacados trabalhos do estado da arte em XAI, especificamente (mas não somente) os estudos que desenvolvem métodos dedicados a explicar predições, abordando suas respectivas vantagens e fragilidades. Na Seção 3.3 é feita uma discussão sobre procedimentos de avaliação em XAI, enquanto a Seção 3.4 apresenta uma visão geral sobre as limitações da explicabilidade. A Seção 3.5 sumariza as abordagens e métodos XAI discutidos. Na Seção 3.6, estão algumas considerações finais sobre este capítulo.

3.2 Estado da Arte em *EXplainable Artificial Intelligence*

O recente aumento nas pesquisas em explicabilidade tem resultado no desenvolvimento e publicação de uma rica variedade de métodos. Com isso, muitos pesquisadores têm se

dedicado em revisar e sumarizar a temática, buscando esclarecer as diversas particularidades e técnicas da área (LINARDATOS; PAPASTEFANOPOULOS; KOTSIANTIS, 2020; BELLE; PAPANTONIS, 2021; BURKART; HUBER, 2021). Lipton (2018) foi um dos pioneiros na tentativa de organizar as principais definições sobre o que é interpretabilidade em Aprendizado de Máquina. Embora a publicação final seja de 2018, a primeira versão do artigo foi disponibilizada em 2016 e reúne uma boa argumentação sobre as necessidade e motivações da interpretabilidade, de acordo com a literatura então disponível. Doshi-Velez e Kim (2017), Chakraborty *et al.* (2017) e Došilović, Brčić e Hlupić (2018) apresentaram introduções em conceitos e taxonomia. Estes trabalhos são seminais e contém visões gerais interessantes quanto aos avanços iniciais no XAI.

Murdoch *et al.* (2019) atualizaram a visão conceitual e Molnar (2019) trouxe uma extensa revisão, não apenas conceitual, mas também sobre as características das principais abordagens XAI. Já Guidotti *et al.* (2018b) analisaram uma ampla variedade de métodos interpretáveis e de explicabilidade, classificando-os de acordo com o tipo de problema para os quais eles são indicados, além de descreverem uma detalhada taxonomia sob o ponto de vista da Mineração de Dados e do Aprendizado de Máquina.

De modo semelhante, Adadi e Berrada (2018) e Arrieta *et al.* (2020) realizaram ricas pesquisas na literatura, apresentando visões abrangentes e aprofundadas do cenário em XAI, desde os fundamentos, contribuições e desafios na área, até as soluções desenvolvidas e aplicadas para lidar com as diferentes necessidades de explicabilidade. Ambas as publicações abordam conceitos éticos de *fairness* (justiça no sentido de imparcialidade) e *compliance* (observância legal) em Aprendizado de Máquina, diferindo quanto ao aspecto crítico, com Adadi e Berrada (2018) levantando questões relativas à (falta de) avaliação das explicações e Arrieta *et al.* (2020) sugerindo diretrizes para a construção de sistemas socialmente responsáveis. Ainda com discussões aprofundadas em XAI, Tjoa e Guan (2020) e Amann *et al.* (2022) também debatem conceitos e aplicações, porém, estes autores direcionaram suas pesquisas na explicabilidade de sistemas caixa-preta utilizados em medicina. Estes dois últimos trabalhos levantam pontos importantes sobre os riscos da falta de explicações esclarecedoras dentro de aplicações médicas, com Amann *et al.* (2022) destacando a necessidade de fixar a terminologia e validar as explicações.

Conforme Molnar (2019), é possível interpretar o processo de decisão de um modelo de aprendizado analisando o quanto cada variável influencia na predição de uma instância de dado. É fácil verificar essas influências (ou importâncias) individuais em um modelo linear f, formalizado do seguinte modo para uma instância n-dimensional, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$:

$$f(\mathbf{x}) = \omega_0 + \omega_1 x_1 + \cdots + \omega_n x_n \tag{3.1}$$

sendo x_i o valor do atributo i da instância $\mathbf{x}$, com ω_i o seu respectivo peso associado e

$\omega_i x_i$ correspondendo ao efeito desta variável (peso multiplicado pelo valor do atributo). A influência ϕ_i que a i-ésima variável implica sobre a predição $f(\mathbf{x})$ é calculada por:

$$\phi_i(f) = \omega_i x_i - \mathbb{E}[\omega_i \mathbf{X}_i] = \omega_i x_i - \omega_i \mathbb{E}[\mathbf{X}_i] \tag{3.2}$$

com $\mathbb{E}[\mathbf{X}_i]$ consistindo no valor esperado da variável i. Ou seja, o quanto cada atributo contribui para uma predição pode ser inferido pela diferença entre seu efeito e seu valor esperado. Somando todas as influências das variáveis de uma instância, tem-se o seguinte:

$$\begin{aligned} \sum_{i=1}^{n} \phi_i(f) &= \sum_{i=1}^{n} (\omega_i x_i - \mathbb{E}[\omega_i \mathbf{X}_i]) \\ &= (\omega_0 + \sum_{i=1}^{n} \omega_i x_i) - (\omega_0 + \sum_{i=1}^{n} \mathbb{E}[\omega_i \mathbf{X}_i]) \\ &= f(\mathbf{x}) - \mathbb{E}[f(\mathbf{X})] \end{aligned} \tag{3.3}$$

sendo esta a diferença entre a predição para a instância $\mathbf{x}$ e o valor esperado da predição.

Note que este conceito de "efeito" apenas funcionou diretamente devido à linearidade do modelo. Seria interessante explicar outras classes de modelos desse modo simplificado, entretanto, uma quantidade moderada de não-linearidade nos relacionamentos entre variáveis pode demandar um aumento na complexidade de formulação da modelagem linear que, consequentemente, acaba reduzindo a sua interpretabilidade. Em outras palavras, mesmo os modelos lineares podem se tornar demasiado complexos para serem interpretáveis (CHEN; LUNDBERG; LEE, 2021). No entanto, *feature effect/impact* (efeito/impacto do atributo), *variable contribution* (contribuição da variável), *feature-level interpretations* (interpretações a nível de atributo) são termos frequentes em XAI utilizados para descrever como ou em que medida cada atributo contribui para a predição do modelo, ou seja, descrever a *feature importance* (importância de atributos).

Breiman (2001) propôs uma das primeiras abordagens para identificar os atributos que mais impactam na decisão de um modelo, em seu clássico artigo definindo o *Random Forest*. A solução de Breiman envolve a permutação dos valores de cada atributo para avaliar sua contribuição individual, calculando a perda sofrida pelo modelo devido às permutações. O processo de permutação assume a independência entre os atributos, considerando que a permutação quebra a conexão entre os atributos de entrada e a saída do modelo, resultando redução do desempenho preditivo para os atributos mais importantes, dado que atributos mais importantes dão origem às maiores perdas. O método de Breiman é específico para *Random Forests* treinadas, um tipo de modelo impenetrável, ou seja, virtualmente impossível de interpretar (BREIMAN, 2001; RIBEIRO; SINGH; GUESTRIN, 2016c).

Embora a gama de metodologias XAI seja variada, explicar predições por meio da

caracterização da importância dos atributos é objetivo comum dentro das abordagens XAI, seja local ou globalmente, explícita ou mesmo implicitamente (TAN *et al.*, 2023). Logo, muitos autores classificam os métodos XAI de acordo com outras propriedades, como os mecanismos aplicados para computar importâncias. A atribuição de importâncias está no centro do *feature importance*, mas esta não é o único modo de se caracterizar importâncias. Em resumo, a terminologia relacionada ao *feature importance* pode ser definida com:

- ***Feature Attribution*** – Mede as contribuições individuais dos atributos de entrada no desempenho de um modelo de aprendizado supervisionado, distribuindo de modo justo o valor predito entre estes atributos para quantificar a sua relevância individual (CASALICCHIO; MOLNAR; BISCHL, 2019; WOJTAS; CHEN, 2020);
- ***Additive Importance*** – Explicação em que o somatório de todas as importâncias atribuídas deve aproximar o valor original da predição (LUNDBERG; LEE, 2017);
- ***Sensitivity*** – Mede como o desempenho preditivo de um modelo de aprendizado varia (aumentando ou diminuindo) ao se perturbar as variáveis de entrada (MISHRA *et al.*, 2021). Do ponto de vista da análise de sensibilidade, quanto mais importante uma variável for, mais significativamente ela contribui para o desempenho preditivo;
- ***Gradient-based*** – Caso particular da abordagem de sensibilidade que avalia como o modelo de aprendizado se comporta a partir de pequenas perturbações nos atributos, de tamanho infinitesimal (BHATT *et al.*, 2020);
- ***Feature Selection*** – A partir de um conjunto de dados original com n atributos, identifica uma combinação ou um subconjunto de p atributos importantes ou que mais contribuem para treinar um modelo com perda mínima de precisão. Na prática, $p \ll n$ para a maioria das tarefas de seleção de atributos (DAS *et al.*, 2022).

Deve-se distinguir a seleção de atributos da extração de atributos. Ambas as metodologias visam melhorar o desempenho de modelos baseados em dados, ao reduzir o espaço de características original. Embora existam exceções em que a extração de atributos não necessariamente reduz dimensionalidade, os métodos de extração de atributos (ou características) estão mais intimamente ligados às tarefas de redução de dimensionalidade, ao gerar conjuntos contendo novos atributos baseados nos dados originais. Isso é feito por meio de transformações lineares ou não-lineares que mapeiam representações significativas de baixa dimensionalidade, preservando informações relevantes e previamente definidas do espaço de alta dimensionalidade (MAATEN; HINTON, 2008; NONATO; AUPETIT, 2018; ORTIGOSSA; DIAS; NASCIMENTO, 2022); enquanto a seleção de atributos, apesar de também reduzir a dimensionalidade, é aplicada de modo a remover atributos com base em projeções canônicas.

3.2.1 XAI Baseada em Aproximações

Quando há a necessidade de aplicar modelos de aprendizado sofisticados e precisos, sabe-se que a sua forte não-linearidade resulta em falta de transparência, requerendo abordagens de explicabilidade. A explicabilidade pode ser alcançada utilizando algoritmos inerentemente interpretáveis que aproximem o desempenho preditivo do modelo original (caixa-preta). Mais especificamente, o modelo caixa-preta pode ser utilizado como um "treinador" para transferir o conhecimento para um modelo mais transparente e interpretável que aproxime e explique as predições do modelo original, abordagem esta conhecida por *model distillation* (HINTON; VINYALS; DEAN, 2015).

Um reconhecido exemplo de interpretabilidade vem das árvores de decisão, visto que a sua sequência lógica pode ser intuitivamente interpretada por um analista humano (AMANN *et al.*, 2022). Outras estratégias interpretáveis incluem a regressão logística e o aprendizado baseado em regras (GUIDOTTI *et al.*, 2018b). Entretanto, estes modelos possuem limitações significativas (cf. Seção 2.3). Tan *et al.* (2023) argumentam que as explicações baseadas em árvores de decisão são pouco precisas e, em alguns casos, se mostram ainda menos precisas do que explicações simples baseadas em modelos lineares.

Friedman e Popescu (2008) e Guidotti *et al.* (2018a) desenvolveram soluções com classificadores baseados em regras. Embora sejam considerados transparentes, os métodos baseados em regras possuem limitações semelhantes às elencadas para a regressão logística quanto à escalabilidade. Em alguns casos, é necessário gerar conjuntos excessivamente grandes de regras para obter um bom nível de classificação, o que acaba inviabilizando a análise. Os modelos baseados em regras apenas são indicados para a aproximação em domínios reduzidos (ZEDNIK, 2021). Logo, modelos mais simples não são eficientes para tratar dados de alta dimensionalidade com relacionamentos complexos.

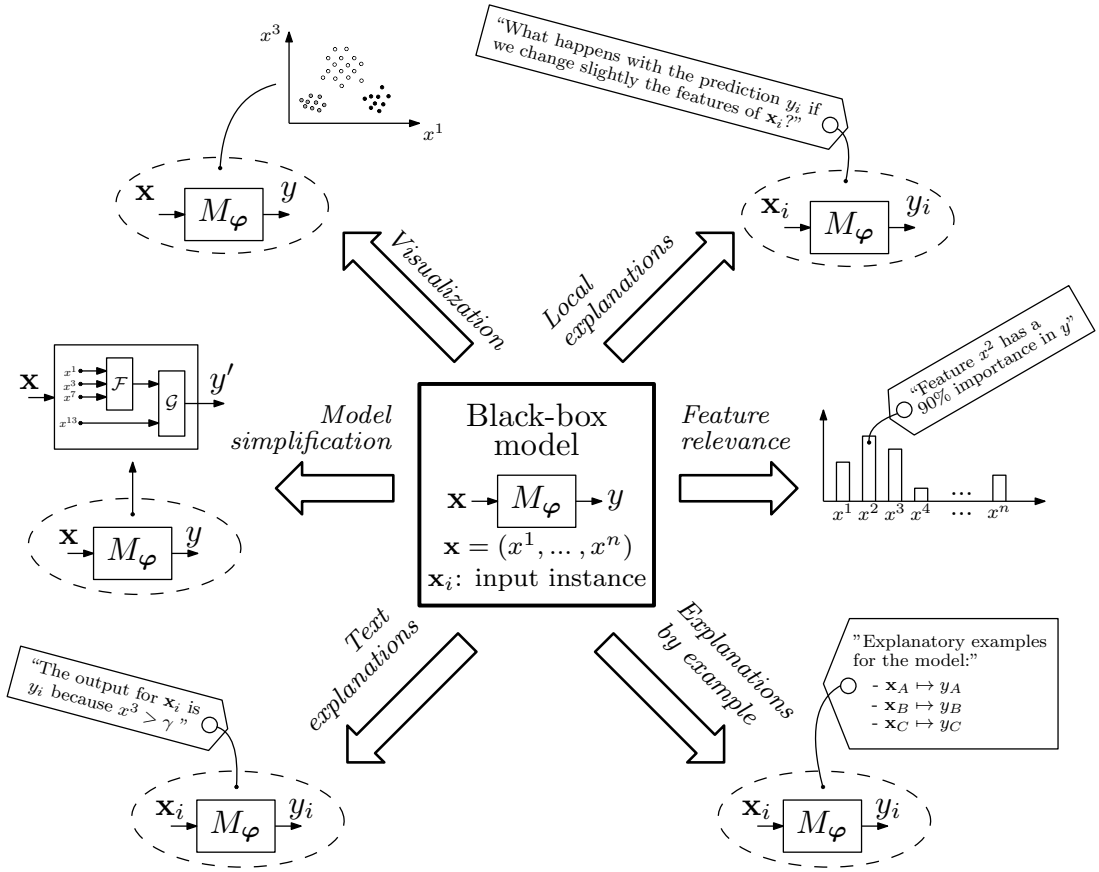
Lou, Caruana e Gehrke (2012) apresentaram uma abordagem baseada em GAMs (*Generalized Additive Models*) como uma alternativa interpretável aos modelos de regressão complexos. GAMs são modelos lineares de suavização que decompõem a função a ser predita em uma agregação de componentes unidimensionais definidos para as variáveis preditivas (HASTIE; TIBSHIRANI, 2017). Com isso, uma GAM é capaz de capturar relacionamentos não-lineares individuais das variáveis modeladas. Lou *et al.* (2013) desenvolveram uma versão aprimorada do método anterior, com uma nova e otimizada formulação matemática. Esta versão aprimorada foi aplicada por Caruana *et al.* (2015) em um estudo para predizer o risco de readmissão médica em casos de pneumonia. O trabalho apresenta casos de estudo bem detalhados e com resultados interessantes, justificando a abordagem baseada em GAM por este ser o "padrão-ouro" em termos de interpretabilidade. Porém, o uso de GAMs é limitado devido à sua baixa *performance* para gerar explicações

em modelagens mais complexas (ARRIETA *et al.*, 2020).

Tan *et al.* (2023) realizaram um estudo comparativo indicando que explicações geradas utilizando métodos XAI baseados em aproximação apresentam resultados precisos dentro do contexto das explicações aditivas. No entanto, quando a aproximação por meio de modelos de aprendizado transparentes não é viável, faz-se necessário partir para a explicabilidade *post-hoc* (MOLNAR, 2019). O universo das técnicas *post-hoc* já é extenso na literatura, com a pesquisa de Arrieta *et al.* (2020) categorizando os métodos *post-hoc* de acordo com o modo em que eles geram as suas explicações.

A Figura 3.1 sintetiza as principais classes de abordagens projetadas para converter dados de modelos não-interpretáveis em informações explicáveis. Note que essa classificação pode ser vista como uma extensão do diagrama apresentado na Figura 2.15. A seguir, serão discutidas técnicas *post-hoc* seguindo uma classificação semelhante, mas de modo não restrito, visto que alguns métodos combinam duas ou mais classes de abordagens, além da constante evolução das propostas de explicabilidade.

Figura 3.1 – Diferentes abordagens *post-hoc*. Os métodos podem ser classificados de acordo com a apresentação das explicações.



Fonte: Arrieta *et al.* (2020).

3.2.2 XAI Baseada em Visualização de Informações

Visualização de informações é o mapeamento de dados dentro de formatos gráficos que simplificam a sua representação, auxiliando os analistas na descoberta visual de tendências, padrões e características (ALEXANDRINA *et al.*, 2019). Técnicas de visualização têm sido utilizadas com sucesso há muitos anos em diversos domínios de aplicação, o que inclui XAI, em que a comunidade de pesquisas em visualização tem dedicado consideráveis esforços em utilizar gráficos para prover elementos interpretativos para modelos complexos, desde as Redes Neurais (KAHNG *et al.*, 2018) até as recentemente introduzidas arquiteturas de *Deep Learning* baseadas em *Transformers* (VASWANI *et al.*, 2017; WU *et al.*, 2023).

Marcílio-Jr, Eler e Breve (2021) apresentaram uma ferramenta *model-agnostic* suportada por visualizações coordenadas para representar a similaridade entre classes. A ferramenta calcula os *feature importances* utilizando uma técnica que introduz perturbações em atributos individuais que minimizam o desempenho do modelo e a perturbação ao mesmo tempo. Chan *et al.* (2020a) desenvolveram uma interface gráfica que permite inspecionar predições a partir de diferentes níveis de densidade. O sistema constrói uma visão global sumarizada permitindo que o usuário "navegue" até explicações locais. Com isso, a interface apresenta a importância de uma instância a partir de contextos gerados pelo agrupamento de instâncias similares em vetores de tópicos com diferentes densidades.

Partial Dependence plots (FRIEDMAN, 2001) são ferramentas gráficas utilizadas para gerar entendimento sobre modelos de aprendizado supervisionado, por meio da visualização do efeito marginal médio (valores de dependência parcial) entre as variáveis de entrada e as predições (ADADI; BERRADA, 2018). A dependência parcial pode capturar relacionamentos monotônicos, embora também possa obscurecer efeitos heterogêneos e relações complexas resultantes das interações entre atributos (MOLNAR, 2019). *Individual Conditional Expectation curves* (GOLDSTEIN *et al.*, 2015) lidam com essa limitação ao desagregar as dependências parciais para visualizar o quanto a predição de uma instância individual se altera quando o valor de um atributo selecionado é alterado (CASALICCHIO; MOLNAR; BISCHL, 2019). Mapas de calor (*heatmaps*) são aplicados para explorar e destacar quais são os pontos de maior relevância para as unidades neurais em problemas de classificação de imagens (ZHANG; ZHU, 2018; LAPUSCHKIN *et al.*, 2019; ZEDNIK, 2021). No entanto, os *heatmaps* são difíceis de agregar, o que dificulta a detecção de falsos positivos (BHATT *et al.*, 2020).

Xenopoulos *et al.* (2022) desenvolveram GALE (*Globally Assessing Local Explanations*), uma metodologia baseada em TDA (*Topological Data Analysis*) (SINGH *et al.*, 2007) para extrair representações simplificadas de conjuntos de explicações locais. O método gera uma assinatura topológica dos relacionamentos entre o espaço de explicações e as predições

de modelo. A partir da inspeção visual dessas representações, é possível avaliar e refinar os parâmetros do método XAI subjacente ou comparar quantitativamente a similaridade entre diferentes técnicas de explicação. Neste sentido, GALE atua mais como uma ferramenta para avaliação visual de explicadores do que como um método explicador em si.

Vig (2019b) discute diversos trabalhos que introduzem metodologias para visualizar mecanismos de atenção em modelos de Processamento de Linguagem Natural (PLN). O BertViz (VIG, 2019a; VIG, 2019b) é uma ferramenta composta por três visualizações para explorar *Transformers* a partir de diferentes níveis: de atenção, de modelo e de neurônio. Com suporte do BertViz, Vig (2019b) demonstrou o caso em que um modelo baseado em atenção carregava viés de gênero. No entanto, o desempenho da ferramenta tende a se deteriorar ao trabalhar com entradas extensas ou modelos grandes. O BertViz apresenta os pesos do mecanismo de atenção em *heatmaps*, mas a abordagem pode levar a interpretações pouco claras. Além disso, experimentos contrafactuais tendem a obter *heatmaps* alternativos que resultam em predições equivalentes (JAIN; WALLACE, 2019), embora Wiegreffe e Pinter (2019) afirmem que o fato de existir mais de uma explicação não implica que estas não tenham sentido ou sejam falsas.

Garde, Kran e Barez (2023) desenvolveram o DeepDecipher, uma interface interativa para a visualização e interpretação de neurônios em camadas MLP de *Transformers*. O método apresenta informações sobre padrões de ativação dos neurônios, objetivando compreender quando e por que ocorre a ativação, apoiado por um banco de dados com sequências predefinidas e um procedimento para gerar gráficos a partir de *tokens* (FOOTE *et al.*, 2023). No entanto, a visualização de um neurônio pode não capturar seu comportamento geral, além do DeepDecipher não introduzir uma nova abordagem de explicabilidade.

Muitos métodos de explicabilidade baseados em visualização fazem uso ativo de técnicas de redução de dimensionalidade ou projeção multidimensional (NONATO; AUPETIT, 2018), para gerar representações interpretáveis dos espaços de características dos modelos de aprendizado, como relacionamentos entre os neurônios e a sua influência nos dados (HOHMAN *et al.*, 2018).

Cantareira, Etemad e Paulovich (2020) desenvolveram um método para representar o fluxo de dados de ativação nas camadas ocultas de uma RNA. Com isso, é possível verificar a evolução da rede durante o treinamento e as representações que são construídas quando uma camada envia informações para a próxima. Utilizar visualização durante o processo de treinamento permite acompanhar o modelo à medida que ele aprende, o que possibilita monitorar o seu desempenho (HOHMAN *et al.*, 2018). Rauber *et al.* (2016) apresentaram um método semelhante, explorando visualmente o modo como os neurônios artificiais transformam os dados de entrada à medida que estes passam pelas camadas ocultas de redes profundas.

O SUBPLEX (CHAN *et al.*, 2020b; YUAN *et al.*, 2022) é uma ferramenta interativa de análise visual que combina projeções multidimensionais e explicações de predições. O método gera agrupamentos que reduzem a complexidade visual e permitem a análise de grandes conjuntos de explicações locais. Com informações agregadas a nível de subpopulação, os usuários podem explorar interativamente os detalhes nas explicações, selecionando instâncias ou atributos, com o objetivo de identificar padrões e facilitar a comparação entre padrões locais de múltiplas subpopulações. O SUBPLEX não introduz uma nova técnica de explicabilidade, mas emprega métodos XAI e de projeção bem estabelecidos para gerar explicações que são agrupadas e exibidas em uma interface gráfica. Entretanto, o SUBPLEX demanda capacidade computacional para processamento em tempo real.

A t-SNE (MAATEN; HINTON, 2008), que é uma robusta técnica de projeção multidimensional, tem presença frequente entre os métodos baseados em visualização (TJOA; GUAN, 2020). Entretanto, as técnicas de redução de dimensionalidade têm um limite de usabilidade no que diz respeito ao número de pontos visualizados simultaneamente (RAUBER *et al.*, 2016; HOHMAN *et al.*, 2018). Logo, a explicabilidade por meio de visualizações enfrenta desafios de escalabilidade ao lidar com grandes quantidades de elementos, bem como o modo de representar apropriadamente os seus relacionamentos.

3.2.3 XAI Baseada em *Decision Boundaries*

Uma abordagem ainda pouco explorada na literatura, é a explicação do comportamento dos modelos de aprendizado por meio da investigação dos seus *decision boundaries* (bordas ou limites de decisão). Karimi, Derr e Tang (2019) desenvolveram o DeepDIG, um método baseado na geração de amostras adversariais suficientemente próximas do *decision boundary*, ou seja, instâncias sintéticas que estejam entre duas classes diferentes. Mais especificamente, o método atua sobre DNNs previamente treinadas e gera instâncias de borda, isto é, amostras com probabilidades de classificação para duas classes distintas tão próximas quanto possível, resultando em incerteza de classificação. Então, essas instâncias são utilizadas para medir a complexidade, ou a não-convexidade, do *decision boundary* gerado pelo espaço de características da DNN treinada.

Uma das propriedades fundamentais das DNNs é seu grande poder de generalização, que é alcançado por meio de sofisticadas combinações de transformações não-lineares. Com isso, as DNNs conseguem mapear dados com relacionamentos complicados e de alta dimensionalidade, o que acaba levantando uma questão: a complexidade dos dados no espaço de entrada se manifesta no espaço transformado pela rede?

No intuito de responder a essa questão, Karimi, Derr e Tang (2019) desenvolveram duas métricas para caracterizar a complexidade dos *decision boundaries*, uma métrica para o espaço original dos dados de entrada e a outra específica para o espaço transformado pela

rede. Com essas ferramentas, foi possível confirmar a hipótese levantada por Li, Ding e Gao (2018), de que o *decision boundary* resultante da última camada de uma DNN treinada com *Backpropagation* converge para a solução de uma SVM linear, treinado sobre os dados transformados. Ainda neste contexto, Guan e Loew (2020) desenvolveram uma métrica para avaliar a complexidade dos *decision boundaries*, alegando que os modelos que geram bordas de decisão mais simples apresentam melhor capacidade de generalização. Englhardt *et al.* (2020) apresentaram um método para encontrar uma amostragem mínima com densidade (quase) uniforme que contenha pontos de borda, a partir dos dados originais. A técnica é baseada em um problema de otimização restrito que retém as amostras necessárias para garantir a delimitação de *decision boundaries* corretos.

Yousefzadeh e O'Leary (2019a) calcularam os *flip points*, que são pontos suficientemente próximos do *decision boundary* de modelos treinados, capazes de serem classificados em ambas as classes (considerando redes neurais para predição binária). Os autores levantam questões interessantes como: os *flip points* podem ser utilizados para determinar a mínima mudança em um dado de entrada, suficiente para alterar uma predição; instâncias de dado classificadas incorretamente tendem a apresentar menores distâncias em relação a um *flip point* do que as instâncias classificadas corretamente; pontos próximos aos seus *flip points* são mais influentes em definir as bordas de decisão entre classes; utilizar *flip points* como dados sintéticos durante o treinamento pode melhorar a acurácia do modelo quando este apresentar viés. *Flip points* existem para qualquer modelo de aprendizado e, se devidamente confirmadas, questões como estas podem transformar os *flip points* em elementos-chave na verificação da confiança de uma predição (YOUSEFZADEH; O'LEARY, 2019a).

3.2.4 XAI Baseada em Exemplos Contrastivos e Contrafactuais

O conceito vem das Ciências Sociais, que estabelecem que as explicações humanas decorrem de processos essencialmente contrastivos (JACOVI *et al.*, 2021). Explicações contrastivas esclarecem o motivo de um evento ter ocorrido em contraste a outro. A propriedade da contrastividade prevê que uma explicação responda à questão do porquê um evento aconteceu em termos das possíveis causas (alternativas hipotéticas) que estão relacionadas a este evento, por exemplo: "por que o evento A aconteceu em vez do evento B?"; uma explicação "razoável" citaria as motivações causais que levaram o modelo ao evento A (STEPIN *et al.*, 2021).

É possível descrever diferentes cenários para uma predição caso ocorressem ligeiras mudanças nos dados de entrada, isto é, explicar as possíveis consequências contrafactuais acarretadas a partir dessas alterações. Explicações contrafactuais têm uma longa história na filosofia e psicologia, que consideram que os processos de racionalização humanos seguem padrões de dependência contrafactual (VERMA; DICKERSON; HINES, 2020). Em XAI,

explicações contrafactuais identificam pontos próximos aos dados de entrada que alteram a saída do classificador (BHATT *et al.*, 2020), descrevendo quais características mudariam a predição, de acordo com alguma perturbação, deleção ou inclusão de informações nos atributos preditivos (LIAO; GRUEN; MILLER, 2020). As explicações contrafactuais não respondem explicitamente o "porquê" um modelo tomou uma certa decisão, em vez disso, o objetivo é descrever uma ligação entre o que poderia ter acontecido caso as entradas fossem alteradas de um modo específico, fornecendo orientações para alcançar algum resultado desejado (VERMA; DICKERSON; HINES, 2020; MISHRA *et al.*, 2021).

Poyiadzi *et al.* (2020) desenvolveram uma abordagem para gerar explicações contrafactuais por meio da construção de um grafo ponderado sobre as instâncias de dados. Em seguida, o método aplica o algoritmo de Dijkstra para encontrar o caminho mais curto entre as instâncias que geraram as explicações. Isso é feito de acordo com métricas ponderadas por densidade e requisitos definidos pelos usuários, apresentando sugestões sobre as mudanças nas entradas que poderiam levar aos resultados desejados. Raimundo, Nonato e Poco (2022) apresentaram o MAPOCAM (*Model-Agnostic Pareto-Optimal Counterfactual Antecedent Mining*), um algoritmo que utiliza otimização multi-objetivo para determinar explicações contrafactuais. Os atributos de entrada do modelo são tratados como funções-objetivo, aplicadas em um mecanismo baseado em árvores que busca as mudanças necessárias que resultam em antecedentes contrafactuais. A otimização multi-objetivo ainda é pouco explorada entre abordagens XAI contrafactuais, mas o MAPOCAM é caro computacionalmente, além de não oferecer suporte a dados categóricos.

As explicações contrafactuais têm um apelo devido ao seu alinhamento com o raciocínio humano (MILLER, 2019; BAROCAS; SELBST; RAGHAVAN, 2020). No entanto, Bhatt *et al.* (2020) argumentam que muitos dos métodos atualmente existentes fazem aproximações grosseiras, dado que determinar explicações contrafactuais plausíveis (viáveis tanto no contexto dos dados como no mundo real) não é algo trivial. Além disso, as implementações contrafactuais disponíveis são *model-specific* ou funcionam apenas para modelos que não são caixas-preta por natureza.

Em geral, as abordagens contrastiva e contrafactual são similares e guardam pequenas diferenças entre si, com ambas buscando por mudanças mínimas nos dados de entrada que alterem o contexto de uma predição, com as contrastivas sendo mais restritivas em relação às explicações contrafactuais (WACHTER; MITTELSTADT; RUSSELL, 2017). Para mais informações a respeito do tema, Verma, Dickerson e Hines (2020) revisaram e categorizaram diversas pesquisas sobre XAI contrafactual, elencando propriedades desejáveis, avaliando as vantagens, desvantagens e questões em aberto entre as diferentes abordagens propostas atualmente. O leitor interessado também pode consultar a pesquisa de Stepin *et al.* (2021), em que os autores elaboraram uma extensa revisão na literatura, desde as características

teóricas, distinções, até o estado da arte em XAI contrastiva e contrafactual.

3.2.5 XAI Baseada em Decomposição de Sinais e Gradientes

Os métodos baseados em gradientes computam gradientes para explicar as predições de modelos de aprendizado. Esses métodos atribuem importância aos atributos de entrada ao verificar a quantidade de mudanças que pequenas perturbações nestes dados impactam na saída do modelo. *Vanilla Gradient* (SIMONYAN; VEDALDI; ZISSERMAN, 2013) calcula diretamente os gradientes da saída do modelo relativos aos atributos de entrada sem quaisquer modificações. Apesar da sua simplicidade, essa abordagem sofre com a falta sensibilidade mais refinada quanto a ruídos nos gradientes.

Bach *et al.* (2015) propuseram o LRP (*Layer-wise Relevance Propagation*), que gera explicações por meio da decomposição de predições de modelos não-lineares complexos, como as RNA, em termos das suas variáveis de entrada. O método é fundamentado em DTD (*Deep Taylor Decomposition*) (SCHRECKENBERGER; BARTELT; STUCKENSCHMIDT, 2019; KAUFFMANN; MÜLLER; MONTAVON, 2020) e assume que o modelo de aprendizado pode ser decomposto em camadas de computação para atribuir um vetor de *scores* de relevância sobre as variáveis (de entrada ou neurônios intermediários). O LRP identifica as propriedades que são pivotais em relação ao estado de máxima incerteza das predições, redistribuindo a função de predição em sentido contrário, por meio da projeção de sinais a partir da camada de saída até chegar na camada de entrada, utilizando um mecanismo de *backpropagation* aplicado uniformemente a todos os parâmetros do modelo (LAPUSCHKIN *et al.*, 2019; HAMILTON *et al.*, 2021).

O LRP é considerado um método *model-agnostic* pois não faz restrições *a priori* a respeito de algoritmos de aprendizado específicos. Entretanto, o LRP foi projetado como um conceito geral para operar sobre arquiteturas de modelos caixa-preta baseados em *kernels* não-lineares, como as Redes Neurais e *Bag of Words*, o que inclui diversas modelagens relacionadas com tarefas de classificação de imagens (BACH *et al.*, 2015). O LRP tem alcançado sucesso em gerar intuições e valores mensuráveis que descrevem o processamento das variáveis em redes neurais, devido ao seu método de redistribuição seguir princípios de conservação de relevância e decomposição proporcional, que mantém forte conexão com a saída do modelo preditor (KOHLBRENNER *et al.*, 2020).

Lapuschkin *et al.* (2019) aplicaram um agrupamento espectral sobre os vetores de *scores* do LRP para identificar padrões e comportamentos atípicos em modelagens de aprendizado. Este trabalho demonstrou a existência de um viés oculto presente no conjunto de treinamento e que foi incorporado pelo modelo de aprendizado. Diversas imagens utilizadas no treinamento continham uma marcação com um endereço *URL* em uma das classes. Então, novas imagens com conteúdo não associado a esta classe, mas manipuladas

para conter uma *URL*, eram classificadas erroneamente pelo modelo. Lapuschkin *et al.* (2019) ainda utilizou *heatmaps* (mapas de calor) com a mesma dimensionalidade dos dados de entrada para interpretar visualmente os *scores* de relevância. Kohlbrenner *et al.* (2020) apresentaram uma revisão sistemática, avaliando e quantificando os efeitos das principais abordagens baseadas em LRP aplicadas sobre RNAs.

SmoothGrad (SMILKOV *et al.*, 2017) é um aprimoramento do *Vanilla Gradient*, que gera múltiplas amostras adicionando pequenos níveis de ruído na instância de interesse. O método calcula os gradientes por meio da média dessas instâncias perturbadas, tornando as explicações menos sensíveis ao ruído nos dados de entrada. *Input* × *Gradient* (SHRIKUMAR *et al.*, 2016) calcula o produto elemento a elemento dos gradientes de saída do modelo com as respectivas entradas. Enquanto o *Integrated Gradients* (SUNDARARAJAN; TALY; YAN, 2017) é semelhante às demais abordagens baseadas em gradientes, diferindo ao determinar um conjunto de amostras interpoladas entre a instância a ser explicada e uma instância-base (geralmente contendo valores neutros que podem vir da média). Então o *Integrated Gradients* calcula os gradientes das amostras e os integra da instância-base até a entrada sob explicação.

Shrikumar, Greenside e Kundaje (2017) desenvolveram o DeepLIFT (*Deep Learning Important FeaTures*), baseado no conceito de *scores* de importância derivado do LRP. O DeepLIFT é voltado para Redes Neurais Profundas (DNNs), propagando atribuições em cada camada do modelo profundo para comparar a diferença entre a ativação de um neurônio em relação a uma "ativação de referência" (valor base). Mais especificamente, o método calcula os multiplicadores dos neurônios em uma camada em relação aos seus sucessores imediatos (neurônios alvo) utilizando o algoritmo de *backpropagation*, algo que é semelhante à aplicação da regra da cadeia (STEWART; CLEGG; WATSON, 2020) para derivadas parciais. Segundo Lundberg e Lee (2017), essa composição pode ser vista como uma linearização dos componentes não-lineares da Rede Neural.

Nesse contexto, os autores do DeepLIFT também definiram o *Rescale rule* como uma melhoria da regra da cadeia no cálculo de gradientes em relação à saída do *backpropagation*. Porém, a escolha da ativação de referência é feita heuristicamente, com os autores deixando questões importantes em aberto, como a computação empírica de uma boa referência, além de como propagar as importâncias além de simplesmente aplicar gradientes (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017).

3.2.6 XAI Baseada em Simplificação

A explicabilidade por simplificação compreende as técnicas em que um novo modelo é construído tendo por base o modelo a ser explicado, já treinado (ARRIETA *et al.*, 2020). O objetivo do modelo simplificado é ter um comportamento semelhante ao original,

porém, com menor complexidade, ou seja, o modelo simplificado deve ter um desempenho preditivo similar ao original, mas a partir de estruturas mais transparentes. Thiagarajan *et al.* (2016) desenvolveu o TreeView, uma ferramenta para interpretação visual de modelos complexos. A abordagem identifica fatores discriminatórios entre as classes dos dados de entrada, utilizando eliminação sequencial por meio do particionamento hierárquico do espaço de atributos, agrupando as instâncias de acordo com cada fator discriminatório de modo que as associações não desejadas são descartadas.

Entre as técnicas de explicabilidade mais conhecidas atualmente, encontra-se o LIME (*Local Interpretable Model-Agnostic Explanations*) (RIBEIRO; SINGH; GUESTRIN, 2016c). Este método determina um modelo interpretável linear que aproxima localmente o modelo original. Para tal, LIME gera uma vizinhança sintética ao redor da instância a ser explicada, por meio de perturbações sobre instâncias do próprio conjunto de dados. As amostras geradas são classificadas utilizando o modelo de aprendizado original, ponderando essas amostras de acordo com a sua proximidade em relação ao ponto sob explicação, aplicando um *kernel* de ponderação. Então, LIME determina um modelo linear sobre a vizinhança ao minimizar uma função de não-fidelidade e, por fim, as predições são explicadas por meio da interpretação deste modelo linear.

Mais especificamente, seja f o modelo de aprendizado treinado, $g \in G$ um modelo interpretável, com G sendo uma classe de modelos potencialmente interpretáveis como a regressão linear, por exemplo. Para uma instância n-dimensional $\mathbf{x} = (x_1, \ldots, x_n)$ a ser explicada, o modelo interpretável g é determinado pela minimização da função de perda $\mathcal{L}$ do seguinte modo:

$$\mathcal{E}(\mathbf{x}) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_{\mathbf{x}}) + \Omega(g) \tag{3.4}$$

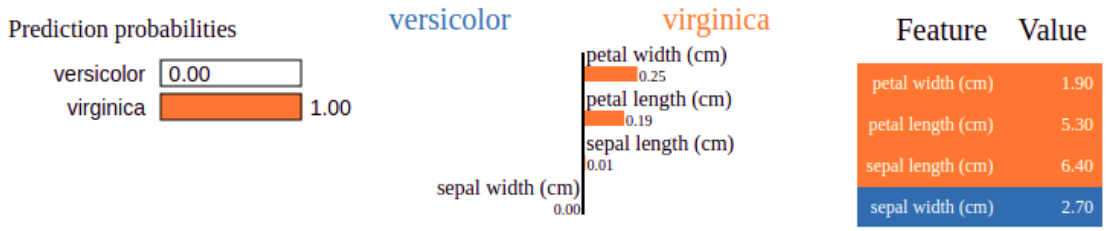
com $\pi_{\mathbf{x}}$ sendo o *kernel* de ponderação centrado em $\mathbf{x}$ responsável por manter a fidelidade local e Ω um termo de complexidade (que deve ser mantido baixo) aplicado sobre g.

Alguns autores também classificam o LIME como um *Local Surrogate Model* (Modelo de Substituição Local), que é a classe de métodos utilizados para explicar predições individuais aplicando um modelo substituto treinado localmente (MOLNAR, 2019). Logo, o modelo substituto deve ser um bom aproximador para as predições do modelo original, localmente, mas não necessariamente precisa ser também uma boa aproximação global, propriedade esta conhecida como fidelidade local. A Figura 3.2 apresenta um exemplo de explicação gerada pelo LIME sobre o bem-conhecido conjunto de dados Iris (FISHER, 1988) que, por simplificação, foi adaptado para conter apenas duas classes. Para tarefas de classificação binária, o LIME apresenta explicações utilizando duas cores (laranja e azul, neste caso).

O LIME utiliza uma interface gráfica simples e informativa. Na Figura 3.2a estão as

Figura 3.2 – Explicação gerada pelo LIME sobre uma amostra do conjunto Iris classificada como pertencente à classe *virginica*.

(a) Probabilidades de classificação. **(b)** Os atributos mais importantes. **(c)** Valores dos atributos mais importantes.



Fonte: Ortigossa, Gonçalves e Nonato (2024).

probabilidades de classificação da instância sob investigação; a Figura 3.2b apresenta os atributos que mais contribuíram para a predição, com os valores nas barras horizontais informando as importâncias determinadas pelo LIME; a Figura 3.2c lista os valores dos atributos. As cores são consistentes entre os gráficos e refletem as contribuições dos atributos para a classificação, ou seja, atributos em laranja suportam a classe *virginica*, enquanto aqueles em azul suportam a classe *versicolor*. No entanto, não é claro como cada atributo pode contribuir positiva ou negativamente no resultado apresentado pelo método. Os autores ainda desenvolveram duas extensões do LIME (RIBEIRO; SINGH; GUESTRIN, 2016a; RIBEIRO; SINGH; GUESTRIN, 2016b), o que inclui uma versão aprimorada que fornece explicações textuais mais esclarecedoras (RIBEIRO; SINGH; GUESTRIN, 2018).

Entre suas principais vantagens de aplicação, é possível citar a flexibilidade do LIME. Mesmo trocando o modelo de aprendizado aplicado sobre um determinado conjunto de dados, ainda é possível gerar explicações para esses dados utilizando o modelo local interpretável gerado pelo LIME. O método é um dos poucos que opera em contextos de dados tabulares, textuais ou imagens (MOLNAR, 2019). Em classificação de imagens, LIME forma vizinhanças segmentando a imagem de entrada em *superpixels* e perturbando a imagem original segmentada, mediante a troca aleatória de *superpixels* por uma cor de fundo. Então os estados booleanos desses *superpixels* são aplicados para compor as variáveis do modelo linear (HAMILTON *et al.*, 2021). Esta estratégia pode ser estendida para habilitar outros métodos XAI a trabalhar com imagens.

Embora seja considerado uma das grandes soluções em XAI, o LIME possui algumas limitações significativas. O método é estável ao explicar classificadores lineares, mas é instável nos demais casos, com as explicações podendo mudar completamente apenas rodando o código algumas vezes (AMPARORE; PEROTTI; BAJARDI, 2021). Segundo Aas, Jullum e Løland (2021), LIME não garante efeitos perfeitamente distribuídos entre as variáveis. Além disso, modelos distintos descrevendo perfis de aproximação diferentes podem servir como aproximador dos dados amostrados, com o LIME escolhendo um desses modelos

aleatoriamente e sem oferecer garantias de que o modelo escolhido seja de fato o melhor aproximador local. Além disso, não existem garantias teóricas sólidas indicando que um modelo de substituição local simples seja adequadamente representativo de modelos mais complexos (ADADI; BERRADA, 2018).

Definir uma vizinhança significativa ao redor de uma instância de interesse é uma tarefa complexa. LIME contorna essa dificuldade construindo a vizinhança ao redor do ponto sob explicação, em relação ao centro de massa dos dados de treinamento. Essa estratégia pode contribuir com a instabilidade da técnica ao gerar amostras consideravelmente distintas do ponto de interesse, embora isso aumente a probabilidade de que o método aprenda ao menos uma explicação (MOLNAR, 2019). Também, LIME se baseia em suposições um tanto simplistas demais com relação às bordas de decisão dos modelos de aprendizado, ao assumir que elas são localmente lineares. No entanto, as bordas de decisão de modelos como as redes neurais, por exemplo, podem ser altamente não-lineares, mesmo localmente, e uma aproximação linear neste contexto pode levar a explicações pouco confiáveis (YOUSEFZADEH; O'LEARY, 2019b).

Existe uma certa falta de significado comparativo para os valores de saída do LIME. Em outras palavras, não é simples de compreender o que os números atribuídos para cada variável representam, ou o relacionamento que esses valores têm com a predição. Também, o modo de ponderação linear do LIME aumenta a influência de amostras não perturbadas (HAMILTON *et al.*, 2021). Não há um bom modo para se estimar o *kernel* de ponderação ou mesmo uma razão de largura apropriada para este *kernel* (MOLNAR, 2019). Assim, parâmetros importantes como o *kernel*, o tamanho da vizinhança e o termo de complexidade são escolhidos heuristicamente pelo método, algo que afeta a fidelidade local e leva a comportamentos inconsistentes (LUNDBERG; LEE, 2017; AMPARORE; PEROTTI; BAJARDI, 2021).

3.2.7 XAI Baseada em *Shapley Values*

Derivados de uma clássica modelagem da Teoria dos Jogos, os *Shapley values* (SHAPLEY, 1953) descrevem um modo de distribuir (compartilhar) os ganhos (custos) totais de um jogo cooperativo entre os jogadores, satisfazendo critérios de justiça (KUMAR; CHANDRAN, 2021). Logo, determinar *Shapley values* é um problema do tipo *cost-sharing* (partilhamento de custos) (SUNDARARAJAN; NAJMI, 2020). Problemas de partilhamento de custos são questões centrais em diversas áreas em que é necessário dividir custos conjuntos e alocar parcelas deste custo de modo proporcional entre cada um dos indivíduos participantes (FRIEDMAN; MOULIN, 1999). Por exemplo, a eletricidade é uma utilidade pública com uma longa cadeia de produção que, de modo simplificado, inicia nas unidades geradoras, passa pelas transmissoras e distribuidoras até chegar ao consumidor final.

Determinar o preço que será pago pelo consumidor e distribuir esse valor de modo justo entre cada elo da cadeia produtiva, é um típico problema de partilhamento de custos.

Um *Shapley value* representa a contribuição marginal média de um jogador, avaliada sobre todas as possíveis combinações de jogadores, ou seja, trata-se de uma média ponderada das contribuições individuais em relação a todas as possíveis composições de indivíduos (MOLNAR, 2019). Uma das características da abordagem reside em sua sólida fundamentação teórica, que garante axiomaticamente uma justa distribuição de ganhos (custos) entre os participantes de um jogo colaborativo. Segundo Kumar *et al.* (2020), um jogo colaborativo é composto por um conjunto de n jogadores e uma função característica v, que mapeia subconjuntos $S \subseteq \{1, \ldots, n\}$ em valores reais, satisfazendo $v(\varnothing) = 0$. A função característica descreve quanto do ganho resultante pode ser atribuído individualmente aos jogadores por cooperarem no jogo. Assim, os *Shapley values* representam um modo de distribuir o valor total de uma cooperação, $v(\{1, \ldots, n\})$, entre os indivíduos.

Considere $v(i)$ a função característica aplicada sobre o atributo i (um jogador) de um subconjunto S (jogadores/atributos), isto é, $i \in S$. Então, o *Shapley value* é calculado como uma média ponderada das contribuições marginais de i em relação aos possíveis subconjuntos $S \subseteq \{1, \ldots, n\}$ e o número de permutações em S:

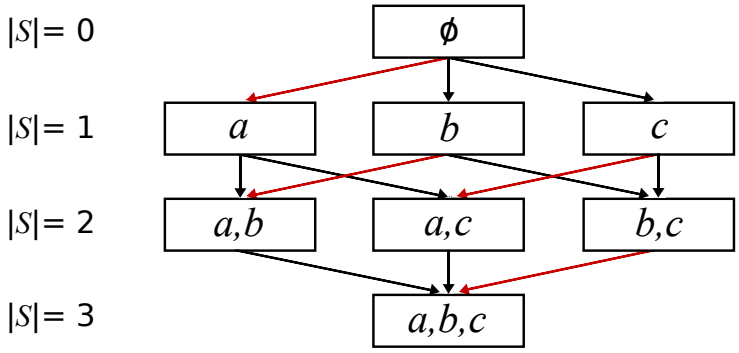
$$\phi_v(i) = \sum_{S \subseteq n \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)) \tag{3.5}$$

em que $\phi_v(i)$ é o *Shapley value* do atributo i, $v(S)$ é o valor esperado da função característica condicional a subconjuntos $S \subseteq \{1, \ldots, n\}$, ou seja, $\mathbb{E}[v(S)]$, n representa a quantidade total de atributos, e $| \, . \, |$ denota cardinalidade (KUMAR; CHANDRAN, 2021). Assim, $v(S \cup \{i\}) - v(S)$ descreve a contribuição marginal de um atributo (jogador) em relação a uma combinação de atributos (jogadores), ou seja, é a variação $\Delta_v(i, S)$ gerada ao se incluir i em S (KUMAR *et al.*, 2020).

A Figura 3.3 ilustra o conceito por trás do cálculo dos *Shapley values* sobre um conjunto com três atributos, $\{a, b, c\}$. Cada possível combinação de atributos deve ser considerada para verificar a contribuição individual de cada atributo, isto é, todos possíveis subconjuntos S, com $|S|$ variando de 0 a n (neste exemplo, $n = 3$).

Cada vértice da Figura 3.3 retrata uma combinação de atributos e cada aresta corresponde à inclusão de um atributo que não estava presente na combinação anterior. Logo, a abordagem original para calcular *Shapley values* requer verificar 2^n combinações (que é o número total de possíveis subconjuntos de $\{1, \ldots, n\}$), tornando a sua computação exata um problema NP-difícil (conforme n cresce, a Equação 3.5 se torna computacionalmente inaplicável). Entretanto, calcular um *Shapley value* em cenários contendo poucos atributos é relativamente simples. Considere o exemplo da Figura 3.3. O *Shapley value* do atributo

Figura 3.3 – Diagrama com as possíveis combinações para um conjunto contendo três atributos.



Fonte: Ortigossa, Gonçalves e Nonato (2024).

a é computado verificando os custos marginais entre todas as combinações de atributos $S \subseteq n \backslash \{a\}$, que levem até um subconjunto que contenha o atributo a que, neste caso, são: $\{\varnothing\}$, $\{b\}$, $\{c\}$ e $\{b, c\}$. As setas vermelhas da Figura 3.3 indicam os caminhos entre combinações com o atributo a e aquelas sem este atributo, ou seja, as contribuições marginais ao se incluir a. Aplicando a Equação 3.5, obtém-se a seguinte configuração de contribuições marginais e fatores de ponderação:

$$\begin{aligned} \phi_v(a) = & \frac{0!\,(3-0-1)}{3!} \times \Delta_v(a, \{\varnothing\}) + \\ & \frac{1!\,(3-1-1)}{3!} \times \Delta_v(a, \{b\}) + \\ & \frac{1!\,(3-1-1)}{3!} \times \Delta_v(a, \{c\}) + \\ & \frac{2!\,(3-2-1)}{3!} \times \Delta_v(a, \{b, c\}) \\ = & \frac{1}{3}\Delta_v(a, \{\varnothing\}) + \frac{1}{6}\Delta_v(a, \{b\}) + \frac{1}{6}\Delta_v(a, \{c\}) + \frac{1}{3}\Delta_v(a, \{b, c\}) \end{aligned} \tag{3.6}$$

com $\Delta_v(a, S)$ sendo a contribuição marginal de a condicional ao subconjunto de atributos S. Observe que um *Shapley value* não é somente a diferença no resultado da predição quando se remove um atributo do modelo, mas sim uma soma ponderada de custos marginais (MOLNAR, 2019).

Lundberg e Lee (2017), entre outros autores também dedicados ao estudo da temática (FRIEDMAN; MOULIN, 1999; KUMAR *et al.*, 2020; SUNDARARAJAN; NAJMI, 2020), citam as seguintes propriedades esperadas para uma solução de problema do tipo partilhamento de custos, satisfeitos pelos *Shapley values*:

- ***Accuracy*** – O somatório dos *Shapley values* de todos os atributos equivale ao valor total da cooperação, $\sum_i^n \phi_v(i) = v(\{1, \ldots, n\})$;
- ***Missingness*** – Para um atributo i, se $\Delta_v(i, S) = 0$ para todos os subconjuntos S, então este atributo não refletirá impacto no resultado do modelo, ou seja, $\phi_v(i) = 0$;

- ***Consistency*** – Para um atributo i e uma função característica não-decrescente v, a contribuição $\phi_v(i)$ somente deveria aumentar se o valor em i aumentar, fixados os valores de todos os demais atributos. Isso significa que se v é monótona em i, $\phi_v(i)$ aumenta se o valor de i aumentar;
- ***Additivity*** – Para um atributo i e duas funções característica, v e t, $\phi_v(i) + \phi_t(i) = \phi_{v+t}(i)$, em que $(v + t)(S) = v(S) + t(S)$. Esse axioma permite a soma aritmética;
- ***Symmetry*** – Para dois atributos i e j, se $\Delta_v(i, S) = \Delta_v(j, S)$ para qualquer subconjunto S, então as contribuições de i e j devem ser iguais, $\phi_v(i) = \phi_v(j)$.

No contexto do Aprendizado de Máquina, os *Shapley values* quantificam, para cada atributo, o valor da mudança na predição esperada quando o modelo de aprendizado é condicionado a combinações deste atributo (LUNDBERG; LEE, 2017). Note que o modelo é aplicado diversas vezes durante a quantificação, porém, sob configurações equivalentes no que diz respeito aos hiperparâmetros e dados de treinamento (que é o conjunto completo). A diferença está nas combinações de variáveis incluídas em cada etapa. A modelagem já tem uma longa história de aplicação, com Lipovetsky e Conklin (2001) utilizando *Shapley values* para analisar a importância global de variáveis em modelos de regressão linear.

Mais especificamente, um *Shapley value* descreve um valor de importância para uma variável de entrada, representando a contribuição sobre a predição ao se incluir este atributo no modelo. Para calcular essa atribuição aplicando a Equação 3.5, utiliza-se o modelo de aprendizado como função característica, aplicado em subconjuntos com e sem o atributo de interesse, extraindo uma média ponderada entre as diferenças (KUMAR; CHANDRAN, 2021). Como todas as combinações de atributos devem ser verificadas para obter os *Shapley values* referentes a todos os atributos do conjunto de dados, o custo computacional cresce exponencialmente conforme o número de atributos aumenta, inviabilizando o uso da abordagem em modelagens que operam dados de alta dimensionalidade (MOLNAR, 2019).

Para contornar essa limitação, Štrumbelj e Kononenko (2014) desenvolveram uma versão aproximada baseada em amostragem de Monte Carlo. A estimativa por amostragem da Equação 3.5 considera que as predições são geradas a partir de instâncias (amostras) contendo seleções aleatórias de atributos (exceto o atributo sob investigação). Com isso, estima-se iterativamente o *Shapley value* de cada atributo a partir de amostras também selecionadas aleatoriamente, em cada iteração. As variações nas predições são ponderadas, para cada amostragem, de acordo com a distribuição de probabilidade dos dados, com o resultado sendo computado em média. O procedimento é repetido para cada atributo para estimar todos os *Shapley values* (MOLNAR, 2019).

Lundberg e Lee (2017) desenvolveram o SHAP (*SHapley Additive exPlanations*), uma das abordagens baseadas em *Shapley values* de maior sucesso atualmente. SHAP

estima os *Shapley values* a partir de uma aproximação do modelo de aprendizado original, utilizando uma função de expectativa condicional sobre vetores de combinações de atributos simplificados. Com isso, SHAP avalia o ganho (ou a perda) em uma predição simulando a presença e a ausência de atributos por meio da amostragem de valores da distribuição marginal de cada atributo. Observe que a expectativa condicional é o estimador mais comum que sumariza a distribuição de probabilidades em aplicações de predição. Porém, SHAP assume a independência entre os atributos, possibilitando que a distribuição condicional seja trocada pela distribuição marginal (AAS; JULLUM; LØLAND, 2021), o que facilita a aproximação das expectativas condicionais para, então, estimar diretamente os *Shapley values* utilizando a estratégia de *Additive Feature Attribution* (atribuição de importâncias aditivas).

Os métodos de explicabilidade aditivos atribuem um efeito para cada atributo (cf. Equação 3.3), sendo que o somatório dos efeitos de todas as *feature attributions* (atribuições de importâncias) aproxima a predição original do modelo (LUNDBERG; LEE, 2017). Note a correspondência entre os problemas de partilhamento de custos e de atribuição: os *Shapley values* distribuem os custos de um jogo entre os jogadores; a função característica é análoga ao modelo de aprendizado; os jogadores aos atributos; e os partilhamentos de custos às atribuições de importância (SUNDARARAJAN; NAJMI, 2020).

Formalmente, seja f um modelo de aprendizado qualquer, g um modelo interpretável de explicação e $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$ uma instância n-dimensional a ser explicada. SHAP define um mapeamento para a instância original, $\mathbf{x} = h_{\mathbf{x}}(\mathbf{x}')$, tal que $g(\mathbf{x}') \approx f(h_{\mathbf{x}}(\mathbf{x}'))$, com $\mathbf{x}'$ sendo uma simplificação de $\mathbf{x}$, ou seja, $\mathbf{x}' \approx \mathbf{x}$. Mesmo que a instância simplificada contenha menos informação do que a original, a função de mapeamento $h_{\mathbf{x}}$ garante que não ocorrerá perda significativa de informação (LUNDBERG; LEE, 2017). Então, SHAP gera um modelo de explicação g aproximando localmente o modelo original f ao determinar os *Shapley values* para cada atributo de $\mathbf{x}$, de modo aditivo, isto é:

$$f(\mathbf{x}) = g(\mathbf{x}') = \phi_0 + \sum_{i \in n} \phi_i x'_i \tag{3.7}$$

com ϕ_0 representando o valor esperado da predição, $\mathbb{E}[f(\mathbf{X})]$, e ϕ_i caracterizando o *Shapley value* para a variável x_i, calculado por meio da Equação 3.5, em que o ganho marginal é estimado por $f(h_x(\mathbf{x}')) = f(\mathbf{x})$ como função característica.

A Equação 3.7 significa que o SHAP aproxima f por meio de uma modelagem aditiva linear g, tornando possível estimar localmente o valor da predição de f baseado nos *Shapley values* dos atributos da respectiva instância como parâmetros do modelo linear g. Porém, pode ser demasiado custoso aplicar a Equação 3.5 diretamente devido à grande quantidade de combinações necessárias. Então, na prática, SHAP também utiliza

a estratégia de aproximação por amostragem baseada em uma versão de permutação da equação clássica de Shapley, com amostragens realizadas separadamente para cada atribuição de importância (LUNDBERG; LEE, 2017).

De acordo com seus autores, SHAP estima localmente a contribuição de cada atributo respeitando os axiomas ***local accuracy*** (precisão/fidelidade local), ***missingness*** (irrelevância) e ***consistency*** (consistência). Entretanto, SHAP também pode ser aplicado para estimar a importância de variáveis a nível global (KUMAR; CHANDRAN, 2021). Diferente do LIME (RIBEIRO; SINGH; GUESTRIN, 2016c), SHAP permite explicações contrastivas por meio da comparação de uma predição sob o contexto de subconjuntos de instâncias ou mesmo de uma única instância, em vez de apenas comparar as predições com a predição média de todo o conjunto de dados (MOLNAR, 2019; KUMAR *et al.*, 2020).

Observe que ao transformar a explicação em um modelo linear aditivo, SHAP permite a conexão entre os *Shapley values* e o LIME (MOLNAR, 2019). Então, Lundberg e Lee (2017) desenvolveram o KernelSHAP, uma variação do SHAP baseada em um conceito semelhante ao do LIME. Embora a formulação do LIME seja diferente da formulação dos *Shapley values*, tanto o LIME quanto o SHAP são métodos de atribuição de importância de natureza aditiva. Porém, lembre-se que o LIME escolhe heuristicamente o *kernel* de ponderação, a função de perda e o termo de complexidade, algo que viola o axioma da consistência e afeta a fidelidade local, resultando em explicações com comportamentos instáveis (LUNDBERG; LEE, 2017; MOLNAR, 2019). De fato, a Equação 3.5 é uma diferença de médias. Como a média é o melhor ponto de mínimos quadrados estimado para um conjunto de dados, é possível encontrar um *kernel* de ponderação utilizando o método dos mínimos quadrados. Com isso, Lundberg e Lee (2017) demonstraram como determinar o *kernel* de ponderação, uma função de perda local e o termo de complexidade (cf. Equação 3.4) sob o contexto dos *Shapley values*.

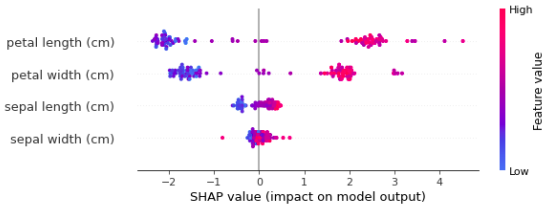
Dada a formulação linear do LIME, torna-se possível estimar os *Shapley values* utilizando o KernelSHAP em soluções baseadas em regressão, algo computacionalmente mais eficientes do que calcular diretamente a equação clássica dos *Shapley values*. Vale destacar que SHAP e LIME explicam as coisas de modos essencialmente diferentes. Ambos os métodos comparam a predição a ser explicada com uma probabilidade média. Entretanto, o SHAP verifica a diferença entre a predição e o valor esperado da predição média global, enquanto o LIME explica a diferença entre a predição e uma predição média local (dada pela amostragem de vizinhança) (AAS; JULLUM; LØLAND, 2021).

SHAP emerge não apenas como um dos principais métodos de *feature importance*, mas de XAI como um todo, devido às características axiomáticas vantajosas derivadas dos *Shapley values* (AMPARORE; PEROTTI; BAJARDI, 2021). A biblioteca *Python* SHAP oferece uma ampla gama de métodos além de um rico conjunto de ferramentas gráficas para

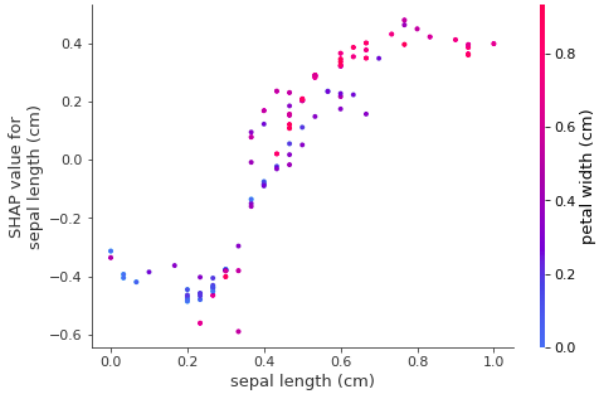
análise e visualização de informações. A Figura 3.4 apresenta as principais ferramentas de visualização oferecidas pelo pacote SHAP. Para gerar os gráficos da Figura 3.4, foi utilizado o conjunto de dados Iris que, por simplificação, foi adaptado para conter apenas duas classes: *versicolor* e *virginica*. Esta configuração de dados é idêntica ao exemplo ilustrando para o LIME (cf. Figura 3.2).

Figura 3.4 – Ferramentas gráficas do SHAP aplicadas em todo o conjunto Iris (global) e também sobre uma amostra classificada como pertencente à classe *virginica* (local).

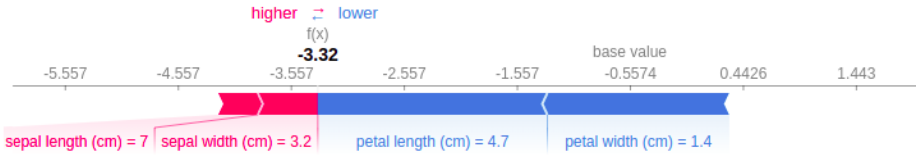
(a) SHAP *summary plot* (global).



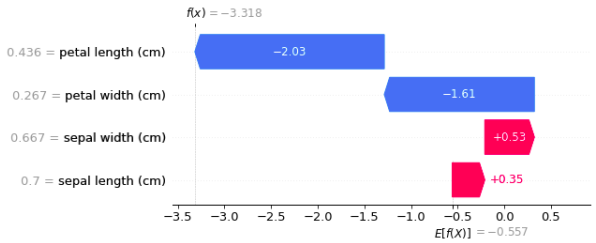
(b) SHAP *dependence plot* (global).



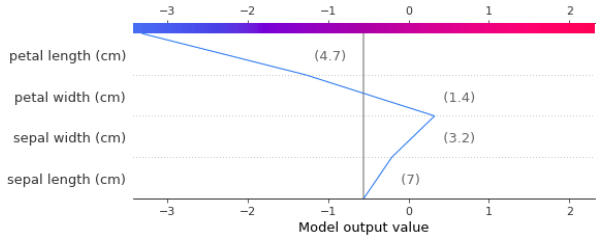
(c) SHAP *force plot* (local).



(d) SHAP *waterfall plot* (local).



(e) SHAP *decision plot* (local).



Fonte: Ortigossa, Gonçalves e Nonato (2024).

Na Figura 3.4a, são geradas explicações para todas as instâncias do conjunto de dados, sumarizando a densidade das importâncias. As variáveis são indicadas na lateral esquerda do gráfico e ordenadas verticalmente, conforme as suas respectivas importâncias médias globais. Cada ponto representa o *Shapley value* atribuído para uma variável pertencente a uma instância. Os pontos são distribuídos horizontalmente segundo a magnitude dos *Shapley values*. Quanto mais distante do zero no sentido positivo, mais importante é a variável dentro da classe em que a instância foi predita. Valores mais distantes de zero para o lado negativo, indicam que o atributo pode influenciar mais dentro de outras classes

que não a predita. Os pontos são acumulados verticalmente para indicar a densidade de distribuição dos *Shapley values* por atributo, e coloridos de acordo com intervalo de valores do atributo, do menor para o maior.

A Figura 3.4b apresenta os efeitos de interação marginal entre duas variáveis, com cada ponto representando a predição de uma instância de dado. O gráfico considera todas as instâncias e descreve o relacionamento global das variáveis com a predição do modelo. Neste exemplo, foram selecionadas instâncias da classe *virginica*. O eixo das abscissas descreve o comprimento da sépala (*sepal length*) e o eixo das ordenadas representa o *Shapley value* atribuído ao respectivo comprimento de sépala, informando o quanto o valor da sépala altera a predição para cada instância. Valores mais distantes de zero (eixo horizontal) indicam que a variável é importante para ser classificada como *virginica*. Os pontos são coloridos de acordo com a largura da pétala (*petal width*), que é a variável que possui maior efeito de interação com o comprimento de sépala.

O gráfico da Figura 3.4c proporciona a análise local de uma instância específica e, com isso, possibilita compreender quanto cada atributo contribuiu para uma predição única. No *force plot* são encontradas informações do valor predito da instância sob explicação, o valor base (resultado esperado para a predição média do modelo sobre os dados de treinamento) e o valor de cada atributo. A cor das "setas" sob a linha horizontal ilustra a influência das variáveis, com o vermelho indicando aquelas que contribuíram positivamente e o azul quais contribuíram negativamente. Quanto maior a seta, maior o impacto da variável. Embora seja uma metáfora concisa, devido ao seu arranjo horizontal, o *force plot* se mostra pouco eficiente em trabalhar com muitas variáveis ao mesmo tempo. Essa limitação é contornada com o gráfico da Figura 3.4d, que apresenta informações semelhantes, mas organizadas verticalmente, em formato cascata (*waterfall*). Já na Figura 3.4e, são utilizadas linhas para representar os efeitos de cada atributo, possibilitando visualizar o perfil das importâncias de uma ou mais instâncias simultaneamente.

Observe que para gerar a sequência de gráficos da Figura 3.4, foi utilizado um modelo de classificação binário (dados com apenas duas classes). Neste cenário, a saída esperada do modelo de aprendizado seria um valor de probabilidade entre 0 e 1, indicando a qual das duas classes as amostras têm maiores chances de classificação. No entanto, o valor final do SHAP pode ser algo bem diferente. Isso ocorre porque o SHAP, por padrão, explica um modelo classificador em termos de seu resultado marginal anterior à aplicação da função de ativação de saída do modelo. Logo, as unidades do SHAP são, na verdade, taxas de classificação (*log odds*) em vez das probabilidades de classificação. Segundo Aas, Jullum e Løland (2021), esta não é uma boa escolha de *design* no sentido de favorecer a interpretabilidade direta dos resultados do SHAP.

Além do artigo inicial, os autores do SHAP publicaram outros trabalhos com otimizações

e novas funcionalidades. Lundberg, Erion e Lee (2018) desenvolveram o TreeSHAP, uma versão específica para modelos baseados em *tree ensembles*, como o *Random Forest*. O TreeSHAP considera a dependência entre os atributos, ao estimar a existência de algum grau de dependência, mas não de toda a dependência (AAS; JULLUM; LØLAND, 2021). O artigo ainda apresenta contribuições interessantes, como o agrupamento supervisionado, que trata uma das questões mais desafiadoras dentro do contexto de agrupamentos não-supervisionados, ao utilizar o conceito de *feature attribution* para converter os atributos de entrada em valores com as mesmas unidades da saída do modelo e, com isso, determinar ponderações (métrica de distância) que permitam comparações diretas da importância relativa entre atributos com diferentes unidades de medida.

Já em Lundberg *et al.* (2018), os autores aplicaram o SHAP sobre modelos do tipo *ensemble*, dentro de um contexto médico voltado à predição de complicações durante procedimentos cirúrgicos. O trabalho foi testado com médicos especialistas por meio de uma interface gráfica *web* e publicado em uma das mais conceituadas revistas científicas da atualidade. Chen, Lundberg e Lee (2021) estenderam o SHAP sob a ótica do DeepLIFT, criando, com isso, o DeepSHAP dedicado a prover explicações para modelos baseados em *Deep Learning* e árvores do tipo *Gradient Boosted*.

Lundberg e Lee (2017) descreveram as conexões entre o DeepLIFT e os *Shapley values*, de modo que o DeepLIFT pode ser visto como uma aproximação do cálculo dos *Shapley values* (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017). Neste contexto, o DeepSHAP explica uma predição a partir de uma referência, neste caso, um subconjunto de amostras com valores ajustados de modo (não necessariamente) aleatório, chamado de *background distribution*. A instância sob análise é explicada a partir da configuração de variáveis a serem "perdidas", com essas variáveis faltantes sendo referenciadas a valores correspondentes nas amostras da *background distribution*. Então, obtém-se um SHAP *value* da instância em relação a cada uma das amostras da *background distribution*, com o SHAP *value* final calculado por meio da média sobre as atribuições de importância relativas à *background distribution*. O DeepSHAP é compatível com *PyTorch* e *TensorFlow*, mas pode ser menos preciso do que o SHAP. Lundberg *et al.* (2020) desenvolveram uma versão aprimorada do SHAP que estende o conceito de explicações locais para separar e capturar os efeitos de interação individuais das variáveis, não apenas sobre instâncias únicas, mas também sobre pares de instâncias, gerando explicações em uma matriz de importâncias.

Chen, Lundberg e Lee (2022) apresentaram o Generalized DeepSHAP (G-DeepSHAP), um método local desenvolvido a partir de um aprimoramento do DeepSHAP e do DeepLIFT, para permitir explicações sobre séries de modelos distribuídos. O G-DeepSHAP determina uma distribuição base (*baseline distribution*) por meio do algoritmo de *k-means*. Então, a instância sob explicação é comparada a esta distribuição base, diminuindo o viés de se

confiar em uma única *baseline* aleatória, além de permitir que as explicações do método executem tarefas contrastivas. G-DeepSHAP generaliza a *Rescale rule* introduzida pelo DeepLIFT, propagando as importâncias para explicar predições de modelos em série de tipos de mistos, em vez de camadas de um único modelo profundo. A *Rescale rule* em grupo do G-DeepSHAP também permite redução de dimensionalidade para atributos altamente correlacionados, entretanto, o G-DeepSHAP não garante a satisfação dos axiomas atendidos pelos *Shapley values*.

Hong *et al.* (2020) utilizaram *force* e *decision plot* para analisar os resultados de um modelo baseado em redes profundas CNN e LSTM, aplicado em dados de sensores para prognóstico de manutenção de motores de aviação do tipo turbofan. Hamilton *et al.* (2021) estenderam o SHAP (e o LIME) para explicar redes CNN profundas aplicadas em busca e recuperação de imagens por similaridade. Xu *et al.* (2021) utilizaram o SHAP em um sistema interativo para analisar o relacionamento entre variáveis de entrada e a saída em modelos de predição para séries temporais multidimensionais. Casalicchio, Molnar e Bischl (2019) propuseram uma abordagem local baseada em *Shapley values* para medir a importância de atributos individuais, provendo informações por meio de ferramentas gráficas como *Partial Dependence* e *Individual Conditional Expectation plots*.

3.2.8 Questões em XAI relacionadas aos *Shapley Values*

Muito embora a explicabilidade de predições por meio de *Shapley values* seja um tema de intensa pesquisa em XAI atualmente, a abordagem apresenta limitações consideráveis. Um *Shapley value* atribui um valor de importância para uma variável e não um modelo interpretável, como o LIME, por exemplo. Por isso, muitos dos métodos derivados de *Shapley values* não permitem verificar as mudanças nas predições por meio de alterações nos dados de entrada. Apesar do KernelSHAP contornar essa limitação ao habilitar o LIME para estimar *Shapley values* (MOLNAR, 2019).

Amparore, Perotti e Bajardi (2021) apontaram as vantagens de se utilizar o SHAP em diferentes cenários, sugerindo o método como a melhor escolha quando o objetivo analítico for a concordância local, com o SHAP atingindo níveis excepcionais de concordância. Mas esta pesquisa também verificou que o SHAP não é muito mais estável do que o LIME, e que a alegada vantagem do SHAP só pode ser explorada na prática para conjuntos de dados com poucas variáveis. O cálculo exato dos *Shapley values* consome muitos recursos, com Aas, Jullum e Løland (2021) indicando a intratabilidade dos *Shapley values* para conjuntos com mais de dez variáveis. Logo, na maior parte dos casos, apenas soluções aproximadas são factíveis (MOLNAR, 2019). Neste sentido, Hooker *et al.* (2018) observaram que o SHAP exibe comportamento determinístico sobre dados de baixa dimensionalidade, mas quando aplicado a dados de alta dimensionalidade, o método faz uso de técnicas estatísticas

de amostragem baseadas em Monte Carlo, tendendo a explicações instáveis.

A correlação pode impor sérias limitações, uma vez que as importâncias tendem a ser divididas entre as variáveis correlacionadas, mascarando a verdadeira importância de cada atributo. Hooker e Mentch (2019) demonstraram que a explicabilidade a partir da atribuição de importância por meio de métodos baseados em permutação, como o SHAP, pode ser algo altamente enganoso. A questão abordada pelos autores se refere à criação de conjuntos de variáveis com combinações improváveis ou impossíveis dentro do contexto de dados original. Essa estratégia funcionaria bem em cenários com independência entre variáveis, mas em estudos observacionais e problemas de aprendizado de máquina, é muito raro que variáveis sejam estatisticamente independentes (AAS; JULLUM; LØLAND, 2021).

Se houver alto grau de correlação entre algumas ou todas as variáveis e se estes atributos correlacionados forem combinados em configurações não-realistas, o modelo de aprendizado será forçado a extrapolar para regiões desconhecidas do espaço de características. Como os métodos de explicabilidade baseados em permutação são sensíveis ao modo segundo o qual o modelo extrapola, o comportamento de extrapolação se torna uma significativa fonte de erros, com as explicações resultantes sendo geradas com base na captura de informações indesejadas ou distorcidas (HOOKER; MENTCH; ZHOU, 2021). Além disso, ignorar as estruturas de dependência entre as variáveis e assumir distribuições independentes, como é feito no SHAP, é uma propriedade cujas consequências ainda não foram cuidadosamente estudadas (SUNDARARAJAN; NAJMI, 2020).

Uma solução seria a amostragem condicional, em que as variáveis são condicionalmente amostradas de acordo com aquelas que já fazem parte da explicação, embora isso viole o axioma da simetria (JANZING; MINORICS; BLÖBAUM, 2019). Aas, Jullum e Løland (2021) demonstraram que apesar de, em tese, considerar a dependência entre variáveis, o TreeSHAP é potencialmente impreciso quando as variáveis são de fato dependentes. Estes autores estenderam o KernelSHAP utilizando elementos do TreeSHAP para lidar com atributos dependentes, mas a solução sofre com a complexidade computacional.

Kumar *et al.* (2020) e Kaur *et al.* (2020) argumentam que os *Shapley values* não são uma solução natural para explicações do tipo *human-centric* (voltadas ao entendimento humano), devido à falta de clareza por trás das análises geradas pelo método, que podem levar a vieses de confirmação ou mesmo a um certo exagero de confiança. Kumar *et al.* (2020) ainda levantaram questões matemáticas sobre o método, observando que as soluções empregadas para contornar essas questões matemáticas acabam introduzindo mais complexidade. Por exemplo, quando o conjunto de variáveis é arbitrariamente grande, é necessário escolher um conjunto significativo. No entanto, as explicações podem mudar consideravelmente de acordo com as variáveis selecionadas. Além do mais, não é óbvio se duas variáveis estatisticamente relacionadas podem ser consideradas separadamente, como

é feito pelas abordagens aditivas, ainda que essa escolha não tenha impacto no resultado. Os autores ainda apontam que não está claro se uma média das importâncias representando "todas as possíveis explicações" é um modo adequado para gerar informações.

Kumar *et al.* (2021) verificaram as perdas de informação dos *Shapley values* em relação às interações entre atributos correlacionados e desenvolveram os *Shapley residuals*. A proposta dos autores não é um método de explicação ou de avaliação propriamente dito, mas sim um quantificador de informação perdida. Os *Shapley residuals* destacam as limitações dos *Shapley values* ao indicar o quanto das importâncias resultantes podem vir de relacionamentos que foram desconsiderados sendo que, nestes cenários, as explicações devem ser tomadas com algum ceticismo por parte dos analistas.

Quais fatores influenciam os *Shapley values*? Como a distribuição de uma variável influencia o seu *Shapley value*? Como as explicações baseadas em *Shapley values* variam para diferentes saídas preditas pelo mesmo modelo? Kumar e Chandran (2021) destacaram a dificuldade em responder essas questões, pois os *Shapley values* não tem uma solução fechada com tempo computacional factível e a sua estimativa numérica é cara. Os autores verificaram que além de depender dos valores do atributo sob explicação e do modelo preditor, os *Shapley values* também dependem da distribuição dos dados. Quando a variância geral é baixa, a maioria das instâncias cai em uma pequena região do espaço, implicando que a curva de probabilidades pode ser aproximada por uma linha naquela região. Quando a variância é alta, tem-se uma situação de volatilidade em que poucas instâncias se concentram em volta do zero, que é ponto de maior derivada sobre a curva de probabilidades, com os *Shapley values* podendo aumentar em magnitude conforme o valor do atributo se afasta da média.

Mesmo sendo uma das abordagens mais proeminentes da atual literatura XAI (BHATT *et al.*, 2020), é preciso considerar que um *framework* de Teoria dos Jogos não resolve automaticamente o problema de atribuição de importâncias, embora seja uma solução geral adequada para quantificar a importância de atributos (KUMAR *et al.*, 2020).

3.3 Métodos e Métricas de Avaliação

Embora já exista um amplo conjunto de métodos XAI, ainda há uma certa dificuldade em avaliar os resultados obtidos pelas explicações. Um problema em avaliar métodos XAI é que, em geral, não há "*ground truth*", ou seja, não há referências confiáveis do que é uma explicação adequada para cada tipo de problema caixa-preta. Os desenvolvedores de *Machine Learning* acabam recorrendo ao conhecimento de especialistas de domínio como *ground truth* implícito para validar as explicações, por exemplo (BHATT *et al.*, 2020). Além disso, os métodos propostos são frequentemente suportados por axiomas que

determinam propriedades desejáveis para as explicações, ou utilizam dados simulados para checar o que é possível computar em termos de explicações (AAS; JULLUM; LØLAND, 2021). No entanto, mesmo dados sintéticos projetados para fornecer *ground truth* nas explicações podem apresentar limitações significativas, pois não há como garantir que modelos de aprendizado treinados sobre esses conjuntos de fato vão aderir ao *ground truth* (FABER; MOGHADDAM; WATTENHOFER, 2021; AGARWAL *et al.*, 2022b).

Confiar apenas em propriedades axiomáticas para explicar predições de modelos altamente complexos pode não ser suficiente. Neste contexto, as metodologias de avaliação têm seguido quatro linhas: medir a sensibilidade das explicações quanto a perturbações no modelo e nos dados de entrada; inferir o comportamento das explicações a partir da remoção de variáveis; avaliar as explicações a partir de configurações controladas em que já se conhece a importância dos atributos; e avaliação baseada no conhecimento de analistas especializados (*humans in the loop*) (YANG; KIM, 2019).

As avaliações qualitativas e quantitativas em XAI geralmente correspondem à plausibilidade das explicações e à fidelidade ao comportamento do modelo, respectivamente (CHEN; LUNDBERG; LEE, 2022). Bodria *et al.* (2021) abordaram testes quantitativos e Yang e Kim (2019) organizaram um bom referencial de trabalhos que consideram cada uma das quatro metodologias de avaliação citadas anteriormente, além de proporem um *framework* de avaliação baseado em perturbações. Yang e Kim (2019) também debateram as poucas discussões sobre as explicações falsas e destacam que avaliar a explicabilidade é tão importante quanto desenvolver métodos XAI. O trabalho ainda apresenta duas métricas para avaliar o quanto e como as explicações devem mudar de acordo com mudanças nas variáveis de entrada (imagens), de modo controlado. Yang e Kim (2019) não garantem que as técnicas XAI com bom desempenho em seu *framework* terão bom desempenho em dados reais, mas afirmam que suas métricas são testes simples e que as técnicas que falham em testes simples, provavelmente falharão em cenários mais difíceis.

Ablation é uma linha de pesquisas em XAI com o objetivo de validar explicações globais ou locais, removendo informações de atributos de acordo com ranqueamentos de importância (HAMEED *et al.*, 2022). Então, uma aplicação de *ablation* verifica o desempenho relativo de um modelo de aprendizado, perturbando suas variáveis de entrada por ordem de importância. Teoricamente, se um método XAI foi aplicado adequadamente, os atributos mais importantes causarão perda no desempenho do modelo quando estes forem perturbados.

Hooker *et al.* (2018) apresentaram ROAR (*RemOve And Retrain*), um método baseado em *ablation* desenvolvido para comparar explicações no contexto de classificação de imagens. O ROAR é caro computacionalmente, uma vez que o modelo é retreinado a cada entrada perturbada, além do processo de retreinamento divergir conceitualmente do paradigma

XAI *post-hoc*. Outro fato que deve ser levado em consideração é que a avaliação por *ablation* depende da definição de aproximações iniciais (*baselines*). De acordo com Haug *et al.* (2021), quanto mais uma *baseline* se aproxima da distribuição original dos dados, mais discriminativa será a aproximação, enquanto que as *baselines* que desviam da distribuição (OOD, *out-of-distribution*) podem resultar em explicações inválidas.

Estreitamente ligados aos algoritmos de *ablation*, os métodos de seleção de atributos (*feature selection*) são aplicados com o objetivo de aprimorar a avaliação do efeito dos atributos em tarefas de predição e explicabilidade (DAS *et al.*, 2022). As metodologias de *feature selection* determinam subconjuntos de atributos importantes a partir de um conjunto original, individualmente (um por um) ou em grupos, de acordo com a importância relativa dos atributos e seguindo abordagens de eliminação (*top-down*) (YE; SUN, 2018) ou inclusão (*bottom-up*) (DAS *et al.*, 2022). Um algoritmo de *feature selection* reduz a dimensionalidade dos dados e a complexidade computacional, mas essa metodologia também depende de processos de retreinamento.

Weerts, Ipenburg e Pechenizkiy (2019) aplicam análises qualitativas e Adadi e Berrada (2018) apresentam uma extensa discussão sobre a falta de avaliação em XAI. Os autores argumentam que a existência de poucos trabalhos de avaliação em XAI se deve ao aspecto ainda subjetivo da explicabilidade, e também destacam que raras pesquisas se dedicam aos desafios de gerar explicações de fato compreensíveis por humanos. A pesquisa também argumenta que as explicações contrastivas podem ser aplicadas em contextos de avaliação, pois a interação social humana é baseada em contrastividade. Porém, Hooker, Mentch e Zhou (2021) apontam que as explicações contrastivas e contrafactuais podem incorrer em problemas de extrapolação (similar ao SHAP) que as tornam potencialmente enganosas. Além disso, identificar conjuntos otimizados de elementos contrafactuais é uma tarefa NP-difícil (TSIRTSIS; RODRIGUEZ, 2020).

Um dos motivos para a falta de trabalhos concretos com analistas humanos, segundo Yang e Kim (2019), é que o processo de avaliação *human in the loop* é complexo e caro, além de envolver considerações de sociologia e psicologia. Entretanto, levar em consideração o conhecimento de especialistas de domínio pode enriquecer o contexto das explicações, tornando-as mais compreensíveis (JEYASOTHY *et al.*, 2022).

Diferindo das avaliações qualitativas, as métricas quantitativas são relativamente independentes de modelos de aprendizado e quase exclusivamente dedicadas ao contexto do *feature importance* (CHEN; LUNDBERG; LEE, 2022). Amparore, Perotti e Bajardi (2021) apontam não haver um consenso sobre métricas fundamentais, além da ausência de definições no sentido de quantificar a efetividade das explicações. Os autores demonstraram comportamentos inesperados em LIME e SHAP e propuseram quatro métricas para verificar diferentes aspectos em técnicas XAI. Liu *et al.* (2021) também demonstraram algumas

falhas nos métodos de explicabilidade mais utilizados e definiram uma série de métricas, além de uma metodologia de geração de dados sintéticos para aplicar em avaliações. Hooker *et al.* (2018) apresentaram um modo de avaliar variáveis importantes e discutiram a dificuldade em se "comprar" diretamente os resultados de métodos como LIME e SHAP, por exemplo. Alvarez-Melis e Jaakkola (2018b) avaliam os efeitos drásticos que perturbações mínimas podem causar em métodos XAI, chegando a alegar que a maioria das abordagens não é robusta o suficiente mesmo quando são aplicadas em modelos de aprendizado que são.

As métricas de estabilidade medem o quanto uma explicação é sensível a modificações nos hiperparâmetros ou nos dados de entrada do modelo (ALVAREZ-MELIS; JAAKKOLA, 2018a). Contudo, não há consenso sobre a estabilidade (também chamada de sensibilidade ou robustez) na literatura XAI. Mishra *et al.* (2021) discutiram os conceitos de diferentes métricas de estabilidade para métodos de atribuição de importâncias e contrafactuais, utilizando a robustez como um termo unificado.

Agarwal *et al.* (2022b) apresentaram recentemente o OpenXAI, um pacote de código aberto para avaliação e *benchmarking* (análise comparativa) de métodos XAI *post-hoc*. O OpenXAI inclui um gerador de dados sintéticos, uma coleção de conjuntos de dados reais, modelos pré-treinados, métodos de atribuição de importâncias e implementações de diversas métricas quantitativas para avaliar fidelidade, estabilidade e imparcialidade dos métodos XAI incluídos no pacote. Embora não seja muito fácil personalizar seus parâmetros para tal, o OpenXAI também tem como proposta ser empregado para testar novas abordagens de explicabilidade.

3.4 Limitações em XAI

Um modo de promover a confiança é aumentando a transparência dos sistemas, sendo que uma importante parte do aumento na transparência vem da aplicação da explicabilidade (AMANN *et al.*, 2022). A explicabilidade tem potencial para desvendar as caixas-pretas que são os algoritmos de aprendizado complexos e isso ajuda a introduzir mais elementos de confiança que suportem os sistemas que fazem uso ativo desses modelos. Porém, também existe muito debate a respeito das limitações do XAI, além das já elencadas questões relativas à falta de avaliação de explicações. Até aqui, foram discutidos diversos conceitos, necessidades, desafios e os métodos XAI que tratam dessas questões. Entretanto Kaur *et al.* (2020) argumentam que nem todos os cientistas de dados sabem como aplicar a explicabilidade corretamente dentro das etapas do *Machine Learning*.

Vários trabalhos na literatura levantam questões quanto a falta de consenso e definições claras na terminologia básica de XAI (AMANN *et al.*, 2022). Ao longo deste texto,

foi feito um trabalho em identificar os principais conceitos e buscar definições o mais esclarecedoras possível, à luz da literatura, para cada um desses elementos. Acredita-se que essas definições contribuam no sentido de elucidar muitas dúvidas conceituais, mas, ainda assim, é preciso haver concordância na área do XAI. Não é necessário que cada novo artigo defina os mesmos termos à sua maneira.

Krishna *et al.* (2022) discutiram o quanto é comum que explicações produzidas por diferentes métodos XAI (considerados estado da arte) discordem entre si, um fenômeno conhecido por *disagreement*. Não é claro os motivos dessa discordância, uma vez que ainda há poucas pesquisas sobre este tema. Os autores ainda conduziram estudos com cientistas de dados visando esclarecer como as discordâncias poderiam ser resolvidas. As técnicas baseadas em permutação de atributos estão entre as principais abordagens XAI de *feature importance*, apresentando resultados bastante apelativos (CASALICCHIO; MOLNAR; BISCHL, 2019). Computar importâncias por meio da permutação de atributos pode ser eficaz sob o ponto de vista global, entretanto, este processo pode falhar em produzir explicações precisas em casos diferentes do global (BARBER; CANDÈS, 2015). Além disso, é possível demonstrar com exemplos simples que explicações baseadas em permutação podem ser enganosas ou carregar distorções (HOOKER; MENTCH; ZHOU, 2021).

Explicações por meio de relacionamentos causais é algo pouco explorado na literatura XAI. Descobrir causalidade em grandes conjuntos de dados está longe de ser trivial, embora seja considerado um objetivo de explicabilidade significativo (BHATT *et al.*, 2020). Outra limitação, que pode ser vista com um desafio a ser enfrentado pelos métodos XAI, tem a ver com as variáveis não numéricas. De fato, tratar adequadamente variáveis categóricas é um desafio também para os algoritmos de aprendizado. A solução tradicionalmente aplicada neste contexto é o *one-hot encoding*, que é um método simples para conversão de variáveis não numéricas em matrizes binárias. Porém, em grandes conjuntos com muitos atributos categóricos, o *one-hot encoding* acaba aumentando significativamente o grau de esparcidade dos dados. Aas, Jullum e Løland (2021) sugerem abordagens alternativas da literatura de clusterização, que definem funções de distribuição que manipulam dados não numéricos (HUANG, 1998), além de generalizações da distância de *Mahalanobis* para misturas de variáveis nominais, ordinais e contínuas (LEON; CARRIERE, 2005).

Alguns trabalhos questionam a real necessidade de explicações (BUNT; LOUNT; LAUZON, 2012; KULESZA *et al.*, 2013). Conforme discutido, nem sempre é necessário haver explicações para todos os contextos contendo modelos de aprendizado. Entretanto, após toda a exposição empreendida até aqui, espera-se ter esclarecido que não é possível simplesmente confiar em sistemas caixa-preta somente com base em bons valores em métricas de *performance*. É necessário haver auditabilidade para compreender as razões por trás das predições. Também são pertinentes os questionamentos sobre a falta de

significância "*human-centric*" na explicabilidade e os impactos causados nos modelos mentais de compreensão dos usuários, tanto como resultado da falta de completude das explicações quanto da sobrecarga de informações geradas pelos métodos XAI (KUMAR *et al.*, 2020). Argumenta-se que estas limitações, ainda que centrais, amadurecerão com a evolução das pesquisas e com o desenvolvimento aprofundado de novas ferramentas XAI. Logo, é importante que essas críticas sejam sim levantadas e não apenas relativizadas.

3.5 Resumo e Caracterização dos Métodos XAI

A Tabela 3.1 apresenta um resumo da literatura revisada neste trabalho, de acordo com os objetivos dos métodos XAI. Note que alguns métodos podem ser classificados dentro de mais de uma categoria, algo natural, visto que alguns métodos se baseiam em uma combinação de abordagens para cumprir com a tarefa da explicabilidade. Esta tabela foi parcialmente inspirada nas categorizações propostas por Amparore, Perotti e Bajardi (2021), Liao, Gruen e Miller (2020) e Guidotti *et al.* (2018b).

De acordo com Bhatt *et al.* (2020), *feature importance* é a classe de técnicas XAI mais utilizada atualmente. Embora sejam tradicionalmente caracterizados em diferentes classes, muitos métodos que "explicam predições" geram informações por meio de tarefas de atribuição de importâncias. Este é o caso do LIME, por exemplo, um método que atribui importância aos atributos de entrada do modelo e que normalmente é classificado como uma técnica de explicabilidade por simplificação. De modo semelhante, o LRP também atribui importâncias, mas baseado em uma estrutura de propagação de sinais.

Os cientistas de dados utilizam *feature importance* para diversas tarefas de explicabilidade, como compreender se há distorções nos conjuntos de dados ou deficiências nos modelos, compreender parte do processo de aprendizado e fornecer novas perspectivas para a engenharia dos atributos (extração e transformação de variáveis a partir de dados brutos). Observe que um método de *feature importance* gera informações sobre a relevância de um atributo específico dentro do contexto de um determinado modelo de aprendizado treinado. No entanto, é importante destacar que isso não generaliza os efeitos desse atributo ou a sua possível relevância quando os dados são aplicados em outros modelos.

3.6 Considerações Finais

Com esta revisão da literatura, espera-se ter demonstrado que a explicabilidade no contexto de modelos de aprendizado de máquina complexos é um campo potencialmente fértil, com diversas estratégias empregadas atualmente. Os espaços de classificação criados por modelos não-lineares moldam regiões para as diferentes classes de dados, que podem

Tabela 3.1 – Sumário das metodologias e pesquisas em *Explainable Artificial Intelligence* revisadas nesta pesquisa.

Metodologia	**Tipo de explicação**	**Tipo de modelo**	**Localidade**	**Referências**
Aproximação	Explica o modelo ou predições	Agnóstico	Local ou Global	Guidotti *et al.* (2018a), Ribeiro, Singh e Guestrin (2018), Tan *et al.* (2023), Hastie e Tibshirani (2017), Lou, Caruana e Gehrke (2012), Lou *et al.* (2013), Caruana *et al.* (2015), Tan *et al.* (2023)
Visualização	Explica o modelo	Específico	Global	Rauber *et al.* (2016), Hohman *et al.* (2018), Zhang e Zhu (2018), Cantareira, Etemad e Paulovich (2020), Vig (2019a), Vig (2019b), Garde, Kran e Barez (2023)
	Explica predições	Agnóstico	Local	Lundberg, Erion e Lee (2018), Lapuschkin *et al.* (2019), Chan *et al.* (2020a), Chan *et al.* (2020b), Yuan *et al.* (2022), Hamilton *et al.* (2021), Xenopoulos *et al.* (2022)
	Inspeção de modelo	Agnóstico	Global ou Local	Friedman (2001), Goldstein *et al.* (2015), Zednik (2021), Marcílio-Jr, Eler e Breve (2021)
Decision boundaries	Explica o modelo	Específico	Global	Li, Ding e Gao (2018), Karimi, Derr e Tang (2019), Yousefzadeh e O'Leary (2019a), Guan e Loew (2020), Englhardt *et al.* (2020), Sohns, Garth e Leitte (2023)
Contrastivas e contrafactuais	Baseada em exemplos	Agnóstico	Local	Wachter, Mittelstadt e Russell (2017), Poyiadzi *et al.* (2020), Barocas, Selbst e Raghavan (2020), Jacovi *et al.* (2021), Stepin *et al.* (2021), Raimundo, Nonato e Poco (2022)
Gradientes	Explica predições	Específico	Local	Simonyan, Vedaldi e Zisserman (2013), Shrikumar *et al.* (2016), Shrikumar, Greenside e Kundaje (2017), Sundararajan, Taly e Yan (2017), Smilkov *et al.* (2017), Bach *et al.* (2015), Lapuschkin *et al.* (2019), Kohlbrenner *et al.* (2020)
Simplificação	Explica predições	Agnóstico	Local	Thiagarajan *et al.* (2016), Ribeiro, Singh e Guestrin (2016a), Ribeiro, Singh e Guestrin (2016b), Ribeiro, Singh e Guestrin (2016c), Turner (2016), Ribeiro, Singh e Guestrin (2018), Guidotti *et al.* (2018a), Amparore, Perotti e Bajardi (2021), Hamilton *et al.* (2021)
Shapley	Explica predições	Agnóstico	Global ou Local	Lipovetsky e Conklin (2001), Štrumbelj e Kononenko (2014), Hooker e Mentch (2019), Sundararajan e Najmi (2020), Kumar *et al.* (2020), Kaur *et al.* (2020), Hooker, Mentch e Zhou (2021), Kumar *et al.* (2021), Kumar e Chandran (2021), Lundberg e Lee (2017), Weerts, Ipenburg e Pechenizkiy (2019), Amparore, Perotti e Bajardi (2021), Hamilton *et al.* (2021), Hong *et al.* (2020), Aas, Jullum e Løland (2021), Xu *et al.* (2021)
TreeSHAP, DeepSHAP e G-DeepSHAP	Explica predições	Específico	Local	Lundberg, Erion e Lee (2018), Lundberg *et al.* (2018), Lundberg *et al.* (2020), Chen, Lundberg e Lee (2021), Chen, Lundberg e Lee (2022)
Avaliação	Testes e validação	Agnóstico	Global ou Local	Hameed *et al.* (2022), Das *et al.* (2022), Liu *et al.* (2021), Faber, Moghaddam e Wattenhofer (2021), Yang e Kim (2019), Weerts, Ipenburg e Pechenizkiy (2019), Amparore, Perotti e Bajardi (2021), Hooker *et al.* (2018), Alvarez-Melis e Jaakkola (2018b), Alvarez-Melis e Jaakkola (2018a)

Fonte: Elaborada pelo autor.

ser completamente entrelaçadas e difíceis de desvendar (KARIMI; DERR; TANG, 2019). A maior parte das técnicas XAI estado da arte não é estável ou possui parâmetros importantes determinados de modo empírico e pouco fundamentado.

Esta pesquisa difere dos trabalhos revisados ao desenvolver uma metodologia XAI mais estável dentro da tarefa de explicar os resultados de modelos de aprendizado por meio da atribuição de importâncias. Com isso, busca-se apresentar quais fatores influenciaram nas predições, mas sem recorrer a heurísticas ou escolhas infundadas.

Neste capítulo, foram discutidos os principais trabalhos em XAI considerados correlatos ao tema desta pesquisa ou que contribuem com o esclarecimento de alguma técnica utilizada ou ferramenta desenvolvida. Lembrando que a comunidade de pesquisas em XAI é produtiva, com novos trabalhos sendo publicados em diversos periódicos e eventos a cada ano, em busca de explicações para os mais variados problemas caixa-preta. Assim, continuamente surgem métodos inovadores e mais eficientes para preencher as lacunas ainda existentes. Mesmo com os esforços empreendidos aqui, seria inviável proceder uma análise sobre todas as técnicas relevantes em XAI. Porém, semelhante ao que se fez no capítulo anterior, sempre que possível, houve a indicação de *surveys* e *reviews* para que o leitor interessado em alguma técnica específica possa consultar a literatura especializada.

No próximo capítulo, serão apresentadas as características do T-Explainer, o método XAI desenvolvido nesta pesquisa que vem para abordar as principais lacunas encontradas nas atuais abordagens de explicabilidade. O T-Explainer é uma nova ferramenta XAI baseada nos sólidos fundamentos matemáticos da expansão de Taylor para gerar atribuições de importância localmente, que possui propriedades desejáveis, como precisão local e consistência, enquanto é estável.

Capítulo 4

Desenvolvimento da Proposta

4.1 Considerações Iniciais

A metodologia XAI apresentada neste capítulo visa a construção de uma nova ferramenta computacional para explicar predições de modelos de aprendizado complexos, baseada na sólida fundamentação matemática da expansão em série de Taylor. A abordagem desenvolvida se chama T-Explainer, uma técnica aditiva de atribuição de importâncias que atende a propriedades desejáveis dentro do contexto XAI. O T-Explainer gera explicações interpretáveis, independentes de modelos, localmente precisas e consistentes com as predições dos modelos de aprendizado sob análise, além de serem estáveis.

Na Seção 4.2, são descritos os requisitos a serem explorados pela explicabilidade, junto de uma visão geral dos componentes necessários para cumprir com o objetivo desta pesquisa. Na Seção 4.3, será introduzida a formulação teórica do T-Explainer. A Seção 4.4 descreve como o T-Explainer satisfaz as propriedades *local accuracy*, *missingness* e *consistency*. Na Seção 4.5, são apresentados os aspectos computacionais e os detalhes de desenvolvimento relacionados com a implementação T-Explainer. As métricas utilizadas para testar e validar o T-Explainer estão descritas na Seção 4.6. Na Seção 4.7, estão algumas considerações sobre este capítulo.

4.2 Visão Geral

Em resumo, uma inteligência artificial explicável é obtida por meio da geração de informações interpretáveis sobre características do processo preditivo de modelos de aprendizado, com o objetivo de esclarecer as decisões de aplicações baseadas em inteligência computacional (ARRIETA *et al.*, 2020). Modelos caixa-preta que apenas entregam predições sem qualquer tipo de explicação sobre os mecanismos de aprendizado por trás, são difíceis de confiar e não fornecem direções contextuais que suportem os seus resultados

quando confrontados pelos usuários (LUNDBERG *et al.*, 2018). Conhecer somente o que entra e o que sai de um modelo de tomada de decisão, sem entender o que realmente está acontecendo nos bastidores, é algo que já não satisfaz os atuais padrões de acesso à informação, além de ir contra as atuais exigências éticas e legais.

No contexto de verificação da corretude das predições, Kumar *et al.* (2020) elencaram questões importantes que uma investigação apoiada por XAI pode ajudar a responder:

- Se um erro foi originado em algum ponto do fluxo de processamento de dados para uma certa variável;
- Se um modelo está operando sobre correlações espúrias dos dados de treinamento;
- Se o modelo exibe algum viés inapropriado;
- Se existe ganho na precisão do modelo para o caso de uma certa variável ser incluída ou excluída.

A lacuna da transparência já foi devidamente identificada dentro da literatura e vem sendo abordada sob as mais diversas perspectivas, com o objetivo de mitigar o conhecido *tradeoff* entre transparência e poder preditivo dos modelos de aprendizado complexos. Entretanto, os benefícios da explicabilidade não vêm de modo irrestrito e mesmo os métodos XAI de maior sucesso atualmente, como LIME e SHAP, por exemplo, sofrem com limitações significativas. Aplicar esses métodos automaticamente para desmistificar modelos caixa-preta, desconhecendo o seu funcionamento e ignorando as suas deficiências, nada mais é do que, ironicamente, utilizá-los também como caixas-pretas.

Conforme discutido no capítulo anterior, LIME se baseia em soluções heurísticas com justificativas frágeis, enquanto o SHAP, embora embasado em uma sólida fundamentação matemática, apresenta formulação complexa e impraticável em certos contextos. No caso do SHAP, a solução determinística apenas é factível para dados com dimensionalidade baixa. Assim, ambos os métodos que despontam como estado da arte em XAI (AMPARORE; PEROTTI; BAJARDI, 2021), baseiam-se em aproximações probabilísticas suscetíveis a aleatoriedades que induzem problemas de estabilidade e confiabilidade. Já os métodos baseados em gradientes dependem menos de processos aleatórios, mas são específicos para aplicações sobre modelos com parâmetros diferenciáveis, como as Redes Neurais, dificultando a sua utilização sobre modelos populares como *Random Forests* e SVMs. Além disso, a estabilidade dos métodos baseados em gradientes é dependente de fatores como a propagação do gradiente, a complexidade do modelo de aprendizado e a escolha apropriada de instâncias de referência (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017).

Isso sugere que a intuição sobre os benefícios da interpretabilidade e as questões que ela poderia ajudar a responder, como as elencadas acima, podem ter apoio em suposições

incorretas (KUMAR *et al.*, 2020). É neste cenário que este trabalho se insere, ao empreender um estudo aprofundado e crítico sobre os atuais métodos XAI, reunindo pesquisas que revelam as propriedades dessas técnicas e destacam seus pontos fortes e fracos. A partir desse conhecimento, aborda-se o objetivo da pesquisa ao desenvolver um novo método para quantificar e atribuir importâncias, o T-Explainer, uma técnica aditiva fundamentada por elementos teóricos e práticos diferentes do que têm sido utilizado até então.

Para compreender as predições de um modelo, é essencial entender a contribuição individual dos seus atributos de entrada. Os problemas de atribuição se encaixam na categoria de métodos de explicabilidade por *feature importance*, isto é, distribuem o valor da predição de um modelo entre os seus atributos, quantificando a sua relevância individualmente. O valor atribuído a cada variável pode ser interpretado como a importância (ou contribuição) desta variável para a predição e, neste sentido, as atribuições têm valor explicativo, visto que indicam o quão influente um atributo é para a decisão tomada sobre uma dada instância (SUNDARARAJAN; NAJMI, 2020).

Em vez de aproximar o modelo de aprendizado original a um modelo mais transparente e interpretável, mas sem garantias de que este seja de fato um modelo adequado, ou de computar importâncias por meio de uma média de contribuições marginais de todas as combinações de atributos, a formulação desenvolvida aqui é mais direta. Lundberg e Lee (2017) demonstraram que LIME, LRP e DeepLIFT aproximam implicitamente os *Shapley values*, com o SHAP fornecendo um entendimento unificado sobre esses métodos (HAMILTON *et al.*, 2021). Porém, o SHAP é pouco estável em contextos com maior dimensionalidade, pois os desenvolvedores têm que recorrer a soluções aproximadas para contornar a grande necessidade por recursos computacionais, que escala exponencialmente, ao se calcular os *Shapley values* de modo determinístico (originalmente um problema NP-difícil). Ou seja, a alta demanda por recursos limita a atribuição de importâncias a bases de dados com poucos atributos (AMPARORE; PEROTTI; BAJARDI, 2021), e o uso de aproximações por amostragem tende à instabilidade.

Entretanto, a aleatoriedade inerente às estratégias de aproximação não é a única fonte de instabilidade da explicabilidade utilizando *Shapley values*. O comportamento de extrapolação derivado de cenários improváveis ou impossíveis que podem ser formados durante o processo de marginalização (quando o modelo de aprendizado é submetido a combinações de atributos correlacionados fora do escopo de treinamento) também adiciona uma camada de instabilidade à metodologia. Neste sentido, argumenta-se que a instabilidade é um ponto crítico em XAI, pois reflete em uma das características mais danosas para a integridade de um método provedor de explicações: a dificuldade em suscitar confiança em seus resultados.

Devido às suas garantias teóricas, o SHAP é considerado um dos mais vantajosos

e indicados métodos para cumprir com requisitos legais de explicabilidade (MILLER, 2019; AMPARORE; PEROTTI; BAJARDI, 2021). No entanto, uma questão de grande importância e que muitas vezes passa ao largo das discussões envolvendo um dos métodos mais populares em XAI atualmente, está relacionada ao real atendimento do SHAP às propriedades *local accuracy*, *missingness* e *consistency*. Talvez devido ao fato dos *Shapley values* observarem a essas propriedades, sendo o SHAP baseado no cálculo de *Shapley values*, compra-se facilmente a ideia de que o SHAP, automaticamente e por definição, "herdaria" as mesmas características, o que pode não ser verdadeiro. As pesquisas de Sundararajan e Najmi (2020) e Kumar *et al.* (2020) analisaram essa questão, apontando na direção oposta. A leitura cuidadosa revela que, na verdade, os autores do SHAP demonstraram o atendimento das propriedades axiomáticas pelo método, de modo conveniente, sem oferecer quaisquer garantias teóricas dos efeitos ao se assumir a independência entre as variáveis. Também não há garantias teóricas sobre a utilização de expectativas condicionais e de aproximações amostrais, algo que, na verdade, poderia violar os axiomas (SUNDARARAJAN; NAJMI, 2020; KUMAR *et al.*, 2020).

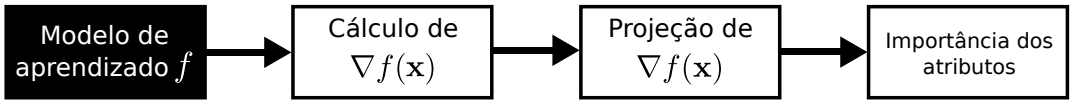
Para enfrentar esses desafios, o T-Explainer difere dos métodos descritos anteriormente em dois aspectos principais: é determinístico, sendo estável por definição; exige a configuração de poucos hiperparâmetros, o que o torna fácil de utilizar. Mais especificamente, o T-Explainer foi construído de modo a não depender de componentes aleatórios, garantindo a sua estabilidade em diferentes execuções, enquanto fundamentado nos sólidos conceitos matemáticos da expansão de Taylor, o que naturalmente lhe confere propriedades desejáveis semelhantes às reivindicadas pelo SHAP, *local accuracy*, *missingness* e *consistency* (LUNDBERG; LEE, 2017), demonstradas nas seções a seguir. Logo, o T-Explainer é um método XAI baseado em gradientes que explica predições de modelos de aprendizado complexos por meio de *feature importance*.

Embora seja possível explicar modelos não diferenciáveis utilizando métodos XAI baseados em gradientes, este procedimento normalmente é feito de modo indireto, aproximando o modelo de aprendizado original por um modelo que seja diferenciável. Mas essa estratégia depende da qualidade da aproximação, algo que nem sempre é fácil de se obter, além de introduzir uma camada a mais de opacidade devido ao fato das explicações serem geradas a partir de um modelo intermediário em vez do original. Então, além da busca por mais estabilidade, o T-Explainer contorna a limitação habitual dos métodos XAI baseados em gradientes ao ser *model-agnostic*, ou seja, o T-Explainer é flexível o suficiente para lidar diretamente com uma gama mais ampla de contextos e modelos de aprendizado.

É claro que computar gradientes não resume toda a metodologia desenvolvida no T-Explainer. A atribuição das *feature importances* é feita localmente por meio da interpretação geométrica entre os espaços de características dos gradientes com os respectivos

espaços de características dos dados de entrada. Na Figura 4.1 estão ilustrados os componentes básicos da metodologia: a partir de um modelo de aprendizado arbitrário e previamente treinado, são localmente calculadas as derivadas parciais de primeira ordem do modelo em relação ao dado de entrada sob explicação; as derivadas parciais formam os componentes do gradiente da função de predição, ou seja, um mapeamento linear que descreve o comportamento do modelo de aprendizado em uma vizinhança suficientemente próxima da instância a ser explicada; por fim, as importâncias são quantificadas e atribuídas por meio da projeção do gradiente sobre os eixos que representam os atributos da instância sob explicação.

Figura 4.1 – Fluxograma com os módulos básicos do T-Explainer.



Fonte: Elaborada pelo autor.

Vales destacar que também foram desenvolvidos métodos para integrar as explicações geradas pelo T-Explainer com as ferramentas gráficas de visualização de informações disponíveis na biblioteca SHAP, tornando o uso do T-Explainer mais fácil e informativo. Embora tenha sido projetado principalmente para explicabilidade local, o T-Explainer também é capaz de gerar visões globais otimizadas.

De modo a oferecer um *framework* XAI mais completo, foram implementadas métricas quantitativas para avaliação comparativa (*benchmarking*) da estabilidade dos métodos XAI, além da introdução de novas métricas para medir a estabilidade e a preservação local de métodos aditivos, e também um método para geração de conjuntos de dados sintéticos, solidificando o T-Explainer como um pacote de ferramentas que promove a compreensão em aplicações de *Machine Learning*. Com isso, espera-se contribuir com a área da explicabilidade, oferecendo uma nova abordagem estável e transparente que atua no sentido de elucidar as predições de modelos de aprendizado não-lineares, enquanto mantém características vantajosas dos métodos XAI estado da arte, mas contornando suas principais deficiências, especificamente no que se refere à complexidade e a desconfiança gerada pela instabilidade.

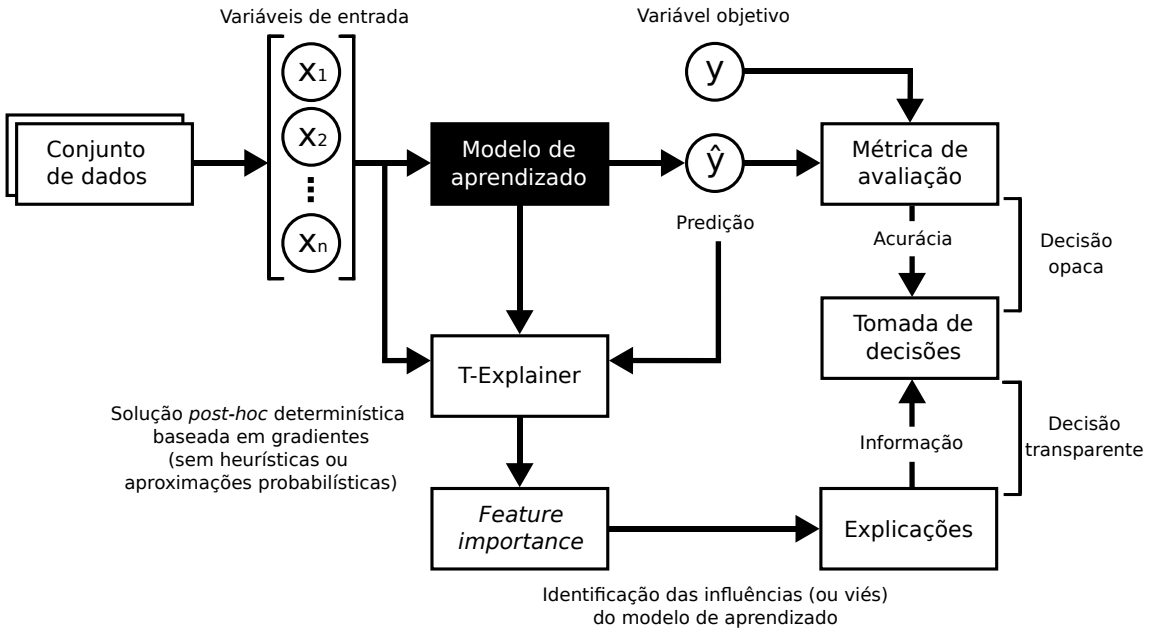
4.3 T-Explainer

Nesta seção, estão descritos os fundamentos matemáticos do T-Explainer. Antes de prosseguir, é preciso estabelecer algumas notações importantes. Seja $\mathbf{X}$ um conjunto de dados multidimensional composto por p instâncias ou observações. Cada observação $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$ é uma instância de dado em $\mathbb{R}^n$, ou seja, caracterizada por n

variáveis ou atributos preditivos. Associada a cada instância, há uma variável objetivo ou classe, $\mathbf{y}$, pertencente ao espaço objetivo, denotado por $\mathbf{Y} \in \mathbb{R}^m$, sendo m o número de classes associadas a $\mathbf{X}$. Esses dados alimentam um modelo classificador do tipo caixa-preta f. Por generalidade, nenhuma definição prévia será feita sobre o modelo f.

Usualmente, um problema de classificação transcorre seguindo os passos ilustrados na parte superior da Figura 4.2. Primeiro, os dados são observados e reunidos em conjunto. Especialistas de domínio (ou métodos automatizados) analisam os n atributos de cada uma das instâncias e as classificam de acordo com as suas características, ou seja, o espaço objetivo $\mathbf{Y}$ deve ser previamente definido para que as instâncias $\mathbf{x}$ sejam classificadas e rotuladas. Um modelo de aprendizado f, também chamado de classificador ou preditor, é uma função que mapeia os atributos preditivos de $\mathbf{x} \in \mathbf{X}$ para o espaço objetivo $\mathbf{Y}$, isto é, $f(\mathbf{x}) = \hat{\mathbf{y}}$, com $\hat{\mathbf{y}}$ sendo a predição ou a classificação feita pelo modelo.

Figura 4.2 – Diagrama esquemático da aplicação do T-Explainer para gerar explicações por meio da descoberta dos atributos mais importantes.



Fonte: Elaborada pelo autor.

O objetivo de um modelo classificador é treinar uma função preditora f sobre uma parte dos dados, o conjunto de treinamento $\mathbf{X}$, para que esta função aprenda e reconheça as características e padrões contidos nos dados de entrada. Durante o processo de aprendizado, f é constantemente otimizada sobre $\mathbf{X}$ de modo que $f(\mathbf{x}) = \hat{\mathbf{y}}$ seja classificada corretamente dentro da respectiva classe $\mathbf{y} \in \mathbf{Y}$. O modelo é então testado em outra parte dos dados, chamada conjunto de teste, com uma medida de acurácia sendo computada para verificar a taxa de predições feitas corretamente, $\hat{\mathbf{y}} = \mathbf{y}$, inferindo se o modelo é satisfatoriamente preciso ou não. Caso não atenda a precisão esperada, é possível proceder com diversas tarefas de engenharia, desde procedimentos de pré-processamento sobre os dados, ajuste

dos hiperparâmetros do modelo, ou até mesmo escolher um modelo de aprendizado diferente. Os procedimentos de treinamento e testes seguem até atingir um nível de precisão preestabelecido, "certificando" que o modelo "aprendeu" sobre os dados. Quanto maior a taxa de predições classificadas corretamente, maior será a acurácia do modelo. Como visto na Seção 2.3, cada modelo de aprendizado possui as suas capacidades e cada contexto de aplicação pode demandar capacidades diferentes, logo, diferentes modelos podem apresentar diferentes acurácias sobre os mesmos dados. As métricas de avaliação guiam os cientistas de dados na escolha do modelo mais preciso para a aplicação em mãos.

Após ser treinado, testado e cumprir com os requisitos de acurácia estabelecidos, o modelo de aprendizado pode ser implantado para classificar (predizer) dados novos e sem classificação prévia (não rotulados). Neste ponto, o modelo estará em terreno desconhecido e aplicará sobre os novos dados de entrada todo o conhecimento adquirido (padrões reconhecidos) durante o procedimento de aprendizado. O monitoramento de modelos é algo desafiador, por causa da grande diversidade em modos de relacionamento entre o espaço de características e o espaço objetivo (LUNDBERG *et al.*, 2020). É essencial verificar se um modelo em operação está classificando os dados novos corretamente ou se, por algum viés de treinamento ou mesmo problema na definição da modelagem, seu desempenho foi considerado satisfatório mas, na prática e sobre dados desconhecidos, o modelo gera classificações incorretas ou por motivos incorretos. A opacidade dos modelos não-lineares aumenta a complexidade da tarefa de verificação desses sistemas e, por isso, problemas em aplicações de Aprendizado de Máquina, como ruídos ou vieses espúrios, por exemplo, podem permanecer ocultos dentro do processo de tomada de decisão.

A partir de uma técnica XAI local e *model-agnostic*, explica-se o valor predito de $f(\mathbf{x})$ provendo informações que descrevem o comportamento de f nas proximidades da instância $\mathbf{x}$, ao passo que uma técnica global explicaria f de um ponto de vista geral, sobre todo o conjunto de dados. O T-Explainer quantifica e atribui a importância para as variáveis de entrada de modelos de aprendizado localmente e, com isso, gera explicações sobre aquilo que está influenciando no processo de predição de cada instância. Então, a tomada de decisões apoiada por Aprendizado de Máquina passa a se basear em informações mais transparentes a respeito do funcionamento do modelo, e não meramente sobre métricas de *performance*.

Conforme Amparore, Perotti e Bajardi (2021), um bom método de explicabilidade é aquele que produz informações confiáveis, com a estabilidade sendo um dos requisitos mínimos neste contexto. Para ser confiável, o método XAI não pode prover explicações (completamente) inconsistentes ou conflitantes, seja quando é executado múltiplas vezes para explicar a mesma instância ou dados similares. A metodologia desenvolvida nesta pesquisa para quantificar importâncias preserva a similaridade das explicações, ao evitar a

dependência de recursos considerados fonte de instabilidade, como amostragens aleatórias.

Voltando à Figura 4.2, mas agora na parte inferior do diagrama, o T-Explainer calcula o *feature importance* localmente, para cada instância, recebendo o modelo preditor treinado f, a instância de dado a ser explicada $\mathbf{x}$ junto da predição feita pelo modelo para esta instância. Por simplicidade, assume-se a partir daqui que f é um modelo de classificação binário treinado sobre $\mathbf{X}$, isto é, $f(\mathbf{x}) \in [0, 1]$ corresponde à probabilidade de $\mathbf{x}$ pertencer à classe 1 e $(1 - f(\mathbf{x}))$ é a probabilidade de $\mathbf{X}$ pertencer à classe 0. Todo o raciocínio desenvolvido a seguir pode ser estendido para modelos de regressão. O classificador f pode ser visto como uma função de variáveis reais (*real-valued function*):

$$f : \mathbf{X} \to \Omega \subset \mathbb{R} \tag{4.1}$$

em que $\Omega = [0, 1]$. Sendo uma função real, f pode ser linearmente aproximada por meio da expansão de Taylor de primeira ordem:

$$f(\mathbf{x} + \mathbf{h}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \mathbf{h} \tag{4.2}$$

onde $\mathbf{h}$ é um vetor deslocamento que corresponde a uma pequena perturbação na vizinhança de $\mathbf{x}$, e $\nabla f(\mathbf{x})$ é o gradiente (transformação linear) de f em $\mathbf{x}$, determinado por:

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right]. \tag{4.3}$$

O gradiente de f em $\mathbf{x}$ corresponde, de fato, à matriz Jacobiana quando f é uma função multivalorada. A matriz Jacobiana, ou simplesmente Jacobiana, é um importante instrumento para o *Machine Learning*, ao agregar as derivadas parciais utilizadas no algoritmo de *Backpropagation* (HAYKIN, 1999). Mas o que exatamente representa a matriz Jacobiana? A Jacobiana de um modelo de aprendizado f é um mapeamento que transforma dados multidimensionais em $\mathbb{R}^n$ para o espaço de classes $\mathbb{R}^m$. Especificamente, a Jacobiana de f é definida como uma matriz $m \times n$ contendo todas as derivadas parciais de primeira ordem de f. Logo, cada um dos elementos da Jacobiana representa a derivada parcial da m-ésima classe de saída do modelo f em relação a n-ésima dimensão (atributo) do vetor de entrada $\mathbf{x}$ (cf. Equação 4.3).

Uma questão ainda mais interessante no contexto desta pesquisa é: o que a Jacobiana faz? O produto escalar entre a Jacobiana e o deslocamento $\mathbf{h}$ é um mapeamento linear de $\mathbb{R}^n$ para $\mathbb{R}^m$, que determina a derivada ou diferencial de f em $\mathbf{x}$. Como uma propriedade fundamental, a Jacobiana descreve a melhor aproximação linear de f em uma vizinhança de $\mathbf{x}$ (NOCEDAL; WRIGHT, 1999; PRESS, 2007). Neste sentido, a matriz Jacobiana descreve como uma perturbação local em um dado de entrada altera o comportamento da saída do modelo. A Jacobiana de f pode ser vista como um descritor da quantidade de

transformação que o modelo sofre localmente em uma vizinhança de $\mathbf{x}$.

De fato, o lado direito da Equação 4.2 aproxima o comportamento de f em uma vizinhança de $\mathbf{x}$ e, sendo esta uma aproximação linear, ela pode ser naturalmente interpretada. Em outras palavras, se é possível tomar a Jacobiana como um descritor, dado que se trata de uma aproximação linear, então é possível utilizar a Jacobiana para gerar explicações. Note que, ao se reduzir o problema de classificação para $m = 1$ (uma classe de saída binária, Equação 4.1), tem-se um caso particular em que a matriz Jacobiana se reduz ao próprio vetor gradiente da Equação 4.3.

A formulação descrita acima se assemelha ao procedimento baseado em expansão de Taylor desenvolvido pelo *Vanilla Gradient* (SIMONYAN; VEDALDI; ZISSERMAN, 2013) para calcular *saliency maps*. A diferença é que, no caso do T-Explainer, as atribuições não dependem de informações de classe e também não utilizam quaisquer parâmetros específicos da arquitetura do modelo. Além disso, o T-Explainer difere dos métodos anteriores, baseados em gradientes, por aplicar uma modelagem de explicação aditiva e um procedimento de otimização determinístico para aproximar os gradientes do modelo, conforme definido a seguir.

Seja $\mathbf{h} = \mathbf{z}' \in \mathbb{R}^n$ uma perturbação em uma pequena vizinhança da instância $\mathbf{x}$, isto é, $\mathbf{x}' = \mathbf{x} + \mathbf{z}'$ é um ponto nesta vizinhança. A perturbação $\mathbf{z}'$ pode ser interpretada como uma entrada de dado simplificada, de acordo com a definição em Ribeiro, Singh e Guestrin (2016c). Então, o T-Explainer é formulado como um método aditivo (cf. Equação 3.7):

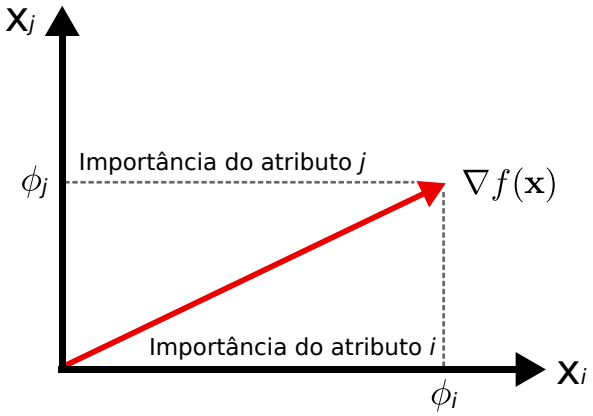
$$g_{\mathbf{x}}(\mathbf{z}') = \phi_0 + \sum_{i=1}^{n} \phi_i z'_i \tag{4.4}$$

em que $g_{\mathbf{x}}$ é a modelagem explicativa, $\phi_0 = \mathbb{E}[f(\mathbf{X})]$ representa o valor esperado para a predição de $\mathbf{X}$, e $\phi_i = \frac{\partial f(\mathbf{x})}{\partial x_i}$. O valor esperado da predição de um modelo de aprendizado sobre um conjunto de dados arbitrário, é um dado estatístico complexo de ser obtido mas que, na prática, é calculado por meio da média das saídas do modelo sobre o conjunto de treinamento $\mathbf{X}$ quando $\mathbf{X}_i$ (valores dos atributos) é desconhecido.

O modelo de explicação $g_{\mathbf{x}}$ é uma modelagem de atribuição local, isto é, existe um $g_{\mathbf{x}}$ para cada $\mathbf{x}$. Isso faz do T-Explainer um método aditivo de atribuição de importâncias (*additive feature attribution*), conforme a definição feita por Lundberg e Lee (2017), o que significa que é possível reconstruir o valor original da predição de $f(\mathbf{x})$, por meio de uma modelagem linear $g_{\mathbf{x}}$ que inclui a soma dos valores de importância atribuídos a cada uma das variáveis de $\mathbf{x}$. Logo, o coeficiente ϕ_i (*feature attribution*) indica a importância do i-ésimo atributo para a predição feita por f sobre $\mathbf{x}$. Portanto, a importância ϕ_i tem uma interpretação geométrica simples e intuitiva no T-Explainer, correspondendo à projeção do gradiente $\nabla f(\mathbf{x})$ sobre o i-ésimo eixo do espaço de atributos. A Figura 4.3 ilustra o

conceito. Quanto mais alinhados o gradiente $\nabla f(\mathbf{x})$ e o i-ésimo eixo do espaço de atributos estiverem, mais importante será o i-ésimo atributo para a predição feita pelo modelo f.

Figura 4.3 – A importância ϕ_i é obtida pela projeção do gradiente de f sobre o i-ésimo eixo do espaço de atributos de $\mathbf{x}$.



Fonte: Elaborada pelo autor.

4.4 T-Explainer – Propriedades

De acordo com Lundberg e Lee (2017), um "bom" método de explicabilidade deve garantir ao menos três importantes propriedades: *local accuracy* (precisão/fidelidade local), *missingness* (irrelevância) e *consistency* (consistência) (cf. Subseção 3.2.7). Nesta seção, será demonstrado que o T-Explainer aproxima a propriedade *local accuracy* enquanto mantém as propriedades *missingness* e *consistency*.

4.4.1 *Local Accuracy*

Para toda instância de dado $\mathbf{x}$ a ser explicada, se um método de atribuição $g_{\mathbf{x}}$ satisfaz a seguinte modelagem:

$$f(\mathbf{x}+\mathbf{z}') = g_{\mathbf{x}}(\mathbf{z}') = \phi_0 + \sum_{i=1}^{n} \phi_i z_i' \tag{4.5}$$

então $g_{\mathbf{x}}$ mantém a propriedade da precisão/fidelidade local (LUNDBERG; LEE, 2017). O T-Explainer não satisfaz estritamente a esta propriedade mas, em vez disso, é possível aproximá-la. Por construção, tem-se o seguinte:

$$f(\mathbf{x}+\mathbf{z}') \approx g_{\mathbf{x}}(\mathbf{z}') = \phi_0 + \sum_{i=1}^{n} \phi_i z_i' \tag{4.6}$$

e, a partir do Teorema do resto na expansão de Taylor (MARSDEN; TROMBA, 2003), existe um limitante superior para o erro de aproximação que é dado por:

$$f(\mathbf{x}+\mathbf{z}') - g_{\mathbf{x}}(\mathbf{z}') = O(\|\mathbf{z}'\|^2). \tag{4.7}$$

Com isso, é possível afirmar que, embora o T-Explainer não satisfaça exatamente a propriedade da precisão local, esta é aproximada com um erro pequeno, da ordem do quadrado da perturbação (que deve ser um valor pequeno, por definição).

4.4.2 *Missingness*

A propriedade da irrelevância, denominada na literatura por *missingness* (LUNDBERG; LEE, 2017) ou *dummy* (KUMAR *et al.*, 2020; SUNDARARAJAN; NAJMI, 2020), trata da tolerância do método de atribuição quanto a valores não impactantes ou nulos. Ou seja, se um atributo x_i não tem impacto sobre o resultado da predição do modelo, então o seu respectivo coeficiente de importância deve ser zero, $\phi_i = 0$ (LUNDBERG; LEE, 2017). No contexto do T-Explainer, um atributo que não impacta em f não causa variação (incremento ou decremento) quando somente este i-ésimo atributo é alterado (caso contrário, o atributo impactaria sim no valor da predição do modelo). Mais especificamente,

$$f(x_1, \ldots, x_i + z'_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n) = 0, \tag{4.8}$$

então, não há variação na i-ésima direção de $\mathbf{x}$ e a derivada parcial $\phi_i = \frac{\partial f(\mathbf{x})}{\partial x_i} = 0$, garantindo que o T-Explainer satisfaz a propriedade da irrelevância. Note que a irrelevância nada mais é do que uma formalização para se dizer que um atributo que não é importante para o modelo de aprendizado, naturalmente requer do método de explicação, a atribuição de valor zero para a importância deste atributo (SUNDARARAJAN; NAJMI, 2020).

4.4.3 *Consistency*

A consistência em métodos de atribuição de importâncias implica que, se um preditor f é não decrescente no atributo x_i, então a atribuição de importância ϕ_i deveria aumentar de magnitude apenas se o valor de x_i aumentar (SUNDARARAJAN; NAJMI, 2020).

Sejam f e $\tilde{f}$ dois modelos classificadores binários. Considere a seguinte notação $\mathbf{x}'\backslash i$ para indicar que o i-ésimo atributo foi desconsiderado em qualquer perturbação de $\mathbf{x}$ (implicando $z'_i = 0$ para qualquer perturbação, então $x'_i\backslash i = x_i$). De acordo com Lundberg e Lee (2017), um método de explicabilidade é consistente se, fixando $\mathbf{x}$, $\tilde{f}(\mathbf{x}') - \tilde{f}(\mathbf{x}'\backslash i) > f(\mathbf{x}') - f(\mathbf{x}'\backslash i)$ isso implica em $\phi_i(\tilde{f}) > \phi_i(f)$.

Suponha que $\tilde{f}(\mathbf{x}') - \tilde{f}(\mathbf{x}'\backslash i) > f(\mathbf{x}') - f(\mathbf{x}'\backslash i)$ está dentro de uma pequena vizinhança de $\mathbf{x}$, em particular,

$$\begin{aligned}\tilde{f}(x_1 + z'_1, \ldots, x_i + z'_i, \ldots, x_n + z'_n) - \tilde{f}(x_1 + z'_1, \ldots, x_i, \ldots, x_n + z'_n) > \\ f(x_1 + z'_1, \ldots, x_i + z'_i, \ldots, x_n + z'_n) - f(x_1 + z'_1, \ldots, x_i, \ldots, x_n + z'_n)\end{aligned} \tag{4.9}$$

para $z'_i \in (-\delta, 0) \cup (0, \delta)$. Então, definem-se

$$\tilde{s}(z'_i) = \frac{\tilde{f}(\mathbf{x}') - \tilde{f}(\mathbf{x}'\backslash i)}{z'_i} \quad \text{e} \quad s(z'_i) = \frac{f(\mathbf{x}') - f(\mathbf{x}'\backslash i)}{z'_i}. \tag{4.10}$$

A partir da Equação 4.9, sabe-se que $\tilde{s}(z'_i) > s(z'_i)$ para $z'_i \in (-\delta, 0) \cup (0, \delta)$. Assumindo que $\tilde{f}$ e f são ambas funções diferenciáveis em $\mathbf{x}$, então

$$\lim_{z'_i \to 0} \tilde{s}(z'_i) = \frac{\partial \tilde{f}(\mathbf{x})}{\partial x_i} \quad \text{e} \quad \lim_{z'_i \to 0} s(z'_i) = \frac{\partial f(\mathbf{x})}{\partial x_i} \tag{4.11}$$

existem. Note que uma função diferenciável em $\mathbf{x}$ implica na (mas não é implicado pela) existência de todas as derivadas parciais de primeira ordem em $\mathbf{x}$ (FITZPATRICK, 2009). Com isso, e a partir do *Limit Inequality Theorem*, tem-se

$$\phi_i(\tilde{f}) = \frac{\partial \tilde{f}(\mathbf{x})}{\partial x_i} > \frac{\partial f(\mathbf{x})}{\partial x_i} = \phi_i(f) \ , \tag{4.12}$$

demonstrando que o T-Explainer mantém a propriedade da Consistência.

O *Limit Inequality Theorem* garante que, dadas duas funções $\tilde{s}, s : (a, c) \cup (c, b) \subset \mathbb{R} \to \mathbb{R}$, se $\tilde{s}(x) > s(x)$ para todo $x \in (a, c) \cup (c, b)$ e, existindo os limites $\lim_{x \to c} \tilde{s}(x) = A$ e $\lim_{x \to c} s(x) = B$, então $A > B$ (WILSON, 2010).

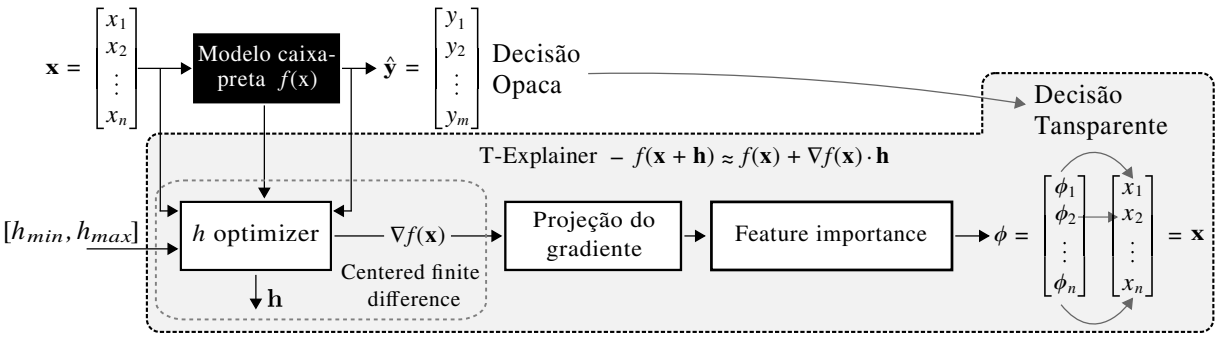
Segundo Lundberg e Lee (2017), as explicações baseadas em *Shapley values* são as únicas modelagens aditivas de atribuição de importância para atributos preditivos de modelos de aprendizado, que são capazes de satisfazer as propriedades *local accuracy*, *missingness* e *consistency* (teoricamente, pode-se dizer que sim, muito embora as soluções práticas desenvolvidas levantem sérios questionamentos quanto a efetividade dessa afirmação, conforme discutido na Subseção 3.2.8). De acordo com o demonstrado ao longo de toda esta seção, o T-Explainer aproxima a *local accuracy*, enquanto satisfaz as propriedades *missingness* e *consistency*. Logo, o T-Explainer se posiciona como um dos poucos métodos XAI para a explicabilidade de predições de modelos caixa-preta a se aproximar do SHAP em termos de garantias teóricas.

4.5 T-Explainer – Aspectos Computacionais

A Figura 4.4 ilustra os detalhes de cada uma das etapas do *pipeline* do T-Explainer para explicar predições por meio da atribuição de importâncias.

Computar o gradiente da Equação 4.3 de uma função real e bem conhecida é (teoricamente) simples, bastando apenas que se calcule as derivadas parciais desta função. No entanto, o T-Explainer demanda derivadas parciais de um modelo caixa-preta arbitrário, previamente treinado sobre dados de natureza diversa e contendo mecanismos internos complexos o suficiente para reconhecer os padrões contidos nesses dados. Lembre-se que

Figura 4.4 – *Pipeline* detalhado do T-Explainer.



Fonte: Adaptada de Ortigossa *et al.* (2024).

é possível descrever como a saída do modelo f é alterada por perturbações no dado de entrada $\mathbf{x}$ com o gradiente $\nabla f(\mathbf{x})$. Então, reorganizando a Equação 4.2, é possível aproximar o gradiente a partir da diferença entre o dado de entrada perturbado e a respectiva saída do modelo f aplicando métodos de diferenças finitas (*finite difference methods*).

Diferenças finitas é uma classe de métodos bem estabelecidos na literatura, que são aplicados para aproximar soluções de equações diferenciais. De acordo com LeVeque (2007), os métodos de diferenças finitas substituem as derivativas nas equações diferenciais por aproximações discretas, com estas transformações resultando em sistemas algébricos de equações computacionalmente factíveis. Ao discretizar as equações diferenciais, a abordagem permite soluções numéricas para diversos tipos de problemas que poderiam ser consideravelmente desafiadores ou mesmo impossíveis de se tratar analiticamente.

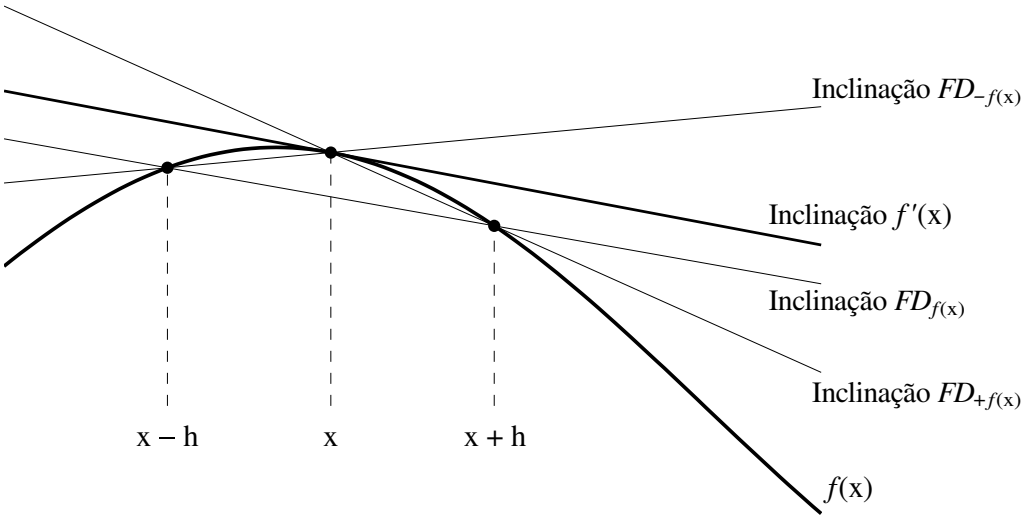
Neste contexto, a solução desenvolvida para aproximar as equações e obter gradientes de modelos de aprendizado no T-Explainer foi a *centered finite differences* (diferenças finitas centradas). Formalmente, as derivadas parciais de $\nabla f(\mathbf{x})$ são aproximadas com base no valor de $f(\mathbf{x})$ de modo que a instância $\mathbf{x}$ é perturbada por um pequeno deslocamento $\mathbf{h}$, atributo a atributo, de ambos os "lados" de $\mathbf{x}$, ou seja, as derivadas parciais são aproximadas em um número finito de pontos próximos a $\mathbf{x}$ por:

$$FD_{f(\mathbf{x})} = \frac{f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}-\mathbf{h})}{2||\mathbf{h}||}. \tag{4.13}$$

A *centered finite differences* foi escolhida pois este método representa uma média de duas aproximações de primeira ordem (*one-sided*), resultando em uma aproximação de segunda ordem de precisão, em que o erro de aproximação é proporcional a $||\mathbf{h}||^2$, menor do que em aproximações de primeira ordem (LEVEQUE, 2007). A Figura 4.5 ilustra o conceito da abordagem. Note que, por definição, é possível calcular derivadas tomando a diferença entre $\mathbf{x}$ e o deslocamento apenas em um dos seus lados, seja $(\mathbf{x}+\mathbf{h})$ ou $(\mathbf{x}-\mathbf{h})$. Entretanto, ao tomar a média da diferença de ambos os lados, a inclinação da reta secante formada pelos dois pontos laterais a $\mathbf{x}$ se aproxima mais rapidamente do perfil da derivada,

mesmo quando o deslocamento não é tão pequeno quanto aquele que seria necessário para alcançar a mesma inclinação da tangente de $f(\mathbf{x})$ somente por um dos lados de $\mathbf{x}$.

Figura 4.5 – Aproximação de $f'(\mathbf{x})$ utilizando a *centered finite difference* é interpretada como uma média das inclinações das linhas secantes laterais a $f(\mathbf{x})$.



Fonte: Adaptada de LeVeque (2007).

Além disso, para aproximar as derivadas parciais necessárias para construir o gradiente do modelo, é preciso determinar um parâmetro fundamental do método de diferenças finitas: a quantidade de deslocamento. O deslocamento $\mathbf{h}$ deve ser um valor pequeno o suficiente para causar a perturbação em uma vizinhança próxima da instância sob explicação $\mathbf{x}$. Seguindo a definição, sabe-se que a derivada de uma função f é computada tomando o limite da Equação 4.13 quando $\mathbf{h} \to 0$, isto é:

$$f'(\mathbf{x}) = \lim_{\mathbf{h}\to 0} \frac{f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}-\mathbf{h})}{2||\mathbf{h}||}. \tag{4.14}$$

Então, por analogia, basta estabelecer um valor pequeno, próximo de zero, para $\mathbf{h}$. No entanto, na prática não é possível seguir esta escolha simplista. Se $\mathbf{h}$ assumir um valor muito pequeno, isso pode gerar erros de arredondamento significativos (*round-off errors*). Por outro lado, se $\mathbf{h}$ for um valor muito grande, isso pode levar a erros de truncamento significativos. Além do mais, o que é considerado muito pequeno ou muito grande vai depender do contexto de f.

4.5.1 Otimizador para Aproximação de Derivadas

Para solucionar a questão em aproximar derivadas parciais, foi desenvolvido um método otimizador de $\mathbf{h}$, baseado em uma busca binária que minimiza o erro quadrático médio (MSE) em uma função de custo. O método otimizador roda sobre um intervalo previamente definido $[h_{min}, h_{max}]$, buscando por um valor otimizado de $\mathbf{h}$ que aproxime o melhor possível

a probabilidade de saída predita pelo modelo $f(\mathbf{x})$ relativo ao valor de $\tilde{f}(\mathbf{x})$ dada pela Equação 4.2. Foi fixado como valor base para h_{min}, a menor distância entre quaisquer duas instâncias do conjunto de dados, ou seja:

$$h_{min} = \min_{i,j=1}^{n} ||\mathbf{x}_i - \mathbf{x}_j||, \ \ \forall i \neq j, \mathbf{x} \in \mathbf{X}. \tag{4.15}$$

Com isso, é possível garantir que o valor otimizado de $\mathbf{h}$ seja determinado respeitando o relacionamento mínimo de similaridade entre as instâncias do conjunto de dados. O método otimizador é formulado do seguinte modo:

$$\nabla f(\mathbf{x}) = \underset{\mathbf{h} \in [h_{min}, h_{max}]}{\arg\min} \mathcal{L}(f, \mathbf{h}, \epsilon_{\mathbf{x}}) + \theta(\nabla f_{\mathbf{x}}) \tag{4.16}$$

em que $\mathcal{L}$ é a função de custo, $\nabla f_{\mathbf{x}}$ é o valor aproximado do gradiente $\nabla f(\mathbf{x})$ obtido a cada ciclo do algoritmo otimizador pela aplicação do método da Equação 4.14 sobre a Equação 4.2, $\epsilon_{\mathbf{x}}$ é o limiar de custo (erro de aproximação) limitado por h_{min}^2 (LEVEQUE, 2007), e θ é um método que garante a estabilidade numérica do processo de otimização, checando se $\mathbf{h}$ preserva a Jacobiana de f como uma matriz não singular (ou *full-rank*, para o caso de matriz não quadrada, obtida quando $n \neq m$). Essa abordagem é capaz de encontrar o gradiente $\nabla f(\mathbf{x})$ de um modelo de classificação binário ou mesmo a matriz Jacobiana completa, derivada de todas as probabilidades de saída do modelo.

É importante destacar que durante o procedimento de otimização de $\mathbf{h}$, é necessário garantir que o valor de $\mathbf{h}$ em cada iteração mantenha a estabilidade numérica do processo. Se $\mathbf{h}$ levar a uma matriz Jacobiana singular (ou *rank-deficient*), o gradiente resultante $\nabla f(\mathbf{x})$ levará o T-Explainer a atribuir valores nulos de importância para atributos potencialmente importantes. Então, θ é um método simples de checagem que verifica se $\mathbf{h}$ implica em um gradiente de f que venha de uma matriz não singular (ou *full-rank*), garantindo, com isso, a estabilidade numérica do procedimento de otimização.

Quando se trabalha com modelos de aprendizado de máquina, é importante garantir que os valores dos atributos estejam dentro do mesmo intervalo para evitar algum viés de escala. A normalização é uma técnica aplicada durante a fase de pré-processamento de dados que tem o objetivo de mapear os valores dos atributos para dentro de um intervalo de escala similar. O processo de normalização garante que os atributos estejam todos na mesma escala, o que, em teoria, contribui com o desempenho e com a estabilidade de treinamento dos modelos (SHANKER; HU; HUNG, 1996). Nem todos os conjuntos de dados necessitam de normalização para serem utilizados no Aprendizado de Máquina, com a normalização sendo requerida quando as escalas dos atributos são discrepantes. O mais comum em Aprendizado de Máquina é o mapeamento dos atributos para o intervalo $[0, 1]$, mas a normalização também pode ser feita em qualquer intervalo, de acordo com os

requisitos do modelo.

Neste sentido, é preciso também observar as propriedades do intervalo de normalização durante a etapa de otimização do parâmetro de perturbação $\mathbf{h}$. A perturbação deve modificar o valor dos atributos mas, ao mesmo tempo, não deve levar estes atributos para fora do intervalo de normalização que foi aplicado sobre o conjunto de treinamento do modelo de aprendizado. Caso o intervalo de normalização seja violado pela perturbação, o modelo será submetido a uma instância completamente desconhecida em relação ao seu espaço de características, algo que acaba forçando o modelo de aprendizado a extrapolar. Como consequência, a extrapolação do modelo pode induzir o método de explicabilidade a erros ou significativas distorções (HOOKER; MENTCH; ZHOU, 2021).

Seja $[a, b]$ o intervalo de normalização dos dados de treinamento do modelo f. Como a *centered finite differences* demanda que a instância sob explicação seja perturbada por ambos os "lados", isto é, $\mathbf{x} + \mathbf{h}$ e $\mathbf{x} - \mathbf{h}$ (Equação 4.13), é necessário garantir que cada atributo da instância perturbada $\mathbf{x}'$ não seja menor que a (no caso da perturbação negativa) e nem maior do que b (lado da perturbação positiva). Antes de enviar $\mathbf{x}'$ para o procedimento de *centered finite differences*, verificam-se os valores mínimo e máximo de $\mathbf{x}'$, $\mathbf{x}'_{min}$ e $\mathbf{x}'_{max}$, respectivamente. Se $x'_i < a, \forall x'_i \in \mathbf{x}'$, então $\mathbf{x}'$ é reescalado dentro do intervalo $[a, \mathbf{x}'_{max}]$. Se $x'_i > b, \forall x'_i \in \mathbf{x}'$, então $\mathbf{x}'$ é reescalado no intervalo $[\mathbf{x}_{min}, b]$. Após esta verificação de escala, $\mathbf{x}'$ é aplicado na *centered finite differences*. Com isso, o otimizador do parâmetro $\mathbf{h}$ desenvolvido, garante que os resultados do T-Explainer sejam baseados em perturbações dentro de uma vizinhança plausível, respeitando o intervalo de dados conhecido pelo modelo de aprendizado.

Por fim, a última etapa do T-Explainer trata da atribuição de importâncias para cada um dos atributos preditivos, seguindo a estratégia geométrica descrita anteriormente, na Seção 4.3. A Equação 4.2 representa a melhor aproximação linear de uma função f em uma vizinhança de pontos perturbados suficientemente próximos a $\mathbf{x}$, descrevendo como essas perturbações no dado de entrada influenciam localmente na predição do modelo. Ao determinar um deslocamento otimizado e aproximar o gradiente do modelo aplicando a expansão de Taylor para cada instância sob explicação, o T-Explainer atribui valores de importância aos atributos que são ajustados de acordo com as características locais de cada instância. Conforme destacado por Ribeiro, Singh e Guestrin (2016c), as explicações devem manter correspondência com o comportamento do modelo em uma vizinhança próxima da instância predita para garantir o significado explicativo.

Porém, garantir a fidelidade local não implica em garantir a fidelidade global simultaneamente, posicionando as explicações local e globalmente consistentes e interpretáveis como um desafio em XAI. A simples agregação de explicações locais pode não ser efetiva a nível global, pois as explicações locais são específicas para cada instância, o que é frequentemente

inconsistente com as explicações globais (WOJTAS; CHEN, 2020). Com a estratégia de otimização adaptativa que ajusta os parâmetros a cada explicação, argumenta-se que o T-Explainer oferece flexibilidade no sentido de gerar visões agregadas para grupos de instâncias ou mesmo para o conjunto de dados como um todo, enquanto preserva os relacionamentos locais. Essa abordagem aprimora as capacidades do T-Explainer em prover explicações consistentes e interpretáveis local e globalmente.

4.5.2 Tratamento de Atributos Categóricos

Os aspectos computacionais descritos até aqui para determinar o gradiente de um modelo de aprendizado, são funcionais quando se trabalha com atributos de natureza numérica. Entretanto, boa parte dos conjuntos de dados empregados no desenvolvimento de soluções baseadas em Aprendizado de Máquina, não estão limitados apenas a dados numéricos, mas também a atributos categóricos. Dados categóricos (ou qualitativos) reúnem atributos que apresentam um grupo com número fixo de possíveis valores, de modo a designar a uma instância em particular, propriedades qualitativas ou categorias nominais. A Figura 4.6a representa um conjunto de dados misto, contendo atributos numéricos e categóricos. Note que o atributo "*Sex*" é categórico, pois carrega os valores qualitativos "*male*" e "*female*" como os possíveis identificadores nominais sobre qual grupo pertence cada indivíduo (instância). Os valores nominais de cada atributo categórico são usualmente definidos durante a etapa de modelagem dos dados.

Embora seja um problema comum em Aprendizado de Máquina, tratar dados categóricos é um desafio. Alguns algoritmos de aprendizado têm a capacidade de trabalhar diretamente com atributos categóricos, como é o caso, por exemplo, das árvores de decisão. Mas boa parte dos modelos de aprendizado mais utilizados não operam diretamente sobre atributos nominais. Os algoritmos que não têm essa habilidade direta, requerem que todos os atributos de entrada sejam numéricos, algo que pode ser visto mais como uma restrição de eficiência desses algoritmos do que como uma limitação em si, uma vez que os atributos categóricos podem ser convertidos em valores numéricos sem grandes dificuldades.

Uma das estratégias mais utilizadas neste contexto é o *one-hot-encoding* (RODRíGUEZ *et al.*, 2018). Derivado da lógica digital, o *one-hot-encoding* é uma técnica de representação de variáveis que transforma categorias nominais em matrizes de valores binários, 0 e 1, indicando a presença ou a ausência de um valor categórico. A Figura 4.6b ilustra o mesmo conjunto apresentado na Figura 4.6a, mas com os valores categóricos transformados em numéricos por meio do *one-hot-encoding*. A coluna *Sex* foi desmembrada em duas novas colunas, "*Sex_male*" e "*Sex_female*", para representar os valores nominais *male* e *female*, respectivamente. Então, quando uma instância possuir o valor *male* dentro do atributo *Sex*, haverá o valor 1 na coluna *Sex_male* e 0 na coluna *Sex_female*, e vice-versa.

Figura 4.6 – Tratamento de dados categóricos utilizando *one-hot-encoding*.

(a) Conjunto de dados misto, com atributos numéricos e categóricos.

Id	Sex	Age	Fare	Embarked
1	male	22	7.2500	S
2	female	38	71.2833	C
3	female	26	7.9250	S
4	female	35	53.1000	S
5	male	35	8.0500	S

(b) Atributos categóricos convertidos em numéricos com *one-hot-encoding*.

Id	Sex_female	Sex_male	Age	Fare	Embarked_C	Embarked_Q	Embarked_S
1	0	1	22	7.2500	0	0	1
2	1	0	38	71.2833	1	0	0
3	1	0	26	7.9250	0	0	1
4	1	0	35	53.1000	0	0	1
5	0	1	35	8.0500	0	0	1

Fonte: Elaborada pelo autor.

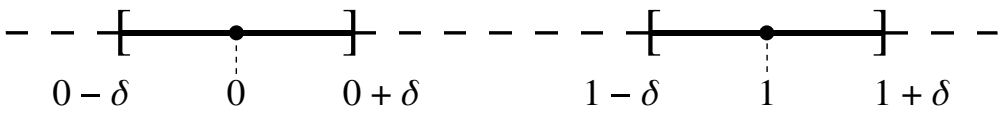
O *one-hot-encoding* é relativamente simples, eficiente e largamente utilizado para a conversão de atributos nominais em dados numéricos. Porém, a técnica tem as suas desvantagens, como o aumento da dimensionalidade e da esparsidade do conjunto de dados, visto que a maior parte das colunas criadas pelo método conterá zeros. Mas uma análise mais aprofundada sobre os benefícios e as desvantagens do *one-hot-encoding*, e outros métodos de conversão de atributos categóricos em valores numéricos, está além do escopo deste trabalho. Aqui, assume-se que a técnica é aplicada para contornar a restrição dos algoritmos de aprendizado quanto às suas habilidades em trabalhar diretamente com variáveis categóricas.

Embora seja uma solução para o Aprendizado de Máquina, o *one-hot-encoding* imprime limitações aos métodos de explicabilidade, especialmente aos baseados em gradientes, como o T-Explainer. Ao substituir os valores nominais por colunas contendo zeros e uns, cria-se uma configuração em que não é possível aproximar as derivadas parciais necessárias para compor os gradientes do T-Explainer. Isso acontece por causa da falta de continuidade entre os pontos 0 e 1 que, embora sejam de fato valores numéricos, são discretos e não contínuos em $\mathbb{R}$. Para que uma função seja derivável em um ponto, este ponto deve estar em um intervalo aberto não-vazio, ou seja, a derivada é calculada sobre funções reais e contínuas, ainda que parcialmente. Aproximar as derivadas parciais de um modelo de aprendizado treinado, na vizinhança de colunas contendo apenas zeros e uns, resultaria em zero, o que induziria o método de explicabilidade a atribuir importância nula a um atributo potencialmente importante.

O T-Explainer foi desenvolvido de maneira modular, dividido em um conjunto de métodos que tratam cada uma das diferentes etapas da geração do *feature importance*. Logo, a adição ou aprimoramento de novas funcionalidades ou capacidades, é facilitada dentro da estrutura do T-Explainer. Para contornar o desafio imposto pelos atributos categóricos, foi desenvolvida uma metodologia de tratamento para este tipo de variáveis que pré-processa as colunas resultantes do procedimento de *one-hot-encoding* imediatamente antes da execução do T-Explainer numérico. O método recebe a informação da presença de atributos categóricos na instância a ser explicada (o usuário discrimina quais são os identificadores das colunas categóricas) e, em vez de executar diretamente o T-Explainer, é feita uma chamada a um método preliminar que ajusta as colunas transformadas pelo *one-hot-encoding* para que estas possam ser manuseadas corretamente pelo T-Explainer.

A metodologia em questão simula a continuidade dos valores 0 e 1 resultantes do *one-hot-encoding*, por meio da criação de uma distribuição ao redor destes valores binários. A Figura 4.7 ilustra o conceito. Os valores 0 e 1 de cada coluna categórica são perturbados por uma distribuição uniforme com raio $\delta \in (0, 0.5)$, que pode ser ajustado pelo usuário.

Figura 4.7 – Indução de continuidade em uma variável categórica transformada por *one-hot-encoding* por meio de perturbação uniforme.



Fonte: Elaborada pelo autor.

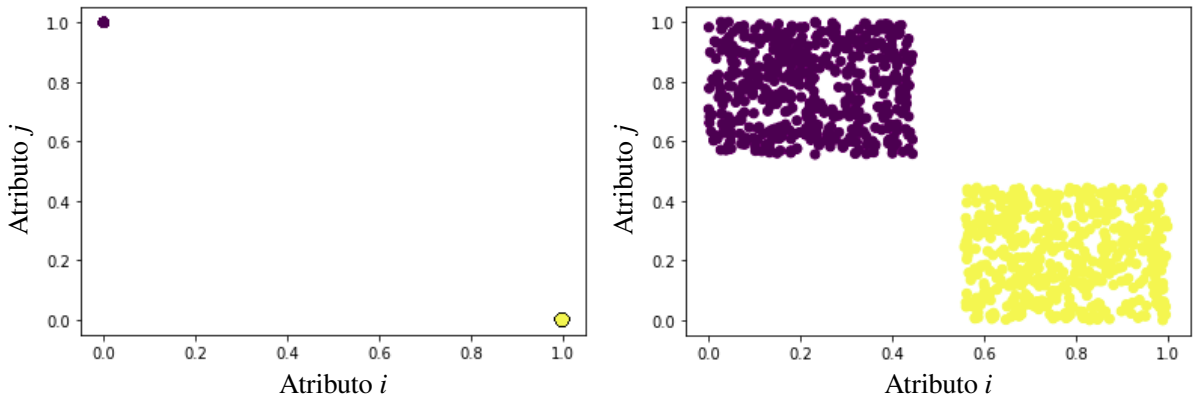
A Figura 4.8b ilustra o resultado da abordagem de indução de continuidade numérica, aplicada sobre um atributo categórico que foi transformado em um par de colunas contendo zeros e uns (Figura 4.8a), como no exemplo descrito anteriormente e ilustrado pela Figura 4.6. Este procedimento cria uma "nuvem" de pontos ao redor dos valores binários, 0 e 1, com abertura $\pm\delta$, simulando a continuidade numérica em uma vizinhança próxima de 0 e de 1, permitindo, com isso, que sejam feitas aproximações de derivadas nesses pontos.

Deve ser destacado que este processo de indução de continuidade dos atributos categóricos é feito sobre o conjunto de treinamento, com as colunas de atributos perturbados sendo posteriormente normalizadas dentro do mesmo intervalo escalar das colunas numéricas. Treina-se uma cópia do modelo de aprendizado original sobre este conjunto de treinamento perturbado. Então, o T-Explainer categórico recebe esta cópia do modelo e a instância a ser explicada (que também foi tratada com perturbações sobre as colunas categóricas) para gerar o *feature importance* relativo à predição desta instância.

Em outras palavras, é possível explicar predições de instâncias contendo atributos categóricos utilizando o T-Explainer, mas sem que fosse necessário realizar alterações

Figura 4.8 – Indução de continuidade sobre atributos com valores discretos binários.

(a) Atributos com valores 0 e 1. **(b)** Atributos perturbados em um intervalo $\pm\delta$.



Fonte: Elaborada pelo autor.

significativas sobre o núcleo do método em sua versão puramente numérica, pois foi desenvolvida e acoplada ao T-Explainer, um módulo preliminar de tratamento para os dados categóricos.

4.6 Métricas para Avaliação Quantitativa

Nesta seção, serão descritas as métricas integradas ao *framework* do T-Explainer para avaliar quantitativamente a qualidade das explicações geradas pelos métodos de atribuição, especialmente quanto aos aspectos da estabilidade. Estabilidade é um requisito fundamental na explicabilidade, uma vez que, para ser confiável, um explicador deve ser no mínimo estável (AMPARORE; PEROTTI; BAJARDI, 2021). Especificamente, a estabilidade é tomada como a geração de explicações consistentes quando o método é executado múltiplas vezes para explicar a mesma instância ou o mesmo conjunto de dados.

4.6.1 *Relative Input/Output Stability*

Relative Input Stability (RIS) e *Relative Output Stability* (ROS) (AGARWAL *et al.*, 2022a; AGARWAL *et al.*, 2022b) são métricas utilizadas para avaliar a estabilidade das explicações quanto a perturbações nos dados de entrada e nas probabilidades de classificação preditas pelo modelo, respectivamente. As implementações desenvolvidas aqui seguiram as definições feitas por Agarwal *et al.* (2022b), entretanto, foram introduzidos alguns melhoramentos para contornar alguns pontos fracos nas propostas originais apresentadas pelos autores. Seja $\mathcal{N}_{\mathbf{x}}$ uma vizinhança de instâncias perturbadas $\mathbf{x}'$ ao redor de $\mathbf{x}$, $e_{\mathbf{x}}$ e $e_{\mathbf{x}'}$ correspondendo às explicações de $\mathbf{x}$ e $\mathbf{x}'$. Considere ainda $f(\mathbf{x})$ e $f(\mathbf{x}')$ como as probabilidades de saída preditas pelo modelo f para $\mathbf{x}$ e $\mathbf{x}'$. Então, as métricas RIS e ROS são formalmente definidas como:

$$\text{RIS}(\mathbf{x}, \mathbf{x}', e_{\mathbf{x}}, e_{\mathbf{x}'}) = \max_{\mathbf{x}'} \frac{\| \frac{e_{\mathbf{x}} - e_{\mathbf{x}'}}{e_{\mathbf{x}}} \|_p}{\max\left(\| \frac{\mathbf{x} - \mathbf{x}'}{\mathbf{x}} \|_p, \epsilon_c\right)} \quad \text{e} \tag{4.17}$$

$$\text{ROS}(\mathbf{x}, \mathbf{x}', e_{\mathbf{x}}, e_{\mathbf{x}'}) = \max_{\mathbf{x}'} \frac{\| \frac{e_{\mathbf{x}} - e_{\mathbf{x}'}}{e_{\mathbf{x}}} \|_p}{\max\left(\| \frac{f(\mathbf{x}) - f(\mathbf{x}')}{f(\mathbf{x})} \|_p, \epsilon_c\right)} \,, \tag{4.18}$$

$\forall \mathbf{x}' \in \mathcal{N}_{\mathbf{x}}$, com p determinando a norma l_p utilizada para medir a mudança na explicação, e ϵ_c sendo um limiar de corte (*clipping threshold*) estabelecido para evitar divisão por zero. Quanto maiores os valores de RIS e ROS, mais instável é o método de explicação com relação a perturbações na entrada ou na saída do modelo.

Note que o numerador de ambas as métricas mede a quantidade de mudança normalizada entre a explicação $e_{\mathbf{x}'}$ gerada sobre uma instância perturbada $\mathbf{x}'$, relativa à explicação $e_{\mathbf{x}}$ gerada sobre a instância de dado original $\mathbf{x}$. Já o denominador mede a norma entre as instâncias $\mathbf{x}$ e $\mathbf{x}'$, para o caso da métrica RIS, e a norma entre as probabilidades de saída preditas pelo modelo para a instância original e a perturbação, $f(\mathbf{x})$ e $f(\mathbf{x}')$, na métrica ROS. Inicialmente, ambas as métricas retornam o valor máximo de mudança aferido ao se perturbar um conjunto de instâncias originais.

Originalmente, RIS e ROS utilizam o método *clip* da biblioteca *numpy* para tratar os valores próximos de zero nos processos de normalização, e também na divisão final. Entretanto, o método *clip*, quando aplicado de modo simplista para cortar valores que estejam fora de um intervalo definido, acaba ignorando valores negativos que podem ser expressivos no contexto do *feature importance*. Assim, optou-se por desenvolver aqui um método *clip* especificamente desenhado para preservar valores significantes, sejam eles positivos ou negativos, descartando os valores de fato próximos de zero (fora do limiar ϵ_c).

Além disso, durante os procedimentos de testes utilizando estas métricas, notou-se que a configuração proposta por Agarwal *et al.* (2022b) poderia ainda ignorar perturbações de instâncias próximas dos *decision boundaries* dos modelos, algo que limitaria a real efetividade das métricas, pois os *decision boundaries* podem naturalmente impor instabilidades às explicações. Esta limitação foi solucionada primeiro perturbando a instância sob avaliação $\mathbf{x}$, ordenando o conjunto perturbado obtido baseado na distância de cada perturbação em relação a $\mathbf{x}$, e então gerando um conjunto final, para cada instância sob avaliação $\mathbf{x}$, contendo a vizinhança com uma quantidade mínima de perturbações. Isso difere da proposta original, que não ordena as perturbações e apenas seleciona aquelas que foram preditas com a mesma classe predita para $\mathbf{x}$. Argumenta-se que a nova abordagem desenvolvida aqui torna as métricas RIS e ROS mais robustas, garantindo a avaliação de todos as instâncias de entrada, ao evitar possíveis omissões na medição da estabilidade ou potenciais quebras do algoritmo devido a situações críticas como as bordas de decisão.

As métricas RIS e ROS foram incorporadas no mesmo ciclo de execução, para que ambas utilizassem exatamente as mesmas configurações de perturbação. Além de permitir comparações diretas entre os resultados, dado que estes são gerados sob as mesmas condições, esta solução economiza tempo de execução dentro do procedimento de *benchmarking* dos métodos de explicabilidade. O ponto mais custoso computacionalmente para a execução de ambas as métricas é a geração das vizinhanças perturbadas, enquanto que os cálculos das estabilidades de entrada e saída podem ser realizados de modo independente a partir da perturbação de $\mathbf{x}$. Então, com ambas as métricas dentro do mesmo ciclo de execução, reduz-se sensivelmente o tempo computacional para obter seus resultados.

Adicionalmente, RIS e ROS foram estendidas para fornecer mais estatísticas além do valor máximo de instabilidade para os métodos testados. Estas métricas agora retornam a instabilidade média e também o desvio padrão, tanto da média quanto da instabilidade máxima. Com esse conjunto de melhoramentos, RIS e ROS se tornam capazes de avaliar a estabilidade e ainda indicar se o modelo de aprendizado (e consequentemente o método XAI que o explica) está sob a possível influência de *outliers* (valores atípicos) nos dados, com mais robustez e confiabilidade.

4.6.2 *Run Explanation Stability*

O uso de componentes aleatórios e algoritmos de amostragem estocásticos por parte de muitos métodos XAI, embora tenha benefícios na aplicabilidade desses métodos, tem como principal consequência o aumento da instabilidade. Isso faz com que as explicações variem mesmo quando o método é aplicado várias vezes para explicar a mesma instância (AMPARORE; PEROTTI; BAJARDI, 2021). Como resultado prático, essa instabilidade levanta incertezas sobre as explicações.

Neste cenário, foi desenvolvida a *Run Explanation Stability* (RES), uma nova métrica adicionada ao *framework* do T-Explainer para medir a estabilidade local dos métodos de atribuição de importância, durante múltiplas explicações geradas para a mesma instância, sob as mesmas configurações de predição. Mais especificamente, a métrica RES avalia a similaridade das explicações quando o método de atribuição é submetido a reiteradas aplicações do procedimento de explicação, individualmente, sobre instâncias $\mathbf{x}$ não perturbadas. Considere $g_{\mathbf{x}i}, \ldots, g_{\mathbf{x}n}$ um conjunto de n explicações para a mesma instância $\mathbf{x}$, e $\bar{g}_{\mathbf{x}}$ sendo a média entre essas n explicações. Então, a métrica RES é formalizada por:

$$\mathrm{RES}(g_{\mathbf{x}}, \bar{g}_{\mathbf{x}}) = \max_{k=1}^{n} \|\bar{g}_{\mathbf{x}} - g_{\mathbf{x}k}\|, \quad \forall \mathbf{x} \in \mathbf{X}. \tag{4.19}$$

Ao final, a métrica retorna o valor máximo de dissimilaridade entre a média e as n explicações de $\mathbf{x}$, considerando todas as instâncias de entrada. Quanto maior o valor

retornado, menos consistente e menos estável são as explicações geradas pelo método avaliado. O conceito geral da RES foi parcialmente inspirado pela métrica *Reiteration Similarity* proposta por Amparore, Perotti e Bajardi (2021). No entanto, a versão desenvolvida aqui simplifica a formulação proposta por esses autores ao utilizar o desvio padrão como medida de similaridade entre as explicações.

4.6.3 *Faithfulness of an Additive Explanator*

Outra novidade incorporada ao conjunto de métricas disponíveis no pacote do T-Explainer é a *Faithfulness of an Additive Explanator* (FAE). Esta métrica avalia a taxa em que um método XAI do tipo *additive feature importance* preserva a fidelidade local (*local accuracy*) nas suas explicações, isto é, a predição do modelo de aprendizado deve ser reconstruída por meio do somatório dos valores de importância atribuídos (LUNDBERG; LEE, 2017). Vale destacar que a fidelidade local é uma propriedade fundamental da classe dos métodos aditivos, que os diferencia das demais abordagens XAI ao propor uma solução que distribui o valor da predição, de modo justo, entre os atributos de entrada. Logo, espera-se que um método baseado em *additive feature importance* respeite o máximo possível esta propriedade.

Seja $f(\mathbf{x})$ a probabilidade de saída predita pelo modelo f sobre uma instância $\mathbf{x}$, e $g_{\mathbf{x}}$ a explicação desta predição, gerada de acordo com a modelagem da Equação 4.5, a métrica FAE é definida como a taxa em que

$$\text{FAE}(\mathbf{X}, g) = \sum_{\forall \mathbf{x} \in \mathbf{X}} F_{\mathbf{x}}, \text{ com } F_{\mathbf{x}} = \begin{cases} 1, & \text{se } \|f(\mathbf{x}) - g_{\mathbf{x}}\| < \epsilon \\ 0, & \text{caso contrário} \end{cases}, \tag{4.20}$$

sendo ϵ um limiar de tolerância definido previamente para ajustar o grau de precisão esperado para a aproximação entre $f(\mathbf{x})$ e $g_{\mathbf{x}}$.

A taxa FAE indica a quantidade percentual de explicações que preservaram a fidelidade local, dentre todas as explicações geradas para todas as instâncias de entrada. Logo, quanto maior o valor da taxa, maior a quantidade de reconstruções de $f(\mathbf{x})$ feitas corretamente e, com isso, mais "fiel" é o método à modelagem da Equação 4.5. Além da taxa, a métrica FAE também computa o valor máximo e a média de fidelidade dos valores $\|f(\mathbf{x}) - g_{\mathbf{x}}\|$. Neste caso, quanto menor o valor da média, mais próximas as aproximações entre $f(\mathbf{x})$ e $g_{\mathbf{x}}$ estarão do limiar ϵ e, por isso, mais fiel será o método XAI aditivo.

4.7 Considerações Finais

Neste capítulo foram descritas as fundamentações da metodologia XAI desenvolvida nesta pesquisa, o T-Explainer. As garantias teóricas demonstradas aqui, posicionam o

T-Explainer em um lugar de destaque entre as principais abordagens para explicabilidade. Com o T-Explainer é possível computar e atribuir a relevância dos atributos dos dados que influenciam no processo de tomada de decisão do modelo de aprendizado de máquina. Isso é feito de modo determinístico para atributos numéricos, ou seja, sem recorrer a aproximações ou algoritmos estocásticos que introduzem instabilidade e desconfiança sobre as explicações geradas. O T-Explainer é baseado em gradientes, calculados por meio da expansão de Taylor, mas não está limitado a uma única classe de modelos de aprendizado, como a maioria dos métodos baseados em gradientes está. Além do mais, o T-Explainer é flexível em gerar explicações locais e também visões gerais de modo otimizado.

Assim, e conforme destacado por Lapuschkin *et al.* (2019), este trabalho contribui com a área do Aprendizado de Máquina ao adicionar uma perspectiva a mais que não tem sido tão bem abordada no discurso sobre as máquinas inteligentes, mas que é fundamental para verificar o comportamento correto de modelos de aprendizado e esclarecer os opacos processos de tomada de decisões que podem afetar os usuários. Além do método T-Explainer em si, o *framework* desenvolvido integra um conjunto de métricas de avaliação de estabilidade e funcionalidades que visam aprimorar os procedimentos de escolha das ferramentas XAI nos mais diversos contextos de aplicação.

Capítulo 5

Resultados Experimentais

5.1 Considerações Iniciais

Este capítulo estabelece as configurações e resultados dos experimentos realizados, colocando em perspectiva a utilidade do T-Explainer ao comparar suas explicações com outros métodos XAI do estado da arte, aplicados sobre uma variedade de conjuntos de dados e modelos classificadores caixa-preta. Além de testar e demonstrar as capacidades da abordagem, o capítulo introduz as ferramentas de pré-processamento desenvolvidas para tratar e padronizar os dados de entrada, os recursos de visualização de informações e os demais métodos integrados ao T-Explainer, utilizados na tarefa de explicar predições.

Na Seção 5.2, são descritas as configurações dos modelos de aprendizado treinados para gerar as predições a serem explicadas, justificando os modelos e as configurações dos hiperparâmetros. A Seção 5.3, apresenta os resultados do T-Explainer face a outros métodos de atribuição de importância sobre dados gerados sinteticamente. Esta seção também define os parâmetros de geração dos dados sintéticos. Na Seção 5.4, estão reportados os testes sobre bases de dados reais. A Seção 5.5 discute os desempenhos computacionais do T-Explainer e a Seção 5.6 contém considerações finais sobre o capítulo.

5.2 Configuração dos Experimentos

Para os experimentos descritos a seguir, foram treinados modelos caixa-preta pertencentes a duas classes diferentes, Redes Neurais *Multi-Layer Perceptions* (MLP), implementados por meio da biblioteca *scikit-learn*, e *Gradient-boosted Tree Ensembles* (baseados em *Random Forests*), utilizando a biblioteca *XGBoost gradient boosting* (CHEN; GUESTRIN, 2016). Vale destacar que outras classes de modelos podem ser consideradas alternativas viáveis aplicadas em testes de métodos XAI, como a Regressão Logística e SVMs (AMPARORE; PEROTTI; BAJARDI, 2021). Entretanto, optou-se por utilizar Redes Neurais e

Random Forests devido à popularidade desses tipos de modelos. Além disso, são modelos altamente não-lineares. De acordo com Breiman (2001), *Random Forests* são impenetráveis no que diz respeito à interpretabilidade. A complexidade dos mecanismos internos torna as aplicações baseadas nesses modelos distantes de serem transparentes, o que acaba levantando a barreira da opacidade, uma vez que é virtualmente impossível interpretar diretamente as razões segundo as quais uma decisão foi tomada por uma Redes Neurais ou *Tree Ensembles* (RIBEIRO; SINGH; GUESTRIN, 2016c).

Tanto as Redes Neurais como os *Tree Ensembles* utilizados foram gerados a partir de conjuntos de dados sintéticos e reais, divididos em porções de 80% das instâncias como dados de treinamento e 20% para testes. Essa divisão foi feita por meio do método *train_test_split* da biblioteca *scikit-learn*, mantendo sempre fixada a mesma variável de *shuffling* (embaralhamento) entre todos os conjuntos. Além disso, foram realizados procedimentos de otimização de hiperparâmetros com o *Grid Search*, de modo a aprimorar o desempenho preditivo dos modelos de aprendizado selecionando hiperparâmetros mais ajustados aos problemas de classificação desta pesquisa.

Para as Redes Neurais, foi inicializada a otimização dos hiperparâmetros configurando a quantidade de neurônios em cada uma das camadas ocultas em potência de 2, seguindo o que é uma prática padrão neste contexto (TAN *et al.*, 2023). Foi definida a ReLU como função de ativação em combinação com o Gradiente Descendente Estocástico (SGD, *Stochastic Gradient Descent*) como otimizador e a *log-loss* como função de perda. Outras combinações de otimizadores e funções de ativação foram experimentadas, mas sem que resultassem em ganhos significativos de desempenho. Os hiperparâmetros dos classificadores *Random Forests* foram otimizados a partir da definição de intervalos contendo quantidades de estimadores (árvores de decisão) e profundidades máximas para cada estimador, utilizando *cross-entropy loss* (entropia cruzada) com métrica de avaliação para a classificação.

De acordo com os resultados dos procedimentos de otimização, foram selecionados três diferentes configurações para as MLPs: uma Rede Neural com três camadas ocultas e 64 unidades neuronais por camada (3H-NN); uma Rede Neural com cinco camadas ocultas e 64 unidades neuronais por camada (5H-64-NN); e uma Rede Neural com cinco camadas ocultas com $[64, 128, 128, 128, 64]$ neurônios em cada camada (5H-128-NN). Estes três modelos utilizam taxas de aprendizado de 0.01, *alpha* 0.0001 e quantidades máximas de épocas de treinamento de 500. As configurações selecionadas foram as que alcançaram os melhores desempenhos nas tarefas de classificação empreendidas aqui, tendo sido baseadas em trabalhos prévios da literatura (BALDI; SADOWSKI; WHITESON, 2014; BORISOV *et al.*, 2022). O treinamento de unidades neuronais e camadas ocultas adicionais aumentou o tempo de treinamento, sem resultar em ganhos significativos de desempenho.

Os classificadores *Random Forest* têm 500 árvores estimadoras com profundidade

máxima de 6 para cada árvore, taxa de aprendizado de 0.01, e *gamma* igual a 1. Os classificadores *Random Forest* treinados seguindo esta configuração de hiperparâmetros serão identificados por XFRC. Todos os demais hiperparâmetros não citados foram definidos como o padrão das respectivas bibliotecas. Esses modelos serão aplicados aqui como as caixas-pretas base para comparar o T-Explainer face a outros métodos XAI em tarefas de explicabilidade por meio de atribuição de importâncias. A Tabela 5.1 sumariza os modelos de aprendizado utilizados nos experimentos.

Tabela 5.1 – Configuração dos modelos classificadores caixa-preta utilizados.

Modelo	Arquitetura	Descrição
3H-NN	Rede Neural	Três camadas ocultas com 64 neurônios por camada. ReLU, SGD, taxa de aprendizado 0.01, *alpha* 0.0001, iterações máximas 500.
5H-64-NN	Rede Neural	Cinco camadas ocultas com 64 neurônios por camada. ReLU, SGD, taxa de aprendizado 0.01, *alpha* 0.0001, iterações máximas 500.
5H-128-NN	Rede Neural	Cinco camadas ocultas com $[64, 128, 128, 128, 64]$ neurônios por camada. ReLU, SGD, taxa de aprendizado 0.01, *alpha* 0.0001, iterações máximas 500.
XRFC	*Tree Ensemble*	500 árvores, profundidade máxima 6, taxa de aprendizado de 0.01, gamma 1.

Fonte: Elaborada pelo autor.

Modelos de aprendizado com arquiteturas baseadas em árvores como os *Tree Ensembles*, impõem desafios extra à explicabilidade por serem classificadores não contínuos, em que valores constantes são armazenados em nós folha. Esta característica torna complexa a aplicação de métodos de explicação, especialmente os baseados em gradientes. Então, é necessário observar que, inicialmente, não se esperava que o T-Explainer apresentasse bons desempenhos sobre os classificadores XRFC.

Os métodos XAI selecionados para servirem de padrão de comparação com o T-Explainer são: SHAP, LIME, *Integrated Gradients*, *Input* $\times$ *Gradient* e DeepLIFT. SHAP e LIME são métodos muito populares dentro do contexto XAI de atribuição de importâncias em que é necessário haver independência do modelo de aprendizado (*model-agnostic*). Embora SHAP e LIME sejam amplamente utilizados, também são reconhecidamente instáveis. Já os métodos baseados em gradientes, como o *Integrated Gradients*, podem ser mais robustos em estabilidade, mas são limitados a modelos com parâmetros diferenciáveis. Então, além da comparação contra métodos consagrados, o T-Explainer será avaliado no contexto de modelos diferenciáveis e não diferenciáveis, testando as habilidades do T-Explainer em contextos demandando independência de modelo.

Especificamente, o SHAP *Explainer* foi utilizado para explicar as Redes Neurais, ao passo que a versão específica do SHAP para modelos baseados em árvores, o SHAP *TreeExplainer* (ou TreeSHAP) (LUNDBERG; ERION; LEE, 2018), foi utilizado sobre os classificadores XRFC (*Tree Ensembles*). Destaca-se ainda que embora SHAP, LIME e os métodos baseados em gradientes citados logo acima tenham sido propostos alguns anos atrás (por volta de 2017), foi decidido comparar o T-Explainer com esses métodos porque eles são atualmente os mais comumente aplicados em tarefas XAI de atribuição de importâncias, tanto em pesquisas como na prática (AGARWAL *et al.*, 2022b).

Para os experimentos requerendo perturbação de dados, foi utilizado o método *NormalPerturbation* da biblioteca OpenXAI (AGARWAL *et al.*, 2022b) para gerar vizinhanças perturbadas, individualmente, a cada instância. Todas as perturbações seguem a mesma configuração, com $\mu = 0$, $\sigma^2 = 0.001$, porcentagem de salto (*flip percentage*) $\varepsilon_p = 0.0001$, e a máxima distância da perturbação de $h_{min}/2$ (h_{min} varia de acordo com cada conjunto de dados), garantindo vizinhanças com pequenas perturbações centradas na instância sob avaliação. Também foram definidos o *clipping threshold* para tratar valores próximos a zero, $\epsilon_c = \pm 10^{-5}$, e a norma l_p utilizada, que é a euclidiana ($p = 2$).

Os resultados dos testes quantitativos serão apresentados em tabelas comparativas. Para facilitar a leitura dessas tabelas e evitar erros de arredondamento, valores superiores a 10^5 serão representados no formato de notação científica, enquanto que valores muito pequenos, isto é, inferiores a 10^{-10}, serão considerados zero. Por conveniência, foi adotado o padrão de separação decimal utilizando ponto, em vez da vírgula.

5.3 Análise Comparativa – Dados Sintéticos

Segundo Amparore, Perotti e Bajardi (2021), as técnicas XAI de maior sucesso atualmente são capazes de explicar uma variada gama de modelos caixa-preta, operando sobre diferentes tipos de dados como, por exemplo, imagens e texto. Porém, esses métodos normalmente mapeiam dados complexos em representações interpretáveis no formato tabular. Neste contexto, imagens podem ser segmentadas em vetores de *superpixels* (regiões ou agrupamentos de *pixels* que guardam alguma característica de similaridade pré-definida) ou mesmo pode haver a conversão direta dos *pixels* em formato tabular; enquanto dados textuais podem ser convertidos em vetores de frequências de palavras. As explicações são geradas em termos das colunas (isto é, atributos) das representações tabulares. Então, e sem perda de generalidade, os resultados deste trabalho se concentram em gerar explicações sobre conjuntos de dados tabulares para classificação binária (duas classes), algo canônico em Aprendizado de Máquina (XENOPOULOS *et al.*, 2022). Além disso, experimentos que cobrem tarefas de classificação binária são uma prática comum em testes sobre métodos

como LIME e SHAP, por exemplo (AMPARORE; PEROTTI; BAJARDI, 2021).

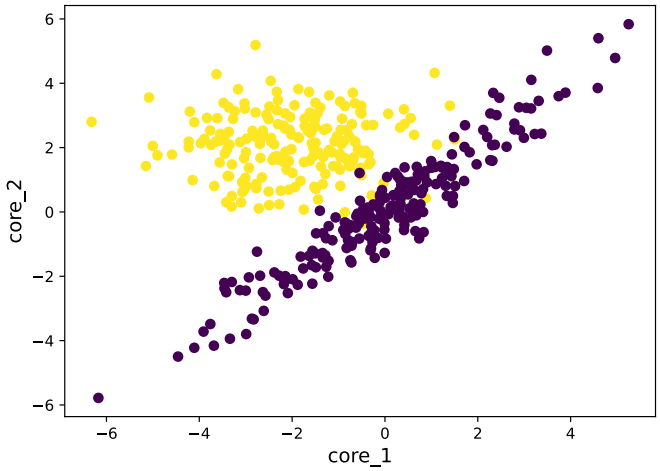
Antes de partir para testes sobre bases de dados reais, com características e distribuições previamente desconhecidas, é interessante proceder uma série de testes utilizando ambientes controlados, induzidos por dados formatados sobre os quais se conhece as suas nuances e comportamentos. Para isso, foram geradas quatro bases de dados sintéticas diferentes, cada uma contendo um total de 1000 instâncias.

5.3.1 Atributos Sintéticos Numéricos

O primeiro conjunto é 4-dimensional, isto é, contém 4 atributos (4-FT) em que cada instância de dado $\mathbf{x}$ é gerada do seguinte modo: as instâncias foram rotuladas distribuindo igualmente a variável objetivo $\mathbf{y} \in [0, 1]$ entre as metades do conjunto e, assim, cada classe tem 500 instâncias. Condicionado ao valor de $\mathbf{y}$, foram amostradas as instâncias $\mathbf{x}$ como $x_{1:2} \sim \mathcal{N}(\mu_y, \Sigma_y)$. Foram definidos $\mu_0 = [0, 0]^{\mathrm{T}}$ e $\mu_1 = [-2, 2]^{\mathrm{T}}$, $\Sigma_0 = [[1, 1], [-1, 1]]^{\mathrm{T}}$ e $\Sigma_1 = \mathbf{I}$, em que μ_0, Σ_0 e μ_1, Σ_1 denotam as médias e as covariâncias das distribuições Gaussianas associadas com as instâncias das classes 0 e 1, respectivamente. Os atributos $x_{1:2}$ foram denominados *core_1* e *core_2*, pois são os atributos de fato preditivos. Valores aleatórios foram atribuídos às variáveis $x_{3:4}$ (*noise_1* e *noise_2*). Essa configuração resulta em um conjunto de dados contendo dois atributos importantes (preditivos) e dois atributos não importantes (ruído aleatório), ou seja, não é esperado que os dois atributos-ruído influenciem na predição de modo determinante ou mesmo significativo.

A Figura 5.1 ilustra as distribuições dos atributos *core_1* e *core_2*. Note que *core_1* e *core_2* foram configuradas de modo a apresentar distribuições com perfis de espalhamento claramente distintos, mantendo um bom nível de separação entre classes, mas com pequena região de mistura entre elas, o que foi feito propositalmente para introduzir um grau de complexidade para o processo preditivo.

Figura 5.1 – Distribuição dos atributos preditivos do conjunto de dados 4-FT.



Fonte: Elaborada pelo autor.

Esses dados foram projetados de modo a facilitar a verificação do desempenho dos métodos de explicabilidade. É esperado que uma ferramenta de atribuição de importâncias seja capaz de identificar quais atributos de fato influenciam nas decisões preditivas do modelo, quantificando sobre estes atributos valores mais significativos de importâncias. Do mesmo modo, aquilo que não é influente (ruído) deve ser caracterizado como tal ao receber valores de importância pouco expressivos.

O conjunto sintético 4-FT descreve um cenário simplificado em que é claro o que se espera identificar. Buscar garantias empíricas e teóricas de que um método XAI discrimina corretamente o que é importante daquilo que não é, é algo de suma importância para o projeto e construção de uma ferramenta dedicada a gerar explicações confiáveis. Isso porque em uma aplicação real, uma vez que se sabe que as explicações são confiáveis, ao se explicar predições feitas por modelos de aprendizado treinados e operando dados reais, quando o método XAI apontar que atributos considerados irrelevantes receberam pesos mais significativos, então será possível inferir sobre a necessidade de alguma revisão na modelagem, seja sobre a definição do problema ou a parametrização do modelo, ou sobre uma possível adequação das técnicas de pré-processamento aplicadas ou mesmo a necessidade de investigação de algum tipo de viés desconhecido nos dados.

O método gerador do conjunto 4-FT (*synthetic_data_generator*) foi integrado ao T-Explainer, podendo ser aplicado para produzir outros dados com quantidades arbitrárias de instâncias. O conjunto especificamente utilizado nos resultados descritos nesta seção, encontra-se arquivado no diretório da pesquisa.

O segundo conjunto sintético é significativamente mais robusto do que o primeiro. Trata-se de um conjunto 20-dimensional (20-FT) criado a partir de amostragens Gaussianas utilizando as ferramentas de geração de dados sintéticos da biblioteca OpenXAI, cuja metodologia e algoritmo estão descritos em Agarwal *et al.* (2022b). De acordo com os autores, o algoritmo garante a criação de dados que encapsulam dependências entre os atributos e vizinhanças locais claramente separadas, propriedades estas que são fundamentais para garantir que as explicações derivadas deste conjunto sintético permaneçam consistentes com o comportamento do modelo treinado sobre tais dados.

Foram utilizados os modelos XRFC e 3H-NN nos experimentos envolvendo dados sintéticos. O modelo XRFC apresentou acurácia de 96% sobre o conjunto 4-FT e 83.5% sobre 20-FT, enquanto o modelo 3H-NN apresentou acurácia de 97.5% sobre o conjunto 4-FT e 83.5% sobre os dados 20-FT. Os experimentos de *benchmark* (testes comparativos) descritos nesta seção não se baseiam em conjuntos de dados sintéticos contendo quantidades massivas de instâncias. É de amplo conhecimento a necessidade de se utilizar uma quantidade de dados significativa para treinar e testar modelos de aprendizado de máquina, de modo que estes representem adequadamente o contexto dos dados. No entanto, de

acordo com Aas, Jullum e Løland (2021), métodos de *feature importance* bem conhecidos tendem a se tornar instáveis quando submetidos a conjuntos de dados com mais de dez dimensões. Neste sentido, a dimensionalidade é um ponto crítico para avaliar a estabilidade dos métodos de *feature importance*. Logo, optou-se por gerar conjuntos de dados sintéticos com uma quantidade de instâncias que fosse suficiente (1000 instâncias para cada conjunto) para treinar e testar os modelos e avaliar a estabilidade dos métodos XAI com eficiência.

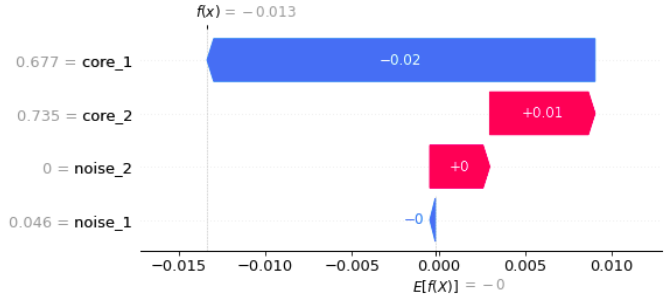
A Figura 5.2 apresenta um cenário interessante gerado por meio do TreeSHAP (Figura 5.2a) e do T-Explainer (Figura 5.2b), explicando a predição da mesma instância do conjunto de dados 4-FT com o modelo XRFC. Conforme esperado para as explicações de instâncias do conjunto 4-FT, o TreeSHAP atribuiu o valor mais significativo de importância para o atributo *core_1* (seguido pelo atributo *core_2*). No entanto, note que o TreeSHAP atribuiu a importância de *core_1* com sinal negativo, fazendo com que a reconstrução aditiva da predição (Equação 4.6) feita pelo TreeSHAP resultasse em -0.013, enquanto o verdadeiro valor predito para esta instância foi 0.0134. Na prática, isso significa que o TreeSHAP está indicando que a instância foi incorretamente predita pelo modelo, pois esta deveria, na verdade, pertencer à classe oposta a qual foi predita. Entretanto, ao checar a verdadeira classe da instância, nota-se que a predição do modelo está correta. Esse resultado do TreeSHAP claramente viola a propriedade da fidelidade local (*local accuracy*). Enquanto isso, o T-Explainer também apontou o atributo *core_1* como o mais relevante para a predição, mas com um sinal positivo correto que mantém a explicação coerente com a predição e também com a verdadeira classe da instância.

Como destacado por Ribeiro, Singh e Guestrin (2016c), explicar uma predição individual fornece algumas noções de entendimento aos usuários a respeito da confiabilidade do classificador. Porém, essa perspectiva singular não é suficiente para validar o modelo como um todo. Então, a partir daqui serão apresentados os resultados dos testes comparativos do T-Explainer sobre volumes maiores de instâncias (a nível de conjunto de dados) e modelos, enfatizando a estabilidade dos métodos de explicabilidade de modo sistemático e não apenas limitado a uma única amostra de dado.

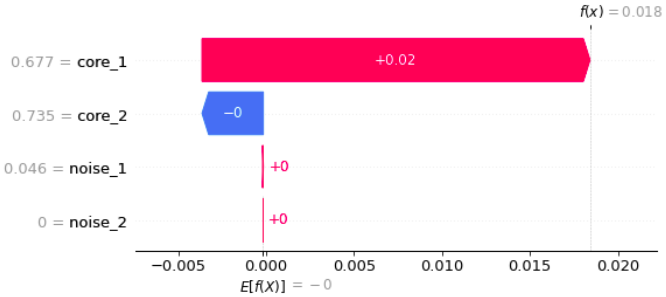
A Tabela 5.2 provê uma visão mais ampla do cenário ilustrado na Figura 5.2. Foi utilizada a métrica FAE (cf. Seção 4.6.3), considerando um limiar de tolerância de 0.01, para avaliar a média e a taxa de preservação da fidelidade local pelos métodos T-Explainer, SHAP e TreeSHAP, quando aplicados sobre as predições do modelo XRFC treinado com o conjunto 4-FT. O TreeSHAP, a versão específica do SHAP para modelos baseados em árvores, apresentou uma taxa de fidelidade aditiva de 0.507 para as suas explicações. O T-Explainer preservou a aditividade a uma taxa de 0.425, ligeiramente menor do que o TreeSHAP. Entretanto, o T-Explainer foi consideravelmente mais preciso nas explicações do que o SHAP *explainer*, a versão *model-agnostic* da biblioteca SHAP.

Figura 5.2 – Importâncias atribuídas por TreeSHAP e T-Explainer para a mesma instância classificada com o modelo XRFC sobre os dados 4-FT.

(a) TreeSHAP *feature importances*.



(b) T-Explainer *feature importances*.



Fonte: Ortigossa *et al.* (2024).

Por outro lado, a Tabela 5.3 também apresenta os resultados da métrica FAE avaliando T-Explainer, SHAP e TreeSHAP, mas agora explicando as predições do modelo XRFC treinado sobre os dados 20-FT. Observe que o T-Explainer superou todos os demais métodos aditivos quanto à preservação da propriedade *local accuracy* no contexto do conjunto 20-FT, com uma taxa de fidelidade aditiva de 0.862, enquanto o segundo melhor colocado, TreeSHAP, obteve 0.620, mais de vinte pontos percentuais inferior ao T-Explainer.

Tabela 5.2 – T-Explainer, SHAP, e TreeSHAP *Local Accuracy* explicando as predições do modelo XRFC treinado no conjunto de dados sintéticos 4-FT.

XRFC	FAE	
XAI	Média	Taxa
T-Explainer	**0.0178**	0.425
SHAP	0.5072	0.007
TreeSHAP	0.0186	**0.507**

Fonte: Ortigossa *et al.* (2024).

A Tabela 5.4 exibe os resultados das métricas RIS, ROS (estabilidade das explicações para perturbações de entrada e saída, cf. Seção 4.6.1) e RES (verifica a similaridade em reiterações, cf. Seção 4.6.2) para os métodos T-Explainer, TreeSHAP e LIME, gerando explicações no contexto do classificador XRFC treinado sobre os dados 4-FT. Note que

Tabela 5.3 – T-Explainer, SHAP, e TreeSHAP *Local Accuracy* explicando as predições do modelo XRFC treinado sobre o conjunto de dados 20-FT.

XRFC	FAE	
XAI	Média	Taxa
T-Explainer	**0.0053**	**0.862**
SHAP	0.4601	0.319
TreeSHAP	0.0103	0.620

Fonte: Ortigossa *et al.* (2024).

o TreeSHAP obteve os melhores resultados em RIS, ROS e RES. O bom desempenho do TreeSHAP era esperado neste cenário, uma vez que o conjunto 4-FT tem baixa dimensionalidade, com o TreeSHAP tirando vantagem de espaços de características com poucas dimensões ao utilizar uma versão determinística do algoritmo de computação dos *Shapley values* (AMPARORE; PEROTTI; BAJARDI, 2021). Observe que a métrica ROS apresenta valores nominais mais elevados se comparados aos da RIS. Isso não é estranho, uma vez que valores próximos a zero são "cortados" pelo *clipping threshold*, definido em $\pm 10^{-5}$, algo que que ocorre com particular frequência nos denominadores da métrica ROS.

Tabela 5.4 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 4-FT.

XRFC	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	13,819	111.4	6.7e+06	38,904	**0**
TreeSHAP	**5,053**	**38.0**	**5.4e+05**	**8,226**	**0**
LIME	10,895	200.4	6.1e+06	86,050	2.7e-04

Fonte: Ortigossa *et al.* (2024).

Já a Tabela 5.5 apresenta as mesmas métricas de estabilidade para o classificador 3H-NN (modelo diferenciável) treinado sobre os dados 4-FT. Neste caso, o T-Explainer obteve desempenhos consistentemente melhores do que os outros métodos XAI em termos de RIS (7 vezes melhor do que o DeepLIFT) e ROS, sendo o segundo mais estável, empatado com o *Integrated Gradients* e SHAP, em termos da métrica RES. O LIME foi o método menos estável nestes testes, para todas as métricas, enquanto os métodos baseados em gradientes apresentaram os melhores desempenhos em estabilidade para perturbações RIS e ROS, algo que pode ser esperado dentro do contexto da explicabilidade de predições feitas por Redes Neurais. Entretanto, observe que DeepLIFT e *Input* × *Gradient* não alcançaram bons resultados para a estabilidade de reiteração (RES) quando comparados aos seus pares, T-Explainer e *Integrated Gradients*, ou quando comparado ao SHAP, que também se posicionou entre os métodos mais estáveis para a reiteração de explicações.

Tabela 5.5 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 4-FT.

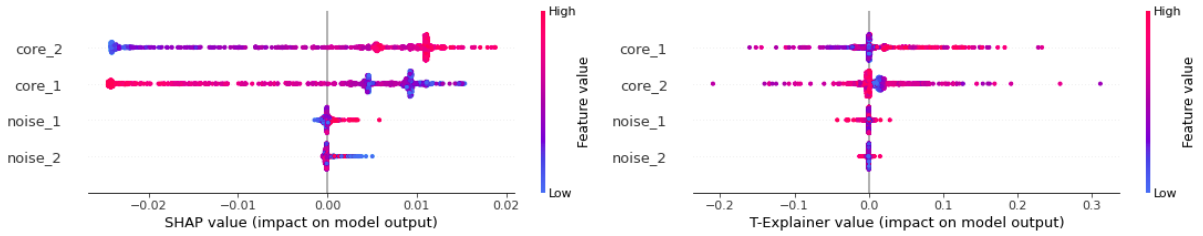
3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**176**	**5.97**	**806**	**5.82**	**0**
SHAP	2,010	38.59	12,122	51.39	**0**
LIME	4,625	127.1	1.7e+07	296.9	2.3e-02
Integrated Gradients	2,465	19.33	1,169	7.41	**0**
Input × Gradient	1,316	9.75	1,535	7.74	1.2e-05
DeepLIFT	1,316	9.75	1,535	7.74	1.1e-05

Fonte: Ortigossa *et al.* (2024).

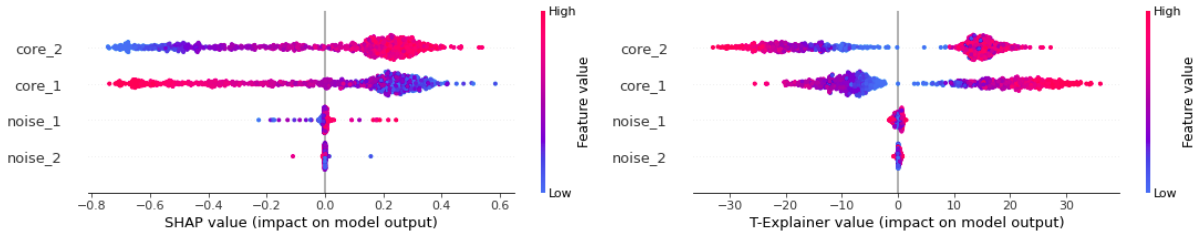
A Figura 5.3 apresenta o perfil geral das explicações geradas por (Tree)SHAP e pelo T-Explainer para as predições sobre o conjunto 4-FT. Tanto o TreeSHAP quanto o T-Explainer identificaram corretamente os atributos *core_1* e *core_2* como os mais importantes globalmente, mas com uma discordância entre estes métodos quanto ao posicionamento destas variáveis. O TreeSHAP destacou *core_2* como o atributo mais importante (Figura 5.3a) enquanto *core_1* foi mais importante para o T-Explainer (Figura 5.3b).

Figura 5.3 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas a partir do conjunto de dados 4-FT.

(a) TreeSHAP e o modelo XRFC. **(b)** T-Explainer e o modelo XRFC.



(c) SHAP e o modelo 3H-NN. **(d)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

Já nas Figuras 5.3c e 5.3d, observa-se que SHAP e T-Explainer também identificaram corretamente os atributos importantes e os irrelevantes nas predições do conjunto 4-FT feitas pela Rede Neural 3H-NN mas, desta vez, ambos os métodos concordaram quanto à ordem de importância dos atributos. Note que, semelhante ao cenário descrito pela

Figura 5.3a, os resultados do SHAP mostram valores mais elevados do lado negativo das importâncias. Já o T-Explainer concentrou as importâncias das variáveis *core*_1 e *core*_2 mais afastadas do eixo-zero, enquanto os atributos contendo ruído aleatório ficaram basicamente centrados ao redor do zero. No geral, os cenários descritos pela Figura 5.3 e pelas Tabelas 5.4 e 5.5, sugerem não somente que o T-Explainer pode ser mais confiável para explicar predições sobre o conjunto 4-FT, mas também que uma Rede Neural é um modelo, em geral, com melhor desempenho preditivo sobre esses dados.

As Tabelas 5.6 e 5.7 descrevem os resultados de estabilidade obtidos a partir da explicação das predições dos modelos XRFC e 3H-NN, treinados sobre o conjunto 20-FT. Analisando a Tabela 5.6, observa-se que o T-Explainer foi superior ao TreeSHAP e ao LIME sob as métricas RIS e ROS, para ambos os cenários de valores, máximos e médios. O TreeSHAP alcançou um desempenho próximo do T-Explainer com relação ao valor máximo de ROS, mas o T-Explainer foi consideravelmente melhor, em média, na ROS. Isso mostra que, embora o T-Explainer tenha oscilado mais dentro de uma quantidade pequena de instâncias, quando tomado o cenário geral, com muitas instâncias, o T-Explainer foi mais estável. Ainda de acordo com a Tabela 5.6, T-Explainer e TreeSHAP apresentaram desempenhos similares na estabilidade RES, tornando ambos os métodos significativamente mais estáveis do que o LIME no contexto de reiteração de explicações.

Tabela 5.6 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 20-FT.

XRFC	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**728**	**29.0**	**2.6e+06**	**35,596**	**0**
TreeSHAP	897	37.8	3.3e+06	64,204	**0**
LIME	1,117	143.6	21.6e+06	3.5e+05	1.7e-04

Fonte: Ortigossa *et al.* (2024).

Curiosamente e diferente do que ocorreu no contexto do conjunto de dados 4-FT, o TreeSHAP apresentou bom resultado para a métrica RES ao explicar as predições feitas pelo classificador XRFC, baseado em *Random Forests* (Tabela 5.6), enquanto o SHAP obteve o pior resultado de estabilidade entre os métodos testados para a reiteração das explicações (RES) de predições feitas pela Rede Neural 3H-NN (Tabela 5.7) sobre o conjunto 20-FT.

Resultados similares podem ser observados na Tabela 5.7, em que o T-Explainer supera todos os outros métodos XAI em termos de RIS (3 vezes mais estável do que o *Integrated Gradients*), com desempenho ligeiramente superior ao *Integrated Gradients* quanto ao ROS médio, mas superando todos os demais métodos segundo o ROS máximo.

Nesta configuração, o SHAP se mostrou o método claramente mais instável, algo que pode ser justificado no contexto do conjunto de dados 20-FT porque, em espaços de alta dimensionalidade, o SHAP utiliza o algoritmo baseado em amostragem em vez do algoritmo determinístico. Embora essa estratégia torne o cálculo de *Shapley values* factível em altas dimensões, isso acaba impondo ao SHAP um comportamento estocástico que leva a explicações instáveis (AMPARORE; PEROTTI; BAJARDI, 2021). Além disso, o SHAP também pode sofrer de modo mais drástico em ambientes complexos por causa das extrapolações (HOOKER; MENTCH, 2019; HOOKER; MENTCH; ZHOU, 2021).

Tabela 5.7 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 20-FT.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**717**	**26.2**	**17,887**	**99.98**	**0**
SHAP	2.8e+05	7,272	1.9e+06	14,076	1.9e-01
LIME	10,182	125.6	28,063	301.9	2.7e-02
Integrated Gradients	2,279	28.4	39,287	107.4	**0**
Input × Gradient	11,815	63.4	60,787	217.2	5.0e-06
DeepLIFT	11,813	63.4	60,781	217.2	4.7e-06

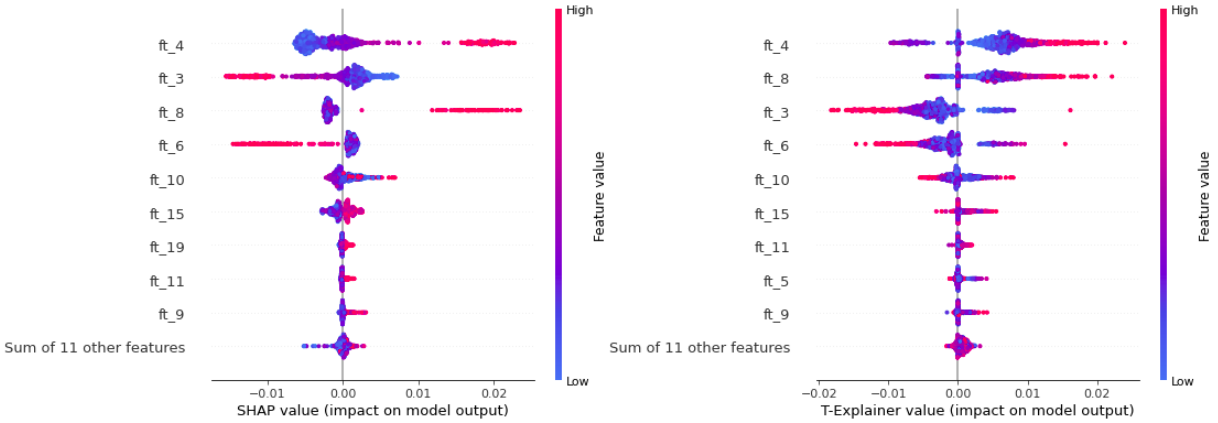
Fonte: Ortigossa *et al.* (2024).

A Figura 5.4 apresenta a visão geral das explicações geradas pelo (Tree)SHAP e pelo T-Explainer para as predições feitas sobre os dados 20-FT. Note que o atributo ft_4 foi identificado como o mais importante em todos os cenários por ambos os métodos de explicação. Embora tenha ocorrido discordância entre os métodos quanto à ordenação das importâncias globais dos demais atributos, com raras exceções, tanto o (Tree)SHAP como o T-Explainer encontraram conjuntos de atributos importantes razoavelmente semelhantes entre si. Vale destacar que houve discordância nos resultados mesmo entre o TreeSHAP e o SHAP, sobre a importância dos atributos nas predições feitas pelo modelo XRFC e 3H-NN, assim como também houve discordância entre as execuções do T-Explainer para os diferentes modelos de aprendizado. Essa discordância não surpreende, uma vez que diferentes modelos geram espaços de classificação diferentes durante o processo de aprendizado. O fenômeno da discordância (*disagreement*) (KRISHNA *et al.*, 2022), que tem motivado diversas pesquisas em XAI, busca os motivos dos métodos de explicabilidade discordarem entre si sob as mesmas configurações de dados e modelos.

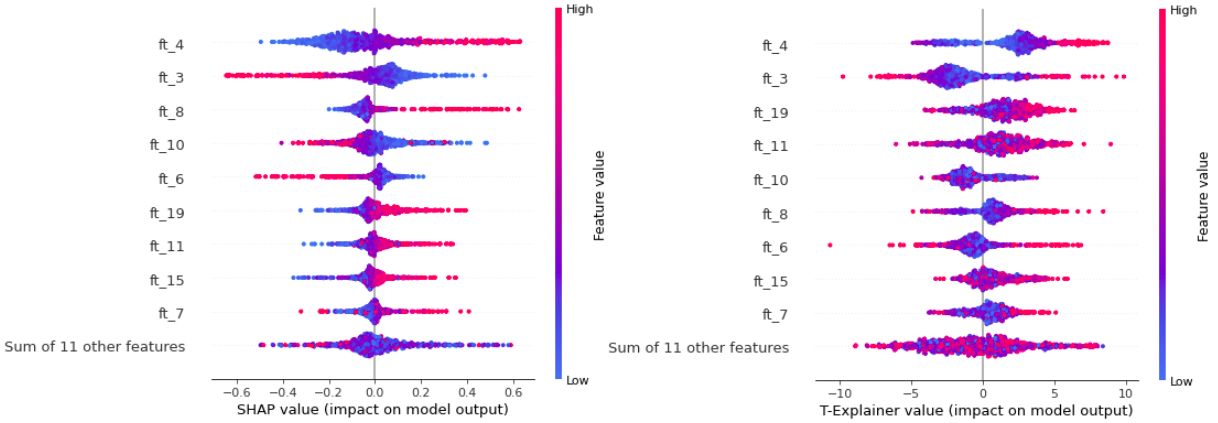
O T-Explainer apresentou um desempenho elevado na preservação da fidelidade local (Tabela 5.3), além de se mostrar como o método mais estável quanto a perturbações ao explicar as predições feitas sobre o conjunto 20-FT. Mas é interessante observar o modo como (Tree)SHAP e T-Explainer concordam no espalhamento das importâncias

Figura 5.4 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre o conjunto 20-FT.

(a) TreeSHAP e o modelo XRFC. **(b)** T-Explainer e o modelo XRFC.



(c) SHAP e o modelo 3H-NN. **(d)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

das demais onze variáveis não representadas nas imagens da Figura 5.4 (última linha de atributos nas imagens da Figura). Para o classificador XRFC, ambos os métodos atribuíram valores reduzidos de importâncias para esses atributos, com o T-Explainer concentrando as importâncias ainda mais próximo ao eixo-zero do que o TreeSHAP (Figuras 5.4a e 5.4b). Enquanto no contexto do modelo 3H-NN, as demais onze variáveis receberam importâncias sob um espectro de valores mais disperso tanto pelo SHAP quanto pelo T-Explainer. Logo, é possível dizer que o (Tree)SHAP e o T-Explainer concordaram razoavelmente bem nas explicações geradas sobre o conjunto 20-FT. Entretanto, o T-Explainer se comportou de modo mais fiel localmente e mais estável do que o (Tree)SHAP.

De modo geral, as Tabelas 5.4, 5.5, 5.6, e 5.7, mostraram que o T-Explainer é robusto em termos da métrica RES, ao se posicionar sempre como o primeiro ou segundo melhor método. O TreeSHAP foi o único método a superar o T-Explainer em alguns dos testes de estabilidade RIS e ROS, muito embora isso tenha ocorrido de modo mais significativo apenas dentro de contextos menos complexos, com dados de baixa dimensionalidade.

5.3.2 Atributos Sintéticos Categóricos

Os demais conjuntos sintéticos utilizados aqui foram gerados do seguinte modo. Primeiro, um conjunto 3-dimensional em que cada instância contém um atributo binário que é transformado em duas colunas, uma para cada valor deste atributo, seguindo o padrão *one-hot encoding* para representação de valores categóricos em numéricos (cf. Subseção 4.5.2). Em outras palavras, as colunas $x_{1:2}$ do conjunto resultante podem ser tomadas como novos atributos, produto da aplicação do *one-hot encoding* sobre um único atributo com dois valores nominais "sim" e "não", por exemplo. As instâncias desse conjunto são rotuladas distribuindo igualmente a variável objetivo $\mathbf{y} \in [0, 1]$ entre as metades do conjunto, de modo que a classe $\mathbf{y} = 1$ esteja associada à coluna x_1, denominada *cat_core_*1, e a classe $\mathbf{y} = 0$ esteja associada à coluna x_2, denominada *cat_core_*2. Valores aleatórios foram atribuídos às variáveis $x_{3:4}$ (*noise_*1 e *noise_*2). Logo, essa configuração gera um conjunto sintético 4-dimensional (identificado por 4-FT-2CAT) contendo duas colunas preditivas derivadas do *encoding* dos valores nominais do atributo categórico, e dois atributos não importantes (ruído aleatório).

No método gerador do conjunto 4-FT-2CAT, foi incluído um parâmetro para a definição de um percentual de mistura entre as instâncias. Esta mistura ocorre invertendo a relação original dos valores nas colunas $x_{1:2}$ com a variável objetivo $\mathbf{y}$. Então, esse parâmetro introduz um grau de incerteza nos dados, de modo a evitar o *overfitting* do modelo de aprendizado, uma vez que, havendo pontos de mistura entre os dois atributos preditivos, estes não podem ser diretamente separados por uma simples reta (modelos lineares). Para o conjunto 4-FT-2CAT utilizado nos resultados descritos a seguir, foi definido um percentual de até 10% de incerteza entre os valores do atributo categórico.

A Tabela 5.8 exibe os resultados das métricas de estabilidade RIS, ROS e RES aplicadas sobre os métodos T-Explainer, TreeSHAP e LIME, explicando as predições do classificador XRFC treinado sobre os dados 4-FT-2CAT. Os desempenhos do T-Explainer e do TreeSHAP sobre estes dados podem ser comparados com os seus respectivos desempenhos sobre o conjunto 4-FT, apresentado na Tabela 5.4. No entanto, destaca-se aqui a forte estabilidade do LIME que, diferente do que foi visto até então, se mostrou um método XAI pouco estável.

Este desempenho do LIME não surpreende e pode ser explicado pela baixa dimensionalidade do conjunto 4-FT-2CAT, que tem apenas quatro dimensões em que, na prática, *cat_core_*1 e *cat_core_*2 derivam da conversão numérica de um único atributo preditivo categórico com dois valores. Como são apenas dois valores preditivos, mesmo havendo alguma mistura entre estes valores, a separação linear de $x_{1:2}$ pode ser feita sem grandes dificuldades. Então, o LIME tem um cenário de atuação favorável neste contexto, por ser

baseado na aproximação local do modelo de aprendizado complexo por modelos lineares.

Tabela 5.8 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 4-FT-2CAT.

XRFC	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	21,211	453	6.9e+06	1.3e+05	**0**
TreeSHAP	2,546	46.2	5.4e+06	22,707	3.9e-09
LIME	**69.1**	**1.7**	**3.1e+04**	**435.9**	**0**

Fonte: Elaborada pelo autor.

A Tabela 5.9 apresenta as métricas de estabilidade aplicadas sobre as explicações de predições feitas pelo modelo 3H-NN. Semelhante aos resultados anteriores, o LIME se destacou. Entretanto, é possível observar que o T-Explainer teve um bom desempenho quando comparado aos seus pares baseados em gradientes e, principalmente, comparado ao SHAP. Note que a estabilidade do T-Explainer foi ligeiramente inferior quanto à estabilidade do *Integrated Gradients* quanto a perturbações RIS, mas superior ao mesmo em relação a perturbações de saída do modelo. Já o *Input* × *Gradient* e o DeepLIFT foram os métodos baseados em gradiente menos estáveis nestes testes mas, ainda assim, com desempenhos consideravelmente melhores do que os apresentados pelo SHAP.

Tabela 5.9 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 4-FT-2CAT.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	652.2	12.8	3,015	57.1	**0**
SHAP	1.3e+05	451.1	2.3e+06	5,640	**0**
LIME	**3.25**	**0.3**	**186.1**	**2.5**	**0**
Integrated Gradients	523.2	7.7	3,938	57.9	**0**
Input × Gradient	2,586	24.9	16,405	116.5	6.0e-07
DeepLIFT	2,586	24.9	16,405	116.5	6.0e-07

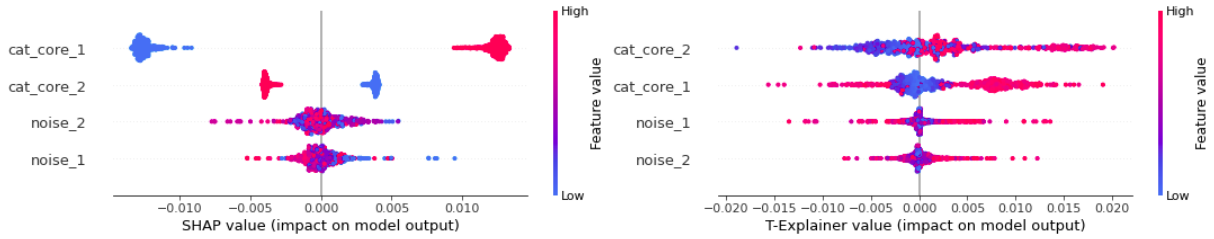
Fonte: Elaborada pelo autor.

A Figura 5.5 representa o perfil global das explicações geradas por (Tree)SHAP e T-Explainer. Note que os métodos XAI identificam corretamente *cat_core_*1 e *cat_core_*2 como os atributos mais importantes, mas há uma significativa diferença entre a visão geral das explicações, de acordo com o modelo de aprendizado subjacente. Segundo as explicações das predições do modelo baseado em *Random Forests*, as importâncias de *cat_core_*1 e *cat_core_*2 geradas pelo T-Explainer (Figura 5.5b) tem um perfil de espalhamento mais uniforme ao longo dos eixos horizontais (impacto sobre a predição modelo), se comparado ao TreeSHAP (Figura 5.5a), que agrupou as importâncias desses

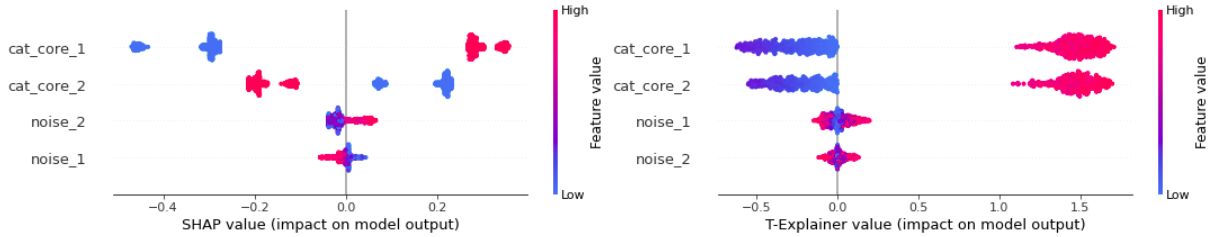
atributos com menor raio de dispersão. No entanto, o TreeSHAP atribuiu importâncias dentro de um intervalo de valores menores para *cat_core_2*, algo que, pela construção do conjunto 4-FT-2CAT, não deveria ocorrer (ambos os atributos têm o mesmo peso).

Figura 5.5 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre o conjunto de dados 4-FT-2CAT.

(a) TreeSHAP e o modelo XRFC. **(b)** T-Explainer e o modelo XRFC.



(c) SHAP e o modelo 3H-NN. **(d)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

Já para as predições da Rede Neural 3H-NN, o T-Explainer apresenta um resultado bastante interessante (Figura 5.5d). Observe que o T-Explainer foi capaz de identificar as duas variáveis preditivas com larga margem dentro dos eixos horizontais de importâncias, comparado aos eixos dos atributos ruído. Mas mais do que isso, o T-Explainer atribuiu importâncias dentro de intervalos semelhantes entre as variáveis preditivas *cat_core_1* e *cat_core_2*, algo que era esperado, já que ambos os atributos têm peso semelhante na distribuição das instâncias do conjunto 4-FT-2CAT, que é balanceado. Neste sentido, o resultado do T-Explainer pode ser considerado superior em comparação ao do SHAP (Figura 5.5c) que, apesar de também identificar corretamente as variáveis preditivas, atribui importâncias dentro de amplitudes de valores que não condizem com o esperado, segundo a construção de *cat_core_1* e *cat_core_2*.

Ambos os modelos, XRFC e 3H-NN, obtiverem acurácias de 90.5% sobre o conjunto 4-FT-2CAT. Analisando os resultados da Figura 5.5 e das Tabelas 5.8 e 5.9, é possível dizer que o desenvolvedor de uma aplicação de aprendizado de máquina trabalhando sobre os dados 4-FT-2CAT, faria uma escolha mais interessante ao implementar uma Rede Neural com explicações geradas por um método baseado em gradientes, como o T-Explainer. Porém, se a escolha fosse o classificador *Random Forest*, o TreeSHAP é mais interessante.

A segunda base de dados sintética com atributos categóricos foi construída como um

híbrido entre os conjuntos 4-FT e 4-FT-2CAT. Mais especificamente, esta segunda base, chamada 6-FT-2CAT, é composta pelo atributo categórico binário de uma base construída com o método gerador dos dados 4-FT-2CAT, e pelos quatro atributos numéricos de uma base construída com o método gerador dos dados 4-FT. Logo, o método responsável por gerar o conjunto 6-FT-2CAT concatena o atributo categórico de uma base do tipo 4-FT-2CAT, com os atributos numéricos de um conjunto do tipo 4-FT. Após o *encoding* do atributo categórico, o resultado é um conjunto sintético 6-dimensional contendo duas colunas preditivas derivadas dos valores nominais (*cat_core_*1 e *cat_core_*2), dois atributos preditivos numéricos baseados em distribuições Gaussianas (*core_*1 e *core_*2), além de duas variáveis com ruído aleatório (*noise_*1 e *noise_*2). Estes atributos seguem as configurações de construção dos dados 4-FT e 4-FT-2CAT.

A Tabela 5.10 apresenta a estabilidade das explicações geradas sobre as predições do modelo XRFC treinado nos dados 6-FT-2CAT. A exemplo da Tabela 5.8, o LIME apresentou o melhor desempenho entre os métodos XAI. Entretanto, observam-se números gerais de estabilidade do T-Explainer e do TreeSHAP bastante diferentes na Tabela 5.10, quando comparados aos números da Tabela 5.8. Sob os dados 6-FT-2CAT, com maior dimensionalidade em relação àqueles do conjunto 4-FT-2CAT, tanto T-Explainer quanto TreeSHAP apresentaram uma significativa melhora na estabilidade geral quanto às perturbações de entrada e saída do modelo (RIS e ROS), com a estabilidade RES se mantendo similar entre as referidas tabelas.

Tabela 5.10 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto de dados sintéticos 6-FT-2CAT.

XRFC	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	5,547	97.1	5.5e+06	58,697	**0**
TreeSHAP	319	11.4	1.4e+06	14,141	6.0e-09
LIME	**65.3**	**3.5**	**2.3e+05**	**2,898**	**0**

Fonte: Elaborada pelo autor.

Na Tabela 5.11 estão os resultados de estabilidade dos métodos XAI aplicados sobre as predições do modelo 3H-NN. Novamente, o LIME se destacou, embora o *Integrated Gradients* também tenha apresentado bons resultados frente aos demais métodos, especialmente em relação ao SHAP que, de longe, foi o método menos estável nesta configuração. O T-Explainer apresentou valores de RIS, ROS e RES próximos aos seus pares baseados em gradientes, superando alguns deles em termos da estabilidade RES. É preciso ponderar que essas métricas trabalham com distâncias entre números pequenos e são bastante sensíveis a perturbações mínimas. Então, valores de estabilidade dentro da mesma ordem de grandeza

podem ser considerados similares. Já diferenças de grande magnitude, como ocorre entre T-Explainer, LIME e *Integrated Gradients* comparados ao SHAP, revelam disparidades mais significativas na capacidade dos métodos em manter a estabilidade das explicações.

Tabela 5.11 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto sintético 6-FT-2CAT.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	14,036	137.5	3,872	36.3	**0**
SHAP	5.5e+06	13,042	2.4e+05	1,462	**0**
LIME	**141.7**	**5.2**	**644.6**	**5.3**	**0**
Integrated Gradients	992	11.9	1,038	9.1	**0**
Input × Gradient	5,833	42.9	5,535	23.9	8.2e-06
DeepLIFT	5,834	42.9	5,535	23.9	9.5e-06

Fonte: Elaborada pelo autor.

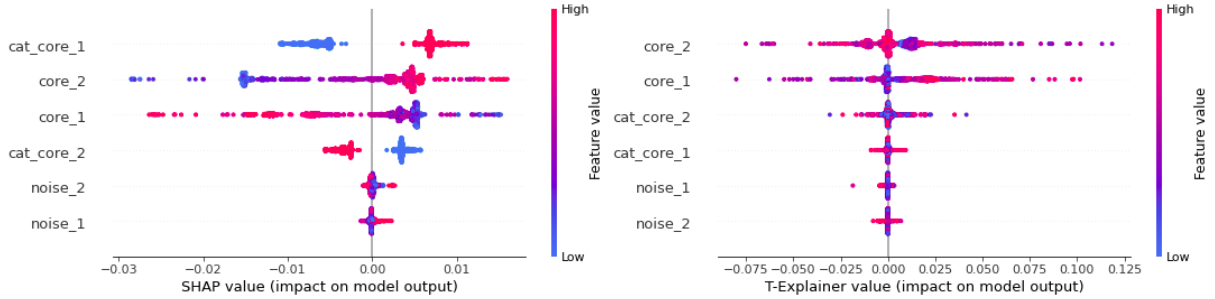
A Figura 5.6 apresenta a visão global das explicações geradas pelo (Tree)SHAP e pelo T-Explainer para instâncias do conjunto 6-FT-2CAT classificadas pelos modelos XRFC e 3H-NN. Tanto TreeSHAP (Figura 5.6a) quanto T-Explainer (Figura 5.6b) identificaram as quatro variáveis que são de fato importantes e os atributos ruído, entretanto, houve uma discordância entre os métodos a respeito do ordenamento das variáveis mais importantes. Para o TreeSHAP, o atributo mais influente dentro das predições do modelo XRFC, foi o valor categórico *cat_core_*1. O T-Explainer posicionou os dois atributos preditivos numéricos, *core_*2 e *core_*1, como os mais importantes, seguidos pelos valores do atributo categórico. Ambos os explicadores atribuíram as menores faixas de importância às variáveis ruído. No entanto, analisando atentamente as Figuras 5.6a e 5.6b, é possível observar que a amplitude horizontal de importâncias atribuídas pelo T-Explainer aos atributos ruído é mais estreita, comparada ao TreeSHAP.

Verificando os resultados do SHAP (Figura 5.6c) e do T-Explainer (Figura 5.6d), explicando as predições feitas pela Rede Neural, observam-se discordâncias mais significativas entre os dois métodos. Para o SHAP, o atributo menos importante é *cat_core_*1, algo que não apenas discorda, mas está em oposição ao resultado gerado pelo próprio TreeSHAP sobre o modelo XRFC (Figura 5.6a). Enquanto isso, o T-Explainer apresentou um comportamento dentro daquilo que se esperava quanto ao posicionamento das variáveis mais e menos importantes, se mostrando consistente com o ordenamento dos atributos preditivos identificados na execução do T-Explainer sobre as predições do classificador XRFC (embora o perfil horizontal das importâncias seja diferente, cf. Figuras 5.6b e 5.6d).

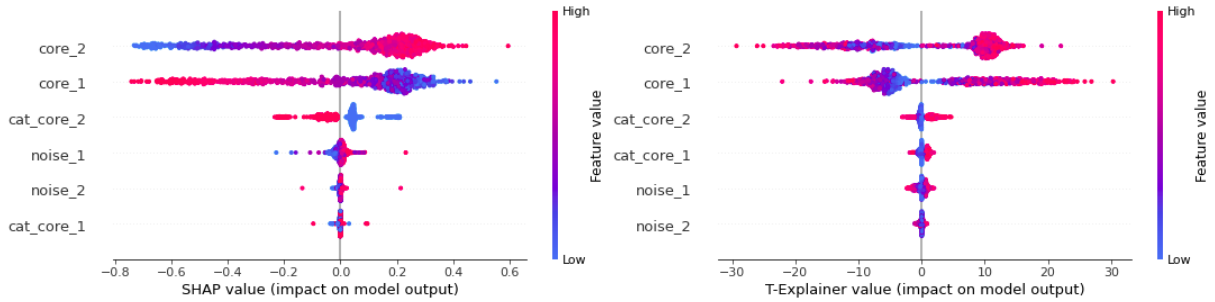
A terceira base de dados sintética contendo atributos categóricos é mais robusta em termos de dimensionalidade. Trata-se de um conjunto com 22 atributos que foi gerado a

Figura 5.6 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre os dados 6-FT-2CAT.

(a) TreeSHAP e o modelo XRFC. **(b)** T-Explainer e o modelo XRFC.



(c) SHAP e o modelo 3H-NN. **(d)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

partir da concatenação de uma base com 20 atributos numéricos gerada com o suporte da biblioteca OpenXAI, seguindo as mesmas definições estabelecidas na Seção 5.3.1 para o conjunto 20-FT, com dois atributos categóricos construídos do seguinte modo: (i) um atributo (cat_noise) contendo dois valores nominais $\{a, b\}$, em que cada um destes valores foi distribuído de modo equivalente entre as instâncias associadas com as classes 0 e 1; (ii) um atributo (cat_core) contendo três valores nominais $\{f, g, h\}$, em que os valores f e h estão associados com a classe 1, enquanto o valor g está associado com a classe 0. Foi definido um percentual de até 25% de incerteza para os valores do atributo cat_core, de modo a introduzir algum grau de mistura entre o mapeamento desses valores com as suas respectivas classes, reduzindo a chance de *overfitting* dos modelos sobre este atributo. Note que, ao distribuir igualmente os valores de cat_noise entre as classes do conjunto, cria-se um atributo sem relevância preditiva, ou seja, cat_noise tem igual importância para a predição dentro da classe 1 ou 0.

Antes de aplicar estes dados no treinamento de modelos de aprendizado, é necessário converter os valores nominais em numéricos. O *encoding* dos atributos categóricos resulta em cinco novas colunas, duas para os valores de cat_noise (cat_noise_a e cat_noise_b) e três para os valores de cat_core (cat_core_f, cat_core_g e cat_core_h). Com isso, o conjunto de dados final, nomeado 25-FT-5CAT, passa a conter 25 dimensões, isto é, 20 numéricas sintetizadas com a OpenXAI e 5 resultantes da conversão dos dois atributos

categóricos. Os modelos XRFC e 3H-NN foram treinados sobre o 25-FT-5CAT, obtendo acurácias de 85% e 82%, respectivamente (valores próximos aos registrados ao treinar esses modelos utilizando o conjunto 20-FT, cf. Seção 5.3.1).

A Tabela 5.12 apresenta os resultados das métricas de estabilidade avaliando os métodos XAI ao explicar as predições do classificador XRFC treinado com o 25-FT-5CAT. O LIME se posicionou como o mais estável. Entretanto, o T-Explainer superou o desempenho do TreeSHAP quanto às perturbações RIS e também em relação à similaridade em reiterações (RES). Embora não tenha superado o TreeSHAP no ROS máximo, em média, o T-Explainer se mostra ligeiramente mais estável do que este método para perturbações de saída. Comparando os resultados da Tabela 5.12 com os resultados anteriores, sobre as predições do classificador XRFC em conjuntos contendo atributos categóricos (Tabelas 5.8 e 5.10), observa-se uma melhora significativa no desempenho do T-Explainer operando em tarefas de explicação para dados em dimensões mais altas.

Tabela 5.12 – Estabilidade dos métodos XAI ao explicar as predições do modelo XRFC treinado no conjunto sintético com atributos categóricos 25-FT-5CAT.

XRFC	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	2,302	46.0	1.4e+07	1.0e+05	**0**
TreeSHAP	2,525	51.0	5.2e+06	1.1e+05	5.7e-09
LIME	**43.9**	**5.3**	**1.4e+06**	**20,025**	**0**

Fonte: Elaborada pelo autor.

Já nos resultados da Tabela 5.13, o T-Explainer se destacou ao explicar as predições da Rede Neural 3H-NN treinada sobre o conjunto 25-FT-5CAT. O T-Explainer foi o único método a manter a estabilidade máxima na métrica RES, alternando com o *Integrated Gradients* a posição de mais estável quanto às perturbações de entrada e saída. Enquanto o T-Explainer foi mais estável para o RIS máximo, o *Integrated Gradients* foi mais estável em média (com os valores médios apresentados por ambos os métodos estando próximos). Já para o ROS máximo, é possível afirmar que T-Explainer e *Integrated Gradients* estejam tecnicamente empatados como os mais estáveis e, novamente, próximos em média.

A Figura 5.7 apresenta a visão geral das explicações geradas por (Tree)SHAP e T-Explainer para as predições feitas com os dados 25-FT-5CAT. A princípio, observa-se que nenhum dos métodos explicadores posicionou qualquer um dos valores do atributo *cat_noise* entre os mais importantes, seja explicando as predições do classificador XRFC ou da Rede Neural 3H-NN. Este resultado é condizente o que se esperava, já que nenhum dos valores de *cat_noise* está associado de modo significativo com uma das classes dos dados e, por isso, estes não deveriam impactar as predições de modo relevante.

Tabela 5.13 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre os dados sintéticos com atributos categóricos 25-FT-5CAT.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**1,877**	40.4	13,958	111.5	**0**
SHAP	4.3e+05	10,417	1.3e+06	19,830	1.4e-01
LIME	6,860	116.8	57,720	348.7	2.5e-02
Integrated Gradients	2,523	**24.9**	**13,815**	**60.6**	4.1e-06
Input × Gradient	3,048	61.4	23,145	159.7	5.3e-06
DeepLIFT	3,048	61.4	23,145	159.7	5.4e-06

Fonte: Elaborada pelo autor.

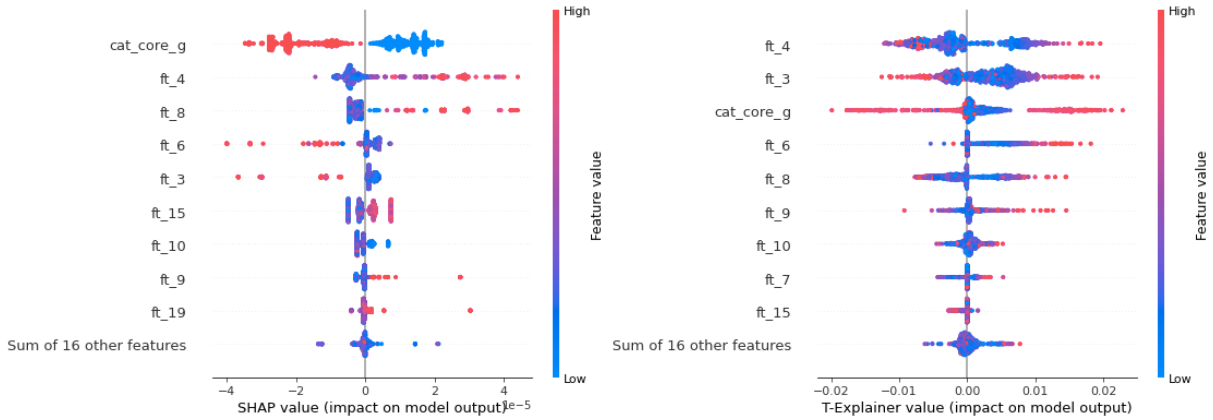
Em relação ao atributo *cat_core*, tanto (Tree)SHAP quanto o T-Explainer identificaram ao menos um dos seus valores entre os mais importantes. Segundo as explicações do T-Explainer, *cat_core_g* está entre os elementos mais relevantes para as predições de XRFC e 3H-NN (Figuras 5.7b e 5.7d), ao passo que este valor está entre os mais importantes segundo o SHAP apenas para as predições de XRFC (Figura 5.7a). Já sobre as explicações resultantes da rede 3H-NN (Figura 5.7c), o SHAP identificou os valores *cat_core_h* e *cat_core_f* entre os elementos mais importantes. Note que, conforme definido logo acima, *cat_core_f* e *cat_core_h* estão associados ao mapeamento de instâncias da classe 1, enquanto *cat_core_g* está associado ao mapeamento de instâncias da classe 0. Embora esta discordância entre os resultados do SHAP seja compreendida como um efeito colateral da arquitetura da variável *cat_core* e do tratamento que o método realiza sobre diferentes modelos, em vez de algum erro de explicação, o T-Explainer se mostrou mais consistente, nesta tarefa, com modelos de aprendizado diferentes.

Comparando os resultados nos *Summary plots* da Figura 5.7, com aqueles alcançados por (Tree)SHAP e T-Explainer explicando as predições de XRFC e 3H-NN treinados sobre os dados 20-FT (Figura 5.4), totalmente numéricos, observa-se um bom nível de concordância geral entre os principais atributos numéricos identificados em ambas as configurações de explicação (algo desejado ao planejar os atributos *cat_core* e *cat_noise*).

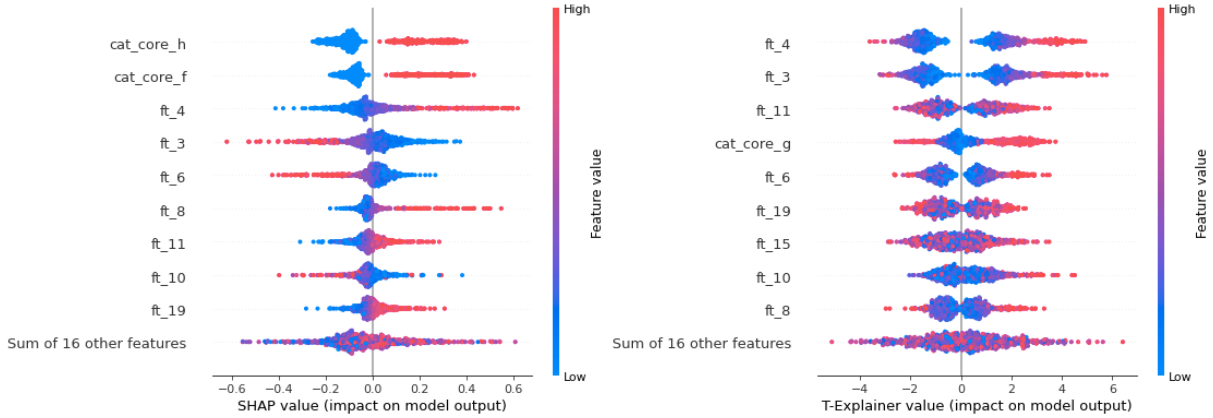
É claro que deve ser levado em consideração o fato que diferentes modelos de aprendizado, mesmo quando treinados sobre o mesmo conjunto de dados, tendem a aprender espaços de características diferentes. Logo, não é inesperado haver divergências entre as explicações geradas sobre predições de modelos diferentes. Avaliar os motivos do *disagreement* entre métodos XAI é tema para uma pesquisa inteira, o que vai além do propósito deste trabalho, que identifica as discordâncias entre as explicações mas evita levantar hipóteses especulativas a respeito deste fenômeno. Porém, é interessante observar a discordância entre os métodos XAI quando atuando dentro das mesmas configurações de modelo e de dados. A partir das propriedades de geração estabelecidas para o conjunto

Figura 5.7 – *Summary plots* com as explicações geradas pelos métodos (Tree)SHAP e T-Explainer para as predições feitas sobre o conjunto 25-FT-5CAT.

(a) TreeSHAP e o modelo XRFC. **(b)** T-Explainer e o modelo XRFC.



(c) SHAP e o modelo 3H-NN. **(d)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

sintético 6-FT-2CAT, esperava-se encontrar cenários como os que foram determinados pelo T-Explainer, LIME e pelo TreeSHAP.

O T-Explainer ainda não está teoricamente desenvolvido para atuar com modelos descontínuos, como os baseados em árvores. Entretanto, seu desempenho foi notável ao explicar as predições feitas pelo classificador XRFC quando a tarefa envolvia dados em dimensões altas, em que o T-Explainer se destacou entre os métodos explicadores mais estáveis (cf. Tabela 5.7). Embora ainda não tenha as garantias teóricas para suportar plenamente modelos não contínuos, o T-Explainer se mostra capaz de obter bons resultados neste contexto, posicionando-se como uma alternativa XAI baseada em gradientes.

5.4 Análise Comparativa – Dados Reais

Nesta seção, serão estendidas as avaliações e comparações realizadas na seção anterior, mas agora aplicando bases de dados reais provenientes de diferentes domínios de

aplicação. Especificamente, foram executados experimentos utilizando seis conjuntos de dados reais bem conhecidos na literatura, contendo propriedades distintas em termos de dimensionalidade, tamanho e presença de atributos categóricos.

O *Breast Cancer Wisconsin* (DUA; GRAFF, 2017) contém 30 atributos sobre parâmetros de medições feitas em massas celulares do peito de pacientes, realizadas durante exames para o diagnóstico de câncer de mama. Cada uma das 569 instâncias foi classificada (diagnosticada) como benigna ou maligna. O *Titanic Disaster* (CUKIERSKI, 2012) reúne informações sobre os passageiros do icônico navio britânico *RMS Titanic*, à época uma embarcação supostamente "inafundável", mas que acabou naufragando na madrugada de 15 de abril de 1912, em sua viagem inaugural pelo Atlântico Norte. Cada uma das 891 instâncias representa um passageiro, que foi rotulado de acordo com o que ocorreu com este passageiro, isto é, se sobreviveu ou não ao naufrágio. O *German Credit Data* (HOFMANN, 1994) contém 1000 instâncias representando indivíduos que buscam por crédito em uma instituição bancária. Cada indivíduo é descrito por atributos categóricos e numéricos, utilizados para classificar seu risco de crédito em bom ou ruim.

Já o *Banknote Authentication* (LOHWEG, 2013) possui 1372 instâncias com 4 atributos contendo informações extraídas de digitalizações feitas sobre cédulas bancárias. Cada instância (cédula) foi classificada como genuína ou falsa. O conjunto *Home Equity Line of Credit* (HELOC) (FICO, 2019) contém atributos financeiros de aplicações anonimizadas para a obtenção de linhas de crédito para o financiamento residencial submetidas por 9871 indivíduos. A tarefa no conjunto HELOC é predizer se um candidato ao crédito tem um bom ou mau risco de pagamento de sua conta HELOC. O maior conjunto de dados desta pesquisa é o HIGGS (BALDI; SADOWSKI; WHITESON, 2014), contendo 28 atributos sobre eventos de colisão simulados para distinguir entre sinais com *Higgs bosons* ou processos de fundo. O conjunto HIGGS original possui 11 milhões de instâncias, mas foi utilizada aqui a versão de 98 mil instâncias disponível na *OpenML* (VANSCHOREN *et al.*, 2014). Esses conjuntos de dados reais foram selecionados por apresentarem grande variedade de tamanho (quantidade de amostras), dimensionalidade (quantidade de atributos) e também devido aos diferentes balanços entre atributos numéricos e categóricos, impondo uma variada gama de desafios aos métodos XAI. A Tabela 5.14 sumariza as características dos conjuntos reais selecionados.

Apesar do T-Explainer ter demonstrado bons resultados ao gerar explicações sobre dados sintéticos, superando os outros explicadores sobre dados contínuos de alta dimensionalidade, os modelos de aprendizado com arquiteturas baseadas em árvores impõem desafios extra para os métodos XAI. Neste sentido, é preciso destacar que a versão atual do T-Explainer não tem respaldo teórico para operar em modelos baseados em árvores, os quais são descontínuos e não diferenciáveis, inviabilizando o cálculo das derivadas. Embora não

Tabela 5.14 – Principais propriedades dos conjuntos de dados reais desta seção. *Cancer* e *German* se referem aos conjuntos *Breast Cancer Wisconsin* e *German Credit*, respectivamente.

	Cancer	*Titanic*	*German*	*Banknote*	HELOC	HIGGS
#Instâncias	569	891	1,000	1,372	9,871	98,050
#Atributos num.	30	4	4	4	21	28
#Atributos cat.	0	3	5	0	2	0
#Classes	2	2	2	2	2	2

Fonte: Elaborada pelo autor.

apropriado para modelos de árvore, o T-Explainer ainda se mostrou competitivo neste cenário adverso. Por esta razão os testes comparativos mais abrangentes envolvendo dados reais utilizam apenas Redes Neurais. Redes Neurais são modelos diferenciáveis, atendendo aos requisitos teóricos que suportam o T-Explainer. As questões relacionadas aos modelos baseados em árvores e o T-Explainer serão abordadas em detalhes na Seção 6.1.

5.4.1 Atributos Reais Numéricos

A Tabela 5.15 apresenta os testes de estabilidade dos métodos XAI explicando as predições feitas pelo modelo 3H-NN treinado sobre o conjunto *Breast Cancer Wisconsin*. Os métodos baseados em gradientes obtiveram os melhores resultados, com amplas margens em relação a LIME e SHAP. O *Integrated Gradients* foi o método mais estável em todos os quesitos avaliados. Embora o T-Explainer tenha ficado atrás dos métodos baseados em gradientes quando são observados os valores máximos das métricas RIS e ROS, é interessante observar que o T-Explainer está próximo do desempenho do *Integrated Gradients* considerando os valores médios de RIS. Além disso, T-Explainer e *Integrated Gradients* foram os únicos a atingir a estabilidade máxima em reiteração de explicações quando (RES). Em relação ao SHAP e ao LIME, o T-Explainer obteve desempenhos superiores e em todas as estatísticas das métricas RIS, ROS e RES.

Tabela 5.15 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto *Breast Cancer Wisconsin*.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	144.3	3.1	1.5e+05	829.0	**0**
SHAP	83,180	2,256	4.7e+07	1.3e+05	1.8e-01
LIME	3,936	39.5	9.9e+05	3,157	2.6e-02
Integrated Gradients	**17.5**	**1.7**	**91,181**	**346.9**	**0**
Input × Gradient	99.3	3.8	1.0e+05	692.8	3.6e-06
DeepLIFT	99.3	3.8	1.0e+05	692.8	3.4e-06

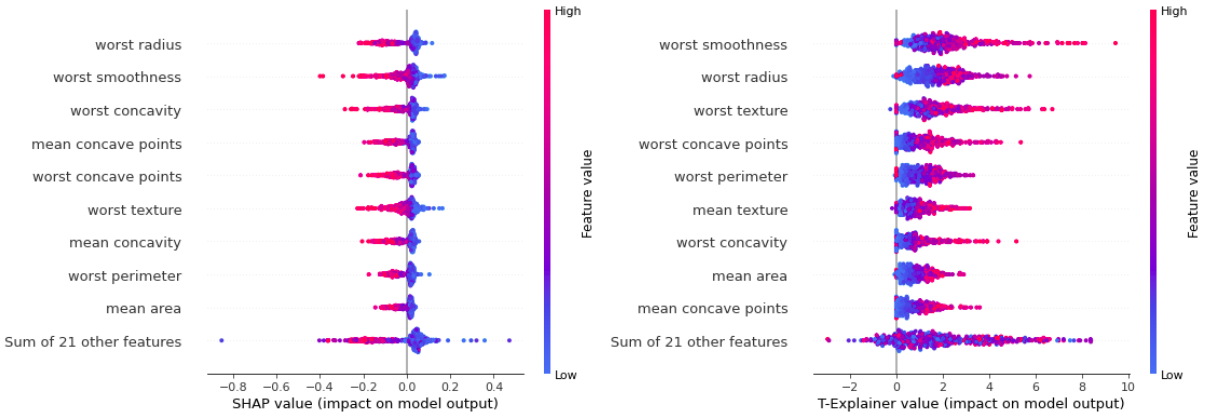
Fonte: Elaborada pelo autor.

A Tabela 5.15 reafirma a boa estabilidade geral do T-Explainer para dados com maior dimensionalidade, como o conjunto *Breast Cancer Wisconsin*, que possui 30 atributos. No geral, os resultados vistos na Tabela 5.15 demonstram que os métodos baseados em gradientes apresentam maior robustez quando a dimensionalidade dos dados aumenta. Esta afirmação se fortalece em comparação aos resultados da Tabela 5.5, em que o desempenho geral de todos os métodos testados foi relativamente próximo operando num espaço de menor dimensão (conjunto 4-FT, com quatro atributos). Neste contexto, sabe-se que o SHAP costuma apresentar bons resultados devido ao uso de seu algoritmo determinístico para o cálculo dos *Shapley values*. No entanto, quando T-Explainer, SHAP e LIME são aplicados em tarefas de explicação sobre dados com vinte ou trinta atributos, torna-se mais clara a vantagem que o T-Explainer apresenta em termos de estabilidade das explicações.

Na Figura 5.8 estão as visões gerais das explicações geradas pelo SHAP e pelo T-Explainer para as predições feitas pelo modelo 3H-NN sobre todo o *Breast Cancer Wisconsin*. Os *Summary plots* da Figura 5.8 listam os nove atributos mais importantes determinados por cada um dos métodos, agregando os demais 21 atributos.

Figura 5.8 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas sobre o conjunto *Breast Cancer Wisconsin*.

(a) SHAP e o modelo 3H-NN. **(b)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

Note que tanto o SHAP quanto o T-Explainer identificaram os atributos "*worst radius*" e "*worst smoothness*" como sendo os atributos mais importantes para classificar casos de câncer de mama. Entretanto, os métodos discordam quanto a qual destes dois atributos é o mais importante. Para os demais atributos importantes que estão listados nas Figuras 5.8a e 5.8b, observa-se um significativo grau de concordância em relação a quais são os atributos identificados como mais importantes. Por exemplo, dentro dos nove atributos mais importantes identificados por SHAP e T-Explainer, oito convergem. Há discordâncias entre estes métodos quanto à ordenação dos atributos mais importantes. Entretanto, desconsiderada a questão das ordenações, é surpreendente o alto grau de concordância

entre os atributos mais importantes sendo que o *Breast Cancer Wisconsin* possui alta dimensionalidade.

Vale destacar que, embora o *Breast Cancer Wisconsin* seja amplamente conhecido e utilizado em problemas de Aprendizado de Máquina, trata-se de um conjunto relativamente pequeno, com menos de 600 instâncias para treinamento e teste de modelo, algo que pode resultar em espaços latente com pouca capacidade de generalização.

Para os experimentos realizados com o conjunto de dados *Banknote Authentication*, foi treinado o modelo 3H-NN, com a Tabela 5.16 trazendo o desempenho dos métodos XAI explicando as predições desta modelagem. Nota-se que o T-Explainer foi menos estável do que o *Integrated Gradients* quanto às métricas RIS e ROS, muito embora tenha se posicionado como o segundo melhor método quando são observados os valores médios destas métricas (com RES similar ao *Integrated Gradients*). A Tabela 5.16 mostra, mais uma vez, que os métodos XAI baseados em gradientes são substancialmente mais estáveis em gerar *feature importances* quanto a perturbações de entrada/saída, em comparação com SHAP. Porém, neste caso, o LIME apresentou um bom desempenho geral, se posicionando como o segundo mais estável para a variação máxima de RIS.

Tabela 5.16 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto *Banknote Authentication*.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	175.3	3.1	429.3	4.1	**0**
SHAP	1.3e+05	468	1.9e+05	543	**0**
LIME	97.6	9.3	849.0	13.4	3.2e-02
Integrated Gradients	**12.4**	**1.5**	429.2	**2.1**	**0**
Input × Gradient	175.3	4.4	**428.0**	4.1	1.6e-05
DeepLIFT	175.3	4.4	**428.0**	4.1	1.6e-05

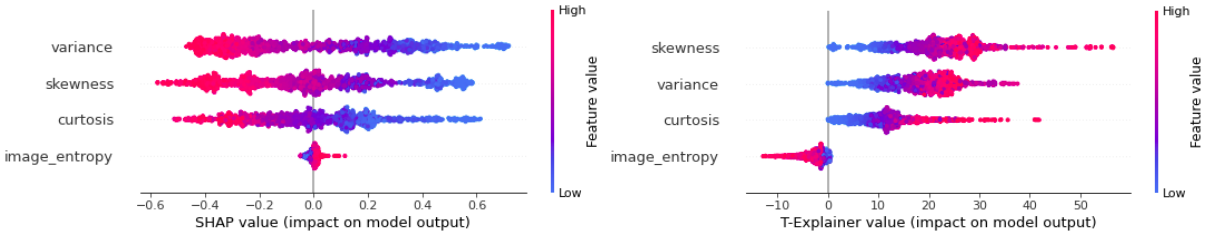
Fonte: Ortigossa *et al.* (2024).

É interessante observar na Tabela 5.16, o desempenho dos métodos baseados em gradientes quanto ao ROS máximo. Neste cenário, DeepLIFT e *Input* × *Gradient* empatam com os mais estáveis para perturbações de saída do modelo, seguidos por *Integrated Gradients* e T-Explainer. Entretanto, a diferença entre os desempenhos destes quatro métodos é pequena, se comparada aos resultados obtidos por LIME e SHAP, que é possível dizer que os quatro métodos baseados em gradientes (classificações esta que inclui o T-Explainer) estão tecnicamente empatados com as melhores estabilidades ROS. Logo, o T-Explainer se revela como um método de explicabilidade estável e capaz de competir muito bem com as técnicas XAI mais eficientes dentro do *feature importance*, tendo, inclusive, superado alguns destes métodos, em média.

A Figura 5.8 apresenta o cenário global das explicações geradas por SHAP e T-Explainer para as predições feitas pelo modelo 3H-NN sobre os dados *Banknote Authentication*. Observe que ambos identificaram *variance* e *skewness* como os atributos preditivos mais importantes. Entretanto, os métodos novamente discordam quanto à ordenação dessas variáveis. De acordo com o T-Explainer, valores nominais mais elevados para os três primeiros atributos, resultam em atribuições de importância mais expressivas (coloração avermelhada dos pontos na Figura 5.9b). Para o SHAP, atributos com valores nominais menores são os que resultam em importâncias positivas maiores, assim como os atributos com valores nominais mais altos impactaram negativamente de modo mais expressivo nas predições (Figura 5.9a). Esse contraste entre os espectros dos valores do atributo (*feature value*) também ocorreu para *image_entropy*, que foi identificado por ambos os métodos, SHAP e T-Explainer, como o atributo menos importante.

Figura 5.9 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas sobre o conjunto *Banknote Authentication*.

(a) SHAP e o modelo 3H-NN. **(b)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

Embora o *Banknote Authentication* tenha uma quantidade maior de instância do que o conjunto de dados anterior, *Breast Cancer Wisconsin*, o *Banknote Authentication* é um conjunto de baixa dimensionalidade (apenas quatro atributos preditivos). Essa baixa dimensionalidade pode ser uma possível explicação para o desempenho razoável obtido pelo LIME nas métricas RIS e ROS, além do SHAP ter se igualado ao *Integrated Gradients* e ao T-Explainer na métrica RES. Ainda assim, no geral, é possível dizer que os métodos XAI baseados em gradientes apresentaram desempenho superior, com o *Integrated Gradients* obtendo o melhor desempenho entre eles.

O conjunto de dados HIGGS é robusto, contando quase 100 mil instâncias e 28 atributos preditivos numéricos. Ou seja, trata-se de um conjunto que reúne as características desafiadoras que os exemplos anteriores apresentam apenas parcialmente – tamanho e alta dimensionalidade. Para avaliar o desempenho dos métodos XAI sobre o HIGGS, foram treinados os classificadores 3H-NN e 5H-64-NN, que obtiveram acurácias de 69.58% e 66.83%, respectivamente. Embora estes valores de acurácia sejam menores do que os observados nos testes anteriores, destaca-se que eles estão próximos de resultados reportados

na literatura (BORISOV *et al.*, 2022). Incluir aqui testes utilizando uma Rede Neural de cinco camadas foi uma decisão tomada com o objetivo de manter similaridade com as arquiteturas propostas em Baldi, Sadowski e Whiteson (2014), em que os pesquisadores exploraram o uso de redes profundas (*Deep Neural Networks*) sobre o conjunto HIGGS.

A Tabela 5.17 apresenta a estabilidade dos métodos XAI explicando as predições do modelo 3H-NN, enquanto na Tabela 5.18 estão os resultados das métricas de estabilidade sobre as explicações geradas para as predições da rede 5H-64-NN. Ambas as tabelas mostram os bons desempenhos em estabilidade dos métodos XAI baseados em gradientes, com o T-Explainer superando todos os seus pares para perturbações de entrada (RIS), enquanto se posiciona como uma alternativa bastante competitiva em termos de estabilidade para perturbações ROS. Note que apenas o T-Explainer e o *Integrated Gradients* alcançaram a estabilidade máxima em reiterações (métrica RES).

Tabela 5.17 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto HIGGS.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**1,063**	**45.0**	69,645	304.4	**0**
SHAP	1.4e+05	5,213	8.8e+05	22,190	2.05e-01
LIME	4,424	90.0	69,458	492.6	3.19e-02
Integrated Gradients	3,833	49.7	**27,719**	**272.4**	**0**
Input × Gradient	2,030	79.9	70,159	604.3	6.62e-06
DeepLIFT	2,031	79.9	70,153	604.3	6.32e-06

Fonte: Ortigossa *et al.* (2024).

Tabela 5.18 – Estabilidade dos métodos XAI sobre as predições do modelo 5H-64-NN treinado no conjunto de dados HIGGS.

5H-64-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**1,359**	**57.7**	65,999	**330.4**	**0**
SHAP	1.3e+05	4,309	7.3e+06	19,999	2.22e-01
LIME	9,582	130.8	46,700	477.2	3.50e-02
Integrated Gradients	15,619	101.1	**37,091**	452.2	**0**
Input × Gradient	4,036	123.9	84,022	558.6	1.67e-05
DeepLIFT	4,036	123.9	84,021	558.6	1.69e-05

Fonte: Ortigossa *et al.* (2024).

Os resultados das Tabelas 5.17 e 5.18 vêm para demonstrar, mais uma vez, a habilidade do T-Explainer em gerar explicações consistentes dentro de tarefas que estão sob diferentes níveis de complexidade de modelos e de dimensionalidade de dados.

A Figura 5.10 descreve a visão geral das explicações geradas por SHAP e T-Explainer para as predições dos modelos 3H-NN e 5H-64-NN sobre o conjunto HIGGS. Neste caso, observa-se um nível maior de discordância entre os métodos XAI quanto aos atributos mais importantes, se comparado ao exemplo da Figura 5.8, que também retrata a visão global dos explicadores sobre um conjunto com alta dimensionalidade. A discordância não se restringe aos cenários entre SHAP e T-Explainer, mas também entre os atributos mais importantes identificados pelo SHAP (T-Explainer) para a rede 3H-NN com aqueles determinados pelo próprio SHAP (T-Explainer) para a rede 5H-64-NN.

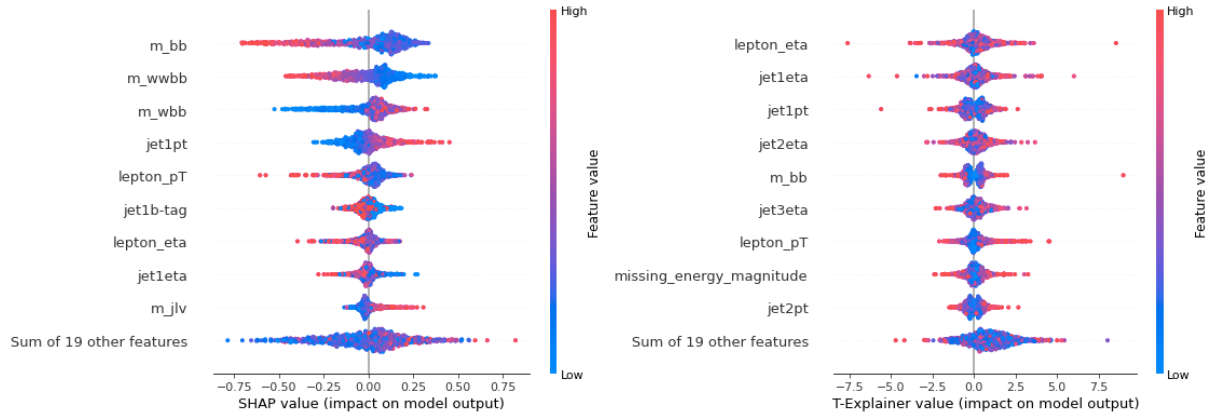
Dos nove atributos mais importantes listados pelos *Summary plots* das Figuras 5.10a e 5.10c, o SHAP concorda em oito deles em suas execuções sobre os modelos diferentes, incluindo os dois atributos identificados como mais importante ("m_bb" e "m_wwbb"). Já o T-Explainer concorda sete vezes entre os nove principais atributos rodando sobre os modelos 3H-NN e 5H-64-NN (Figuras 5.10b e 5.10d), incluindo os dois atributos mais importantes, "lepton_eta" e "jet1eta", embora estes não apresentem a mesma ordenação nos dois gráficos.

Este tipo de discordância entre modelos de aprendizado diferentes pode ser esperada, pois modelos com arquiteturas deferentes tendem a construir espaços latentes que podem ser significativamente diferentes, mesmo quando treinados sobre os mesmos dados. Quanto às discordâncias entre SHAP e T-Explainer dentro de tarefas com o mesmo modelo, há cinco concordâncias (entre os nove atributos mais importantes) para as explicações geradas sobre as predições do modelo 3H-NN (Figuras 5.10a e 5.10b), enquanto para as explicações feitas sobre o modelo 5H-64-NN, foram seis concordâncias (Figuras 5.10c e 5.10d). Um aprofundamento sobre as motivações destas discordâncias pode ser elencado como trabalho para o futuro. Neste contexto, seria valioso ter o suporte de um especialista no domínio dos dados, auxiliando no ajuste do classificador e também na escolha do método XAI explicando as predições, num processo para adequar ambos às características esperadas na modelagem, com o *framework* do T-Explainer oferecendo ferramentas para tal.

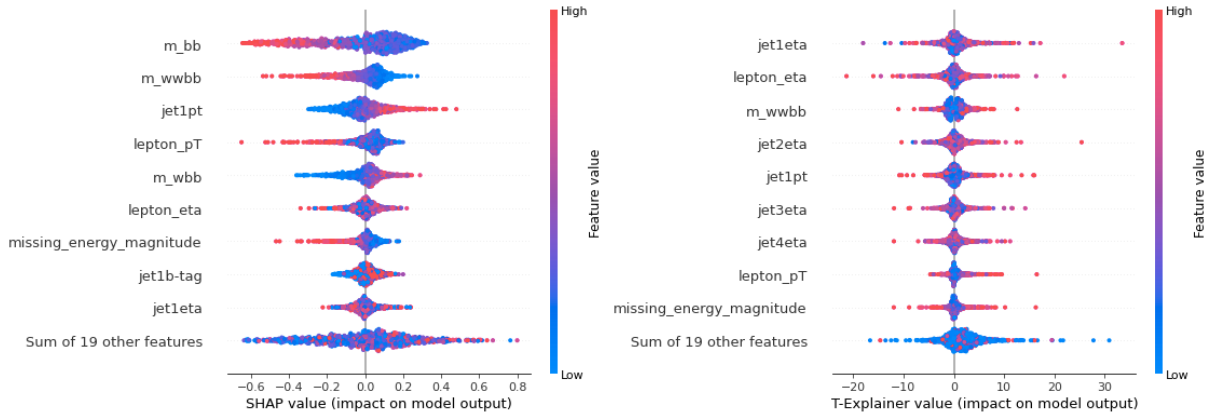
De acordo com as análises reportadas por Amparore, Perotti e Bajardi (2021), o SHAP pode ser, em geral, tão ou mais instável do que o LIME, com as alegadas vantagens sobre a eficiência explicativa do SHAP sendo predominantemente obtidas dentro de tarefas explicativas com dados de baixa dimensionalidade. Os experimentos apresentados ao longo desta seção demonstram que o T-Explainer pode alcançar bons níveis de desempenho, em geral, tendo superando LIME e SHAP *Explainer* (que estão entre os mais bem conhecidos métodos de explicabilidade *model-agnostic*) em tarefas compreendendo dados de alta dimensionalidade e Redes Neurais mais profundas e complexas. Além disso, o T-Explainer também se mostra competitivo em relação aos métodos XAI baseados em gradientes. Com isso, é possível afirmar que o T-Explainer é uma nova e valiosa alternativa dentro

Figura 5.10 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas a partir do conjunto HIGGS.

(a) SHAP e o modelo 3H-NN. **(b)** T-Explainer e o modelo 3H-NN.



(c) SHAP e o modelo 5H-64-NN. **(d)** T-Explainer e o modelo 5H-64-NN.



Fonte: Elaborada pelo autor.

do universo da atribuição de importâncias para a explicação de predições em aplicações apoiadas por Aprendizado de Máquina no contexto de dados numéricos.

5.4.2 Atributos Reais Categóricos

Para completar as análises deste capítulo, foram selecionadas bases de dados reais com atributos categóricos. O conjunto *Titanic Disaster* reúne informações numéricas sobre os passageiros do navio, como a idade e a taxa paga pelo bilhete de embarque, além de atributos categóricos, como o sexo, a classe da cabine e o porto de embarque do passageiro. Alguns atributos desse conjunto não são relevantes para propósitos de classificação apoiada por Aprendizado de Máquina, com o nome do passageiro e o número de série do bilhete de embarque. Por não acrescentarem valor de decisão no treinamento e construção do espaço de características do modelo, estas variáveis foram removidas do conjunto na etapa de preparação dos dados.

Um problema que teve que ser tratado durante o pré-processamento foi com relação aos dados faltantes. Detectou-se a presença de valores faltantes em diversas instâncias, tanto em atributos numéricos quanto categóricos. Existem algumas estratégias bem conhecidas no Aprendizado de Máquina para lidar este tipo de problema, entre elas, a remoção da instância com informação faltante ou o preenchimento desta informação faltante por um valor estatístico.

Excluir instâncias incompletas implica em perda de dados que podem ser importantes para o treinamento do modelo. Esta perda de informações nem sempre acarreta em uma redução significativa no desempenho preditivo. Entretanto, se a quantidade de instâncias com valores faltantes for expressiva, excluir todas estas instâncias pode empobrecer o processo de treinamento do modelo por causa da redução no espaço amostral. Como a base *Titanic Disaster* contém um número relativamente pequeno de amostras (cf. Tabela 5.14), optou-se pela estratégia de preenchimento dos valores faltantes.

Atributos numéricos com valores faltantes foram preenchidos com a média calculada sobre os respectivos atributos (excluindo deste cálculo os valores faltantes). A escolha do valor médio foi motivada porque a média pode ser estatisticamente tomada como um valor "neutro", ou sobre a curva dos dados. Com isso, a instância que teve um atributo preenchido sinteticamente (com um valor que, eventualmente, pode não ser o mais próximo do valor que seria o real), tende a não impor sobre o modelo de aprendizado, para aquele atributo, um comportamento divergente da média. Isso evita comportamentos "fora da curva" (extrapolação) durante o treinamento do modelo, para instâncias com atributos sobre os quais o valor real é inicialmente desconhecido.

Já para valores faltantes em atributos categóricos, optou-se pelo uso da moda. Neste caso, faria pouco sentido usar a média, por exemplo, que tende a ser um valor numérico não inteiro. A moda representa o valor encontrado com maior frequência dentro de cada atributo. No caso de atributos categóricos, a moda também tende a não afetar o comportamento do modelo pois, assim como a média, a moda não representa um valor inesperado (*outlier*) dentro dos dados. Vale destacar que os métodos de pré-processamento implementados no *framework* do T-Explainer para tratamento de dados faltantes, podem preencher valores faltantes tanto em atributos numéricos quanto categóricos, com a média, mediana, moda ou ainda com zeros, a depender das necessidade que o analista tenha identificado dentro da aplicação em desenvolvimento. Existem outras estratégias para o tratamento de dados faltantes, mas uma análise mais acurada destas soluções vai além dos objetivos desta pesquisa.

Com os dados pré-processados, a Rede Neural 3H-NN foi treinada e testada, apresentando acurácia 83.2%. A Tabela 5.19 apresenta a estabilidade dos métodos XAI testados sobre as explicações das predições feitas pelo modelo 3H-NN. Curiosamente, desta vez, o

LIME não se destacou como o método mais estável dentro de um contexto com atributos categóricos, como visto nos conjuntos sintéticos discutidos na Seção 5.3.2. As explicações feitas pelos métodos baseados em gradientes se mostraram as mais estáveis quanto a perturbações de entrada/saída, com o *Integrated Gradients* obtendo o melhor desempenho nas métricas RIS e ROS.

Tabela 5.19 – Estabilidade dos métodos XAI ao explicar as predições do modelo 3H-NN treinado sobre o conjunto *Titanic Disaster*.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	3,165	26.9	27,782	238.3	**0**
SHAP	105,826	1,027	4.1e+06	11,738	1.3e-01
LIME	18,948	43.8	1.2e+05	283.5	1.6e-02
Integrated Gradients	**122.7**	**3.7**	**22,381**	**108.4**	**0**
Input × Gradient	1,923	12.7	47,119	222.1	1.2e-06
DeepLIFT	1,923	12.7	47,119	222.1	1.1e-06

Fonte: Elaborada pelo autor.

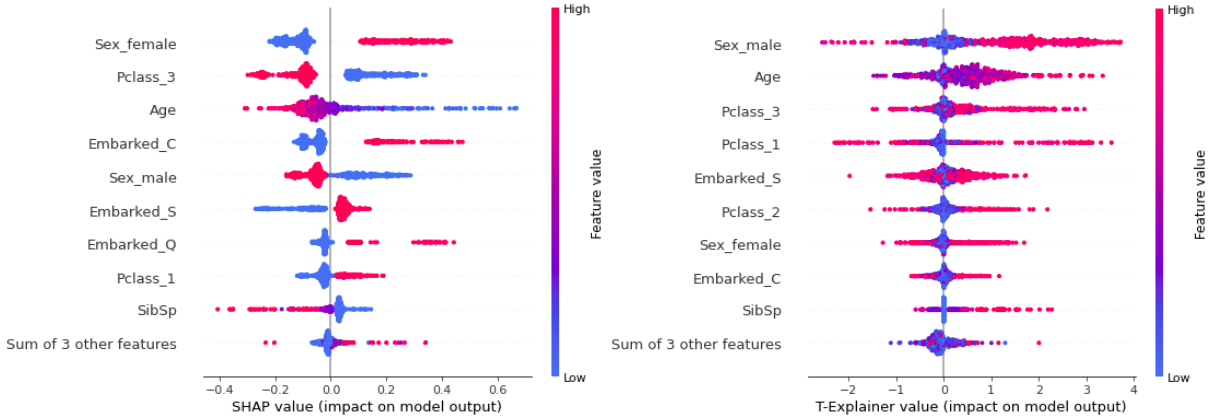
Embora o T-Explainer tenha ficado atrás dos seus pares do paradigma baseado em gradientes, em termos de valores máximos de RIS e ROS, quando são analisados os valores médios destas métricas, nota-se uma proximidade entre o T-Explainer e os demais métodos baseados em gradientes, que diferem entre 3.7 até 26.9 unidades, no caso do RIS médio, e 108.4 a 238.3 unidades para o ROS médio. Comparando estes valores com o desempenho do SHAP, por exemplo, é possível afirmar que os métodos baseados em gradientes são as opções mais estáveis para explicar predições do conjunto *Titanic Disaster* feitas pela Rede Neural, com o *Integrated Gradients* despontando como o mais estável. Note ainda que T-Explainer e *Integrated Gradients* se igualaram como os mais resilientes quanto à similaridade em reiterações (RES).

Na Figura 5.11 estão ilustradas as visões gerais das explicações feitas pelos métodos SHAP e T-Explainer. O atributo indicando o gênero (*Sex*) do passageiro foi desmembrado em duas colunas pelo *one-hot encoding*, "*Sex_male*" e "*Sex_female*". Analisando todas as imagens da Figura 5.11, é possível dizer que "sexo" foi a característica mais importante para determinar a sobrevivência ou não de um passageiro embarcado no *Titanic*. O fato de SHAP e T-Explainer discordarem entre qual atributo é o mais importante, se *Sex_male* ou se *Sex_female*, neste contexto, pode ser considerado um efeito colateral da transformação feita pelo *one-hot encoding*, mas que deve ser esclarecido pelo analista durante o procedimento de explicação.

Analisando essa questão, tem-se que o atributo sexo, que contém apenas os valores binários "*male*" e "*female*", aparenta ser importante dentro da classificação (binária)

Figura 5.11 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para as predições feitas sobre o conjunto *Titanic Disaster*.

(a) SHAP e o modelo 3H-NN. **(b)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

dos passageiros entre "sobrevivente" ou "não sobrevivente". Neste sentido, levanta-se a hipótese de que a discordância entre os explicadores pode ter ocorrido porque o valor *Sex_male* tem forte correlação com uma das classes, enquanto o valor *Sex_female* tem maior correlação com a outra classe. Ou seja, ambos os valores do atributo sexo são importantes para o classificador. Os métodos XAI têm mecanismos de identificação de importâncias diferentes e, em uma situação como esta, com um atributo binário relevante dentro de uma classificação binária, pode haver uma discordância entre os métodos SHAP e T-Explainer quanto a qual valor do atributo sexo, mas ao se re-agregar as importâncias atribuídas a *Sex_male* e *Sex_female* dentro do mesmo atributo original (sexo), tem-se que este é um dos mais importantes para a classificação dos passageiros. Uma solução para este cenário seria que os métodos XAI oferecessem a possibilidade de se atribuir importâncias direcionadas a uma classe especificada pelo usuário, por exemplo. Com isso, os métodos gerariam suas explicações com base em um objetivo de explicação em comum.

De modo geral, os resultados apresentados na Figura 5.11 indicam que os fatores mais impactantes para a sobrevivência de um passageiro no naufrágio do *Titanic*, de acordo com as explicações do SHAP e do T-Explainer, foram o sexo e a classe da cabine em que cada passageiro viajava. Estatisticamente, sabe-se que cerca de 75% do total de mulheres abordo do *Titanic* sobreviveram ao naufrágio, enquanto apenas 20% dos homens sobreviveram. Entre as mulheres, 97% das passageiras da primeira classe se salvaram. Entre os passageiros da terceira classe, a mais numerosa e com maior percentual de homens, cerca de 84% dos passageiros homens não sobreviveram. Por fim, o atributo "*Embarked*", que indica o porto em que cada passageiro embarcou no navio (com três possíveis valores, S, Q e C – Southampton, Queenstown e Cherbourg), mostrou-se menos relevante do que as características sociais (sexo) e econômicas (classe) dos passageiros.

Para os experimentos com o conjunto de dados *German Credit*, foi selecionado o classificador 3H-NN. A versão original deste conjunto possui 20 atributos numéricos e categóricos, mas o conjunto original é praticamente impossível de ser compreendido e aplicado diretamente em um modelo de aprendizado, devido ao complexo sistema de categorias e símbolos utilizados para representar as informações. Por isso, é necessário um bom trabalho de pré-processamento para aplicar o *German Credit* no treinamento de modelos classificadores. Alguns atributos foram excluídos, por serem pouco úteis, com significado obscuro ou translação muito complexa. Após o procedimento de limpeza dos dados, foi obtida uma base resultante com nove atributos (cf. Tabela 5.14):

- *Age* – Numérico, idade do indivíduo solicitante do crédito;
- *Sex* – Categórico, com os possíveis valores "*male*" e "*female*";
- *Job* – Numérico, indica a escala de qualificação profissional do indivíduo;
- *Housing* – Categórico, indica a situação de moradia do indivíduo, com os valores possíveis "*own*", "*rent*" e "*free*";
- *Saving accounts* – Categórico, indica a quantidade de poupança do indivíduo, com os valores possíveis "*little*", "*moderate*", "*rich*" e "*quite rich*";
- *Checking account* – Categórico, indica a situação da conta corrente do indivíduo;
- *Credit amount* – Numérico, valor do crédito solicitado pelo indivíduo;
- *Duration* – Numérico, quantidade de meses do empréstimo solicitado;
- *Purpose* – Categórico, indica a finalidade do empréstimo, com os possíveis valores "*car*", "*furniture/equipment*", "*radio/TV*", "*domestic appliances*", "*repairs*", "*education*", "*business*", "*vacation/others*".

Semelhante ao ocorrido com as instâncias do conjunto *Titanic Disaster*, também foram detectados valores faltantes no *German Credit Data*. A estratégia de tratamento aplicada aqui foi a mesma adotada anteriormente, isto é, os valores numéricos faltantes foram substituídos pela média calculada sobre o respectivo atributo em que cada valor faltante se localiza, ao passo que para os valores faltantes em atributos categóricos, optou-se pela substituição utilizando a moda. Após estas substituições, os atributos categóricos do conjunto foram transformados em colunas numéricas com o *one-hot encoding*, resultando em uma base com 24 colunas numéricas. Então, o classificador 3H-NN foi treinado sobre esta base transformada, obtendo acurácia de 66%. Há que se observar que essa acurácia pode ser aprimoradas por meio do refinamento dos hiperparâmetros do classificador ou mesmo de uma engenharia de dados mais refinada sobre o conjunto *German Credit*, que é complexo, visando estabelecer configurações que favoreçam a criação de espaços de

características mais adequados às particularidades específicas dos dados. Entretanto, o *tuning* (ajuste) de modelos de aprendizado é um procedimento que consome tempo, mesmo utilizando ferramentas dedicadas como o *Grid Search*.

A Tabela 5.20 traz os resultados dos testes de estabilidade dos métodos XAI explicando as predições do modelo 3H-NN sobre o *German Credit Data*. O T-Explainer se destacou como o método mais estável com boa margem em quase todas as estatísticas verificadas pelas métricas. Entre as perturbações de entrada, o LIME foi o que mais se aproximou do T-Explainer quanto ao valor máximo de RIS. Porém, observando o RIS médio, os métodos baseados em gradientes foram mais estáveis do que o LIME, com destaque para o *Integrated Gradients* que, em média, foi quase tão estável quanto o T-Explainer. Curiosamente, este cenário não se repetiu para as perturbações de saída, em que os métodos baseados em gradientes foram todos mais estáveis do que o LIME para a métrica ROS. O T-Explainer foi o mais estável no ROS máximo, seguido pelo *Integrated Gradients*. Já para o ROS médio, o destaque ficou com o *Integrated Gradients* como o método mais estável, seguido com proximidade pelo T-Explainer.

Tabela 5.20 – Estabilidade dos métodos XAI sobre as predições do modelo 3H-NN treinado no conjunto *German Credit*.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**1,971**	**49.4**	**15,735**	239.9	**0**
SHAP	1.0e+06	14,044	2.5e+07	62,883	2.1e-01
LIME	7,547	131.1	1.8e+05	739.9	2.4e-02
Integrated Gradients	8,709	59.5	58,721	**185.7**	**0**
Input × Gradient	10,933	63.2	1.0e+05	350.8	2.8e-06
DeepLIFT	10,934	63.2	1.0e+05	350.8	2.4e-06

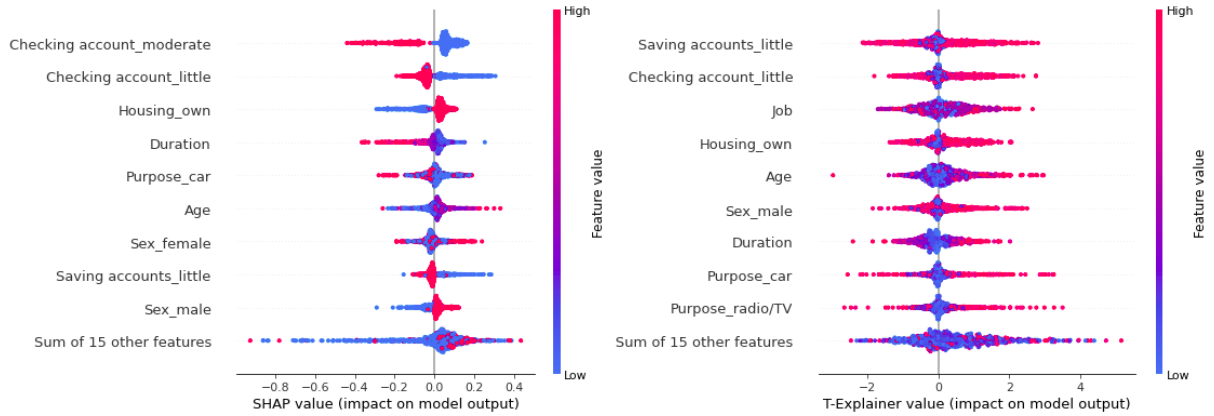
Fonte: Ortigossa *et al.* (2024).

Os gráficos da Figura 5.12 comparam visualmente SHAP e T-Explainer, explicando as predições do modelo 3H-NN sobre o conjunto *German Credit*. Em geral, é possível observar que houve discordância entre os cenários apresentados, principalmente com relação à ordenação dos conjuntos de variáveis mais importantes identificados por SHAP e T-Explainer. Porém, ainda que não exista uma concordância exata quanto ao ordenamento dos atributos mais importantes, é possível encontrar elementos de concordância entre as explicações, que coincidem em seis dos nove atributos mais importantes. Por exemplo, nos *Summary plots* da Figura 5.12, "*Housing_own*" (um valor dentro do atributo categórico "*Housing*", sinalizando que o indivíduo solicitante do crédito possui casa própria) tem presença entre os elementos que mais receberam atribuições significativas. Sabe-se que dentro do contexto da concessão de empréstimos bancários, a posse de patrimônio, como

um imóvel, é um elemento importante para garantir lastro por quem demanda crédito.

Figura 5.12 – *Summary plots* com as explicações feitas pelos métodos SHAP e T-Explainer para as predições geradas sobre o conjunto *German Credit*.

(a) SHAP e o modelo 3H-NN. **(b)** T-Explainer e o modelo 3H-NN.



Fonte: Elaborada pelo autor.

Demais atributos, como "*Duration*" (duração) e "*Age*", também estão entre os mais importantes. Empréstimos de longo prazo tendem a ser considerados mais arriscados pelas instituições financeiras, bem como clientes mais jovens que, a princípio, têm menos histórico financeiro, podem ser considerados como créditos de maior risco. A situação financeira dos clientes se mostra relevante para o SHAP e o T-Explainer, com valores de atributos que definem a situação da conta e o nível de poupança dos indivíduos ("*Checking account*" e "*Saving accounts*") figurando entre os itens mais importantes. Curiosamente, categorias que descrevem as razões dos empréstimos, em geral, não se mostram tão relevantes para a classificação do crédito, ainda que algum valor esteja entre os mais importantes, apenas "*Purpose_car*" (quando o aplicante tem a intenção de utilizar o empréstimo para adquirir um carro) foi relevante para o SHAP e o T-Explainer. Analisando os valores definidos para o atributo "*Purpose*", é possível inferir que "carro" é aquele que demandaria maiores recursos, justificando sua relevância identificada por ambos os métodos.

Outro fato que chama a atenção tem relação com o atributo "*Sex*", que carrega a informação sobre o gênero do indivíduo aplicante. SHAP e T-Explainer posicionam este atributo de modo semelhante entre os mais relevantes. No entanto, "*Sex_female*" (feminino) foi mais importante para o SHAP, ao passo que "*Sex_male*" (masculino) foi mais relevante para o T-Explainer. *Sex* é um atributo binário dentro do conjunto *German Credit*, isto é, apenas os valores feminino e masculino estão definidos. Ainda que não figure nas primeiras posições, estar presente entre os os nove elementos mais importantes em ambos os gráficos da Figura 5.12, faz com que seja possível dizer que tanto SHAP quanto T-Explainer apontam o gênero dos aplicantes como uma influência significativa dentro das predições do modelo 3H-NN. Essa informação revelada pela explicabilidade é

de grande valor pois, neste contexto, conceder ou não crédito para um indivíduo, devem ser observadas questões éticas (*fairness*) e de conformidade (*compliance*), evitando que os sistemas computacionais apresentem viés de gênero, étnicos ou religiosos. Então, a explicabilidade é uma aliada dos desenvolvedores, apontando o que deve ser ajustado nos modelos que integram aplicações de tomada de decisões envolvendo humanos.

O HELOC é um conjunto de dados mais robusto em termos de tamanho e dimensionalidade. Similar ao que ocorreu com os conjuntos *Titanic Disaster* e *German Credit*, foi necessário realizar trabalhos de pré-processamento sobre os dados HELOC. Originalmente, o conjunto tem $10,459$ instâncias, $5,000$ das quais pertencem a classe "*Good*" (bom), ou seja, clientes que tem boas chances de saldar suas contas HELOC dentro de dois anos, enquanto $5,459$ instâncias são classificadas como "*Bad*" (mau). Isso se traduz em uma distribuição razoavelmente balanceada, com 48% de indivíduos bons pagadores e 52% de indivíduos maus pagadores. Entretanto, nem todas as instâncias são únicas. Então, foram excluídas as instâncias repetidas, resultando em um conjunto de dados com $9,871$ amostras únicas (cf. Tabela 5.14), com a mesma distribuição de classes original. Além disso, existem amostras com valores faltantes (marcados dentro do conjunto com o caractere "?"). Seguindo a estratégia aplicada anteriormente, valores faltantes em atributos numéricos foram preenchidos com a média do respectivo atributo, ao passo que valores faltantes em atributos categóricos foram substituídos pela moda.

Para comparar o desempenho do T-Explainer com os demais métodos explicadores, foram treinadas as Redes Neurais 3H-NN e 5H-128-NN sobre o HELOC, obtendo acurácias de 71.34% e 73.27%, respectivamente. Embora estes valores de acurácia sejam menores do que alguns valores vistos anteriormente neste capítulo, são desempenhos que estão em linha com resultados reportados na literatura (BORISOV *et al.*, 2022).

A Tabela 5.21 apresenta o desempenho dos métodos XAI em testes de estabilidade ao explicar as predições do modelo 3H-NN. O *Integrated Gradients* se mostra como o método mais estável para quase todas as estatísticas das métricas RIS, ROS e RES (atrás do LIME apenas para o RIS médio, mas com números bastante próximos). Entretanto, observe o desempenho do T-Explainer, especialmente com relação aos valores médios das métricas. Apenas *Integrated Gradients*, LIME e T-Explainer foram capazes de manter o RIS médio abaixo de dez unidades, algo que pode ser considerado como virtualmente a mesma estabilidade média para perturbações de entrada. Resultados similares são observados no ROS médio, em que *Integrated Gradients* alcançou os menores valores de instabilidade, seguido pelo T-Explainer e LIME.

Na Tabela 5.22 estão os números de estabilidade dos métodos XAI quando aplicados na explicação das predições feitas pela Rede Neural de cinco camadas, 5H-128-NN. Neste caso, o T-Explainer é o mais estável, com o *Integrated Gradients* se posicionando como

Tabela 5.21 – Estabilidade dos métodos XAI explicando as predições do modelo 3H-NN treinado sobre o conjunto HELOC.

3H-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	1,443	8.49	94,943	527.9	**0**
SHAP	84,794	1,330	1.7e+07	64,693	1.5e-02
LIME	509.7	**3.84**	3.7e+05	626.5	9.3e-02
Integrated Gradients	**159.8**	4.21	**85,463**	**401.1**	**0**
Input × Gradient	3,754	33.67	3.5e+05	1,879	8.7e-07
DeepLIFT	3,749	33.64	3.5e+05	1,878	9.0e-07

Fonte: Ortigossa *et al.* (2024).

o segundo melhor quanto a perturbações de entrada (RIS) e o LIME como o segundo melhor para perturbações de saída (ROS). Note que, mais uma vez, apenas T-Explainer e *Integrated Gradients* foram os métodos que alcançaram os maiores níveis de estabilidade RES, em ambos os cenários das Tabelas 5.21 e 5.22.

Tabela 5.22 – Estabilidade dos métodos XAI sobre as predições do modelo 5H-128-NN treinado no conjunto de dados HELOC.

5H-128-NN	RIS		ROS		RES
XAI	Máximo	Média	Máximo	Média	Máximo
T-Explainer	**154.0**	**5.37**	**1.4e+05**	**362.5**	**0**
SHAP	52,924	1,757	1.6e+07	67,451	1.9e-01
LIME	5,083	39.96	2.5e+05	935.0	1.6e-02
Integrated Gradients	255.7	7.21	1.9e+07	19,283	**0**
Input × Gradient	11,769	66.01	4.3e+05	3,215	1.8e-06
DeepLIFT	11,785	66.06	4.3e+05	3,218	1.7e-06

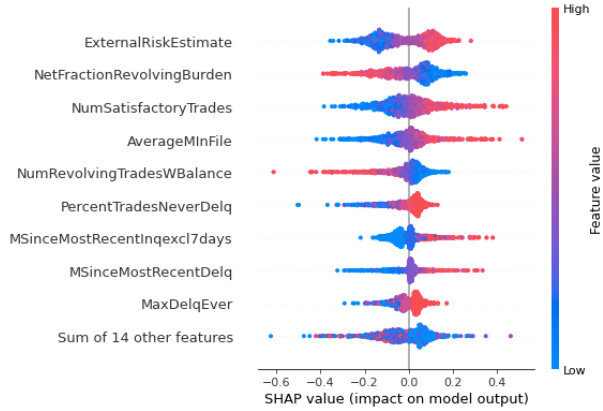
Fonte: Ortigossa *et al.* (2024).

A Figura 5.13 ilustra as visões gerais das explicações geradas por SHAP e T-Explainer para as predições feitas pelos modelos 3H-NN e 5H-128-NN sobre os dados HELOC. Embora à primeira vista esteja evidenciada uma grande diferença entre as distribuições de importância atribuídas por SHAP e T-Explainer, para cada um dos atributos, mesmo quando comparados os resultados do mesmo método explicador mas sobre diferentes modelos. Entretanto, após uma análise mais aprofundada entre os gráficos da Figura 5.13, observa-se que os métodos XAI apresentam boa concordância.

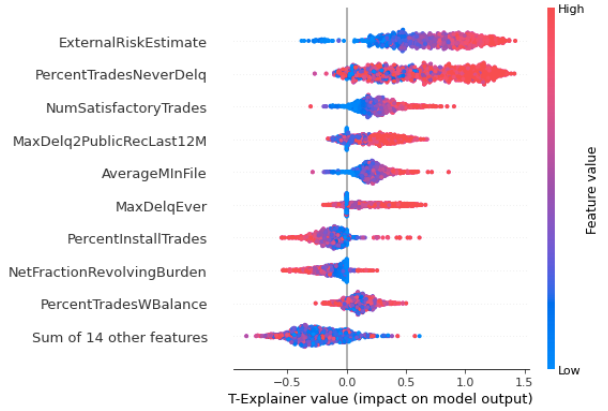
Para o modelo 3H-NN (Figuras 5.13a e 5.13b), SHAP e T-Explainer concordam em seis dos nove atributos mais importantes, ao passo que para a Rede Neural de cinco camadas (Figuras 5.13c e 5.13d), a concordância foi ligeiramente menor, isto é, cinco dos nove atributos mais relevantes. Por exemplo, nas explicações sobre o modelo 3H-NN, SHAP

Figura 5.13 – *Summary plots* com as explicações geradas pelos métodos SHAP e T-Explainer para predições feitas sobre o conjunto HELOC.

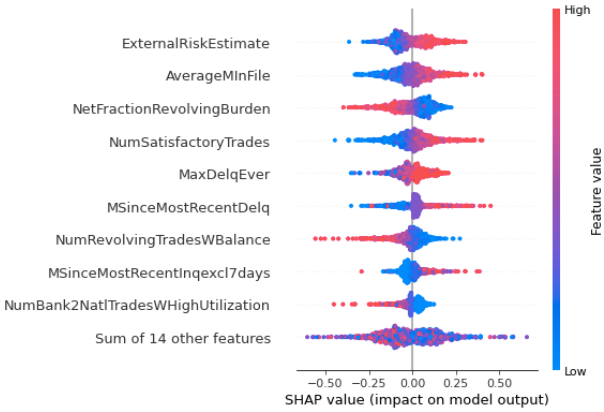
(a) SHAP e o modelo 3H-NN.



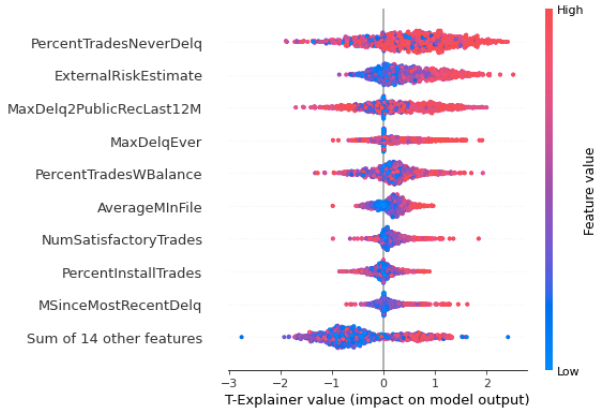
(b) T-Explainer e o modelo 3H-NN.



(c) SHAP e o modelo 5H-128-NN.



(d) T-Explainer e o modelo 5H-128-NN.



Fonte: Elaborada pelo autor.

e T-Explainer chegam a concordar quanto ao posicionamento de alguns dos principais atributos, como foi o caso de "*ExternalRiskEstimate*" e "*NumSatisfactoryTrades*", primeiro e terceiro atributos identificados como mais importantes por SHAP e T-Explainer. Já nas explicações do modelo 5H-128-NN, SHAP e T-Explainer posicionaram o mesmo atributo "*ExternalRiskEstimate*" entre os dois mais relevantes.

É necessário salientar que algum grau de discordância entre métodos XAI é esperado. Entretanto, o que chama a atenção é a boa concordância entre SHAP e T-Explainer, mesmo operando sobre dados de alta dimensionalidade. Desde que foi apresentado, em 2017, o SHAP, bem como as demais técnicas derivadas e que integram o *framework* SHAP, se tornou um dos principais métodos XAI (senão o principal) do paradigma de atribuição de importâncias (*feature importance*). Neste sentido, o T-Explainer ter alcançando bons níveis de concordância com o SHAP é um feito importante alcançado por esta pesquisa. Entretanto, a instabilidade do SHAP em gerar explicações locais é algo conhecido na literatura, foi amplamente debatido neste texto, além de evidenciado pelos resultados de

estabilidade apresentados pelo método ao longo deste capítulo.

Os experimentos acima demonstram que o T-Explainer apresenta bom desempenho, em geral, superando métodos *model-agnostic* bem conhecidos como o SHAP e, ainda que o LIME tenha surpreendido em alguns testes, estes se concentraram em configurações com dados de menor dimensionalidade. Além disso, o T-Explainer se mostra competitivo entre técnicas *model-specific* como as baseadas em gradientes, especialmente o *Integrated Gradients*, sendo uma nova e alternativa ferramenta para explicar predições de modelos caixa-preta (*black boxes*).

5.5 Desempenho Computacional

Durante os experimentos realizados neste capítulo, buscou-se manter as mesmas configurações para todos os testes, prática comum na literatura para comparação entre métodos XAI (UPADHYAY; JOSHI; LAKKARAJU, 2021; TAN *et al.*, 2023). Os modelos de aprendizado construídos aqui foram otimizados, mas não são os mais ajustados para extrair deles o melhor desempenho preditivo sobre todos os dados ao mesmo tempo, de modo individualizado. Claro que as explicações geradas pelos métodos XAI são tão boas quanto os modelos sobre os quais estes métodos são aplicados. Então, seria interessante também proceder testes com modelos especificamente parametrizados e treinados sobre cada um dos conjuntos de dados para verificar o comportamento de cada método. Entretanto, há a necessidade de se fazer um balanço entre a quantidade de tempo que seria necessário dedicar à construção de modelos individualizados e o resultado prático para a análise dos métodos XAI.

Optou-se por proceder a parametrização utilizando o *Grid Search*, selecionando as configurações de modelos que apresentaram as melhores acurácias dento das tarefas em mãos. Entretanto, seria necessário ampliar ainda mais o leque de opções estabelecido para os hiperparâmetros das Redes Neurais e classificadores XRFC na busca de modelos mais finamente ajustados para cada contexto de dados. Esse tipo de refinamento consome tempo e recursos. A construção de modelos preditivos altamente acurados e dedicados às nuances mais afinadas dos conjuntos de dados aos quais estes modelos se aplicam, embora seja o cenário "ideal" em problemas de Aprendizado de Máquina, não é a motivação deste trabalho. Então, optou-se por treinar modelos de aprendizado capazes de alcançar boas taxas de acurácia em termos gerais, respeitando, quando possível, configurações reportadas na literatura. A individualização e o afinamento dos modelos preditivos dedicados a cada uma das base de dados utilizadas nesta pesquisa, fica como trabalho futuro.

Os testes discutidos neste capítulo foram feitos a partir de vários conjuntos de dados sintéticos e reais. Cada base utilizada apresenta as suas características particulares, que

são distintas entre os diferentes conjuntos. Entretanto, utilizou-se o mesmo recorte de dados para cada bateria de testes realizada dentro do contexto de um conjunto, seja esta bateria apresentada ora em tabela ou figura. Esta configuração de equidade nos dados em cada grupo de testes busca oferecer tabelas e figuras contendo comparações justas entre os métodos XAI, uma vez que discutir resultados para a mesma métrica de estabilidade mas com métodos sob diferentes recortes de dados, poderia suscitar questões sobre a validade das comparações estabelecidas para estes resultados.

Os tempos de execução das explicações locais geradas pelo T-Explainer oscilaram entre 0.5 e 1.5 segundo, em média, com 7 a 13 ciclos do algoritmo de otimização sendo necessários para determinar o gradiente (o limite máximo de iterações foi fixado em 30). Esses valores representam um desempenho computacional bastante atrativo, quando comparado aos primeiros desenvolvimentos desta pesquisa que tinham métodos baseados no conceito de remoção e retreinamento de Hooker *et al.* (2018), que exigem consideravelmente mais recursos computacionais (fato pontuado pelos próprios autores), fazendo com que os algoritmos iniciais chegassem a executar em tempos superiores aos 60 segundos, aproximadamente, dentro das configurações de dados e modelos de aprendizado estabelecidas. As versões atuais do métodos XAI comparados com o T-Explainer apresentam tempos de execução comparáveis, gerando uma atribuição local de importâncias em uma fração de segundo.

Para comparar os desempenhos quanto ao tempo de execução entre SHAP e T-Explainer, selecionou-se o modelo 3H-NN que foi treinado sobre um conjunto de dados sintético com 16 atributos numéricos (16-FT), gerado com suporte da biblioteca OpenXAI. O SHAP é uma biblioteca XAI que oferece aos analistas um conjunto de métodos explicadores independentes e também específicos a certos modelos de aprendizado. Conforme definido anteriormente, na Seção 5.2, foi utilizado o SHAP *Explainer* para explicar predições feitas por Redes Neurais. O SHAP *Explainer* é a versão *model-agnostic* do SHAP, que utiliza diferentes algoritmos para gerar explicações, algoritmos estes, otimizados de acordo com o modelo e os dados sob explicação. Para dados de alta dimensão, o SHAP *Explainer* utiliza uma versão baseada em permutações, detalhada na Seção 3.2.7.

A história de sucesso relativamente longa do SHAP dentro do XAI, seja em aplicações acadêmicas ou comerciais, deriva, além de suas garantias teóricas, de seu amplo conjunto de ferramentas e do avançado nível de otimização no desenvolvimento de seus métodos, implementados por uma equipe de programadores dedicados ao projeto. Apesar dos bons resultados e dos esforços empreendidos, o T-Explainer é fruto de uma única pesquisa de doutorado. Logo, sua implementação ainda não alcançou o grau de maturação de ferramentas consagradas como o SHAP.

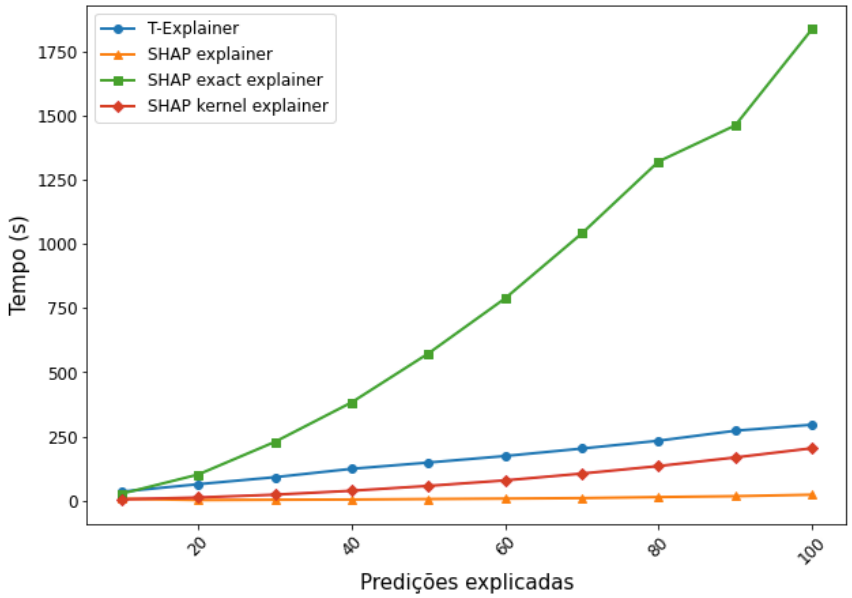
Por isso, para estabelecer comparações mais justas entre os tempos de execução, além do SHAP *Explainer*, compara-se o T-Explainer com o *KernelSHAP* (LUNDBERG; LEE,

2017), versão do método que aproxima uma regressão linear pra calcular *Shapley values* e o SHAP *Exact Explainer*, versão que aplica o algoritmo determinístico do explicador. Embora também seja uma versão com alto nível de otimização algorítmica, o SHAP *Exact Explainer* não faz uso de mecanismos de amostragem, gerando explicações com base no cálculo dos *Shapley values* aplicando diretamente a Equação 3.5.

Note que este experimento utiliza um conjunto de dados sintético de 16 atributos (16-FT), que foi gerado seguindo as mesmas especificações apresentadas na Seção 5.3.1, para o conjunto 20-FT gerado pela biblioteca OpenXAI. No entanto, houve um motivo para gerar um conjunto de menor dimensionalidade. Segundo as especificações, o SHAP *Exact Explainer* é capaz de trabalhar bem com dados contendo até 15 atributos. Além disso, o algoritmo se torna intratável. Inicialmente, planejou-se realizar a comparação de tempos de execução com o conjunto 20-FT, que é familiar ao leitor pois já havia sido introduzido, definido e experimentado. Entretanto, o SHAP *Exact Explainer* apresentou erro de execução ao operar o conjunto 20-FT. O número de 16 atributos foi o máximo que o SHAP *Exact Explainer* foi capaz de rodar sem quebras na execução. O modelo 3H-NN foi treinado sobre os dados 16-FT e obteve acurácia de 85%.

O gráfico da Figura 5.14 apresenta um comparativo entre os tempos de execução do T-Explainer e do SHAP, nas versões otimizada por amostragens, SHAP *Explainer*, e determinística, SHAP *Exact Explainer*. Foram realizados dez ciclos de execução variando, em cada ciclo, a quantidade de explicações que os métodos deveriam gerar, de acordo com aumentos gradativos no tamanho das amostras de instâncias selecionadas no conjunto 16-FT e preditas pelo modelo 3H-NN.

Figura 5.14 – Tempos de execução dos métodos SHAP e T-Explainer explicando as predições do modelo 3H-NN treinado sobre o conjunto 16-FT.



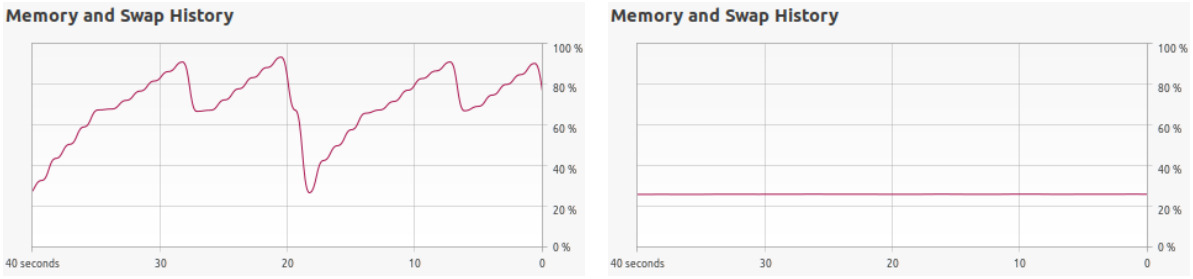
Fonte: Ortigossa *et al.* (2024).

Quanto aos tempos de execução, o T-Explainer teve um desempenho em linha com o *KernelSHAP*, ambos notadamente mais eficientes do que o SHAP *Exact Explainer*, com este método apresentando uma curva de tempo com trajetória exponencial, enquanto o T-Explainer mantém um aumento próximo ao linear no tempo, seguindo o incremento das amostras. Mas o SHAP *Explainer* foi o método de execução mais rápida. Entretanto, deve ser reforçado, mais uma vez, o uso de ferramentas de amostragem para obter tal desempenho, que seria impraticável de outro modo, dada a complexidade do método exato. Além disso, uma informação importante a ser registrada tem relação com o número de instâncias utilizadas. Observando o eixo das abscissas (horizontal) na Figura 5.14, tem-se que a quantidade máxima foi de 100 instâncias preditas e explicadas, um número relativamente baixo. Isso se deu porque não foi possível executar o SHAP *Exact Explainer* sem quebras sobre amostras contendo mais instâncias de dado, ou seja, além da limitação quanto ao número de atributos, existe também a limitação quanto ao tamanho da amostra que pode ser explicada pelo SHAP *Exact Explainer*.

Algo que chamou a atenção ao longo dos comparativos de tempo, foi a intensa demanda por memória do SHAP *Exact Explainer*. A Figura 5.15 ilustra o histórico de uso de memória RAM durante a execução do SHAP *Exact Explainer* (Figura 5.15a) e do T-Explainer (Figura 5.15b), durante as explicações das predições do modelo 3H-NN sobre os dados 16-FT realizadas para os comparativos de tempo da Figura 5.14. O Sistema Operacional e as demais aplicações ativas no momento da captura desses históricos consumiam por volta de 24.6% da memória RAM do computador. Note que o SHAP *Exact Explainer* apresenta uma curva com variações abruptas, com picos chegando a atingir quase o limite de memória RAM do computador. Por outro lado, o T-Explainer implicou em pouca variação no consumo de memória.

Figura 5.15 – Demanda por memória RAM na execução dos métodos SHAP e T-Explainer explicando as predições do modelo 3H-NN sobre os dados 16-FT.

(a) SHAP *Exact Explainer*. **(b)** T-Explainer.



Fonte: Elaborada pelo autor.

Conforme cresceram os conjuntos de amostras sob explicação, cresceram as demandas por memória do SHAP *Exact Explainer*. Uma aplicação que requer variações entre pouco

mais de 20% para até quase 100% da memória disponível, levando em consideração que o computador utilizado possuía 16GB de RAM, é impeditiva em muitos cenários, ainda mais se observada a limitação na quantidade de predições explicadas. Neste sentido, a aplicação do SHAP como método XAI em cenários com dados mais robustos em termos de dimensionalidade e tamanho, parece estar sujeita a um *tradeoff* semelhante ao discutido na Seção 2.4. Ou seja, aplicar a versão mais precisa do SHAP é apenas possível em cenários muito restritos, ao passo que a versão factível do método tende a apresentar resultados altamente instáveis, como os vistos neste capítulo. Esse tipo de *tradeoff* não é observado no T-Explainer que, embora também demande processamento intenso, é uma aplicação leve em termos de consumo de memória.

Os testes de tempo de execução foram realizados mantendo o computador e o Sistema Operacional dedicados em executar os métodos XAI, sem que houvesse qualquer outra ação durante este procedimento que pudesse demandar uso de memória ou processamento, algo que fatalmente interferiria nos resultados obtidos. As Figuras 5.14 e 5.15 refletem os desempenhos dos métodos XAI sob as mesmas condições de modelo e dados.

Apenas a título de curiosidade, renderizar um gráfico como o da Figura 5.10c aplicando o SHAP *Explainer* para explicar as predições da Rede Neural 5H-64-NN sobre todo o conjunto HIGGS (cf. Tabela 5.14), é uma tarefa que requer pouco mais de 23 horas de processamento ininterrupto. Ainda que fosse computacionalmente possível utilizar o SHAP *Exact Explainer* para explicar predições sob esta configuração de modelo e dados (o que não é), seguindo os desempenhos descritos na Figura 5.14, seria difícil estimar a quantidade de tempo necessário para gerar uma visão sobre todo um conjunto com as proporções do HIGGS a partir do SHAP *Exact Explainer*.

A execução das métricas quantitativas para avaliar métodos XAI consome tempo e recursos, pois são feitos repetidos procedimentos de amostragem e perturbação para cada instância de dado, com diversos outros cálculos sendo requeridos para computar cada uma das métricas. A decisão de integrar a RIS e a ROS foi acertada pois, além da questão técnica em se utilizar as mesmas perturbações para ambas, houve o ganho de tempo, que foi significativo. Ainda assim, os testes levam tempo para serem concluídos, o que é esperado. Uma opção quando se trabalha com conjuntos de dados com grande volume de instâncias, seria dividir os testes selecionando conjuntos de amostras (ou *batches*).

5.6 Considerações Finais

Não é surpresa que o T-Explainer tenha se comportado de modo mais próximo, em termos de estabilidade, aos métodos baseados em gradientes que foram selecionados para contrapor com seus resultados (*Integrated Gradients*, *Input* $\times$ *Gradient* e DeepLIFT).

Porém, diferente destes, o T-Explainer se mostrou mais flexível para atuar em contextos com classificadores não diferenciáveis. Conforme visto ao longo deste capítulo, nem sempre o T-Explainer apresentou o melhor desempenho entre os métodos XAI testados, algo que não necessariamente é considerado um demérito. Em muitos casos, ainda que não tenha se mostrado o mais estável dos explicadores, o T-Explainer ficou longe de obter resultados altamente instáveis, como os apresentados consistentemente pelo SHAP.

De modo geral, o T-Explainer apresentou desempenhos robustos em relação aos demais métodos XAI, principalmente atuando sobre o contexto de dados com atributos numéricos de maior dimensionalidade. Quando a tarefa foi explicar predições sobre dados contendo atributos categóricos, o *Integrated Gradients* se destacou, notadamente em termos dos valores máximos das métricas RIS e ROS, com o T-Explainer apresentando resultados encorajadores, figurando regularmente entre os métodos mais estáveis. Embora também deva ser destacado o bom desempenho do T-Explainer quanto à estabilidade ao gerar explicações sobre dados categóricos de maior dimensionalidade, como os conjuntos *German Credit* e HELOC.

Tratar adequadamente atributos categóricos não é um assunto trivial em Aprendizado de Máquina. A solução desenvolvida para habilitar o T-Explainer a manusear este tipo de dados envolveu uma estratégia de simulação de continuidade numérica, que foi aplicada nos valores binários resultantes da aplicação do *one-hot encoding* sobre os valores categóricos. Esta solução se mostrou eficiente, porém, abre um largo caminho para pesquisas e aprimoramentos, principalmente na escolha do raio de perturbação δ, que pode variar de acordo com as características do conjunto de dados e do modelo de aprendizado sob explicação. O valor do raio de perturbação δ pode impactar na acurácia do modelo ajustado aos dados categóricos, fazendo com que este discorde do modelo original. Nos experimentos propostos nesta pesquisa, a transformação de atributos categóricos em numéricos não causou impacto significativo na acurácia do modelo ajustado.

Destaca-se que a tarefa de escolha do valor δ pode ser feita pelo analista humano responsável pelos procedimentos de explicação, de acordo com o conhecimento de características dos dados. Mas esta escolha também poderia ser feita automaticamente, integrando no T-Explainer ferramentas que façam a extração de propriedades dos dados de modo a determinar um raio de perturbação adequado, sem a necessidade de intervenção humana.

Amparore, Perotti e Bajardi (2021) apontam que a inclusão de ferramentas de análise úteis também é um dos motivos de sucesso das principais abordagens XAI. Todo o desenvolvimento realizado nesta pesquisa foi estruturado sob o formato de *framework* em uma biblioteca *Python* que oferece aos desenvolvedores e usuários interessados pelo T-Explainer como abordagem XAI, uma gama de ferramentas analíticas para a tarefa de atribuição de importâncias em predições de modelos de aprendizado. Em outras

palavras, além do T-Explainer em si, o pacote computacional desenvolvido inclui um conjunto de métodos de pré-processamento e inspeção de resultados. Para a etapa de pré-processamento, foram integrados métodos para tratar dados faltantes (em variáveis numéricas ou categóricas), normalização, métodos para a obtenção de estatísticas dos dados, como distâncias e valor esperado de predição. Para a inspeção dos resultados, o T-Explainer foi integrado com as principais ferramentas de visualização de informações disponibilizadas pelo SHAP, além de oferecer um conjunto de métricas quantitativas e geração de dados sintéticos para *benchmark*.

É claro que cada teste empreendido aqui, utilizando modelos baseados em algoritmos de aprendizado e arquiteturas diferentes, treinados sobre conjuntos de dados diversos, representa um cenário em particular com configurações e espaços de características distintos. É tarefa do cientista de dados verificar as melhores alternativas de explicabilidade para o problema em mãos, algo que pode ser feito por meio de refinamentos nos hiperparâmetros, seja dos modelos de aprendizado subjacentes ou também dos métodos XAI utilizados para explicar as predições desses modelos, contando com as ferramentas de *benchmarking* integradas no *framework* do T-Explainer para tal.

Neste capítulo foram apresentados os resultados obtidos nesta pesquisa de doutorado. Estes resultados são motivadores e indicam que os objetivos elencados no início deste documento foram cumpridos. Cada passo de projeto e construção empreendido aqui, envolveu estudos sobre as principais metodologias existentes, visando compreender os detalhes de seus mecanismos de funcionamento para que, então, fosse possível tirar proveito das suas capacidades e encontrar soluções para contornar as suas deficiências. Claro que o tema da explicabilidade é vasto e não há uma solução definitiva que resolva todas as questões em aberto. Há muito espaço para a evolução no *Explainable Artificial Intelligence* em oferecer abordagens compreensivas e com uma gama de ferramentas para tratar e verificar as predições de modelos de aprendizado de máquina. Mas este trabalho dá um passo no sentido do progresso na área, com o T-Explainer apresentando resultados encorajadores, inclusive, superando alguns de seus pares.

Capítulo 6

Conclusão

6.1 Limitações e Trabalhos Futuros

Considerando a redação da GDPR no que se refere ao direito à explicação, caso uma aplicação suportada por modelos de decisão forneça aos seus usuários explicações inconsistentes para instâncias de dado similares (ou para a mesma instância), então essas explicações não podem ser consideradas confiáveis (AMPARORE; PEROTTI; BAJARDI, 2021). Neste sentido, a metodologia apresentada nesta pesquisa dá um passo à frente em relação à geração de explicações que tenham maior grau de confiabilidade. Ao não se apoiar em parametrizações complexas que requerem heurísticas, escolhas empíricas pouco fundamentadas ou aproximações probabilísticas, a abordagem desenvolvida apresenta comportamento determinístico na tarefa de atribuição de importâncias, demonstrando o valor desta pesquisa ao adicionar uma alternativa sólida no recente e produtivo universo XAI. Quando cuidadosamente desenvolvida e aplicada, a explicabilidade contribui ao adicionar uma perspectiva a mais no vasto horizonte do Aprendizado de Máquina, o que pode enriquecer o debate futuro sobre se as máquinas são realmente capazes de apresentar comportamentos inteligentes (LAPUSCHKIN *et al.*, 2019).

Os experimentos descritos aqui se concentraram em dois tipos de modelos classificadores binários, que são conhecidos por apresentarem bons resultados em tarefas de aprendizado supervisionado (BORISOV *et al.*, 2022). Porém, não há restrições em aplicar o T-Explainer em outros modelos mais complexos. Em particular, o T-Explainer naturalmente pode ser estendido para suportar regressão ou classificação multi-classe a partir de pequenas adaptações, que estão em desenvolvimento.

Em geral, os resultados das avaliações de estabilidade indicam que o T-Explainer é mais robusto quando aplicado em cenários com dados de alta dimensionalidade preditos utilizando Redes Neurais de maior complexidade. O T-Explainer apresentou bons resultados quando aplicado sobre modelos baseados em árvores, mas existem desafios que devem ser

considerados ao tratar deste tipo de modelo de aprendizado. Árvores possuem arquiteturas não contínuas em que valores constantes são armazenados dentro de cada nó folha, o que dificulta a atuação de métodos explicadores baseados em gradientes. Para se aplicar um método como o *Integrated Gradients* ou DeepLIFT na explicação de predições feitas por um modelo baseado em árvores, é necessário antes gerar um modelo intermediário (normalmente uma Rede Neural) que aproxime o comportamento do modelo original, e então gerar as explicações a partir deste modelo intermediário. Mas pode ser demasiado complexo obter uma aproximação fiel de um modelo de aprendizado.

Ainda que a atual versão do T-Explainer não esteja completamente operacional para explicar aplicações baseadas em árvores, os experimentos utilizando dados sintéticos se mostraram encorajadores. Como trabalho futuro, será desenvolvido um mecanismo para habilitar o T-Explainer para trabalhar com árvores de modo pleno, utilizando um interpolador de funções aplicado na árvore, com o T-Explainer rodando sobre o modelo interpolado.

O T-Explainer está habilitada para trabalhar com dados contendo atributos categóricos, por meio de perturbações contínuas que possibilitam o cálculo das derivadas parciais. Entretanto, a metodologia desenvolvida requer que seja feito o retreino do modelo de aprendizado, para que este seja ajustado aos intervalos contínuos induzidos nas colunas de zeros e uns resultantes da aplicação do *one-hot-encoding* sobre os atributos categóricos. O procedimento de retreino implica em complexidade computacional.

Diferente da solução apresentada por Hooker *et al.* (2018), que exige o retreino do modelo a cada iteração do método, a solução desenvolvida para o T-Explainer categórico requer um único retreino, sendo, por isso, mais eficiente. Porém, ainda há a necessidade de um retreino, o que pode representar custo extra considerável em cenários com grandes volumes de dados e modelos altamente complexos. Por isso, o refinamento da metodologia de tratamento de dados categóricos de modo confiável e mais eficiente computacionalmente é uma tarefa que está em planejamento teórico e prático. Entre as alternativas neste contexto, destacam-se as transformações baseadas em *target encoding* (BANACHEWICZ; MASSARON; GOLDBLOOM, 2022), que convertem cada elemento nominal de um atributo categórico em seu correspondente valor esperado.

Também está em projeto a expansão do T-Explainer para trabalhar com dados não tabulares. Os testes apresentados se concentraram em torno da explicação de modelos de aprendizado treinados sobre dados tabulares, pois este é o formato de dados mais utilizado em problemas de aprendizado (BORISOV *et al.*, 2022). Entretanto, o T-Explainer foi projetado considerando estruturas mais complexas de representação de dados, como imagens 2D, nuvens de pontos 3D, vídeos ou segmentações semânticas. Para processar estes dados, extensões no T-Explainer podem ser implementadas a partir de estratégias de

simplificação com máscaras, algo que não é novidade e tem sido amplamente utilizado em outros métodos XAI (RIBEIRO; SINGH; GUESTRIN, 2016a; LUNDBERG; LEE, 2017).

Durante o processo de testes, foi observado que o cálculo de diferenças finitas apresenta certa instabilidade para explicar instâncias próximas aos limites de decisão do modelo (*decision boundaries*), devido às descontinuidades geradas pelos limites de decisão. Qualquer método de diferenciação numérica introduz erros que podem ser difíceis de controlar, especialmente em situações críticas, como a explicação de predições de instâncias de borda. Para resolver esta preocupação, estamos desenvolvendo um método mais robusto e personalizado para lidar com a instabilidade da diferenciação e aumentar a precisão das explicações. A literatura sobre métodos numéricos contém uma série de alternativas para lidar com descontinuidades na aproximação de derivadas utilizando diferenças finitas (TOWERS, 2009; SCHEINBERG, 2022). Atualmente, algumas dessas alternativas estão sob investigação para serem implementadas e introduzir mais robustez ao T-Explainer.

Outro aspecto em aprimoramento é o módulo de otimização de $\mathbf{h}$. A quantificação de uma estimativa razoável de deslocamento não tem solução fechada na literatura e muitas vezes depende de cálculos iterativos que aproximam o gradiente. Para grandes conjuntos de dados, estas abordagens iterativas podem impor limitações computacionais. É importante considerar que um valor ótimo de $\mathbf{h}$ pode variar dependendo da função preditiva f e da precisão de aproximação desejada. Como parte central do T-Explainer, o processo de otimização $\mathbf{h}$ está em refinamento visando aprimorar a precisão enquanto se mantém a eficiência computacional. A literatura sobre métodos numéricos traz uma série de alternativas para lidar com descontinuidades na aproximação de derivadas por meio de diferenças finitas (TOWERS, 2009; SCHEINBERG, 2022). Entre as alternativas em estudo, estão os métodos que desenvolvem diferenças finitas generalizadas (GFD – *Generalized Finite Differences*) (BENITO *et al.*, 2003; SONG *et al.*, 2020). Também relacionado ao processamento, otimizações computacionais, como processamento paralelo e uso de GPU, podem ser exploradas para tornar o T-Explainer mais eficiente.

Optou-se por dar maior atenção sobre o SHAP *Explainer* (e também TreeSHAP) nos testes comparativos. Esta escolha foi motivada pelo inegável sucesso que a biblioteca SHAP tem obtido dentro da explicabilidade, atualmente. O SHAP *Explainer* é a versão otimizada do método baseado em *Shapley values* que é a mais flexível em termos de modelos de aprendizado. Outro motivo para esta escolha está na disponibilidade de ferramentas gráficas de alta qualidade pela biblioteca SHAP, como os *Summary plots* apresentados no capítulo anterior. Foram desenvolvidas interfaces para que o T-Explainer pudesse aplicar muitas das ferramentas gráficas oferecidas pelo SHAP. Embora fosse interessante desenvolver as mesmas interfaces dedicadas aos demais métodos XAI utilizados nos experimentos, de modo a direcionar as comparações visuais entre o T-Explainer e os métodos mais

estáveis em cada experimento (em vez de restringir as visualizações ao SHAP), esses desenvolvimentos fogem ao escopo desta pesquisa, mas ficam como trabalho futuro.

Outras versões do SHAP serão exploradas futuramente, de modo a ampliar a gama de comparações com o T-Explainer. Entre estas versões, estão o SHAP *Exact Explainer* (quando possível), *KernelSHAP*, DeepSHAP e o SHAP *GradientExplainer*, que explica predições com base em gradientes (apesar desta última versão ser, na verdade, uma extensão do *Integrated Gradients*, utilizado nesta pesquisa). Entretanto, também fica aqui registrada a necessidade de ir além do SHAP em problemas de explicabilidade. Apesar de ser uma abordagem que oferece uma biblioteca XAI rica em ferramentas, o SHAP não é a única solução viável, além de apresentar uma série de limitações (cf. Seção 3.2.8). As avaliações empreendidas aqui evidenciaram a sua instabilidade. Existem diversas alternativas viáveis em XAI, com o T-Explainer se posicionando como uma delas.

A falta de procedimentos de avaliação em XAI foi destacada por Adadi e Berrada (2018). Uma avaliação abrangente de uma ferramenta de explicabilidade requer testes extensivos usando métricas quantitativas (BODRIA *et al.*, 2021) e qualitativas (WEERTS; IPENBURG; PECHENIZKIY, 2019), além de incorporar o conhecimento de analistas humanos. Neste trabalho, foram priorizadas as métricas quantitativas, especialmente aquelas relativas à avaliação da estabilidade, mas há de se reconhecer o valor do processo de avaliação qualitativa. Logo, há o compromisso futuro em validar de modo abrangente o T-Explainer usando métricas além das discutidas aqui.

Lundberg *et al.* (2018) observaram que as variáveis importantes não implicam um relacionamento causal e então não representam um cenário completo para explicação de modelos de aprendizado. Em outras palavras, gerar explicações apenas sobre aspectos ou ferramentas únicas dentro do contexto dos sofisticados e multifacetados sistemas de aprendizado atuais, é uma parte da contribuição com a interpretabilidade. Conhecer as variáveis que influenciam em uma decisão é um tipo de informação que pode ser utilizada pelos analistas como meio para formular e melhor fundamentar explicações sobre as verdadeiras razões ou viés que podem estar guiando as predições dos modelos, o que é difícil de inferir sem o apoio de um método XAI.

Neste sentido, um ferramental XAI abrangente, capaz de produzir informações consistentes e estáveis sobre as diferentes características do processo de aprendizado, pode sim cobrir um cenário de explicabilidade mais completo. Logo, a abordagem desenvolvida nesta pesquisa, que tem como objetivo desenvolver um método para atribuir o *feature importance* para variáveis preditivas, insere-se em um contexto mais ambicioso de integrar um *framework* XAI completo, contendo ferramentas analíticas voltadas à explicabilidade, desde métodos específicos para pré-processamento e formatação de dados, até a geração, testes e visualização de informações. Como um projeto em evolução, o T-Explainer conti-

nuará a expandir e se atualizar, incorporando funcionalidades e refinamentos adicionais. Este desenvolvimento contínuo fará com que o T-Explainer avance em suas capacidades de "abrir" e interpretar os intrincados e complexos modelos de aprendizado caixa-preta.

6.2 Considerações Finais

A Inteligência Artificial não é o futuro, é o presente. Redes Neurais, *Machine Learning*, *Deep Learning* e outros temas relacionados ao universo da Inteligência Artificial têm transitado de um conceito futurista para uma realidade cada vez mais recorrente na vida cotidiana. Hoje, a Inteligência Artificial não está mais restrita à pesquisa acadêmica, filmes de ficção científica, ou recomendações em lojas *online*, com os rendimentos relacionados ao mercado da Inteligência Artificial superando, em 2021, o Produto Interno Bruto (PIB) de grandes economias mundiais como Canadá, Itália e Brasil (ADADI; BERRADA, 2018; United Nations, 2021). Os novos modelos de aprendizado são cada vez mais capazes, precisos e complexos, alcançando desempenhos que eram inimagináveis até poucos anos atrás (LECUN; BENGIO; HINTON, 2015; DOŠILOVIĆ; BRČIĆ; HLUPIĆ, 2018; VASWANI *et al.*, 2017; WIEGREFFE; PINTER, 2019).

Cresce a quantidade de decisões tomadas com o apoio de sistemas inteligentes, com novas aplicações surgindo a todo momento. Muitos domínios em que os algoritmos de aprendizado se inserem têm o potencial de impactar o modo como a sociedade interage, desafiando o desenvolvimento de novas ferramentas destinadas a mitigar os possíveis impactos negativos causados. Embora a comunidade científica dedicada ao Aprendizado de Máquina tenha se concentrado exitosamente em aprimorar o desempenho preditivo dos seus modelos, existe a necessidade de ajustar o equilíbrio entre precisão e transparência. Alta precisão significa altas taxas de valores classificados positivamente e, por consequência, baixos índices de decisões falsas. Entretanto, não é aceitável ignorar o entendimento do processo racional segundo o qual as decisões foram geradas (LUNDBERG *et al.*, 2018).

Vale destacar que explicar predições é algo de particular interesse, pois os padrões que o modelo descobre podem ser ainda mais importantes do que o desempenho preditivo em si (LUNDBERG *et al.*, 2020). A verificação de padrões não é capturada pelas métricas de validação comumente utilizadas no Aprendizado de Máquina, o que coloca uma questão sensível sobre o amplo, e por vezes não refletido, atual uso de modelos não-lineares complexos para a tomada de decisões em muitos domínios, desde a ciência até a indústria (LAPUSCHKIN *et al.*, 2019).

A necessidade de ir além da mera crença em métricas de *performance* é benéfica para todos os envolvidos (e afetados) por aplicações que se baseiam em tomadas de decisão que são suportadas por Inteligência Artificial. O Aprendizado de Máquina é visto como

algo revolucionário e que pode ser aplicado como parte importante das mudanças de paradigma que levarão a sociedade às evoluções futuras. Entretanto, nem tudo o que se refere aos avanços tecnológicos é perfeito, com pesquisas citadas ao longo deste texto abordando as muitas fragilidades dos algoritmos de aprendizado que, inclusive, vão além dos questionamentos gerados pela sua dificuldade de compreensão. Portanto, novos algoritmos e novas metodologias de modelagem mais robustas e em conformidade com as exigências éticas e legais podem emergir com o apoio das técnicas XAI.

Usualmente, os modelos de aprendizado são treinados e validados com dados passados, que refletem o contexto presente em que estes dados foram coletados, algo que por si só já representa algum grau de viés. A realidade parece muito mais óbvia em retrospectiva do que quando se infere o futuro. Aqueles que experimentam o viés retrospectivo podem aplicar erroneamente a retrospectiva atual à previsão passada (SHEFRIN, 2002). Os usuários que optam por acreditar em modelos caixa-preta, sem um questionamento sobre o processo racional por trás, podem acabar, em última análise, sendo iludidos pelo acaso e pelo viés do resultado (TALEB, 2005).

Nesta pesquisa, foi apresentado o T-Explainer, uma técnica XAI baseada em expansão em série de Taylor. T-Explainer é um método determinístico independente de modelo (*model-agnostic*) para gerar explicações locais sobre predições por meio da atribuição de importância, construído sobre uma sólida fundamentação matemática que lhe garante a observância de relevantes propriedades, como precisão local e consistência. O T-Explainer é capaz de fornecer explicações mesmo para aqueles modelos de aprendizado que tradicionalmente não podem ser explicados diretamente utilizando abordagens baseadas em gradientes. Os resultados experimentais aqui descritos, demonstram que as explicações do T-Explainer são estáveis e competitivas com os principais métodos de explicabilidade, sejam elas *model-agnostic* ou especificamente projetadas, em diversos contextos. Além de um novo método, os desenvolvimentos que integram o T-Explainer atuam como um *framework* XAI contendo ferramentas de *benchmarking* aplicáveis a diferentes modelos e domínios de dados, auxiliando os usuários do Aprendizado de Máquina a escolherem os métodos XAI mais adequados, a depender das suas necessidades contextuais.

Referências Bibliográficas

AAS, K.; JULLUM, M.; LØLAND, A. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, Elsevier, v. 298, p. 103502, 2021. Citado nas páginas 68, 89, 94, 95, 97, 98, 99, 100, 102, 105, 107 e 139.

ADADI, A.; BERRADA, M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, v. 6, p. 52138–52160, 2018. Citado nas páginas 28, 29, 31, 37, 56, 57, 60, 66, 68, 69, 76, 81, 90, 103, 182 e 183.

AGARWAL, C.; JOHNSON, N.; PAWELCZYK, M.; KRISHNA, S.; SAXENA, E.; ZITNIK, M.; LAKKARAJU, H. Rethinking stability for attribution-based explanations. *Preprint arXiv:2203.06877*, 2022. Citado na página 128.

AGARWAL, C.; SAXENA, E.; KRISHNA, S.; PAWELCZYK, M.; JOHNSON, N.; PURI, I.; ZITNIK, M.; LAKKARAJU, H. OpenXAI: Towards a transparent evaluation of model explanations. *Preprint arXiv:2206.11104*, 2022. Citado nas páginas 102, 104, 128, 129, 136 e 138.

ALEXANDRINA, E. C.; ORTIGOSSA, E. S.; LUI, E. S.; GONÇALVES, J. A. S.; CORREA, N. A.; NONATO, L. G.; AGUIAR, M. L. Analysis and visualization of multidimensional time series: Particulate matter (PM10) from São Carlos-SP (Brazil). *Atmospheric Pollution Research*, Elsevier, v. 10, n. 4, p. 1299–1311, 2019. Citado na página 81.

ALVAREZ-MELIS, D.; JAAKKOLA, T. S. On the robustness of interpretability methods. *Preprint arXiv:1806.08049*, 2018. Citado nas páginas 104 e 107.

_____. Towards robust interpretability with self-explaining neural networks. *Preprint arXiv:1806.07538*, 2018. Citado nas páginas 104 e 107.

AMANN, J.; VETTER, D.; BLOMBERG, S. N.; CHRISTENSEN, H. C.; COFFEE, M.; GERKE, S.; GILBERT, T. K.; HAGENDORFF, T.; HOLM, S.; LIVNE, M. *et al.* To explain or not to explain? – Artificial intelligence explainability in clinical decision support systems. *PLOS Digital Health*, Public Library of Science San Francisco, CA USA, v. 1, n. 2, p. e0000016, 2022. Citado nas páginas 56, 57, 59, 63, 69, 76, 79 e 104.

AMPARORE, E.; PEROTTI, A.; BAJARDI, P. To trust or not to trust an explanation: Using LEAF to evaluate local linear XAI methods. *PeerJ Computer Science*, PeerJ Inc., v. 7, p. e479, 2021. Citado nas páginas 30, 31, 62, 70, 89, 90, 95, 99, 103, 106, 107, 110, 111, 112, 115, 128, 130, 131, 133, 136, 137, 141, 144, 161, 177 e 179.

ARORA, S.; BARAK, B. *Computational complexity: A modern approach*. Cambridge, UK: Cambridge University Press, 2009. Citado na página 51.

ARRIETA, A. B.; DÍAZ-RODRÍGUEZ, N.; SER, J. D.; BENNETOT, A.; TABIK, S.; BARBADO, A.; GARCÍA, S.; GIL-LÓPEZ, S.; MOLINA, D.; BENJAMINS, R. *et al.* Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, Elsevier, v. 58, p. 82–115, 2020. Citado nas páginas 27, 28, 31, 41, 46, 47, 48, 56, 57, 58, 63, 64, 65, 66, 70, 71, 73, 76, 80, 87 e 109.

ASIMOV. *The Neural Network Zoo – The Asimov Institute*. 2016. <https://www.asimovinstitute.org/neural-network-zoo/>. Acesso em: 02-01-2022. Citado na página 47.

BACH, S.; BINDER, A.; MONTAVON, G.; KLAUSCHEN, F.; MÜLLER, K.-R.; SAMEK, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS one*, Public Library of Science San Francisco, CA USA, v. 10, n. 7, p. e0130140, 2015. Citado nas páginas 86 e 107.

BALDI, P.; SADOWSKI, P.; WHITESON, D. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, Nature Publishing Group UK London, v. 5, n. 1, p. 4308, 2014. Citado nas páginas 134, 155 e 160.

BANACHEWICZ, K.; MASSARON, L.; GOLDBLOOM, A. *The Kaggle Book: Data analysis and machine learning for competitive data science*. Birmingham, UK: Packt Publishing Ltd, 2022. Citado na página 180.

BARBER, R. F.; CANDÈS, E. J. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 43, n. 5, p. 2055–2085, 2015. Citado na página 105.

BAROCAS, S.; SELBST, A. D.; RAGHAVAN, M. The hidden assumptions behind counterfactual explanations and principal reasons. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2020. (FAccT '20), p. 80–89. Citado nas páginas 85 e 107.

BARTLETT, P. L.; LONG, P. M.; LUGOSI, G.; TSIGLER, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 117, n. 48, p. 30063–30070, 2020. Citado na página 54.

BELLE, V.; PAPANTONIS, I. Principles and practice of explainable machine learning. *Frontiers in Big Data*, Frontiers, p. 39, 2021. Citado na página 76.

BENITO, J.; URENA, F.; GAVETE, L.; ALVAREZ, R. An h-adaptive method in the generalized finite differences. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 192, n. 5-6, p. 735–759, 2003. Citado na página 181.

BHATT, U.; XIANG, A.; SHARMA, S.; WELLER, A.; TALY, A.; JIA, Y.; GHOSH, J.; PURI, R.; MOURA, J. M. F.; ECKERSLEY, P. Explainable machine learning in deployment. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2020. p. 648–657. Citado nas páginas 63, 78, 81, 85, 101, 105 e 106.

BHAVSAR, P.; SAFRO, I.; BOUAYNAYA, N.; POLIKAR, R.; DERA, D. Chapter 12 - machine learning in transportation data analytics. In: *Data Analytics for Intelligent Transportation Systems*. Amsterdam, The Netherlands: Elsevier, 2017. p. 283–307. Citado nas páginas 38, 39, 40, 42 e 48.

BODRIA, F.; GIANNOTTI, F.; GUIDOTTI, R.; NARETTO, F.; PEDRESCHI, D.; RINZIVILLO, S. Benchmarking and survey of explanation methods for black box models. *Preprint arXiv:2102.13076*, 2021. Citado nas páginas 102 e 182.

BORISOV, V.; LEEMANN, T.; SESSLER, K.; HAUG, J.; PAWELCZYK, M.; KASNECI, G. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 7, p. 1–41, 2022. Citado nas páginas 134, 160, 169, 179 e 180.

BREIMAN, L. Bagging predictors. *Machine Learning*, Springer, v. 24, n. 2, p. 123–140, 1996. Citado na página 49.

_____. Random forests. *Machine Learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado nas páginas 49, 77 e 134.

BRISCOE, E.; FELDMAN, J. Conceptual complexity and the bias/variance tradeoff. *Cognition*, Elsevier, v. 118, n. 1, p. 2–16, 2011. Citado nas páginas 51, 52 e 53.

BUNT, A.; LOUNT, M.; LAUZON, C. Are explanations always important? A study of deployed, low-cost intelligent interactive systems. In: *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*. New York, NY, USA: Association for Computing Machinery, 2012. (IUI '12), p. 169–178. Citado na página 105.

BURKART, N.; HUBER, M. F. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, v. 70, p. 245–317, 2021. Citado na página 76.

BURRELL, J. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, Sage Publications Sage UK: London, England, v. 3, n. 1, p. 1–12, 2016. Citado na página 28.

CALIFORNIA. *California Consumer Privacy Act (CCPA)*. Department of Justice, Office of the Attorney General: State of California, 2021. <https://oag.ca.gov/privacy/ccpa>. Acesso em: 12-07-2023. Citado nas páginas 30 e 73.

CANTAREIRA, G. D.; ETEMAD, E.; PAULOVICH, F. V. Exploring neural network hidden layer activity using vector fields. *Information*, MDPI, v. 11, n. 9, p. 426, 2020. Citado nas páginas 46, 47, 82 e 107.

CARBONELL, J. G.; MICHALSKI, R. S.; MITCHELL, T. M. An overview of machine learning. *Machine Learning*, Elsevier, p. 3–23, 1983. Citado na página 38.

CARUANA, R.; LOU, Y.; GEHRKE, J.; KOCH, P.; STURM, M.; ELHADAD, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Virtual Conference: NY ACM, 2015. p. 1721–1730. Citado nas páginas 28, 65, 79 e 107.

CASALICCHIO, G.; MOLNAR, C.; BISCHL, B. Visualizing the feature importance for black box models. In: SPRINGER. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018*. Cham: Springer International Publishing, 2019. p. 655–670. Citado nas páginas 78, 81, 99 e 105.

CHAKRABORTY, S.; TOMSETT, R.; RAGHAVENDRA, R.; HARBORNE, D.; ALZANTOT, M.; CERUTTI, F.; SRIVASTAVA, M.; PREECE, A.; JULIER, S.; RAO, R. M. *et al.* Interpretability of deep learning models: A survey of results. In: IEEE. *2017 IEEE smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI*. San Francisco, CA, USA, 2017. p. 1–6. Citado nas páginas 28, 29 e 76.

CHAN, G. Y.-Y.; BERTINI, E.; NONATO, L. G.; BARR, B.; SILVA, C. T. Melody: Generating and visualizing machine learning model summary to understand data and classifiers together. *Preprint arXiv:2007.10614*, 2020. Citado nas páginas 68, 81 e 107.

CHAN, G. Y.-Y.; YUAN, J.; OVERTON, K.; BARR, B.; REES, K.; NONATO, L. G.; BERTINI, E.; SILVA, C. T. SUBPLEX: Towards a better understanding of black box model explanations at the subpopulation level. *Preprint arXiv:2007.10609*, 2020. Citado nas páginas 68, 83 e 107.

CHEN, H.; LUNDBERG, S.; LEE, S.-I. Explaining models by propagating Shapley values of local components. In: *Explainable AI in Healthcare and Medicine*. New York City, NY, USA: Springer, 2021. p. 261–270. Citado nas páginas 66, 77, 98 e 107.

CHEN, H.; LUNDBERG, S. M.; LEE, S.-I. Explaining a series of models by propagating Shapley values. *Nature Communications*, Nature Publishing Group, v. 13, n. 1, p. 1–15, 2022. Citado nas páginas 66, 67, 98, 102, 103 e 107.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. Citado nas páginas 49, 72 e 133.

COVER, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, IEEE, n. 3, p. 326–334, 1965. Citado na página 41.

CSÁJI, B. C. *et al.* Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, Citeseer, v. 24, n. 48, p. 7, 2001. Citado na página 46.

CUKIERSKI, W. *Titanic - Machine Learning from Disaster*. Kaggle, 2012. Acesso em: 06-10-2022. Disponível em: <https://kaggle.com/competitions/titanic>. Citado na página 155.

DAS, S.; JAVID, A. M.; GOHAIN, P. B.; ELDAR, Y. C.; CHATTERJEE, S. Neural greedy pursuit for feature selection. *Preprint arXiv:2207.09390*, 2022. Citado nas páginas 78, 103 e 107.

DORAN, D.; SCHULZ, S.; BESOLD, T. R. What does explainable AI really mean? A new conceptualization of perspectives. *Preprint arXiv:1710.00794*, 2017. Citado na página 31.

DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. *Preprint arXiv:1702.08608*, 2017. Citado nas páginas 29, 48, 57, 61, 73 e 76.

DOŠILOVIĆ, F. K.; BRČIĆ, M.; HLUPIĆ, N. Explainable artificial intelligence: A survey. In: IEEE. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija, Croatia, 2018. p. 0210–0215. Citado nas páginas 28, 39, 41, 48, 50, 57, 61, 63, 76 e 183.

DOUMPOS, M.; ZOPOUNIDIS, C. Model combination for credit risk assessment: A stacked generalization approach. *Annals of Operations Research*, Springer, v. 151, n. 1, p. 289–306, 2007. Citado na página 66.

DUA, D.; GRAFF, C. *Breast Cancer Wisconsin (Diagnostic)*. 2017. UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Sciences. Citado na página 155.

ENGLHARDT, A.; TRITTENBACH, H.; KOTTKE, D.; SICK, B.; BÖHM, K. Efficient SVDD sampling with approximation guarantees for the decision boundary. *Preprint arXiv:2009.13853*, 2020. Citado nas páginas 84 e 107.

EU Regulation. *2016/679 of the European Parliament and of the Council of 27 April 2016 on the General Data Protection Regulation*. Council of the European Union: European Parliament, 2016. <http://data.europa.eu/eli/reg/2016/679/oj>. Acesso em: 10-05-2023. Citado nas páginas 30 e 62.

FABER, L.; MOGHADDAM, A. K.; WATTENHOFER, R. When comparing to ground truth is wrong: On evaluating GNN explanation methods. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Virtual Event, Singapore: Association for Computing Machinery, 2021. (KDD '21), p. 332–341. Citado nas páginas 102 e 107.

FICO. *Home Equity Line of Credit (HELOC) Dataset*. 2019. <https://community.fico.com/s/explainable-machine-learning-challenge>. Acesso em: 10-09-2023. Citado na página 155.

FISHER, R. A. *Iris*. 1988. UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Sciences. DOI: https://doi.org/10.24432/C56C76. Citado na página 88.

FITZPATRICK, P. *Advanced calculus*. Providence, Rhode Island, EUA: American Mathematical Society, 2009. v. 5. Citado na página 120.

FOOTE, A.; NANDA, N.; KRAN, E.; KONSTAS, I.; COHEN, S.; BAREZ, F. Neuron to graph: Interpreting language model neurons at scale. *Preprint arXiv:2305.19911*, 2023. Citado na página 82.

FORTMANN-ROE, S. *Accurately measuring model prediction error*. 2012. <http://scott.fortmann-roe.com/docs/MeasuringError.html>. Acesso em: 15-06-2023. Citado nas páginas 51 e 59.

_____. *Understanding the Bias-Variance Tradeoff*. 2012. <http://scott.fortmann-roe.com/docs/BiasVariance.html>. Acesso em: 20-04-2023. Citado na página 52.

FRIEDMAN, E.; MOULIN, H. Three methods to share joint costs or surplus. *Journal of Economic Theory*, Elsevier, v. 87, n. 2, p. 275–312, 1999. Citado nas páginas 90 e 92.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Atatistics*, JSTOR, p. 1189–1232, 2001. Citado nas páginas 55, 81 e 107.

FRIEDMAN, J. H.; POPESCU, B. E. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, Institute of Mathematical Statistics, v. 2, n. 3, p. 916–954, 2008. Citado nas páginas 55 e 79.

GARDE, A.; KRAN, E.; BAREZ, F. DeepDecipher: Accessing and investigating neuron activation in large language models. *Preprint arXiv:2310.01870*, 2023. Citado nas páginas 82 e 107.

GARETH, J.; DANIELA, W.; TREVOR, H.; ROBERT, T. *An introduction to statistical learning: With applications in R*. Heidelberg, Germany: Spinger, 2017. v. 1. Citado na página 59.

GARNELO, M.; SHANAHAN, M. Reconciling deep learning with symbolic artificial intelligence: Representing objects and relations. *Current Opinion in Behavioral Sciences*, v. 29, p. 17–23, 2019. ISSN 2352-1546. Artificial Intelligence. Citado nas páginas 37 e 38.

GEIFMAN, A.; GALUN, M.; JACOBS, D.; RONEN, B. On the spectral bias of convolutional neural tangent and Gaussian process kernels. *Advances in Neural Information Processing Systems*, v. 35, p. 11253–11265, 2022. Citado na página 46.

GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias/variance dilemma. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 4, n. 1, p. 1–58, 1992. Citado nas páginas 52, 53 e 54.

GHOJOGH, B.; CROWLEY, M. The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial. *Preprint arXiv:1905.12787*, 2019. Citado nas páginas 49, 50, 52 e 54.

GILPIN, L. H.; BAU, D.; YUAN, B. Z.; BAJWA, A.; SPECTER, M.; KAGAL, L. Explaining explanations: An overview of interpretability of machine learning. In: IEEE. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. Turin, Italy, 2018. p. 80–89. Citado na página 30.

GLEICHER, M. A framework for considering comprehensibility in modeling. *Big Data*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 4, n. 2, p. 75–88, 2016. Citado na página 58.

GOLDREICH, O. Computational complexity: A conceptual perspective. *ACM Sigact News*, ACM New York, NY, USA, v. 39, n. 3, p. 35–39, 2008. Citado na página 51.

GOLDSTEIN, A.; KAPELNER, A.; BLEICH, J.; PITKIN, E. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, Taylor & Francis, v. 24, n. 1, p. 44–65, 2015. Citado nas páginas 55, 81 e 107.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado nas páginas 45 e 52.

GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. Explaining and harnessing adversarial examples. *Preprint arXiv:1412.6572*, 2014. Citado nas páginas 28, 46 e 61.

GUAN, S.; LOEW, M. Analysis of generalizability of deep neural networks based on the complexity of decision boundary. In: IEEE. *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Miami, FL, USA, 2020. p. 101–106. Citado nas páginas 84 e 107.

GUIDOTTI, R.; MONREALE, A.; RUGGIERI, S.; PEDRESCHI, D.; TURINI, F.; GIANNOTTI, F. Local rule-based explanations of black box decision systems. *Preprint arXiv:1805.10820*, 2018. Citado nas páginas 79 e 107.

GUIDOTTI, R.; MONREALE, A.; RUGGIERI, S.; TURINI, F.; GIANNOTTI, F.; PEDRESCHI, D. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 51, n. 5, p. 1–42, 2018. Citado nas páginas 61, 63, 65, 68, 76, 79 e 106.

GUNNING, D.; AHA, D. DARPA's explainable artificial intelligence (XAI) program. *AI Magazine*, v. 40, n. 2, p. 44–58, 2019. Citado nas páginas 29, 63 e 64.

HAIM, N.; VARDI, G.; YEHUDAI, G.; SHAMIR, O.; IRANI, M. Reconstructing training data from trained neural networks. *Preprint arXiv:2206.07758*, 2022. Citado na página 53.

HAMEED, I.; SHARPE, S.; BARCKLOW, D.; AU-YEUNG, J.; VERMA, S.; HUANG, J.; BARR, B.; BRUSS, C. B. BASED-XAI: Breaking ablation studies down for explainable artificial intelligence. *Preprint arXiv:2207.05566*, 2022. Citado nas páginas 102 e 107.

HAMILTON, M.; LUNDBERG, S.; ZHANG, L.; FU, S.; FREEMAN, W. T. Model-agnostic explainability for visual search. *Preprint arXiv:2103.00370*, 2021. Citado nas páginas 86, 89, 90, 99, 107 e 111.

HARNAD, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, Elsevier, v. 42, n. 1-3, p. 335–346, 1990. Citado na página 38.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H. *The elements of statistical learning: Data mining, inference, and prediction*. Heidelberg, Germany: Springer, 2009. v. 2. Citado nas páginas 44 e 52.

HASTIE, T. J.; TIBSHIRANI, R. J. *Generalized additive models*. Oxfordshire, England, UK: Routledge, 2017. Citado nas páginas 79 e 107.

HAUG, J.; ZÜRN, S.; EL-JIZ, P.; KASNECI, G. On baselines for local feature attributions. *Preprint arXiv:2101.00905*, 2021. Citado na página 103.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. New Jersey, USA: Prentice Hall, 1999. Citado na página 116.

HEALEY, S. P.; COHEN, W. B.; YANG, Z.; BREWER, C. K.; BROOKS, E. B.; GORELICK, N.; HERNANDEZ, A. J.; HUANG, C.; HUGHES, M. J.; KENNEDY, R. E. *et al.* Mapping forest change using stacked generalization: An ensemble approach. *Remote Sensing of Environment*, Elsevier, v. 204, p. 717–728, 2018. Citado na página 66.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. *Preprint arXiv:1503.02531*, 2015. Citado na página 79.

HO, T. K. Random decision forests. In: IEEE. *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Montreal, QC, Canada, 1995. v. 1, p. 278–282. Citado na página 48.

HOFMANN, H. *Statlog (German Credit Data)*. 1994. UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Sciences. Citado na página 155.

HOHMAN, F.; KAHNG, M.; PIENTA, R.; CHAU, D. H. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, IEEE, v. 25, n. 8, p. 2674–2693, 2018. Citado nas páginas 82, 83 e 107.

HONG, C. W.; LEE, C.; LEE, K.; KO, M.-S.; HUR, K. Explainable artificial intelligence for the remaining useful life prognosis of the turbofan engines. In: IEEE. *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*. Kaohsiung, Taiwan, 2020. p. 144–147. Citado nas páginas 99 e 107.

HOOKER, G.; MENTCH, L. Please stop permuting features: An explanation and alternatives. *Preprint arXiv:1905.03151*, 2019. Citado nas páginas 100, 107 e 144.

HOOKER, G.; MENTCH, L.; ZHOU, S. Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, Springer, v. 31, n. 6, p. 1–16, 2021. Citado nas páginas 100, 103, 105, 107, 124 e 144.

HOOKER, S.; ERHAN, D.; KINDERMANS, P.-J.; KIM, B. A benchmark for interpretability methods in deep neural networks. *Preprint arXiv:1806.10758*, 2018. Citado nas páginas 99, 102, 104, 107, 173 e 180.

HUANG, Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, Springer, v. 2, n. 3, p. 283–304, 1998. Citado na página 105.

INTEL. *Intel's Pohoiki Beach, a 64-Chip Neuromorphic System*. 2020. <https://intel.ly/30dAps3>. Acesso em: 20-04-2023. Citado na página 47.

JACOVI, A.; SWAYAMDIPTA, S.; RAVFOGEL, S.; ELAZAR, Y.; CHOI, Y.; GOLDBERG, Y. Contrastive explanations for model interpretability. *Preprint arXiv:2103.01378*, 2021. Citado nas páginas 84 e 107.

JAIN, S.; WALLACE, B. C. Attention is not explanation. *Preprint arXiv:1902.10186*, 2019. Citado na página 82.

JANZING, D.; MINORICS, L.; BLÖBAUM, P. Feature relevance quantification in explainable AI: A causality problem. *Preprint arXiv:1910.13413*, 2019. Citado na página 100.

JEYASOTHY, A.; LAUGEL, T.; LESOT, M.-J.; MARSALA, C.; DETYNIECKI, M. Integrating prior knowledge in post-hoc explanations. *Preprint arXiv:2204.11634*, 2022. Citado na página 103.

KAHNEMAN, D.; SIBONY, O.; SUNSTEIN, C. R. *Noise: A flaw in human judgment*. New York City, NY, USA: Little, Brown and Company, 2021. Citado na página 56.

KAHNG, M.; ANDREWS, P. Y.; KALRO, A.; CHAU, D. H. ActiVis: Visual exploration of industry-scale deep neural network models. *IEEE Transactions on Visualization and Computer Graphics*, v. 24, n. 1, p. 88–97, 2018. Citado na página 81.

KARIMI, H.; DERR, T.; TANG, J. Characterizing the decision boundary of deep neural networks. *Preprint arXiv:1912.11460*, 2019. Citado nas páginas 28, 60, 83 e 107.

KASS, R.; FININ, T. The need for user models in generating expert system explanations. *International Journal of Expert Systems*, v. 1, n. 4, 1988. Citado na página 64.

KAUFFMANN, J.; MÜLLER, K.-R.; MONTAVON, G. Towards explaining anomalies: A deep Taylor decomposition of one-class models. *Pattern Recognition*, Elsevier, v. 101, p. 107198, 2020. Citado na página 86.

KAUR, H.; NORI, H.; JENKINS, S.; CARUANA, R.; WALLACH, H.; VAUGHAN, J. W. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. Honolulu, Hawai'i, USA: ACM SIGCHI, 2020. p. 1–14. Citado nas páginas 100, 104 e 107.

KOHLBRENNER, M.; BAUER, A.; NAKAJIMA, S.; BINDER, A.; SAMEK, W.; LAPUSCHKIN, S. Towards best practice in explaining neural network decisions with LRP. In: IEEE. *2020 International Joint Conference on Neural Networks (IJCNN)*. Glasgow, UK, 2020. p. 1–7. Citado nas páginas 86, 87 e 107.

KRISHNA, S.; HAN, T.; GU, A.; POMBRA, J.; JABBARI, S.; WU, S.; LAKKARAJU, H. The disagreement problem in explainable machine learning: A practitioner's perspective. *Preprint arXiv:2202.01602*, 2022. Citado nas páginas 105 e 144.

KUCHARSKI, A. Study epidemiology of fake news. *Nature*, Nature Publishing Group, v. 540, n. 7634, p. 525–525, 2016. Citado na página 73.

KULESZA, T.; STUMPF, S.; BURNETT, M.; YANG, S.; KWAN, I.; WONG, W.-K. Too much, too little, or just right? Ways explanations impact end users' mental models. In: *2013 IEEE Symposium on Visual Languages and Human-Centric Computing*. San Jose, CA, USA: IEEE, 2013. p. 3–10. Citado na página 105.

KUMAR, H.; CHANDRAN, J. Is Shapley explanation for a model unique? *Preprint arXiv:2111.11946*, 2021. Citado nas páginas 68, 90, 91, 93, 95, 101 e 107.

KUMAR, I.; SCHEIDEGGER, C.; VENKATASUBRAMANIAN, S.; FRIEDLER, S. Shapley residuals: Quantifying the limits of the Shapley value for explanations. *Advances in Neural Information Processing Systems*, v. 34, 2021. Citado nas páginas 64, 101 e 107.

KUMAR, I. E.; VENKATASUBRAMANIAN, S.; SCHEIDEGGER, C.; FRIEDLER, S. Problems with Shapley-value-based explanations as feature importance measures. In: PMLR. *37th International Conference on Machine Learning*. Vienna, Austria, 2020. p. 5491–5500. Citado nas páginas 73, 91, 92, 95, 100, 101, 106, 107, 110, 111, 112 e 119.

LAPUSCHKIN, S.; WÄLDCHEN, S.; BINDER, A.; MONTAVON, G.; SAMEK, W.; MÜLLER, K.-R. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, Nature Publishing Group, v. 10, n. 1, p. 1–8, 2019. Citado nas páginas 31, 61, 81, 86, 87, 107, 132, 179 e 183.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado nas páginas 28, 46, 47, 61 e 183.

LEON, A. D.; CARRIERE, K. A generalized mahalanobis distance for mixed data. *Journal of Multivariate Analysis*, Elsevier, v. 92, n. 1, p. 174–185, 2005. Citado na página 105.

LEVEQUE, R. J. *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. Philadelphia, PA, USA: SIAM, 2007. Citado nas páginas 121, 122 e 123.

LI, Y.; DING, L.; GAO, X. On the decision boundary of deep neural networks. *Preprint arXiv:1808.05385*, 2018. Citado nas páginas 84 e 107.

LIAO, Q. V.; GRUEN, D.; MILLER, S. Questioning the AI: Informing design practices for explainable AI user experiences. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu, Hawai'i, USA: ACM SIGCHI, 2020. p. 1–15. Citado nas páginas 70, 85 e 106.

LINARDATOS, P.; PAPASTEFANOPOULOS, V.; KOTSIANTIS, S. Explainable AI: A review of machine learning interpretability methods. *Entropy*, MDPI, v. 23, n. 1, p. 18, 2020. Citado na página 76.

LIPOVETSKY, S.; CONKLIN, M. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, Wiley Online Library, v. 17, n. 4, p. 319–330, 2001. Citado nas páginas 93 e 107.

LIPTON, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, ACM New York, NY, USA, v. 16, n. 3, p. 31–57, 2018. Citado nas páginas 29, 41, 57, 58 e 76.

LIU, Y.; KHANDAGALE, S.; WHITE, C.; NEISWANGER, W. Synthetic benchmarks for scientific research in explainable machine learning. *Preprint arXiv:2106.12543*, 2021. Citado nas páginas 103 e 107.

LOHWEG, V. *Banknote Authentication*. 2013. UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Sciences. Citado na página 155.

LOMBROZO, T. The structure and function of explanations. *Trends in Cognitive Sciences*, Elsevier, v. 10, n. 10, p. 464–470, 2006. Citado na página 57.

LOU, Y.; CARUANA, R.; GEHRKE, J. Intelligible models for classification and regression. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Beijing, China: NY ACM, 2012. p. 150–158. Citado nas páginas 79 e 107.

LOU, Y.; CARUANA, R.; GEHRKE, J.; HOOKER, G. Accurate intelligible models with pairwise interactions. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Chicago, Illinois, USA: NY ACM, 2013. p. 623–631. Citado nas páginas 65, 79 e 107.

LUNDBERG, S. M.; ERION, G.; CHEN, H.; DEGRAVE, A.; PRUTKIN, J. M.; NAIR, B.; KATZ, R.; HIMMELFARB, J.; BANSAL, N.; LEE, S.-I. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, Nature Publishing Group, v. 2, n. 1, p. 56–67, 2020. Citado nas páginas 30, 31, 41, 50, 98, 107, 115 e 183.

LUNDBERG, S. M.; ERION, G. G.; LEE, S.-I. Consistent individualized feature attribution for tree ensembles. *Preprint arXiv:1802.03888*, 2018. Citado nas páginas 98, 107 e 136.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, CA, USA: Curran Associates Inc., 2017. p. 4768–4777. Citado nas páginas 30, 32, 75, 78, 87, 90, 92, 93, 94, 95, 98, 107, 111, 112, 117, 118, 119, 120, 131, 174 e 181.

LUNDBERG, S. M.; NAIR, B.; VAVILALA, M. S.; HORIBE, M.; EISSES, M. J.; ADAMS, T.; LISTON, D. E.; LOW, D. K.-W.; NEWMAN, S.-F.; KIM, J. *et al.* Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, Nature Publishing Group, v. 2, n. 10, p. 749–760, 2018. Citado nas páginas 31, 41, 98, 107, 110, 182 e 183.

MAATEN, L. V. D.; HINTON, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, JMLR.org, v. 9, n. 1, p. 2579–2605, 2008. Citado nas páginas 78 e 83.

MAIER, O.; HANDELS, H. Predicting stroke lesion and clinical outcome with random forests. In: SPRINGER. *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Athens, Greece, 2016. p. 219–230. Citado na página 50.

MARCÍLIO-JR, W. E.; ELER, D. M.; BREVE, F. Model-agnostic interpretation by visualization of feature perturbations. *Preprint arXiv:2101.10502*, 2021. Citado nas páginas 81 e 107.

MARSDEN, J. E.; TROMBA, A. *Vector calculus*. New York, NY, USA: Macmillan, 2003. Citado na página 118.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado nas páginas 15, 42 e 43.

MCKINNEY, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 1st. ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2012. Citado na página 72.

MILLER, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, Elsevier, v. 267, p. 1–38, 2019. Citado nas páginas 29, 85 e 112.

MISHRA, S.; DUTTA, S.; LONG, J.; MAGAZZENI, D. A survey on the robustness of feature importance and counterfactual explanations. *Preprint arXiv:2111.00358*, 2021. Citado nas páginas 78, 85 e 104.

MOLNAR, C. *Interpretable Machine Learning*: A guide for making black box models explainable. Durham, NC, USA: Lulu Press, Inc., 2019. Citado nas páginas 76, 80, 81, 88, 89, 90, 91, 92, 93, 95 e 99.

MUNROE, R. *Xkcd – A webcomic of romance, sarcasm, math, and language*. 2010. <https://xkcd.com/1838/>. Acesso em: 10-07-2023. Citado na página 60.

MURDOCH, W. J.; SINGH, C.; KUMBIER, K.; ABBASI-ASL, R.; YU, B. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 116, n. 44, p. 22071–22080, 2019. Citado nas páginas 27 e 76.

MURPHY, K. P. *Machine learning: A probabilistic perspective*. Cambridge, MA, USA: MIT press, 2012. Citado na página 49.

NASCIMENTO, D. C.; BARBOSA, B.; PEREZ, A. M.; CAIRES, D. O.; HIRAMA, E.; RAMOS, P. L.; LOUZADA, F. Risk management in e-commerce–A fraud study case using acoustic analysis through its complexity. *Entropy*, MDPI, v. 21, n. 11, p. 1087, 2019. Citado na página 40.

NEAL, B.; MITTAL, S.; BARATIN, A.; TANTIA, V.; SCICLUNA, M.; LACOSTE-JULIEN, S.; MITLIAGKAS, I. A modern take on the bias-variance tradeoff in neural networks. *Preprint arXiv:1810.08591*, 2018. Citado nas páginas 52, 53 e 54.

NOCEDAL, J.; WRIGHT, S. J. *Numerical optimization*. New York, NY, USA: Springer, 1999. Citado na página 116.

NONATO, L. G.; AUPETIT, M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Computer Society, Washington, DC, USA, v. 25, n. 8, p. 2650–2673, 2018. Citado nas páginas 78 e 82.

ORTIGOSSA, E. S.; DIAS, F. F.; BARR, B.; SILVA, C. T.; NONATO, L. G. T-Explainer: A model-agnostic explainability framework based on gradients. *Preprint arXiv:2404.16495*, 2024. Citado nas páginas 121, 140, 141, 142, 143, 144, 158, 160, 167, 170 e 174.

ORTIGOSSA, E. S.; DIAS, F. F.; NASCIMENTO, D. C. d. Getting over high-dimensionality: How multidimensional projection methods can assist data science. *Applied Sciences*, v. 12, n. 13, 2022. Citado na página 78.

ORTIGOSSA, E. S.; GONÇALVES, T.; NONATO, L. G. EXplainable artificial intelligence (XAI)–From theory to methods and applications. *IEEE Access*, v. 12, p. 80799–80846, 2024. Citado nas páginas 40, 53, 58, 61, 64, 65, 89, 92 e 96.

PASSI, S.; JACKSON, S. J. Trust in data science: Collaboration, translation, and accountability in corporate data science projects. *Proceedings of the ACM on Human-Computer Interaction*, ACM New York, NY, USA, v. 2, n. CSCW, p. 1–28, 2018. Citado na página 31.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, JMLR.org, v. 12, p. 2825–2830, nov 2011. ISSN 1532-4435. Citado nas páginas 71 e 72.

POOLE, D.; MACKWORTH, A.; GOEBEL, R. *Computational Intelligence: A Logical Approach*. Oxford, England: Oxford University Press, 1998. Citado na página 37.

POURSABZI-SANGDEH, F.; GOLDSTEIN, D. G.; HOFMAN, J. M.; VAUGHAN, J. W. W.; WALLACH, H. Manipulating and measuring model interpretability. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2021. (CHI '21), p. 1–52. Citado nas páginas 62, 63 e 69.

POYIADZI, R.; SOKOL, K.; SANTOS-RODRIGUEZ, R.; BIE, T. D.; FLACH, P. FACE: Feasible and actionable counterfactual explanations. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, NY, USA: Association for Computing Machinery, 2020. (AIES '20), p. 344–350. Citado nas páginas 85 e 107.

PREECE, A.; HARBORNE, D.; BRAINES, D.; TOMSETT, R.; CHAKRABORTY, S. Stakeholders in explainable AI. *Preprint arXiv:1810.00184*, 2018. Citado na página 73.

PRESS, W. H. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge, England: Cambridge University Press, 2007. Citado na página 116.

RAIMUNDO, M. M.; NONATO, L. G.; POCO, J. Mining pareto-optimal counterfactual antecedents with a branch-and-bound model-agnostic algorithm. *Data Mining and Knowledge Discovery*, Springer, p. 1–33, 2022. Citado nas páginas 85 e 107.

RAUBER, P. E.; FADEL, S. G.; FALCAO, A. X.; TELEA, A. C. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, IEEE, v. 23, n. 1, p. 101–110, 2016. Citado nas páginas 44, 82, 83 e 107.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. Model-agnostic interpretability of machine learning. *Preprint arXiv:1606.05386*, 2016. Citado nas páginas 89, 107 e 181.

_____. Nothing else matters: Model-agnostic explanations by identifying prediction invariance. *Preprint arXiv:1611.05817*, 2016. Citado nas páginas 89 e 107.

_____. "Why should I trust you?" Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA: NY ACM, 2016. p. 1135–1144. Citado nas páginas 28, 30, 60, 62, 66, 67, 68, 69, 70, 77, 88, 95, 107, 117, 124, 134 e 139.

_____. Anchors: High-precision model-agnostic explanations. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, USA: AAAI Press, 2018. v. 32, n. 1. Citado nas páginas 89 e 107.

RODRíGUEZ, P.; BAUTISTA, M. A.; GONZàLEZ, J.; ESCALERA, S. Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, v. 75, p. 21–31, 2018. Citado na página 125.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 42.

RUDER, S. An overview of gradient descent optimization algorithms. *Preprint arXiv:1609.04747*, 2016. Citado na página 45.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. Citado na página 45.

RUSSELL, S.; NORVIG, P. *Artificial intelligence: A modern approach*. 3rd. ed. Upper Saddle River, NJ, USA: Pearson Education, Inc., 2010. Citado na página 27.

SAMEK, W.; WIEGAND, T.; MÜLLER, K. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR*, abs/1708.08296, 2017. Citado na página 28.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, IBM, v. 3, n. 3, p. 210–229, 1959. Citado na página 38.

SCHEINBERG, K. Finite difference gradient approximation: To randomize or not? *INFORMS Journal on Computing*, INFORMS, v. 34, n. 5, p. 2384–2388, 2022. Citado na página 181.

SCHRECKENBERGER, C.; BARTELT, C.; STUCKENSCHMIDT, H. iDropout: Leveraging deep Taylor decomposition for the robustness of deep neural networks. In: SPRINGER. *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*. Rhodes, Greece, 2019. p. 113–126. Citado na página 86.

SHANKER, M.; HU, M. Y.; HUNG, M. S. Effect of data standardization on neural network training. *Omega*, Elsevier, v. 24, n. 4, p. 385–397, 1996. Citado na página 123.

SHAPLEY, L. S. A value for n-person games. In: *Contributions to the Theory of Games (AM-28), Volume II*. Princeton, New Jersey, USA: Princeton University Press, 1953. p. 2. Citado na página 90.

SHEFRIN, H. *Beyond greed and fear: Understanding behavioral finance and the psychology of investing*. Oxford, England: Oxford University Press, 2002. Citado na página 184.

SHRIKUMAR, A.; GREENSIDE, P.; KUNDAJE, A. Learning important features through propagating activation differences. In: PMLR. *34th International Conference on Machine Learning*. Sydney, Australia, 2017. p. 3145–3153. Citado nas páginas 45, 87, 98, 107 e 110.

SHRIKUMAR, A.; GREENSIDE, P.; SHCHERBINA, A.; KUNDAJE, A. Not just a black box: Learning important features through propagating activation differences. *Preprint arXiv:1605.01713*, 2016. Citado nas páginas 87 e 107.

SIMONYAN, K.; VEDALDI, A.; ZISSERMAN, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Preprint arXiv:1312.6034*, 2013. Citado nas páginas 86, 107 e 117.

SINGH, G.; MÉMOLI, F.; CARLSSON, G. E. *et al.* Topological methods for the analysis of high dimensional data sets and 3D object recognition. *PBG@ Eurographics*, v. 2, p. 091–100, 2007. Citado na página 81.

SMILKOV, D.; THORAT, N.; KIM, B.; VIÉGAS, F.; WATTENBERG, M. SmoothGrad: Removing noise by adding noise. *Preprint arXiv:1706.03825*, 2017. Citado nas páginas 30, 87 e 107.

SOHNS, J.-T.; GARTH, C.; LEITTE, H. Decision boundary visualization for counterfactual reasoning. In: WILEY ONLINE LIBRARY. Hoboken, New Jersey, USA, 2023. v. 42, n. 1, p. 7–20. Citado na página 107.

SONG, L.; LI, P.-W.; GU, Y.; FAN, C.-M. Generalized finite difference method for solving stationary 2D and 3D stokes equations with a mixed boundary condition. *Computers & Mathematics with Applications*, Elsevier, v. 80, n. 6, p. 1726–1743, 2020. Citado na página 181.

STEPIN, I.; ALONSO, J. M.; CATALA, A.; PEREIRA-FARIÑA, M. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, IEEE, v. 9, p. 11974–12001, 2021. Citado nas páginas 69, 84, 85 e 107.

STEWART, J.; CLEGG, D. K.; WATSON, S. *Calculus: Early transcendentals*. Boston, MA, USA: Cengage Learning, 2020. Citado na página 87.

ŠTRUMBELJ, E.; KONONENKO, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, Springer, v. 41, n. 3, p. 647–665, 2014. Citado nas páginas 93 e 107.

SU, J.; VARGAS, D. V.; SAKURAI, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 23, n. 5, p. 828–841, 2019. Citado na página 28.

SUNDARARAJAN, M.; NAJMI, A. The many Shapley values for model explanation. In: PMLR. *37th International Conference on Machine Learning*. Vienna, Austria, 2020. p. 9269–9278. Citado nas páginas 90, 92, 94, 100, 107, 111, 112 e 119.

SUNDARARAJAN, M.; TALY, A.; YAN, Q. Axiomatic attribution for deep networks. In: PMLR. *International Conference on Machine Learning*. Brookline, MA, USA, 2017. p. 3319–3328. Citado nas páginas 87 e 107.

TALEB, N. N. *Fooled by randomness: The hidden role of chance in life and in the markets*. New York City, NY, USA: Random House Incorporated, 2005. v. 1. Citado na página 184.

TAN, S.; HOOKER, G.; KOCH, P.; GORDO, A.; CARUANA, R. Considerations when learning additive explanations for black-box models. *Machine Learning*, Springer, p. 1–27, 2023. Citado nas páginas 29, 30, 69, 78, 79, 80, 107, 134 e 172.

THEODOROU, A.; WORTHAM, R. H.; BRYSON, J. J. Designing and implementing transparency for real time inspection of autonomous robots. *Connection Science*, Taylor & Francis, v. 29, n. 3, p. 230–241, 2017. Citado na página 73.

THIAGARAJAN, J. J.; KAILKHURA, B.; SATTIGERI, P.; RAMAMURTHY, K. N. TreeView: Peeking into deep neural networks via feature-space partitioning. *Preprint arXiv:1611.07429*, 2016. Citado nas páginas 88 e 107.

TJOA, E.; GUAN, C. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, 2020. Citado nas páginas 28, 46, 59, 61, 63, 71, 76 e 83.

TOWERS, J. D. Finite difference methods for approximating heaviside functions. *Journal of Computational Physics*, Elsevier, v. 228, n. 9, p. 3478–3489, 2009. Citado na página 181.

TSIRTSIS, S.; RODRIGUEZ, M. G. Decisions, counterfactual explanations and strategic behavior. *Advances in Neural Information Processing Systems*, v. 33, p. 16749–16760, 2020. Citado na página 103.

TURNER, R. A model explanation system. In: IEEE. *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. Salerno, Italy, 2016. p. 1–6. Citado na página 107.

United Nations. *Statistics Division - National Accounts*. New York, NY 10017, USA: UN, 2021. <https://unstats.un.org/unsd/snaama/Basic>. Acesso em: 10-06-2023. Citado na página 183.

UPADHYAY, S.; JOSHI, S.; LAKKARAJU, H. Towards robust and reliable algorithmic recourse. *Advances in Neural Information Processing Systems*, v. 34, p. 16926–16937, 2021. Citado na página 172.

VANSCHOREN, J.; RIJN, J. N. V.; BISCHL, B.; TORGO, L. OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, ACM New York, NY, USA, v. 15, n. 2, p. 49–60, 2014. Citado na página 155.

VAPNIK, V. *The nature of statistical learning theory*. Heidelberg, Germany: Springer Science & Business Media, 1999. Citado na página 41.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. *Advances in Neural Information Processing Systems*, v. 30, 2017. Citado nas páginas 61, 81 e 183.

VERMA, S.; DICKERSON, J.; HINES, K. Counterfactual explanations for machine learning: A review. *Preprint arXiv:2010.10596*, 2020. Citado nas páginas 84 e 85.

VIG, J. BertViz: A tool for visualizing multihead self-attention in the BERT model. In: ICLR. *ICLR workshop: Debugging machine learning models*. New Orleans, LA, USA, 2019. v. 23, p. 1–6. Citado nas páginas 82 e 107.

_____. A multiscale visualization of attention in the transformer model. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, 2019. p. 37–42. Citado nas páginas 82 e 107.

WACHTER, S.; MITTELSTADT, B.; RUSSELL, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, HeinOnline, v. 31, p. 841, 2017. Citado nas páginas 85 e 107.

WANG, K.; MUTHUKUMAR, V.; THRAMPOULIDIS, C. Benign overfitting in multiclass classification: All roads lead to interpolation. *Advances in Neural Information Processing Systems*, v. 34, p. 24164–24179, 2021. Citado na página 54.

WEERTS, H. J.; IPENBURG, W. van; PECHENIZKIY, M. A human-grounded evaluation of SHAP for alert processing. *Preprint arXiv:1907.03324*, 2019. Citado nas páginas 103, 107 e 182.

WERBOS, P. J. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, IEEE, v. 78, n. 10, p. 1550–1560, 1990. Citado na página 45.

WEST, D. M. *The future of work: Robots, AI, and automation*. Washington, DC, USA: Brookings Institution Press, 2018. Citado na página 27.

WIEGREFFE, S.; PINTER, Y. Attention is not not explanation. *Preprint arXiv:1908.04626*, 2019. Citado nas páginas 61, 82 e 183.

WILSON, S. J. *Proofs of the Limit Inequalities*. 2010. <http://www.milefoot.com/math/calculus/limits/LimitInequalityProofs05.htm>. Acesso em: 09-02-2023. Citado na página 120.

WOJTAS, M.; CHEN, K. Feature importance ranking for deep learning. *Advances in Neural Information Processing Systems*, v. 33, p. 5105–5114, 2020. Citado nas páginas 69, 78 e 125.

WOLPERT, D. H. Stacked generalization. *Neural Networks*, Elsevier, v. 5, n. 2, p. 241–259, 1992. Citado na página 66.

WU, J.-L.; CHANG, P.-C.; WANG, C.; WANG, K.-C. ATICVis: A visual analytics system for asymmetric transformer models interpretation and comparison. *Applied Sciences*, MDPI, v. 13, n. 3, p. 1595, 2023. Citado na página 81.

XENOPOULOS, P.; CHAN, G.; DORAISWAMY, H.; NONATO, L. G.; BARR, B.; SILVA, C. GALE: Globally assessing local explanations. In: *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*. Virtual: PMLR, 2022. (Proceedings of Machine Learning Research, v. 196), p. 322–331. Citado nas páginas 81, 107 e 136.

XU, K.; YUAN, J.; WANG, Y.; SILVA, C.; BERTINI, E. MTSeer: Interactive visual exploration of models on multivariate time-series forecast. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2021. (CHI '21), p. 1–15. Citado nas páginas 99 e 107.

YANG, M.; KIM, B. Benchmarking attribution methods with relative feature importance. *Preprint arXiv:1907.09701*, 2019. Citado nas páginas 63, 102, 103 e 107.

YE, M.; SUN, Y. Variable selection via penalized neural network: A drop-out-one loss approach. In: PMLR. *International Conference on Machine Learning*. Stockholm, Sweden, 2018. p. 5620–5629. Citado na página 103.

YOUSEFZADEH, R.; O'LEARY, D. P. Interpreting neural networks using flip points. *Preprint arXiv:1903.08789*, 2019. Citado nas páginas 47, 84 e 107.

_____. Investigating decision boundaries of trained neural networks. *Preprint arXiv:1908.02802*, 2019. Citado na página 90.

YUAN, J.; CHAN, G. Y.-Y.; BARR, B.; OVERTON, K.; REES, K.; NONATO, L. G.; BERTINI, E.; SILVA, C. T. SUBPLEX: A visual analytics approach to understand local model explanations at the subpopulation level. *IEEE Computer Graphics and Applications*, v. 42, n. 6, p. 24–36, 2022. Citado nas páginas 83 e 107.

ZEDNIK, C. Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy & Technology*, Springer, v. 34, n. 2, p. 265–288, 2021. Citado nas páginas 79, 81 e 107.

ZHANG, C.; BENGIO, S.; HARDT, M.; RECHT, B.; VINYALS, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, ACM New York, NY, USA, v. 64, n. 3, p. 107–115, 2021. Citado na página 54.

ZHANG, Q.; ZHU, S.-C. Visual interpretability for deep learning: A survey. *Preprint arXiv:1802.00614*, 2018. Citado nas páginas 28, 29, 81 e 107.


ICMC
USP
SÃO CARLOS