UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sistema antifraude para detecção de não conformidades em gastos corporativos

Luiz Gustavo Ribeiro

Dissertação de Mestrado do Programa de Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria (MECAI)

ICMC USP SÃO CARLOS

Luiz Gustavo Ribeiro

Sistema antifraude para detecção de não conformidades em gastos corporativos

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria. *VERSÃO REVISADA*

Área de Concentração: Matemática, Estatística e Computação

Orientador: Prof. Dr. Fabricio Simeoni de Sousa

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)

R484s

Ribeiro, Luiz Gustavo
Sistema antifraude para detecção de não conformidades em gastos corporativos / Luiz Gustavo Ribeiro; orientador Fabricio Simeoni de Sousa. -- São Carlos, 2024.
81 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Fraude. 2. Reembolso de despesas. 3. Aprendizado de máquina. 4. Análise exploratória. I. Sousa, Fabricio Simeoni de, 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

Luiz Gustavo Ribeiro

Anti-fraud system for detecting non-compliance in corporate expenses

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Professional Master's Program in Mathematics Statistics and Computing Applied to Industry, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Mathematics, Statistics and Computing

Advisor: Prof. Dr. Fabricio Simeoni de Sousa

USP – São Carlos
June 2024

Este trabalho é dedicado aos meus familiares e amigos
que me apoiram nos momentos mais difíceis dessa jornada.
Em especial, aos funcionários do Instituto de Ciências Matemáticas e Computação (ICMC).

AGRADECIMENTOS

Os agradecimentos principais são direcionados ao meu filho Kauê Ribeiro. Sua presença em minha vida é a maior força que me impulsiona. Aos meus irmãos, Allan e Lucas, pela motivação, companheirismo e por sempre acreditarem em mim. À minha mãe, Maria Luiza, pelo amor infinito, pelas palavras de incentivo e por me ensinar os valores que me guiam.

Aos amigos que me apoiam e se preocupam com meu futuro e bem estar. Em destaque para João Coimbra e Guyan Di Bonis.

Aos professores que sempre foram dedicados e pacientes na minha jornada de aprendizado e formação profissional.

Por fim, agradeço ao professor Fabricio Simeoni de Sousa pela orientação e pelas ricas discussões realizadas ao longo deste trabalho.

“Nesse grande futuro,
não podemos esquecer do nosso passado.”
(Bob Marley)

RESUMO

RIBEIRO, L. G. **Sistema antifraude para detecção de não conformidades em gastos corporativos**. 2024. 81 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

Os colaboradores de uma organização frequentemente submetem solicitações de reembolso para despesas relacionados ao desenvolvimento do seu trabalho. Dessa forma, é de grande importância a validação e análise dessas transações, na intenção de verificar se essas fazem parte do desenvolvimento do trabalho e com isso, devem ser reembolsadas. A correta identificação dessas transações é uma tarefa complexa e cara, uma vez que existe a necessidade de uma pessoa auditar cada transação, analisando um contexto macro, relacionando a função do colaborador com o gasto realizado. A presente contribuição contorna esses desafios utilizando conceitos de análise de dados e aprendizado de máquina, na intenção de criar modelo capaz de analisar e classificar uma transação como em conformidade e não conformidade com as politicas organizacionais. Os resultados para os modelos desenvolvidos mostram uma melhoria acentuada em relação aos esforços convencionais para identificação das transações, com precisão se aproximando dos níveis de potencial prático.

Palavras-chave: Análise Exploratória, Aprendizado de Máquina, Fraude, Reembolso de Despesas.

ABSTRACT

RIBEIRO, L. G. **Anti-fraud system for detecting non-compliance in corporate expenses**. 2024. 81 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

Employees of an organization often submit reimbursement requests for expenses related to the development of their work. The validation and analysis of these transactions are critical to ensure proper reimbursement for work development. The correct identification of these transactions is a complex and costly task. It requires a person to audit each transaction, relating it to the employee's role, and analyze the transaction in the macro context of the organization's overall finances and budget. This contribution overcomes these challenges by using data analysis and machine learning concepts to create a model capable of analyzing and classifying a transaction as compliant or non-compliant with organizational policies. The results for the developed models show a marked improvement over conventional efforts to identify transactions with accuracy approaching practical potential levels.

Keywords: Exploratory Analysis, Machine Learning, Fraud, Reimbursement Expenses.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AED	Análise Exploratória de Dados
AUC	*Area Under the Curve*
FN	Falsos Negativos
FP	Falsos Positivos
IA	Inteligência Artificial
IQR	Intervalo Interquartil
ML	*machine learning*
NLP	*Natural Language Processing*
RF	*Random Forest*
RFE	*Recursive Feature Elimination*
ROC	*Receiver Operating Characteristic*
SHAP	*SHapley Additive exPlanations*
TFP	Taxa de Falsos Positivos
TVP	Taxa de Verdadeiros Positivos
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

1.1 Contexto

A gestão de despesas corporativas desempenha um papel fundamental na saúde financeira e no desempenho geral de uma organização. Ela permite um controle financeiro eficiente, redução de custos, tomada de decisões informadas, conformidade com políticas e regulamentações, além de melhorar a eficiência operacional. Em contrapartida, as fraudes em gastos corporativos representam uma preocupação significativa para as organizações, pois podem apresentar consequências financeiras e reputacionais negativas.

O relatório global ***Occupational Fraud 2022: Report to the Nations*** (ACFE, 2023), em sua 12ª edição, fornece informações valiosas sobre os custos, métodos, perpetradores, impacto e resultados de esquemas de fraude ocupacional derivados de 2.110 casos reais de fraude que afetaram organizações em 133 países e 23 setores industriais, entre janeiro de 2020 e setembro de 2021. O estudo revela que:

- Estima-se que as organizações percam 5% da receita anual para fraudes. Neste relatório, em mais de 2000 casos estudados, totalizou-se $3,6 bilhões em perdas;

- Fraudes em reembolso de despesas representam 11% do total de casos estudados, com uma perda mediana de $40.000 e duração média de até 18 meses;

- Entre várias técnicas de detecção de fraudes, a denúncia foi o método mais comum, com 42% dos casos, seguida pela auditoria interna com 16% dos casos. Mais da metade das denúncias vieram dos próprios funcionários;

- Fraudes detectadas por meio de denúncia ou auditoria interna têm uma duração média de 12 meses e uma perda mediana entre $108.000 e $117.000. Monitoramento de dados

e transações reduzem pela metade tanto a perda mediana quanto a duração média do esquema de fraude;

- A maior incidência de esquemas de fraude de reembolso se encontram nas áreas da construção, energia, tecnologia e administração pública.

De acordo com Scarinci (2021), a ocorrência de fraudes em empresas brasileiras de capital aberto é altamente influenciada por características tanto internas quanto externas dessas empresas. A autora relata que as características do ambiente, como o setor de atuação e o cenário econômico, e certas características das empresas, como baixa rentabilidade, risco de falência, utilização dos serviços de auditoria de uma firma renomada, tamanho da empresa e montante destinado a doações políticas, exerceram um papel determinante na probabilidade de ocorrência de fraudes corporativas no mercado brasileiro.

Para prevenir fraudes em gastos corporativos, as organizações devem implementar medidas eficazes, incluindo criação de políticas claras, implementação de controles internos adequados, realização de auditorias regulares, promoção de uma cultura ética, treinamento dos funcionários, punição adequada para os casos de fraude e a utilização de ferramentas de software para facilitar uma análise mais aprofundada do fluxo de caixa da empresa, como despesas relacionadas a viagens corporativas e reembolso de despesas.

Os reembolsos de despesas são um caso especial que consiste na devolução de despesas devidamente comprovadas por um colaborador. Essas despesas geralmente são feitas em reuniões com clientes externos e viagens de negócios que cobrem alimentação, hospedagem, passagens aéreas ou outros transportes, combustível e material de trabalho. Uma política de reembolso bem definida é a chave para garantir a transparência no fluxo de trabalho financeiro da empresa e pode ajudar a prevenir casos de não conformidades e fraudes. Neste caso, o descumprimento e a fraude podem ser entendidos como qualquer omissão ou ato doloso promovido para prejudicar e ludibriar a empresa.

1.2 Motivação

Auditar bases de dados complexas com uma quantidade massiva de informações manualmente apresenta desafios significativos. Conforme a quantidade de dados aumenta, a auditoria manual pode se tornar inviável em termos de eficiência e cobertura. O conjunto de bases de dados de uma organização podem conter informações provenientes de diversas fontes, com diferentes formatos e estruturas, requerendo o uso de ferramentas específicas e técnicas analíticas avançadas para identificar padrões, relações não óbvias entre os dados e comportamentos incomuns.

Mesmo que os auditores sejam especialistas, eles podem estar sujeitos a viéses inconscientes que podem influenciar suas decisões e interpretações. Além disso, os seres humanos são

suscetíveis a limitações cognitivas, como fadiga e falta de atenção, especialmente ao lidar com grandes volumes de dados.

Na terceira edição do relatório, ***2024 Anti-Fraud Technology Benchmarking Report*** (ACFE, 2024), são examinadas as tendências atuais e esperadas na adoção de análises tradicionais, Inteligência Artificial (IA) e IA generativa, além de outras tecnologias utilizadas para combater fraudes. O levantamento de 22 perguntas realizado em outubro de 2023 resultou em 1187 respostas, representando organizações em 111 países ao redor do mundo, com a maioria delas concentradas nos Estados Unidos e Canadá (42%). Os respondentes foram solicitados a fornecer informações sobre o uso de várias tecnologias em suas organizações como parte de suas iniciativas de combate à fraude.

As principais descobertas desse relatório são:

- Nove em cada dez organizações (91%) utilizam técnicas de análise de dados em seus programas anti-fraude;
- Estima-se triplicar o uso de IA e aprendizado de máquina nos próximos dois anos;
- Espera-se que 83% das organizações implementem IA generativa como parte de seus programas anti-fraude nos próximos dois anos;
- Nos próximos dois anos, três em cada cinco organizações (59%) planejam aumentar seus orçamentos em tecnologias anti-fraude.

Avanços recentes em inteligência artificial permitem construir sistemas robustos de predição para realizar a detecção de não conformidade em reembolsos de despesas corporativas de forma rápida e eficaz, reduzindo o tempo necessário para auditorias e aumentando a confiabilidade das informações. Além disso, linguagens de programação modernas e eficientes oferecem uma ampla variedade de bibliotecas e frameworks dedicados à inteligência artificial, facilitando o desenvolvimento de modelos e sua integração com outras tecnologias e sistemas em ambientes corporativos complexos.

1.3 Objetivo do projeto

1.3.1 Objetivo geral

Esta dissertação visa abordar conceitos e metodologias de aprendizado de máquina para a criação de um modelo preditivo capaz de detectar não conformidades em reembolso de despesas corporativas com objetivo principal de melhorar a detecção de fraudes, reduzir perdas financeiras, garantir conformidade regulatória e aumentar a eficiência operacional nas organizações.

1.3.2 Objetivos específicos

- Identificar e coletar dados sobre reembolsos de despesas, incluindo informações sobre transações, funcionários, departamentos e políticas da empresa.
- Selecionar e pré-processar os atributos relevantes dos dados, incluindo limpeza, normalização e engenharia de características, para preparar o conjunto de dados para modelagem.
- Treinar e otmizar algoritmos de aprendizado de máquina para maximizar sua performance.
- Avaliar e comparar o desempenho dos classificadores utilizando métricas adequadas para quantificar sua capacidade de detectar efetivamente casos de não conformidade.
- Interpretar as previsões do classificador escolhido.

1.4 Estrutura da Pesquisa

A pesquisa está estruturada em cinco seções, incluindo essa introdução. Na segunda seção será apresentado o referencial teórico, que contem os principais conceitos relacionados com o projeto. Na terceira seção, apresenta-se o desenho metodológico para solucionar o problema, em seguida na quarta seção, expõem se os resultados e suas análises. Por último, na quinta seção, é apresentado as conclusões e considerações finais sobre os pontos mais relevantes apontados na pesquisa.

CAPÍTULO

2

REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos fundamentais de análise estatística e aprendizado de máquina para o desenvolvimento, avaliação e interpretação do modelo preditivo destinado à detecção de não conformidades em gastos corporativos. Estes conceitos serão essenciais para alcançar os objetivos desta pesquisa.

2.1 Análise Exploratória de Dados para Seleção de Atributos

Análise Exploratória de Dados (AED) é uma abordagem à análise de conjuntos de dados para entender e resumir suas principais características através de medidas de estatística descritiva e técnicas de visualização. AED permite compreender a estrutura dos dados, identificar padrões, tendências e anomalias, além de fornecer as bases para as etapas subsequentes de seleção e engenharia de atributos no desenvolvimento de modelos preditivos de aprendizado de máquina.

2.1.1 Análise Estatística de Dados

A análise estatística tem como objetivo determinar a qualidade e o poder preditivo dos atributos em relação a classe alvo de um conjunto de dados. Essa abordagem proporciona uma compreensão abrangente dos dados, destacando a importância de ser a primeira etapa em qualquer estudo de conjunto de dados.

*2.1.1.1 **Análise Descritiva***

Análise descritiva (ou análise univariada) fornece uma compreensão das características de cada atributo do conjunto de dados (FACELI *et al.*, 2011). Existem diferentes tipos de atributos, cada um representando um tipo específico de informação:

1. **Atributos Categóricos**

    - **Nominais:** Representam categorias qualitativas sem ordenação natural, como sexo, cor, tipo de produto. São geralmente representados por strings ou códigos numéricos.
    - **Ordinais:** Representam categorias qualitativas com ordenação natural, como nível de escolaridade, números de estrela de locais ou produtos. São geralmente representados por números inteiros.

2. **Atributos Numéricos**

    - **Discretos:** Assumem um conjunto finito de valores, como quantidade de filhos e quantidade de banheiros de uma casa. São geralmente representados por números inteiros.
    - **Contínuos:** Assumem qualquer valor dentro de um intervalo, como altura, peso e temperatura. São geralmente representados por números reais.

3. **Textual**

    Representam textos ou sequências de caracteres. Exemplos incluem descrições, nomes de produtos, comentários de clientes. Esses atributos são frequentemente tratados de maneira especial, utilizando técnicas específicas de Processamento de Linguagem Natural, ou do inglês *Natural Language Processing* (NLP), para extração de recursos e análise.

Medidas de Centralidade

As medidas de centralidade visam resumir um conjunto de dados em um único valor, fornecendo uma ideia geral do centro da distribuição dos dados. Segundo Gama *et al.* (2015, p. 27), as medidas de centralidade definem pontos de referência nos dados e variam para dados numéricos e simbólicos. Para atributos categórico ou simbólicos, usualmente utiliza-se a moda, e para atributos numéricos são empregadas a média aritmética, mediana e percentil.

- **Média aritmética:** A soma de todos os valores dividida pelo número total de valores. É a medida mais conhecida e utilizada.
- **Mediana:** O valor que divide o conjunto de dados em dois grupos com o mesmo número de elementos. É menos sensível a pontos de dados que se desviam significativamente do restante do conjunto de dados (*outliers*) em comparação com a média.
- **Moda:** O valor que ocorre com mais frequência em um conjunto de dados. É útil para dados categóricos.
- **Percentis:** Fatias da distribuição de dados, revelando a porcentagem de valores abaixo de um determinado ponto. Por exemplo, o percentil 50, também conhecido como mediana, é o valor abaixo do qual 50% dos dados estão contidos. Já o percentil 25 indica o valor abaixo do qual 25% dos dados estão contidos, e assim por diante.

Medidas de Espalhamento

Medidas de espalhamento, também conhecidas como medidas de dispersão, são estatísticas que indicam a variabilidade ou a dispersão dos valores de um conjunto de dados em relação a um valor central, como a média (FACELI *et al.*, 2011). Elas fornecem informações sobre a homogeneidade ou heterogeneidade do conjunto de dados. Algumas das medidas de espalhamento mais comuns incluem:

- **Amplitude:** A diferença entre o maior e o menor valor do conjunto de dados. É uma medida simples, mas sensível a *outliers*.
- **Variância:** A média dos quadrados das diferenças entre cada valor e a média do conjunto de dados. Indica a dispersão média dos valores em relação à média.
- **Desvio Padrão:** A raiz quadrada da variância. Fornece uma medida de dispersão em uma escala semelhante à dos dados originais.
- **Intervalo Interquartil (IQR):** A diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1). É uma medida de dispersão que utiliza percentis para indicar a dispersão dos dados em torno da mediana, sendo menos sensível a *outliers* do que a amplitude.

2.1.1.2 *Análise de Correlação*

A análise de correlação, também conhecida como análise bivariada, é um método estatístico que busca quantificar a associação entre duas variáveis. Essa análise permite identificar se existe uma relação entre as variáveis e, em caso afirmativo, qual a força e a direção dessa relação.

Coeficiente de Correlação de Pearson

O coeficiente de correlação de Pearson (r) é a medida mais utilizada para quantificar a força da correlação linear entre duas variáveis (PRESS *et al.*, 2007, p. 745):

$$r = \rho_{XY} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{2.1}$$

Onde:

- ρ_{XY} representa a correlação entre as variáveis X e Y;
- n é o número de observações;
- X_i e Y_i são as observações individuais para as variáveis X e Y;
- $\bar{X}$ e $\bar{Y}$ são as médias das variáveis X e Y, respectivamente.

Na equação (2.1), o valor de r varia entre -1 e 1, sendo:

- $r = 0$: Ausência de relação significativa entre as variáveis. Correlação nula;
- $0 < r < 1$: Correlação linear positiva, com a força da correlação aumentando à medida que r se aproxima de 1, ou seja, o aumento de uma variável está associado ao aumento da outra;
- $-1 < r < 0$: Correlação linear negativa, com a força da correlação aumentando à medida que r se aproxima de -1, ou seja, o aumento de uma variável está associado à diminuição da outra.

Técnicas de visualização de dados, como histogramas, boxplots e gráficos de dispersão, são ferramentas valiosas para complementar a análise de estatística univariada e multivariada. A escolha da técnica mais adequada depende do tipo de atributo, da distribuição dos dados e do objetivo da análise. A Quadro 1 ilustra como cada técnica pode ser utilizada em relação ao tipo de atributo dos dados.

Quadro 1 – Técnicas de Estatística Descritiva

Tipo do Atributo	**Técnicas de Estatística Descritiva**	**Exemplos**
Categórico Nominal	Frequência absoluta e relativa Moda Gráfico de barras	Sexo cor
Categórico Ordinal	Frequência absoluta e relativa Mediana Quartis Gráfico de boxplot	Classificação de qualidade (ruim, bom, ótimo)
Numérico Discreto	Média, Mediana, Moda Desvio padrão Variância Quartis Gráfico de histograma	Número de filhos
Numérico Contínuo	Média, Mediana, Moda Desvio padrão Variância Coeficiente de assimetria Coeficiente de curtose Gráfico de histograma	Altura Peso Temperatura

2.2 Aprendizado de Máquina

Ao longo das décadas, o campo do aprendizado de máquina, ou do inglês *machine learning* (ML) passou por uma evolução notável, impulsionada por progressos na capacidade computacional, disponibilidade abundante de dados e o desenvolvimento de algoritmos cada vez mais sofisticados.

Aprendizado de máquina acontece quando um programa de computador aprende de uma experiência E com respeito a alguma tarefa T e com alguma medida de performance P, se a performance em T, medida por P, melhora com a experiência E (MITCHELL, 1997, p. 2).

O processo de aprendizado envolve a capacidade do modelo em aprimorar seu desempenho em uma tarefa específica à medida que adquire experiência, enquanto essa melhoria é mensurada por algum critério de desempenho.

Para Faceli *et al.* (2011, p. 54), um algoritmo de ML é uma função F que, dado um conjunto de exemplos rotulados, constrói um estimador. Se o rótulo pertence a um domínio de valores nominais e não ordenados, o estimador gerado é um classificador. Se o rótulo pertence a um domínio de valores infinitos e ordenados, tem-se um problema de regressão, que induz um regressor.

Um algoritmo de ML aprende uma aproximação $\hat{F}$ que permite estimar o valor produzido pelo classificador F para novas observações com o objetivo de minimizar o valor esperado de alguma função de perda especificada $L(y, F(x))$. A função de perda é uma medida que quantifica o quão bem o modelo está performando em relação aos rótulos verdadeiros dos dados.

Modelos de ML são frequentemente considerados "caixas pretas" porque, embora possam fazer previsões altamente precisas, o processo interno de como essas previsões são feitas pode ser complexo e não transparente. A teoria do aprendizado estatístico é um conjunto de princípios matemáticos e estatísticos que fornecem uma base teórica para compreender como os modelos de aprendizado de máquina generalizam a partir de dados de treinamento para fazer previsões ou tomar decisões em dados não vistos.

Hastie, Tibshirani e Friedman (2009) propõem uma metodologia unificada que abrange diversos métodos de aprendizado, enfatizando a interpretação e a compreensão dos princípios fundamentais. Além disso, discutem temas como seleção de variáveis, técnicas de regularização e avaliação de modelos.

2.3 Aprendizado Supervisionado para Classificação Binária

Aprendizado de Máquina Supervisionado para Classificação Binária é uma subárea do aprendizado de máquina que se concentra em treinar modelos para realizar a tarefa de categorização de dados em duas classes distintas e mutuamente exclusivas.

Nesse cenário, um modelo é alimentado com um conjunto de dados de treinamento, onde cada exemplo é descrito por um vetor de atributos ou características associado a um rótulo da classe alvo já conhecida. O objetivo central é capacitar o modelo a aprender padrões e relações nos dados, e usar esse conhecimento para classificar novas instâncias não rotuladas em uma das duas classes alvo (MONARD; BARANAUSKAS, 2003).

Por exemplo, em um cenário de detecção de fraude em transações financeiras, o aprendizado de máquina supervisionado para classificação binária seria usado para determinar se uma transação é fraudulenta (classe positiva) ou não (classe negativa) com base em características da transação, como valor, localização e histórico do usuário.

2.3.1 Árvore de Decisão

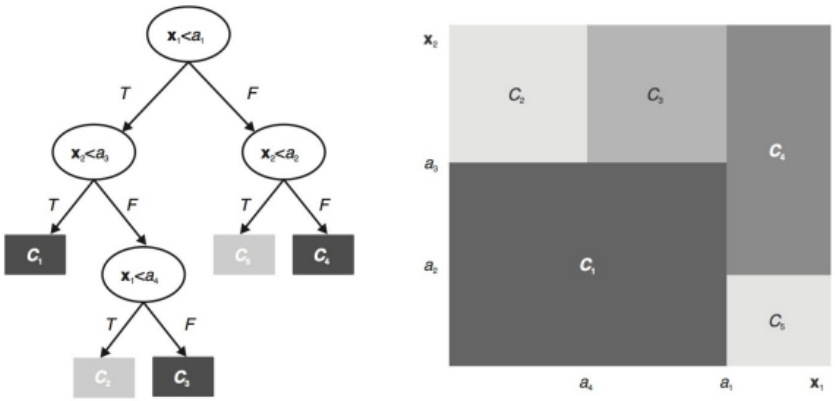
É um método de aprendizado de máquina supervisionado que representa decisões e suas consequências em uma estrutura hierárquica semelhante a uma árvore. Árvores de decisão são amplamente usadas para tarefas de classificação e regressão e são conhecidas por sua interpretabilidade e facilidade de compreensão, tornando-as uma opção popular em muitos domínios.

Árvore de decisão usa a estratégia de dividir para conquistar, dividindo um problema complexo em problemas mais simples, aos quais são aplicadas regras de decisões para solucioná-los, por fim quando combinados, podem produzir a solução do problema complexo inicial. Formalmente, uma árvore de decisão é um grafo acíclico direcionado, onde cada nó ou é um nó de divisão, com dois ou mais sucessores, ou um nó folha (FACELI *et al.*, 2011, p. 83).

Cada nó de divisão da árvore representa um teste em uma ou mais características dos dados e as ramificações da árvore correspondem às possíveis respostas a esse teste. As folhas da árvore representam as classes alvo.

Na Figura 1, Faceli *et al.* (2011, p. 84) representa uma árvore de decisão e a divisão correspondente no espaço definida pelos atributos x_1 e x_2, sendo a união das regiões definidas pelos nós folhas o espaço todo definido pelos atributos. Observa-se que as regiões definidas pelos nós folhas são mutuamente excludentes, e assim, qualquer interseção das regiões cobertas entre duas folhas é vazia.

Figura 1 – Uma árvore de decisão e as regiões de decisão no espaço de objetos.



Fonte: Faceli *et al.* (2011).

O algoritmo de construção de árvores seleciona recursivamente as melhores caracterís-

ticas para dividir os dados de treinamento, com o objetivo de maximizar a pureza das classes nas folhas, medida, por exemplo, pela entropia. A árvore resultante pode ser usada para fazer previsões sobre a classe de novos exemplos, percorrendo a árvore de acordo com os testes de características até alcançar uma folha, que determina a classe prevista.

O Algoritmo 1 descreve os principais passos utilizado para construir uma árvore de decisão. A função **GeraÁrvore** recebe o conjunto de dados **D** como parâmetro de entrada. No Passo 3, o algoritmo avalia o critério de parada. Se mais divisões do conjunto de dados forem necessárias, é selecionado o atributo que maximiza alguma medida de impureza, como descrito no Passo 5. Em seguida, no Passo 7, a função **GeraÁrvore** é aplicada recursivamente a cada partição do conjunto de dados **D** (FACELI *et al.*, 2011, p. 84).

Algoritmo 1 – Algoritmo para Construção de uma Árvore de Decisão

Requer: Conjunto de dados $D = \{(X_i, y_i), i = 1, ..., n\}$
Assegure: Árvore de decisão
1: **/*Função GeraÁrvore(D)*/**
2: **se** *critério de parada(***D***) = Verdadeiro* **então**
3: **retorna** : um nó folha rotulado com a constante que minimiza a função de perda
4: **fim se**
5: **para cada** partição dos exemplos D_i, baseado nos valores do atributo escolhido **faça**
6: Induz uma subárvore *Árvore = GeraÁrvore(*D_i*);*
7: **fim para**
8: **retorna** : *Árvore* contendo um nó de decisão baseado no atributo escolhido, e descendentes *Árvore*$_i$;

A **entropia** é uma medida de incerteza e, no contexto de árvore de decisão, é usada para medir a aleatoriedade da classe alvo. A característica que mais reduz a aleatoriedade da classe alvo será escolhida como nó de divisão. Cover e Thomas (2006) definem entropia pela seguinte equação:

$$H(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i) \tag{2.2}$$

Onde:

- $H(S)$ é a entropia do conjunto de dados S;
- c é o número de classes;
- p_i é a proporção de instâncias da classe i no conjunto de dados S.

Note que não são considerados os valores da variável aleatória, mas a probabilidade de ocorrência. A entropia será nula se o resultado possuir certeza e valor máximo quando os resultados forem equiprováveis.

O **ganho de informação** é um critério para mensurar como a divisão de um conjunto de dados em subconjuntos com base em uma determinada característica reduz a incerteza ou a impureza das classes nas folhas da árvore (QUINLAN, 1986). É calculado como a diferença entre a entropia do conjunto de dados original e a entropia ponderada dos subconjuntos resultantes da divisão.

$$IG(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot H(S_v) \tag{2.3}$$

Onde:

- $IG(S,A)$ é o ganho de informação ao dividir o conjunto de dados S com base na característica A;
- $H(S)$ é a entropia do conjunto de dados original;
- $Values(A)$ são os valores possíveis da característica A;
- $|S_v|$ é o número de instâncias no subconjunto S_v resultante da divisão pelo valor v da característica A;
- $|S|$ é o número total de instâncias no conjunto de dados S;
- $H(S_v)$ é a entropia do subconjunto S_v.

As árvores de decisão são modelos robustos, flexíveis, interpretáveis e eficiente, podendo ser eficazes em uma variedade de cenários. No entanto, é importante citar suas principais desvantagens como:

- Tendência ao sobreajuste: O sobreajuste ou o *overfiting* ocorre quando o modelo se ajusta excessivamente aos dados de treinamento, incluindo ruídos e não consegue generalizar para dados nunca visto. Árvores de decisão profundas são mais propensas ao *overfitting*. Podar uma árvore, que consiste em trocar nós profundos por folhas, é utilizada pra reduzir esse comportamento indesejado (FACELI *et al.*, 2011, p. 91).
- Instabilidade: Pequenas mudanças nos dados de treinamento podem resultar em grandes variações na árvore final, tornando os modelos instáveis (FACELI *et al.*, 2011, p. 96).

2.3.2 Aprendizado Ensemble

Para Hastie, Tibshirani e Friedman (2009, p. 605), a ideia do aprendizado *ensemble* é construir um modelo de previsão combinando as forças de uma coleção de modelos base mais simples para melhorar o desempenho geral na predição da classe alvo. Seu objetivo principal é mitigar erros, tendências e fraquezas que podem existir em modelos individuais, aproveitando

a inteligência coletiva do conjunto para alcançar um modelo mais robusto e geralmente mais preciso em suas predições.

No artigo de Opitz e Maclin (1999), é evidenciado que os classificadores *ensembles* tendem a ter um desempenho superior em relação aos classificadores individuais.

2.3.2.1 *Bagging ou Bootstrap Aggregating*

A técnica de *Bootstrap Aggregation*, ou simplesmente *Bagging*, introduzida por Breiman (1996), consiste em criar várias instâncias de um mesmo algoritmo de aprendizado de máquina, treinando cada uma delas em uma replica aleatória dos dados originais com reposição, o que significa que alguns exemplos podem aparecer mais de uma vez na mesma amostra. Em contrapartida, exemplos não selecionados compõem o subconjunto de teste. Esse processo de amostragem com reposição é denominado de *bootstrap*.

Faceli *et al.* (2011, p. 144) explica que para um conjunto de treinamento com n exemplos, a probabilidade de um exemplo ser selecionado é $1-(1-\frac{1}{n})^n$, e para um n grande, torna-se $1-\frac{1}{e}$, onde e é a base de logaritmos naturais. Cada amostra contém, em média, $38,6\%$ $(\frac{1}{e})$ de exemplos duplicados. Todos os classificadores são empregados para classificar cada exemplo no conjunto de teste, e a classificação final é geralmente realizada utilizando um esquema de voto uniforme. O autor detalha o funcionamento da técnica de *bagging* no Algoritmo 2.

A agregação de classificadores homogêneos, ou seja, gerados por um único algoritmo, com o processo de amostragem *bootstrap* visa melhorar a precisão e a estabilidade dos modelos preditivos pois reduz a variância de modelos que tendem a se ajustar demais aos dados, como árvore de decisão.

Algoritmo 2 – Algoritmo de *Bagging*

Requer: Um algoritmo de aprendizado ϕ
Um conjunto de treinamento $\mathbf{D} = \{(X_i, y_i), i = 1, ..., n\}$
Número de Iterações Nr
Um conjunto de teste com nt exemplos $\mathbf{T} = \{(X_j, ?), j = 1, ..., nt\}$
Assegure: Previsões para o conjunto de teste

/*Fase de aprendizado*/
para cada $l = 1$ **to** Nr **faça**
  D' $\leftarrow$ amostra com repetição de **D**
  $\hat{f}_l \leftarrow \phi$ (**D'**)
fim para
/*Fase de classificação*/
para cada $j = 1$ **to** nt **faça**
  $\hat{y}_j = argmax_{y \in Y} \sum_{l=1}^{Nr} \hat{f}_l(X_j \in T)$
fim para
retorna : Vetor de previsões $\hat{y}$

2.3.2.2 Random Forest

Desenvolvido por Breiman (2001), o método *Random Forest* (RF), ou Floresta Aleatória, representa uma extensão da técnica de bagging. Neste método, as árvores de decisão servem como os estimadores base e introduzem aleatoriedade adicional para fortalecer a diversidade e a robustez do conjunto.

O processo se inicia com a criação de subconjuntos aleatórios com reposição (*bootstrap*) a partir dos conjuntos de dados originais, permitindo a inclusão de múltiplas instâncias do mesmo ponto de dado.

Durante a construção de cada árvore de decisão, apenas um conjunto aleatório de características é considerado em cada nó para determinar a melhor divisão. Essa seleção aleatória assegura que diferentes árvores se concentrem em aspectos distintos dos dados, introduzindo variações no conjunto.

Cada subconjunto resulta na construção independente de um modelo de árvore de decisão. Estas árvores individuais capturam padrões e relações distintas nos dados, fazendo uso do conjunto específico de características disponíveis em cada nó.

Para problemas de classificação binária, o RF faz previsões por meio de votação majoritária das saídas das árvores individuais. Cada árvore emite sua própria previsão e a classe mais votada é selecionada como a previsão final. Esta agregação capitaliza a sabedoria coletiva dos modelos, integrando perspectivas diversas para aprimorar a precisão global e a capacidade de generalização do modelo ensemble.

2.3.2.3 Gradient Boosting

O algoritmo de *Gradient Boosting* combina de maneira iterativa modelos de aprendizado base ou fracos, geralmente árvores de decisão, em um único modelo de aprendizagem forte. O objetivo é melhorar iterativamente a precisão do modelo aprendendo com os erros cometidos em iterações anteriores (FRIEDMAN, 2001).

Inicialmente, um modelo base é treinado no conjunto de dados e suas previsões são comparadas com os valores reais alvo. As diferenças entre as previsões e os valores verdadeiros são os erros residuais.

Em cada iteração, um novo modelo base é treinado para minimizar o erro residual do modelo anterior. Entretanto, em vez de usar os valores alvo originais como rótulos, o novo modelo é treinado com base nos erros cometidos pelos modelos do conjunto até o momento. Essa abordagem permite que o novo modelo se concentre nas áreas onde os modelos anteriores tiveram desempenho inferior e seja altamente correlacionados com o gradiente negativo da função de perda associada a todo o conjunto de modelos.

As previsões do novo modelo são então adicionadas ao conjunto e o processo é repetido

até que o erro residual não possa ser mais reduzido ou até que um critério de parada pré-definido seja atingido. Ao adicionar iterativamente novos modelos, o conjunto melhora gradualmente seu desempenho preditivo.

O Algoritmo 3 é uma adaptação genérica do pseudocódigo de *Gradient Boosting* apresentado por Friedman (2001, p. 5), onde L é a função de perda, $F_m(x)$ é o modelo preditivo no estágio m, γ é a taxa de aprendizado que controla a contribuição de cada modelo base e $h_m(x)$ é o modelo base ajustado aos resíduos no estágio m. O algoritmo itera sobre M estágios (árvores), atualizando o modelo preditivo a cada passo.

Algoritmo 3 – Algoritmo de *Gradient Boosting*

Requer: Um conjunto de treinamento $\mathbf{D} = \{(X_i, y_i), i = 1, ..., n\}$
Uma função de perda diferenciável $L(y, F(x))$
Número de iterações M (número de árvores)

1: Inicialize o modelo com um valor constante:
$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$
2: **para cada** $m = 1$ **to** M **faça**
3: Calcule os resíduos:
$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \ldots, n$
4: Ajuste um modelo de aprendizagem básico (ou fraco, tal como uma árvore) aos resíduos $h_m(x)$, isto é, treine-o utilizando o conjunto de treino $\{(X_i, r_{im}), i = 1, ..., n\}$
5: Calcule o multiplicador γ_m minimizando a função de perda:
$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$
6: Atualize o modelo: $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$
7: **fim para**
8: **retorna** : $F_m(x)$

2.4 Medidas de Desempenho

Medidas de desempenho para classificação binária em aprendizado de máquina são utilizadas para avaliar quão bem um modelo de classificação é capaz de distinguir entre duas classes distintas, geralmente chamadas de classe positiva (+) e classe negativa (-). A matriz de confusão é uma tabela que permite a visualização das frequências de classificação para cada classe do modelo em comparação com os valores reais dos dados. A matriz de confusão, representada no Quadro 2, é organizada da seguinte forma (FACELI *et al.*, 2011):

- Verdadeiros Positivos (VP) são os casos em que o modelo previu corretamente a classe positiva. Isso significa que o modelo acertou quando a instância realmente pertencia à classe positiva.
- Falsos Positivos (FP) representam os casos em que o modelo previu erroneamente a classe positiva quando a instância pertencia à classe negativa. Em outras palavras, o modelo

cometeu um erro ao classificar a instância como positiva.

- Falsos Negativos (FN) são os casos em que o modelo previu erroneamente a classe negativa quando a instância pertencia à classe positiva. O modelo errou ao não reconhecer a classe positiva.
- Verdadeiros Negativos (VN) são os casos em que o modelo previu corretamente a classe negativa. O modelo acertou ao identificar a instância como pertencente à classe negativa.

Quadro 2 – Matriz de Confusão

		Valores Preditos	
		+	-
Valores Reais	+	VP	FN
	-	FP	VN

Com base na matriz de confusão, várias medidas de desempenho podem ser derivadas para avaliar a qualidade do modelo (MONARD; BARANAUSKAS, 2003):

- Acurácia: A acurácia (Acc) é a proporção de todas as previsões corretas em relação ao total de amostras. Ela fornece uma medida geral do quão bem o modelo está se saindo.

$$Acc = \frac{VP+VN}{VP+VN+FP+FN} \tag{2.4}$$

- Taxa de erro total (Err): A taxa de todas as previsões incorretas em relação ao total de amostras.

$$Err = \frac{FP+FN}{VP+VN+FP+FN} \tag{2.5}$$

- Precisão: A precisão mede a proporção de verdadeiros positivos em relação ao total de previsões positivas. É útil quando o foco está na minimização de falsos positivos.

$$Precisão = \frac{VP}{VP+FP} \tag{2.6}$$

- Sensibilidade: A Sensibilidade (*recall)* mede a proporção de verdadeiros positivos em relação ao total de casos reais positivos. É útil quando o objetivo é minimizar falsos negativos. Também conhecida como Taxa de Verdadeiros Positivos (TVP).

$$Sensibilidade = \frac{VP}{VP+FN} \tag{2.7}$$

- F1-Score: Média harmônica da precisão e da revocação, fornecendo uma métrica que combina ambas as medidas. É útil quando o equilíbrio entre a precisão e a sensibilidade é importante.

$$F1-Score = \frac{2*(Precisão*Sensibilidade)}{Precisão+Sensibilidade} \tag{2.8}$$

- Especificidade: A especificidade mede a proporção de verdadeiros negativos em relação ao total de casos reais negativos. É útil quando se deseja avaliar o desempenho na detecção da classe negativa.

$$Especificidade = \frac{VN}{VN + FP} \tag{2.9}$$

- Taxa de Falsos Positivos (TFP): Essa taxa mede a proporção de falsos positivos em relação ao total de casos reais negativos. Ela é complementar à especificidade.

$$TFP = \frac{FP}{FP + VN} \tag{2.10}$$

Os modelos frequentemente produzem como resultado uma probabilidade estimada que varia entre 0 e 1, entretanto, para a construção da matriz de confusão, é necessário transformar essas probabilidades em previsões binárias. Importante destacar que um único modelo de classificação pode estar associado a uma variedade de matrizes de confusão. Tal diversidade decorre do fato de que a determinação de uma previsão como 0 ou 1 depende da seleção do ponto de corte (*threshold*) aplicado à probabilidade estimada pelo modelo.

2.5 Análise ROC

A análise *Receiver Operating Characteristic* (ROC) é uma técnica amplamente utilizada para avaliar a eficácia de modelos de classificação binária (FACELI *et al.*, 2011). A curva ROC é uma representação gráfica da análise ROC que ilustra o desempenho de um modelo de classificação à medida que o *threshold* varia. Essa curva traça a taxa de verdadeiros positivos (TVP) em relação à taxa de falsos positivos (TFP) para diferentes valores de *threshold*.

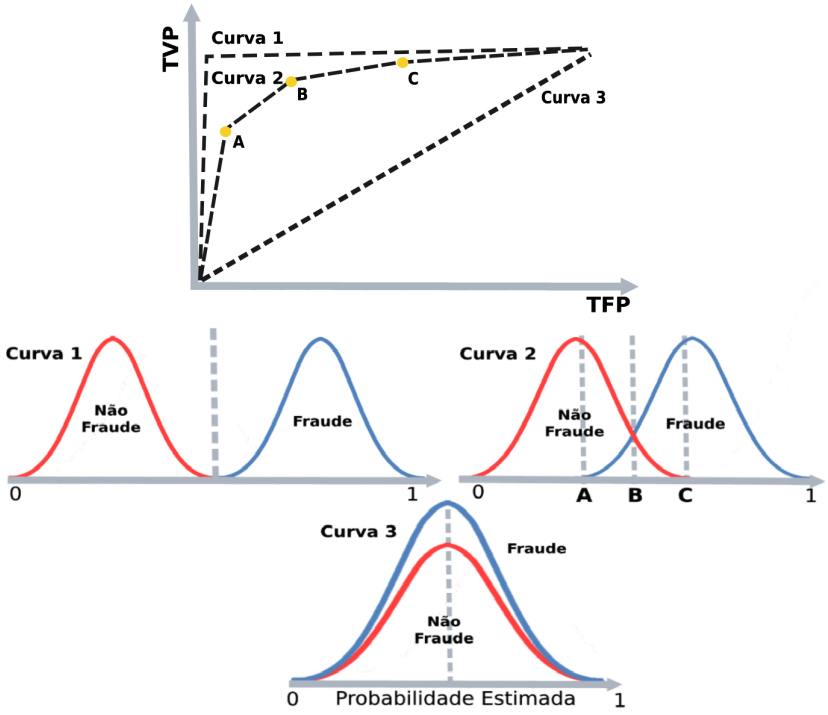
A curva ROC é construída ao variar o *threshold* e registrar os pares TVP e TFP, plotando-os em um gráfico. Essa curva é especialmente útil para avaliar o desempenho do modelo quando o equilíbrio entre precisão e sensibilidade é importante.

A Figura 2 representa alguns exemplos hipotéticos de curvas ROC.

- Curva 1 – Modelo perfeito: a medida que o threshold vai diminuindo o modelo nunca comete um falso positivo e a taxa de verdadeiro positivo está sempre em 100%.
- Curva 2 – Modelo usual: comete alguns falsos positivos e falsos negativos.
- Curva 3 – Modelo ineficiente: um modelo que não agrega nada a mais em relação a chutes aleatórios.

Area Under the Curve (AUC) é uma métrica resumida da curva ROC que fornece uma única pontuação para o desempenho do modelo. A AUC mede a área sob a curva ROC, variando de 0.5 a 1. Quanto maior a AUC, melhor o desempenho do modelo. A Figura 3 representa graficamente possíveis áreas de curvas ROC.

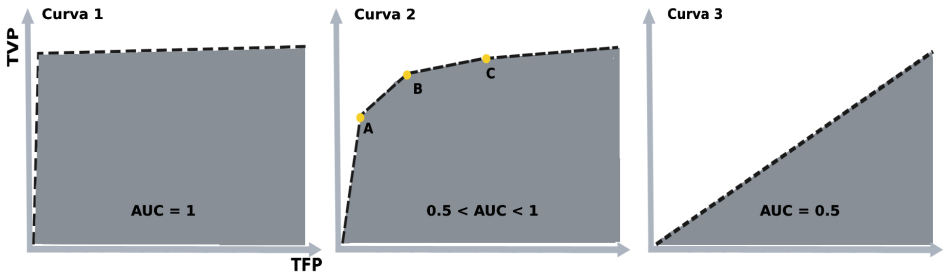
Figura 2 – Curva ROC



Fonte: Elaborada pelo autor.

- AUC = 1: A curva 1 indica um modelo perfeito capaz de separar as classes sem erros.
- 0.5 < AUC < 1: A curva 2 indica que o modelo é capaz de distinguir entre as classes melhor do que um classificador aleatório. Quanto mais próximo de 1, melhor o desempenho do modelo.
- AUC = 0.5: Por fim na curva 3, o modelo tem desempenho similar ao de uma escolha aleatória, ou seja, não é melhor do que um classificador aleatório.

Figura 3 – AUC



Fonte: Elaborada pelo autor.

Em resumo, a análise da curva ROC e a métrica AUC, fornecem uma visão abrangente do desempenho de modelos de classificação em problemas de classificação binária, permitindo a adaptação do ponto de corte de acordo com as necessidades específicas do problema.

2.6 Interpretação de Modelos Preditivos

As decisões dos modelos preditivos têm impacto direto e substancial sobre indivíduos e sociedade, por isso a interpretabilidade desempenha um papel fundamental ao assegurar a equidade, confiabilidade, transparência e responsabilidade dos modelos principalmente em áreas como a justiça, saúde, segurança, finanças e meio ambiente. Além disso, a interpretabilidade ajuda os desenvolvedores a identificar erros e falhas nos modelos, possibilitando ajustes e melhorias mais eficazes.

Miller (2019) define interpretabilidade como o grau em que um humano pode entender a causa de uma decisão em um modelo.

SHapley Additive exPlanations (SHAP) foi implementada por Lundberg e Lee (2017), e seus métodos tem sido amplamente utilizado para explicar a saída de modelos de aprendizado de máquina em termos de importância de atributos. O objetivo principal do método SHAP é tornar os modelos de ML mais transparentes e interpretáveis, permitindo que os usuários compreendam como os atributos contribuem para as previsões do modelo.

O SHAP baseia-se na teoria dos jogos cooperativos e, mais especificamente, no **valor de Shapley** introduzida por Shapley (1953). O valor de Shapley é um conceito da teoria dos jogos cooperativos que atribui um valor de pagamento justo a cada participante (jogador) em uma disputa (coalisão), com base na sua contribuição individual:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{2.11}$$

Onde:

- N é o conjunto de todos os atributos;
- S é um subconjunto de N que não inclui o atributo i;
- f é a função que estamos tentando explicar;
- $\phi_i(f)$ é a contribuição do atributo i para a previsão.

Esta fórmula calcula a contribuição média do atributo i para todas as combinações possíveis de atributos. A ideia é que, ao somar todas essas contribuições para todos os atributos, obtemos a previsão original da função f.

Molnar (2022) enfatiza que o valor de Shapley é o único método de atribuição que satisfaz as propriedades de eficiência, simetria, *dummy* e aditividade, que juntas podem ser consideradas uma definição de pagamento justo.

- Eficiência: O valor total atribuído aos jogadores é igual ao valor total do jogo. Isso significa que não há perda de valor na atribuição.

- Simetria: Jogadores que contribuem igualmente para o jogo recebem o mesmo valor. Garantindo que a atribuição seja justa e não favoreça nenhum jogador em particular.
- *Dummy*: Um jogador que não contribui em nada para o jogo recebe um valor zero.
- Aditividade: O valor que um jogador recebe por participar de um jogo é igual à soma dos valores que ele recebe por participar de cada subjogo. Essa propriedade garante que, para um valor de característica, você pode calcular o valor de Shapley para cada árvore individualmente dentro de uma floresta aleatória, média-los e obter o valor de Shapley para o valor de característica para a floresta aleatória.

O SHAP é um método de atribuição de características aditivas explicada pela equação linear de variáveis binárias:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j \tag{2.12}$$

onde g é o modelo de explicação, z' é um vetor binário que indica se a classe foi prevista ou não, M é o vetor da coalizão, ϕ_j são os valores de atribuição das características para um atributo j e ϕ_0 é o valor inicial da coalizão.

O SHAP atribui valores a cada atributo de entrada, mostrando o impacto de cada atributo na predição do modelo. Isso permite entender não apenas a importância de cada atributo individualmente, mas também como a presença ou ausência de diferentes atributos afetam as predições.

CAPÍTULO

3

MATERIAIS E MÉTODOS

Neste capítulo, será apresentada a metodologia para a construção de um projeto completo de aprendizado de máquina. Serão aplicados os conceitos de análise exploratória de dados e técnicas de aprendizado de máquina vistos no capítulo anterior para o pré-processamento dos dados e treinamento do modelo.

Construir um projeto de aprendizado de máquina é um processo complexo que envolve uma série de etapas fundamentais:

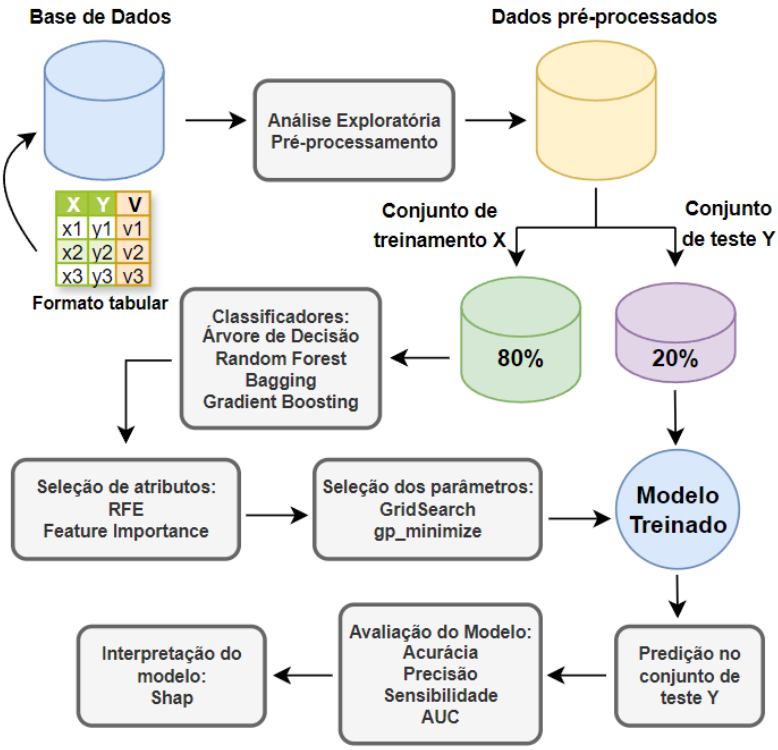
- Formulação do problema: Entender claramente o problema a ser resolvido. Definir objetivos e compreender como a aplicação do aprendizado de máquina pode oferecer soluções. Este passo é crucial para direcionar todo o processo.
- Aquisição de dados: Essa fase envolve o desenvolvimento de programas auxiliares para acessar os diferentes bancos de dados da organização e executar consultas para extrair uma grande quantidade de dados, garantindo que o funcionamento do sistema produtivo não seja comprometido.
- Análise exploratória de dados: Processo de descoberta, extração e transformação de dados brutos em informações úteis e não triviais de grandes conjuntos de dados.
- Seleção e treinamento dos modelos: Fase de seleção e treinamento dos modelos nos dados pré-processados. Isso pode envolver a criação de conjuntos de treinamento e teste, ajuste de hiperparâmetros e seleção de atributos mais importantes do modelo para alcançar o melhor desempenho.
- Avaliação do classificador: Avaliar o desempenho do classificador com métricas como precisão, acurácia, F1-score, curva ROC, entre outras. Também é importante realizar validações cruzadas para garantir que o modelo seja robusto e generalize bem para dados não vistos.

- Interpretação: Métodos como SHAP permitem interpretar e explicar as previsões do modelo, fornecendo *insights* sobre quais atributos são mais importantes para as decisões do modelo.

- Implantação e Monitoramento: Após a avaliação, o modelo precisa ser implantado em um ambiente de produção. É importante monitorar seu desempenho em tempo real e realizar ajustes conforme necessário para garantir que ele continue sendo eficaz ao longo do tempo.

Importante no decorrer do desenvolvimento do projeto de ML revisitar todos os passos anteriores para aprimorar novas ideias, análises e entender se a entrega do projeto está convergindo para a solução final. A Figura 4 apresenta o roteiro de desenvolvimento da solução de ML proposta para este projeto.

Figura 4 – Roteiro de desenvolvimento do projeto de ML



Fonte: Elaborada pelo autor.

Foram utilizadas no desenvolvimento do projeto a linguagem de programação Python Core Team (2019) e as bibliotecas Numpy (HARRIS *et al.*, 2020), Scikit-Learn (PEDREGOSA *et al.*, 2011), NLTK (BIRD STEVEN; KLEIN, 2009) e Pandas (MCKINNEY, 2010). Essas ferramentas oferecem uma ampla seleção de recursos eficientes para a construção de projetos de aprendizado de máquina, abrangendo desde modelagem estatística até a avaliação de desempenho de modelos.

3.1 Conjunto de Dados

A construção da base de dados envolveu múltiplas reuniões com especialistas da área e administradores de banco de dados para compreender quais tabelas seriam pertinentes para a análise. Em diversos estágios desse estudo, foi necessário revisitar o banco de dados para incluir mais tabelas contendo informações adicionais. O conjunto de dados inicial resultou da união de 8 bases de dados estruturados de um cliente, totalizando 273460 registros com 142 atributos. Para preservar o anonimato, este cliente será denominada DespesasX.

Uma gestão eficaz de despesas demanda processos rigorosos, como mencionado no Capítulo 1. Nesse contexto, uma solução de gestão de despesas deve considerar diversos perfis empresariais, incluindo o porte, cultura organizacional e outros fatores relevantes. A empresa DespesasX opera com múltiplas filiais internacionalmente, possuindo centros de custo em diferentes moedas e centros de conferência para a análise e aprovação de solicitações de despesas. Cada solicitação pode conter várias despesas associadas a um solicitante, este pode ser um funcionário, gerente ou um departamento específico. Cada despesa contém informações relevantes sobre sua categoria, juntamente com atributos de identificação, localização e horário.

Um especialista em auditoria de despesas contribuiu com sua experiência para criar três novos atributos. Esses atributos visam identificar se as despesas têm valores inteiros, se são múltiplos de cinco e identificar o gênero dos envolvidos pelo primeiro nome, distinguindo entre masculino e feminino. Além disso, durante a fase de análise exploratória de dados, outros atributos foram adicionados.

Foram excluídos atributos contendo informações sensíveis, assim como aqueles em que a maioria dos registros continha valores nulos. A base de dados foi filtrada para incluir apenas registros que apresentavam valores de despesas.

3.2 Análise Exploratória de Dados

A análise exploratória ajuda a compreender a natureza e a estrutura dos dados (Seção 2.1). Isso inclui a identificação de tipos de variáveis, distribuições, valores ausentes, duplicados, incompletos e inconsistentes.

Os autores Brownlee (2016) e Shixin (2020) forneceram diretrizes abrangentes sobre AED, abordando tanto aspectos intuitivos, por meio de visualização, quanto aspectos mais rigorosos, com base em análise estatística. Esses autores apresentam abordagens práticas e exemplos utilizando a linguagem de programação *Python*, o que facilita a compreensão e implementação das técnicas de análise de dados e seleção de atributos.

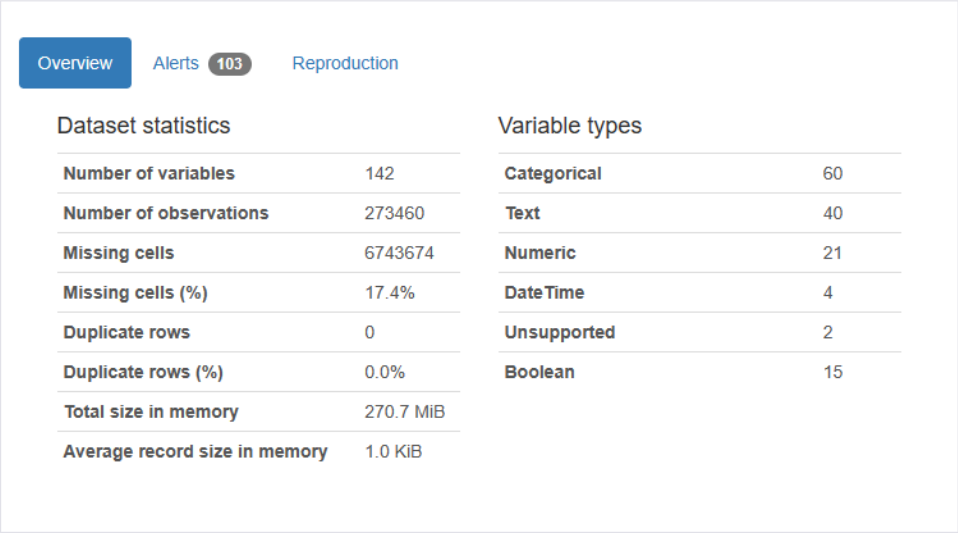
3.2.1 Caracterização dos Dados

Essa etapa consiste em descrever, analisar e interpretar as características fundamentais de um conjunto de dados, explorando seus tipos e escalas.

Com a biblioteca ydata-profiling de Brugman (2019), é possível gerar automaticamente um relatório detalhado da análise exploratória de dados. Esse relatório inclui estatísticas descritivas para cada coluna, como média, mediana, desvio padrão, valores mínimos e máximos, entre outros. Além disso, o relatório fornece informações sobre a presença de valores ausentes, distribuições de valores e correlações entre variáveis. Essa ferramenta é extremamente útil para entender a composição do conjunto de dados e identificar padrões e tendências importantes em uma interface gráfica intuitiva.

A Figura 5 fornece uma visão geral da análise estatística descritiva do conjunto de dados, obtida a partir do relatório da biblioteca ydata-profiling.

Figura 5 – Resumo estatístico do conjunto de dados



Fonte: Elaborada pelo autor.

3.2.2 Exploração dos Dados

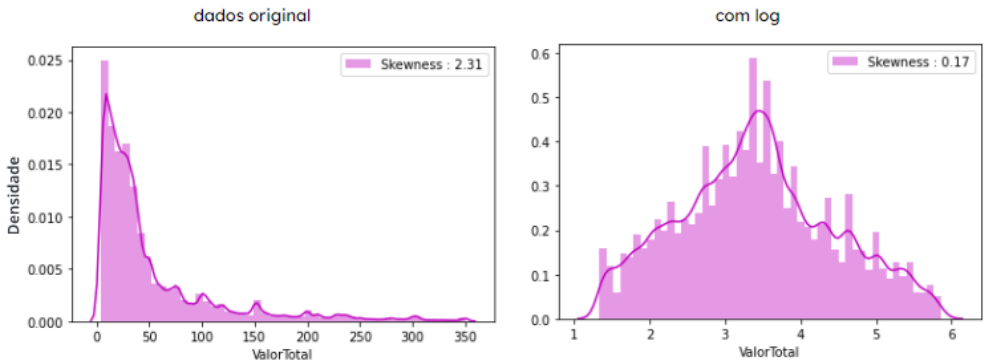
Análise de Obliquidade

Segundo Faceli *et al.* (2011), a obliquidade mede a assimetria da distribuição dos dados em torno da média. A aplicação de uma função logarítmica pode ser benéfica em situações em que os dados apresentam alta assimetria ou uma distribuição com cauda longa. Ao aplicar o logaritmo aos valores, ocorre uma compressão na escala, o que pode tornar a relação entre as variáveis mais linear e aprimorar o desempenho de determinados modelos.

Na Figura 6, o gráfico à direita ilustra uma distribuição assimétrica do atributo *ValorTotal*, enquanto o gráfico à esquerda representa a distribuição com simetria após a transformação

logarítmica.

Figura 6 – Análise de obliquidade

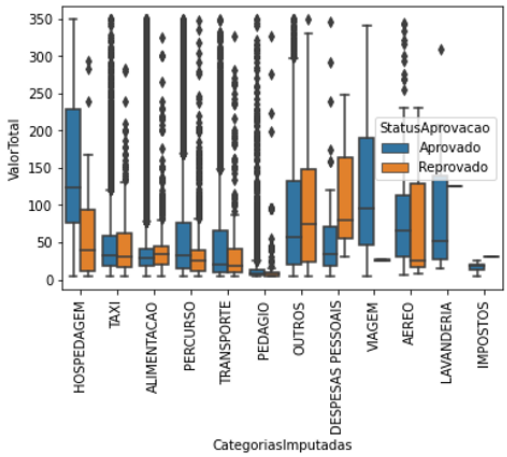


Fonte: Elaborada pelo autor.

Análises boxplot

Na Figura 7 é possível identificar correlação entre a categoria da despesa e o valor total da despesa. Do mesmo modo, o valor influencia o *status* de aprovação de uma nova despesa dentro de uma categoria.

Figura 7 – Análise Boxplot 1



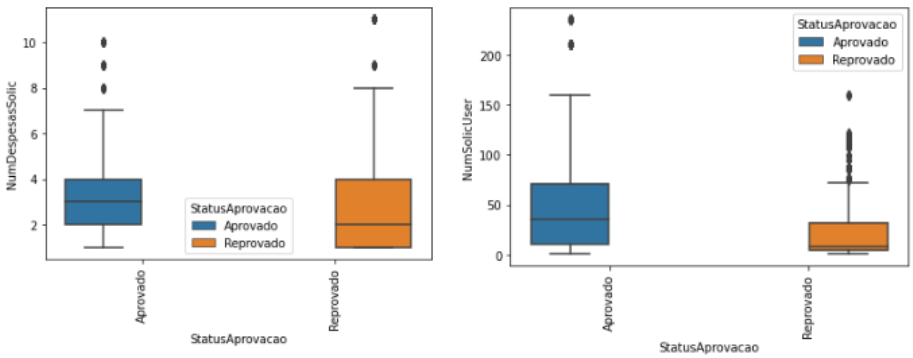
Fonte: Elaborada pelo autor.

De um modo geral, ilustrada pelo gráfico a direita da Figura 8, menos despesas por solicitação indicam possível reprovação. Da mesma forma, analisando o gráfico esquerdo na Figura 8, usuários com menos solicitações de reembolso têm maior chance de terem registros reprovados. No entanto, esse comportamento varia conforme a categoria da despesa e o cargo do viajante (Figura 9).

Extração de Características

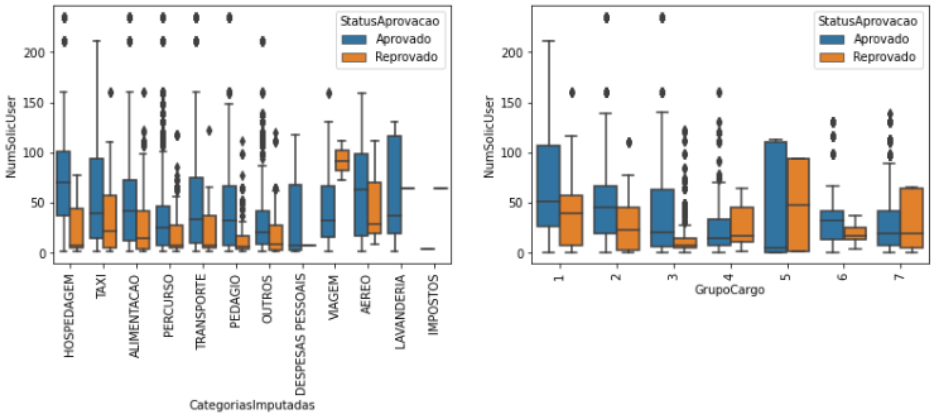
A extração de características cria novos atributos a partir dos dados originais com o objetivo de enriquecer o conjunto de dados, proporcionando uma representação mais rica dos

Figura 8 – Análise boxplot 2



Fonte: Elaborada pelo autor.

Figura 9 – Análise boxplot 3



Fonte: Elaborada pelo autor.

padrões e relações subjacentes nos dados. Isso não só melhora o desempenho dos modelos, mas também aumenta sua interpretabilidade, permitindo uma compreensão mais clara dos fatores que influenciam as previsões.

A geração de novos atributos pode compreender tanto a agregação e sumarização de informações quanto a criação de variáveis com base no conhecimento do domínio ou na intuição sobre o problema em questão.

A primeira abordagem envolve a computação de estatísticas descritivas, como média, mediana e desvio padrão, para diferentes grupos de dados ou períodos de tempo. Já a segunda estratégia consiste em criar novas variáveis combinando atributos existentes ou introduzindo indicadores ou flags para representar condições específicas.

Ao criar atributos que capturam as características mais relevantes e informativas dos dados, é possível reduzir a quantidade de variáveis necessárias para descrever o problema, sem comprometer a qualidade das previsões.

Os atributos criados a partir dos dados originais foram:

- *SolicitanteViajante*: Indica se o solicitante da despesa e o viajante são a mesma pessoa.
- *MesmoSolAprov*: se o solicitante do reembolso é o aprovador da solicitação.
- *NumDespesasSolic*: Quantidade de despesas para cada solicitação.
- *NumSolicUser*: Quantidade total de solicitações feitas pelo usuário.
- *NumDespesasUser*: Quantidade total de despesas feitas pelo usuário.
- *NumAprovadores*: Quantidade de aprovadores de determinada prestação de contas.
- *StatusAprovado*: Quantidade de aprovações de determinada prestação de contas.
- *StatusReprovado*: Quantidade de reprovações de determinada prestação de contas.
- *MesmoAprovador*: Indica se o aprovador é a pessoa reembolsada.
- *IntervaloSolicitante*: Intervalo de tempo em dias entre a solicitação de reembolso e data da despesa.
- *GrupoCargo*: Cargo do reembolsado.
- *ValorInteiro*: Indica se o valor da despesa é um valor inteiro.
- *ValorMultiplo5*: Indica se o valor da despesa é múltiplo de 5.

No entanto, é importante conduzir uma análise cuidadosa e validar os novos atributos gerados para garantir que eles realmente contribuam para a capacidade preditiva do modelo. Além disso, é essencial considerar o impacto da dimensionalidade dos dados e evitar a introdução de atributos redundantes ou irrelevantes.

Matriz de Correlação

A matriz de Correlação, mapeia as relações entre as variáveis númericas em um conjunto de dados. Cada célula da matriz representa o coeficiente de correlação entre duas variáveis específicas, indicando a força e a direção da associação entre elas (Subsubseção 2.1.1.2).

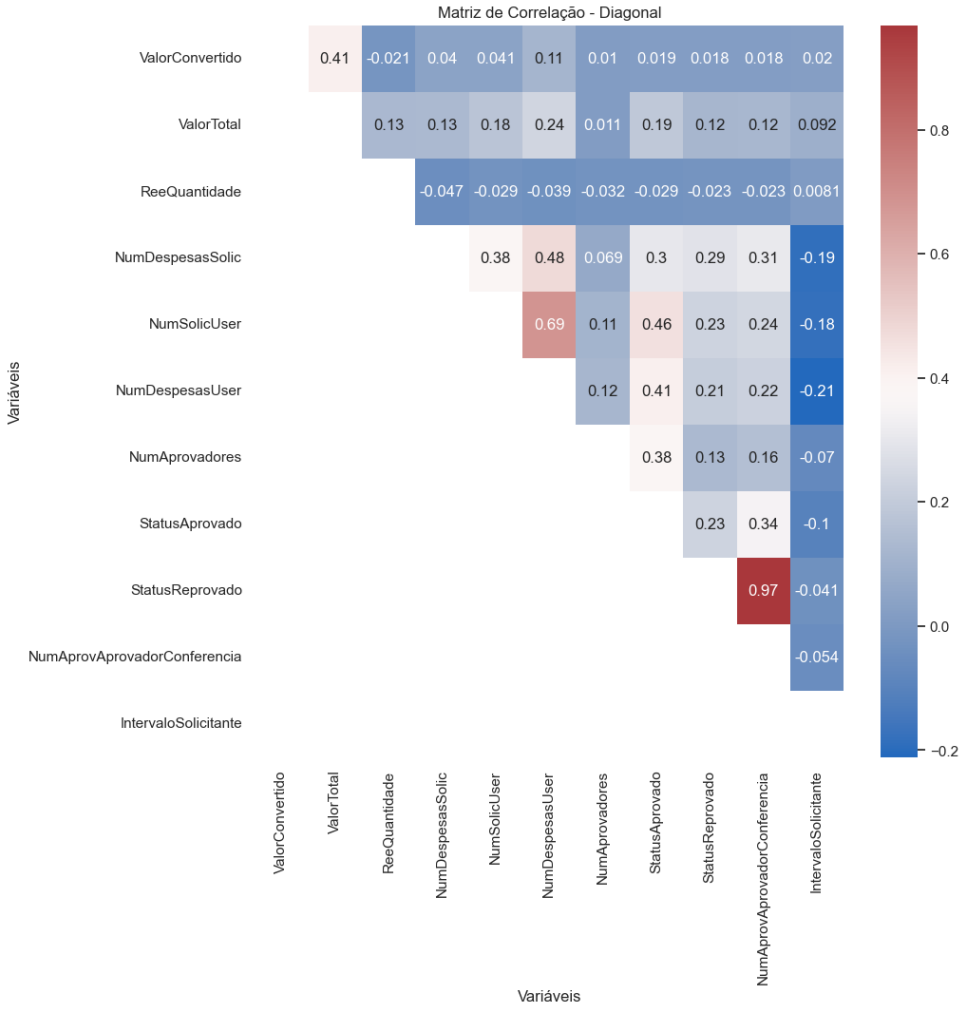
A Figura 10 revela variáveis redundantes ou altamente correlacionadas, permitindo a eliminação de colunas irrelevantes, otimizando o conjunto de dados.

3.3 Pré-processamento de Dados

3.3.1 Transformação de dados

Transformando o Atributo *"Categoria de Despesas"*

Figura 10 – Matriz de correlação



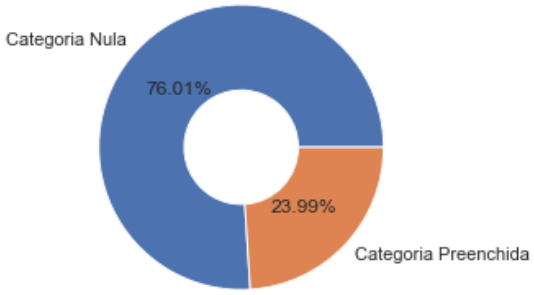
Fonte: Elaborada pelo autor.

Durante a análise da base de dados em questão, identificou-se que o atributo referente a categoria de despesa encontrava-se frequentemente não preenchido (Figura 11). No entanto, observou-se que atributos com o descritivo de despesas e o nome da despesa poderiam conter informações relevantes para categorizar as despesas. Diante disso, foi realizado um trabalho para preencher o atributo de categoria de despesas utilizando os dados desses dois atributos mencionados.

Na Figura 12, é possível observar a distribuição do número de despesas por categoria. Notavelmente, as categorias "Percurso", "Alimentação", "Transporte", "Pedágio" e "Outros" se destacam, apresentando um número significativo de observações entre as cinco categorias mais frequentes.

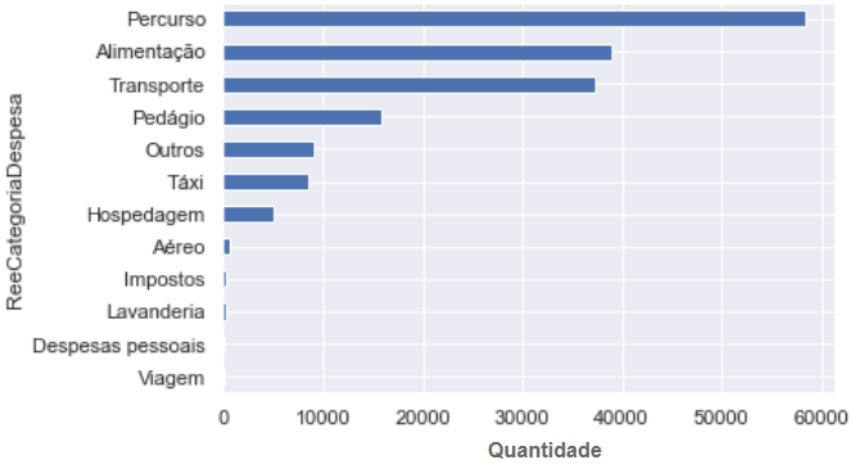
Para alcançar essa tarefa, foram empregadas técnicas de NLP, utilizando a biblioteca NLTK. Esta biblioteca oferece um conjunto diversificado de funcionalidades, permitindo manipular e processar textos de maneira eficiente. Entre as funções utilizadas destacam-se:

Figura 11 – Porcentagem de valores preenchidos na categoria de despesas



Fonte: Elaborada pelo autor.

Figura 12 – Quantidade de despesas por categoria



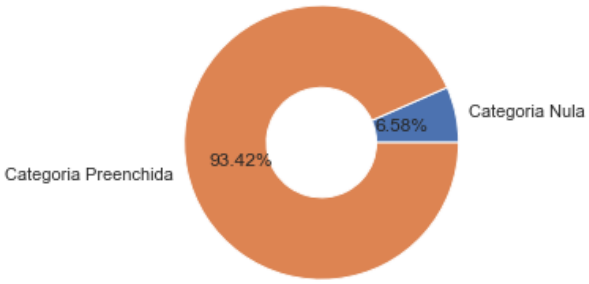
Fonte: Elaborada pelo autor.

- **RSLPStemmer**: Este módulo é responsável por realizar a stemming, isto é, reduzir palavras à sua forma raiz, o que é essencial para normalizar o texto e extrair seu significado essencial.
- **word_tokenize**: Utilizado para dividir o texto em *tokens*, facilitando a análise individual de palavras.
- **RegexpTokenizer**: Função utlizada para segmentar o texto com base em expressões regulares, permitindo uma segmentação mais flexível e adaptável ao padrão desejado.

Essas técnicas permitiram um processamento eficaz do descritivo e do nome da despesa, possibilitando a identificação e o preenchimento correto da categoria de despesas ausente (Figura 13), contribuindo assim para a melhoria da qualidade dos dados e para uma análise mais precisa.

Após o uso das técnicas de NLP e a correta categorização das despesas, houve uma mudança expressiva nas categorias mais frequentes. Em uma ordenação decrescente como observado na Figura 14, as cinco posições mais utilizadas em despesas corporativas passaram a ser

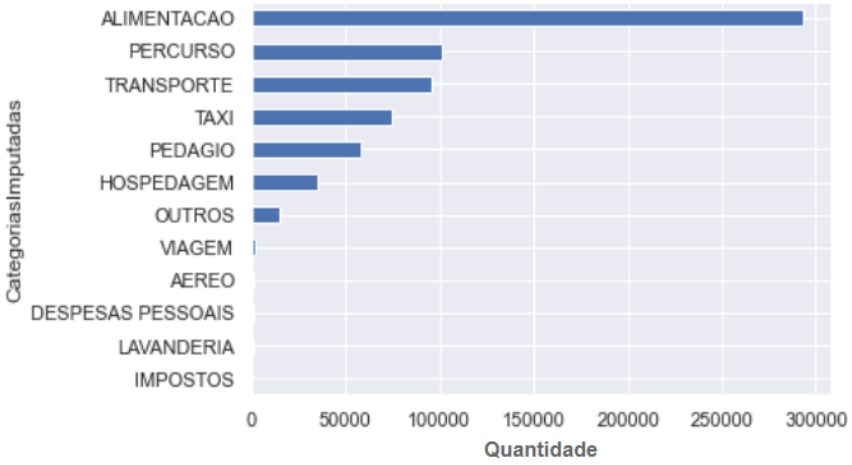
Figura 13 – Porcentagem de valores preenchidos na categoria de despesas depois da NLP



Fonte: Elaborada pelo autor.

ocupadas por "Alimentação", "Percurso", "Transporte", "Taxi" e "Pedágio". Essa reorganização evidencia uma significativa alteração nas prioridades de gastos corporativos após o processo de categorização aprimorada.

Figura 14 – Quantidade de despesas por categoria depois da NLP



Fonte: Elaborada pelo autor.

A Figura 15 exibe as diferentes categorias de despesas em relação à quantidade total de gastos e à média de valor por despesa. Observa-se uma variação significativa entre as categorias: algumas possuem uma frequência elevada na base de dados, porém apresentam um baixo valor médio gasto por despesa, enquanto outras têm uma frequência menor, porém com valores médios de gastos por despesa mais elevados.

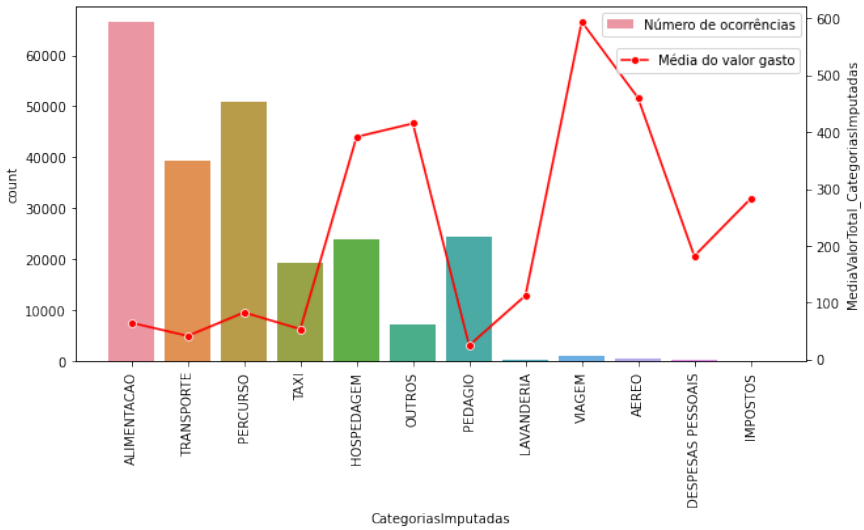
Um atributo importante criado a partir dessa transformação foi:

- *MediaValorTotal_CategoriasImputadas*: Média de gastos por categoria de despesa.

Transformando Atributo "Nome" em "Gênero"

Transformar nome de usuário em seu gênero é uma técnica comumente utilizada para criar novos atributos e extrair características relevantes em projetos de machine learning. Embora

Figura 15 – Quantidade total de despesas e média do valor gasto por tipo de despesa



Fonte: Elaborada pelo autor.

os nomes de usuário em si possam não ter valor direto para o modelo, o gênero associado a esses nomes pode fornecer informações úteis sobre os usuários e ajudar a enriquecer o conjunto de dados.

Os atributos criados a partir dessa transformação foram:

- *SexoViajante*: Classificação de gênero do viajante. Valor "M", para masculino e valor "F" para feminino.
- *SexoAprovador*: Idem ao item anterior mas para o aprovador.
- *SexoSolicitante*: Idem ao item anterior mas para o solicitante. A Figura 16 fornece a análise descritiva desse novo atributo.

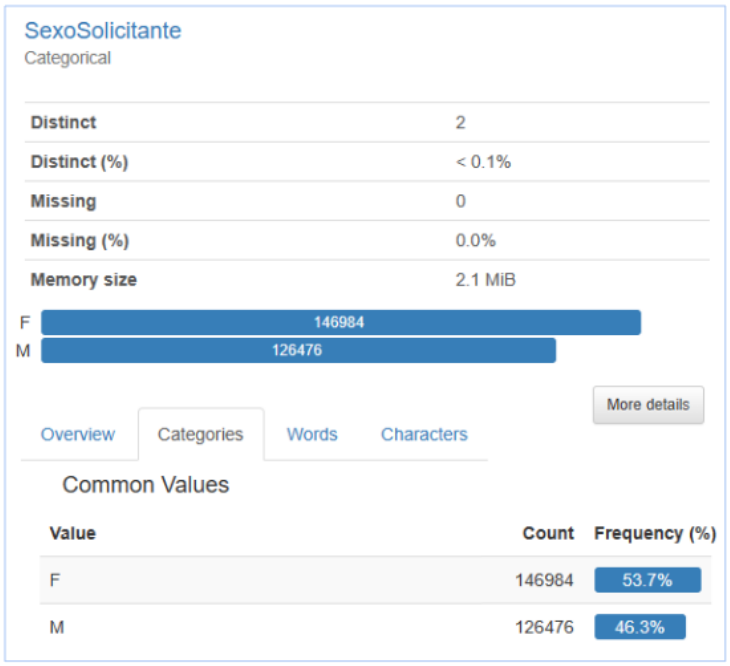
Categorização de Atributos Categóricos

O método *get_dummies* da *scikit-learn* cria colunas separadas para cada categoria presente na variável categórica original e atribui valores binários (0 ou 1) para indicar a presença ou ausência de cada categoria em uma observação específica.

Por exemplo, para a variável categórica chamada *"SexoViajante"* , com as categorias "M" e "F", ao aplicar o método duas novas colunas são criadas: *"SexoViajante_M"* e *"SexoViajante_F"*. Para cada observação no conjunto de dados, a coluna correspondente ao gênero presente recebe o valor 1, enquanto a outra coluna o valor 0.

Essa abordagem permite que algoritmos de machine learning processem e interpretem facilmente variáveis categóricas, sem a necessidade de atribuir arbitrariamente valores numéricos às categorias, o que poderia introduzir uma noção de ordem ou hierarquia onde não existe.

Figura 16 – Análise descritiva do atributo "SexoSolicitante"



Fonte: Elaborada pelo autor.

Padronização de Dados

A padronização visa transformar os dados em uma escala comum. Foi utilizado o algoritmo *StandardScaler* da biblioteca *scikit-learn*. Esse algoritmo transforma cada variável numérica em uma nova com média zero e desvio padrão igual a um. Ao padronizar as variáveis, o *StandardScaler* diminui o impacto de outliers e garante que nenhuma variável domine o modelo, reduzindo o viés.

3.3.2 Limpeza de dados

Os conjuntos de dados frequentemente enfrentam desafios em relação à qualidade dos dados, incluindo informações ruidosas (com erros ou valores inesperados), registros redundantes (com valores duplicados) e campos incompletos (com ausência de valores). Identificar e corrigir essas imperfeições é fundamental para garantir a precisão e a confiabilidade dos dados utilizados nos modelos preditivos.

Nesta etapa, a base de dados foi reduzida em 20% do total de registros e o número de colunas foi reduzido de 131 para 42 colunas. Esse processo gerou um conjunto de dados mais enxuto, eficiente e interpretável, sem comprometer a representatividade dos dados originais.

Atributos com desvio padrão igual a zero

Se o desvio padrão de uma coluna é zero, significa que todos os seus valores são iguais. Essa coluna não fornece nenhuma informação útil para o modelo de ML, pois não há variação nos dados. Portanto, não contribui para a capacidade do modelo de fazer previsões precisas.

Dados Inconsitentes

Na detecção de anomalias nos dados, o uso de expressões regulares, conhecida como *regex*, é uma prática comum para identificar valores que não estejam em conformidade com os padrões esperados. Esse método é particularmente eficaz em situações em que os dados sofrem com erros de entrada ou possuem formatação inconsistente. No caso dos atributos do tipo *float*, a expressão regular `^[0-9]+\.?[0-9]+$`, foi aplicada para verificar se os valores correspondiam apenas a números inteiros ou decimais.

Removendo *outliers*

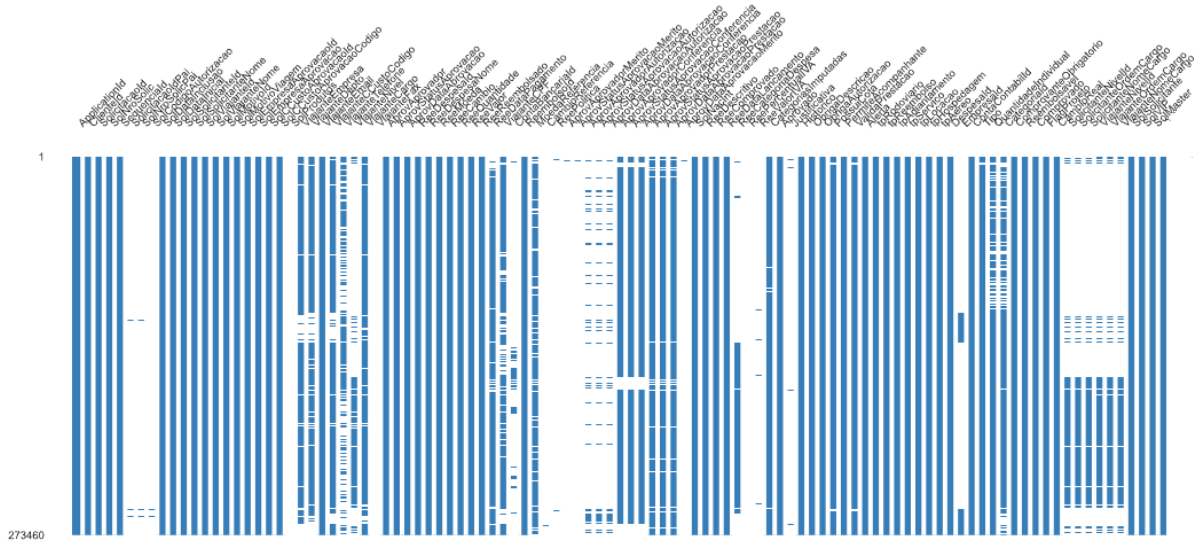
Valores *outliers* podem distorcer a análise estatística e prejudicar o desempenho dos modelos de ML. O método IQR foi utilizado para a remoção desses valores, no qual os valores abaixo de $Q1 - 1,5 * IQR$ ou acima de $Q3 + 1,5 * IQR$ são identificados como *outliers*.

Análises de Valores Nulos ou Ausentes

Remover valores nulos sem uma análise detalhada pode levar à perda de informações importantes e distorcer os resultados. Além disso, é importante considerar se a remoção desses valores não prejudica a representatividade do conjunto de dados.

A matriz de valores nulos e ausentes da biblioteca ydata-profiling, ilustrada na Figura 17, é uma representação densa dos dados que torna mais fácil identificar visualmente os atributos com uma grande quantidade de valores nulos (espaçamentos brancos).

Figura 17 – Visualização de valores ausentes



Fonte: Elaborada pelo autor.

3.3.3 Dados Desbalanceados

Quando as classes são desbalanceadas, ou seja, quando há uma grande disparidade no número de exemplos de cada classe, os modelos de ML podem tender a favorecer essa classe

dominante, resultando em uma baixa capacidade de generalização e desempenho insatisfatório na classificação das classes minoritárias.

Foram aplicadas técnicas de subamostragem da classe majoritária e de sobreamostragem da classe minoritária para atingir um melhor equilíbrio. A Tabela 1 exibe a diminuição de registros de valor 'Aprovado' e o aumento de valor 'Reprovado' da classe alvo após o processo de balanceamento.

Tabela 1 – Balanceamento dos dados

	Aprovado	Reprovado
dados desbalanceados	215937	3322
dados balanceados	**43186**	**21593**

Fonte: Dados da pesquisa.

3.3.4 Seleção de Atributos

A seleção de atributos contribui para a redução da dimensionalidade dos dados, simplificando o modelo e melhorando a eficiência computacional. Além disso, essa prática ajuda a prevenir o *overfitting*, garantindo que o modelo se ajuste adequadamente aos dados de treinamento e generalize bem para novos dados.

Alguns métodos populares para selecionar os atributos de maior relevância ou contribuição para a predição do modelo de aprendizado incluem: informação mútua, importância de atributos e eliminação recursiva de atributos.

Em um projeto de aprendizado de máquina, não há uma solução única e correta, mas sim aquela que, após extensa experimentação e avaliação, produzirá os melhores resultados para o problema em questão.

3.3.4.1 Informação Mútua

A informação mútua mede a informação compartilhada entre duas variáveis aleatórias X e Y, ou seja, o quanto o conhecimento de uma destas variáveis reduza incerteza sobre a outra. Cover e Thomas (2006) descreve que para duas variáveis aleatórias X e Y, com distribuição conjunta de probabilidade $p(x,y)$ e distribuição de probabilidade marginal $p(x)$ e $p(y)$, a sua a informação mútua é definida pela Equação 3.1:

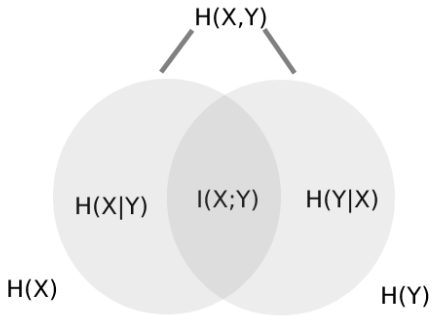
$$I(X,Y) = \sum\sum p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{3.1}$$

Os autores mencionados anteriormente provam que a informação mútua pode ser escrita em termos de entropia (Equação 2.2):

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{3.2}$$

A apresentação desta medida pelo diagrama de Venn (Figura 18) ilustra de forma intuitiva as medidas de informação. A informação mútua somente será nula se os eventos forem independentes. Quando a probabilidade do evento H(X) acontecer dado que já aconteceu o evento H(Y) for menor que a probabilidade do evento H(X) de acontecer, essa condição aumenta a informação mútua e reduz a entropia.

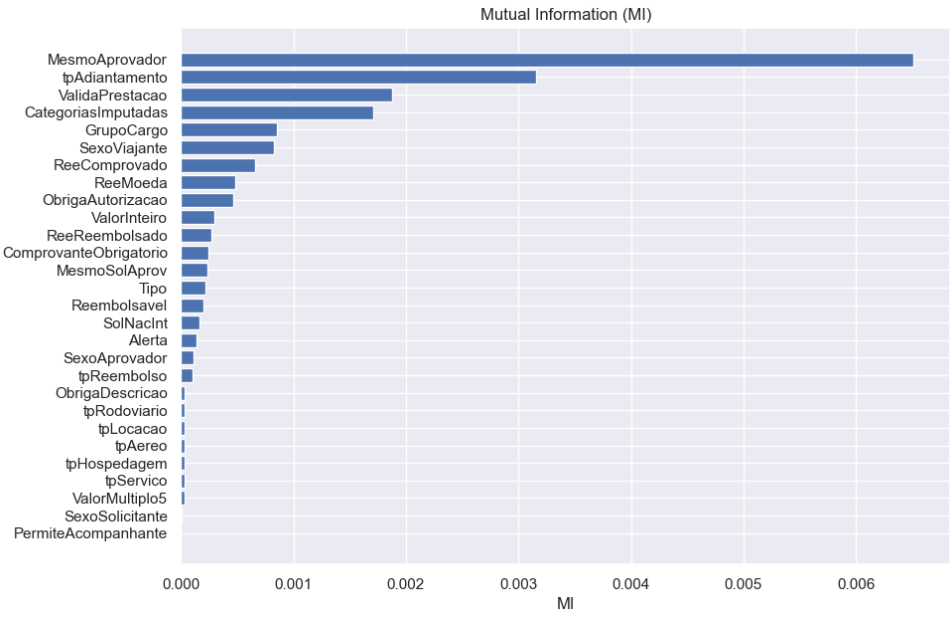
Figura 18 – Diagrama de Venn



Fonte: Elaborada pelo autor.

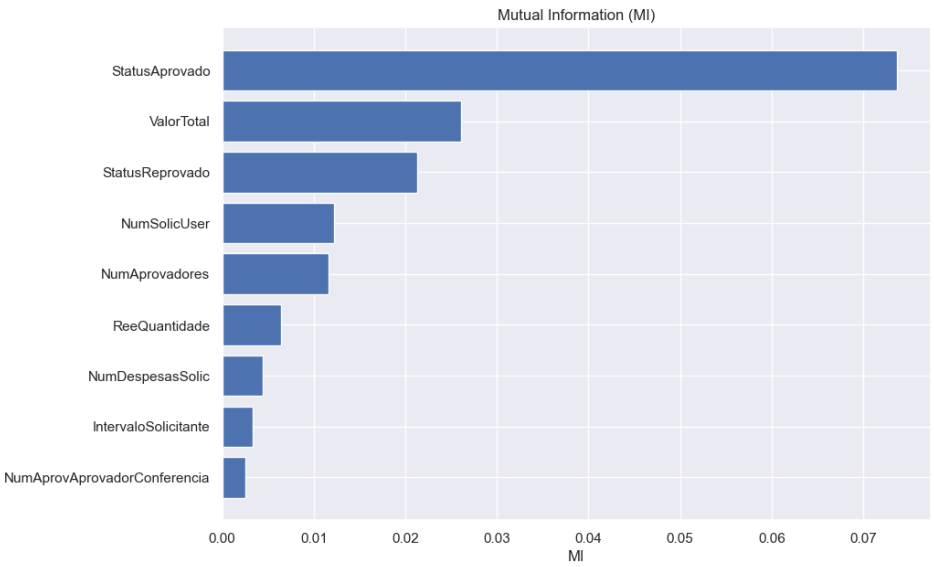
A Figura 19 ilustra a informação mútua de cada atributo categórico com a classe alvo e a Figura 20 entre cada atributo numérico com a classe alvo.

Figura 19 – Informação mútua entre atributos categóricos



Fonte: Elaborada pelo autor.

Figura 20 – Informação mútua entre atributos numéricos



Fonte: Elaborada pelo autor.

3.3.4.2 *Feature Importance*

Nos modelos baseados em árvores de decisão, como *Random Forest*, a importância dos atributos, do inglês *feature importance*, é determinada pela contribuição para a redução da impureza ou ganho de informação em cada nó da árvore. Cada atributo recebe uma pontuação de importância (Figura 21) que reflete sua contribuição específica para a divisão dos dados e a melhoria da qualidade das previsões. O modelo *Random Forest* possui métodos internos para calcular a importância das variáveis.

3.3.4.3 *Eliminação Recursiva de Atributos*

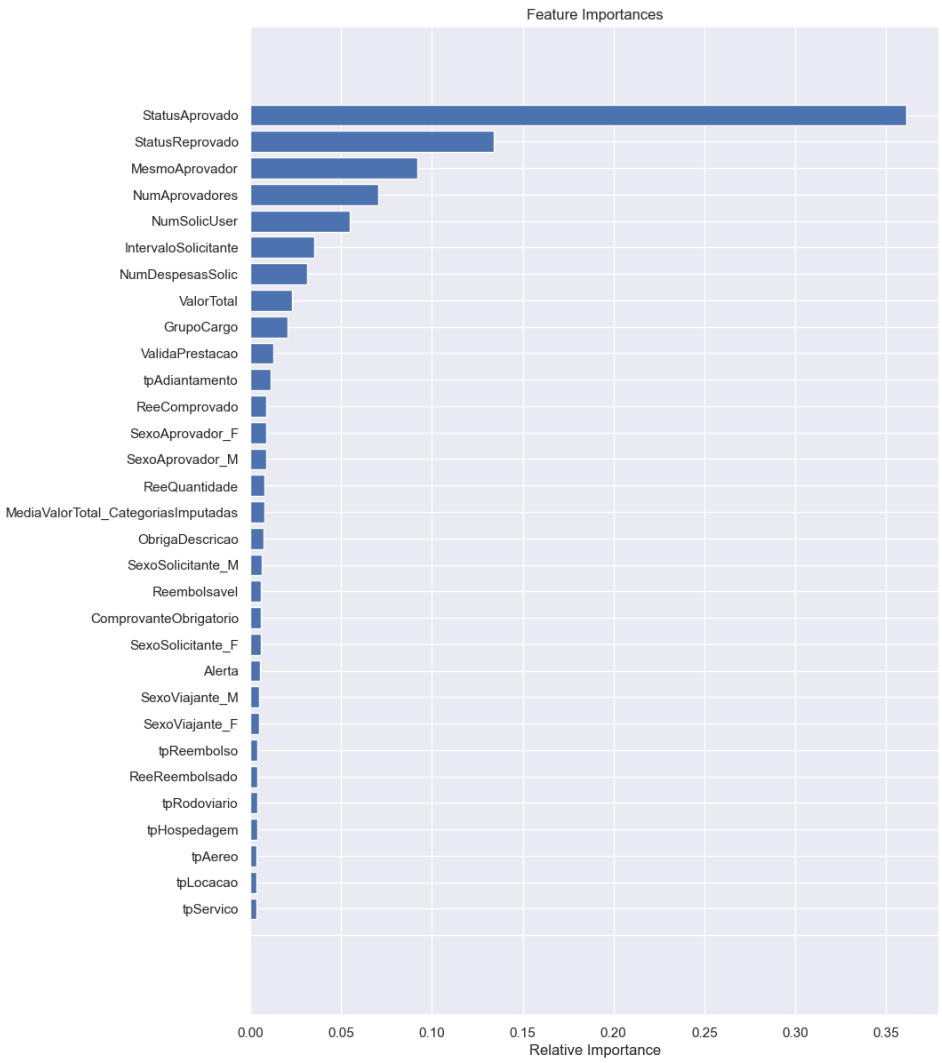
O método *Recursive Feature Elimination* (RFE), ou Eliminação Recursiva de Atributos, é um algoritmo de seleção de atributos disponível na biblioteca *scikit-learn*. O RFE funciona removendo recursivamente os atributos e construindo um modelo com os atributos restantes. Ele usa a acurácia do modelo treinado para identificar quais atributos contribuem mais para prever a classe alvo.

Os principais passos do RFE são:

1. O modelo é treinado no conjunto inicial de atributos e a importância de cada atributo é obtida.

2. Os atributos de menor importância são removidos do conjunto atual de atributos. Esse procedimento é repetido recursivamente no conjunto reduzido até que o número desejado de atributos a selecionar seja eventualmente alcançado.

Figura 21 – Feature importance



Fonte: Elaborada pelo autor.

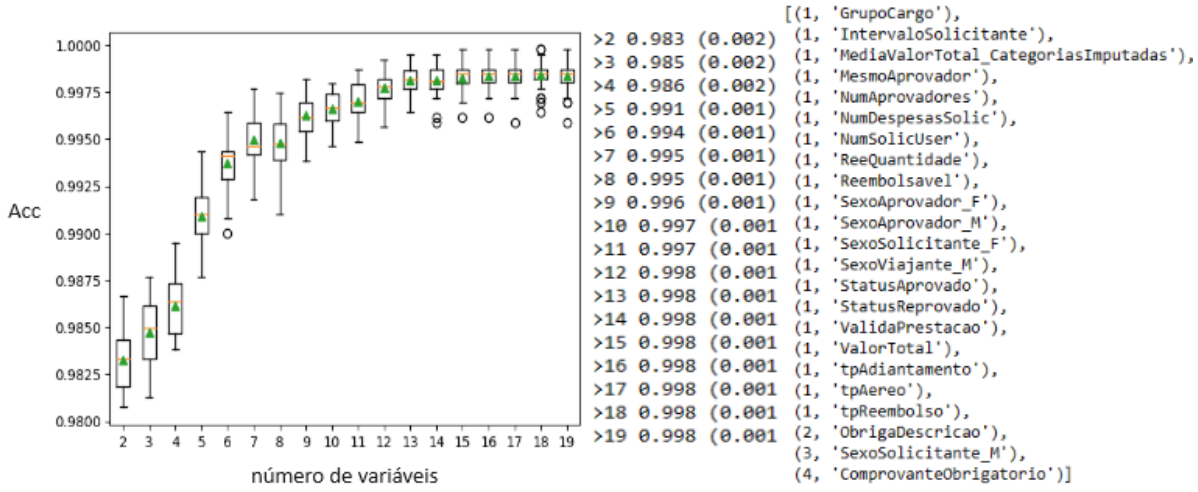
Os resultados podem variar devido à natureza estocástica do algoritmo ou devido a diferenças na precisão numérica. Recomenda-se executar o exemplo várias vezes e comparar o resultado médio.

Neste caso, observa-se na Figura 22 que a acurácia (eixo y) geralmente melhora à medida que o número de variáveis (eixo x) aumenta e possivelmente atinge um pico em torno de 14 a 16 atributos. Na mesma figura o gráfico de boxplot representa a distribuição das pontuações de precisão para cada número configurado de atributos e a classificação dos atributos mais relevantes.

3.4 Treinamento

Para este estudo, foram selecionados os quatro classificadores abordados na Seção 2.3: Árvore de Decisão, *Random Forest*, *Bagging* e *Gradient Boosting*.

Figura 22 – REF



Fonte: Elaborada pelo autor.

3.4.1 Amostragem

A divisão do conjunto de dados em subconjuntos disjuntos para treinamento, validação e teste garante que o modelo seja treinado com dados limpos, avaliado de forma imparcial e testado em um cenário real sem viés. Dois métodos comumente utilizados para particionar os dados são o *Holdout* e a validação cruzada estratificada.

Método *Holdout*

O método *Holdout* particiona o conjunto de dados em uma porcentagem p para treinamento e $(1 - p)$ para teste (MONARD; BARANAUSKAS, 2003). A partição de treinamento (p) pode ser dividido para criação de um conjunto de validação. A proporção de divisão entre os conjuntos de treinamento e teste é geralmente definida como 80% e 20%, respectivamente. Essa proporção pode ser ajustada de acordo com o tamanho do conjunto de dados e a necessidade de um conjunto de teste maior para uma avaliação mais robusta.

A principal desvantagem do método *Holdout* é sua variabilidade, uma vez que envolve apenas uma iteração. Isso significa que o desempenho do modelo pode depender significativamente da qualidade da divisão escolhida.

Método *Validação Cruzada Estratificada*

Na validação cruzada estratificada, do inglês *Stratified K-Fold Cross Validation*, o conjunto de dados é dividido em k partes, conhecidos como dobras (*folds*), de modo que a distribuição das classes alvo seja preservada em cada *fold*. Isso implica que a proporção de cada classe nos folds é aproximadamente a mesma que a proporção no conjunto de dados original (MONARD; BARANAUSKAS, 2003).

Durante o processo de validação cruzada estratificada, o modelo é treinado k vezes, em cada iteração utilizando k-1 *folds* como conjunto de treinamento e 1 *fold* como conjunto

de validação. A média dos desempenhos obtidos em cada *fold* é calculada para fornecer uma estimativa imparcial do desempenho do modelo. Isso assegura que o modelo seja testado em todos os exemplos do conjunto de dados, tornando a avaliação do desempenho mais confiável, pois não depende de uma única divisão arbitrária dos dados em conjuntos de treinamento e teste.

3.4.2 Otimização de Hiperparâmetros

Hiperparâmetros controlam o comportamento geral do modelo e são ajustados manualmente antes do treinamento, diferentemente dos parâmetros que são aprendidos pelo modelo durante o treinamento. Exemplos de hiperparâmetros incluem a taxa de aprendizado, o número de árvores em uma floresta aleatória ou a profundidade máxima de uma árvore de decisão. Foram utilizados dois métodos para selecionar os melhores hiperparâmetros durante o treinamento dos classificadores: *GridSearchCV* e *gp_minimize*.

É importante ressaltar que, embora as duas bibliotecas retornem resultados semelhantes, elas possuem métodos e funções peculiares que se complementam no estudo. Isso reforça a ideia de que não há uma solução única e correta, e que a experimentação com diferentes abordagens é fundamental para encontrar a melhor estratégia para o problema em questão.

Método *GridSearchCV*

O *GridSearchCV* é uma função da biblioteca *scikit-learn* que implementa um método de pesquisa exaustiva sobre a grade de valores que são testados para os hiperparâmetros do modelo. Para cada combinação única de hiperparâmetros, o *GridSearchCV* realiza uma validação cruzada. Isso envolve dividir o conjunto de dados de treinamento em um número especificado de subconjuntos. O modelo é então treinado em todos os subconjuntos, exceto no subconjunto de teste, no qual o desempenho do modelo é avaliado. Este processo é repetido para cada subconjunto de treinamento.

O *GridSearchCV* é uma maneira eficiente de ajustar os hiperparâmetros de um modelo e pode melhorar significativamente o seu desempenho. No entanto, também pode ser computacionalmente intensivo, especialmente se a grade de valores for grande e o modelo for complexo. Portanto, é importante considerar o trade-off entre o tempo de computação e a melhoria do desempenho do modelo ao usar *GridSearchCV*.

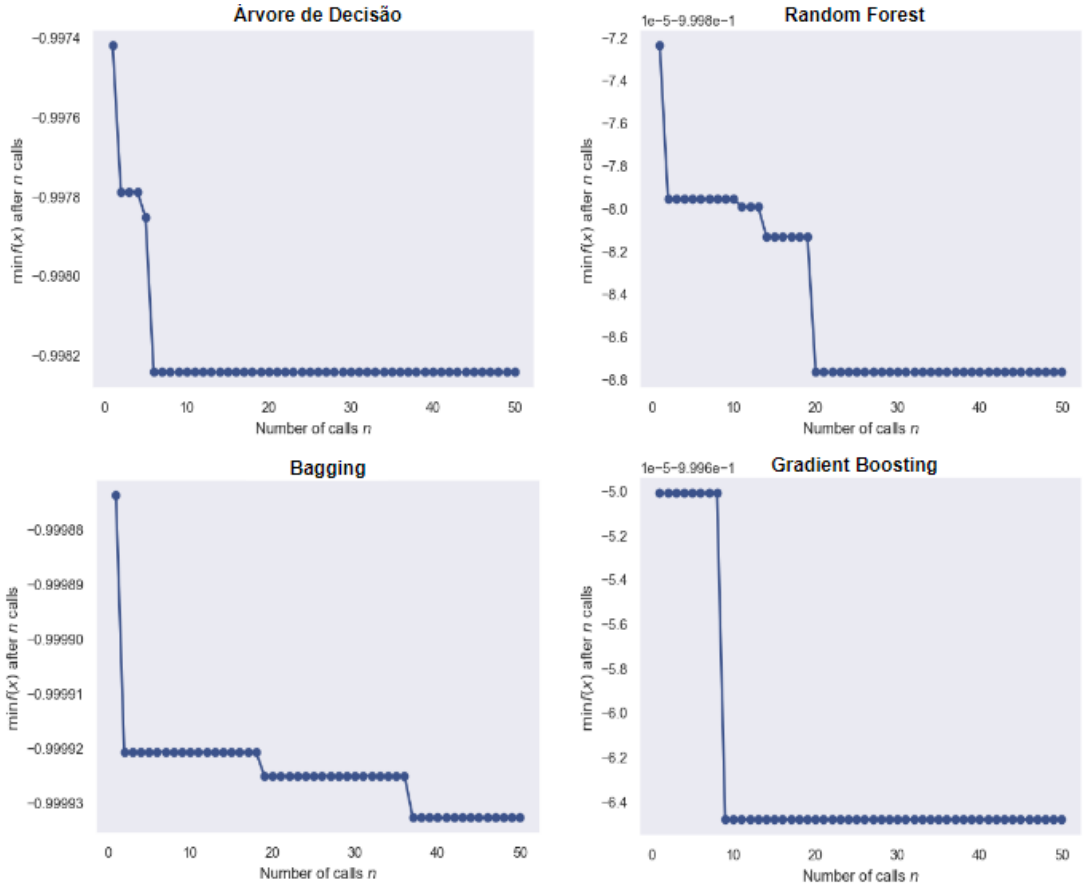
Método *gp_minimize*

A função *gp_minimize* da biblioteca *scikit-optimize* (HEAD *et al.*, 2018) é utilizada para realizar otimização sequencial baseada em modelo, usando processos gaussianos. Ela é especialmente útil para otimizar funções de custo que são muito caras para avaliar. A função *gp_minimize* busca os valores de hiperparâmetros que minimizam uma função objetivo, dentro de um espaço de busca definido. A função *gp_minimize* é geralmente mais rápida que o Grid Search por várias razões:

- **Eficiência na Busca:** A *gp_minimize* usa processos gaussianos para criar um modelo probabilístico da função objetivo. Isso permite que ele identifique regiões promissoras do espaço de hiperparâmetros mais rapidamente e concentre a busca nessas áreas.
- **Menos Avaliações:** Ao contrário do *GridSearchCV*, que avalia todos os pontos em uma grade pré-definida, *gp_minimize* seleciona apenas pontos que são mais prováveis de melhorar o resultado. Isso reduz significativamente o número de avaliações necessárias.
- **Aprendizado a partir de Iterações Anteriores:** A *gp_minimize* aprende com cada iteração e ajusta sua estratégia de busca, enquanto o *GridSearchCV* não tem essa capacidade adaptativa.

A função **plot_convergence** da biblioteca *scikit-optimize* é utilizada para visualizar a convergência do processo de otimização. Essa função gera um gráfico que mostra o melhor valor da função objetivo encontrado a cada iteração da pesquisa. O eixo x do gráfico gerado representa o número de chamadas para a função objetivo, e o eixo y mostra o melhor valor da função objetivo encontrado até aquele ponto. A Figura 23 representa o gráfico de convergência dos quatro classificadores escolhidos nesse estudo.

Figura 23 – Convergência



Fonte: Elaborada pelo autor.

CAPÍTULO

4

RESULTADOS E DISCUSSÕES

Neste capítulo, serão apresentados os resultados obtidos a partir da aplicação dos classificadores de aprendizado de máquina ao conjunto de dados. Inicialmente, serão destacados os principais resultados observados, seguidos de uma análise mais detalhada das conclusões obtidas. Por fim, serão aplicadas técnicas de interpretação SHAP no classificador de melhor desempenho.

4.1 Desempenho dos Classificadores

O método ***predict_proba*** é fornecida por algoritmos de classificação da biblioteca *scikit-learn*. Este método retorna a probabilidade da amostra pertencer à classe positiva e à classe negativa.

Essas probabilidades são úteis para ajustar o ponto de corte (*threshold*) para a classificação. Por exemplo, ao escolher um *threshold* diferente, é possível ajustar a sensibilidade (*recall)* e a especificidade do modelo. Isso pode ser útil em casos onde é mais importante reduzir falsos positivos ou falsos negativos, dependendo do contexto do problema.

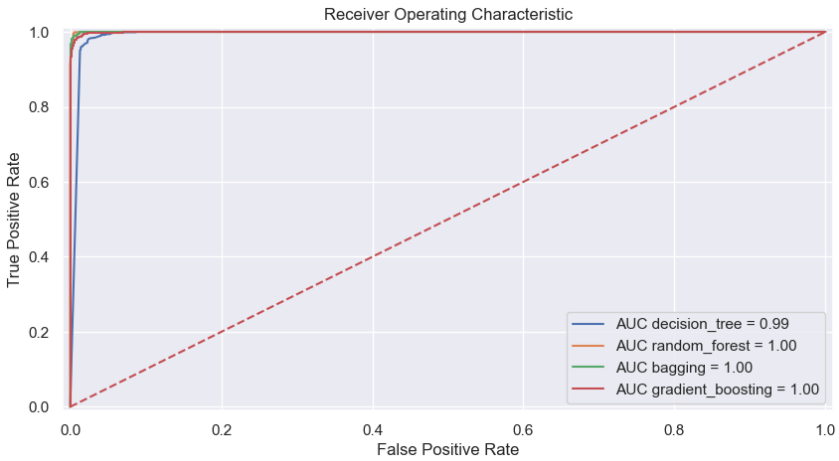
A Tabela 2 resume os resultados das métricas de desempenho dos modelos treinados em uma amostra de 2000 exemplos, contendo 1000 casos positivos e 1000 casos negativos, com um *threshold* de 0.9.

Tabela 2 – Métricas de desempenho.

Models	Acc	AUC	Precision	Recall	F1-Score
Decision Tree	0.977	0.977	0.9772	0.977	0.9772
Random Forest	**0.9945**	**0.9945**	**0.9945**	**0.9944**	**0.9944**
Bagging	0.985	0.985	0.9853	0.985	0.9847
Gradient Boosting	0.993	0.993	0.993	0.993	0.993

A Figura 24 apresenta o gráfico da curva ROC e a métrica AUC de cada classificador.

Figura 24 – Curva ROC e métrica AUC.

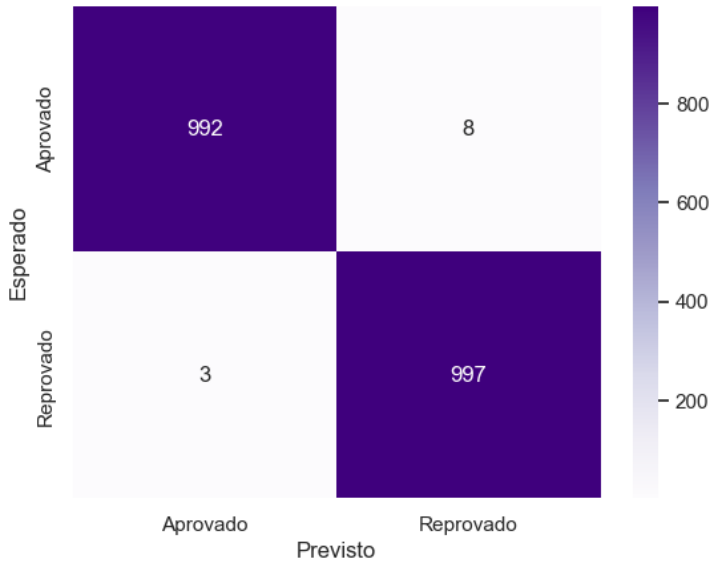


Fonte: Elaborada pelo autor.

Random Forest

Como evidenciado na Tabela 2, o classificador *Random Forest* obteve os melhores resultados na etapa de avaliação com dados nunca vistos pelo modelo anteriormente. A matriz de confusão (Figura 25) proporciona uma visualização clara da quantidade de acertos e erros para cada classe. Além disso, na Figura 26, observa-se que a curva ROC se aproxima de uma curva ideal, com a medida AUC próxima de 1.

Figura 25 – Matriz de Confusão Random Forest com threshold 0.9



Fonte: Elaborada pelo autor.

Aumentar o threshold para 0.95 resultou em uma pequena degradação no desempenho geral do modelo em comparação com o threshold de 0.9 (Tabela 3). No entanto, uma análise

Figura 26 – Curva ROC Random Forest

Fonte: Elaborada pelo autor.

mais detalhada das métricas revela uma diminuição na taxa de falsos positivos (TFP) (Tabela 4).

Tabela 3 – Desempenho do classificador Random Forest.

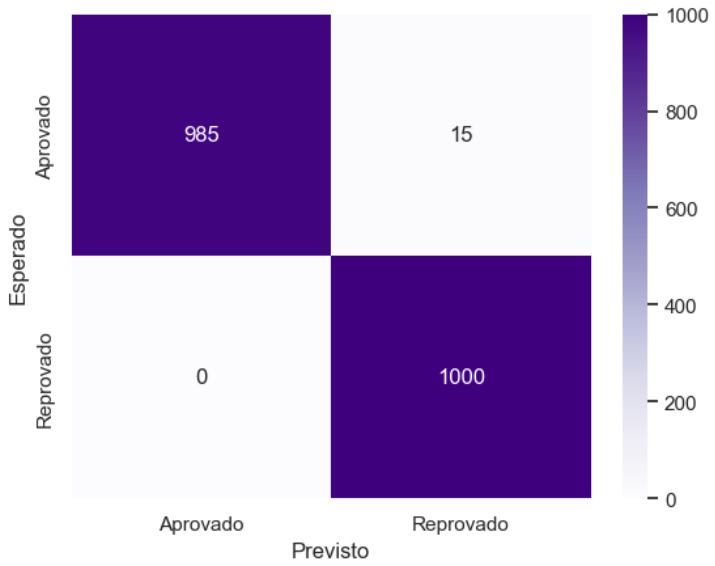
Random Forest					
Threshold	Acc	Auc	Precision	Recall	F1-Score
0.9	**0.9945**	**0.9945**	**0.9945**	**0.9944**	**0.9944**
0.95	0.9925	0.9925	0.9926	0.9925	0.9924

Tabela 4 – TVP e TFP do Random Forest.

Random Forest		
Threshold	TVP	TFP
0.9	0.992	0.003
0.95	**0.985**	**0**

A matriz de confusão (Figura 27) do modelo *Randon Forest* com *threshold* em 0.95 demonstra que 100% dos registros em não conformidade foram detectados. Essa informação converge com o objetivo principal do projeto, que é identificar registros em não conformidade com baixa taxa de falsos positivos e alta taxa de confiabilidade.

Figura 27 – Matriz de Confusão Random Forest com threshold 0.95



Fonte: Elaborada pelo autor.

4.2 Interpretabilidade dos Classificadores

Através de diversas opções de visualização, o SHAP permite compreender o funcionamento dos modelos fornecendo explicações globais e locais de como as variáveis contribuem para as previsões.

4.2.1 Interpretabilidade Local

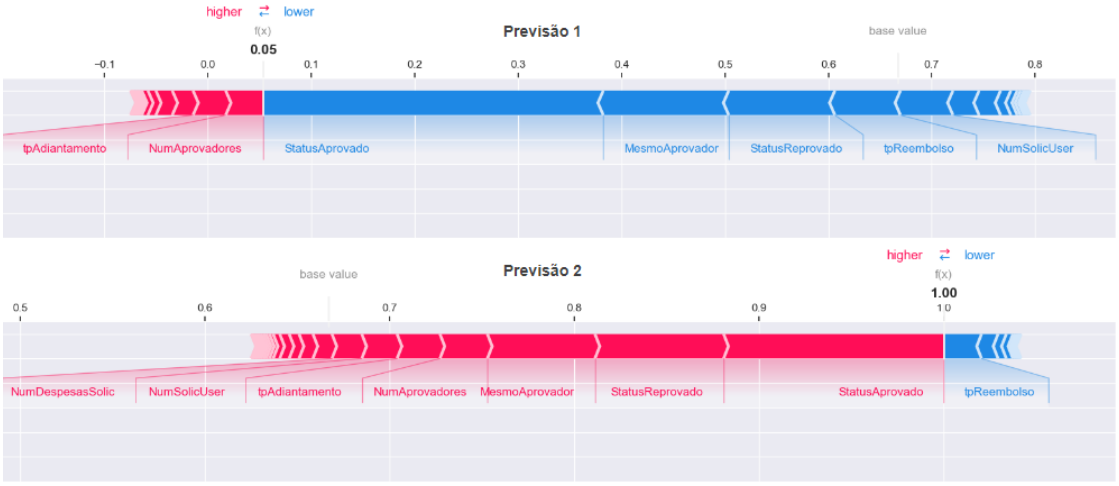
O SHAP fornece valores que explicam a contribuição de cada variável para a previsão de uma instância específica, o que é útil para entender o comportamento do modelo em nível granular.

SHAP Force Plot

Gráfico de interpretação de força representa as contribuições de cada variável como "forças" que aumentam ou diminuem a previsão do modelo. Cada seta no gráfico representa a contribuição de uma variável para uma previsão específica, ajudando a visualizar como as variáveis individuais afetam o resultado final.

A Figura 28 mostra duas previsões com seus respectivos gráficos de força SHAP. No gráfico da Previsão 1, os atributos com valores baixos (em azul) têm forte influência na reprovação do reembolso. Já no gráfico da Previsão 2, o comportamento é oposto.

Figura 28 – SHAP Forces



Fonte: Elaborada pelo autor.

4.2.2 Interpretabilidade Global

Além das explicações locais, o SHAP oferece uma visão geral das contribuições de cada variável para o modelo como um todo.

SHAP Feature Importance Plot

A importância dos atributos no SHAP é determinada pela média dos valores absolutos de Shapley para cada atributo em todo o conjunto de dados (MOLNAR, 2022).

Ao comparar os resultados da importância dos atributos entre o modelo *Random Forest* (Figura 21) e o método SHAP (Figura 29), observa-se uma semelhança, com os quatro atributos mais importantes ocupando as primeiras posições em ambas as análises.

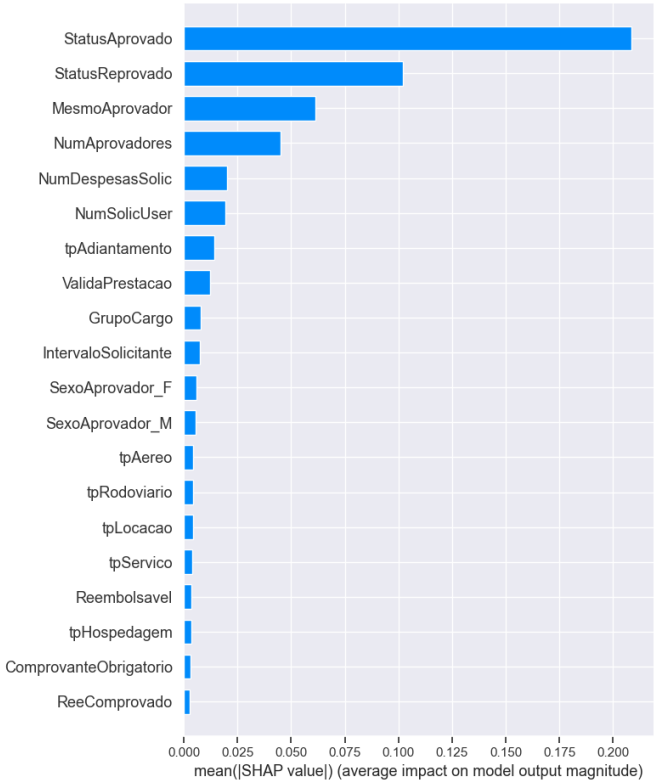
SHAP Summary Plot

Molnar (2022) explica que o gráfico de resumo (*summary plot*) combina a importância dos atributos com os efeitos desses atributos. Cada ponto no gráfico representa um valor de Shapley para um atributo e uma instância específica. A posição no eixo y é determinada pelo atributo, com cada atributo possuindo 2000 pontos distribuídos horizontalmente, correspondentes às 2000 instâncias separadas para a fase de testes. No eixo x, a posição é determinada pelo valor de Shapley; valores positivos contribuem para que o modelo responda com a categoria positiva, enquanto valores negativos indicam suporte à categoria oposta.

A cor do ponto indica o valor do atributo, variando de baixo (azul) para alto (vermelho). Para evitar sobreposições, os pontos são deslocados na direção do eixo y, permitindo uma visualização clara da distribuição dos valores de Shapley para cada atributo. Os atributos são ordenados com base em sua importância.

Atributos com uma divisão bem clara de cores são considerados bons preditores, pois ao mudar seu valor, o modelo consegue identificar de forma mais simples sua contribuição para

Figura 29 – SHAP Feature Importance



Fonte: Elaborada pelo autor.

cada classe. Na Figura 30, essa característica de pontos vermelhos e azuis em lugares opostos está acentuada no atributo *"MesmoAprovador"*.

Além da divisão clara de cores, quanto maior o intervalo de alcance dos valores de SHAP, maior será a importância desse atributo para o modelo. Na Figura 30, o atributo *"StatusAprovado"* tem seu valor Shap variando entre -0.4 a 0.3.

SHAP Dependence Plot

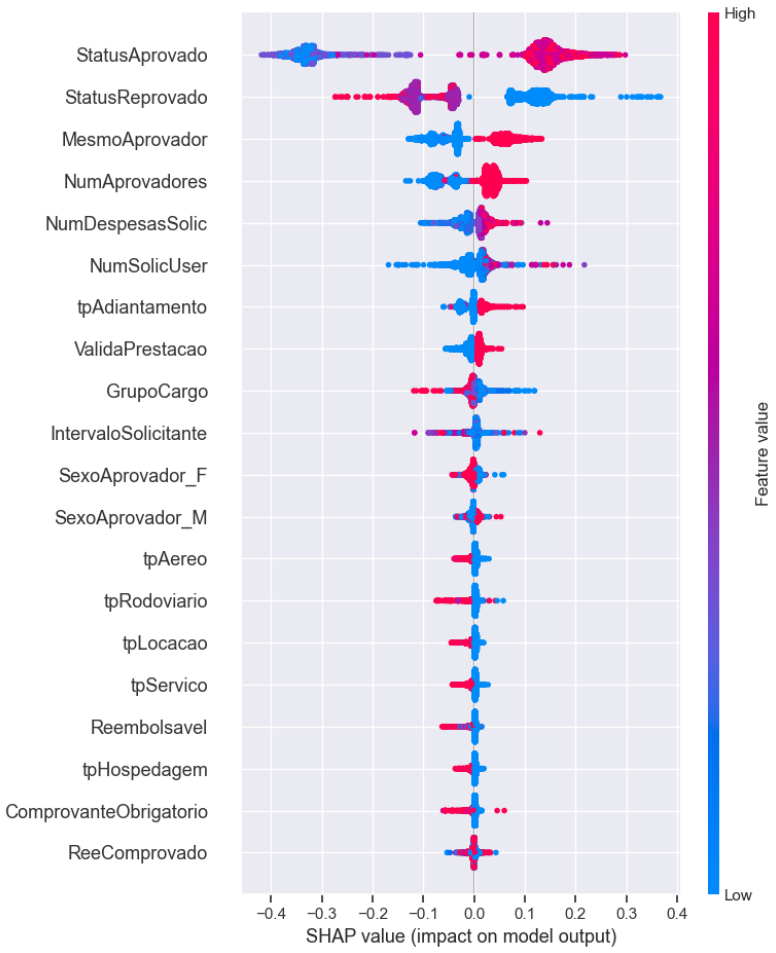
O gráfico de dependência do SHAP com interação mostra como o efeito de um atributo em uma previsão varia de acordo com o valor de outro atributo.

Na Figura 31, demonstra a relação entre o atributo *"StatusAprovado"* e seu valor de Shapley, interagindo com o atributo *"StatusReprovado"*. Se a quantidade de vezes em que um reembolso passou por conferência e foi aprovado (*"StatusAprovado"*) for maior do que o número de vezes que foi reprovado, é provável que o valor de Shapley do atributo *"StatusAprovado"* seja maior.

4.3 Aplicação Pós-Análises

Os modelos treinados são comumente armazenados em formato de bytes usando a biblioteca *pickle* do *Python*. Isso permite que sejam salvos em disco e reutilizados posteriormente,

Figura 30 – SHAP Summary Plot



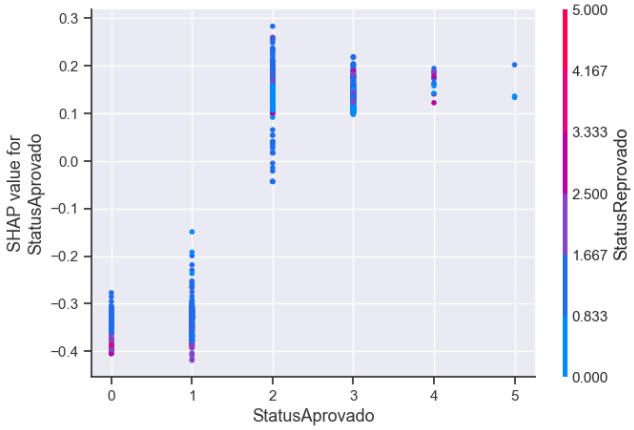
Fonte: Elaborada pelo autor.

eliminando a necessidade de re-treinar os modelos sempre que são necessários. Essa prática economiza tempo e recursos, especialmente em modelos que requerem horas ou dias para serem treinados.

A Figura 32 ilustra um sistema de formato tabular simples desenvolvido para o monitoramento de transações de reembolso em gastos corporativos. Junto às informações originais da transação, há a inclusão da coluna categórica ***"Risco"***, indicando o risco de não conformidade e da coluna ***"Score"*** com a sua respectiva pontuação atribuída pelo classificador. Ademais, a Figura 33 demonstra a possibilidade de análise detalhada de cada transação individualmente.

Com a inclusão do novo atributo de classificação de despesas ao conjunto de dados, é possível realizar novas análises para extrair insights significativos sobre a ocorrência de casos de não conformidade. Essas análises podem revelar quais áreas, sazonalidades e tipos de despesas são mais suscetíveis a esses casos, além de identificar possíveis falhas nos processos internos. Com essas informações, torna-se viável implementar medidas corretivas e preventivas eficazes, como mencionado na Seção 1.1, garantindo uma resposta ágil e aprimorando a conformidade organizacional.

Figura 31 – SHAP Dependence Plot



Fonte: Elaborada pelo autor.

Figura 32 – Sistema de Análise de Reembolsos

	Risco	Score	Nome Solicitante	Nome Viajante	Data Inicial	Empresa Viajante	Categoria Despesa Orig…	Ca
0	Baixo	0.995	User 246	User 246	14/02/2019	Company 7	Percurso	
1	Baixo	1.000	User 81	User 48	01/08/2020	Company 12	Não Informado	
2	Moderado	0.935	User 226	User 232	02/03/2020	Company 7	Alimentacao	
3	Moderado	0.941	User 93	User 93	09/12/2019	Company 7	Transporte	
4	Baixo	1.000	User 358	User 321	01/03/2020	Company 14	Não Informado	
5	Moderado	0.900	User 248	User 248	06/05/2019	Company 10	Alimentacao	
6	Baixo	0.995	User 215	User 215	20/12/2018	Company 7	Percurso	
7	Baixo	0.990	User 18	User 18	16/11/2018	Company 7	Percurso	
8	Baixo	1.000	User 319	User 319	23/08/2020	Company 5	Não Informado	
9	Baixo	1.000	User 292	User 292	18/09/2019	Company 7	Percurso	
10	Moderado	0.945	User 179	User 230	09/09/2019	Company 13	Percurso	

Fonte: Elaborada pelo autor.

Figura 33 – Gráfico de Medição

Fonte: Elaborada pelo autor.

CAPÍTULO

5

CONCLUSÃO

A detecção de não conformidade em gastos corporativos por meio de técnicas avançadas de aprendizado de máquina demonstrou-se uma abordagem altamente eficaz e promissora. Os resultados obtidos ao longo deste estudo reforçam a viabilidade e a importância de empregar tais técnicas para identificar padrões anômalos e potenciais irregularidades nos processos de despesas corporativas.

O sistema de detecção desenvolvido, baseado em técnicas de aprendizado de máquina, evidenciou excelentes níveis de precisão na identificação de transações em não conformidade. Os modelos construídos foram capazes de capturar com sucesso padrões complexos nos dados, permitindo uma detecção precisa de anomalias, contribuindo significativamente para mitigar riscos financeiros e melhorar a integridade dos processos de gastos corporativos.

É importante ressaltar, no entanto, que embora o sistema detector tenha se mostrado uma ferramenta poderosa no fluxo de análise e aprovação de despesas corporativas, não deve ser considerado o único método. A gestão eficaz de gastos corporativos requer uma abordagem multifacetada, na qual diferentes técnicas e métodos são empregados em conjunto para garantir a integridade e a transparência dos processos.

Por exemplo, além do sistema de detecção de não conformidades, a implementação de políticas claras de despesas, a realização de auditorias regulares, a adoção de sistemas de controle de gastos automatizados e a educação contínua dos colaboradores são aspectos fundamentais para uma gestão sólida e eficiente das despesas corporativas.

Portanto, o presente estudo destaca a importância de integrar o sistema detector desenvolvido como parte de um conjunto de ferramentas e práticas que visam garantir a integridade, transparência e eficiência na gestão de gastos corporativos.

Esta dissertação contribui não apenas com um modelo preditivo associado a avanços na detecção de fraudes, mas também ressalta a necessidade de uma abordagem holística para a

gestão financeira corporativa, onde a combinação de tecnologias avançadas e práticas robustas desempenham um papel importante na promoção de uma cultura organizacional de compliance e responsabilidade financeira.

REFERÊNCIAS

ACFE. **Occupational Fraud 2022: Report to the Nations"**. [S.l.], 2023. Disponível em: <https://acfepublic.s3.us-west-2.amazonaws.com/2022+Report+to+the+Nations.pdf>. Acesso em: 18 nov. 2023. Citado na página 27.

____. **2024 Anti-Fraud Technology Benchmarking Report"**. [S.l.], 2024. Disponível em: <https://www.acfe.com/-/media/files/acfe/pdfs/sas_benchmarkingreport_2024.pdf>. Acesso em: 18 mar. 2024. Citado na página 29.

BIRD STEVEN, E. L.; KLEIN, E. **Natural Language Processing with Python**. [S.l.]: O'Reilly Media Inc., 2009. Citado na página 48.

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, p. 123–140, 1996. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00058655>. Citado na página 39.

____. Random forests. Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 40.

BROWNLEE, J. **Machine Learning Mastery**. 2016. Disponível em: <https://machinelearningmastery.com>. Acesso em: 14/10/2023. Citado na página 49.

BRUGMAN, S. **ydata-profiling: Exploratory Data Analysis for Python**. 2019. <https://github.com/pandas-profiling/pandas-profiling>. Acesso em: 03/11/2023. Citado na página 50.

COVER, T. M.; THOMAS, J. A. **Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)**. USA: Wiley-Interscience, 2006. ISBN 0471241954. Citado nas páginas 37 e 60.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011. Citado nas páginas 31, 33, 35, 36, 37, 38, 39, 41, 43 e 50.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189 – 1232, 2001. Disponível em: <https://doi.org/10.1214/aos/1013203451>. Citado nas páginas 40 e 41.

GAMA, J.; CARVALHO, A. C. P. d. L. F. d.; FACELI, K.; LORENA, A. C.; OLIVEIRA, M. **Extração de conhecimento de dados: data mining**. [S.l.]: Edições Sílabo, 2015. Citado na página 32.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with numpy. **Nature**, v. 585, n. 7825, p. 357–362, 2020. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>. Citado na página 48.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer, 2009. (Springer series in statistics). ISBN 9780387848846. Disponível em: <https://hastie.su.domains/Papers/ESLII.pdf>. Citado nas páginas 35 e 38.

HEAD, T.; MECHCODER, G. L.; SHCHERBATYI, F. I.; VINíCIUS, c. Z.; SCHRöDER, N. C.; CAMPOS, T. Y. N.; CEREDA, T. F. S.; RENE-REX, K. K. S.; SCHWABEDAL, C. D. C. S. J.; HVASS-LABS, M. P.; FABISCH, A. **scikit-optimize/scikit-optimize: v0.5.2**. 2018. <https://doi.org/10.5281/zenodo.1207017>. Acesso em: 25/11/2023. Citado na página 65.

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**. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 4768–4777. ISBN 9781510860964. Citado na página 45.

MCKINNEY, W. Data structures for statistical computing in python. In: WALT, S. van der; MILLMAN, J. (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 51 – 56. Citado na página 48.

MILLER, T. Explanation in artificial intelligence: Insights from the social sciences. **Artificial Intelligence**, v. 267, p. 1–38, February 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>. Citado na página 45.

MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <http://www.cs.cmu.edu/~tom/mlbook.html>. Citado na página 35.

MOLNAR, C. **Interpretable Machine Learning**: A guide for making black box models explainable. 2. ed. [s.n.], 2022. Disponível em: <https://christophm.github.io/interpretable-ml-book>. Citado nas páginas 45 e 71.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: **Sistemas Inteligentes-Fundamentos e Aplicações**. [S.l.]: Editora Manole, 2003. p. 89–114. Citado nas páginas 35, 42 e 64.

OPITZ, D.; MACLIN, R. Popular ensemble methods: An empirical study. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 11, p. 169–198, ago. 1999. ISSN 1076-9757. Disponível em: <http://dx.doi.org/10.1613/jair.614>. Citado na página 39.

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**, v. 12, p. 2825–2830, 2011. Citado na página 48.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical Recipes 3rd Edition: The Art of Scientific Computing**. 3. ed. USA: Cambridge University Press, 2007. ISBN 0521880688. Citado na página 33.

Python Core Team. **Python: A dynamic, open source programming language**. [S.l.], 2019. Python version 3.9. Disponível em: <https://www.python.org/>. Citado na página 48.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, n. 1, p. 81–106, 1986. Disponível em: <https://doi.org/10.1007/BF00116251>. Citado na página 38.

SCARINCI, T. F. B. **Fraudes corporativas: uma análise dos seus determinantes e do seu efeito sobre o desempenho das empresas brasileiras de capital aberto.** Dissertação (Mestrado) — Universidade Federal de Goiás, 2021. Available at <https://repositorio.bc.ufg.br/tedeserver/api/core/bitstreams/9d032415-5b46-41e4-92ff-d3b7af18dada/content>. Citado na página 28.

SHAPLEY, L. S. A value for n-person games. In: KUHN, H. W.; TUCKER, A. W. (Ed.). **Contributions to the Theory of Games II**. Princeton: Princeton University Press, 1953. p. 307–317. Citado na página 45.

SHIXIN, L. **Exploratory Data Analysis for Feature Selection in Machine Learning**. 2020. Disponível em: <https://cloud.google.com/blog/products/ai-machine-learning/building-ml-models-with-eda-feature-selection>. Acesso em: 05/11/2023. Citado na página 49.

GLOSSÁRIO

Compliance: cumprimento das leis, regulamentos, políticas internas e padrões éticos relevantes para uma determinada organização ou setor. Isso envolve garantir que a empresa esteja em conformidade com todas as regras e regulamentos aplicáveis, tanto em nível nacional quanto internacional.

Framework: é uma abstração que une códigos comuns entre vários projetos de *software* provendo uma funcionalidade genérica. *Frameworks* são projetados com a intenção de facilitar o desenvolvimento de *software*, habilitando designers e programadores a gastarem mais tempo determinando as exigências do *software* do que com detalhes de baixo nível do sistema.

Insights: refere-se a percepções ou entendimentos profundos e perspicazes derivados da análise de dados, observações ou experiências. No contexto de análise de dados, os *insights* são descobertas significativas e úteis obtidas ao examinar padrões, tendências e relações nos dados.

Interface: em um contexto tecnológico, uma interface geralmente se refere à maneira pela qual um usuário interage com um sistema ou programa de computador. Isso pode incluir elementos visuais, como menus, botões e telas de exibição, bem como métodos de entrada, como teclado, *mouse*, tela sensível ao toque ou reconhecimento de voz.

Trade-off: refere-se à situação em que se precisa escolher entre duas ou mais opções que têm vantagens e desvantagens diferentes. Dessa forma, ao fazer uma escolha, é necessário ponderar esses aspectos e estar ciente de que a melhoria em uma área pode resultar em uma piora em outra.


ICMC
USP
SÃO CARLOS