GUILHERME GOTO ESCUDERO

Pycausal-explorer: uma biblioteca de inferência causal para dados observacionais

Versão Revisada

São Paulo
2024

GUILHERME GOTO ESCUDERO

Pycausal-explorer: uma biblioteca de inferência causal para dados observacionais

Versão Revisada

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Ciências

Área de Concentração:
Engenharia Elétrica

Orientadora:
Profa. Dra. Roseli de Deus Lopes

São Paulo
2024

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Escudero, Guilherme
Pycausal-explorer: uma biblioteca de inferência causal para dados observacionais / G. Escudero -- versão corr. -- São Paulo, 2024.
87 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.CAUSALIDADE 2.SOFTWARE LIVRE 3.INFERÊNCIA ESTATÍSTICA I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

Resumo

A crescente disseminação dos algoritmos de *machine learning* trouxe avanços notáveis em diversas áreas do conhecimento. Esses progressos foram impulsionados pela expansão da capacidade de coleta, armazenamento e processamento de dados. No entanto, à medida que os métodos de *machine learning* se desenvolvem e encontram novas aplicações, surge uma questão fundamental e frequentemente negligenciada: se existe apenas correlação entre as variáveis ou se elas têm uma relação causal. A necessidade de responder à pergunta "E se?"se torna cada vez mais urgente. Nesse contexto, as técnicas de inferência causal, como as usadas em experimentos controlados aleatórios, desempenham um papel fundamental na obtenção de *insights* confiáveis. No entanto, esses experimentos controlados aleatórios enfrentam desafios como altos custos e duração prolongada, enquanto os dados observacionais (coletados sem manipulação deliberada) são uma alternativa viável, mas que por sua vez apresentam complexidades próprias, como a falta de controle sobre o tratamento aplicado. A questão do contrafactual, que envolve considerar "E se uma ação alternativa tivesse sido tomada em vez daquela observada?", torna-se central na inferência causal. Nesta pesquisa, foi realizada uma revisão dos conceitos de causalidade e inferência causal, seguida do detalhamento e comparação entre os *frameworks* de modelagem causal de Neyman-Rubin e de Pearl. Tomando como base o *framework* de Neyman-Rubin, foi revisada a teoria por trás dos principais modelos utilizados em inferência causal de dados observacionais. Outra contribuição desta pesquisa foi a elaboração da Pycausal-explorer, uma biblioteca em Python de código aberto, que, além de implementar os modelos descritos e analisados neste texto, permite a integração com o scikit-learn que é uma das principais bibliotecas de *machine learning* em Python. Com isso, espera-se promover a compreensão e aplicação desses modelos em análises de dados observacionais nas mais diversas áreas, proporcionando *insights* valiosos e embasados em relações de causa e efeito mais robustas e sólidas.

Palavras chaves: inferência causal, código aberto, dados observacionais

Abstract

The increasing prevalence of machine learning algorithms has brought remarkable advancements in various fields of knowledge. These progressions are driven by the expansion of data collection, storage, and processing capabilities. However, as machine learning methods continue to evolve and find new applications, a fundamental and often overlooked question arises: whether there is only a correlation between variables or if they have a causal relationship. The need to answer the "what if?"question becomes increasingly urgent. In this context, causal inference techniques, such as those used in randomized controlled experiments, play a fundamental role in obtaining reliable insights. However, these randomized controlled experiments face challenges such as high costs and extended duration, while observational data (collected without deliberate manipulation) are a viable alternative but come with their own complexities, such as the lack of control over the applied treatment. The issue of counterfactuals, involving considering "What if an alternative action had been taken instead of the one observed?"becomes central in causal inference. In this research, a review of causality and causal inference concepts was conducted, followed by a detailed comparison between the Neyman-Rubin and Pearl causal modeling frameworks. Building upon the Neyman-Rubin framework, the theory behind the main models used in causal inference from observational data was reviewed. Another contribution of this research was the development of Pycausal-explorer, an open-source Python library that not only implements the models described and analyzed in this text but also allows integration with scikit-learn, one of the leading machine learning libraries in Python. With this, it is expected to promote the understanding and application of these models in observational data analysis in various fields, providing valuable insights based on more robust and solid cause-and-effect relationships.

Palavras chaves: causal inference, open source, observational data

Lista de Acrônimos

ATE *Average Treatment Effect*

ACE *Average Causal Effect*

CATE *Conditional Average Treatment Effect*

DAG *Directed Acyclic Graph*

IHDP *Infant Health and Development Program*

IPTW *Inverse Propensity Treatment Weighting*

ITE *Individual Treatment Effect*

MAPE *Mean Average Percentage Error*

PEHE *Precision in Estimation of Heterogeneous Effect*

Lista de Tabelas

Lista de Figuras

Sumário

1 Introdução

1.1 Motivação

Algoritmos de *machine learning* estão cada vez mais difundidos e encontram aplicações nas mais diversas áreas do conhecimento, passando desde a detecção de *spam* em *e-mails*[1], sistemas de recomendação[2], identificação de dígitos em imagens[3], até mesmo a detecção de patologias em imagens de raio-X[4]. Esses avanços foram possíveis graças aos recentes desenvolvimentos tecnológicos que viabilizaram expressivo aumento da capacidade de coleta, armazenamento e processamento de dados.

Apesar da crescente exploração e desenvolvimento dos métodos de machine learning em diversas aplicações, um aspecto ainda não muito explorado é a compreensão das relações causais entre as variáveis de entrada e a variável de resposta. Essa abordagem se torna cada vez mais essencial para responder à pergunta crucial "E se?"em relação aos modelos construídos. O paradoxo de Simpson[5] ilustra vividamente a importância desse estudo de inferência causal. Considere o cenário em que a taxa de sucesso em um tratamento específico é maior quando avaliada globalmente, mas, quando dividida por subgrupos, a taxa é menor em cada um desses subgrupos. Qual taxa é a correta? A global ou a dividida em subgrupos? Isso ocorre devido a uma variável de confusão que afeta a relação entre o tratamento e o resultado e a resposta depende da estrutura causal do sistema. O paradoxo de Simpson destaca como correlações superficiais podem conduzir a conclusões errôneas, ressaltando a necessidade de métodos que considerem as verdadeiras relações de causa e efeito [5].

Um exemplo do paradoxo de Simpson está no estudo conduzido por Charig et al. [6] onde foram comparadas as taxas de sucesso de dois tratamentos para cálculos renais, um usando cirurgia aberta e outro usando nefrolitotripsia percutânea. Cirurgia aberta (1972-1980) teve uma taxa de sucesso de 78% (273/350), enquanto a nefrolitotripsia percutânea (1980-1985) teve uma taxa de sucesso de 83% (289/350), mostrando uma melhora em relação ao uso da cirurgia aberta. No entanto, as taxas de sucesso se mostraram diferentes quando considerado o diâmetro da pedra. Isso mostrou que, para pedras com < 2 cm, 93% (81/87) dos casos de cirurgia aberta foram bem-sucedidos, enquanto apenas 83% (234/270) dos casos de nefrolitotripsia percutânea. Da mesma forma, para pedras com ≥ 2 cm, as taxas de sucesso foram de 73% (192/263) e 69% (55/80) para cirurgia aberta e nefrolitotripsia percutânea, respectivamente [7]. A tabela 1 mostra a taxa de sucesso por grupo e agregada. O principal motivo para ter essa inversão da taxa de sucesso quando olhado no agrupado e nos grupos separados é porque a probabilidade de realizar cirurgia aberta ou nefrolitotripsia percutânea variou conforme o diâmetro das pedras. Pedras menores (ou seja, menos complexos) tiveram maior probabilidade de ter a nefrolitotripsia pergutânea como tratamento, enquanto pedras maiores tiveram a cirurgia aberta como

Tabela 1: Tabela comparativa do índice de sucesso do tratamento de cálculos renais

	Cirurgia Aberta	Nefrolitotripsia Percutânea
Pedras $< 2cm$	93% (81/87)	83% (234/270)
Pedras $\geq 2cm$	73% (192/263)	69% (55/80)
Total	78% (273/350)	83% (289/350)

tratamento dominante.

Questões como se o ato de fumar realmente causa um aumento na propensão ao câncer de pulmão, ou se essa correlação é meramente uma consequência de outras características, ganham destaque. Para tais perguntas, a mera identificação de correlações entre variáveis por meio de algoritmos de *machine learning* não é suficiente. É essencial discernir as relações de causa e efeito do que está sendo estudado. Essa capacidade não apenas permite avaliar a eficácia de tratamentos, mas também pode orientar a formulação de políticas públicas embasadas em entendimentos dos mecanismos que geram esses resultados.

Nesse contexto, as técnicas de inferência causal, como aquelas empregadas em experimentos controlados aleatórios, desempenham um papel fundamental na obtenção de *insights* confiáveis. Apesar da importância dos experimentos controlados, eles enfrentam desafios significativos, como altos custos, duração prolongada e possíveis problemas de seleção de amostras.

Em contraste, a abordagem de dados observacionais (conjuntos de dados coletados sem a manipulação deliberada das variáveis de interesse) surge como uma alternativa viável. No entanto, os dados observacionais apresentam suas próprias complexidades, como a falta de controle sobre o tratamento aplicado e as interações subjacentes. Esses conjuntos de dados levantam a questão fundamental do contrafactual: "E se uma ação alternativa tivesse sido tomada em vez daquela observada?" Esta questão inerente à inferência causal é desafiadora, uma vez que os resultados contrafactuais nunca podem ser diretamente observados.

Durante a realização desta pesquisa, foi identificada uma carência de bibliotecas de inferência causal com foco em dados observacionais, o que provavelmente prejudica os estudos e atrasado o desenvolvimento de aplicações dessas técnicas nas mais diversas áreas. Abordagens convencionais muitas vezes carecem de facilidade de implementação, impedindo sua adoção generalizada. Isso limita a capacidade das empresas de aproveitar os benefícios das análises causais em suas operações, tomada de decisões e desenvolvimento de estratégias.

Para endereçar essa lacuna, esta pesquisa concentra-se em estudos do estado da arte e na criação da Pycausal-explorer, uma nova biblioteca em Python voltada para estudos

e desenvolvimento de aplicações de inferência causal.

Com o lançamento desta biblioteca, o objetivo é fornecer uma ferramenta de código aberto abrangente para a exploração da inferência causal que estimule a compreensão e aplicação nas mais diversas áreas, assim como desenvolvimentos complementares no tema inferência causal. Ao integrar essa funcionalidade com o já consolidado framework sklearn, a biblioteca oferecerá uma experiência familiar para profissionais já acostumados com essa plataforma. Isso reduzirá a curva de aprendizado e incentivará a adoção generalizada dessas técnicas, ampliando o potencial para *insights* valiosos, descobertas substanciais e tomada de decisões fundamentadas em relações de causa e efeito robustas e confiáveis.

1.2 Objetivos

Esta pesquisa de mestrado tem como objetivo geral mapear o estado da arte em inferência causal com foco em dados observacionais, assim como elaborar e disponibilizar a biblioteca de código aberto Pycausal-explorer com os principais métodos identificados, a fim de contribuir para a ampliação do conhecimento e desenvolvimento de novas aplicações nas mais diversas áreas. Para tal, esta pesquisa teve os seguintes objetivos específicos:

1. **Exploração de Conceitos Fundamentais**: Investigar conceitos fundamentais que sustentam a inferência causal, como correlação, causalidade, contrafactuais e efeitos de tratamento. Essa exploração permite discernir os princípios que distinguem a causalidade da mera correlação e proporcionar um entendimento sólido dos desafios inerentes à inferência causal.

2. **Aprofundamento na Teoria de Inferência Causal**: Explorar em detalhes a teoria da inferência causal, o que envolverá a análise das premissas, a formulação dos estimadores de efeito causal e a exploração das limitações e alcance dos modelos.

3. **Revisão dos Principais Modelos de Inferência Causal**: Realizar uma revisão crítica dos modelos de inferência causal mais relevantes e amplamente utilizados na análise de dados observacionais. Essa revisão incluiu a análise das abordagens baseadas em *matching*, árvores de decisão, *metalearners* e outras técnicas, avaliando tanto a teoria subjacente quanto a aplicação prática.

4. **Desenvolvimento de Biblioteca Open-Source**: Elaborar, implementar e disponibilizar a biblioteca open-source Pycausal-explorer que englobe os modelos de inferência causal abordados na revisão teórica. A disponibilização dessa biblioteca visa democratizar o acesso a esses métodos, tornando-os acessíveis e aplicáveis em cenários nas mais diversas áreas.

5. **Exemplo de Uso**: Demonstrar o uso de uma biblioteca em um exemplo de uso, ilustrando como ela pode ser utilizada para obter conclusões na área da educação.

Neste estudo, foi realizada uma análise para entender como a renda per capita de uma família afeta a nota do ENEM de egressos do ensino médio, influenciando o acesso ao ensino superior nas universidades públicas.

Ao cumprir esses objetivos, espera-se que esta pesquisa contribua para a disseminação do conhecimento sobre inferência causal, aprofundando a compreensão da teoria e fornecendo ferramentas práticas que permitam que as mais diversas áreas utilizem eficazmente esses métodos em suas análises de dados observacionais, ampliando o potencial para *insights* valiosos, descobertas substanciais e tomada de decisões fundamentadas em relações de causa e efeito mais robustas e confiáveis.

1.3 Estrutura do Texto

Os demais capítulos desta dissertação estão organizados da seguinte forma:

- **Capítulo 2**: Aprofundamento dos fundamentos teóricos da inferência causal. No capítulo é discutida a teoria por trás dos modelos de inferência causal e, em particular, os modelos de desfechos potenciais de Neyman-Rubin e o framework proposto por Pearl de modelos causais estruturais. Uma análise comparativa entre esses dois frameworks é conduzida para destacar suas semelhanças e diferenças. Essa exploração conceitual estabelece a base necessária para uma compreensão sólida dos métodos abordados posteriormente no texto. A exploração desses conceitos fornece a base necessária para a compreensão dos métodos abordados posteriormente no trabalho.

- **Capítulo 3**: Apresentação e detalhamento dos principais métodos de cálculo de efeito de tratamento utilizados em dados observacionais. São explicadas as abordagens baseadas em *matching*, árvores de decisão, *metalearners* e outras técnicas relevantes. A teoria de cada método é exposta, assim como exemplos práticos de aplicação.

- **Capítulo 4**: descrição do processo de desenvolvimento da biblioteca de inferência causal. São analisadas bibliotecas relacionadas e qual a motivação por trás de criar uma nova biblioteca. São detalhados os passos envolvidos na implementação dos modelos abordados na revisão teórica, bem como os desafios enfrentados e as soluções encontradas. Também é descrito o processo de testes e validação da biblioteca.

- **Capítulo 5**: conclusão do trabalho. São resumidos os principais resultados, *insights* e contribuições do estudo. Além disso, são discutidas possíveis trabalhos futuros. A importância da nova biblioteca e seu potencial impacto na indústria também são reiterados neste capítulo final.

2 Conceitos e Fundamentos Teóricos

2.1 Diagramas Causais

Diagramas causais são grafos direcionados acíclicos que auxiliam no entendimento de como as variáveis se relacionam. Consiste em nós e setas unidirecionais ligando esses nós. Cada nó representa uma variável e a seta qual relação de causalidade entre elas. A figura 1 mostra um exemplo de diagrama causal. A variável X tem uma relação de causalidade com a variável T e Y, enquanto a T tem uma relação de causalidade apenas com a Y. Isso quer dizer que, enquanto uma mudança em X pode causar uma mudança em T e em Y, uma mudança em T pode causar apenas mudanças em Y.

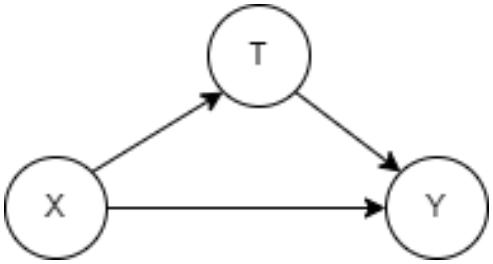

Figura 1: Grafo que mostra relação entre causa e efeito das variáveis.

Os grafos são direcionados, pois implica uma relação de causa e efeito e acíclicos, pois uma variável não pode causar ela mesma [8].

Outro grafo pode ser observado na figura 2. Nele agora as variáveis T e Y não estão com um caminho direto de causa e efeito, mas existe um caminho associativo passando por X. Isso quer dizer que variando X, pode-se observar variação tanto em T como em Y, mas que T e Y não são diretamente relacionadas. Também é possível identificar a variável W que é independente das outras variáveis, por não ter nenhuma flecha saindo dela ou entrando.

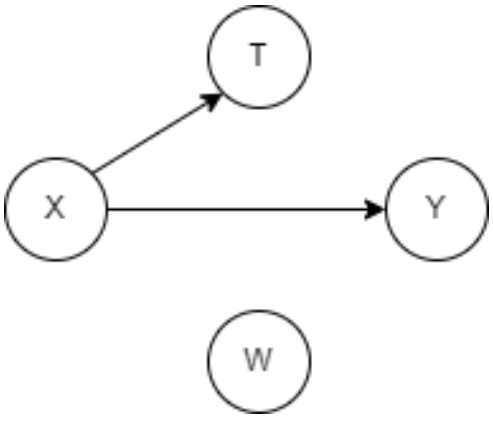

Figura 2: Grafo que mostra relação entre causa e efeito das variáveis, agora com a variável T e Y parcialmente independentes e a variável W independente.

2.2 Independência, Efeitos Associativos e Causais

Quando se estuda causa e efeito entre duas variáveis, primeiro é preciso entender qual o efeito associativo, ou grau de dependência, que as duas variáveis têm. Sendo X e Y duas variáveis, estas são chamadas de independentes quando as duas variáveis se comportam de forma em que o valor de uma variável não influencia no valor da outra e vice-versa. Pode-se definir variáveis independentes por meio da probabilidade condicional:

$$P(X|Y) = P(X) \tag{1}$$

Um exemplo de independência está na sucessiva jogada de uma moeda. Independente de qual lado da moeda caiu em jogadas anteriores, a probabilidade de sair cara é sempre a mesma $P(H) = 0.5$, pois as jogadas anteriores não influenciam nas jogadas futuras. Uma ocasião diferente está na retirada de cartas sem reposição. A probabilidade de sair uma carta preta depende de quais cartas saíram anteriormente, então os eventos são dependentes, ou, associados [9].

Associação e causalidade não devem ser confundidas. Se X causa Y, existe uma associação entre as duas variáveis. Mas uma associação pode surgir sem, necessariamente, existir uma ligação de causa e efeito. Suponha que saiu um artigo que mostrou que tomar mais de 4 copos de café por dia está associado a um risco menor de adquirir câncer de pele. Essa associação não está necessariamente relacionada com o café ter algum mecanismo que evita o câncer de pele. Uma interpretação poderia ser que pessoas que tomam muito café ficam mais tempo em casa e assim ficam menos expostas ao Sol, que de fato reduz o risco de câncer de pele [9]. A figura 3 mostra o grafo do sistema:

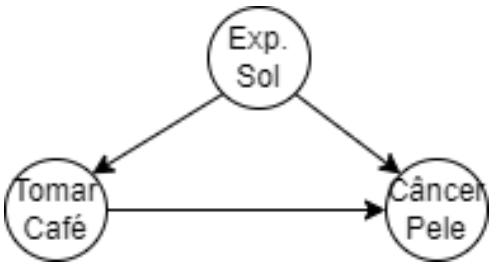

Figura 3: Grafo contendo o relacionamento entre exposição ao Sol, hábito de tomar muito café e câncer de pele

No grafo existe uma seta saindo de "tomar café"até "câncer de pele". Essa seta mostra o suposto efeito causal entre as duas variáveis relacionadas. Mas existe a variável "exposição ao Sol"que afeta tanto "tomar café"quanto "câncer de pele". O caminho de "tomar café"até "câncer de pele"passando por "exposição ao Sol"é a associação *confounding*, ou de confusão. A associação total é dada pela associação causal mais a associação de confusão. Por isso o artigo não consegue inferir causalidade no contexto do texto, como poderia ser feito em um estudo controlado randomizado: todo o efeito associativo pode

vir pela exposição ao Sol, e nenhum efeito indo diretamente ao câncer de pele.

Um estudo controlado randomizado (ERC) é um tipo de pesquisa experimental muito utilizado na medicina para avaliar a eficácia e segurança de intervenções médicas. Ele envolve a alocação aleatória de participantes em grupos de tratamento e controle, sendo que o primeiro recebe a intervenção em estudo e o último, um placebo ou tratamento padrão. Os ECRs são frequentemente conduzidos de forma duplo-cega, onde tanto participantes quanto pesquisadores desconhecem a atribuição dos grupos. Os dados são coletados ao longo do tempo e analisados estatisticamente para determinar se há diferenças significativas nos resultados entre os grupos, com base nas quais conclusões são tiradas sobre a eficácia e segurança da intervenção. Atribuindo os grupos de forma aleatória, evita-se a existência de variáveis que possam estar correlacionadas com essa alocação, eliminando assim potenciais variáveis de confusão que afetem o resultado e o tratamento. Isso torna a diferença do resultado entre os grupos a diferença causal do tratamento.

2.3 Frameworks de Modelagem Causal

Por se tratar de dados observacionais, não podemos usar técnicas como estudo controlado randomizado para inferir efeito de tratamento nas variáveis de interesse. O desafio está em como considerar efeitos associativos e extrair apenas o componente causal. Isso se deve ao fato de existirem variáveis *confounding*, variáveis que se relacionam tanto com a variável de tratamento quanto com a variável resposta de interesse. A modelagem causal é um campo complexo com vários frameworks, sendo os dois mais reconhecidos e influentes os de Pearl e de Neyman-Rubin, considerados pilares fundamentais da modelagem causal com grande impacto na investigação e, na prática, no campo da inferência causal. A escolha de um ou outro por pesquisadores, ou profissionais se dá com base na natureza dos dados, nas questões de investigação e nos pressupostos que se alinham com o contexto específico.

2.3.1 Modelos Causais Estruturais

O Framework de Modelagem Causal proposto por Pearl [10] para estudo de inferência causal, também conhecido como Modelos causais estruturais (*structured causal models* - SCM, em inglês), se apoia em grafos direcionados acíclicos para representar as relações causais entre variáveis.

Formalmente, um modelo causal estrutural consiste em dois conjuntos de variáveis U e V e um conjunto de funções f que atribui cada variável em V um valor baseado nos valores das outras variáveis do modelo. Uma variável X é uma causa direta de Y se X estiver na função que atribui valor a Y. X causa Y se ele for uma causa direta de Y ou de qualquer causa direta de Y.

As variáveis em U são chamadas de variáveis exógenas, o que significa que elas são

externas ao modelo; não sendo explicadas como são causadas. As variáveis em V são endógenas. Cada variável endógena em um modelo é descendente de pelo menos uma variável exógena. Variáveis exógenas não podem ser descendentes de nenhuma outra variável; elas não têm ancestrais e são representadas como nós raiz em grafos. Se é conhecido o valor de cada variável exógena, então, usando as funções em f, é possível determinar o valor de cada variável endógena.

Cada modelo causal estrutural é associado a um grafo, em que as variáveis U e V são representadas como nós e os vértices entre os nós como as funções f. Se em um grafo G a variável X é causa direta de Y, o vértice é direcionado de X para Y. A figura 4 mostra um exemplo de grafo representando um modelo causal estrutural.

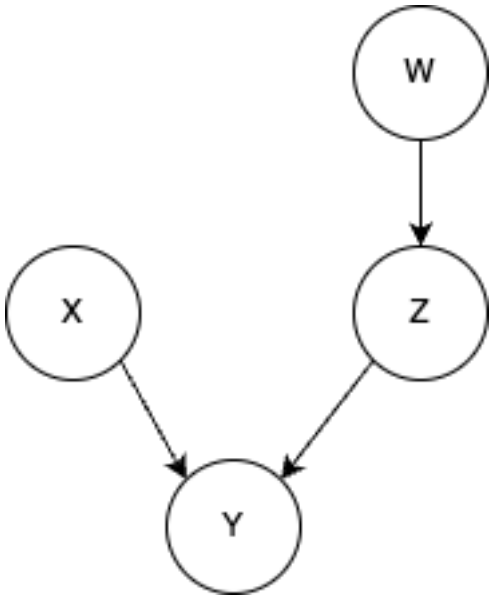

Figura 4: Exemplo de grafo representando um modelo causal estrutural

Nesse caso as variáveis X e W são variáveis exógenas por serem nós raiz, as variáveis Z e Y são endógenas por serem descendentes de pelo menos outra variável. Tanto X como Z como W são causas direta de Y. O SCM é especificado da seguinte forma:

- $U = \{X, W\}$, $V = \{Y, Z\}$
- $f_Z = f(W)$
- $f_Y = g(X, Z)$

2.3.1.1 Decomposição de Produtos

Uma vantagem no uso de grafos está na eficiência em demonstrar probabilidades conjuntas. Para qualquer modelo cujo grafo é acíclico, a distribuição conjunta das variáveis pode ser calculada pelo produto da probabilidade das variáveis x_i dado os pais (nós cujos descendentes é o x_i).

$$P(x_1, x_2, \ldots, x_n) = \prod_i P(x_i|pa_i) \tag{2}$$

No exemplo da figura 4 a probabilidade conjunta é tida como:

$$P(X, Y, Z, W) = P(X)P(W)P(Z|W)P(Y|X, Z) \tag{3}$$

Dessa forma não é necessário calcular a frequência de todas as combinações possíveis de (x, y, z, w) para conseguir calcular a probabilidade conjunta, apenas as de $P(X)$, $P(W)$, $P(Z|W)$ e $P(Y|X, Z)$, diminuindo consideravelmente o espaço de possibilidades necessário para seu cálculo.

O uso de grafos também facilita enxergar quais variáveis são dependentes, independentes e condicionalmente dependentes:

- Regra 0 - Dependência de nós: duas variáveis X e Y que estiverem ligadas por um vértice são consideradas dependentes ($P(Y|X) \neq P(Y)$).
- Regra 1 - Dependência condicional em cadeia: duas variáveis X e Y são condicionalmente independentes dado Z, se houver apenas um caminho unidirecional entre X e Y, e Z for um conjunto de variáveis que intercepta esse caminho.
- Regra 2 - Independência condicional em bifurcações: Se uma variável X é uma causa comum das variáveis Y e Z, e existe apenas um caminho entre Y e Z, então Y e Z são independentes condicionalmente em relação a X.

A figura 5 mostra como a dependência condicional em cadeia funciona. No caso, a variável Y é independente de X dado Z.

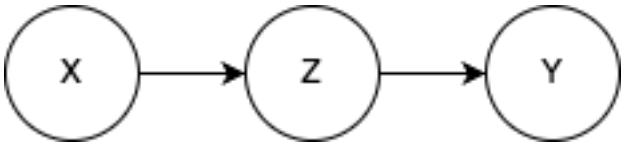

Figura 5: Grafo demonstrando a dependência condicional em cadeia

A figura 6 mostra como a dependência condicional em bifurcações funciona. No caso, a variável X é independente de Z dado Y.

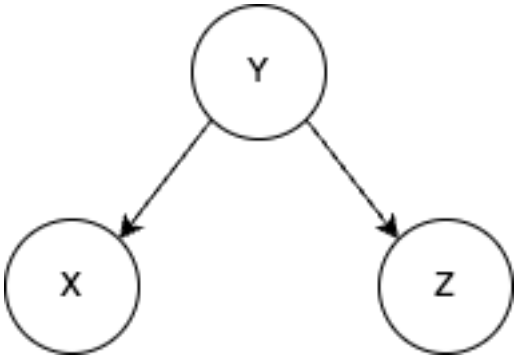

Figura 6: Grafo demonstrando a dependência condicional em bifurcações

2.3.1.2 Colisores

Além das configurações de cadeia e bifurcação, uma terceira configuração de grafos pode ser considerada, que é o caso quando um nó recebe vértices de dois ou mais nós. A figura 7 mostra a configuração, em que a variável Y tem como causadores diretos X, Z.

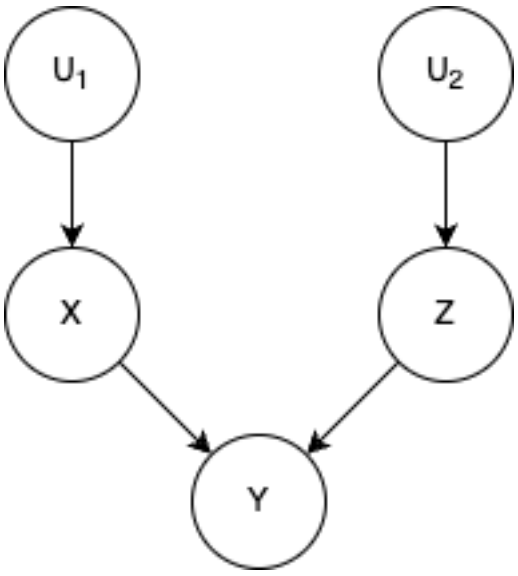

Figura 7: Grafo demonstrando colisores

Considerando U_1 e U_2 independentes, X e Z também são (não existe nenhum caminho ligando diretamente X ao Z). Mas quando condicionado a Y, as duas variáveis se tornam dependentes, o que a princípio parece ser contra-intuitivo. Em uma hipótese em que $Y = X + Z$, saber que $X = 3$ não altera em nada o que se sabe sobre Z, já que as duas variáveis são independentes. Mas ao saber que $Z = 10$, existe apenas um conjunto de tuplas (X, Y) que torna o resultado verdade, tornando então as duas variáveis, que a princípio eram independentes, dependentes condicionadas a Y.

A regra de colisores pode ser escrita da seguinte forma:

- Regra 3 - Colisores: Se uma variável Y é o nó de colisão entre duas variáveis X e Z, e existe apenas um caminho entre X e Z, então X e Z são independentes incon-

dicionalmente, mas são dependentes condicionalmente em relação a Y e quaisquer descendentes de Y.

2.3.1.3 D-separação

Modelos causais geralmente são mais complexos que os grafos usados como exemplos até então. Existem muito mais variáveis envolvidas e múltiplos caminhos entre elas. Ter um processo para analisar esse tipo de grafo usando as regras já estabelecidas se torna necessário. D-separação é o processo que é criado a partir dessas regras.

Esse processo nos permite determinar, para qualquer par de nós, se os nós estão d-conectados, ou seja, existe um caminho de conexão entre eles, ou d-separados, o que significa que não há tal caminho. Quando um par de nós está d-separado, as suas variáveis são independentes; enquanto um par de nós d-conectado suas variáveis estão provavelmente dependentes.

Dois nós, X e Y, estão d-separados se todos os caminhos entre eles (caso existam) estiverem bloqueados, caso contrário eles estão d-conectados. Alguns tipos de nós podem bloquear o caminho. Caso não sendo condicionada nenhuma variável, apenas colisores podem bloquear caminho. Caso seja condicionado a um conjunto de variáveis C, os seguintes tipos de nós podem ser bloqueados:

- Colisores em que seu nó nem seus descendentes estejam em C
- Cadeias ou bifurcações em que seus nós estejam em C

A lógica por trás da d-separação se baseia na ideia de que, ao bloquear determinados caminhos em um grafo causal, é possível isolar o efeito direto de uma variável sobre outra, evitando a influência de variáveis intermediárias. Isso é crucial para entender as relações causais subjacentes e determinar a independência entre variáveis em um sistema complexo.

O conceito de colisores merece uma atenção especial. Quando um nó atua como um colisor em um caminho, significa que ele é um ponto onde diferentes fluxos de influência se encontram. Isso é especialmente importante quando considerado o condicionamento em um conjunto de variáveis. Se um colisor ou qualquer um de seus descendentes estiverem incluídos no conjunto de condicionamento, o caminho entre as variáveis de interesse se torna aberto, tornando variáveis antes dependentes, independentes.

Pode-se então definir d-separação da seguinte forma:

Definição 2.1 (D-separação)**.** Um caminho p está bloqueado por um conjunto de nós Z se e somente se:

1. p contém uma cadeia de nós $A \rightarrow B \rightarrow C$ ou um garfo $A \leftarrow B \rightarrow C$, tal que o nó intermediário B esteja em Z (ou seja, B está condicionado), ou

2. p contém um colisor $A \rightarrow B \leftarrow C$ tal que o nó de colisão B não esteja em Z, e nenhum descendente de B esteja em Z.

A figura 8 mostra um grafo mais complexo onde é possível fazer uma análise de d-separação. Por ter uma cadeia $A \rightarrow B \rightarrow C$ os nós A e C são independentes se condicionado a B. Por ter um colisor em D, os nós G e E são independentes de A, B, C e F. Caso seja condicionado a D ou H, o caminho se abre e essas variáveis se tornam dependentes. Em C existe uma bifurcação, então F é independente de D e H se condicionado C.

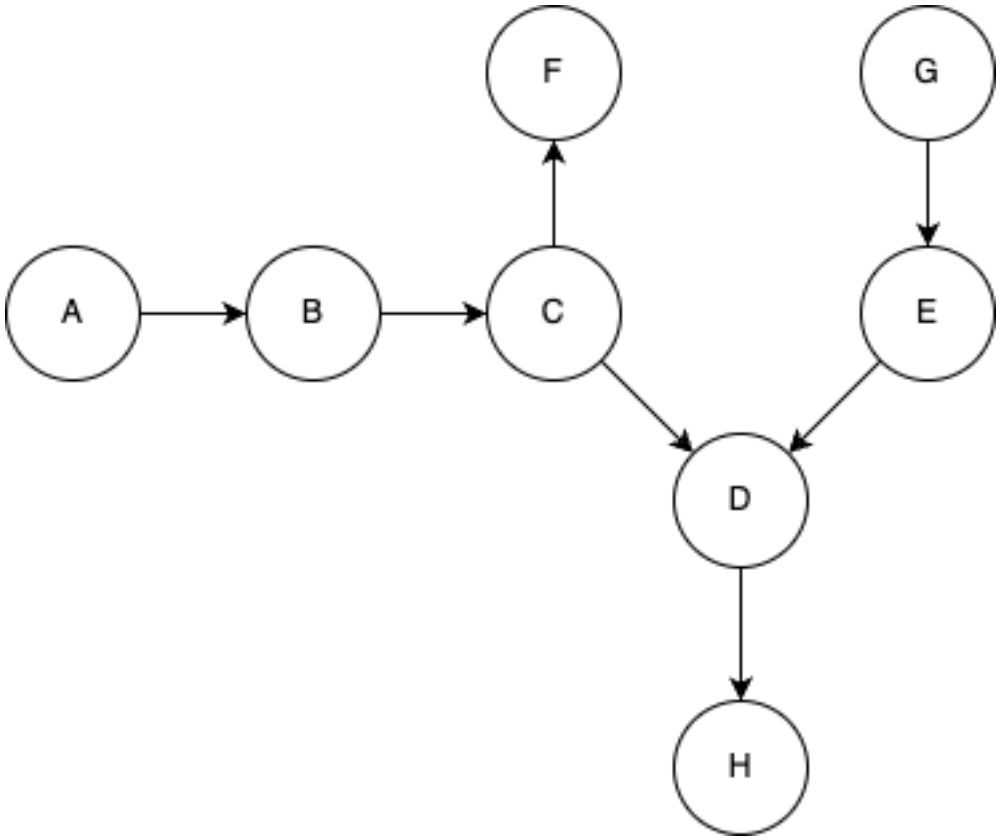

Figura 8: Exemplo de um grafo complexo

2.3.1.4 Critério da Porta dos Fundos

Um dos principais objetivos no estudo de inferência causal está em entender como as variáveis se comportam quando é feita intervenção em uma variável de tratamento T. Quando uma intervenção é feita em uma variável T, apenas a variável de interesse deve ser manipulada, deixando as outras variáveis estáticas. Isso difere de simplesmente calcular a probabilidade condicional $P(Y|T)$ pois estamos apenas estreitando a visão para apenas um subgrupo, e não criando intervenções no grupo que não foi realizado o tratamento. Para diferenciar os dois casos é uada a notação $P(Y|do(X = x))$ para representar a distribuição de Y dado se toda população alterasse X para o valor x.

Para se calcular o efeito causal de uma variável T em um desfecho $Y = y$, podemos utilizar a seguinte fórmula:

$$ACE = P(Y = y|do(T = 1)) - P(Y = y|do(T = 0)) \tag{4}$$

Para realizar o estudo de como a intervenção T afeta Y, precisamos isolar os efeitos de caminhos associativos e apenas levar em consideração o caminho causal. O critério da porta dos fundos nos permite identificar quais variáveis precisamos condicionar para calcular o efeito causal.

Definição 2.2 (Critério da Porta dos Fundos)**.** Dado um par ordenado de variáveis (X, Y) em um grafo acíclico dirigido G, um conjunto de variáveis Z satisfaz o critério da porta dos fundos em relação a (X, Y) se nenhum nó em Z for descendente de X, e Z bloqueia todos os caminhos entre X e Y que contenham uma seta em direção a X.

Esse critério explica que, para entender a causa de X em Y, é necessário bloquear todos os caminhos que levam a essas duas variáveis, menos o caminho $X \rightarrow Y$. Se satisfeito, o efeito causal $P(Y|do(X))$ pode ser calculado como:

$$P(Y|do(X)) = \sum_{z} P(Y|X, Z)P(Z) \tag{5}$$

Isso transforma probabilidades causais em estimadores estatísticos, que podem ser obtidos através dos dados observacionais. O critério determina quais variáveis condicionar e, mais importante, quais não condicionar para conseguir estimar o efeito causal.

2.3.1.5 Critério da Porta da Frente

O critério da porta dos fundos é uma ferramenta poderosa para transformar estimadores causais em estimadores estatísticos, mas tem limitações. Quando não é possível medir variáveis que são confundidoras, não é possível satisfazer o critério da porta dos fundos. Apesar disso, é possível ter casos em que mesmo não conseguindo medir todas as variáveis confundidoras, é possível ainda calcular o efeito causal. A figura 9 mostra um grafo no qual a variável U não é mensurável. É possível ainda mensurar o efeito causal do tratamento T com relação a Y com o auxílio de M, que não é diretamente afetada pela variável não mensurável.

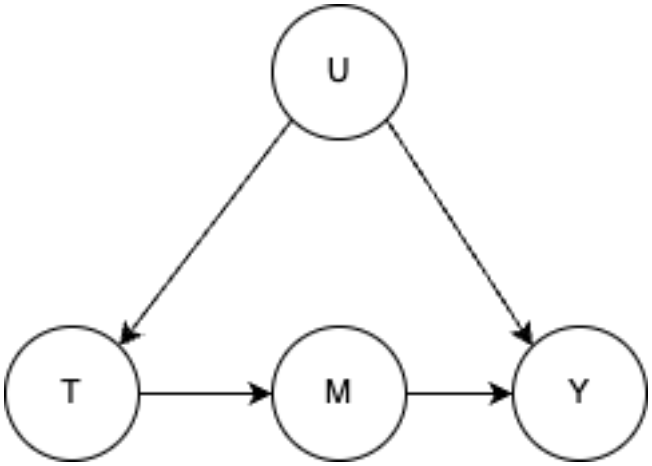

Figura 9: Exemplo de grafo com variável imensurável

O primeiro passo está em calcular o efeito causal de t em m $P(m|do(t))$.

$$P(m|do(t)) = P(m|t) \tag{6}$$

O segundo passo está no cálculo de $P(y|do(m))$. Como existe relação com a variável não mensurável U, usa-se o critério da porta dos fundos condicionando em T:

$$P(y|do(m)) = \sum_t P(y|m,t)P(t) \tag{7}$$

Por fim o primeiro e segundo passo são unificados para calcular o valor de $P(Y|do(T))$.

$$P(y|do(t)) = \sum_m P(m|do(t))P(y|do(m)) \tag{8}$$

$$P(y|do(t)) = \sum_m P(m|t) \sum_{t'} P(y|m,t')P(t') \tag{9}$$

Com isso os efeitos causais são calculados usando o critério da porta da frente. O critério da porta da frente é definido da seguinte maneira:

Definição 2.3 (Critério da Porta da Frente)**.** Um conjunto de variáveis Z é considerado a satisfazer o critério da porta da frente em relação a um par ordenado de variáveis (X, Y) se:

1. Z intercepta todos os caminhos direcionados de X para Y.
2. Não há caminho de porta dos fundos de X para Z.
3. Todos os caminhos da porta dos fundos de Z para Y são bloqueados por X.

2.3.1.6 Do-calculus

Do-calculus é uma ferramenta poderosa, discutida por Pearl em [11] cujo objetivo é identificar causalidade entre as variáveis que são identificáveis, independente se obedecem

ao critério da porta dos fundos ou da porta da frente. Ou seja, determinar $P(Y|do(T = t), X = x)$ caso seja identificável.

Antes de introduzir as regras do do-calculus é necessária a definição de alguns conceitos com relação a grafos manipulados:

Definição 2.4 (Grafos Manipulados)**.** Sendo G uma DAG e X um nó dela com pais e descendentes, temos:

- $G_{\overline{X}}$ o grafo onde as arestas de X com seus pais são excluídas
- $G_{\underline{X}}$ o grafo onde as arestas de X com seus descendentes são excluídas

Com essas definições é possível enunciar as 3 regras do do-calculus:

- Regra 1: $P(y|do(t), z, w) = P(y|do(t), w)$ se $\{Y \perp\!\!\!\perp_{G_{\overline{T}}} Z|T, W\}$
- Regra 2: $P(y|do(t), do(z), w) = P(y|do(t), z, w)$ se $\{Y \perp\!\!\!\perp_{G_{\overline{T}\underline{Z}}} Z|T, W\}$
- Regra 3: $P(y|do(t), do(z), w) = P(y|do(t), w)$ se $\{Y \perp\!\!\!\perp_{G_{\overline{T}\overline{Z(W)}}} Z|T, W\}$ onde $Z(W)$ é o conjunto de nós de Z que não são ancestrais de nenhum nó de W em $G_{\overline{T}}$

A primeira regra explicita a independência de Z em $P(y|do(t), z, w)$ quando Y é d-separado de $Z|T, W$ no grafo em que as arestas que chegam em T são excluídas. Ela é uma generalização da d-separação em distribuições intervencionais, enquanto a segunda regra é uma generalização do critério da porta dos fundos.

As 3 regras de do-calculus torna completa a busca por estimadores causais que são identificáveis, ou seja, se um estimador causal é identificável, existe um número de passos finito de transformações das regras que transforma em um estimador estatístico [12].

2.3.2 Framework de Modelagem Causal de Neyman-Rubin

O Framework de Modelagem Causal de Neyman-Rubin, cada vez mais utilizado para tratar de dados observacionais, consiste em um modelo não-paramétrico em que cada unidade tem dois desfechos potenciais, um em que a unidade recebeu o tratamento e outro em que a unidade não recebeu [13]. Sendo $Y_i(1)$ o desfecho potencial quando a unidade é tratada e $Y_i(0)$ o desfecho potencial quando a unidade não é tratada, o efeito causal, também chamado de efeito de tratamento individual (ITE) é definido como:

$$\tau_i = Y_i(1) - Y_i(0) \tag{10}$$

Como $Y_i(0)$ e $Y_i(1)$ não podem ser observados ao mesmo tempo (ou a unidade recebeu o tratamento, ou não recebeu), isso é considerado como um problema de dados faltantes. No âmbito da inferência causal, também é chamado de *problema fundamental da inferência*

causal. Se uma unidade recebeu o tratamento $T = 1$, $Y_i(1)$ é chamado de factual e $Y_i(0)$ de contrafactual. O oposto ocorre se a unidade não recebeu o tratamento $T = 0$.

Para poder calcular o resultado da equação 10 em dados observacionais é necessário adotar algumas premissas: ignorabilidade, ignorabilidade condicionada, positividade e estabilidade do valor unidade-tratamento. Estas premissas e seu uso para o cálculo do efeito de tratamento médio são descritos nos itens a seguir.

2.3.2.1 Ignorabilidade

A premissa da ignorabilidade assume que a atribuição do tratamento é independente dos desfechos potenciais $Y_i(0)$ e $Y_i(1)$, ou seja:

$$(Y(0), Y(1)) \perp\!\!\!\perp T \tag{11}$$

Com essa premissa é possível calcular o efeito de tratamento médio (ATE) através de estimativas associativas:

$$\begin{aligned} E[\tau] &= E[Y(1) - Y(0)] \\ &= E[Y(1)] - E[Y(0)] \\ &= E[Y(1)|T = 1] - E[Y(0)|T = 0] \text{ (ignorabilidade)} \\ &= E[Y|T = 1] - E[Y|T = 0] \end{aligned} \tag{12}$$

Com essa premissa satisfeita é possível calcular o efeito de tratamento apenas tirando a esperança do grupo tratado menos a do grupo não tratado. Essa é a premissa satisfeita quando é feito um *randomized controlled trial*. Essa premissa também pode ser chamada de permutabilidade, pois assegura que se o grupo dos não tratados e o dos tratados fossem trocados, ou seja, o grupo dos tratados não tivesse recebido o tratamento e vice-versa, o resultado continuaria o mesmo (pois $E[Y(1)] = E[Y(1)|T = 1] = E[Y(1)|T = 0]$).

2.3.2.2 Ignorabilidade Condicionada

Pela premissa da ignorabilidade ser muito forte, raramente ela acontece em dados observacionais. Uma premissa mais relaxada assume a independência dos desfechos potenciais com o tratamento condicionada nas outras variáveis:

$$(Y(0), Y(1)) \perp\!\!\!\perp T|X \tag{13}$$

Essa premissa garante que todas as variáveis confundidoras (que se relacionam tanto com o tratamento quanto com a variável resposta) estão devidamente condicionadas. Com essa premissa é possível calcular o ATE condicionado da seguinte forma:

$$\begin{aligned} E[Y(1) - Y(0)|X] &= E[Y(1)|X] - E[Y(0)|X] \\ &= E[Y(1)|T = 1, X] - E[Y(0)|T = 0, X] \text{ (ignorabilidade cond.)} \\ &= E[Y|T = 1, X] - E[Y|T = 0, X] \end{aligned} \tag{14}$$

Calcula-se o valor do ATE através da marginalização em X:

$$\begin{aligned} E[Y(1) - Y(0)] &= E_X[E[Y(1) - Y(0)|X]] \\ &= E_X[E[Y|T = 1, X] - E[Y|T = 0, X]] \end{aligned} \tag{15}$$

Apesar de ser uma premissa importante, não é possível garantir que todas as variáveis confundidoras estão sendo condicionadas (podem ter variáveis que afetam tanto o tratamento quanto a variável resposta que não estão sendo monitoradas). Por isso, essa é uma premissa não-testável.

2.3.2.3 Positividade

A premissa da positividade diz que para todos os valores x das variáveis presentes na população (ou seja, que $P(X = x) > 0$), tem-se que:

$$0 < P(T = 1|X = x) < 1 \tag{16}$$

Essa premissa garante que existem exemplos de unidades tratadas e não tratadas em todo o espaço das variáveis. Note que quanto menos variáveis eu condicionar meu modelo causal, mais fácil é de garantir essa premissa, enquanto na premissa da ignorabilidade condicional, quanto mais variáveis condicionadas, mais fácil é de garanti-la. Esse *trade-off* está relacionado à maldição da dimensionalidade.

Essa premissa é importante, pois para o cálculo do ATE é necessária a marginalização em X:

$$E_X[E[Y|T = t, X]] = \sum_x P(X = x)(\sum_y y * \frac{P(Y = y, T = t, X = x)}{P(T = t|X = x) * P(X = x)}) \tag{17}$$

No desenvolvimento da equação 17 existe a divisão por $P(T = t|X = x)$, tendo então a necessidade de ser um valor positivo.

2.3.2.4 Estabilidade do valor unidade-tratamento

A última premissa é dividida em duas partes. A primeira é a não interferência, em que a atribuição de tratamento de uma unidade não afeta o desfecho de outra.

$$Y_i(t_0, t_1, ..., t_{i-1}, t_i, t_{i+1}, ..., t_n) = Y_i(t_i) \tag{18}$$

A segunda parte diz respeito à consistência da variável resposta. Se a unidade recebeu a o tratamento $T = t$, então o único resultado possível é $Y = Y(t)$. Ela também indica que não há múltiplas versões para o tratamento $T = 1$.

$$T = t \implies Y = Y(t), t = \{0, 1\} \tag{19}$$

2.3.2.5 Usando as Premissas

A demonstração abaixo mostra como cada uma das premissas são utilizadas no cálculo do efeito de tratamento médio:

$$\begin{aligned} E[Y(1) - Y(0)] &= E[Y(1)] - E[Y(0)] \\ &= E_X[E[Y(1)|X] - E[Y(0)|X]] \text{ (positividade)} \\ &= E_X[E[Y(1)|T = 1, X] - E[Y(0)|T = 0, X]] \text{ (ignor. cond.)} \\ &= E_X[E[Y|T = 1, X] - E[Y|T = 0, X]] \text{ (consistência)} \end{aligned} \tag{20}$$

Com essas premissas, é possível criar um estimador causal a partir de estimadores estatísticos. Para o cálculo desses estimadores existem diversas técnicas de inferência causal, como regressão linear, métodos de combinação, ponderação ou no uso de *metalearners*.

2.3.3 Comparação entre os Frameworks

Ambos os *frameworks* têm como objetivo primordial a análise e identificação de relações causais utilizando dados observacionais. No entanto, suas abordagens e bases teóricas apresentam distinções significativas.

O *framework* de Pearl adota uma abordagem mais estrutural, fundamentada na teoria dos grafos causais e no do-calculus. Pearl propõe a representação das relações causais através de grafos direcionados, nos quais os nós representam variáveis e as arestas indicam influências direcionadas entre elas. O uso do do-calculus, uma linguagem formal para manipulação de expressões causais, possibilita a identificação e estimação dos efeitos do tratamento. Uma vantagem notável deste *framework* é que ele oferece ferramentas para lidar com vieses de seleção, permitindo a modelagem explícita de processos de confusão. No entanto, é necessário saber como é o grafo que se quer estudar para utilizar essa metodologia.

Por outro lado, o *framework* de Neyman-Rubin centra-se na ideia de desfechos potenciais. Esse enfoque envolve a criação de cenários hipotéticos em que cada indivíduo é avaliado sob duas condições: uma em que recebe o tratamento e outra em que não o recebe. Dessa forma, é possível estimar a diferença entre os desfechos, que representa o efeito causal do tratamento. Este *framework* exige a suposição de ignorabilidade, que pressupõe que a alocação do tratamento é independente do resultado potencial, dadas as covariáveis

observadas. O desafio muitas vezes reside na obtenção dessa condição de ignorabilidade, que não pode ser testada, tornando a aplicação do *framework* de Neyman-Rubin sensível a vieses não observados.

Estudos estão sendo conduzidos para compreender se existe uma equivalência entre os dois frameworks. O estudo de Markus [14] compara os dois modelos, ressaltando suas diferenças e apresentando razões para que os dois frameworks não tenham uma equivalência forte. Isso quer dizer que eles não estão abordando a mesma questão de formas diferentes, apesar de ser possível traduzir a linguagem de um framework para outro. No entanto, esse estudo é contestado por Weinberger em [15], apresentando argumentos contrários aos levantados por Markus e demonstrando que ainda não há consenso nessa área.

Após realizar o estudo e comparação entre os dois *frameworks*, decidiu-se escolher o Framework de Modelagem Causal de Neyman-Rubin para o aprofundamento da pesquisa e desenvolvimento da biblioteca. Os principais motivos que sustentam esta decisão são apresentados a seguir:

- Conhecimento Prévio do Grafo Causal: O *framework* de Pearl requer a especificação prévia de um grafo causal que representa as relações entre as variáveis. Isso significa que é necessário ter um conhecimento sólido das relações causais envolvidas no problema. Se não se tem certeza sobre as relações causais ou se o problema é complexo, pode ser difícil construir um grafo causal preciso. Já o modelo de Rubin se baseia na suposição de ignorabilidade condicional às covariáveis, o que pode ser mais flexível em cenários onde o conhecimento causal detalhado é limitado.

- Estrutura Estatística Tradicional: O *framework* de Rubin é fundamentado em notação estatística convencional, o que pode ser mais familiar e acessível para aqueles que já possuem experiência em trabalhar com métodos estatísticos tradicionais.

No entanto, é essencial reconhecer que a escolha entre os frameworks de Neyman-Rubin e de Pearl é altamente influenciada pelo contexto específico do problema e pelas informações disponíveis. Embora o framework de Pearl seja poderoso para representar relações causais complexas, utilizando grafos para expressar as relações entre as variáveis e adotando uma metodologia robusta para a identificação e cálculo dos efeitos causais, a opção pelo framework de Neyman-Rubin é respaldada pela sua aderência à abordagem estatística tradicional e pela sua adequação a diversos cenários de análise causal em dados observacionais. Portanto, a escolha depende das nuances do problema e das preferências metodológicas dos pesquisadores.

3 Principais Modelos de Inferência Causal

Nesta seção serão detalhados os métodos de inferência causal implementados na biblioteca:

3.1 Regressão Linear

O modelo de regressão linear é uma ferramenta amplamente utilizada na análise de dados para entender a relação entre uma variável dependente Y e uma ou mais variáveis independentes $X_1, X_2, \ldots, X_p$. Apesar de não ser um modelo específico de inferência causal, ele também pode ser utilizado para esse fim, buscando entender não apenas a relação estatística entre essas variáveis, mas também se há um efeito causal direto.

O modelo de regressão linear pode ser expresso como:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \beta_t T + \epsilon \tag{21}$$

onde:

- Y é o desfecho potencial.
- $X_1, X_2, \ldots, X_p$ são as variáveis independentes.
- T é a variável de tratamento.
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_p, \beta_t$ são os coeficientes de regressão que representam o efeito da variável independente correspondente.
- ϵ é o termo de erro, que captura a variabilidade não explicada pelo modelo.

Na perspectiva de inferência causal, o interesse está em determinar se a relação entre uma variável independente (aqui chamada de tratamento) T e a desfecho potencial Y é causal.

O uso do modelo de regressão linear pressupõe algumas premissas fundamentais para os resultados serem válidos e interpretáveis:

1. Linearidade: A relação entre as variáveis independentes e dependentes é assumida como linear. Desvios significativos dessa linearidade podem afetar os resultados.
2. Independência dos Resíduos: Os resíduos (diferença entre os valores observados e os valores previstos pelo modelo) devem ser independentes e não mostrar padrões sistemáticos.
3. Homocedasticidade: A variância dos resíduos deve ser constante em todos os níveis das variáveis independentes.

4. Normalidade dos Resíduos: A distribuição dos resíduos deve ser aproximadamente normal.

No caso, o efeito de tratamento médio $\hat{\tau} = E[Y|T = 1] - E[Y|T = 0] = \beta_t$, ou seja, o coeficiente de regressão associado ao tratamento indica qual o efeito de tratamento médio.

3.2 Inverse Propensity Treatment Weighting

Esse método, também chamado de IPTW, é um modelo que tem como origem a criação de uma pseudo-população onde o viés de seleção do tratamento é cancelado. Viés de seleção ocorre quando a forma como os indivíduos são selecionados para um grupo de tratamento ou controle não é aleatória, resultando em grupos que não são equivalentes em suas características.

3.2.1 Escore de Propensão

O IPTW é aplicado em várias etapas. Primeiro é necessário estimar o escore de propensão, que é a probabilidade da observação ter sido tratada dadas as variáveis X $e(X) = Pr[T = 1|X]$. Isso é geralmente feito usando um modelo de regressão logística.

O escore de propensão tem uma propriedade interessante de que, se a atribuição de tratamento é independente dos desfechos potenciais condicionada nas outras variáveis (ou seja, não existem variáveis confundidoras não observadas) então a atribuição de tratamento é independente dos desfechos potenciais condicionado ao escore de propensão $e(X)$:

$$(Y(0), Y(1)) \perp\!\!\!\perp T|X \longrightarrow (Y(0), Y(1)) \perp\!\!\!\perp T|e(X) \tag{22}$$

Prova: a proposição acima é provada mostrando que $P(T = 1|Y(t), e(X)) = P(T = 1|e(X))$ independente do desfecho potencial $Y(t)$.

Como o tratamento é binário:

$$P(T = 1|Y(t), e(X)) = E[T|Y(t), e(X)] \tag{23}$$

Usando a esperança iterada:

$$E[T|Y(t), e(X)] = E[E[T|Y(t), e(X), X]|Y(t), e(X)] \tag{24}$$

Como $e(X)$ é uma função de X, é possível retirar do primeiro condicional:

$$E[E[T|Y(t), e(X), X]|Y(t), e(X)] = E[E[T|Y(t), X]|Y(t), e(X)] \tag{25}$$

Tendo como premissa a ignorabilidade condicionada, é removido $Y(t)$ da esperança interior:

$$E[E[T|Y(t), X]|Y(t), e(X)] = E[E[T|X]|Y(t), e(X)] \tag{26}$$

Para retornar $E[T|X]$ para probabilidade:

$$E[E[T|X]|Y(t), e(X)] = E[P(T = 1|X)|Y(t), e(X)] \tag{27}$$

$$E[P(T = 1|X)|Y(t), e(X)] = E[e(X)|Y(t), e(X)] = e(X) \tag{28}$$

Como $e(X)$ não depende de $Y(t)$, T é independente de $Y(t)$ dado $e(X)$[5].

Essa propriedade é interessante, pois reduz um problema de multidimensionalidade para um problema de uma dimensão só ($e(X)$).

3.2.2 Ponderação das Observações

A segunda etapa consiste em calcular os Pesos de Tratamento. Os pesos de tratamento são o inverso das propensões estimadas. Ou seja, indivíduos tratados receberão um peso de $1/e(X)$ e não tratados um peso de $1/(1 - e(X))$.

3.2.3 Cálculo do Efeito de Tratamento Médio

Por fim a equação para cálculo do efeito de tratamento médio se torna a média ponderada dos desfechos Y:

$$\hat{\tau} = E[Y \frac{T - e(X)}{e(X)(1 - e(X))}] \tag{29}$$

O IPTW pode criar grupos tratados e não tratados equilibrados em relação às covariáveis observáveis, simulando um experimento controlado. A análise subsequente considera os pesos, ajudando a produzir estimativas corrigidas que consideram a seleção não aleatória para o tratamento.

O IPTW requer uma estimativa precisa do escore de propensão para funcionar efetivamente, ou seja, necessita estar bem especificado. Valores extremos dos escores de propensão podem resultar em pesos excessivamente altos ou baixos, podendo introduzir instabilidade nos resultados.

Em resumo, o *Inverse Probability of Treatment Weighting* (IPTW) é uma técnica que ajusta os pesos das observações com base nas probabilidades de tratamento estimadas, visando equilibrar os grupos tratados e não tratados em estudos observacionais. Isso ajuda a mitigar o viés de seleção e a produzir estimativas mais confiáveis dos efeitos de tratamento.

3.3 Metalearners

Metalearner é uma classe de algoritmos para estimar efeito de tratamento. Esse conjunto de métodos se baseia em primeiro criar um modelo, chamado de *base learner*, que estima $E[Y|X = x]$. A segunda etapa consiste em calcular o efeito de tratamento com base no modelo previamente construído. Eles têm a vantagem de usar qualquer modelo de aprendizado supervisionado para ajustar a curva nos dados, podendo utilizar de informações a priori diversas na escolha desses modelos, e no desmembramento do problema em sub tarefas para poder estimar o efeito de tratamento condicionado [16].

Com o intuito de aprimorar a análise dos *metalearners*, foi elaborada uma classificação abrangente que divide esses aprendizes em duas categorias distintas, conforme apresentado por Curth [17].

A primeira categoria, denominada modelos *plugins* de uma etapa, envolve a obtenção das regressões $\hat{\mu_w}$ a partir dos dados observacionais. O cálculo do efeito de tratamento condicional, conhecido como CATE (Conditional Average Treatment Effect), é então realizado como $\hat{\tau}(x) = \hat{\mu_1} - \hat{\mu_0}$, onde $\hat{\mu_0}$ é um estimador do resultado potencial $Y(0)$ e $\hat{\mu_1}$ é um estimador do resultado potencial $Y(1)$. Nessa categoria, estão incluídos os modelos *S-Learner* e *T-Learner*.

Por outro lado, os modelos de duas etapas operam por meio de dois processos sequenciais. Inicialmente, se obtém um subconjunto dos estimadores $\hat{\eta} = \hat{\mu_0}, \hat{\mu_1}, \hat{e}$, onde $\hat{e}$ representa o escore de propensão.

A segunda etapa consiste em criar um pseudo efeito de tratamento $\tilde{Y}\hat{\eta}$, utilizando os estimadores calculados na primeira etapa. Esse resultado é empregado na construção de um modelo que realiza regressão diretamente nas variáveis X, considerando que $E[\tilde{Y}\hat{\eta}|X = x] = \tau(x)$. Pertencem a essa categoria os modelos *X-Learner*, *RA-Learner* e *DR-Learner*.

Nas seções subsequentes, serão delineados os principais algoritmos dessa categorização, abordando suas vantagens e as precauções necessárias ao empregá-los.

3.3.1 S-Learner

O *S-Learner* consiste na criação de um modelo $\mu(x, t)$ para estimar Y:

$$\mu(x, t) = E[Y|X = x, T = t] \tag{30}$$

Um estimador $\hat{\mu}$ pode ser criado a partir dos dados disponíveis. O efeito de tratamento para uma determinada unidade com variáveis x e um efeito de tratamento binário pode ser calculada como:

$$\hat{\tau}(x) = \hat{\mu}(x, 1) - \hat{\mu}(x, 0) \tag{31}$$

Esse tipo de método é interessante por ser simples e altamente explicável dependendo

de qual modelo usar para o treinamento. Esse método pode enviesar o efeito de tratamento para zero, principalmente quando usado métodos de árvore ou métodos que se utilizam fortemente de regularização, como o lasso. Isso ocorre quando as variáveis são muito mais preditivas no desfecho do que o tratamento, fazendo com que métodos de árvore escolham muito menos a variável de tratamento para o corte do que outras variáveis, ou que a regularização acaba subestimando a influência da variável de tratamento, ou até mesmo tirando ela do modelo final, em detrimento das outras variáveis.

3.3.2 T-Learner

A metodologia do *T-Learner* (abreviação de Two-Learner) consiste em primeiro criar um modelo controle:

$$\mu_0(x) = E[Y(0)|X = x] \tag{32}$$

Esse modelo é treinado usando a população X_i, Y_i tal que não receberam tratamento ($T = 0$). O estimador resultado do treinamento será denominado $\hat{\mu}_0(x)$.

Em seguida é criado o modelo tratamento:

$$\mu_1(x) = E[Y(1)|X = x] \tag{33}$$

Esse modelo é treinado usando a população $\{X_i, Y_i\}$ tal que de fato receberam tratamento ($T = 1$). O estimador resultado do treinamento será denominado $\hat{\mu}_1(x)$.

Note que os dois modelos são independentes, podendo utilizar modelos diferentes para realizar as sub tarefas. Por fim o efeito de tratamento de uma determinada unidade é calculado tirando a diferença da saída do modelo tratamento com a saída do modelo controle:

$$\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x) \tag{34}$$

Esse método é interessante por separar o cálculo do efeito de tratamento em dois modelos diferentes, retirando assim a necessidade de se preocupar em como o modelo está levando a variável de tratamento em consideração. Apesar disso, esse método pode se tornar falho quando a população dos tratados é muito menor que a população controle. O modelo de tratamento pode ter poucos dados para que um modelo que represente o fenômeno seja adequadamente treinado, sendo necessário o uso de um modelo mais simples para evitar o *overfit*. Isso pode acabar piorando a estimativa do efeito de tratamento, principalmente quando o foco da modelagem é o efeito de tratamento heterogêneo.

3.3.3 X-Learner

Esse método endereça o problema que pode ocorrer usando dados desbalanceados. Ele primeiro consiste no treinamento do modelo controle e tratamento, na mesma forma que o *T-Learner*:

$$\mu_0(x) = E[Y(0)|X = x] \tag{35}$$

$$\mu_1(x) = E[Y(1)|X = x] \tag{36}$$

A segunda etapa está em calcular o efeito de tratamento individual do grupo tratado $\{X_i^1, Y_i^1\}$, usando o modelo controle para calcular $Y(0)$:

$$\tilde{D}_i^1 = Y_i^1 - \hat{\mu}_0(X_i^1) \tag{37}$$

O mesmo processo é realizado para calcular o efeito de tratamento individual do grupo controle $\{X_i^0, Y_i^0\}$, mas usando o modelo tratamento para calcular $Y(1)$:

$$\tilde{D}_i^0 = \hat{\mu}_1(X_i^0) - Y_i^0 \tag{38}$$

Para o caso em que $\hat{\mu}_0 = \mu_0$ e $\hat{\mu}_1 = \mu_1$:

$$\tau(x) = E[\tilde{D}^0|X = x] = E[\tilde{D}^1|X = x] \tag{39}$$

A terceira etapa está em estimar $\tau(x)$ diretamente de duas formas: usando os dados do grupo controle $\{X_i^0, \tilde{D}_i^0\}$ e da segunda forma usando os dados do grupo tratamento $\{X_i^1, \tilde{D}_i^1\}$:

$$\tau_0(x) = E[\tilde{D}^0|X^0] \tag{40}$$

$$\tau_1(x) = E[\tilde{D}^1|X^1] \tag{41}$$

O racional deste cálculo está em adicionar fenômenos observados no grupo controle, mas que não se observam no grupo dos tratados e vice-versa no efeito de tratamento. Quando há um desbalanço de classe grande, pode-se optar por um modelo mais simples na classe minoritária, mas a classe majoritária também mostram relações complexas que ocorrem em toda a população, mas que na classe minoritária não aparece pela carência de dados e pela escolha de um modelo mais simples para não ocorrer *overfit*.

A última etapa está na determinação de $\tau(x)$ pela ponderação dos modelos que calculam o efeito de tratamento:

$$\tau(x) = g(x) * \tau_0(x) + (1 - g(x)) * \tau_1(x) \tag{42}$$

Aqui $g(x) \in [0, 1]$ representa uma função peso. Normalmente se utiliza o escore de propensão como função, também podendo ser modelado por qualquer modelo supervisionado:

$$g(x) = E[T = 1|X] \tag{43}$$

O uso do escore de propensão calibra o peso em que cada um dos modelos tem no resultado final, em que regiões muito desbalanceadas, como quando existem muito mais exemplos não tratados do que tratados e $g(x)$ é baixo, consegue ponderar para que $\tau_1(x)$, que foi modelado usando os dados dos tratados, seja mais importante do que $\tau_0(x)$.

3.3.4 RA-Learner

O *Regrression Adjusted Learner* foi proposto por Curth [17] tem como primeira etapa a criação dos dois modelos-base $\hat{\mu_0}$ e $\hat{\mu_1}$ utilizando o mesmo método do *X-Learner*. A segunda etapa é a regressão em X onde o pseudo efeito de tratamento $\tilde{Y}\hat{\eta}$ é calculado da seguinte forma:

$$\tilde{Y}_{RA} = T(Y - \hat{\mu_0}(X)) + (1 - T)(\hat{\mu_1}(X) - Y) \tag{44}$$

Nos casos em que o indivíduo foi tratado, o pseudo efeito de tratamento vai ser $(Y - \hat{\mu_0}(X)$, enquanto se o indivíduo não for tratado, $(\hat{\mu_1}(X) - Y$.

O *RE-Learner* pode ser considerado um caso particular no *X-Learner*, onde se usa $g(x) = T$ em vez do comumente usado escore de propensão e não é realizada a regressão em $Y - \hat{\mu_0}(X)$ e $\hat{\mu_1}(X) - Y$, sendo usado os valores direto.

3.3.5 DR-Learner

O *doubly-robust learner*, proposta por Kennedy em [18] e usa a mesma ideia dos *metalearners* de duas etapas, no qual a primeira se cria tanto $\mu_0(x)$ e $\mu_1(x)$ quanto o escore de propensão $e(x)$. O cálculo do pseudo efeito de tratamento é dado da seguinte forma:

$$\tilde{Y}_{DR} = \frac{T - \hat{e}(X)}{\hat{e}(X)(1 - \hat{e}(X))}\{Y - \hat{\mu_T}(X)\} + \hat{\mu_1}(X) - \hat{\mu_0}(X) \tag{45}$$

O nome "doubly-robust"deriva da natureza dupla do método: ele visa ser robusto tanto à especificação errada do modelo de propensão quanto de μ_0 e μ_1. Em outras palavras, mesmo que um dos modelos esteja errado, o DR-Learner compensa a falha no outro modelo.

No caso $\tilde{Y}_{DR}$ é composto por duas componentes principais. A primeira baseada no escore de propensão e a segunda nos modelos de desfecho potenciais. Isso torna o modelo robusto tanto a má especificação do escore de propensão quanto aos modelos de desfecho potenciais.

3.3.6 Cross-fitting

Para os modelos de duas etapas é possível usar a ideia de separar o *dataset* em dois conjuntos[18]. O primeiro conjunto é utilizado para calcular $Y\hat{\mu_0}$, $\hat{\mu_1}$ e $\hat{e}$ e o segundo conjunto usado para construir o modelo que faz a regressão no pseudo efeito de tratamento. Para o cálculo do modelo final é repetido o mesmo processo, mas trocando os conjuntos utilizados em cada etapa. O modelo final é a média dos modelos calculados anteriormente.

A utilidade do *cross-fitting* está relacionada à mitigação do *overfit*. Modelos complexos que podem ser utilizados como base nos *metalearners* podem ser muito flexíveis e ajustar-se demais aos dados de treinamento, capturando até mesmo variações irrelevantes ou ruído. Isso pode levar a estimativas enviesadas quando aplicadas a novos dados. O *cross-fitting* ajuda a evitar esse problema, pois utiliza dados independentes para o treinamento de cada etapa. Isso proporciona uma estimativa mais realista da capacidade do modelo de generalizar para dados não observados.

Esse modelo pode facilmente ser generalizado usando *K-fold* [19], separando em um número arbitrário de conjuntos e realizando treinamentos de forma independente e utilizando a média dos modelos obtidos como resultado final. Um resultado interessante desse método está na possibilidade de conseguir calcular a incerteza do modelo, já que é obtido vários resultados diferentes, que são obtidos a partir de cada processo de treinamento com um conjunto de dados distinto, podendo assim calcular estatísticas como desvio padrão.

3.4 Tabela Comparativa

A tabela 2 mostra um comparativo dos principais modelos implementados na Pycausal-explorer, com suas características, pontos fortes e fracos:

Tabela 2: Tabela comparativa dos principais modelos do Pycausal-explorer

Algoritmo	Características	Pontos Fortes	Pontos Fracos
Regressão Linear	Simplicidade e interpretabilidade; premissa de relação linear entre as variáveis	Útil para cenários simples e bem compreendidos.	Não captura relações não lineares ou efeitos heterogêneos complexos.
IPTW	Atribui pesos a observações com base nas probabilidades de tratamento estimadas.	Pode controlar vieses de seleção.	Sensível a estimativas imprecisas do escore de propensão.
S-Learner	Simplicidade na modelagem, é apenas necessário um modelo base; facilmente interpretável; pode usar uma variedade de modelos-base.	Pode capturar efeitos heterogêneos do tratamento.	Sensível a suposições incorretas sobre a relação entre tratamento e resultado.
T-Learner	Simplicidade e interpretabilidade; pode ser usado com vários modelos-base.	Robusto em situações onde os dois modelos-base são bem especificados.	Sensível à presença de modelos-base mal especificados.
X-Learner	Combina os pontos fortes do T-learner e do S-learner; modelo de duas etapas, prevê diretamente o efeito de tratamento; pode ser usado com vários modelos-base.	Robusto em diferentes cenários causais.	Pode ser computacionalmente intensivo devido à necessidade de ajustar 4 modelos
RA-Learner	Combinação de regressão e ajuste de tratamento (propensity score); pode ser usado com vários modelos-base.	Pode capturar efeitos heterogêneos do tratamento e controlar vieses de seleção.	Exige a estimativa do propensity score, que pode ser sensível a modelos mal especificados.
DR-Learner	Estima o efeito de tratamento levando em consideração o modelo de propensão e os modelos de desfecho potencial; pode ser usado com vários modelos-base.	Funciona bem apenas quando o modelo de propensão ou o modelo de desfechos potenciais estiverem bem especificados	Maior complexidade.

4 Desenvolvimento da Biblioteca Pycausal-explorer

Foi desenvolvida uma biblioteca em Python chamada pycausal-explorer, destinada a facilitar a realização de inferência causal, proporcionando uma ferramenta valiosa para profissionais e pesquisadores engajados em análises de causas e efeitos em estudos observacionais.

A escolha do Python como linguagem base para a biblioteca foi embasada em sua ampla utilização na indústria e na sua grande comunidade de desenvolvedores. Python é conhecido por ser altamente versátil, especialmente em análise de dados e *machine learning*, o que o torna uma plataforma sólida para a construção de modelos de inferência causal.

A biblioteca foi criada para ser compatível com a amplamente utilizada biblioteca scikit-learn (sklearn), que se estabeleceu como a escolha padrão para muitas tarefas de *machine learning*. Essa abordagem visa minimizar as barreiras de entrada para profissionais já familiarizados com o *sklearn*, eliminando a necessidade de aprender um novo framework. Além disso, a integração da biblioteca com o sklearn permitirá que ela seja facilmente incorporada em casos de uso existentes, ampliando sua aplicabilidade e potencial de impacto em diversas áreas.

A biblioteca é de código-fonte aberto e todo o seu código está disponível em um repositório no GitHub[1], promovendo colaboração e transparência no desenvolvimento.

As principais dependências da biblioteca incluem pandas para tratamento de *dataframes*, *numpy* para operações vetoriais, além da integração com o próprio *sklearn*.

A biblioteca também implementa todos os modelos descritos na seção anterior, proporcionando aos usuários uma variedade de opções para realizar suas análises de inferência causal de maneira abrangente e robusta.

4.1 Bibliotecas Relacionadas

Embora o Pycausal-Explorer seja o único que permite a exploração do modelo causal na forma de uma ferramenta Python compatível com scikit-learn, existem outros kits de ferramentas proeminentes voltados para inferência que também trazem muito valor para aplicações de pesquisa e negócios. Uma visão geral desses kits de ferramentas e como eles se comparam ao Pycausal-Explorer podem ser encontrados na Tabela 3.

*Causal explorer*l [20] é um pacote MATLAB com ferramentas de inferência causal destinado a aplicações biomédicas, e é talvez uma das primeiras bibliotecas publicadas destinadas trabalhando com causalidade. Ele fornece ferramentas para modelar relações causais entre variáveis como redes probabilísticas causais (que são redes bayesianas) e algoritmos de descoberta de grafos que ajudam a escolher a melhor suposição de grafo

[1] https://github.com/gotolino/pycausal-explorer

Tabela 3: Comparação do Pycausal-explorer com outras bibliotecas importantes de modelagem causal.

Biblioteca	**Pycausal-Explorer**	**Causal explorer** [20]	**Generalized random forests** [21]	**DoWhy** [22]	**EconML** [23]	**CausalML** [24]
Linguagem	Python	MATLAB	R	Python	Python	Python
Principal propósito	Exploração e avaliação de modelos causais, incluindo modelos de efeito de tratamento heterogêneo	Exploração de modelos causais e seleção de confundidores usando redes bayesianas	Lidando com efeito de tratamento heterogêneo e fornecendo intervalos de confiança para estimativa de efeito de tratamento usando algoritmos baseados em árvore	Ferramentas para identificação minuciosa do efeito causal e verificação de suposições para modelos causais	Uso de ferramentas de aprendizado de máquina para construir modelos causais interpretáveis, especialmente em economia	Estimativa do efeito de tratamento médio condicional (CATE) ou efeito de tratamento individual (ITE) usando modelos de *uplifting* e de inferência causal
Compatível com scikit-learn?	Sim	Não	Não	Não	Não	Não
Algoritmos	Modelos lineares, *metalearners*, uso de modelos base do scikit-learn	Redes bayesianas	Árvores causais, florestas causais	Modelos causais mais populares e todos os modelos do EconML e CausalML	*Double machine learning*, *metalearners* and variáveis instrumentais	Algoritmos de *uplift*, *metalearners*, variáveis instrumentais

causal.

Generalized random forests [21] é um pacote em R de código aberto que implementa modelos causais baseados em *ensembles* de árvores, suportando a chamada estimativa honesta de efeitos causais. Embora seja uma linguagem muito popular nos meios acadêmicos, especialmente no campo da econometria, Python tende a ser mais usado do que R em aplicativos de negócios e é mais fácil de integrar com APIs e software externo, então pode haver demanda por uma implementação em Python de *generalized random forests*.

Além disso, também há exemplos de bibliotecas Python criadas pela indústria para fins modelagem causal. CausalML [24] é uma biblioteca feita pela Uber focada em modelos de *uplift* (ou seja, estimar os efeitos das intervenções em cenários como testes A/B). EconML [23] é uma biblioteca feita pela Microsoft que utiliza ferramentas de aprendizado de máquina para estimar os efeitos do tratamento, particularmente para aplicações em Economia. Outra biblioteca criada pela Microsoft chamada DoWhy [22] implementa uma ferramenta de análise causal de ponta a ponta, abstraindo a maior parte do trabalho de processamento de dados. Esta última biblioteca foca na identificação de efeito causal (ou seja, inspecionar as suposições dos modelos causais) e permite a integração com algoritmos implementados por CausalML e EconML.

Por fim, como visto na Tabela 3, embora a DoWhy seja a biblioteca com maior número de algoritmos implementados, Pycausal-Explorer é o único compatível com Scikit-learn e possui recursos para exploração causal e seleção de confundidores. Nossa biblioteca é uma nova abordagem em tornar a inferência causal e a identificação do efeito do tratamento heterogêneo mais popular, usando Python como linguagem principal e com a novidade de

torná-lo compatível com o scikit-learn. Ao fazer isso, pycausal-explore habilita todos os recursos já implementados no scikit-learn para serem usados, expandindo uma ferramenta tão conhecida na comunidade para uso na exploração causal.

4.2 Classe Base para Modelo de Inferência Causal

A classe base, chamada de *BaseCausalModel* presente no arquivo base.py, mostra uma classe abstrata em que todos os modelos de inferência causal devem herdar. Ela herda do *BaseEstimator*, sendo a classe base para estimadores do scikit-learn, e do ABC que é a classe abstrata do Python. Uma classe abstrata é uma classe que não pode ser instanciada diretamente sendo projetada para ser usada como uma classe base para outras classes. Ela serve como um modelo que define um conjunto de métodos e propriedades que suas subclasses devem implementar. A figura 10 mostra um pedaço do código implementado.

```
class BaseCausalModel(BaseEstimator, ABC):
    """Base class for causal model.

    All models should inherit from this base class.
    All models should at least implement a fit and predict_ite methods.
    """

    @abstractmethod
    def fit(self, X, y, *, treatment):
        """
        Fit model with variables X and target y.

        Parameters
        ----------
        X : {array-like, sparse matrix} of shape (n_samples, n_features)
            Features to control for when estimating causal effect.

        y : array-like of shape (n_samples,)
            Outcome of samples.

        treatment : array-like of shape (n_samples,)
            Binary array. Describes wether or not treatment was applied on a given sample.

        Returns
        -------
        self : object
            Fitted model.
        """
```

Figura 10: Pedaço da implementação do BaseCausalModel

O código de um novo modelo de inferência causal deve herdar do *BaseCausalModel* e

implementar pelo menos os métodos *fit* e *predict_ite*.

O método *fit* é responsável por treinar o modelo de inferência causal. Ele recebe como parâmetro uma matriz com as covariáveis X, um vetor com o desfecho y e um vetor com o tratamento *treatment*, que devem ter todos o mesmo número de linhas. Na definição da função, o asterisco (*) é usado para separar os argumentos posicionais dos argumentos nomeados. Os argumentos posicionais são aqueles passados na ordem em que a função espera, enquanto os argumentos nomeados são especificados com seus nomes. A variável de tratamento está em argumentos nomeados e não posicional para ser compatível com o BaseEstimator do scikit-learn, que espera receber apenas os valores X e y.

O método *predict_ite* é responsável por calcular o efeito de tratamento individual de uma ou um conjunto de observações. Ele recebe como parâmetro uma matriz com as covariáveis X. Mesmo modelos que por padrão não são projetados a calcular o efeito de tratamento individual devem implementar o método, retornando o efeito de tratamento médio para aquele caso.

O método *predict_ate* também é descrito e implementado na classe, recebendo X como parâmetro de entrada e com o resultado sendo a média dos valores de *predict_ite*. Caso seja necessário, também pode-se mudar a implementação de tal classe.

4.3 Datasets

A biblioteca também implementa uma série de *datasets* propícios para a análise de inferência causal. Esses *datasets* diferem de *datasets* de *machine learning* tradicionais, pois o é avaliado é o efeito de tratamento, ou seja, a diferença dos desfechos potenciais.

O problema fundamental da inferência causal está na impossibilidade de se observar os dois desfechos potenciais ao mesmo tempo, o que torna a validação desse tipo de problema mais complexo que aprendizado supervisionado padrão. O resultado $Y(1) - Y(0)$ não é conhecido para nenhuma observação X_i, impossibilitando a criação ou coleta de *datasets* sem ter outras premissas consideradas.

Para avaliar tais modelos, a literatura se baseia na criação de conjuntos de dados sintéticos ou semi-sintéticos, para simular os dois desfechos potenciais, sendo então possível calcular o efeito de tratamento. Alguns conjuntos de dados foram disponibilizados na biblioteca para facilitar a avaliação de modelos, alguns amplamente utilizados na literatura.

4.3.1 Datasets Sintéticos

Foi implementada uma função que retorna uma tabela com n amostras com uma coluna de uma variável aleatória x, uma coluna de tratamento e uma coluna com o desfecho y. A variável x é calculada a partir de uma normal de média 1 e desvio padrão 1 e o tratamento a partir de um binomial. O desfecho y então é calculado através da seguinte fórmula, tendo como efeito de tratamento igual a 1:

$$y = 0,5x + efeito\_tratamento * tratamento \tag{46}$$

Caso o usuário queira que o desfecho seja binário, é possível passando um parâmetro e o desfecho y sofre mais uma transformação y':

$$y' = \begin{cases} 1 & \text{se } y \geq \text{mediana}(Y) \\ 0 & \text{caso contrário} \end{cases} \tag{47}$$

Sendo y a observação e Y o vetor com todas as observações.

4.3.2 Datasets Semi-sintéticos

Alguns *datasets* semi-sintéticos também foram incorporados à biblioteca. Eles são famosos, pois são bastante utilizados na literatura para testar e avaliar desempenho de modelos de inferência causal, sendo importante sua utilização na comparação de métodos já desenvolvidos. Foram agregados os dados do IHDP e *Jobs*.

O *Infant Health and Development Program* (IHDP) é um estudo controlado randomizado projetado para avaliar o efeito de visitas domiciliares de médicos especialistas nas pontuações de testes cognitivos de bebês prematuros. O conjunto de dados foi utilizado primeiramente para avaliar algoritmos de inferência causal usando modelagem bayesiana não paramétrica, descrita em Hill [25]. Nesse contexto, viés de seleção é induzido ao remover subconjuntos não aleatórios dos indivíduos tratados para criação semi-sintética de um conjunto de dados observacionais. Os resultados são gerados usando as covariáveis e tratamentos originais. Ele contém 747 indivíduos e 25 variáveis.

O conjunto de dados *Jobs* consiste na análise do aumento de renda nos indivíduos que passaram por um treinamento financiado pela *National Supported Work* (NSW). O conjunto de dados foi primeiro utilizado em LaLonde[26] e consiste em variáveis demográficas como educação e raça, o tratamento sendo se passou ou não por um treinamento, além da renda em 1975, antes do treinamento, e em 1978, depois do treinamento. Assim como o IHDP, ele foi um estudo controlado randomizado, onde grupos controle e que receberam o treinamento foram escolhidos de forma aleatória. Ao conjunto de dados foram adicionados novos indivíduos que não tiveram treinamento, retirados dos *Population Survey of Income Dynamics* e do *Current Population Survey*, tornando ele um conjunto observacional semi-sintético.

4.3.3 Considerações do Uso

Apesar de datasets semi-sintéticos, como os implementados na biblioteca, serem amplamente utilizados para inferência causal, estudos mais recentes mostram que eles podem não refletir dados reais. Tais conjuntos de dados em que se sabe parte do seu processo

de geração podem gerar uma vantagem esperada para alguns estimadores em detrimento de outros, pelo fato deles conseguirem se adaptar melhor aos dados fornecidos na hora do treinamento.

Os estudos se baseiam na forma em que os conjuntos de dados sintéticos e semi-sintéticos são construídos e na relação entre $Y(0)$ e $Y(1)$. De uma forma genérica o desfecho potencial $Y_w(x)$ pode ser escrito como:

$$Y_w(x) = \begin{cases} f_0(x) & \text{se } w = 0 \\ f_1(x) & \text{se } w = 1 \end{cases} \tag{48}$$

Sendo $f_0(x)$ e $f_1(x)$ funções arbitrárias. Uma abordagem comum na simulação dessas curvas é na decomposição de $f_1(x)$ como parte de $f_0(x)$ mais uma composição aditiva independente de $f_0(x)$ que representa o efeito de tratamento:

$$Y_w(x) = f_0(x) + wf_\tau(x) \tag{49}$$

Na vida real, as curvas de desfecho potenciais podem assumir quaisquer características $f_1(x) = g(f_0(x))$. Alguns conjuntos de dados, como no caso do IHDP, fogem dessa decomposição. No caso do IHDP $g(x)$ é uma transformação logarítmica.

Dependendo de qual curva de funções foi utilizada para criar os *datasets* podem afetar diretamente quais modelos podem performar melhor ou não. Modelos lineares podem se beneficiar quando a premissa para criar as curvas de desfechos potenciais for aditiva, enquanto modelos que se beneficiam na modelagem de curvas exponenciais podem ter uma melhor desempenho no IHDP. Isso não necessariamente quer dizer que um modelo é melhor que outro, apenas que as premissas utilizadas para criação das DGPs beneficiaram tais modelos, mas que podem não ser comparáveis com uma situação do mundo real [27].

Compreender os vieses inerentes aos conjuntos de dados e estimadores utilizados é essencial para determinar se as medições obtidas verdadeiramente refletem a realidade. Uma abordagem que merece consideração é a utilização de conjuntos de dados mais realísticos, uma vez que isso pode proporcionar uma avaliação mais precisa do desempenho dos modelos. Uma alternativa promissora para alcançar esse objetivo é a biblioteca RealCause [28], que visa criar conjuntos de dados mais fiéis à realidade por meio de modelos generativos. Ao adotar essa abordagem, é possível mitigar alguns dos desafios relacionados aos vieses e obter estimativas mais confiáveis dos efeitos causais.

4.4 Métricas

A avaliação adequada dos métodos de inferência causal requer a utilização de métricas que permitam medir a eficácia desses métodos em estimar os efeitos de tratamento de forma precisa e confiável. A seguir são descritas duas métricas implementadas na

Pycausal-Explorer. Outras métricas, disponíveis no sklearn, também podem ser utilizadas como, por exemplo, erro quadrático médio, raiz do erro quadrático médio, erro médio absoluto.

4.4.1 Precision in Estimation of Heterogeneous Effect

A métrica "PEHE"(Precision in Estimation of Heterogeneous Effect) é uma métrica utilizada para avaliar a qualidade das estimativas de efeito de tratamento em problemas de inferência causal. Ela se concentra na diferença entre os valores potenciais (ou valores de resultado) observados e os valores potenciais estimados, que representam o que teria acontecido em termos de resultados se um indivíduo tivesse sido tratado ou não tratado. Ela foi primeiramente definida em [25] e desde então é uma métrica utilizada para calcular desempenho de modelos de inferência causal. Ela é calculada da seguinte forma:

$$PEHE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i(1) - \hat{Y}_i(0) - (Y_i(1) - Y_i(0)))^2} \tag{50}$$

Onde:

- N: é o número de observações
- $Y(0)$: é o desfecho potencial quando não há tratamento
- $Y(1)$: é o desfecho potencial quando há tratamento
- $\hat{Y}_i(0)$: é o desfecho potencial estimado quando não há tratamento
- $\hat{Y}_i(1)$: é o desfecho potencial estimado quando há tratamento

Como $Y_i(1) - Y_i(0) = \tau(X_i)$ e $(\hat{Y}_i(1) - \hat{Y}_i(0) = \hat{\tau}(X_i)$ a equação pode ser reescrita como:

$$PEHE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\tau}(X_i) - \tau(X_i))^2} \tag{51}$$

Onde $\hat{\tau}(X_i)$ é o efeito de tratamento estimado e $\tau(X_i)$ o efeito de tratamento considerando os desfechos potenciais da observação X_i.

4.4.2 Mean Average Percentage Error

A métrica é usada para validar erro percentual do desfecho das observações. Não é uma métrica utilizada normalmente no contexto de inferência causal, mas é uma ferramenta poderosa para mensurar desempenho do efeito de tratamento quando ele não é nulo, mas tem grande variação e queremos dar pesos semelhantes a erros pequenos em efeitos de tratamento pequeno com erros grandes com efeitos de tratamento maiores. Ela é definida como:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\tau(X_i) - \hat{\tau}(X_i)}{\tau(X_i)} \tag{52}$$

Como a métrica tem $\tau(X_i)$ como denominador, ela não é indicada quando existe a hipótese do efeito de tratamento ser nulo em alguma observação.

4.5 Padronização de Código

A manutenção de um código-fonte bem formatado e intuitivo é fundamental para a colaboração eficaz entre desenvolvedores e para a manutenção de projetos ao longo do tempo. Na biblioteca é usado o Black como ferramenta de formatação de código, que se destaca pela sua abordagem rigorosa e automatizada de padronização.

O Black[2] se diferencia por sua abordagem "com opinião" para a formatação de código. Ao invés de depender de decisões subjetivas sobre estilo, o Black adota regras de formatação estritas e executa elas sem margem para personalizações. Isso possibilita uma melhor colaboração entre os desenvolvedores, já que o estilo do código vai estar em um formato conhecido. Também agiliza a colaboração e desenvolvimento, deixando de lado discussões sobre formatação.

Também foi utilizado o PEP8[3] como guia de estilo. PEP significa "Python Enhancement Proposal"(Proposta de Aprimoramento do Python) e o número 8 se refere a esse documento específico na série de PEPs. O PEP 8 estabelece diretrizes e recomendações para a formatação do código Python, incluindo a escolha de nomes de variáveis, organização de imports, espaçamento, indentação e outros aspectos relacionados à legibilidade e consistência do código. Diferentemente do Black, PEP8 não atualiza automaticamente seu código, apenas indica quais diretrizes ele está violando. Além de diretrizes de formatação, também existem outras diretrizes como complexidade do código, declaração de variáveis e importação de bibliotecas.

Por fim foi utilizado o isort[4] como organizador das importações da biblioteca. Ele é responsável por organizar as importações em ordem alfabética, separar em grupos (bibliotecas internas do Python, bibliotecas externas e módulos do próprio código fonte) e tratamento de importações muito longas.

Para garantir que todos os desenvolvedores sempre deixem o código formatado, foi criada uma automação utilizando o pre-commit[5]. Pre-commit é uma biblioteca que possibilita a automação de funções antes de cada *commit* no código. Então o código é formatado usando Black, as importações organizadas usando o isort e é checado se está conforme o guia PEP8. Se falhar o *commit* não é realizado e indicado em quais pontos falhou. Isso

[2]https://github.com/psf/black
[3]https://peps.python.org/pep-0008/
[4]https://pycqa.github.io/isort/
[5]https://pre-commit.com/

permite sempre deixar o código no estilo adequado antes mesmo de entrar no *github*.

4.6 Testes

Para testes automatizados do código foi utilizado o pytest[6]. O pytest é um framework de teste em Python que facilita a escrita e execução de testes unitários e de integração. Ele é amplamente utilizado na comunidade Python devido à sua simplicidade, flexibilidade e recursos avançados.

Existe uma série de vantagens de se utilizar o pytest como *framework* para testes, entre eles:

1. Simplicidade no uso: Testes são implementados como funções, em arquivos começados com *test_*. Essas funções chamam as classes e verificam seu funcionamento.

2. Relatórios: Seus relatórios ao rodar os testes indicam quantas linhas foram testadas, quais testes retornaram erro e como está a cobertura geral de sua biblioteca.

3. Extensível: Pode ser facilmente extensível, como no uso de *plugins* para relatórios mais elaborados.

Todos os módulos devem ter testes adequados, a fim de garantir o funcionamento correto e especificado para cada modelo. São criados testes unitários que verificam inicialização das classes, manuseio de erros e convergência no treinamento usando *datasets* sintéticos. A figura 11 exemplifica a implementação de alguns testes relacionados a inicialização do DRLearner.

Para garantir que a implementação dos modelos foi feita corretamente também são criados testes de desempenho, onde os modelos fazem regressão em um conjunto de dados sintético e é comparado o efeito de tratamento esperado com o obtido pelo modelo. Caso o erro seja maior que 1% pode-se concluir que existe algum erro de implementação.

Também é utilizado o Codecov como serviço de análise de cobertura de código que oferece uma visão detalhada da cobertura do código-fonte de um projeto. Ele é integrado com o pytest para coletar informações de cobertura durante a execução dos testes e, em seguida, enviar esses dados para a plataforma web Codecov[7], onde é possível visualizar relatórios detalhados sobre a cobertura do código. Existem opções de visualizações por arquivo ou agregados. A figura 12 mostra parte da plataforma web.

4.7 Ferramentas de CI/CD

Ferramentas de *continuous integration* (CI) e *continuous delivery* (CD) são utilizadas para automação do processo de criação de verificação de testes da biblioteca, que são

[6]https://docs.pytest.org/en/7.4.x/
[7]https://app.codecov.io/gh/gotolino/pycausal-explorer

```python
def test_drlearner_init_learner():
    learner = LinearRegression()
    drlearner = DRLearner(LinearRegression())
    assert type(drlearner.u0[0]) is type(learner)
    assert type(drlearner.u1[0]) is type(learner)
    assert type(drlearner.tau[0]) is type(learner)

    assert type(drlearner.u0[1]) is type(learner)
    assert type(drlearner.u1[1]) is type(learner)
    assert type(drlearner.tau[1]) is type(learner)


def test_drlearner_init_custom_learner():
    drlearner = DRLearner(
        learner=None,
        u0=LinearRegression(),
        u1=LinearRegression(),
        tau=RandomForestRegressor(n_estimators=100),
    )
    assert isinstance(drlearner.u0[0], LinearRegression)
    assert isinstance(drlearner.u1[0], LinearRegression)
    assert isinstance(drlearner.tau[0], RandomForestRegressor)

    assert isinstance(drlearner.u0[1], LinearRegression)
    assert isinstance(drlearner.u1[1], LinearRegression)
    assert isinstance(drlearner.tau[1], RandomForestRegressor)
```

Figura 11: Exemplo de implementação de testes em DRLearner

norma para acelerar o processo de desenvolvimento, automatizando processos que podem ser maçantes e demorados caso feitos manualmente.

Foi utilizado o GitHub Actions como principal ferramenta de CI/CD da biblioteca, que oferece diversas vantagens quando usado na implementação de uma biblioteca de código aberto:

1. Integração nativa com o GitHub: Como já se usa o GitHub como ferramenta de versionamento de código, o uso do Github Actions é um caminho lógico, com sua facilidade de implementação e recursos já disponibilizados.

2. Rastreabilidade: Através da própria interface do GitHub é possível ver quando houve falha no *pipeline*, oferecendo ferramentas de investigação e reexecução.

3. Transparência no uso de máquina para rodar os pipelines: Não necessita instanciar máquinas específicas para rodar os pipelines, sendo transparente para o usuário esse

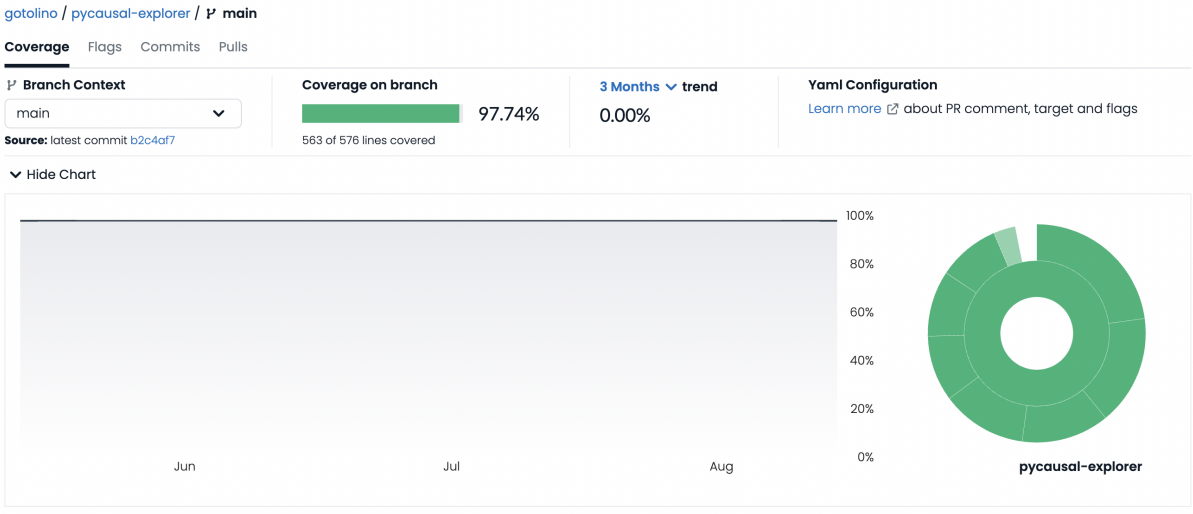

Figura 12: Plataforma web de Codecov

processo.

O Github Actions é utilizado para automatizar alguns processos cruciais: a automação da conformidade de códigos, testes e a implantação de novas versões da biblioteca.

4.7.1 Conformidade de Códigos e Verificação de Testes

O arquivo *.github/workflows/tests.yml* é responsável por verificar a conformidade de código e rodar os testes, gerando os relatórios de cobertura. O *pipeline* roda sempre que é feito um *pull request* (PR), que é a solicitação de mudança de código principal (aqui sendo a *branch* main) ou quando modificam o código da main diretamente.

Sua função está separada em duas partes. A primeira está em verificar se o código está no estilo da biblioteca Black e Isort e se está conforme o guia PEP8. Caso alguma dessas checagens falhe, o processo de modificar o código principal é interrompido até que sejam feitas as devidas alterações no código.

A segunda está em rodar os testes implementados em pytest e na criação do relatório de cobertura para ser enviado ao Codecov para atualização da interface web. Caso algum teste falhe, o processo de modificar o código principal é interrompido até que sejam corrigidos. Para códigos novos também é verificado se testes suficientes foram criados para garantir seu correto funcionamento. Caso a cobertura fique abaixo do mínimo aceitável, o processo também é interrompido.

4.7.2 Publicação de Novas Versões

O segundo *pipeline* está na publicação de novas versões da biblioteca. A ação de publicação da biblioteca é configurada para ser manual, ou seja, não existe nenhum *trigger* em que o pipeline rode automaticamente, sendo necessária a inicialização através de um

mantenedor da biblioteca. Isso permite um controle mais explícito do que é lançado, permitindo revisões e preparações necessárias antes de tornar a nova versão pública.

O *pipeline* consiste no empacotamento do código e geração de um arquivo de distribuição, garantindo que não existam falhas inesperadas, e na sua publicação no PyPI (Python Package Index). O PyPI é um repositório centralizado de pacotes de software para Python. Ele atua como um *hub* onde os desenvolvedores podem compartilhar, distribuir e instalar bibliotecas e pacotes Python. Com ele é possível que outros desenvolvedores obtenham a biblioteca a partir de um gerenciador de bibliotecas, como o *pip*.

4.8 Outros Recursos

Também foram implementados alguns outros recursos menores para melhorar a experiência do desenvolvedor com o uso da biblioteca. Foi criada uma nova classe Pipeline a partir da classe do sklearn. A principal diferença está no fato de poder utilizar o método *predict_ite*, um dos principais métodos quando falando em inferência causal. Assim é possível criar *pipelines* de dados.

Também foi criada documentação automática com acesso web[8] para as classes implementadas, de forma a facilitar o uso de tais módulos. É possível verificar o funcionamento de todas as classes.

[8]https://pycausal-explorer.readthedocs.io/en/latest/

5 Estudo de Caso - Relação entre Renda e Nota do ENEM

Como estudo de caso do uso de biblioteca, foi realizada uma análise para entender como a renda per capita de uma família afeta a nota do ENEM de egressos do ensino médio, mudando o acesso ao ensino superior nas universidades públicas. Para isso, primeiro é necessário entender o que significa igualdade de oportunidade e em que contexto já foi utilizado.

5.1 Educação e Igualdade de Oportunidade

Uma das concepções de justiça mais discutidas é a de igualdade de oportunidade. O êxito de um indivíduo pode ser considerado a consequência de fatores circunstanciais, esforço e políticas, em que fatores circunstanciais são aspectos do ambiente do indivíduo que ele não consegue controlar e que a sociedade não gostaria de responsabilizá-lo (origem dos pais, raça, local de nascimento), esforço compreende em ações que o indivíduo toma que seriam de sua responsabilidade e políticas é o instrumento utilizado para influenciar o êxito do indivíduo. Políticas de igualdade de oportunidade mira em criar um balanço para que fatores circunstanciais sejam balanceados para deixar que o êxito de um indivíduo seja, da melhor forma possível, apenas influenciado pelo esforço [29].

Uma adaptação desses conceitos ao estudo do desempenho educacional está na verificação de anos de estudo do indivíduo. Esse tipo de medida se torna problemática, pois não há garantia que o mesmo tempo de estudo resulta na mesma quantidade de aprendizado. Pessoas que estudaram em escolas diferentes podem ter diferentes níveis de educação, mesmo estudando a mesma quantidade de tempo. Uma medida mais realística do desempenho educacional é a nota de testes padronizados [30].

Devido à natureza dos desfechos ser considerada a soma de aspectos não circunstanciais (ou de não responsabilidade), esforço (ou de responsabilidade) e de políticas, quando não há uma política presente, o resultado de testes padronizados podem ser altamente influenciáveis pelos aspectos de não responsabilidade. Por esse motivo, a nota por si só pode não ser o melhor critério para a entrada de um estudante na universidade, sendo mais uma nota de "riqueza"do que esforço [31].

Existe uma relação substancial entre nota nesse tipo de teste e renda familiar. O estudo conduzido por Dixon-Roman utiliza modelo de equações estruturais para mostrar como a raça e a renda familiar se relacionam com as notas do SAT (prova realizada nos Estados Unidos com o intuito de selecionar os candidatos para ingressar na universidade), mostrando que existe uma relação monotônica crescente entre as notas e a renda familiar, e uma menor nota dos estudantes negros quando comparados na mesma faixa de renda com estudantes brancos [31].

O uso de dados do ENEM para estudo de desigualdade já foi abordado de uma forma mais geral. Usando os dados de 2010, junto com os dados do SAEB, foi quantificado como diferentes *backgrounds* afetavam na nota final do ENEM [32].

Nesse trabalho será focado especificamente como a renda familiar, que é um dos aspectos de não responsabilidade, influencia na nota final do ENEM, tendo um impacto tanto no desempenho educacional adquirido durante os anos de estudos e no acesso ao ensino superior.

5.2 Fontes de dados

5.2.1 IDEB

O Índice de Desenvolvimento da Educação Básica (IDEB) é um índice calculado a partir dos dados sobre aprovação escolar, obtidos no Censo Escolar, e das médias do desempenho no Sistema de Avaliação da Educação Básica (Saeb). Na fonte de dados se encontram as notas no IDEB, indicador de rendimento, nota padronizada, e notas do Saeb, separadas por ensino fundamental (anos iniciais), ensino fundamental (anos finais) e ensino médio[33].

5.2.2 Censo Escolar

O Censo Escolar coleta informações sobre a educação básica no país. Ele abrange o ensino regular, educação especial, educação de jovens e adultos (EJA) e educação profissional (cursos técnicos e cursos de formação inicial continuada ou qualificação profissional).

A coleta de dados é feita de forma declaratória e dividida em duas etapas. A primeira etapa ocorre a coleta de informações sobre os estabelecimentos de ensino, gestores, turmas, alunos e profissionais escolares em sala de aula. A segunda etapa ocorre ao final do ano letivo, coletando informações do aluno, considerando os dados sobre movimentação e rendimento escolar. São disponibilizados dados deste 1995[34].

5.2.3 ENEM

O Exame Nacional do Ensino Médio (ENEM) avalia o desempenho escolar ao final da educação básica. Hoje é o principal meio de acesso ao ensino superior, por meio do Sistema de Seleção Unificado (Sisu), Programa Universidade para Todos (Prouni) e por convênios com instituições portuguesas. Também permite acesso a programas de financiamento estudantil (Fies).

Os microdados do enem disponibilizam dados das inscrições: dados do questionário socioeconômico, dados da cidade onde foi realizado o exame, dados da escola onde foi concluída a educação básica e dados das provas realizadas e notas obtidas[35].

5.3 Extração e Processamento dos Dados

Os dados coletados em sua forma *raw* está na extensão .csv. Um pré-processamento foi feito para transformar os dados em um formato colunar .parquet. Esse formato facilita na leitura e busca de dados que estão em formato de tabela, sendo mais performático que o .csv. Os dados, tanto em csv quanto em parquet, foram depositados no serviço S3 da AWS.

O pré-processamento dos dados foram realizados em clusters *spark* com o seu serviço sendo disponibilizado pela Amazon Glue, que é o serviço de ETL (extração, transformação, carregamento) disponibilizado. Ele envolve, além da transformação das extensões, a identificação das colunas que são numéricas, categóricas ou texto livre. Em colunas que são respostas de questionário, foi feita a transformação da resposta dada pela pessoa que o respondeu (A, B, C, etc.) foi substituída pelo texto que a alternativa representa.

5.4 Bases Utilizadas para o Exemplo de Uso

O ano base utilizado foi o ENEM 2019, contendo:

1. dados do participante
2. dados da escola
3. pedidos de atendimento personalizado
4. pedidos de atendimento específico
5. pedidos de recursos especializados e específicos para a realização das provas
6. dados do local de aplicação da prova
7. dados das provas objetivas
8. dados da redação
9. questionário socioeconômico

Os dados principais para essa análise vêm dos itens 1, 2, 7, 8 e 9.

Também foram relacionadas bases exógenas ao banco de dados principal, a fim de enriquecer o modelo com outras variáveis pertinentes. Foram coletados dados relacionados ao IDH municipal [36] e dados do IDEB [33].

5.5 Análise Exploratória dos Dados

A tabela dos microdados do ENEM 2019 consiste em 5095270 linhas com 136 colunas. Cada linha representa uma inscrição única. Primeiro é necessário definir qual o público que será utilizado para o estudo de caso. Então, nas próximas seções serão feitas análises descritivas das colunas mais importantes para o estudo de caso.

5.5.1 Definição do Público

Para o estudo de caso o objetivo é entender qual o impacto da renda familiar na nota final do ENEM nos egressos do ensino médio. Primeiro precisamos filtrar candidatos que marcaram conclusão do ensino médio em 2019. A figura 13 mostra o número de inscrições por situação de conclusão do ensino médio, baseada na coluna TP_ST_CONCLUSAO. Cerca de 28,8% dos candidatos marcaram conclusão co curso em 2019. Filtrando apenas esses candidatos temos um número atualizado de 1465895 inscrições no público.

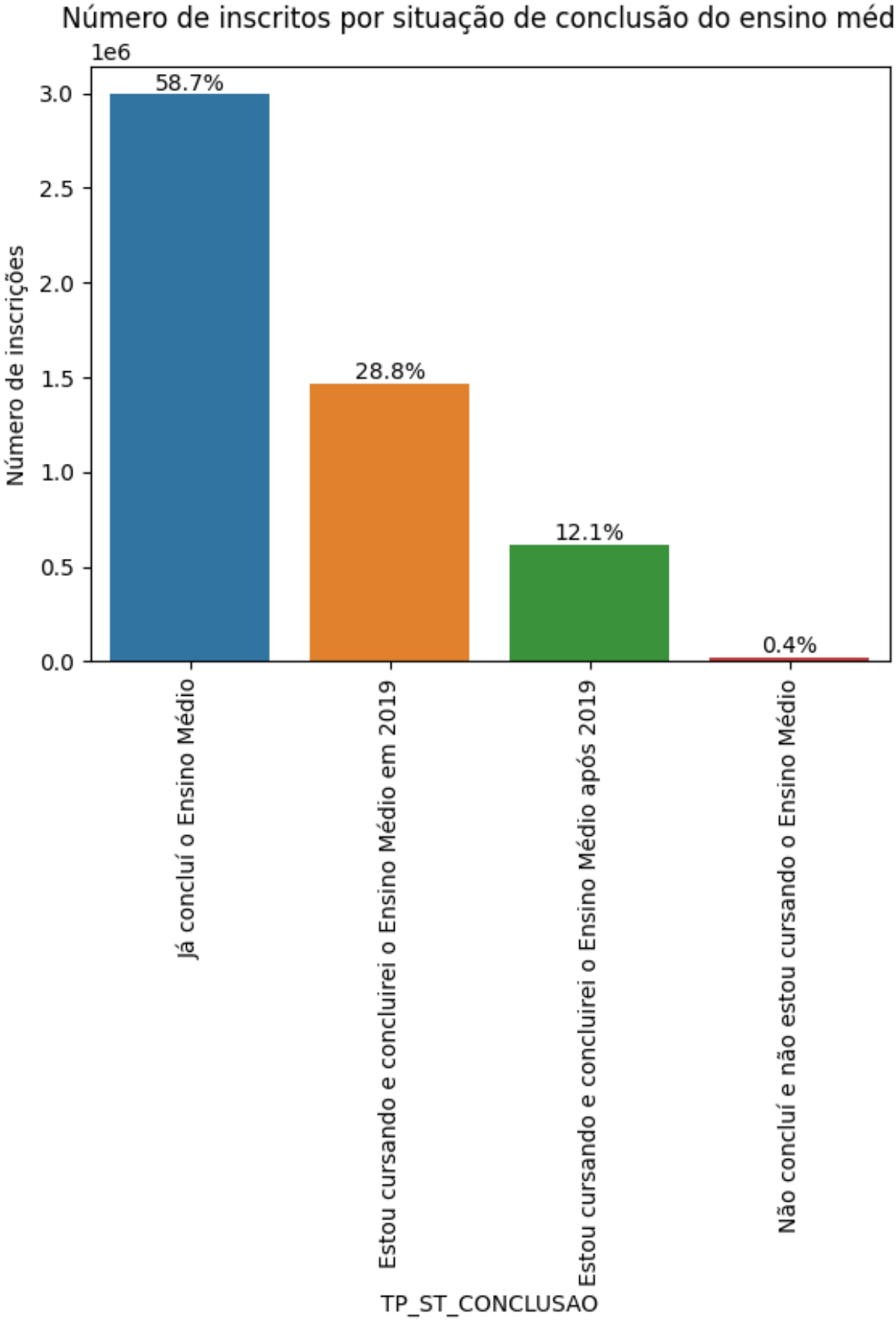

Figura 13: Gráfico mostrando o número de inscrições por situação de conclusão do ensino médio

O próximo passo é identificar as pessoas que de fato foram realizar a prova. Na tabela existem as colunas que representam a presença em cada uma das provas (TP_PRESENCA_CN, TP_PRESENCA_CH, TP_PRESENCA_LC e TP_PRESENCA_MT para a prova de ciência das naturezas, ciências humanas, linguagens e códigos e matemática, respectivamente). Também tem a coluna TP_STATUS_REDACAO que, além de identificar se a redação ficou em branco. Como a prova acontece em dois dias (um para ciências humanas e linguagens e códigos, e outro para ciências da natureza, matemática e redação), apenas a presença de uma prova por dia assegura a presença nos dois dias. 83% dos inscritos egressos do ensino médio compareceram aos dois dias de prova, totalizando um público de 1217156 inscritos, 23,9% da tabela inicial. A análise exploratória dos dados a seguir, modelagem e discussão dos resultados serão baseadas nesse público.

5.5.2 Dados do Participante

Os dados do participante configuram todos os dados relacionados ao candidato. São dados demográficos do local de residência, local de nascimento, idade, sexo, estado civil e dados relacionados ao tipo de ensino médio feito.

A figura 14 mostra a distribuição das inscrições nos estados de residência, representada pela coluna SG_UF_RESIDENCIA. Nota-se que São Paulo e Minas Gerais são os estados com a maior porcentagem de inscritos, enquanto Roraima, Acre e Amapá ocupam as últimas posições. Isso é esperado, já que segundo o IBGE[37], São Paulo e Minas Gerais são os estados mais populosos do Brasil, enquanto Roraima, Amapá e Acre são os menos populosos. A figura 15 mostra o número de inscrições por habitante do estado. Nota-se que o Ceará tem o maior número de inscrições no ENEM per capita, enquanto Roraima fica na última posição.

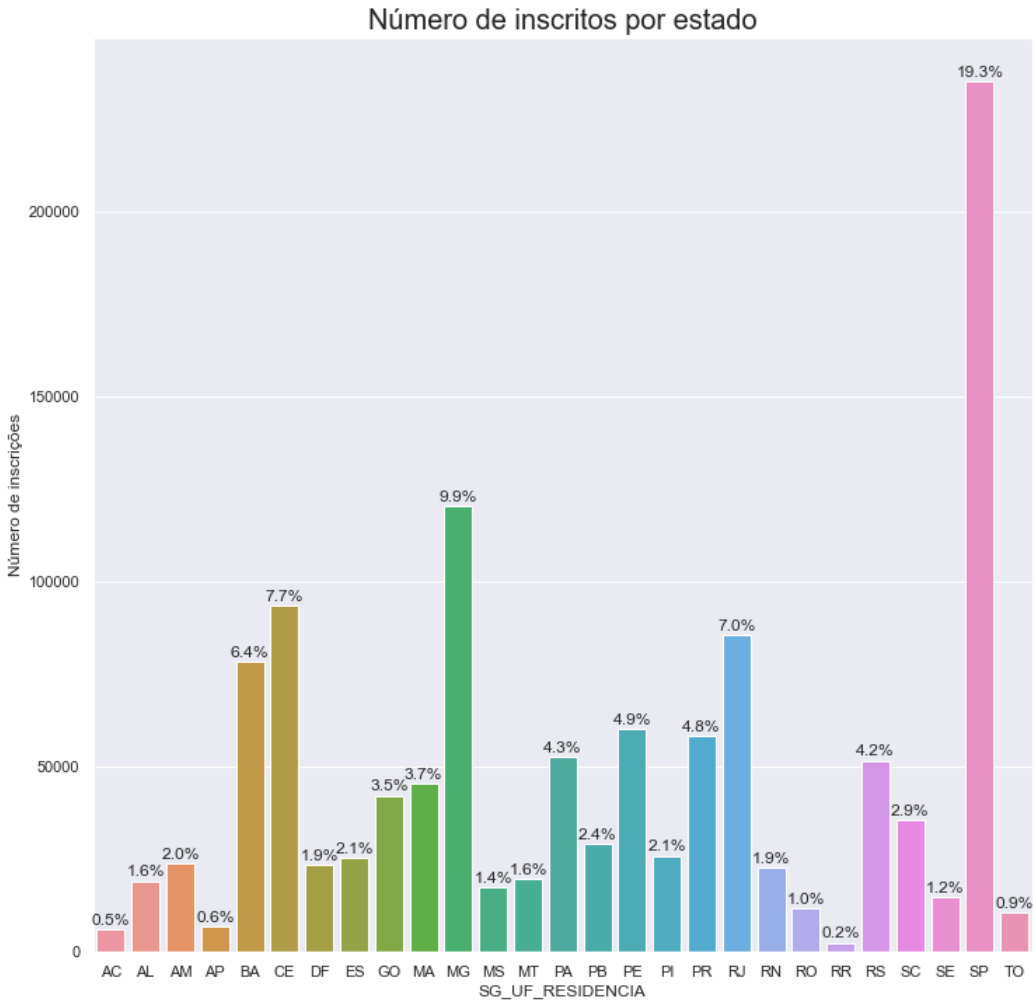

Figura 14: Gráfico mostrando o número de inscrições por unidade federativa

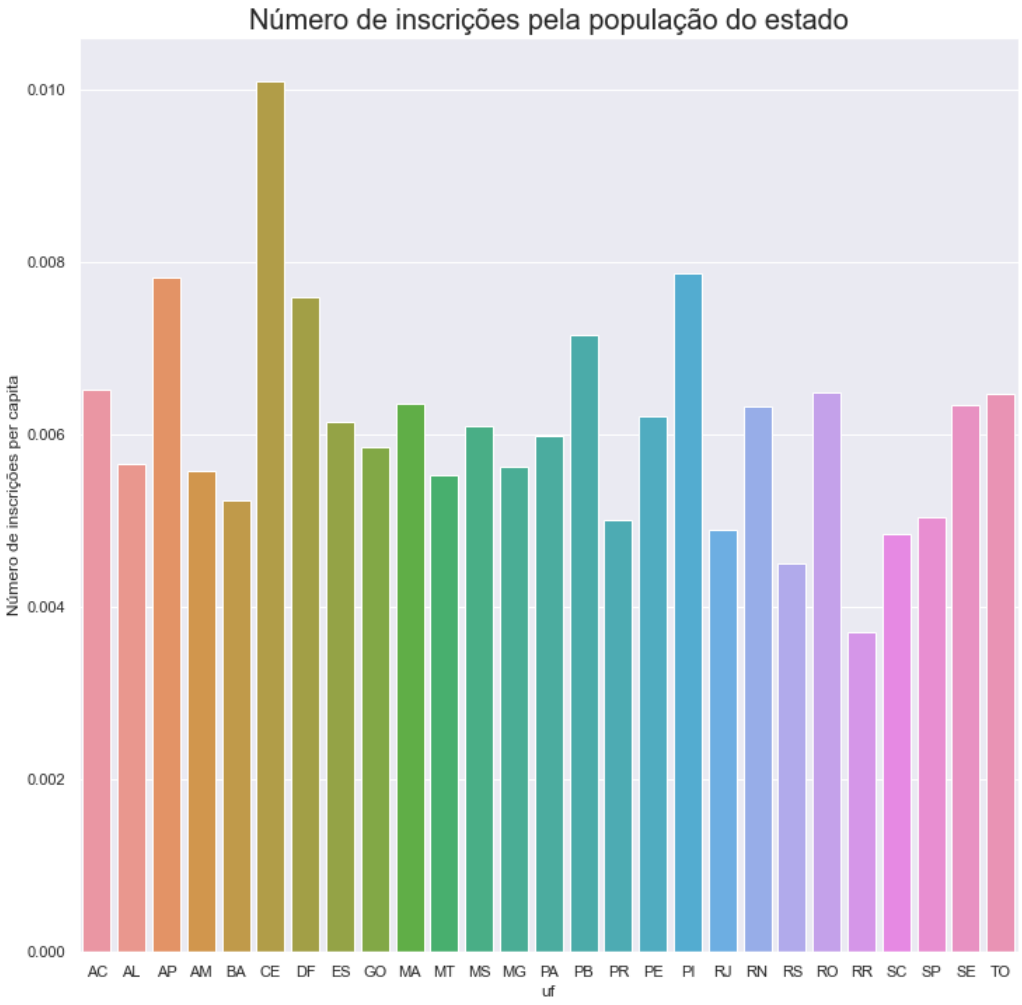

Figura 15: Gráfico mostrando o número de inscrições per capita por unidade federativa

As figuras 16, 17, 18 e 19 mostram a distribuição das inscrições por sexo, estado civil, cor/raça e nacionalidade, representadas respectivamente pelas colunas TP_SEXO, TP_ESTADO_CIVIL, TP_COR_RACA e TP_NACIONALIDADE.

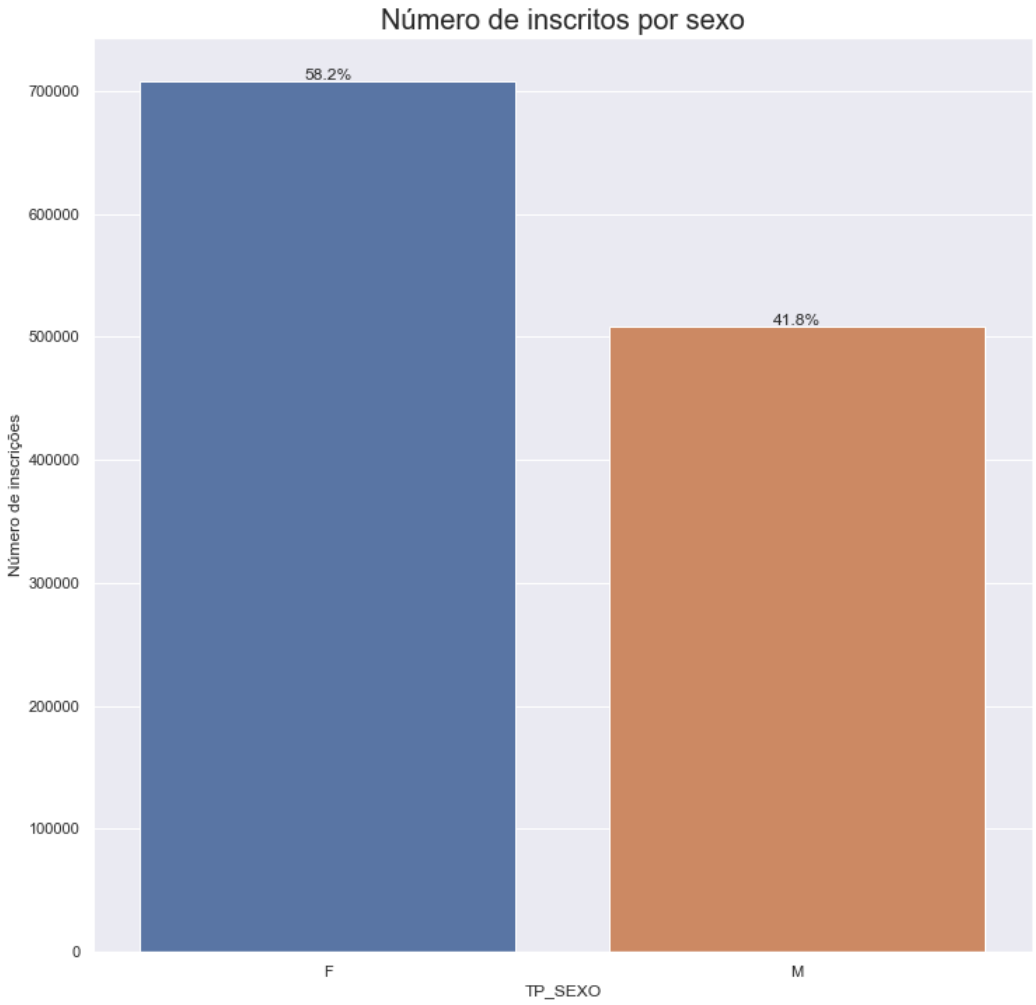

Figura 16: Gráfico mostrando o número de inscrições por sexo

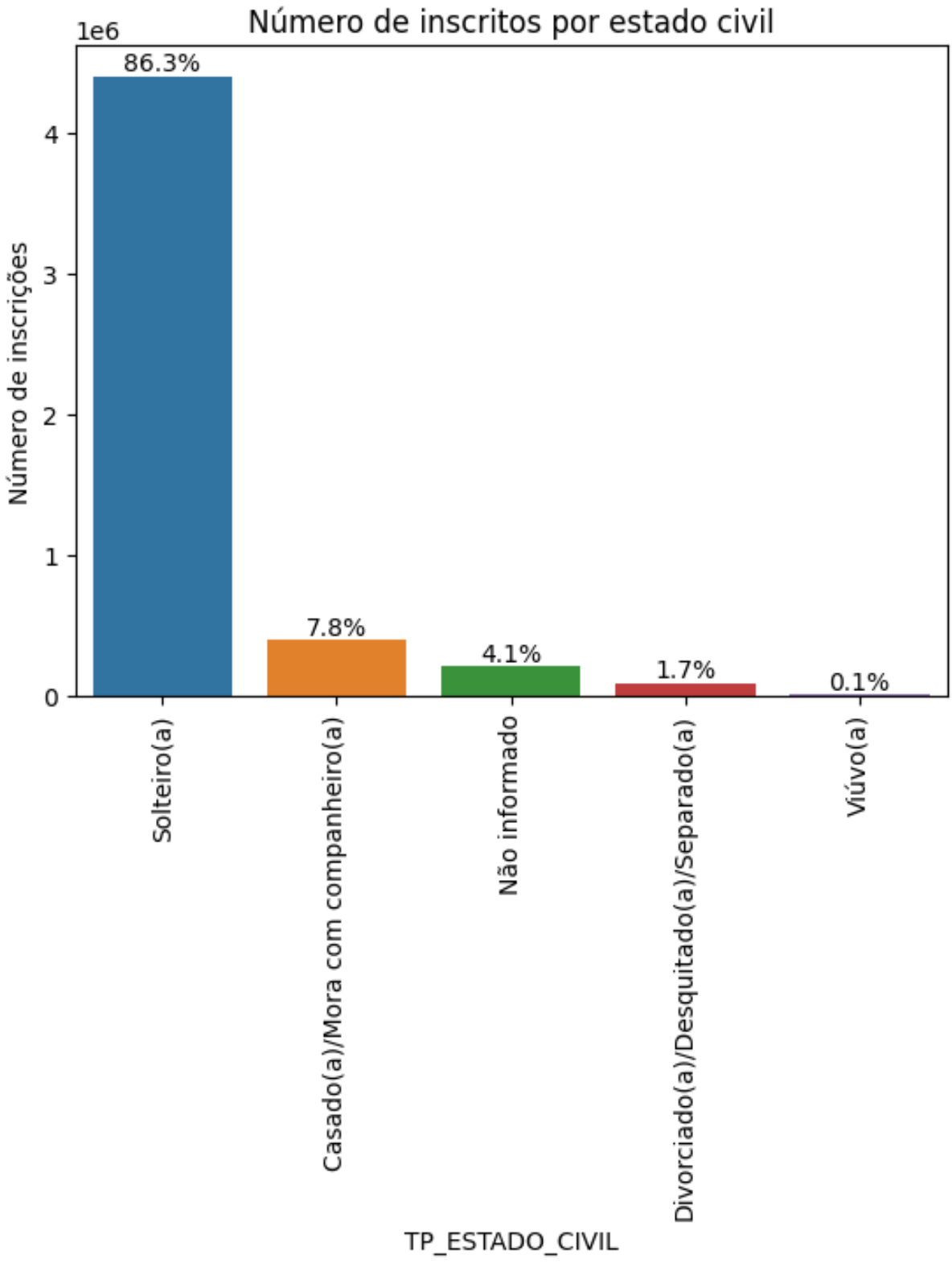

Figura 17: Gráfico mostrando o número de inscrições por estado civil

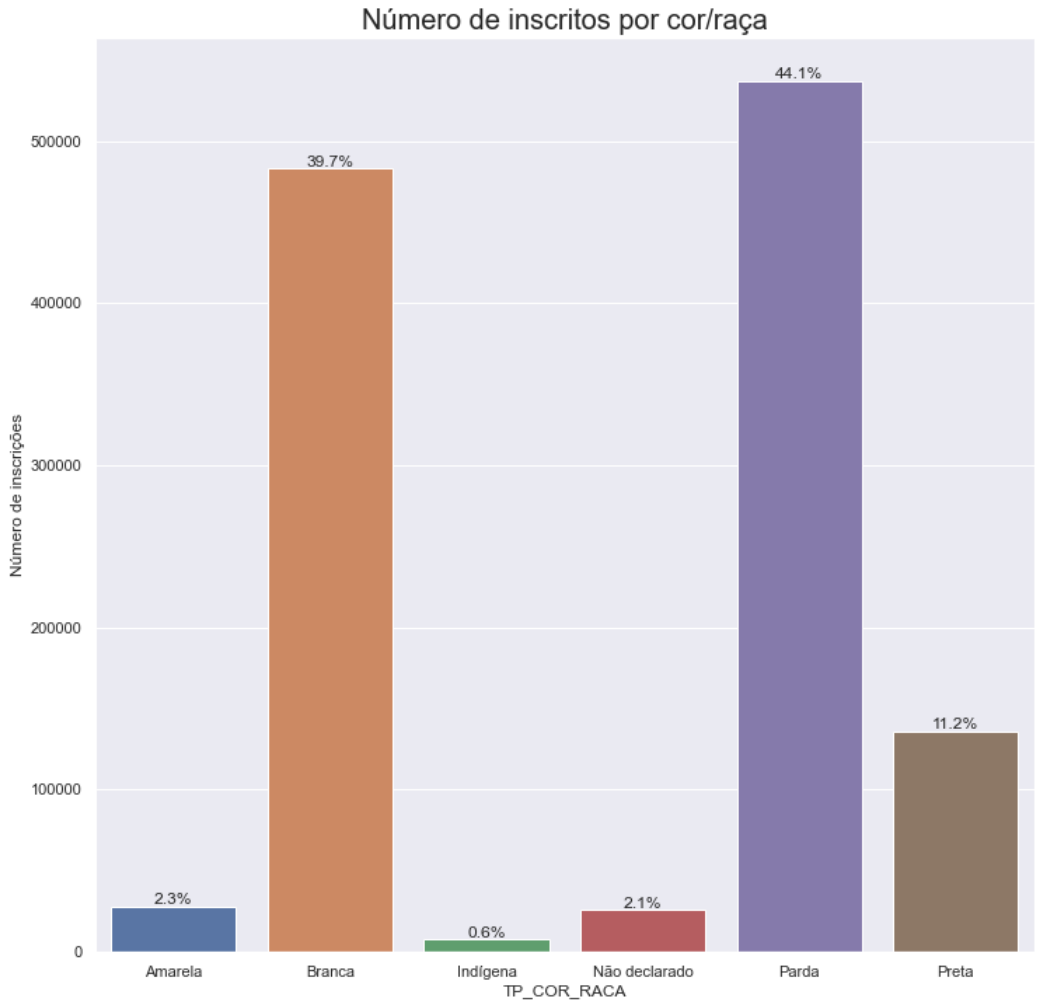

Figura 18: Gráfico mostrando o número de inscrições por cor/raça

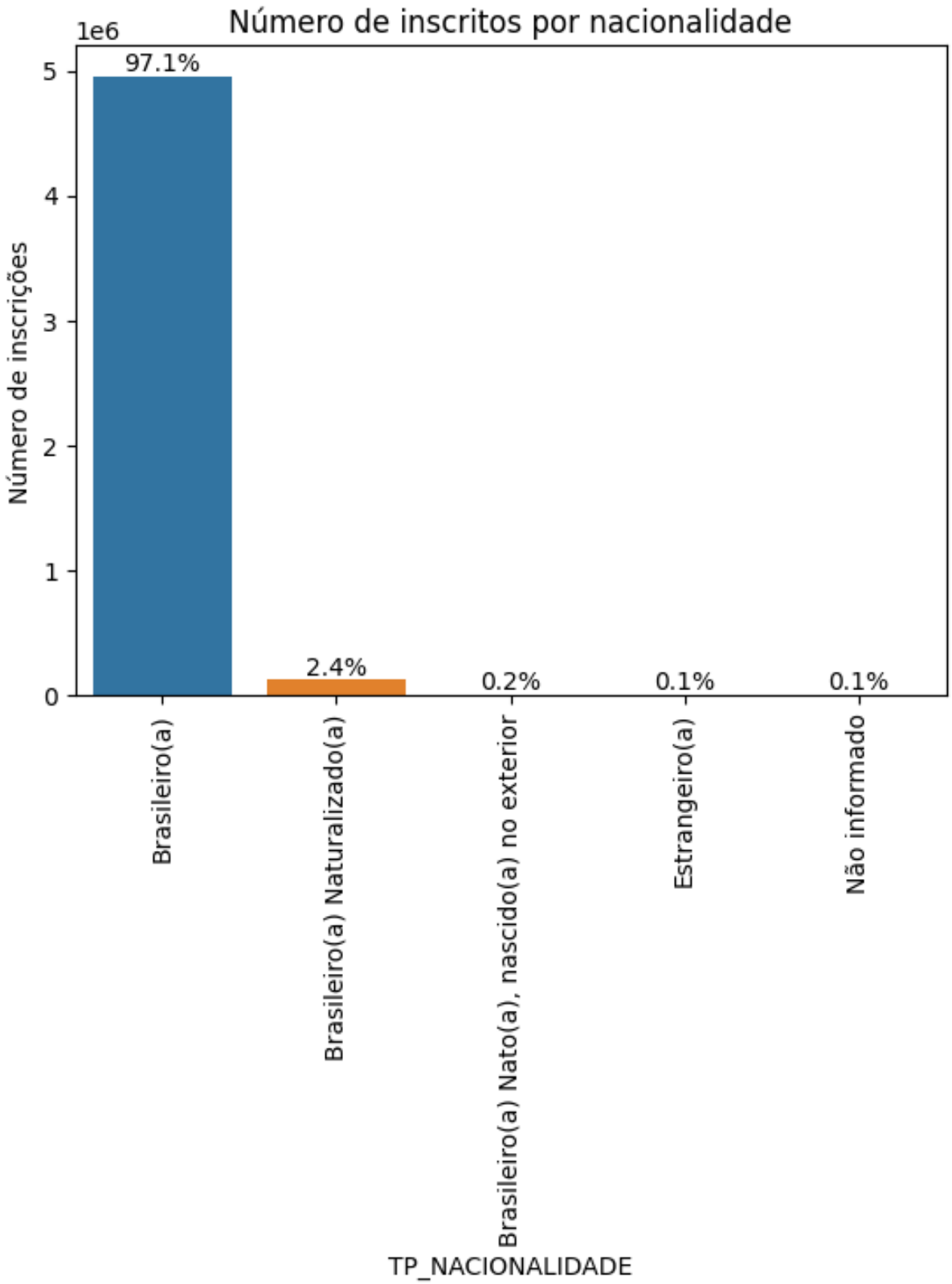

Figura 19: Gráfico mostrando o número de inscrições por nacionalidade

A figura 20 mostra a distribuição das inscrições pelo tipo de escola frequentada e a figura 21 pelo tipo de ensino, representadas pelas colunas TP_ESCOLA e TP_ENSINO, respectivamente.

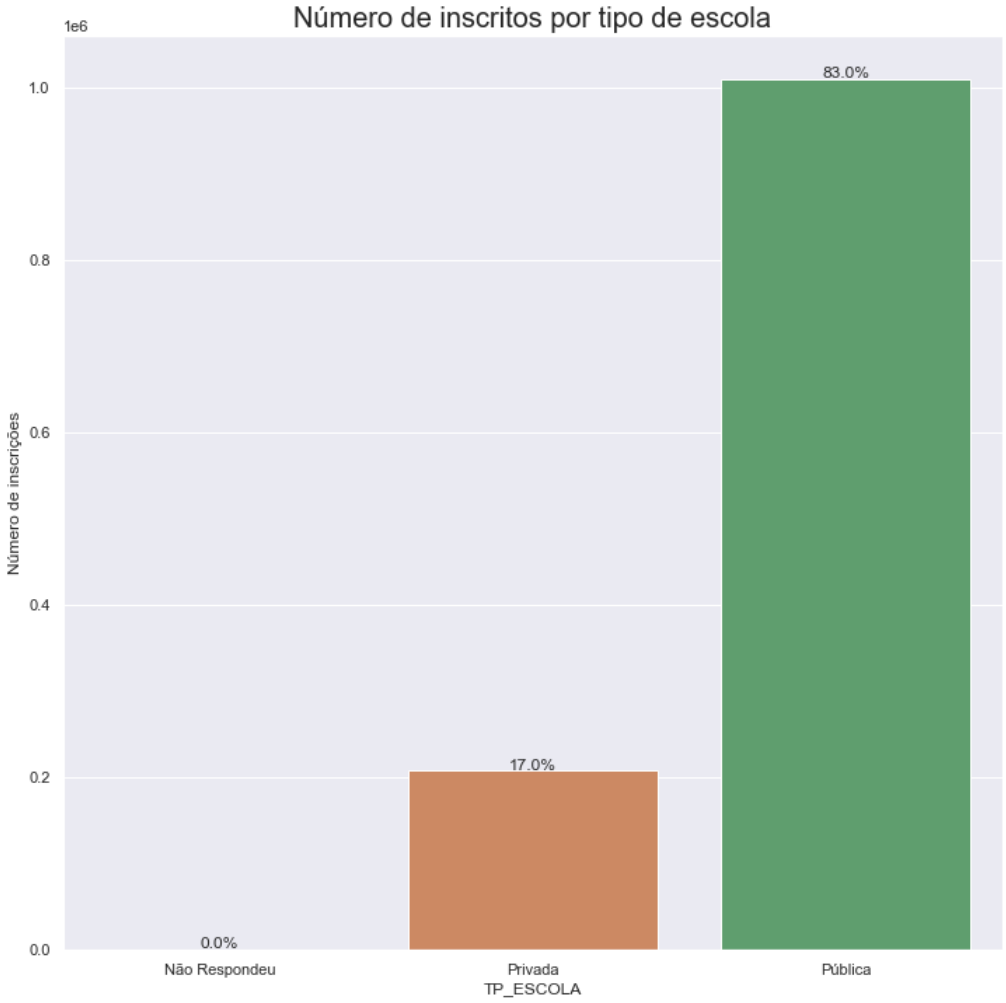

Figura 20: Gráfico mostrando o número de inscrições por tipo de escola

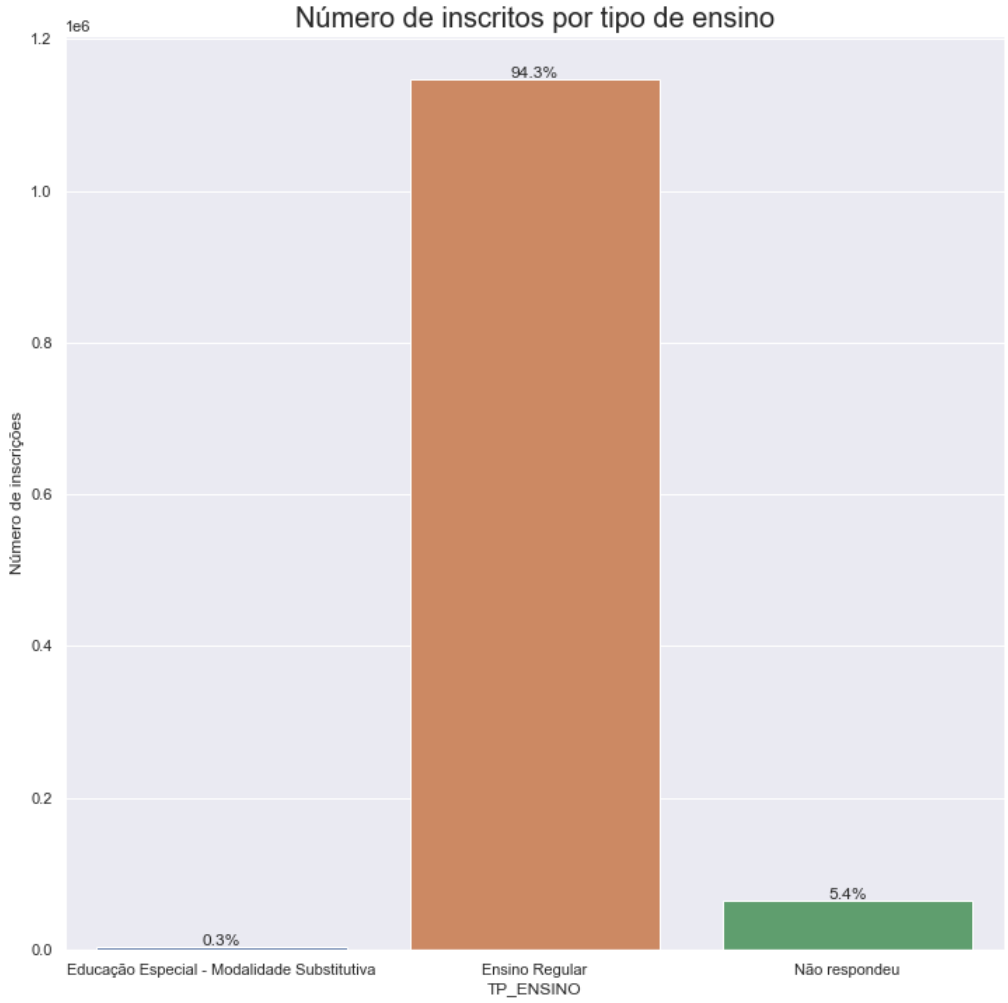

Figura 21: Gráfico mostrando o número de inscrições por tipo de ensino

Por fim, a figura 22 mostra o histograma de idade dos candidatos. A maior parte concentrada entre 16 e 20 anos, o que é esperado já que são egressos do ensino médio. A idade mínima é de 2 anos e a máxima de 83, provavelmente proveniente de erros na coleta dos dados.

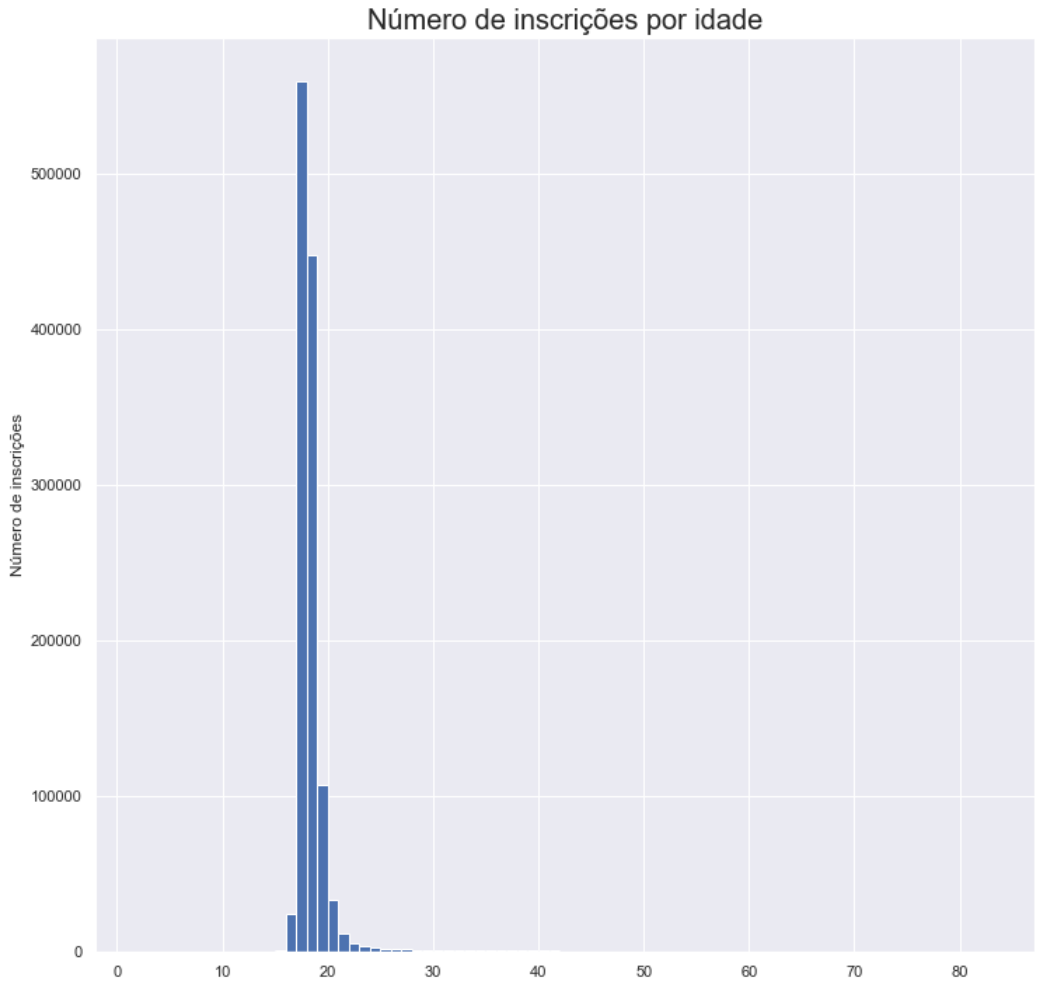

Figura 22: Gráfico mostrando o número de inscrições por idade

5.5.3 Dados da Prova Objetiva e Redação

Os dados da prova objetiva e redação consideram as variáveis relacionadas às provas. São dados de presença dos candidatos nos dias da prova, tipo de prova que realizaram, notas obtidas e respostas que o candidato assinalou e o gabarito.

As figuras 23, 24, 25, 26 e 27 mostram os histogramas das notas das seguintes provas: Linguagens, Códigos e suas Tecnologias (LC); Matemática e suas Tecnologias (MT); Ciências da Natureza e suas Tecnologias (CN); Ciências Humanas e suas Tecnologias (CH) e Redação (REDACAO), representadas pelas respectivas colunas: NU_NOTA_LC; NU_NOTA_MT; NU_NOTA_CN; NU_NOTA_CH; NU_NOTA_REDACAO. A figura 28 mostra o histograma da média simples de todas as notas.

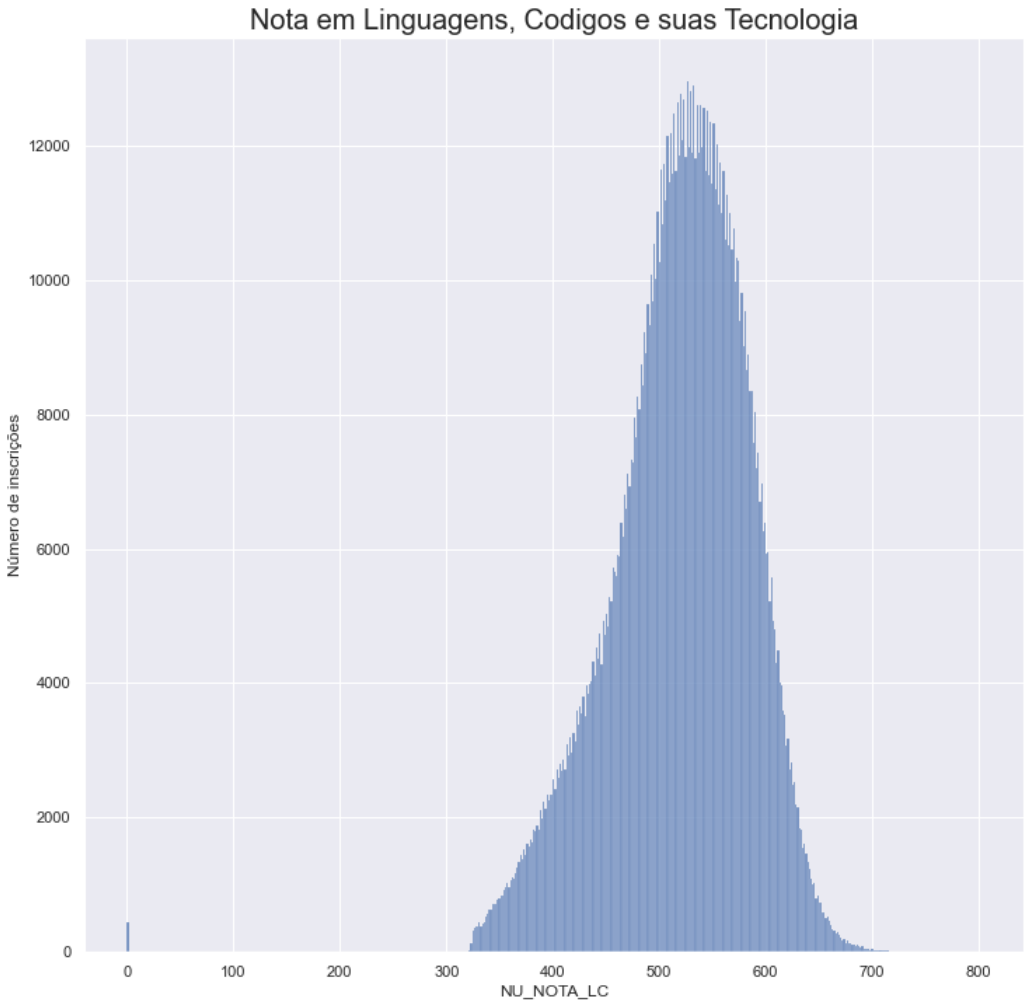

Figura 23: Gráfico mostrando o histograma de notas de Linguagens, Códigos e suas Tecnologias

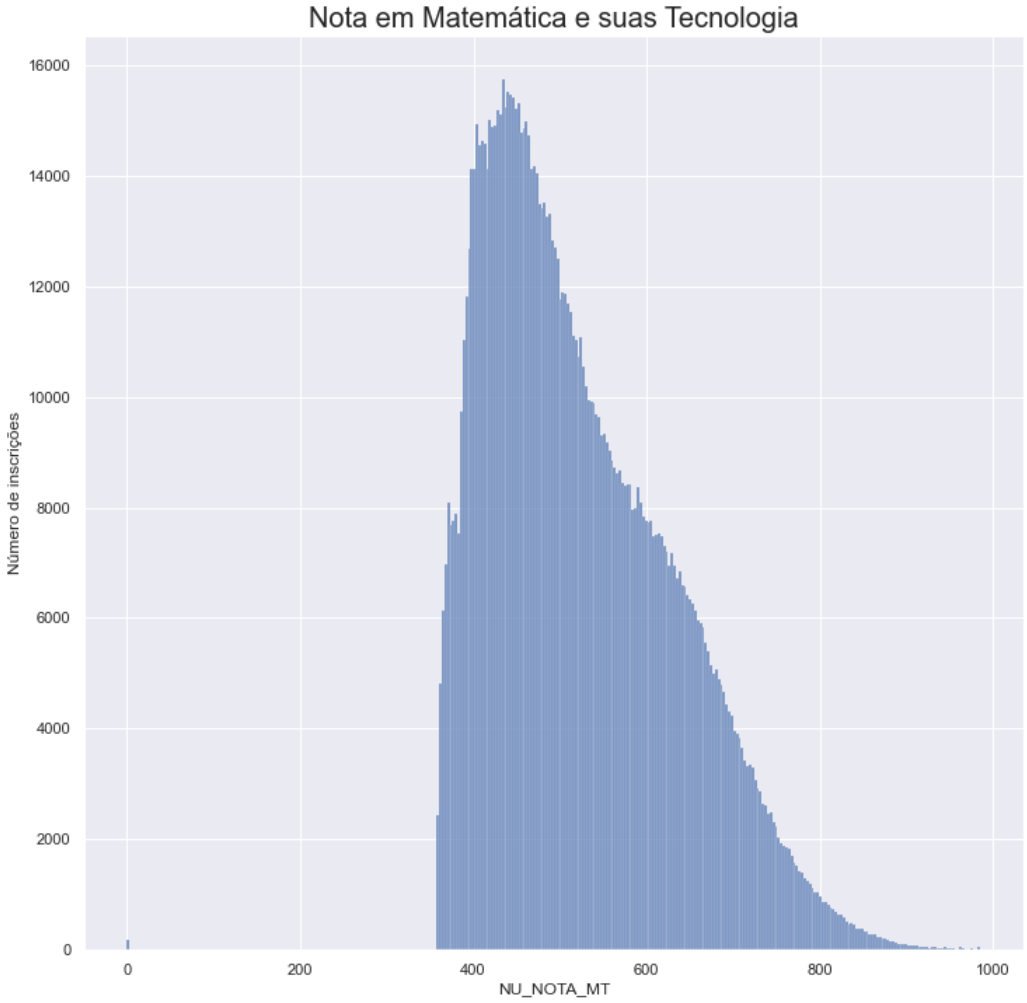

Figura 24: Gráfico mostrando o histograma de notas de Matemática e suas Tecnologias

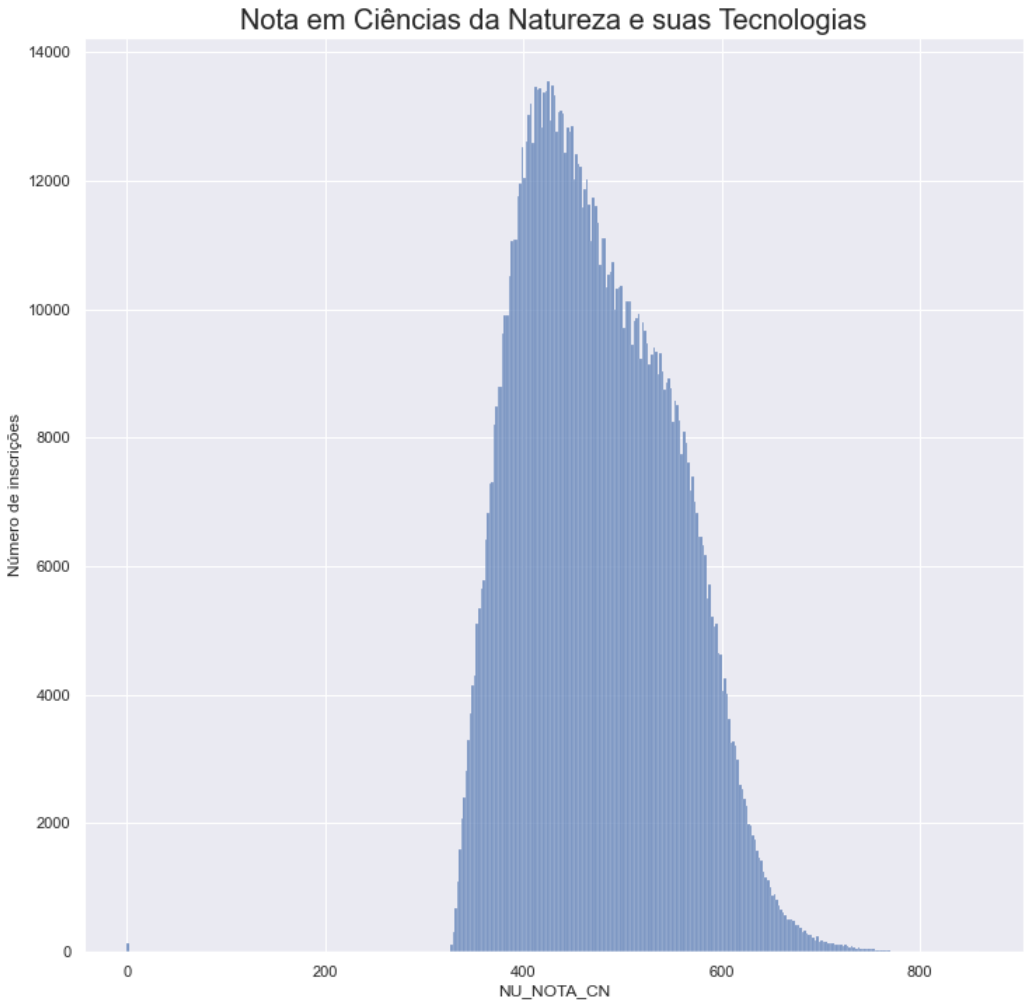

Figura 25: Gráfico mostrando o histograma de notas de Ciências da Natureza e suas Tecnologias

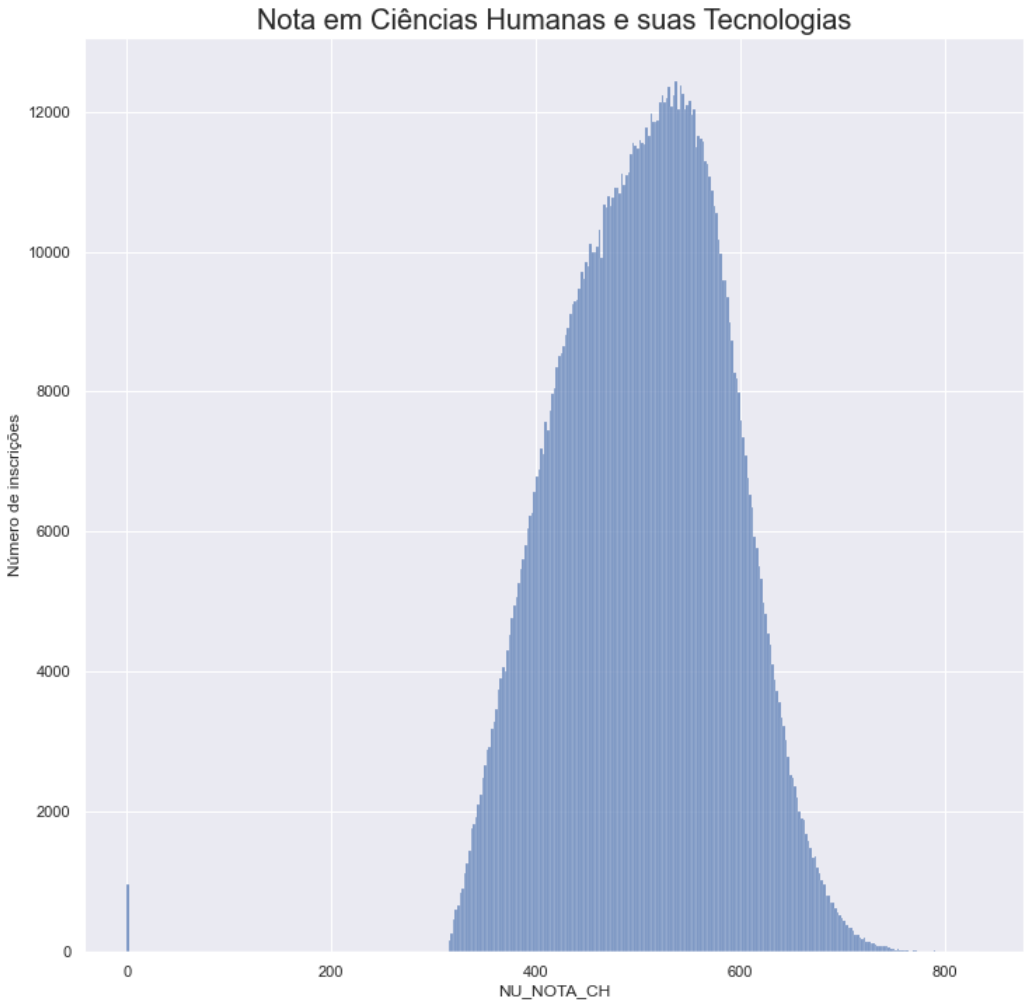

Figura 26: Gráfico mostrando o histograma de notas de Ciências Humanas e suas Tecnologias

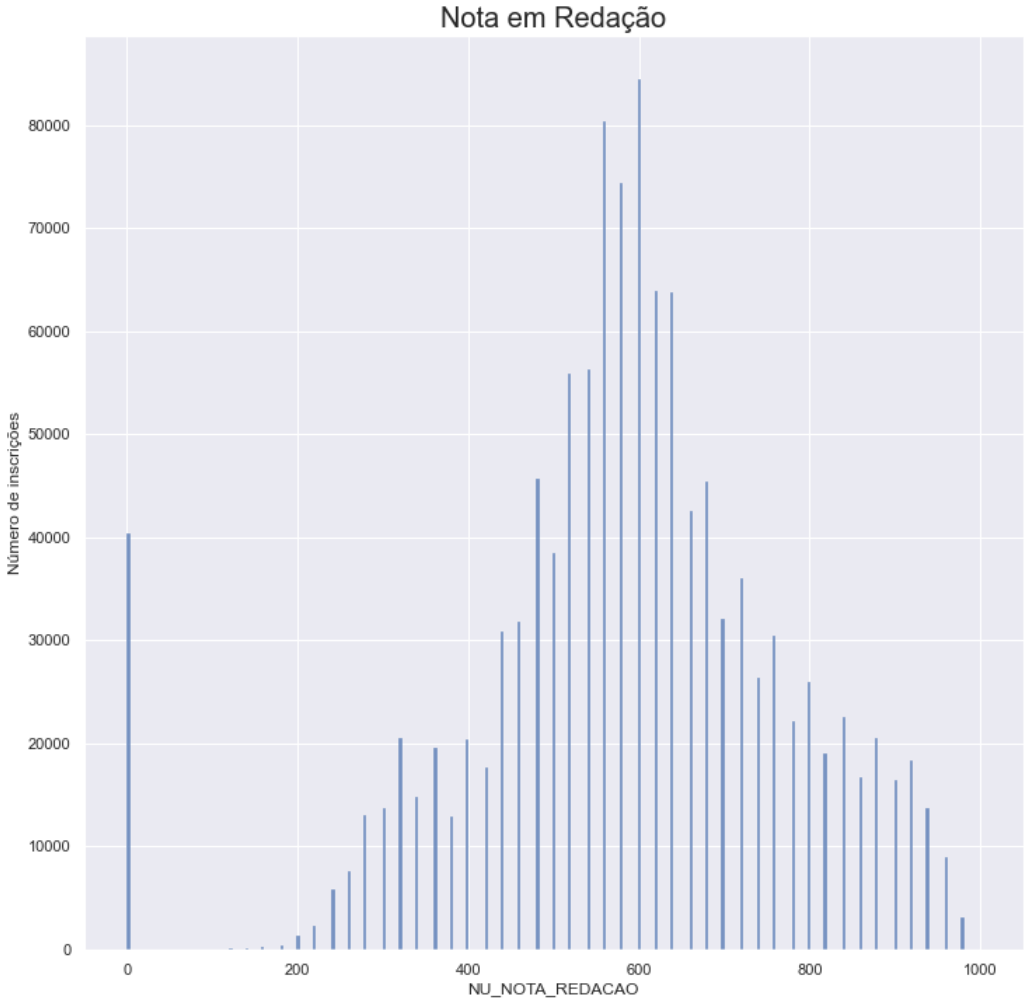

Figura 27: Gráfico mostrando o histograma de notas da Redação

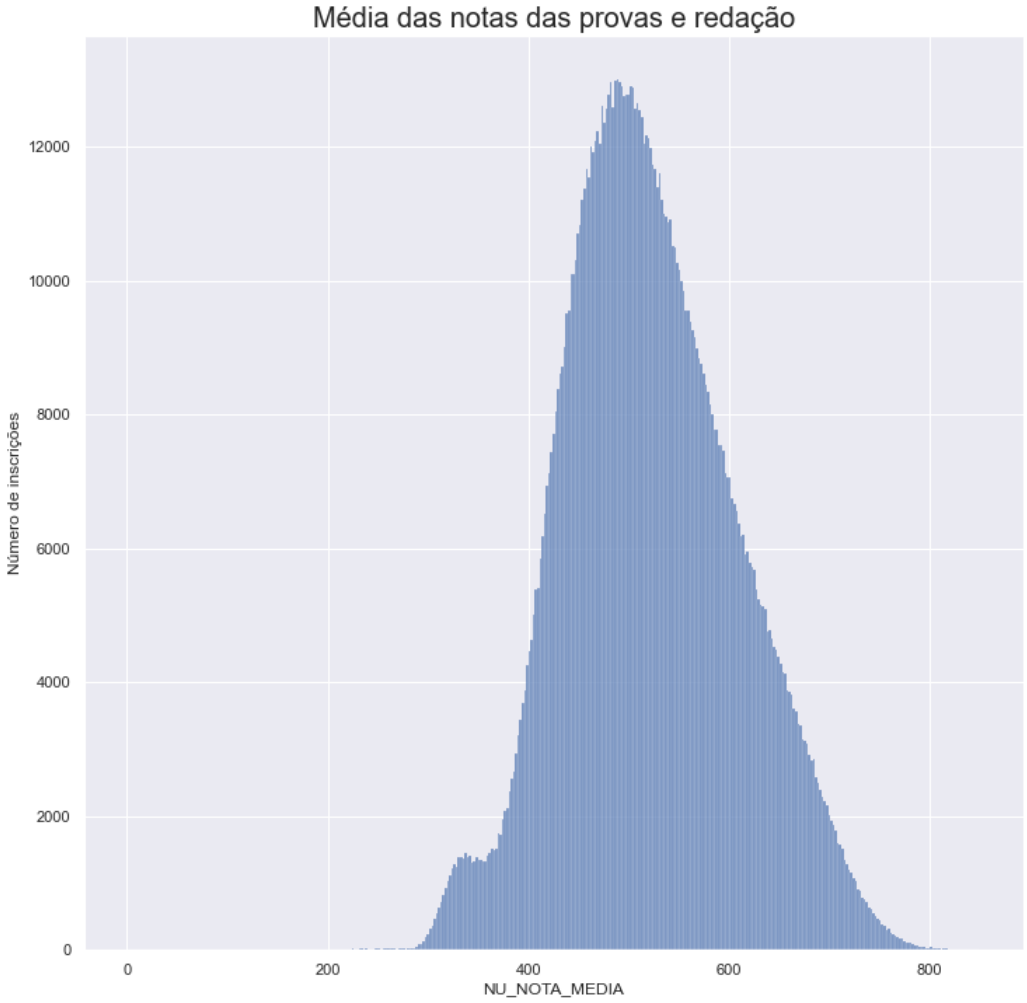

Figura 28: Gráfico mostrando o histograma da média das notas

A tabela 4 mostra algumas estatísticas acerca dessas notas, com valores mínimos e máximos, média, mediana e desvio padrão:

Tabela 4: Tabela com principais estatísticas das notas das provas e redação

prova	média	std	min	mediana	max
Linguagens, Códigos e suas Tecnologias	519	64	0	525	802
Matemática e suas Tecnologias	524	108	0	502	986
Ciências da Natureza e suas Tecnologias	474	75	0	466	861
Ciências Humanas e suas Tecnologias	506	80	0	509	835
Redação	584	190	0	600	1000
Média	521	86	0	514	806

5.5.4 Dados do Questionário Socioeconômico

O questionário socioeconômico compreende em 25 questões acerca das condições familiares e de residência, como ocupação dos pais, número de residentes e se contém diversos equipamentos na residência. Em particular, a questão Q006 é sobre a renda mensal familiar, utilizada como variável de tratamento nesse estudo. A figura 29 mostra a quantidade de inscrições por renda:

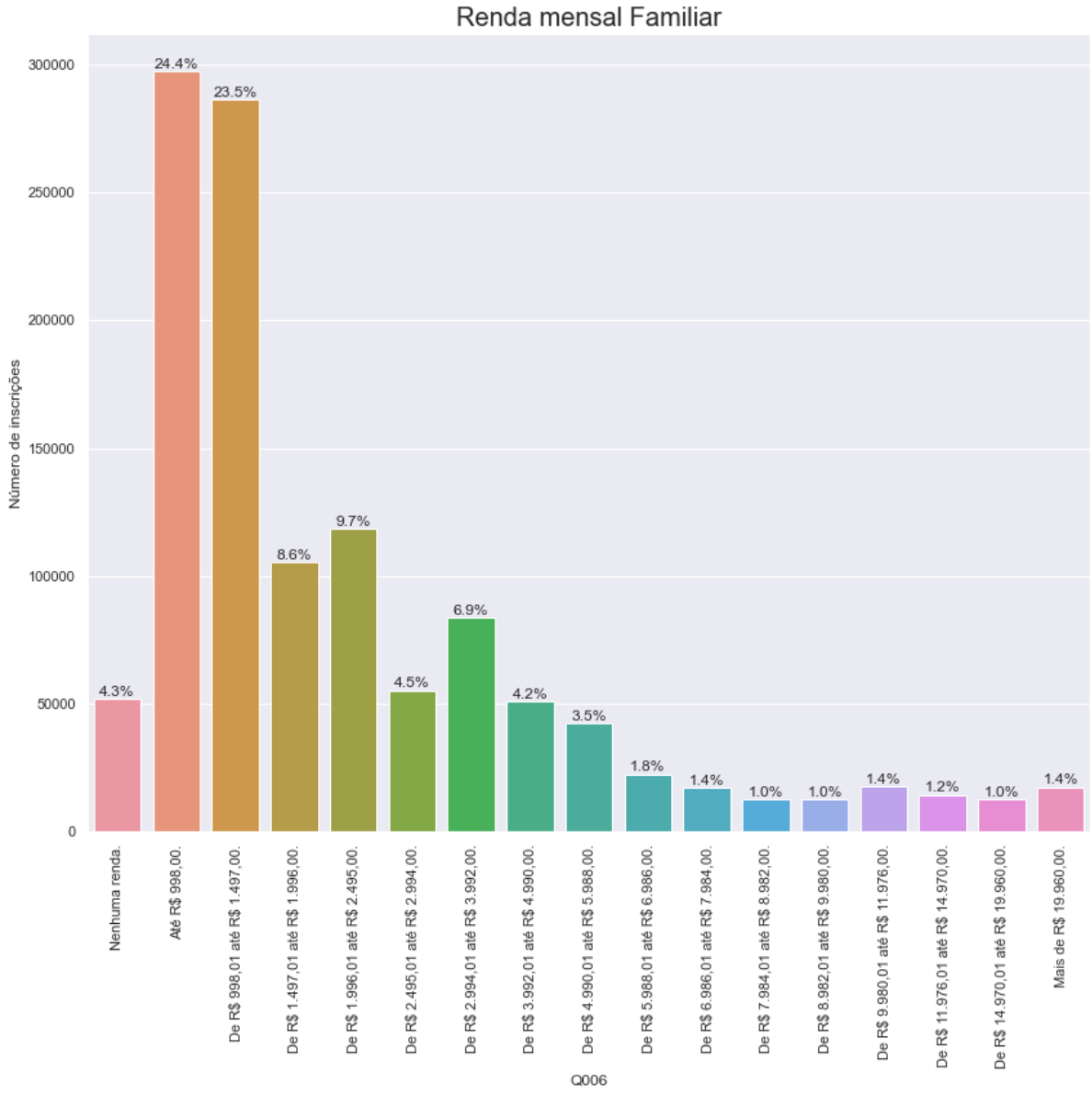

Figura 29: Gráfico mostrando o número de inscrições por renda mensal familiar

Mais da metade dos inscritos têm uma renda familiar de até R$ 1497,00, equivalendo a 1,5 salário mínimo da época em que o exame foi aplicado.

No questionário existem questões com respeito ao grau de escolaridade dos pais (representados pelas questões Q001 para o pai e Q002 para a mãe). Os gráficos 30 e 31 mostram

como foram as distribuições das respostas. Quase metade dos respondentes (47%) tem o pai com grau de escolaridade menor que o ensino médio completo, enquanto para as mães esse valor é de 39,8%.

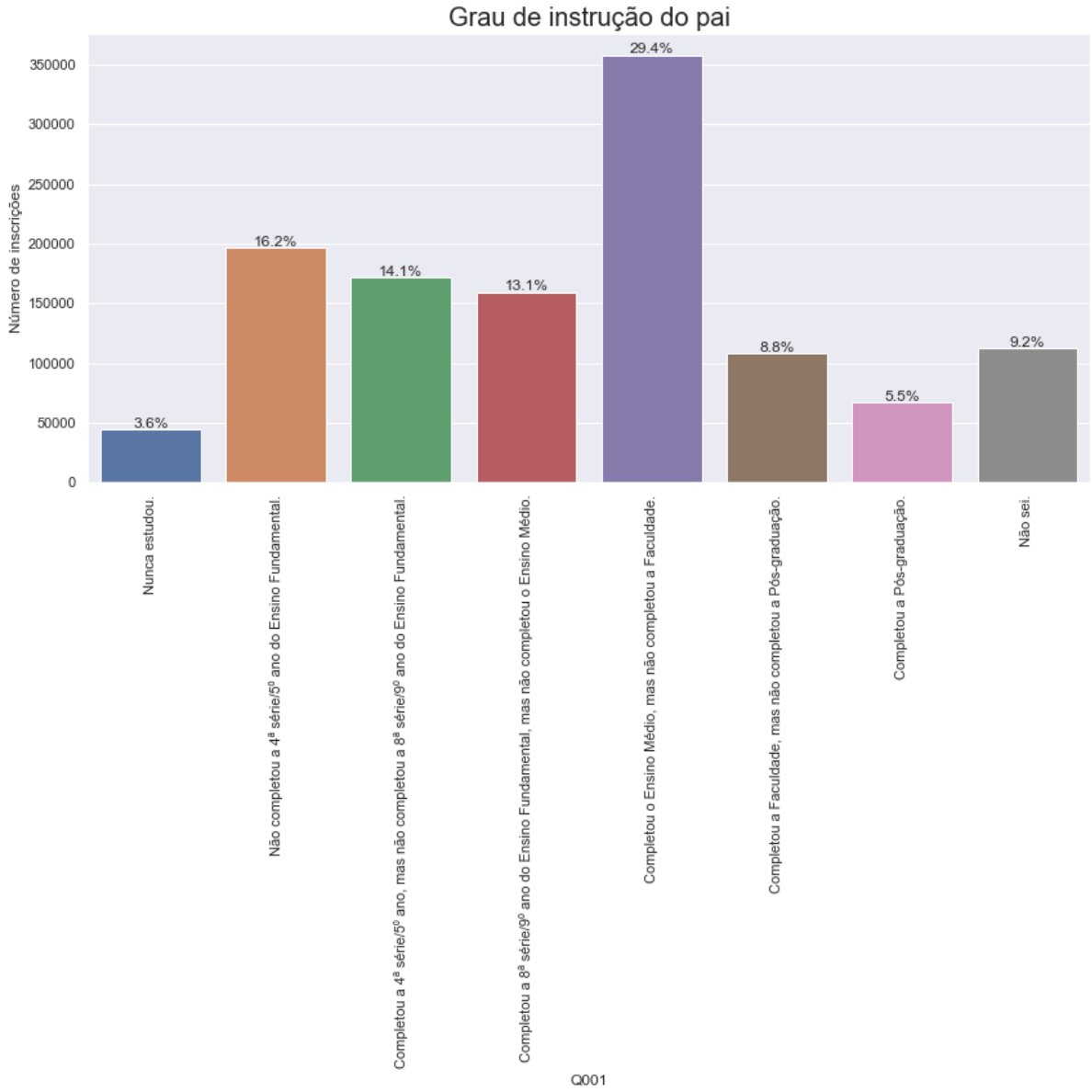

Figura 30: Gráfico mostrando as respostas da pergunta: Até que série seu pai, ou o homem responsável por você, estudou?

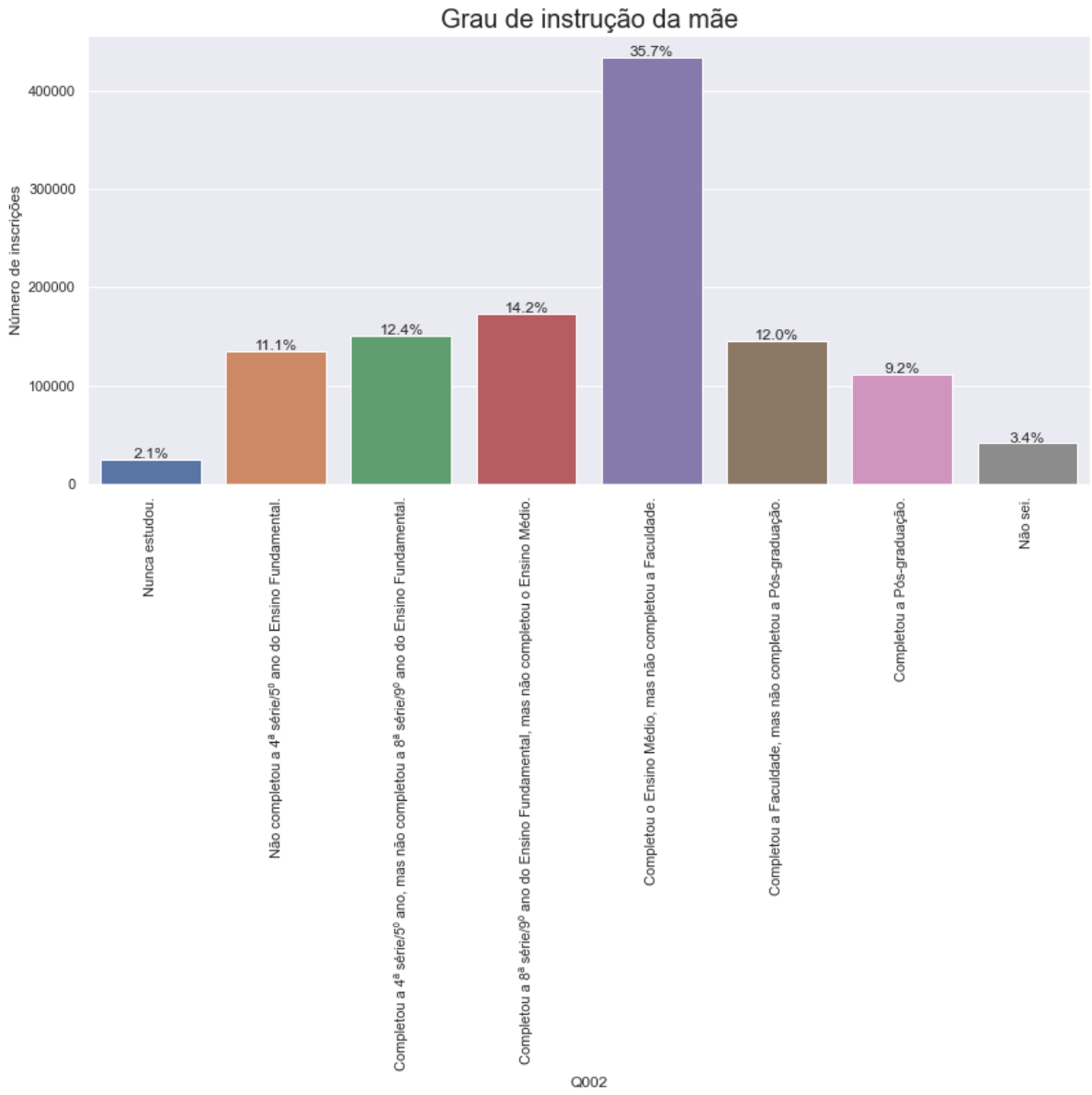

Figura 31: Gráfico mostrando as respostas da pergunta: Até que série sua mãe, ou a mulher responsável por você, estudou?

5.6 Dados Exógenos

Também foram utilizadas algumas variáveis exógenas aos microdados do ENEM, se aproveitando da estrutura do *DataLake* construídos. A primeira é o IDH municipal. O IDH (índice de desenvolvimento humano) é uma métrica de 0 a 1 composta de indicadores de três dimensões: longevidade, educação e renda [36]. Cada uma das dimensões têm o seu próprio valor de IDH, assim como o valor agregado. As figuras 32 e 33 mostram o IDH municipal e o IDH municipal de educação, que serão utilizados nesse estudo. Nota-se que a forma do IDH municipal é normal, enquanto o IDH municipal de educação tem um formato de uma bimodal.

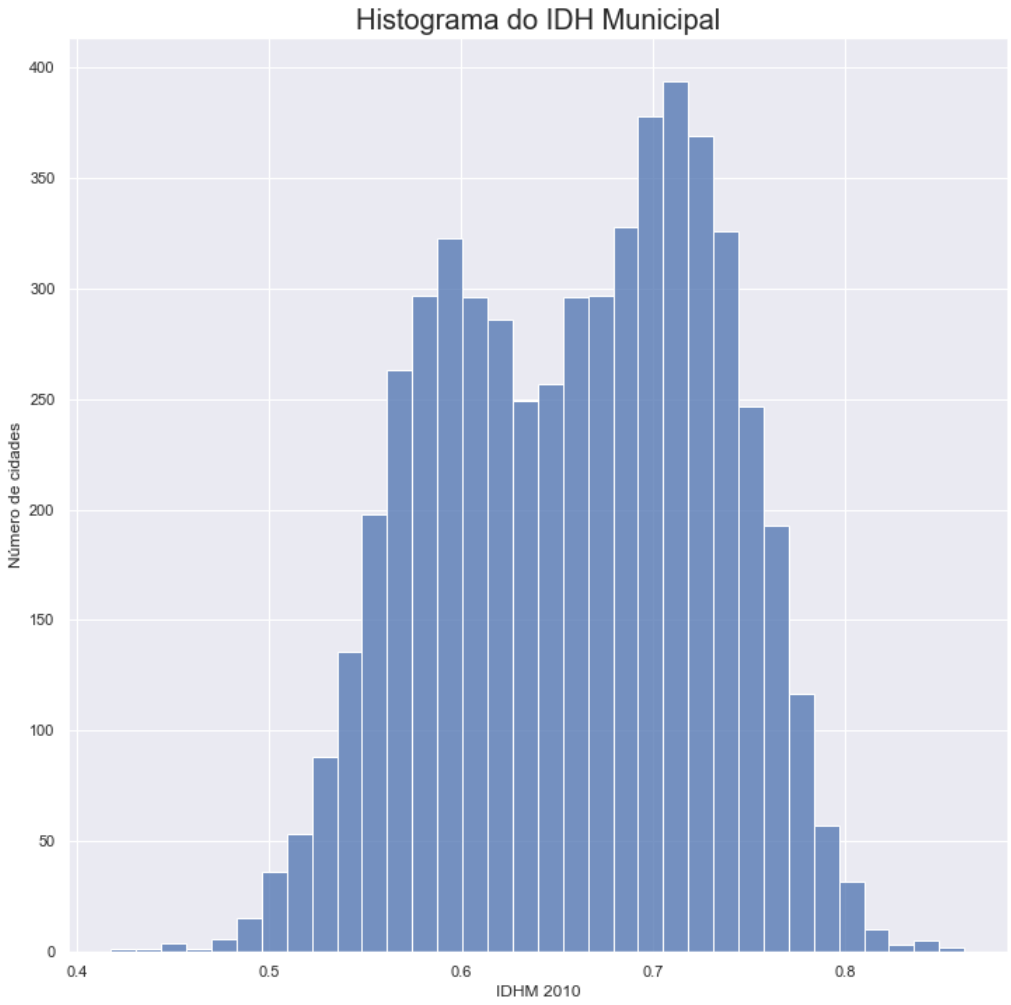

Figura 32: Histograma do IDH municipal

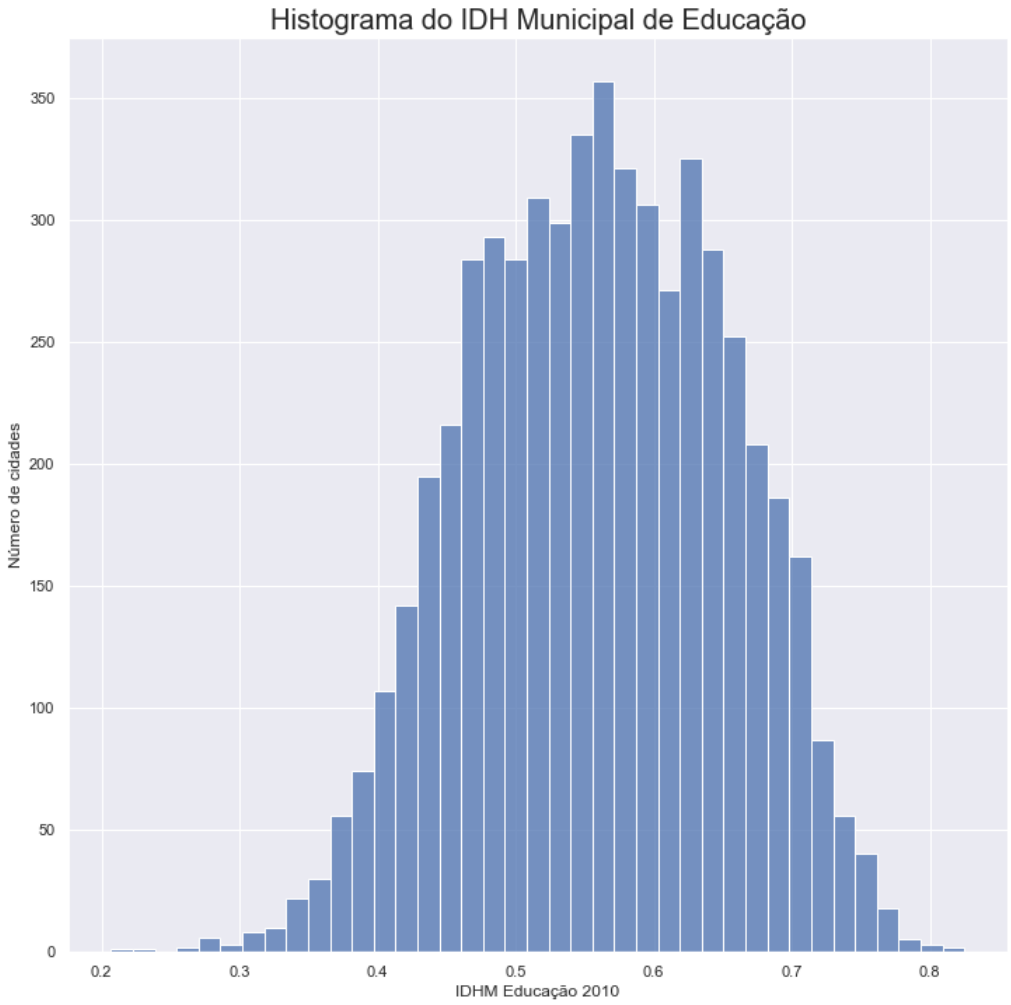

Figura 33: Histograma do IDH municipal de educação

Outro dado exógeno importante para esse estudo é a nota do Índice de Desenvolvimento da Educação Básica (IDEB) [33]. Diferentes colégios podem ter qualidades de ensino variadas. O IDEB pode funcionar como uma *proxy* para a qualidade da educação do colégio. A figura 34 mostra o histograma das notas do IDEB, com a nota mínima sendo 0 e a máxima 7,77.

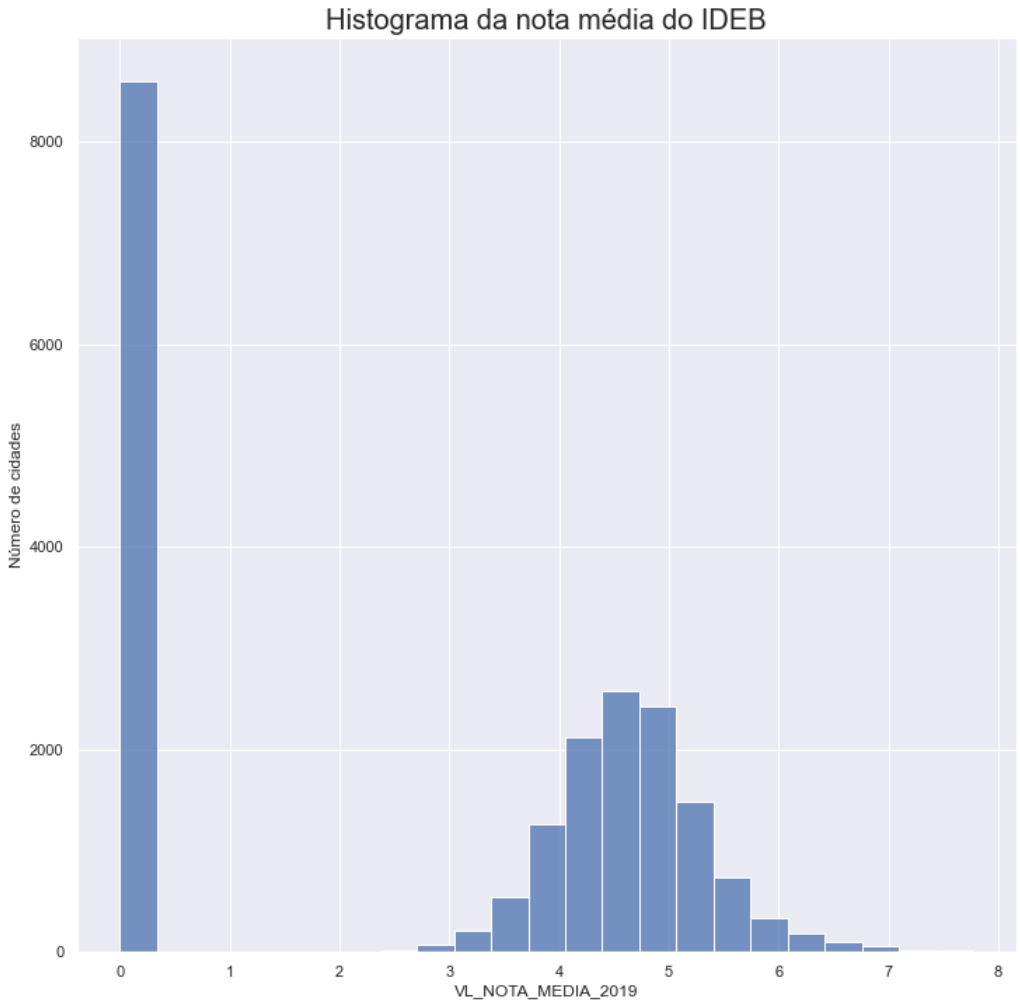

Figura 34: Histograma do IDEB médio

5.7 Modelagem

Para calcular o efeito causal da renda na nota do Enem, foi usada a renda familiar mensal como variável de tratamento e a nota média das provas com a redação como variável *target*. Foi utilizado o método do *x-learner* para o cálculo do efeito de tratamento condicionado, com o modelo base sendo um *gradient boosting*.

Foi realizado um random-search [38] usando *k-fold* igual a 5 e 60 iterações em cada um dos submodelos como forma de tunar os hiperparâmetros. Os hiperparâmetros tunados e os valores testados estão na tabela 5. A métrica alvo utilizada foi a raiz do erro quadrático médio (RMSE em inglês).

As variáveis utilizadas no modelo fora a de tratamento estão na tabela 6:

Tabela 5: Tabela mostrando os hiperparâmetros que foram otimizados e as faixas de valores restadas

hiperparâmetro	**faixa de valores**
n_estimators	randint(20, 1000)
learning_rate	uniform(0.01, 0.3)
num_leaves	randint(20, 3000)
max_depth	randint(3, 13)
min_data_in_leaf	randint(200, 10000)
lambda_l1	randint(0, 100)
lambda_l2	randint(0, 100)

Tabela 6: Tabela com as variáveis usadas como entrada do modelo

Variável	Descrição
SG_UF_RESIDENCIA	Estado de residência do candidato
TP_SEXO	Sexo do candidato
TP_COR_RACA	Cor/raça do candidato
TP_ESCOLA	Tipo de escola do ensino médio
NU_IDADE	Idade do candidato
IDHM (2010)	IDH municipal
IDHM Educação (2010)	IDH municipal educação
IDEB_2019	Nota IDEB 2019

Casos em que a pessoa não respondeu às questões foram tratados como nulas. Foi realizado *one-hot-encoding* nas variáveis categóricas. Essas variáveis foram escolhidas de forma a colocar no modelo todas as variáveis possíveis de serem *confounders* sem inflacionar o modelo com muitas variáveis, já que quanto mais variáveis estiverem no modelo, mais difícil garantir a premissa da positividade.

Como o modelo necessita uma variável de tratamento binária, ela foi construída de forma que todas as pessoas que marcaram que recebiam como renda familiar até R$ 3992,00 (indicativo de 4 salários mínimos, que pertencem às classes D e E) receberam $T = 0$ e, se não, receberam $T = 1$ (pertencentes as classes A, B e C).

A variável objetivo é a nota média do Enem, sendo então o efeito de tratamento a diferença de nota entre ter uma renda familiar maior que R$ 3992,00 e menor.

5.8 Softwares e Bibliotecas Utilizadas

Foi adotado Python como linguagem de programação padrão. Para o *DataLake* foi usado o serviço da Amazon Athena. Os modelos criados na biblioteca pycausal-explorer[9] foram utilizados para esse estudo de caso.

Outras bibliotecas importantes utilizadas são *numpy* para cálculo vetorial, *pandas* para manipulação de dados, *scikit-learn* e *xgboost* para modelos de *machine learning*, *matplotlib*

[9]https://github.com/gotolino/pycausal-explorer

Tabela 7: Tabela com resultado da hiperotimização

	μ_0	μ_1	τ_0	τ_1
n_estimators	539	963	921	847
learning_rate	0.05	0.17	0.11	0.20
num_leaves	362	1355	216	1334
max_depth	11	10	9	12
min_data_in_leaf	233	2851	788	3664
lambda_l1	77	78	23	86
lambda_l2	59	20	26	65
RMSE	63.7	68.7	63.7	68.7

e *seaborn* para visualização.

5.9 Resultados

A tabela 7 mostra o resultado da hiperotimização realizada nos modelos, com o valor de cada hiperparâmetro e o RMSE na validação cruzada.

5.10 Análise dos Resultados

Como não existe um padrão-ouro para comparar os efeitos de tratamento na nota do ENEM pelo fato intrínseco de ser um problema de dados faltantes, a análise do efeito de tratamento será dada qualitativamente, observando como ocorreu o efeito de tratamento em diversos cortes.

Usando uma abordagem ingênua para o cálculo do efeito de tratamento (a média de todos que receberam o tratamento menos a média dos que não receberam) obtém-se o valor de $ATE = 91,2$. Esse *baseline* é utilizado a fim de comparação com o resultado obtido considerando possíveis correlações entre o tratamento e as outras variáveis observadas.

O modelo treinado identificou uma média do efeito de tratamento $ATE = 40,1$. Esse valor se mostra menos que a metade do valor calculado de forma ingênua, mostrando que poderia ter superestimado o valor caso tivesse considerado que os grupos eram equivalentes.

Como o modelo prevê heterogeneidade do resultado com relação aos dados de entrada do modelo, é possível identificar diferentes valores para diferentes grupos de indivíduos. O gráfico 35 mostra como ficaram as distribuições do efeito de tratamento em todos os estados. É possível ver uma diferença desse valor dependendo do estado, onde a mediana pode ir de 32,48 no caso do estado de São Paulo até 54,71 no estado de Roraima.

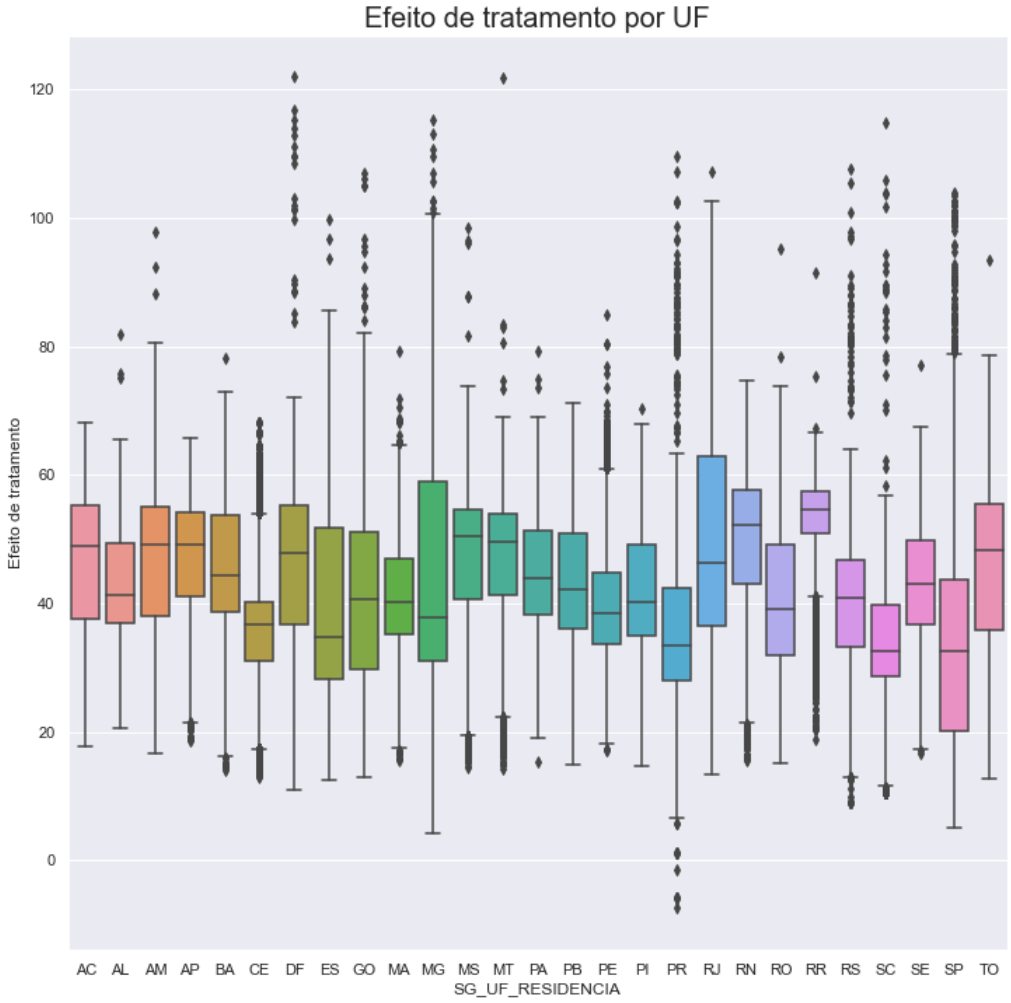

Figura 35: Boxplot do efeito de tratamento por UF

O gráfico 36 mostra usando corte de cor/raça. Apesar do efeito de tratamento ser parecido com esse corte, ainda pode se notar diferença entre brancos e as outras opções. Também há o recorte contra pessoas que nem chegaram a fazer a inscrição para o ENEM, seja por falta de recursos ou falta de incentivo/desinteresse, que também pode mudar esse cenário.

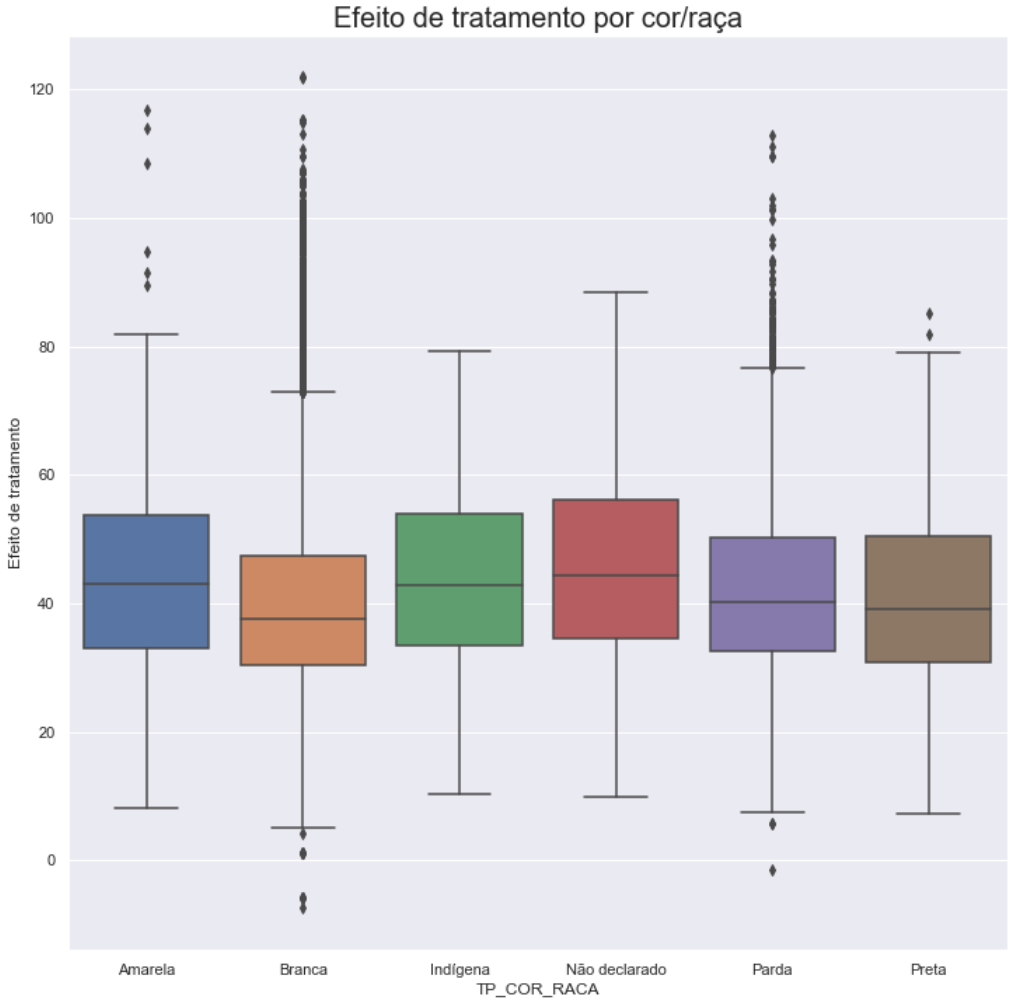

Figura 36: Boxplot do efeito de tratamento por cor/raça

Em geral, esse método pode ser utilizado para guiar políticas públicas, por exemplo, criando políticas específicas para regiões mais afetadas e com um efeito de tratamento maior. Também é mensurável e é possível acompanhar a evolução durante os anos se aplicada a mesma metodologia.

Como trabalhos futuros podem ser realizados diversos cortes de renda e entender como o efeito de tratamento se comporta. Também fazer seleções específicas (por exemplo, pegando apenas a classe E e comparando com a classe A) para entender melhor a dinâmica envolvida.

6 Conclusão

6.1 Considerações Finais

Os avanços na área de *machine learning* trouxeram inúmeras aplicações em diversos campos, desde detecção de spam em emails até diagnósticos médicos em imagens de raio-x. No entanto, à medida que esses métodos ganham popularidade, cresce a necessidade de entender não apenas as correlações entre as variáveis, mas também as relações causais subjacentes. A capacidade de responder à pergunta "E se?"tornou-se crucial em muitos contextos. Por exemplo, será que fumar causa câncer de pulmão, ou essa correlação é influenciada por outras variáveis?

A inferência causal aborda essa necessidade, permitindo a compreensão e identificação de relações de causa e efeito nos dados. O paradoxo de Simpson é um exemplo impactante que ilustra a importância dessa abordagem. Ao considerar cenários em que uma taxa de sucesso global difere das taxas de sucesso em subgrupos, fica evidente que as correlações superficiais podem levar a conclusões enganosas. O estudo da causalidade fornece o arcabouço necessário para discernir essas complexas relações e tomar decisões informadas.

Enquanto experimentos controlados são uma abordagem confiável, eles frequentemente enfrentam desafios práticos. Nesse contexto, os dados observacionais se tornam uma alternativa viável, mas também desafiadora. A falta de controle direto sobre o tratamento e as possíveis interações introduzem incertezas e dificuldades na identificação do contrafactual. Isso significa considerar o que teria acontecido se uma ação alternativa tivesse sido tomada.

Esse trabalho abordou detalhadamente os conceitos teóricos envolvendo a identificação e mensuração de causalidade entre as variáveis. Discutiu os dois maiores *frameworks* utilizados no estudo de inferência causal e fez uma análise comparativa, apontando vantagens e desvantagens do uso de cada *framework*. Além disso, fez uma análise sistemática dos principais modelos utilizados para o cálculo do efeito de tratamento, com sua base teórica e características.

A biblioteca implementada no trabalho não apenas oferece uma ampla gama de métodos de inferência causal, mas também inclui recursos para avaliação e comparação de modelos. Isso permite que os usuários escolham as abordagens mais apropriadas para suas análises, considerando a complexidade dos conjuntos de dados e as perguntas específicas que desejam responder. A compatibilidade com o *sklearn* permite, além de uma entrada facilitada no uso da biblioteca, a possibilidade do uso dos modelos já implementados nela, principalmente quando *metalearners* se utilizam de modelos genéricos como base de sua implementação.

O estudo de caso demonstrou a eficácia do uso da biblioteca para realizar análises

complexas. A investigação sobre a influência da renda per capita familiar nas notas do ENEM e, consequentemente, no acesso ao ensino superior público, destacou a importância de compreender e aplicar o conceito da causalidade.

Espera-se que sendo a sua distribuição *open-source* incentive outros desenvolvedores a contribuir com a biblioteca, tanto com implementações de modelos novos como na criação de novas validações e sugestões de melhorias.

6.2 Trabalhos Futuros

Uma vez que se trata de uma iniciativa de código aberto, fica aberto o convite a todos os desenvolvedores para contribuírem com a biblioteca, disseminando a pesquisa e aplicando-a a novos casos de uso. Como trabalhos futuros está a criação de módulos para seleção de variáveis de confusão. Também está o desenvolvimento de novos modelos, já que o desenvolvimento dos modelos no curso do trabalho não foi exaustivo.

Também pode ser considerada a implementação de módulos que possam ser utilizados em forma de DAGs, levantando a possibilidade de estudo e uso de modelos que têm como base o *framework* de Pearl.

Referências

[1] Crawford, Michael, et al. "Survey of review spam detection using machine learning techniques."Journal of Big Data 2.1 (2015).

[2] Wei, Jian, et al. "Collaborative filtering and deep learning based recommendation system for cold start items."Expert Systems with Applications 69 (2017): 29-39.

[3] LeCun, Yann, et al. "Gradient-based learning applied to document recognition."Proceedings of the IEEE 86.11 (1998): 2278-2324.

[4] Rajpurkar, Pranav, et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning."arXiv preprint arXiv:1711.05225 (2017).

[5] B Neal. Introduction to causal inference: From a machine learning perspective. course lect. *Notes*, 2020.

[6] Clive R Charig, David R Webb, Stephen Richard Payne, and John E Wickham. Comparison of treatment of renal calculi by open surgery, percutaneous nephrolithotomy, and extracorporeal shockwave lithotripsy. *Br Med J (Clin Res Ed)*, 292(6524):879–882, 1986.

[7] Steven A Julious and Mark A Mullee. Confounding and simpson's paradox. *Bmj*, 309(6967):1480–1481, 1994.

[8] Hernán MA, Robins JM (2020). Causal Inference: What If. Boca Raton: Chapman & Hall/CRC.

[9] Naomi Altman and Martin Krzywinski. Points of significance: Association, correlation and causation. *Nature methods*, 12(10), 2015.

[10] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[11] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[12] Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1219. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[13] Jasjeet S Sekhon. The neyman-rubin model of causal inference and estimation via matching methods. *The Oxford handbook of political methodology*, 2:1–32, 2008.

[14] Keith A Markus. Causal effects and counterfactual conditionals: contrasting rubin, lewis and pearl. *Economics & Philosophy*, 37(3):441–461, 2021.

[15] Naftali Weinberger. Comparing rubin and pearl's causal modelling frameworks: a commentary on markus (2021). *Economics & Philosophy*, pages 1–9, 2021.

[16] Künzel, Sören R., et al. "Metalearners for estimating heterogeneous treatment effects using machine learning."Proceedings of the national academy of sciences 116.10 (2019): 4156-4165.

[17] Alicia Curth and Mihaela van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1810–1818. PMLR, 2021.

[18] Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

[19] Daniela Witten and Gareth James. *An introduction to statistical learning with applications in R.* springer publication, 2013.

[20] Constantin F Aliferis, Ioannis Tsamardinos, Alexander R Statnikov, and Laura E Brown. Causal explorer: A causal probabilistic network learning toolkit for biomedical discovery. In *METMBS*, volume 3, pages 371–376. Citeseer, 2003.

[21] Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

[22] Amit Sharma, Emre Kiciman, et al. DoWhy: A Python package for causal inference. https://github.com/microsoft/dowhy, 2019.

[23] Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/microsoft/EconML, 2019. Version 0.x.

[24] Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. Causalml: Python package for causal machine learning, 2020.

[25] Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[26] Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620, 1986.

[27] Alicia Curth, David Svensson, Jim Weatherall, and Mihaela van der Schaar. Really doing great at estimating cate? a critical look at ml benchmarking practices in treatment effect estimation. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*, 2021.

[28] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. Realcause: Realistic causal inference benchmarking. *arXiv preprint arXiv:2011.15007*, 2020.

[29] John E Roemer. Equality of opportunity: A progress report. *Social Choice and Welfare*, pages 455–471, 2002.

[30] Francisco HG Ferreira and Jérémie Gignoux. The measurement of educational inequality: Achievement and opportunity. *The World Bank Economic Review*, 28(2):210–246, 2014.

[31] Ezekiel J Dixon-Román, Howard T Everson, and John J McArdle. Race, poverty and sat scores: Modeling the influences of family income on black and white high school students' sat performance. *Teachers College Record*, 115(4):1–33, 2013.

[32] Erik Figueirêdo, Lauro Nogueira, and Fernanda Leite Santana. Igualdade de oportunidades: Analisando o papel das circunstâncias no desempenho do enem. *Revista Brasileira de Economia*, 68(3):373–392, 2014.

[33] `http://portal.inep.gov.br/ideb` (acessado em 16/08/2020).

[34] `http://portal.inep.gov.br/censo-escolar` (acessado em 16/08/2020).

[35] `http://portal.inep.gov.br/enem` (acessado em 16/08/2020).

[36] `https://www.br.undp.org/content/brazil/pt/home/idh0/conceitos/o-que-e-o-idhm.html` (acessado em 17/05/2022).

[37] `https://www.ibge.gov.br/cidades-e-estados` (acessado em 08/05/2022).

[38] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.