UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Trabalho de Conclusão de Curso

Eduardo Augusto Marinho

Aplicando XAI na comparação de redes neurais e árvores de decisão

São Carlos - São Paulo

2024

Eduardo Augusto Marinho

Aplicando XAI na comparação de redes neurais e árvores de decisão

Monografia apresentada à Universidade Federal de São Carlos como parte dos requisitos necessários para a obtenção do grau no curso de Engenharia de Computação.

Orientação Profa. Dra. Marcela Xavier Ribeiro

São Carlos - São Paulo

2024

Agradecimentos

Agradeço a meu pai, José e minha mãe, Mariza, por todo apoio e cuidado. Agradeço minha irmã, Geíza, e minhas sobrinhas, Betina e Gabriela, pelas ajudas nos momentos mais difíceis. Também agradeço aos meus amigos que acompanharam esta jornada da graduação, pelo companheirismo e momentos de descontração. Por fim, agradeço aos professores da UFSCar, em especial aos professores do Departamento de Computação, pela formação profissional e acadêmica.

Resumo

O novo enfoque em inteligência artificial nos depara com uma preocupação de longa data sobre tais algoritmos, a distinção entre modelos de inteligência artificial *white-box* e *black-box*. Este estudo busca explorar técnicas de análise, conhecidas como IA Explicável (XAI) para tais modelos, em especial nos algoritmos chamados *black-box*, aqueles em que os detalhes de sua tomada de decisão não são completamente conhecidos. É importante a explicabilidade do modelo de IA, pois os modelos opacos podem ocultamente infligir questões éticas e de confiabilidade, incluindo a possibilidade de viés, discriminação, questões de privacidade e violações de direitos. A abordagem escolhida visa estudar e aplicar algumas destas técnicas de XAI para desvendar a lógica de um algoritmo considerado *black-box*, usando uma abordagem progressiva, que explora os fundamentos de uma rede neural até a aplicação de técnicas explicáveis, apresentado uma comparação de comportamento entre redes neurais e árvore de decisão. Por fim, são feitas comparações entre métricas dos modelos e os resultados encontrados são discutidos.

Palavras-chave: XAI, IA explicável, redes neurais, white-box, black-box, árvore de decisão.

Abstract

The new focus on artificial intelligence brings us face to face with a long-standing concern about such algorithms: the distinction between *white-box* and *black-box* artificial intelligence models. This study aims to explore analysis techniques, known as Explainable AI (XAI), for such models, especially focusing on *black-box* algorithms, where the details of their decision-making process are not fully known. It highlights ethical concerns and reliability issues associated with these opaque algorithms, including the possibility of bias, discrimination, privacy issues, and rights violations. The chosen approach seeks to study and apply some of these XAI techniques to unravel the logic of a *black-box* algorithm. The progressive work begins exploring the fundamentals of neural networks until we reach the Explainable AI techniques, including a comparison between neural networks and decision trees. Finally, comparisons between model metrics are made, and the results are discussed.

Keywords: XAI, Explainable AI, neural network, white-box, black-box, decision tree.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Apresentação do tema

A inteligência artificial (IA) é uma área de pesquisa em constante evolução que busca desenvolver sistemas capazes de realizar tarefas que, normalmente, demandam inteligência humana. Este campo interdisciplinar combina conceitos da computação, matemática e estatística para criar algoritmos e modelos que possam aprender padrões, tomar decisões e realizar ações autônomas.

Desde suas raízes na década de 1950, a IA experimentou avanços significativos, influenciando diversas esferas da sociedade, desde assistentes virtuais até diagnósticos médicos. Recentemente, IA tomou um novo protagonismo nas discussões no mundo da tecnologia com o lançamento público de diversos *large language models* (LLM), impulsionados por imensos conjuntos de dados, além de avanços computacionais e nos algoritmos que viabilizaram esses novos modelos, sendo estes muito mais capazes do que os modelos acessíveis ao público geral até então.

Ao mesmo tempo, diversos modelos menos conhecidos e mais restritos são utilizados por serviços e instituições, e muitas vezes a lógica por trás deles é também completamente desconhecida, e com isso surgem diversas preocupações sobre esses modelos, como questões de viés, sociais e éticas (GUIDOTTI et al., 2018), já que muitas vezes eles funcionam de maneira ofuscada, sem um entendimento claro de seu funcionamento.

A IA Explicável (XAI) procura empregar técnicas sobre esses modelos com o objetivo de aprimorar o entendimento de suas tomadas de decisões. Essas técnicas envolvem comparações com modelos conhecidos e mais facilmente compreensíveis, análise por meio de modificações de variáveis, elaboração de um modelo aproximado a partir dos resultados de predição, e diversas outras estratégias (ADADI; BERRADA, 2018).

1.2 Justificativa

A falta de clareza dos modelos de IA *black-box* geram preocupações sobre como suas decisões são tomadas, e assim, destacamos as preocupações éticas e questões de confiabilidade associadas a estes algoritmos opacos, incluindo a possibilidade de viés, discriminação, questões de privacidade, violações de direitos, entre outros vícios que podem estar escondidos, de maneira não intencional, na tomada de decisão do algoritmo.

Então, a existência desses algoritmos *black-box* pode ser considerada uma barreira na adoção desses modelos de aprendizado de máquina, devido a essas questões de cunho

ético e de confiabilidade, em especial considerando-se que conforme suas complexidades aumentam, eles ficam ainda mais ofuscados.

Assim, com a explicabilidade e interpretabilidade buscamos elevar a confiança em modelos de IA, para que possam ser utilizados sem receios, possibilitar a criação de ferramentas para controle dos modelos por meio de um entendimento mais profundo de seu funcionamento e ajudar a guiar desenvolvimentos futuros em IA direcionados pela ética.

1.3 Objetivos

Os modelos de inteligência artificial são classificados de duas formas, os que são considerados *white-box*, ou seja, aqueles em que o resultado é inteligível, ou seja, um especialista no assunto pode entender o que o algoritmo está propondo e os *black-box*, aqueles em que não é compreensível, ou seja, não é possível ter o conhecimento de como o algoritmo chegou ao seu resultado.

Este trabalho busca estudar, utilizar e avaliar técnicas de inteligência artificial explicável, com entendimento teórico e prático, para aumentar a interpretabilidade e auxiliar no entendimento de técnicas *black-box*. Aqui é explorado a compreensão sobre redes neurais em si, desde sua forma mais básica até mais complexa, usando métricas de avaliação para comparar com modelos *"white-box"*. Mais especificamente, este trabalho objetiva aplicar técnicas de IA explicável (XAI) em redes neurais ("black box"), adquirindo conhecimento de como funcionam e como podem ser aplicadas, e compará-las com árvores de decisão ("white-box"), verificando similaridades e diferenças entre os modelos de aprendizado e dos resultados alcançados.

1.4 Metodologia

A metodologia do estudo foi baseada em revisão bibliográfica e aplicação em diferentes conjuntos de dados de redes neurais e árvores de decisão e técnicas de inteligência artificial explicável - XAI.

Para a aplicação de redes neurais, foram feitas escolhas de bases de dados que possibilitariam identificar parcialidade nas decisões dos algoritmos, com bases que levam em consideração fatores circunstanciais de pessoas, como dados demográficos contendo etnia, gênero, nacionalidade entre outros fatores que possam gerar parcialidade.

Neste trabalho utilizou-se uma metodologia incremental: inicialmente fez-se o estudo e aplicação de uma rede neural elementar (perceptron) em conjuntos de dados simplificados. Posteriormente, foram feitas aplicações de redes neurais mais complexas para bases de dados mais complexas. Também foram gerados modelos *white-box*, ou entendíveis, árvores

de decisão, para estas mesmas bases de dados. Esses modelos *white-box* foram usados para fins de comparação. Nesse ponto, foi feita a avaliação do modelo por meio de métricas de eficácia de modelos de classificação para fins comparativos.

Posteriormente, iniciou-se a fase de compreensão, empregando técnicas de inteligência artificial explicável. Nesta etapa buscamos entender ou tornar transparentes os processos de decisão das redes neurais, proporcionando interpretação e entendimento sobre como o modelo opera.

Esse enfoque incremental e progressivo, desde conceitos fundamentais até a implementação de técnicas de explicabilidade, visou aprofundar o entendimento sobre o funcionamento interno dos modelos de aprendizado de máquina, contribuindo para uma abordagem mais explicável e ética no desenvolvimento de modelos de IA.

1.5 Estrutura do trabalho

O trabalho está estruturado em 5 partes, começando capítulo 1 com a introdução ao tema e sua justificativa e a motivação do estudo, metodologia e os objetivos a serem alcançados. O capítulo 2 possui a fundamentação teórica, focando nos princípios teóricos necessários para o entendimento de redes neurais e técnicas explicáveis. O capítulo 3 contém a revisão bibliográfica de técnicas explicáveis utilizadas no estudo. O capítulo 4 detalha todo o desenvolvimento feito ao longo do estudo, com detalhes sobre as redes neurais, árvores de decisão, bases de dados escolhidas, descrições dos testes e exposição dos resultados encontrados. E, por fim, o capítulo 5 apresenta as conclusões encontrados, discussão sobre os resultados e sugestões para pesquisas futuras.

2 Fundamentação Teórica

2.1 Machine Learning

Machine learning (ML), ou aprendizado de máquina, é uma das áreas contidas no estudo de inteligência artificial, e visa desenvolver algoritmos e modelos capazes de aprender padrões e realizar predições a partir de dados. Esses algoritmos, diferente dos algoritmos clássicos, são feitos de maneira que seu comportamento se ajusta conforme eles adquirem experiência durante seus treinamentos. Essa capacidade de adaptação é o que torna a área de machine learning essencial em aplicações que envolvem complexidade e grandes volumes de dados.

A fase de treinamento desses modelos tipicamente envolve expor o algoritmo a um conjunto de dados conhecidos, e similares aos dados aos quais o modelo é aplicado, assim possibilitando o reconhecimento de padrões e relações entre as características desses dados, e essa capacidade de fazer predições com base nesses dados de treino é chamada de generalização.

Existem diversas categorias de aprendizado, como o aprendizado supervisionado e o não supervisionado. No aprendizado supervisionado, o algoritmo tem acesso aos rótulos de classificação da base de dados em que ele é treinado, enquanto no não supervisionado o algoritmo tenta identificar padrões sem saber a rotulação dos dados.

Entre os algoritmos mais comuns de machine learning temos redes neurais, árvore de decisão, k-nearest neighbors (k-NN) e máquina de vetores de suporte (SVM), cada um tendo fundamentos e características de funcionamento distintas, também diferindo em suas eficácias e interpretabilidade. Neste trabalho são explorados os dois primeiros algoritmos mencionados.

2.2 Redes neurais

Propostas inicialmente por Warren McCulloch e Walter Pitts em 1943 (MCCULLOCH; PITTS, 1943), redes neurais artificiais são modelos computacionais inspirados na estrutura de um neurônio que busca reproduzir a forma como um cérebro humano toma decisões.

Em sua forma mais simples, o neurônio artificial é uma estrutura com saída binária, que recebe uma série de sinais de entrada I_n, sendo cada uma dessas entradas multiplicadas por um peso distinto W_n, que uma vez somadas, passam por uma função de ativação (Figura 1), que definirá se o neurônio será ativado, equivalente a um valor 1, ou não,

equivalente ao valor 0. O aprendizado de máquina ocorre na adequada atualização do valor de cada um dos pesos, com base na diferença entre as saídas previstas e os rótulos reais dos dados de treinamento a cada iteração, usando uma determinada taxa de aprendizado.

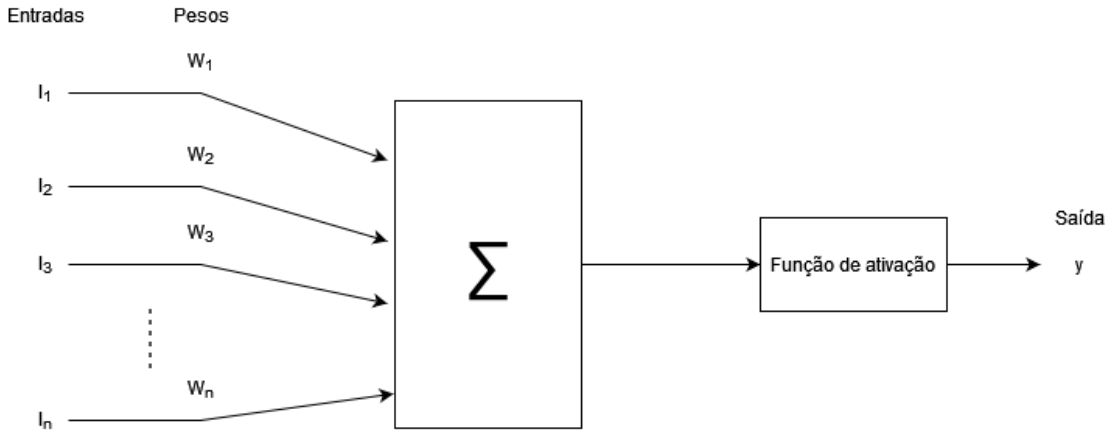

Figura 1 – Modelo simples de um perceptron. Fonte: Autor

Sistemas mais complexos baseados neste conceito podem dispor de diversos neurônios organizados em uma camada, conhecido como single layer perceptron, ou múltiplas camadas de diversos neurônios, chamado de multilayer perceptron.

2.2.1 Perceptron

A aplicação mais básica de uma rede neural se dá na forma de um perceptron, ele é uma aplicação direta de um neurônio com entradas, pesos e taxa de aprendizado.

Por se tratar de um modelo simples, sua capacidade de classificação não é muito elevada, sendo usado como um classificador linear, ou seja, uma reta, ou um plano ou um hiperplano, como o exibido na Figura 2.

Em geral, o perceptron irá dividir a base de dados em que ele está sendo aplicado em duas partes separáveis, sendo esta sua principal limitação. As funções de ativação para modelos de perceptron também costumam ser simples, como a função de ativação com limiar, comumente valor zero:

$$\phi(x) = \begin{cases} 0 & \text{se } x < 0 \\ 1 & \text{se } x \geq 0 \end{cases}$$

2.2.2 Multilayer Perceptron

O Multilayer Perceptron (MLP) é uma extensão do conceito inicial do perceptron, visando contornar as limitações de classificação do modelo simples.

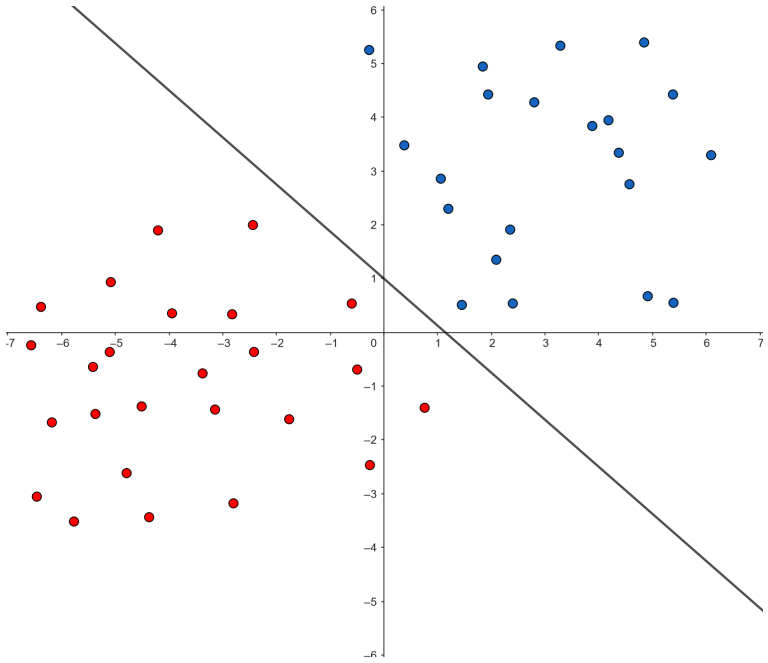

Figura 2 – Exemplo linearmente separável. Fonte: Autor

Consiste em uma rede neural, onde os neurônios se organizam em camadas, possuindo uma camada de entrada, uma camada de saída, e camadas intermediárias, também conhecidas como camadas ocultas. Um dos fundamentos de *Deep Learning* é a presença de grandes quantidades de camadas intermediárias, permitindo modelos de alta complexidade e possibilitando a extração de relações entre as características e reconhecimento de padrões que os modelos mais simples não são capazes de detectar, e assim sendo capaz de resolver problemas mais desafiadores.

Devido a esta estrutura de camadas, a saída de cada neurônio de uma determinada camada está conectada a entrada de todos os neurônios da camada seguinte, cada conexão possuindo seu próprio peso que é ajustado durante o treinamento. Assim, com sua complexidade adicional, modelos MLP não se limitam a classificações separáveis linearmente.

2.2.2.1 Função de ativação

A conexão entre os neurônios das diferentes camadas também passa por uma função de ativação, e diferente do perceptron simples, geralmente são utilizadas funções de ativação não lineares, como a função sigmoide e a unidade linear retificada (ReLU). Estas funções permitem um treinamento melhor para redes mais profundas, sendo a função ReLU a mais utilizada devido a suas vantagens nos resultados dos treinamentos.

$$ReLU(x) = max\{0, x\}$$

2.3 Métricas de avaliação

Métricas de avaliação são fundamentais para comparação e análise de modelos, e dentro da área de redes neurais algumas das métricas mais conhecidas são, precisão, acurácia, recall, medida F1 e Jaccard.

- **Precisão**. É a proporção de acertos positivos em relação ao número total de predições positivas.
- **Acurácia**. É a proporção de acertos positivos e negativos em relação ao total de predições.
- **Recall**. É a taxa de acertos positivos em relação ao total de instâncias que deveriam ser positivas, ou seja os acertos positivos e os falsos negativos.
- **Pontuação F1**. A pontuação pode ser vista como a média harmônica entre a precisão e o recall.
- **Jaccard**. O índice de Jaccard indica a similaridade dos conjuntos de predições e dos resultados esperados, dado pela divisão da interseção pela união desses conjuntos.

Em seguida temos as definições básicas delas (TAHA; HANBURY, 2015).

$$Precisao = \frac{AcertosPos}{AcertosPos + FalsosPos}$$

$$Acuracia = \frac{AcertosPos + AcertosNeg}{AcertosPos + AcertosNeg + FalsosPos + FalsosNeg}$$

$$Recall = \frac{AcertosPos}{AcertosPos + FalsosNeg}$$

$$F1Score = \frac{2 \cdot Precisao \cdot Recall}{Precisao + Recall}$$

$$Jaccard = \frac{|Predicoes \cap ResultadosEsperados|}{|Predicoes \cup ResultadosEsperados|}$$

2.4 Inteligência artificial explicável

A IA explicável, ou XAI, representa o conjunto de técnicas aplicadas em sistemas de inteligência artificial a fim de torná-los mais transparentes e entendíveis por humanos. Embora o termo seja relativamente novo (LENT; FISHER; MANCUSO, 2004), a

preocupação é de longa data, e apesar da área de inteligência artificial ter visto avanços consideráveis recentes e o interesse na explicabilidade dos modelos de IA continuar alto, o campo de IA explicável não tem acompanhado os desenvolvimentos, seja por ter ficado em segundo plano devido aos avanços de IA ou por falta de um foco na pesquisa (ADADI; BERRADA, 2018; SAEED; OMLIN, 2023).

Os métodos de IA Explicável não possuem uma fundamentação teórica geral, cada técnica explora uma aplicação de conceitos distintos, com focos específicos, sendo que se assemelham no seu objetivo que é a explicabilidade ou interpretabilidade de algoritmos de *machine learning* (ML).

2.5 Árvore de Decisão

A árvore de decisão é uma técnica de aprendizado de máquina que é frequentemente utilizada em problemas de classificação ou regressão, que é considerada interpretável e entendível por humanos.

Sua abordagem se baseia em uma estrutura de fluxograma com formato de árvore, onde cada nó representa uma tomada de decisão com base em um dos atributos da base de dados em que ela é aplicada. Seus nós internos podem ser representado como uma condicional baseada nesses atributos, e pode levar a um outro nó interno com outra condição ou a uma folha, com uma decisão da classificação ou predição, como visto na Figura 3.

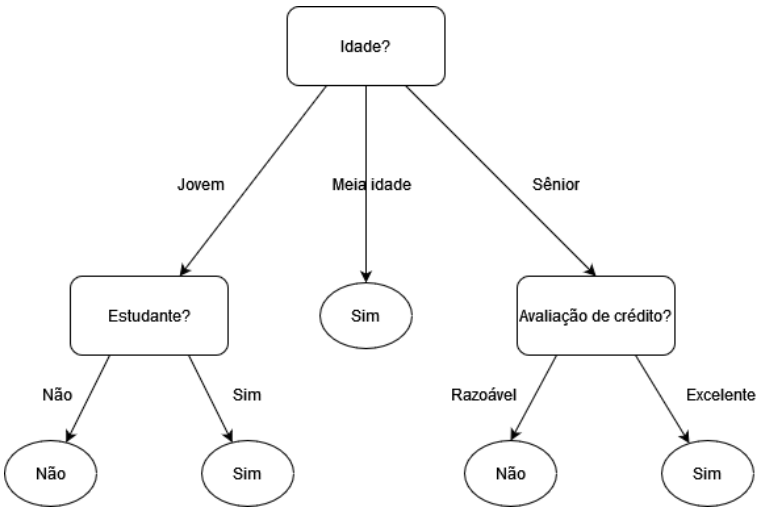

Figura 3 – Exemplo de árvore de decisão para predição de compra de um computador por clientes de uma loja de eletrônicos. Fonte: Adaptada de (HAN; KAMBER; PEI, 2012)

O processo de modelagem de uma árvore de decisão é chamado de indução de árvore de decisão, e pode ser feito atráves de algoritmos como o ID3, C4.5 e CART (HAN; KAMBER; PEI, 2012).

Tipicamente os algoritmos buscam construir as árvores de cima para baixo utilizando diferentes métodos, como a abordagem de algoritmo guloso, vista nos três algoritmos mencionados acima. Também, a depender do algoritmo, são utilizadas métricas variadas para auxiliar na formação das árvores, dentre elas Ganho de informação e impureza de Gini.

3 Revisão Bibliográfica

3.1 Inteligência artificial explicável

No artigo *Peeking inside the black-box: A survey on explainable artificial intelligence (xai)* (ADADI; BERRADA, 2018), os autores propõe a classificação de métodos de XAI em três critérios, a complexidade da interpretabilidade, o escopo da interpretabilidade, e o nível de dependência do modelo de ML utilizado. Sendo estas classes não mutualmente exclusivas e não exaustivas.

3.1.1 Métodos relacionados a complexidade

É esperado que quanto mais complexo o modelo de ML, mais difícil será interpretar e explicar suas decisões. Os métodos relacionados a complexidade propõem o uso de modelos de ML inerentemente mais simples e interpretáveis para a explicação dos métodos mais complexos.

Alguns modelos de ML propostos para a explicação dos modelos complexos são, Lista de Regras Bayesianas (LETHAM et al., 2015), modelo aditivo com interações emparelhadas (CARUANA et al., 2015) e o modelo linear super-esparso (USTUN; RUDIN, 2015). Uma desvantagem associada a estes métodos é a possível perda da acurácia das predições, já que a complexidade do algoritmo muitas vezes pode estar relacionada com sua capacidade de predição (BREIMAN, 2001).

3.1.2 Métodos relacionados ao escopo

Estes métodos são divididos em duas subclasses, interpretabilidade global e interpretabilidade local. Os métodos globais buscam interpretar o comportamento do modelo como um tudo, enquanto os locais buscam o entendimento de predições singulares.

Na prática, o entendimento global de um algoritmo é difícil de ser atingido, em especial conforme o número de parâmetros e complexidade dos modelos aumentam (ADADI; BERRADA, 2018), assim, os métodos de interpretabilidade global possuem utilidade limitada.

Por outro lado, a interpretabilidade local é mais promissora, com o desenvolvimento de ferramentas como o LIME, *Local Interpretable Model-Agnostic Explanation*, capazes de extrair uma explicação de predições de qualquer tipo de classificador ou regressor, utilizando uma aproximação por um modelo interpretável (RIBEIRO; SINGH; GUESTRIN, 2016).

3.1.3 Métodos relacionados ao modelo

Por fim, temos a classificação entre métodos por modelo do algoritmo de ML utilizado, também dividido em duas subclasses, os métodos que são específico de modelos, e os métodos agnósticos de modelos.

Devido a serem, por característica, limitados a modelos específicos, os métodos específicos em grande parte tem sido deixados de lado, em favor de métodos agnósticos, e assim em sua grande maioria os métodos de XAI desenvolvidos e estudados atualmente são os agnósticos (ADADI; BERRADA, 2018).

Desta forma, estes métodos agnósticos também podem ser divididos, e Adadi (ADADI; BERRADA, 2018) propõe quatro categorias: métodos de visualização; métodos de extração de conhecimento; métodos de influências; e, métodos baseados em exemplos.

Os métodos de visualização buscam gerar uma representação visual dos padrões escondidos nos algoritmos de ML.

Para os métodos de extração de conhecimento, seu objetivo é tentar extrair uma informação da estrutura interna da rede neural que possa ajudar na explicação, como alguma regra que possa estar codificada internamente.

Os modelos de influências tentam avaliar a importância de uma característica da base de dados fazendo alterações nos valores de entrada ou nos componentes internos e observando as mudanças geradas, como nas análises de sensibilidade (ZHANG; WALLACE, 2016).

E por fim, os modelos baseados em exemplos, que são técnicas para selecionar exemplos específicos dentro da base de dados e tentar explicar a decisão baseado neles.

Dentro da classificação de métodos de visualização temos o método de Surrogate Model (Substituição de modelo), que vamos explorar de maneira um pouco mais profunda. O método de Surrogate Model propõe utilizar um modelo alternativo mais simples no lugar de um modelo complexo, assim como nos métodos relacionados a complexidade, e com esses modelos mais simples gerar uma interpretação para a decisão do algoritmo original.

Isto pode ser feito com uma substituição por árvore de decisão, criando uma árvore de decisão equivalente ou transformando uma rede neural diretamente em uma árvore de decisão, como proposto por Aytekin (AYTEKIN, 2022), ou com um modelo local interpretável como o LIME.

O algoritmo LIME foi proposta em um artigo por Ribeiro (RIBEIRO; SINGH; GUESTRIN, 2016), e consiste numa técnica explicável de um modelo de classificação ou regressão, através da aproximação do modelo baseado em uma predição local (baseada em uma instância e seu redor), ou seja um modelo aproximado válido localmente para uma

predição específica. Sua saída ordena de maneira decrescente as características com seus respectivos pesos, representando quais delas foram as que mais influenciaram a decisão de predição para aquela instância.

4 Metodologia e Desenvolvimento

4.1 Desenvolvimento

O desenvolvimento feito neste trabalho buscou um entendimento progressivo tanto de redes neurais quanto de técnicas explicáveis.

Para atender a este objetivo, foi seguido uma linha de estudo e prática, começando com o desenvolvimento e aplicação de uma rede neural mais simples possível, o perceptron.

4.1.1 Perceptron

De maneira inicial, e por se tratar de um modelo simples, a aplicação prática de um perceptron foi desenvolvida na linguagem *python*, utilizando somente bibliotecas básicas para tratamento de números e plotagem gráfica.

Este perceptron consegue classificar qualquer conjunto de dados que seja possível separar linearmente, e em seu código estão definidos alguns casos de estudo básicos que podem ser escolhidos no início da execução, os operadores lógicos AND e OR, que são separáveis linearmente, e o operador XOR, onde o perceptron simples falha por não ser possíveis de separar suas saídas linearmente. Também está disponível para escolha uma base de dados linearmente separável.

Os pesos iniciais neste perceptron são escolhidos de maneira aleatória, possuindo valores entre 0 e 1, e taxa de aprendizado de 0.2.

A função de ativação escolhida foi a função de ativação com limiar, sendo o limiar 0, ou seja, acontece a ativação para quaisquer saídas positivas e estritamente maiores que 0.

O número de iterações de treinamento é definido ao início do programa, e é recomendado um valor acima de 20 iterações para as bases de dados presentes no código.

Ao final das iterações de treinamento, o programa fica disponível para testes de tuplas arbitrárias, sendo também gerado um gráfico (4) com os pontos da base de dados escolhida, sua classificação (representada por sua cor) e a reta encontrada que divide a base dados linearmente, conforme o objetivo do perceptron.

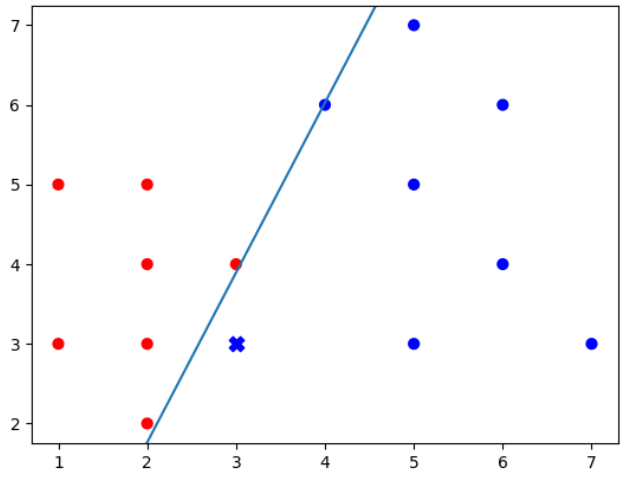

Figura 4 – Resultado do perceptron simples utilizando os dados predefinidos com 100 iterações e tupla de teste (3,3). Fonte: Autor

4.1.2 Multilayer Perceptron

Em sequência, foram elaborados modelos de classificadores multilayer perceptron para as bases de dados escolhidas para o trabalho.

Para isto, foi utilizada a biblioteca *sklearn* da linguagem *python*, que integra diversas ferramentas úteis e relevantes a machine learning, como algoritmos de classificação, regressão, ferramentas de divisão de dados para testes, além de métricas úteis que também são utilizadas neste trabalho.

O classificador MLP da biblioteca *sklearn* é configurável com diversos parâmetros, os mais notáveis são, *hidden_layer_sizes*, que define a quantidade camadas ocultas e quantas unidades de neurônios por camada, sendo uma tupla em que o valor representa a quantidade de neurônios, e sua presença representa a existência daquela camada, ou seja, a tupla (35,35) representa duas camadas com 35 neurônios em cada camada, e a tupla (30,20,10) representa 3 camadas ocultas, a primeira com 30 neurônios, a segunda com 20, e por fim a terceira com 10 neurônios. Também temos o parâmetro *activation*, para a seleção da função de ativação, sendo utilizado a ReLU para padronização dos experimentos, e *learning_rate_init*, para a taxa de aprendizado inicial.

Para padronização, a tupla que indica a quantidade de neurônios e camadas foi estabelecida utilizando 2 camadas escondidas para todas as bases de dados, para um total de 4 camadas com as camadas de entrada e saída, e a quantidade de neurônios em cada camada igual à quantidade de características da base de dados.

As bases de dados foram divididas aleatoriamente na proporção 70% para treinamento e 30% para testes de validação utilizando o *train_test_split* da *sklearn*, e para as

comparações foi utilizado o parâmetro *random_state*, que força uma divisão específica e possibilita uma comparação direta de resultados.

Os critérios de parada são os padrões do classificador, a execução é interrompida quando não há variação maior do que 0.0001 na métrica *training loss*, que é a soma dos erros para os casos de testes.

Por fim, os tempos de execução do modelo e as métricas como precisão, acurácia, recall, pontuação f1 e jaccard foram calculadas para serem utilizadas em comparações na seção de resultados. As bases utilizadas foram multiclasses, e por não ser binárias, para algumas das métricas foram necessárias o cálculo da média de cada rótulo, referente ao parâmetro *average* das métricas, e a abordagem utilizada nesses cálculos foi a *macro*, que computa a métrica para cada rótulo, e acha a média simples deles para as classes não alvo, desconsiderando possíveis desequilíbrios dos rótulos.

4.1.3 Árvore de decisão

Também foram elaborados os programas que geram as árvores de decisão (Figura 5) equivalente aos classificadores MLP, utilizando as mesmas bibliotecas, para cada uma das bases de dados, e o algoritmo de formação das árvores foi uma otimização do algoritmo CART. As mesmas métricas foram calculadas e foram utilizadas na seção de resultados.

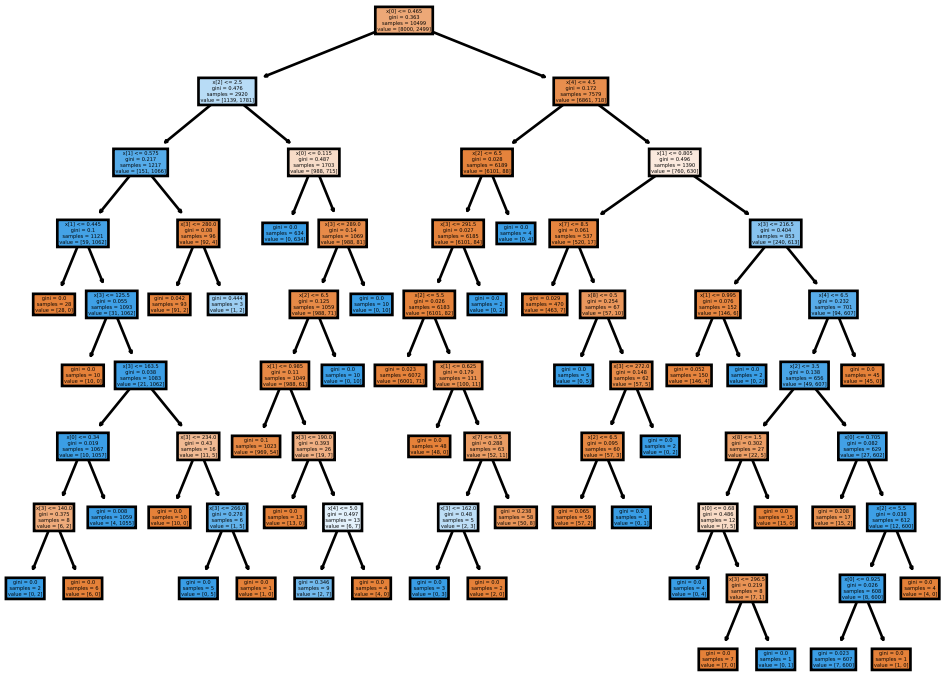

Figura 5 – Exemplo da Árvore de decisão gerada pelo código, aplicada a base de dados de Rotatividade de funcionários. Fonte: Autor

4.1.4 LIME

O algoritmo LIME foi implementado utilizando a biblioteca *lime* da linguagem *python*, que é uma biblioteca específica para a técnica, capaz de fazer a aproximação local e gerar gráficos com a explicação da predição.

4.1.5 Repositório

Os códigos implementados podem ser encontrados no repositório do github disponível na seguinte URL <https://github.com/eduardomarinho/neuralnetworkxai>.

4.2 Escolha das bases de dados

Com base em nossos objetivos de explorar técnicas que possam identificar possíveis discriminações e outros vieses em algoritmos de IA, foram escolhidas bases de dados para algoritmos de classificação com características compatíveis com cenários em que tais preocupações poderiam se manifestar. Essas bases são descritas na Tabela 1.

Nome	**Instâncias**	**Características**
Predict students' dropout and academic success	4424	36
HR Analytics for employee rentension	14999	10
Census Income dataset	48842	14

Tabela 1 – Tabela com informações básicas das bases de dados.

4.2.1 Base de dados - Estudantes

Predict students' dropout and academic success (REALINHO et al., 2021): base de dados referente a desistências e sucesso acadêmico de alunos do ensino superior. Possui características como, o curso do aluno, sua nacionalidade, gênero, status de unidades curriculares do primeiro e segundo semestres, idade, entre outros dados demográficos.

4.2.2 Base de dados - Rotatividade de funcionários

HR Analytics for employee rentension (RETENTION, 2019): base de dados da área de recursos humanos de uma empresa para a classificação de rotatividade de funcionários. Esta base possui indicadores relacionados aos funcionários, como o nível de satisfação, número de projetos, media de horas trabalhadas no mês, quantos anos trabalhando na empresa, entre outros.

4.2.3 Base de dados - Renda

Census Income dataset (KOHAVI, 1996): base com dados de censo demográfico para a classificação de renda anual. Possui como características informações demográficas como idade, nível de educação, ocupação, etnia, gênero, país de origem, entre outros.

4.3 Resultados experimentais

Os modelos de classificação foram treinados utilizando as bases de dados apresentadas, e com isso obtiveram-se métricas de performance, precisão, acurácia, recall, F1 e jaccard.

Como o resultado do treinamento pode variar dependendo de valores aleatórios durante a execução do código, para cada caso do classificador MLP, foi computado a média dos valores de cada uma das métricas referentes a 5 execuções.

Nos casos dos modelos de árvores de decisão, a variação dos valores das métricas em diferentes execuções é menor, mas para manter a padronização, foi computado a média dos valores de cada uma das métricas referentes a 5 execuções.

4.3.1 Métricas de qualidade

As tabelas 2, 3, 4 apresentam as médias das medidas de qualidade de cada uma das bases e modelos, com arredondamento para 2 casas decimais. As métricas também estão representadas de forma gráfica nas figuras 6, 7, 8.

Primeiramente temos os resultados para a base de dados de Estudantes:

Métrica	**Classificador MLP**	**Árvore de Decisão**
Tempo de Exec.	0.54s	0.04s
Acurácia	0.65	0.67
Precisão	0.57	0.61
Recall	0.53	0.61
F1 Score	0.49	0.61
Jaccard	0.38	0.46

Tabela 2 – Tabela com as métricas dos modelos para a base de dados de Estudantes.

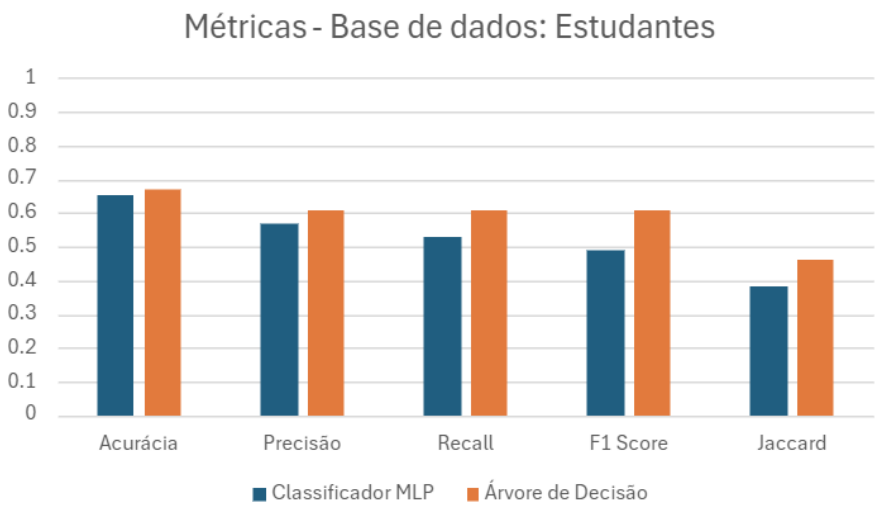

Figura 6 – Gráficos das métricas para a base de Estudantes. Fonte: Autor

Notam-se as proximidades dos resultados, com uma pequena vantagem nas métricas de classificação da árvore de decisão e a diferença em relação ao tempo de execução do treinamento modelo.

Em seguida, temos os resultados para a base de dados de Rotatividade de funcionários:

Métrica	**Classificador MLP**	**Árvore de Decisão**
Tempo de Exec.	1.90s	0.02s
Acurácia	0.90	0.97
Precisão	0.86	0.96
Recall	0.89	0.97
F1 Score	0.87	0.96
Jaccard	0.78	0.93

Tabela 3 – Tabela com as métricas dos modelos para a base de dados de Rotatividade de funcionários.

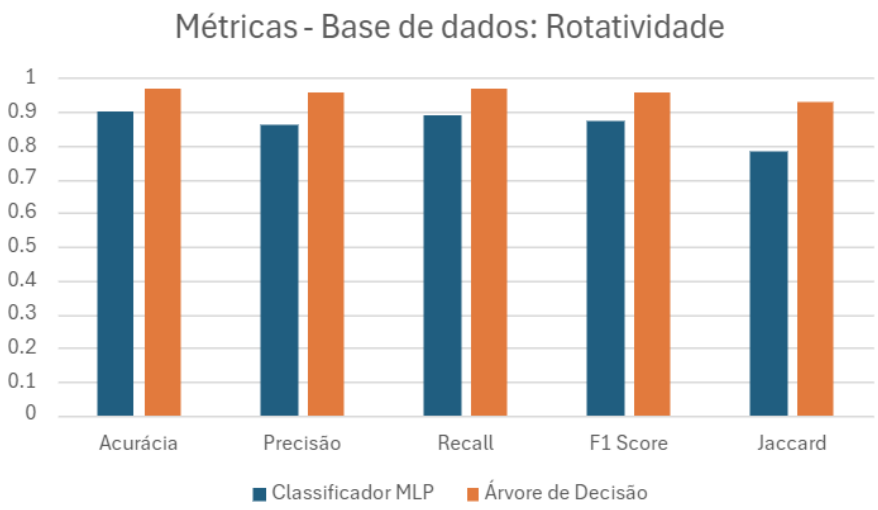

Figura 7 – Gráficos das métricas para a base de Rotatividade de funcionários. Fonte: Autor

Os resultados obtidos novamente indicam a vantagem vista anteriormente nas métricas da árvore de decisão.

E por fim, os resultados para a base de dados de Renda:

Métrica	**Classificador MLP**	**Árvore de Decisão**
Tempo de Exec.	2.54s	0.36s
Acurácia	0.51	0.44
Precisão	0.48	0.36
Recall	0.25	0.36
F1 Score	0.17	0.36
Jaccard	0.13	0.23

Tabela 4 – Tabela com as métricas dos modelos para a base de dados de Renda.

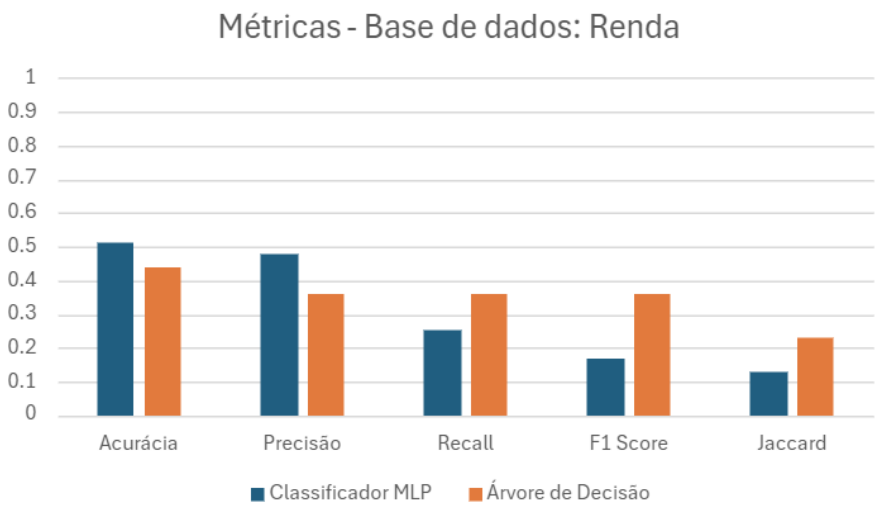

Figura 8 – Gráficos das métricas para a base de Renda. Fonte: Autor

Para esta base de dados, o classificador MLP teve resultados melhores para algumas das métricas, mas mantendo seu tempo de execução elevado em relação à árvore de decisão.

4.3.2 Técnica LIME

O algoritmo LIME é capaz de gerar um peso que reflete a importância de cada característica para a predição gerada pelo modelo de aprendizado. Neste trabalho foi utilizado o LIME para refletir a importância das características no modelo MLP e no modelo de árvore de decisão.

Nesta etapa, foram geradas os gráficos de importância de características, que correspondem às explicações com o LIME, para as instâncias das bases de dados. Esses gráficos apresentados nas Figuras 9 e 10 foram usados para a análise da diferença entre esses dois modelos de aprendizado.

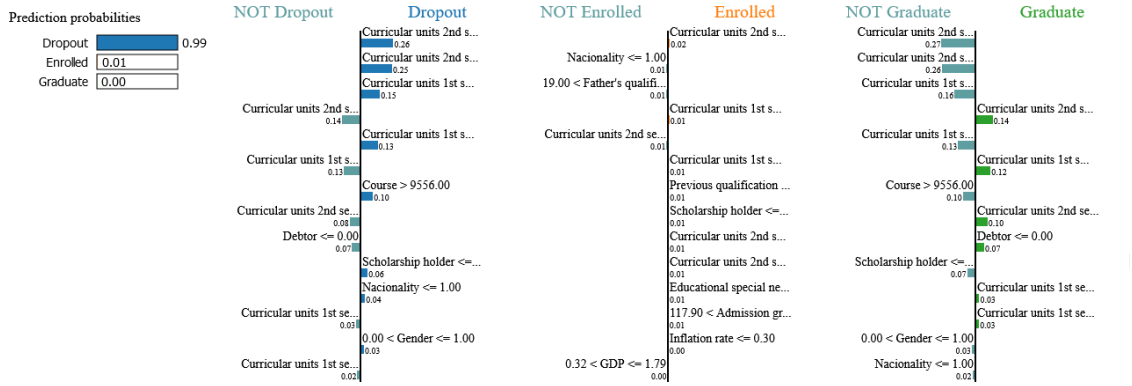

Figura 9 – Exemplo de explicação LIME da instância 533 da base de Estudantes para o modelo MLP. Fonte: Autor

Uma lista completa dos gráficos produzidos nos experimentos se encontra no Apêndice B.1.

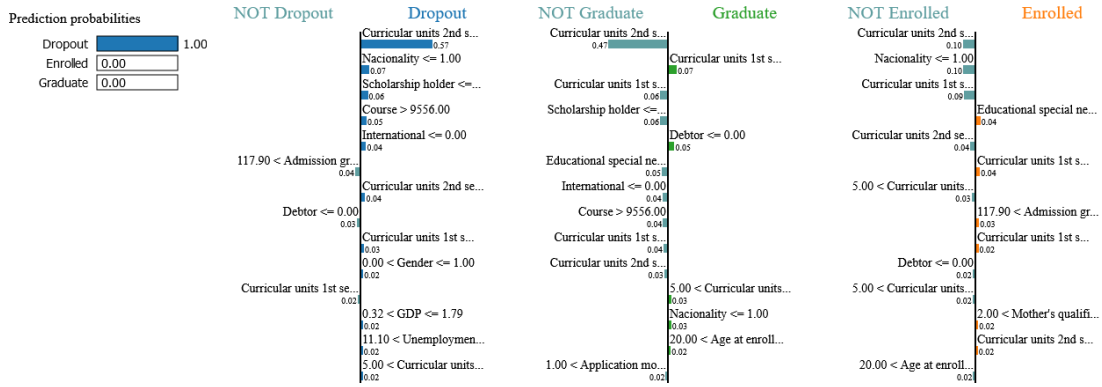

Figura 10 – Exemplo de explicação LIME da instância 533 da base de Estudantes para o modelo de Árvore de decisão. Fonte: Autor

4.4 Análise dos resultados

Os resultados encontrados nos mostra um cenário incomum, mas que se repete em nossas bases de dados, neles as métricas de qualidade (acurácia, precisão, recall, f1, jaccard) da árvore de decisão excedendo àquelas do classificador MLP em quase todos os casos estudados.

Breiman pontua que acurácia, de maneira geral, requer métodos de predição mais complexos (BREIMAN, 2001), porém esta observação não se mostrou em nossos experimentos. Isto não indica que a afirmação é falsa, mas nos mostra que existem casos em que modelos mais simples são capazes de substituir modelos complexos sem muitas perdas, como conclui Sarkar (SARKAR et al., 2016).

Destacam-se também as diferenças de custo computacional, representados pelo tempo de execução do treinamento dos modelos, nos indicando outra vantagem das árvores de decisão.

A abordagem de explicabilidade nos gráficos gerados por LIME nos mostrou quais características de uma instância específica das nossas bases de dados teve maior peso para a decisão da classificação que foi tomada.

Em uma comparação direta entra a explicação LIME para o modelo MLP e modelo de árvore de decisão, observa-se que os pesos e a ordem de importância das características podem diferir. Por exemplo, na comparação das características relevantes entre modelo MLP e Árvore de Decisão para a classificação da instância 533 da base de Estudantes, percebe-se que enquanto MLP lista como as 3 características mais relevantes para determinar a evasão do estudante as unidades curriculares 1 e 2 (ver Figura 9) , já para o Modelo de árvore de decisão, as características mais relevantes para determinar a evasão do estudante são a unidade curricular 2, a nacionalidade e a ausência de bolsa de estudo (ver Figura 10).

5 Conclusão

Recapitulando os objetivos deste trabalho, foi proposto um estudo de técnicas de IA Explicáveis, e para isso foi traçado uma linha de desenvolvimento para entendimento e aplicação de redes neurais e técnicas explicáveis. Durante o percurso desta linha, estudou-se e desenvolveu-se uma rede neural mais simples, denominada perceptron, entendendo seu funcionamento e a desenvolvendo e aplicando de maneira prática, com resultados de classificação linear satisfatórios e pertinentes.

Foram desenvolvidos algoritmos para redes neurais de múltiplas camadas, e com sua aplicação foi possível observar onde surgem as dificuldade com opacidade dos algoritmos *black-box*, com o aumento do número de camadas e neurônios, o entendimento de como uma decisão é tomada fica ofuscado ao cérebro humano.

As técnicas explicáveis utilizadas foram substituição de modelo por um mais simples e explicável, utilizando uma árvore de decisão, e LIME, um modelo de aproximação local de predição.

Com os resultados experimentais obtidos, concluímos que é possível extrair informações de decisão de um algoritmo *black-box* por meio da técnica LIME e também que a substituição por modelos mais simples é uma abordagem viável, e possivelmente preferível. As métricas calculadas para as bases de dados utilizadas mostram que a árvore de decisão na grande maioria dos casos foi capaz de se igualar ou superar a acurácia, precisão, recall, f1 e jaccard do modelo MLP.

E além dessas métricas, também foi visto pelo tempo de execução que a árvore de decisão possui custo computacional menor, em alguns casos com diferenças de ordens de magnitude, em relação a um modelo de MLP. Uma outra característica notável é sua estabilidade em relação a suas métricas entre diferentes execuções do algoritmo.

Nota-se, porém, que os modelos de MLP utilizados não foram otimizados, e também que estes resultados são particulares aos casos estudados. Conforme a literatura indica, para casos mais complexos e com quantidades maiores de dados, é provável que o MLP tenha resultados melhores.

Os resultados obtidos mostram as possibilidades das técnicas de IA Explicável, e como elas podem ser usadas para a explicabilidade e comparação dos modelos de aprendizado. No entanto, observa-se que a técnica LIME utilizada permite gerar uma explicabilidade local os para modelos de inteligência artificial *black-box*.

Contudo, os resultados obtidos neste trabalho são promissores e indicam a viabilidade dessas técnicas. Novos algoritmos explicáveis podem ser desenvolvidos e utilizados

para substituição, e dado um algoritmo suficientemente eficaz, os algoritmos *black-box* poderiam ser deixados de lado, em favor dos explicáveis.

5.1 Pesquisas futuras

A técnica de substituição de modelo se mostrou promissora, e uma pesquisa futura poderia explorar outros modelos na substituição, além da árvore de decisão. Outro foco seria explorar outras técnicas explicáveis.

A aplicação das técnicas em bases de dados maiores e mais complexas também pode ser uma abordagem de estudo.

Por fim, uma comparação entre uma árvore de decisão natural, e uma árvore de decisão criada através da transformação de um modelo de rede neural em uma árvore de decisão, como proposto por Aytekin (AYTEKIN, 2022), pode mostrar resultados interessantes, em especial nas comparações de métricas.

Referências

ADADI, A.; BERRADA, M. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access Volume: 6*, 2018. Citado 4 vezes nas páginas 10, 17, 19 e 20.

AYTEKIN, C. *Neural Networks are Decision Trees*. 2022. Citado 2 vezes nas páginas 20 e 32.

BREIMAN, L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, Institute of Mathematical Statistics, v. 16, n. 3, p. 199 – 231, 2001. Disponível em: <https://doi.org/10.1214/ss/1009213726>. Citado 2 vezes nas páginas 19 e 30.

CARUANA, R. et al. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2015. (KDD '15), p. 1721–1730. ISBN 9781450336642. Disponível em: <https://doi.org/10.1145/2783258.2788613>. Citado na página 19.

GUIDOTTI, R. et al. *A Survey Of Methods For Explaining Black Box Models*. 2018. Citado na página 10.

HAN, J.; KAMBER, M.; PEI, J. *Data mining concepts and techniques, third edition*. Waltham, Mass.: Morgan Kaufmann Publishers, 2012. ISBN 0123814790. Disponível em: <http://www.amazon.de/Data-Mining-Concepts-Techniques-Management/dp/0123814790/ref=tmm_hrd_title_0?ie=UTF8&qid=1366039033&sr=1-1>. Citado 2 vezes nas páginas 5 e 17.

KOHAVI, R. *Census Income*. 1996. UCI Machine Learning Repository. Acessado em Janeiro de 2024. Disponível em: <https://doi.org/10.24432/C5GP7S>. Citado na página 26.

LENT, M. van; FISHER, W.; MANCUSO, M. An explainable artificial intelligence system for small-unit tactical behavior. In: *Proceedings of the 16th Conference on Innovative Applications of Artifical Intelligence*. [S.l.]: AAAI Press, 2004. (IAAI'04), p. 900–907. ISBN 0262511835. Citado na página 16.

LETHAM, B. et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, Institute of Mathematical Statistics, v. 9, n. 3, set. 2015. ISSN 1932-6157. Disponível em: <http://dx.doi.org/10.1214/15-AOAS848>. Citado na página 19.

MCCULLOCH, W.; PITTS, W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 127–147, 1943. Citado na página 13.

REALINHO, V. et al. *Predict students' dropout and academic success*. 2021. UCI Machine Learning Repository. Acessado em Janeiro de 2024. Disponível em: <https://doi.org/10.24432/C5MC89>. Citado na página 25.

RETENTION, H. *HR Analytics for employee rentension*. 2019. Kaggle. Acessado em Janeiro de 2024. Disponível em: <https://www.kaggle.com/datasets/pankeshpatel/hrcommasep>. Citado na página 25.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. Citado 2 vezes nas páginas 19 e 20.

SAEED, W.; OMLIN, C. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems Volume 263*, 2023. Citado na página 17.

SARKAR, S. et al. Accuracy and interpretability trade-offs in machine learning applied to safer gambling. In: *CoCo@NIPS*. [s.n.], 2016. Disponível em: <https://api.semanticscholar.org/CorpusID:14941215>. Citado na página 30.

TAHA, A. A.; HANBURY, A. *Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool*. 2015. Disponível em: <https://doi.org/10.1186/s12880-015-0068-x>. Citado na página 16.

USTUN, B.; RUDIN, C. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, Springer Science and Business Media LLC, v. 102, n. 3, p. 349–391, nov. 2015. ISSN 1573-0565. Disponível em: <http://dx.doi.org/10.1007/s10994-015-5528-6>. Citado na página 19.

ZHANG, Y.; WALLACE, B. *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*. 2016. Citado na página 20.

APÊNDICE A – Tabelas

A.1 Tabelas - Classificadores

Métrica	**1ª**	**2ª**	**3ª**	**4ª**	**5ª**
Tempo de Exec.	0.52111s	0.63513s	0.33607s	0.43918s	0.75617s
Acurácia	0.69653	0.70256	0.64307	0.52033	0.6988
Precisão	0.57149	0.62467	0.49429	0.46779	0.69064
Recall	0.54226	0.55324	0.48148	0.39736	0.68098
F1 Score	0.50747	0.52887	0.44902	0.32374	0.66362
Jaccard	0.40599	0.42140	0.34128	0.24310	0.51268

Tabela 5 – Tabela com valores das métricas do Classificador MLP para a base de dados de Estudantes.

Métrica	**1ª**	**2ª**	**3ª**	**4ª**	**5ª**
Tempo de Exec.	0.03806s	0.03801s	0.03801s	0.04001s	0.03801s
Acurácia	0.66642	0.67922	0.67395	0.68148	0.67093
Precisão	0.60279	0.61119	0.61005	0.61561	0.60747
Recall	0.60605	0.61325	0.61142	0.61802	0.61063
F1 Score	0.60383	0.61209	0.61042	0.61651	0.60855
Jaccard	0.45574	0.46569	0.46268	0.47059	0.46016

Tabela 6 – Tabela com valores das métricas da Árvore de decisão para a base de dados de Estudantes.

Métrica	**1ª**	**2ª**	**3ª**	**4ª**	**5ª**
Tempo de Exec.	1.48033s	2.00245s	0.9626s	2.84364s	2.2045s
Acurácia	0.85356	0.91578	0.90778	0.91412	0.92644
Precisão	0.80146	0.88494	0.8697	0.87031	0.89455
Recall	0.87567	0.88221	0.88017	0.91278	0.90620
F1 Score	0.82287	0.88356	0.87474	0.88822	0.90016
Jaccard	0.70574	0.79682	0.7832	0.80312	0.8224

Tabela 7 – Tabela com valores das métricas do Classificador MLP para a base de dados de Rotatividade de funcionários.

Métrica	**1ª**	**2ª**	**3ª**	**4ª**	**5ª**
Tempo de Exec.	0.02501s	0.02501s	0.02501s	0.02501s	0.02501s
Acurácia	0.97289	0.97533	0.97311	0.973784	0.97333
Precisão	0.95826	0.96295	0.95868	0.95995	0.9591
Recall	0.96810	0.96971	0.96825	0.96868	0.96839
F1 Score	0.96305	0.96627	0.96334	0.96422	0.96363
Jaccard	0.92939	0.93529	0.92992	0.93152	0.93045

Tabela 8 – Tabela com valores das métricas da Árvore de decisão para a base de dados de Rotatividade de funcionários.

Métrica	**1ª**	**2ª**	**3ª**	**4ª**	**5ª**
Tempo de Exec.	2.33853s	1.96044s	3.38911s	3.29742s	1.70738s
Acurácia	0.50577	0.50365	0.50556	0.51095	0.51123
Precisão	0.29149	0.50365	0.51543	0.51095	0.56808
Recall	0.25685	0.25001	0.25345	0.25007	0.2505
F1 Score	0.18094	0.16748	0.1747	0.16908	0.17019
Jaccard	0.13323	0.12591	0.12977	0.12774	0.12832

Tabela 9 – Tabela com valores das métricas do Classificador MLP para a base de dados de Renda.

Métrica	**1ª**	**2ª**	**3ª**	**4ª**	**5ª**
Tempo de Exec.	0.34508s	0.36208s	0.37963s	0.37909s	0.33208s
Acurácia	0.44141	0.44708	0.44339	0.44537	0.44155
Precisão	0.36068	0.36388	0.36073	0.36332	0.36163
Recall	0.36177	0.36565	0.36145	0.36544	0.36304
F1 Score	0.361	0.36463	0.36088	0.36422	0.36208
Jaccard	0.23055	0.23333	0.23062	0.23303	0.23109

Tabela 10 – Tabela com valores das métricas da Árvore de decisão para a base de dados de Renda.

Métrica	**Classificador MLP**	**Árvore de Decisão**
Tempo de Exec.	0.5375s	0.03842s
Acurácia	0.652258	0.6744
Precisão	0.569776	0.609422
Recall	0.531064	0.611874
F1 Score	0.494544	0.61028
Jaccard	0.38489	0.462972

Tabela 11 – Tabela com médias dos valores dos modelos aplicados a base de dados de Estudantes.

Métrica	**Classificador MLP**	**Árvore de Decisão**
Tempo de Exec.	1.8987s	0.02501s
Acurácia	0.903536	0.9736888
Precisão	0.864192	0.959788
Recall	0.891406	0.968626
F1 Score	0.87391	0.964102
Jaccard	0.782256	0.931314

Tabela 12 – Tabela com médias dos valores dos modelos aplicados a base de dados de Rotatividade de funcionários.

Métrica	**Classificador MLP**	**Árvore de Decisão**
Tempo de Exec.	2.53858s	0.3596s
Acurácia	0.507432	0.44376
Precisão	0.47792	0.362048
Recall	0.252176	0.36347
F1 Score	0.172478	0.362562
Jaccard	0.128994	0.231724

Tabela 13 – Tabela com médias dos valores dos modelos aplicados a base de dados de Renda.

APÊNDICE B – Gráficos LIME

B.1 Figuras - Modelo LIME

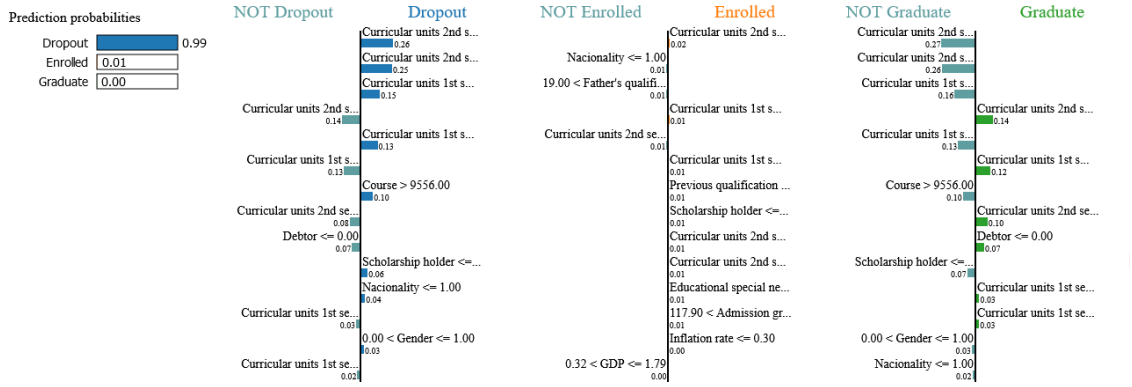

Figura 11 – Exemplo de explicação LIME da instância 533 da base de Estudantes para o modelo MLP. Fonte: Autor

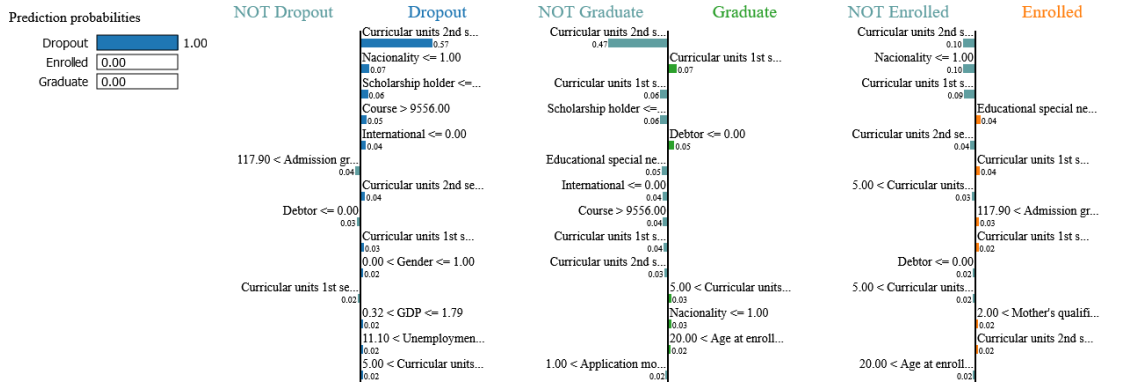

Figura 12 – Exemplo de explicação LIME da instância 533 da base de Estudantes para o modelo de Árvore de decisão. Fonte: Autor

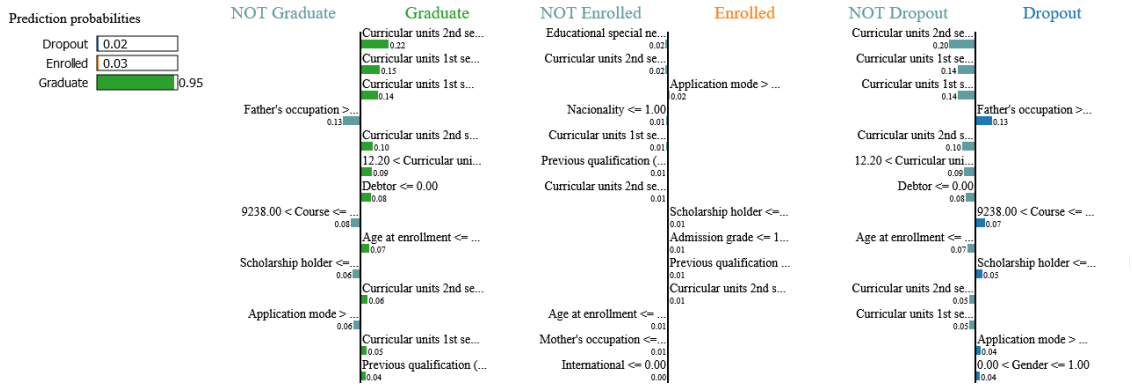

Figura 13 – Exemplo de explicação LIME da instância 2027 da base de Estudantes para o modelo MLP. Fonte: Autor

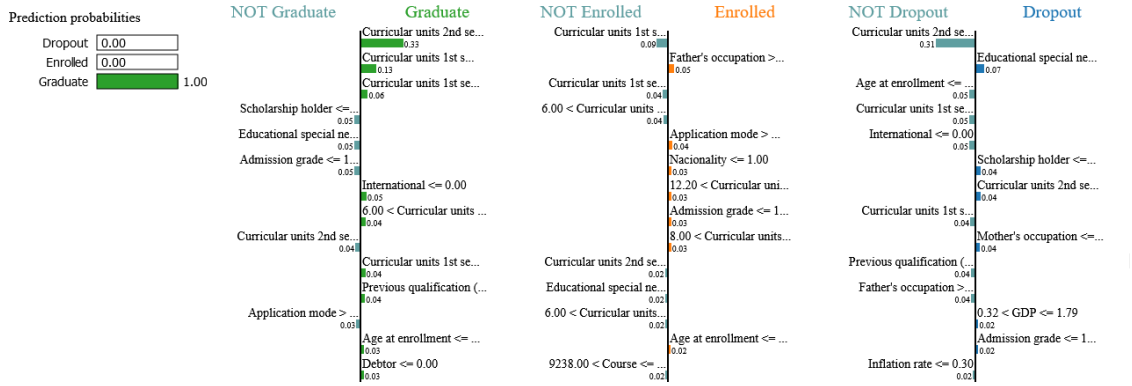

Figura 14 – Exemplo de explicação LIME da instância 2027 da base de Estudantes para o modelo de Árvore de decisão. Fonte: Autor

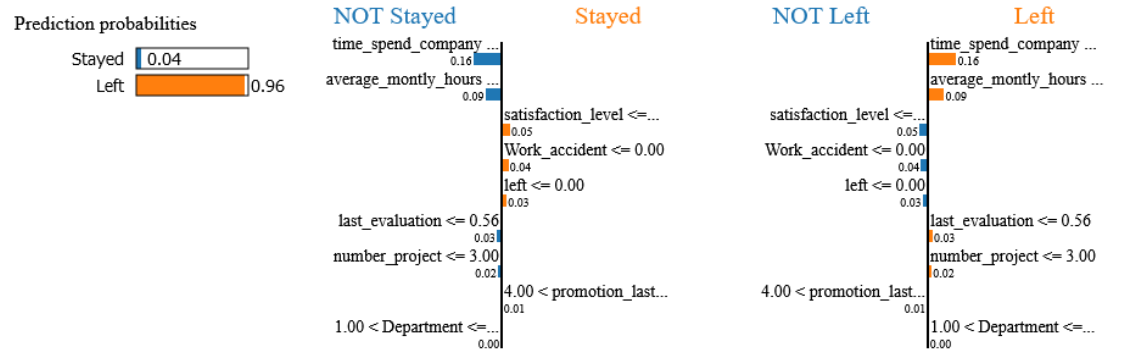

Figura 15 – Exemplo de explicação LIME da instância 1012 da base de Rotatividade de funcionários para o modelo MLP. Fonte: Autor

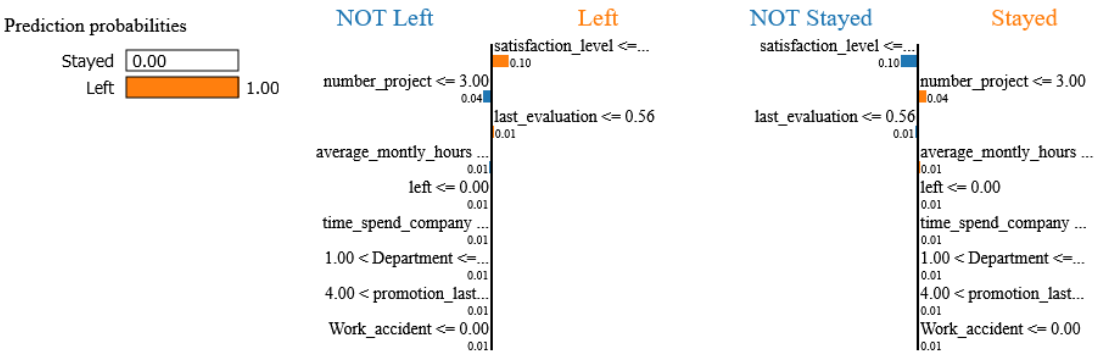

Figura 16 – Exemplo de explicação LIME da instância 1012 da base de Rotatividade de funcionários para o modelo de Árvore de decisão. Fonte: Autor

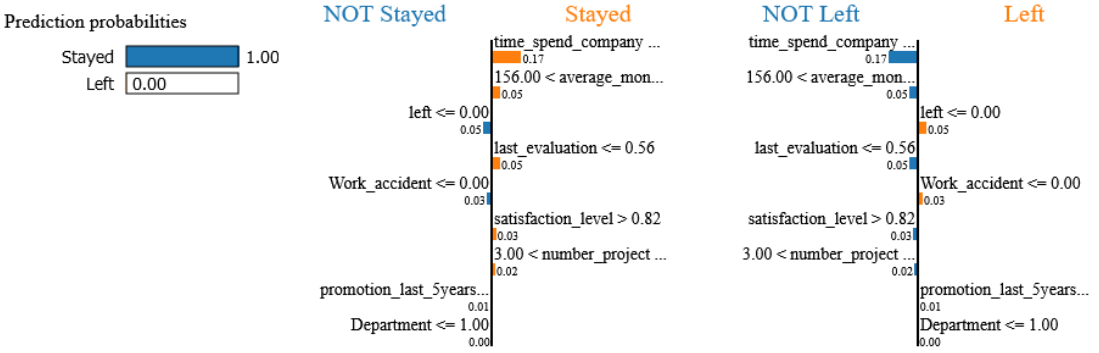

Figura 17 – Exemplo de explicação LIME da instância 3325 da base de Rotatividade de funcionários para o modelo MLP. Fonte: Autor

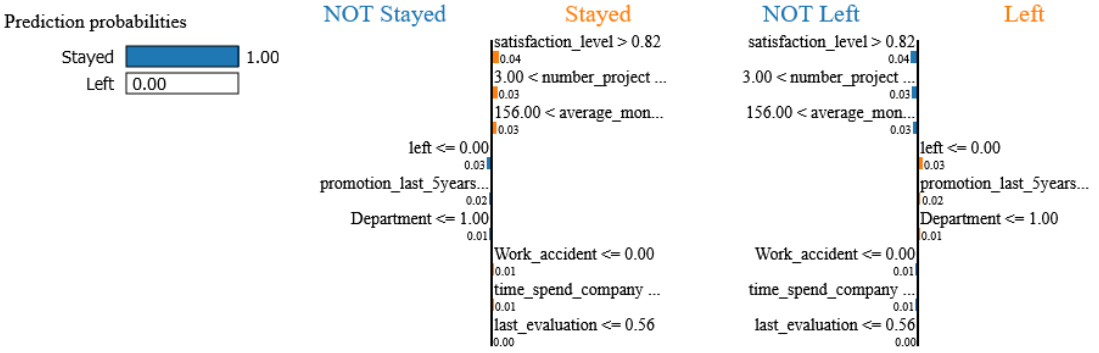

Figura 18 – Exemplo de explicação LIME da instância 3325 da base de Rotatividade de funcionários para o modelo de Árvore de decisão. Fonte: Autor

APÊNDICE C – Árvores de Decisão

C.1 Figuras - Árvores de Decisão

Os arquivo vetoriais para as árvore de decisão, assim como o arquivo em formato de texto estão disponíveis no repositório do github na URL <https://github.com/eduardomarinho/neuralnetworkxai>.

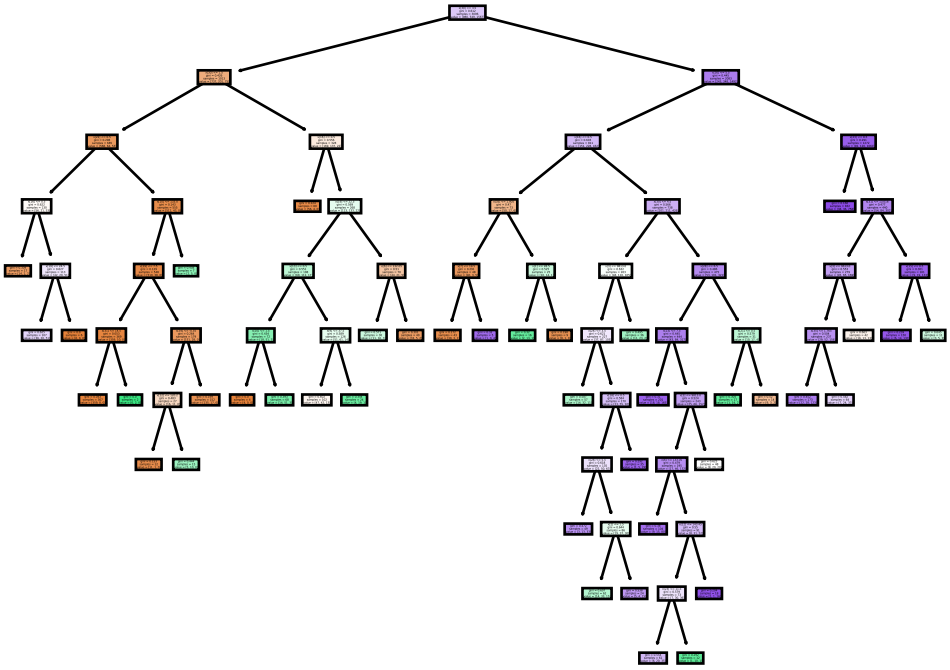

Figura 19 – Árvore de decisão aplicada a base de dados de Estudantes. Fonte: Autor

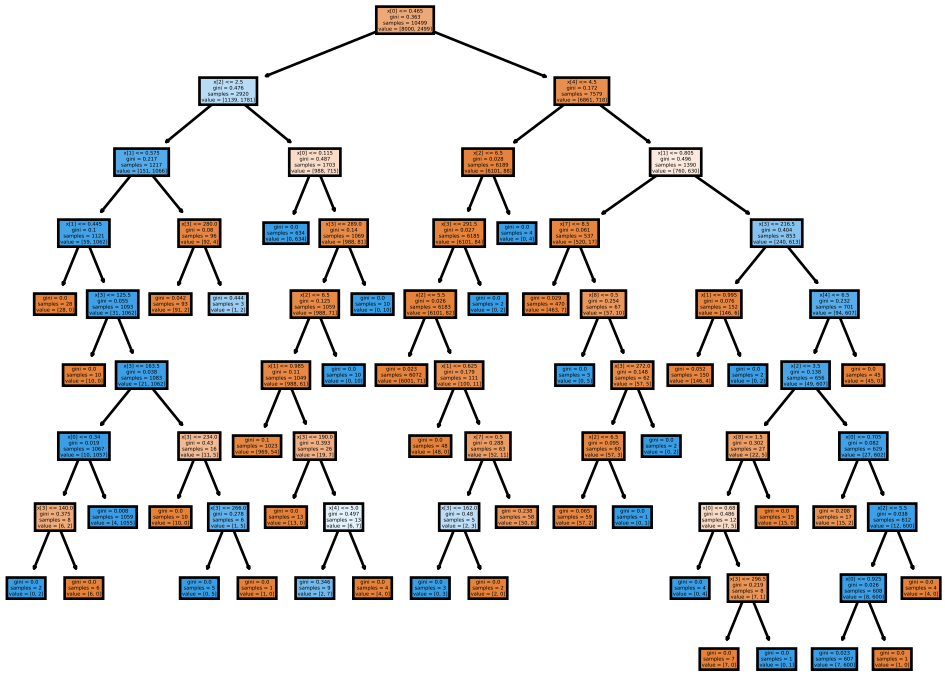

Figura 20 – Árvore de decisão aplicada a base de dados de Rotatividade de funcionários. Fonte: Autor

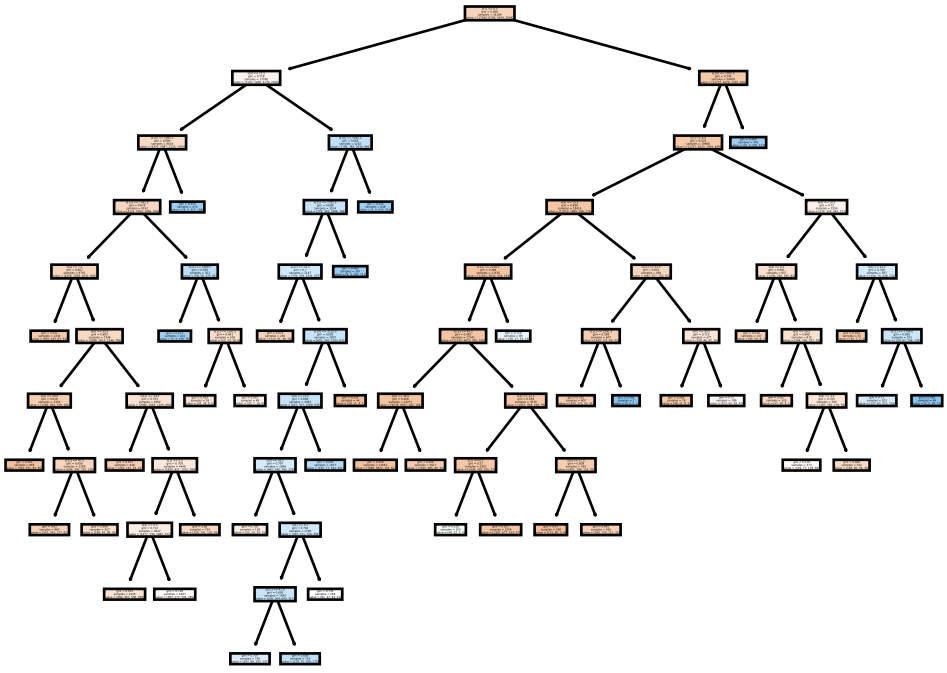

Figura 21 – Árvore de decisão aplicada a base de dados de Renda. Fonte: Autor