UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

VINICIUS GONÇALVES ARRUDA

Aplicação do Algoritmo LIME para explicar como classificadores black-box usam atributos na tomada de decisão

São Carlos - SP

2025

VINICIUS GONÇALVES ARRUDA

Aplicação do Algoritmo LIME para explicar como classificadores black-box usam atributos na tomada de decisão

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Universidade Federal de São Carlos, como requisito para a obtenção do título de Bacharel em Engenharia de Computação.

Orientação Profa. Dra. Marcela Xavier Ribeiro

São Carlos - SP

2025

Dedico este trabalho à minha família e aos meus amigos

Agradecimentos

Agradeço principalmente aos meus pais e aos meus avós, que acreditaram em mim até mesmo quando eu duvidei. Agradeço também aos meus amigos, que melhoraram imensamente todo o caminho.

Por fim, agradeço à minha orientadora, por toda a paciência durante o desenvolvimento deste trabalho. Que a próxima fase da vida seja sempre melhor que a anterior.

Resumo

Com a recente popularização de inteligências artificiais generativas, houve um aumento na preocupação com tornar compreensíveis quais os critérios determinantes utilizados pelos modelos para responder às questões propostas pelos usuários. Neste trabalho, procurou-se utilizar técnicas de IA Explicável (XAI) para esclarecer o funcionamento de modelos black-box, aqueles em que não está claro o caminho trilhado pelo algoritmo para encontrar a resposta oferecida. A preocupação acerca da explicabilidade de um modelo de IA se dá principalmente para garantir confiabilidade e monitorar possíveis falhas éticas e de direitos, como: preconceito, discriminação, invasão de privacidade, entre outros problemas que treinamentos podem causar. O modelo black-box escolhido foi o Multilayer Perceptron (MLP) e buscou-se aumentar sua explicabilidade aplicando o Algoritmo LIME. A aplicação visou compreender o peso designado aos atributos na decisão do MLP. Para isso, ajustou-se um algoritmo kNN ponderado com a atribuição dos pesos encontrados pelo ranking de importância dos atributos gerados pelo LIME ao ser aplicado no MLP. Por fim, foram analisadas medidas de desempenho como tempo de execução, precisão, acurácia, Recall, F1 Score e Jaccard para comparar o MLP com um classificador white-box, o kNN. Os resultados obtidos sugerem que há aumento de explicabilidade do MLP utilizando técnicas de XAI e que a média do desempenho do LIME melhora a expressividade da influência global dos atributos no modelo de classificação.

Palavras-chave: Inteligência Artificial; XAI; kNN; Modelos black-box; MLP;.

Abstract

With the recent popularization of generative artificial intelligence, there has been an increase in concern about making understandable the determining criteria used by models to answer questions proposed by users. In this work, we sought to use Explainable AI (XAI) techniques to clarify the functioning of black-box models, those in which the path taken by the algorithm to find the answer offered is not clear. The concern about the explainability of an AI model is mainly to ensure reliability and monitor possible ethical and rights failures, such as: prejudice, discrimination, invasion of privacy, among other problems that training can cause. The black-box model chosen was the Multilayer Perceptron (MLP) and we sought to increase its explainability by applying the LIME Algorithm. The application aimed to understand the weight assigned to the attributes in the MLP decision. To this end, a weighted kNN algorithm was adjusted with the assignment of the weights found by the importance ranking of the attributes generated by LIME when applied to the MLP. Finally, performance measures such as execution time, precision, accuracy, recall, F1 Score and Jaccard were analyzed to compare the MLP with a white-box classifier, kNN. The results obtained suggest that there is an increase in the explainability of the MLP using XAI techniques and that the average performance of LIME improves the expressiveness of the global influence of the attributes in the classification model.

Keywords: AI; XAI; kNN; black-box model; MLP.

Lista de ilustrações

Sumário

1 Introdução

1.1 Apresentação do tema

De acordo com (NORVIG, 2009) inteligência artificial (IA) se refere a uma máquina, que opera com conhecimento codificado em uma linguagem interna, cujo pensamento pode ser usado para escolher corretamente as ações a serem tomadas. Recentemente, a área teve crescimento em sua evolução devido ao aumento de investimentos consequente da popularização de IAs generativas como o Chat-GPT, o Deepseek e o Copilot. O campo combina principalmente teorias de computação e de estatística para criar modelos que possam aprender padrões e, com isso, tomar as melhores decisões possíveis para o contexto.

Uma das áreas de maior destaque abrangidas pelo estudo de IAs é a de Aprendizado de Máquina ou Machine Learning (ML), que procura criar algoritmos capazes de fazer predições partindo de bases de dados. É possível citar, por exemplo, tradução automática e processamento de imagens como aplicações comuns de Aprendizado de Máquina (WU; FENG, 2018).

No campo de ML existem algumas categorias de treinamento, nele destacam-se o aprendizado supervisionado, em que o modelo tem acesso às respostas esperadas de classificação diretamente da base de dados em que ele tem seu treinamento, e o não supervisionado, onde o modelo tenta identificar padrões sem as respostas fornecidas pela base (MORALES; ESCALANTE, 2022)

Entre os classificadores mais comuns de ML temos redes neurais, algoritmos Random Forest, k-Nearest Neighbors (kNN) e árvores de decisão, cada um tendo características distintas, oferecendo eficácias e interpretabilidades específicas de seus escopos.

Porém, como citado em (GUIDOTTI, 2018), existe uma preocupação crescente com questões de viés, sociais e éticas devido à forma desconhecida com que uma parte considerável desses classificadores chega ao seu resultado final.

Portanto, há uma necessidade cada vez maior de usar modelos interpretáveis cujas previsões possam ser compreendidas pelos desenvolvedores e por outras partes interessadas como reguladores e usuários finais. Nos casos em que os modelos ainda não são interpretáveis, denominados modelos black-box, muitas vezes é necessário criar métodos para explicar seu comportamento e suas previsões. A área de Inteligência Artificial Explicável (XAI) busca atingir esse objetivo.

Uma das maneiras de aumentar a explicabilidade de modelos black-box é a utilização de técnicas de comparações com modelos já conhecidos e, portanto, compreensíveis. Outra

maneira é a análise dos efeitos de modificações de parâmetros e variáveis nos modelos black-box e a elaboração de modelos aproximados a partir dos resultados de predição (ADADI; BERRADA, 2018).

1.2 Justificativa

Os chamados modelos de caixa preta (black-box) de inteligência artificial são aqueles em que não está claro qual foi o processo utilizado para a tomada de decisão. Exemplos de modelos black-box são redes neurais artificiais, redes neurais profundas, algoritmos de boosting, máquinas de vetor de suporte e florestas aleatórias (RIBEIRO; SINGH; GUESTRIN, 2016).

Modelos assim permitem apreensões éticas e questões de confiabilidade. Destacam-se a possibilidade de viés, preconceitos, questões de privacidade, violações de direitos, entre outros problemas causados involuntariamente nos resultados. É possível citar, por exemplo, problemas graves de discriminação racial realizados por estes algoritmos (FREITAS, 2018).

Portanto, neste trabalho, com o foco em explicabilidade e interpretabilidade, buscou-se atacar a falta de clareza dos algoritmos black-box para viabilizar uma maior adoção desses modelos de aprendizado de máquina em mais aplicações de forma coesa, considerando ainda que os desafios da área de XAI só têm crescido, visto que a complexidade dos modelos tem aumentado (HOFFMAN et al., 2019).

Também se visou aumentar a confiabilidade de modelos de IA, para que seja possível depurar e entender se os critérios adotados para a tomada de decisão são pertinentes, implicando em ponderar os atributos mais relevantes da base de dados (LUNDBERG; LEE, 2017). Possibilitando assim a validação dessas decisões.

1.3 Objetivos

1.3.1 Objetivo geral

Aplicar o Algoritmo LIME para aumentar a explicabilidade do classificador black-box Multilayer Perceptron (MLP). Com isso, busca-se encontrar o grau de contribuição de cada atributo na decisão gerada pelo MLP.

1.3.2 Objetivos específicos

Para verificar o comportamento do MLP usando um modelo mais explicável, foram escolhidos um kNN Simples e um kNN Ponderado, que apresenta graus de importância (pesos) gerados pelo LIME aplicado ao MLP. Comparou-se os resultados do MLP original com os de ambos os algoritmos kNN.

Ainda na aplicação do Algoritmo LIME ao MLP, deseja-se realizar testes com um número considerável de instâncias para verificar a contribuição das instâncias na decisão dada pelo modelo.

Os objetivos específicos deste trabalho são:

- Fazer o levantamento das principais ferramentas de XAI e abordar seu funcionamento.
- Investigar os benefícios da aplicação de técnicas de explicabilidade em modelos de IA.
- Aplicar o MLP em uma base representativa de dados reais.
- Aplicar o algoritmo LIME no modelo MLP gerado no objetivo anterior.
- Rodar o kNN simples sobre a base de dados escolhida.
- Rodar o kNN ponderado com os pesos de atributos atribuídos pelo LIME.
- Realizar testes com um número representativo de instâncias para verificar a interferência das instâncias no resultado do LIME.
- Discutir os resultados obtidos com as aplicações.

2 Fundamentação Teórica

2.1 Mineração de Dados

Com o aumento da capacidade de armazenamento de dados das últimas décadas, cresceu também a disponibilidade de quantidades enormes de dados para serem tratados por grandes empresas (HECKERMAN, 1997). Neste contexto, foi criada e potencializada a área de Mineração de Dados, que busca transformar tal volume de dados em informação útil e conhecimento (MENA, 1999).

O campo de mineração de dados abrange áreas como IA, estatística, processamento de imagens e banco de dados e seu objetivo geralmente é verificação de hipóteses ou obtenção de novos padrões sobre os dados (BURL; FOWLKES; RODEN, 1999). As técnicas de mineração podem ser de predição ou descrição, em que na primeira são construídos modelos para determinar o comportamento de dados futuros, enquanto na segunda os algoritmos tentam revelar padrões e propriedades nos dados já obtidos.

É importante para a aplicação de mineração de dados definir anteriormente fatores como: atributos relevantes para análise, medidas de interesse e também qual a técnica de mineração adequada.

A definição de atributos relevantes permite enxugar a grande quantidade de padrões desinteressantes e sem significado que podem resultar do processo de mineração.

A importância das medidas de interesse se dá para mensurar a qualidade da mineração a partir de parâmetros como precisão e acurácia, revocação, verdadeiro positivo, verdadeiro negativo, f-measure, curva ROC, entre outras.

As principais técnicas trabalhadas pela mineração de dados para extrair conhecimento de bases de dados são classificação, agrupamento e associação. Nos classificadores, um treinamento é realizado para a criação de modelos de aprendizado que permitem atribuir instâncias de dados analisados a classes definidas. No agrupamento, uma coleção de instâncias de dados é agrupada de acordo com a similaridade de seu conteúdo. Enquanto na associação, busca-se encontrar padrões de co-ocorrência entre itens de dados em uma base de dados (AGRAWAL; IMIELINSKI; SWAMI, 1993).

2.2 Classificadores

Classificadores são modelos construídos para atribuir uma classe a instâncias descritas por um conjunto de atributos (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997).

Para tal, é preciso realizar o aprendizado do modelo a partir de dados rotulados da base de dados.

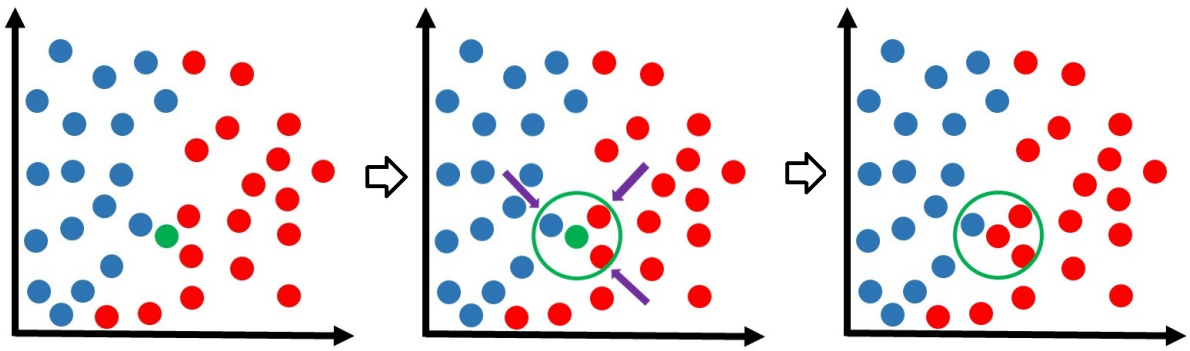
2.2.1 Classificador kNN (K-Nearest-Neighbor Classifier)

O classificador k-Nearest-Neighbor (kNN), ou k-vizinhos mais próximos, é um classificador baseado na aprendizagem por comparação: basicamente classifica uma dada instância de teste comparando-a com instâncias consolidadas no treinamento que são consideradas parecidas, vizinhas. Considerando que no treinamento as instâncias são descritas por um número n de atributos, cada instância representa um ponto num espaço n-dimensional. Desta forma, todas as instâncias de treino são armazenadas num espaço n-dimensional (LI; YU; LU, 2003).

Dada uma nova instância cuja classificação é desconhecida, o kNN retorna a classe dominante entre as k instâncias vizinhas mais próximas (LODWICH; FRASCH; BREUEL, 2007).

Na figura 1 há um exemplo simplificado do funcionamento do kNN para k=3, ou seja, 3 vizinhos, em que a instância verde é a amostra que se deseja classificar e as instâncias azuis e vermelhas são instâncias já classificadas da base de dados. É possível observar que, dos 3 vizinhos mais próximos, dois são classificados como vermelhos e um classificado como azul; portanto, nossa amostra de interesse é classificada como vermelha.

Figura 1 – Modelo de funcionamento do kNN para k=3



Fonte: Tech (2024)

A "proximidade" é definida em termos de uma métrica de distância, como a distância Euclidiana. À medida que o número de instâncias de treino se aproxima do infinito e k = 1, a taxa de erro do classificador KNN não é pior do que o dobro da taxa de erro mínimo teórico do classificador Bayes. Se k também se aproximar do infinito, a taxa de erro aproxima-se da taxa de erro de Bayes (WANG; HAMZA; SONG, 2017).

Os classificadores kNN utilizam comparações baseadas na distância e atribuem o mesmo peso a cada atributo. Portanto, podem apresentar precisão baixa quando existem

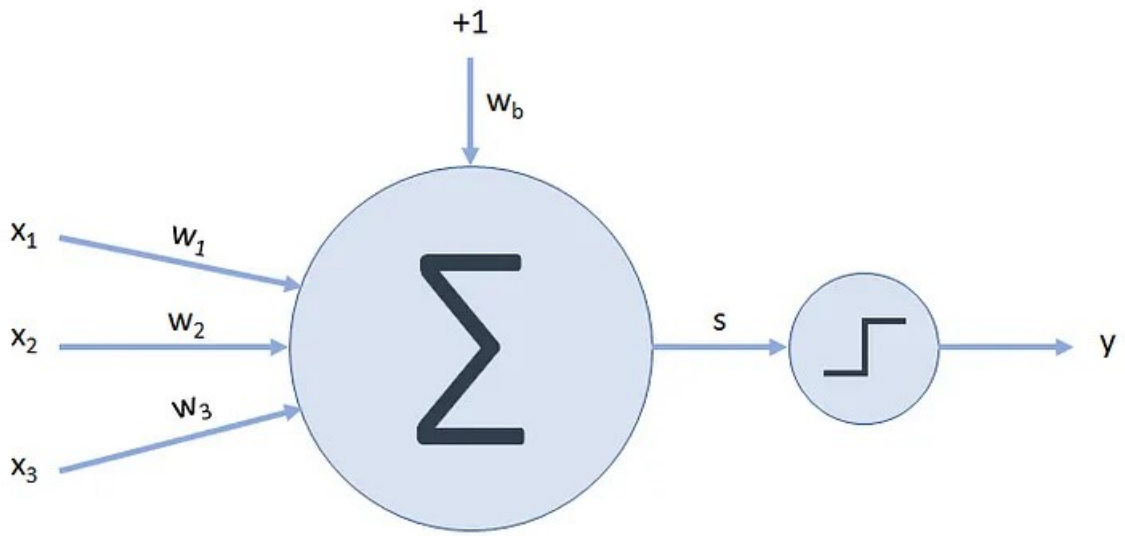
atributos ruidosos ou irrelevantes. O método, no entanto, foi modificado para incorporar a ponderação de atributos e a eliminação de instâncias de dados com ruído. O algoritmo, por calcular distâncias para cada classificação realizada, tende a perder desempenho conforme a base de dados cresce.

2.2.2 Redes neurais artificiais

Redes neurais artificiais são modelos computacionais inspirados na estrutura de um neurônio humano que tentam reproduzir a forma como um cérebro real toma decisões (MCCULLOCH, 1943).

Perceptron é o modelo mais simples de rede neural. É um modelo de rede neural proposto no final da década de 1950 que faz classificações de forma binária, funcionando como um modelo matemático de um neurônio. A figura 2 mostra um exemplo de Perceptron. Neste modelo de neurônio, as entradas são sinais capturados pelo neurônio (x1, x2 e x3) e, após serem modulados pela sinapse, resultam em uma saída (y), que pode ser ativada ou não. Portanto, a saída é definida exclusivamente pela combinação de sinais na entrada e pesos sinápticos.

Figura 2 – Modelo de funcionamento do Perceptron.



Fonte: Matsumura (2020)

Na figura 2 há um somador que pondera as entradas por meio de pesos (w1, w2 e w3) e aceita uma entrada adicional constante, ponderada por um peso wb. O resultado dessa soma ponderada alimenta a função de ativação que mapeia s para o resultado y. Os parâmetros do perceptron são seus pesos, que representam as sinapses em um neurônio. Ao alterar estes pesos, altera-se a função representada pelo perceptron.

O Multilayer Perceptron (MLP) é uma evolução do perceptron, funciona em uma organização de camadas, possuindo uma camada de entrada, camadas intermediárias e uma camada de saída.

O sistema do MLP funciona de forma que a saída de cada neurônio de uma camada específica fica conectada com a entrada de todos os neurônios da camada seguinte, onde cada conexão possui seu próprio peso que é regulado durante o aprendizado.

A função de ativação no MLP geralmente são funções de ativação não lineares, como a função sigmoide e a unidade linear retificada (ReLU).

Já no Deep Learning há uma quantidade grande de camadas intermediárias, o que aumenta a complexidade desses modelos e permite a extração de relações entre os atributos e reconhecimento de padrões que os modelos mais simples não conseguem obter, aumentando o domínio de aplicação destes modelos.

2.3 Inteligência artificial explicável (XAI)

IA Explicável é o campo que abrange o conjunto de ferramentas e técnicas utilizadas para ajudar no processo de compreensão dos resultados obtidos através de uma Inteligência Artificial.

Os métodos de IA Explicável se dividem entre aqueles que buscam interpretabilidade global e aqueles que apresentam interpretabilidade local. Enquanto métodos globais buscam interpretar o comportamento do modelo de forma ampla, os locais buscam o entendimento de predições em torno de um recorte ou até de instâncias singulares.

Os métodos de interpretabilidade global funcionam de maneira limitada conforme o número de parâmetros e a complexidade dos modelos aumentam, pois o entendimento geral de um algoritmo é difícil de ser atingido (ADADI; BERRADA, 2018).

Estes métodos também são divididos entre outras duas categorias: os métodos agnósticos, que servem para qualquer tipo de modelo (black-box e white-box), e os métodos que são específicos de cada modelo.

Os métodos agnósticos tendem a trabalhar com interpretabilidade local, fornecendo uma explicação para predições de qualquer tipo de classificador, e por isso têm sido mais estudados do que os métodos globais que tendem a ter a aplicabilidade limitada a um número pequeno de modelos de classificação (BAEHRENS et al., 2010).

Os autores de (ADADI; BERRADA, 2018) ainda criaram categorias para os métodos de XAI:

- Métodos de visualização: buscam criar uma representação visual dos padrões não revelados nos modelos de ML.

- Métodos de extração de conhecimento: tenta extrair informações que possam ajudar na explicação de dentro da estrutura interna da rede neural, como uma regra codificada internamente.

- Métodos de influências: avaliam a importância de um atributo da base de dados fazendo mudanças nos valores de entrada ou nos componentes internos e verificando as mudanças geradas.

- Métodos baseados em exemplos: selecionam instâncias específicas dentro da base de dados e buscam explicar a decisão baseado nelas.

- Surrogate Model (Substituição de modelo): utilizam modelos alternativos mais simples no lugar de um modelo complexo, em que os modelos mais simples possam criar uma interpretação para a decisão do algoritmo original.

Entre as formas de *Surrogate Model* estão transformar uma rede neural diretamente em uma árvore de decisão (AYTEKIN, 2022) e criar um modelo local interpretável como o LIME, este último sendo o método de explicabilidade escolhido para ser usado neste trabalho por ranquear atributos de acordo com sua importância no modelo a ser explicado.

2.3.1 Algoritmo LIME (*Local Interpretable Model Agnostic Explanations*)

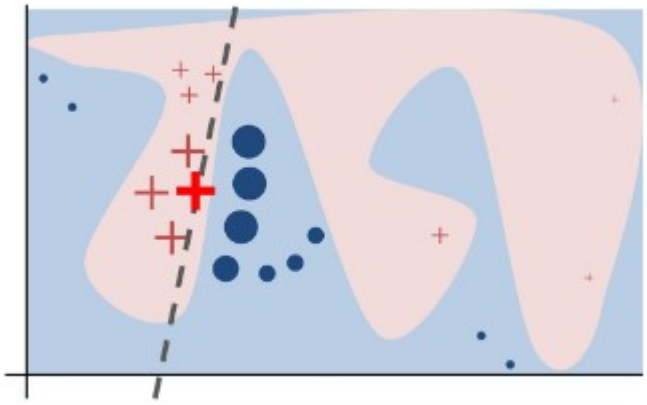
O algoritmo *Local Interpretable Model Agnostic Explanations* (LIME) visa aproximar qualquer classificador black-box de um modelo local e interpretável, aprendendo um modelo de previsão localmente em torno de cada amostra (RIBEIRO; SINGH; GUESTRIN, 2016).

O funcionamento do algoritmo LIME ocorre de forma a fornecer valores variados de dados para o modelo black-box e testar o que ocorre com tais predições, para então, no passo seguinte, gerar um novo conjunto de dados que consiste em amostras "perturbadas"e, assim, obter as correspondentes revisões para elas do modelo black-box.

Em sequência, o método pondera os resultados desses novos dados como uma função de sua proximidade com os dados originais, e, em uma última análise, o método gera o modelo substituto com base no conjunto de dados com perturbações usando os pesos de cada amostra (MOLNAR, 2020).

No exemplo da Figura 3 busca-se definir se a amostra é cruz vermelha ou ponto azul. A região com o fundo pintado em vermelho ou azul são as chamadas fronteiras de decisão. As amostras já classificadas são as geradas pela perturbação; quanto mais próxima geometricamente da amostra de interesse, mais importante para a decisão e maior ela será representada na imagem. O tracejado preto representa o modelo substituto gerado pelo LIME, um modelo linear utilizado para criar a explicação.

Figura 3 – Modelo visual de funcionamento do LIME.



Fonte: (RIBEIRO; SINGH; GUESTRIN, 2016)

Abaixo estão listadas as etapas do processo para treinamento de modelos substitutos locais usando o LIME:

a) Selecionar uma amostra de interesse para a qual deseja-se ter a explicação de seu rótulo pelo modelo black-box;

b) Gerar a partir de perturbações na instância de interesse, um novo conjunto de dados e utilizar o modelo caixa preta para esse novo conjunto, obtendo os rótulos do novo conjunto de dados;

c) Gerar pesos para as novas amostras geradas conforme a sua proximidade com a amostra de interesse;

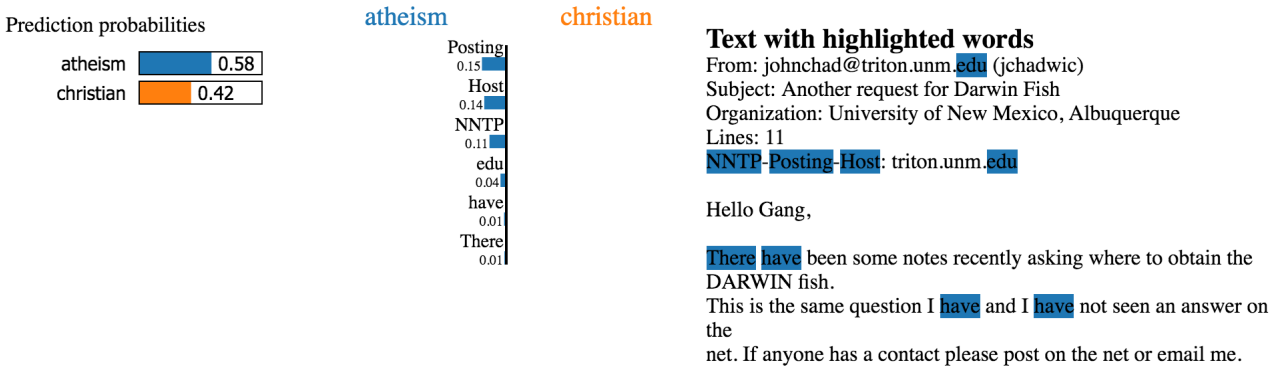
d) Fazer o treinamento de um modelo ponderado/linear e interpretável no conjunto de dados com os pesos gerados no passo anterior;

e) Gerar a explicação resultante da previsão de interpretação do modelo local.

O LIME usa, para definir a vizinhança significativa em torno de um ponto, o kernel de suavização exponencial. O kernel de suavização é uma função que recebe dois exemplos de dados e devolve uma medida de proximidade. O que determina o quão grande é uma vizinhança é a largura do kernel; quando a mesma é pequena, significa que o objeto de interesse deve estar muito próximo para influenciar o modelo local; quando é muito grande, significa que os objetos de interesse mais distantes também influenciam no modelo.

Um exemplo de simples entendimento do funcionamento do LIME fornecido por (RIBEIRO; SINGH; GUESTRIN, 2016) aborda a classificação da religião abordada em um texto. Na Figura 4, palavras negativas (azuis) indicam ateísmo, enquanto palavras positivas (laranja) indicam cristianismo.

Figura 4 – Exemplo de funcionamento do LIME com textos.



Fonte: (RIBEIRO; SINGH; GUESTRIN, 2016)

A maneira de interpretar os pesos é aplicando-os às probabilidades de previsão. Por exemplo, se removermos as palavras Host e NNTP do texto, esperamos que o classificador preveja o ateísmo com probabilidade de 0.58 - 0.14 - 0.1 = 0.31.

2.3.2 SHAP (*Shapley Additive Explanations*)

Shapley Additive Explanations, ou SHAP, é um framework com ferramentas para calcular os valores de Shapley, um conceito utilizado na teoria dos jogos cooperativos para determinar a relação entre a contribuição de cada jogador com a pontuação que cada um deve alcançar, sendo introduzido por Lloyd Shapley (LUNDBERG; LEE, 2017).

O SHAP apresenta diferentes módulos focados em várias abordagens de modelos. É possível citar Linear SHAP, Deep SHAP e Kernel SHAP, por exemplo. Baseado em (MOLNAR, 2020) sabe-se que para um número N de atributos, a quantidade de combinações possíveis é de 2^N. Portanto em um exemplo com N=3 temos o cálculo dos valores de Shapley definido por:

$$\begin{aligned}\phi = &\omega_1 \left(f(x_A, -, -) - f(-, -, -)\right) \\ &+ \omega_2 \left(f(x_A, x_B, -) - f(-, x_B, -)\right) \\ &+ \omega_3 \left(f(x_A, -, x_C) - f(-, -, x_C)\right) \\ &+ \omega_4 \left(f(x_A, x_B, x_C) - f(-, x_B, x_C)\right)\end{aligned}$$

Equação 1.

Onde o traço (-) dentro dos parâmetros de f() denota a ausência de um atributo nos cálculos que serão explicados a frente. Por exemplo, $f(x_A, -, x_C)$ é um exemplo com a ausência do atributo B, e $f(-, -, -)$ é um exemplo com ausência dos três atributos. Os pesos ω são definidos por:

$$\omega = -\frac{(|S| - 1)!(3 - |S|)!}{3!}$$

Equação 2.

Em que $|S|$ é o número de atributos na combinação. Para as quatro linhas na Equação 1 temos $|S| = 1, 2, 2, 3$ e portanto respectivamente $\omega_1, 2, 3, 4 = 1/3, 1/6, 1/6, 1/3$.

Vamos detalhar o exemplo de um modelo com três atributos A, B e C. A predição que tentamos explicar é, por exemplo, f(5, 3, 10) = 7. Uma das etapas para obter os valores de Shapley para cada recurso é calcular f (5, 3, -).

Substituir isso por um valor de expectativa equivale a fazer a seguinte pergunta: "Considerando que xA = 5 e xB = 3, qual é a predição que esperamos?". Em outras palavras, calculamos as previsões f(5, 3, xC) para cada valor xC do conjunto de treinamento e, em seguida, obtemos a média geral das previsões.

Se mais de um recurso for deixado de fora, serão realizados mais cálculos, mas o princípio permanece o mesmo. Ao calcular f(-, -, 10), perguntamos: "Qual é a predição esperada do modelo quando xC = 10?". Novamente, pegamos todos os valores possíveis para xA e xB do conjunto de treinamento, mas agora criamos todos os pares possíveis de (xA, xB) antes de criar todas as predições e calcular a média. Se o conjunto de treinamento consistir em 100 linhas de dados, teremos 100 × 100 = 10.000 pares de valores para fazer predições.

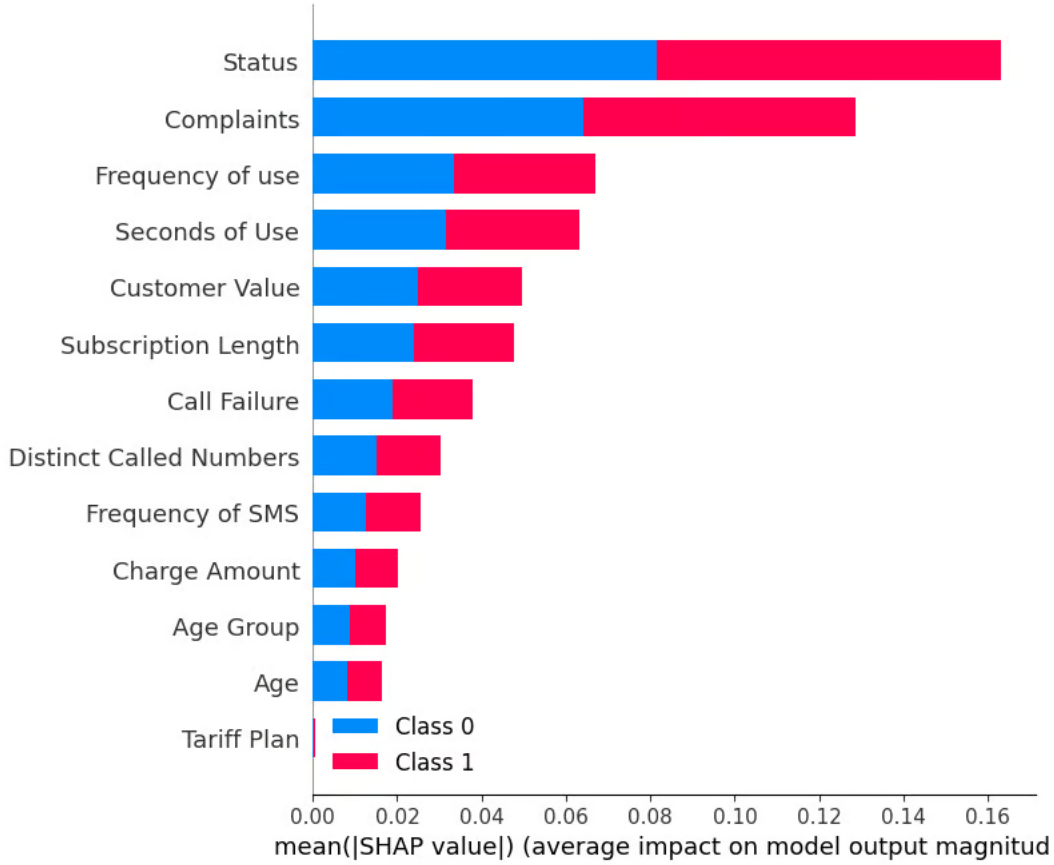
Por fim, f (-, -, -) é calculada de forma semelhante. Como antes, pegamos todos os valores possíveis xA, xB , xC do conjunto de treinamento, criamos todos os trios possíveis, geramos 100 × 100 × 100 = 1.000.000 previsões e calculamos a média. Agora que aproximamos as previsões do modelo para cada combinação, podemos inserir esses números na Equação 1 e obter os valores de Shapley para cada atributo (MOLNAR, 2020).

No contexto de ML, os valores de Shapley para os atributos podem ser utilizados para descobrir como a predição se distribui.

O cálculo dos valores de Shapley no entanto podem ter queda de desempenho no contexto de Deep Learning, pelo volume de dados apresentado. Portanto, a ferramenta SHAP é implementada eficientemente e permite que seja aplicada a uma variedade de modelos, mas não atende a todos os escopos (LUNDBERG; LEE, 2017).

As explicações oferecidas pelo SHAP são geradas em formatos de gráficos como: barra, sumário, beeswarm, importância, mapa de calor, dispersão de imagem e texto. Para ilustrar observaremos um exemplo de um conjunto de dados de uma empresa de telecomunicações iraniana, com cada instância representando um cliente durante um período de um ano. Junto com a classificação de rotatividade, há informações sobre a atividade dos clientes, como falhas de chamadas e duração da assinatura.

Figura 5 – Explicação do SHAP: Gráfico de importância dos atributos - Interpretação Global.



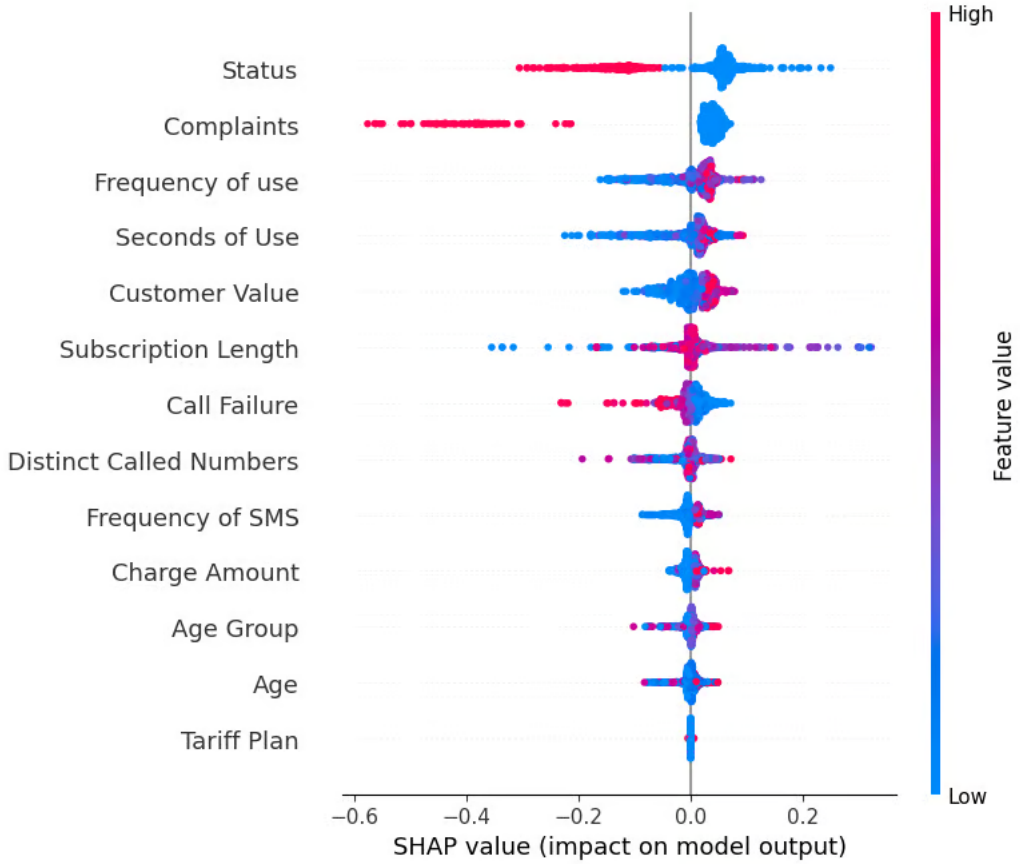
Fonte: (AWAN, 2023)

No gráfico de importância dos atributos da Figura 5 , o tamanho da barra azul diz respeito ao peso daquele atributo para classificações de instâncias como 0, enquanto a barra vermelha mostra a importância daquele atributo em classificações de instâncias como 1. O resultado do gráfico mostra que “Status”, “Reclamações” e “Frequência de uso” desempenham papéis importantes na determinação dos resultados para a base do exemplo.

Enquanto no gráfico sumário da Figura 6, o eixo Y indica os nomes dos atributos em ordem de importância, de cima para baixo. O eixo X representa o valor SHAP, a cor de cada ponto no gráfico representa o valor do atributo correspondente, com vermelho indicando valores altos e azul indicando valores baixos.

Ainda na Figura 6 observa-se que cada ponto representa uma instância dos dados do conjunto original. Ao observar o atributo “Reclamações”, vemos que apresenta valores altos com um valor SHAP negativo. Isso significa que contagens mais altas de reclamações tendem a afetar negativamente o resultado.

Figura 6 – Explicação do SHAP: Gráfico sumário - Interpretação de uma instância.



Fonte: (AWAN, 2023)

2.4 Considerações Finais

Neste trabalho buscou-se estudar técnicas de XAI e a escolha final se deu entre dois métodos agnósticos: o Método SHAP e o Algoritmo LIME. O SHAP apresenta vantagens como o fato da diferença entre uma previsão e a previsão média ser distribuída de forma justa entre os pesos dos atributos da instância. Em situações em que a lei exige explicabilidade, como o "direito a explicações" da União Europeia, tal vantagem faz do SHAP um dos únicos métodos em conformidade com a lei (MOLNAR, 2020).

Os valores de Shapley também permitem explicações mais contrastantes. Em vez de comparar uma previsão com a previsão média de todo o conjunto de dados, você pode compará-la com um subconjunto ou até mesmo com um único ponto de dados. O SHAP é um dos únicos métodos de explicação com uma teoria baseada nos axiomas - eficiência, simetria, dummy, aditividade. Dando a técnica uma base teórica razoável (MOLNAR, 2020).

A principal desvantagem do Método SHAP é que os cálculos dos valores de Shapley exigem muito recurso computacional porque o tempo de treinamento cresce exponencialmente com o número de atributos. Outra limitação é que, para uma única instância, o

valor de Shapley retorna um único valor de Shapley por atributo, não fornecendo o modelo de previsão como o LIME faz (LIU et al., 2017).

Outra desvantagem é que você precisa de acesso aos dados se quiser calcular o valor Shapley para novos dados. Não é suficiente acessar a função de previsão porque você precisa dos dados para substituir partes da instância de interesse com valores de instâncias dos dados originais como citado anteriormente no exemplo com 3 atributos.

A explicação gerada pelo Algoritmo LIME é bastante simples. Entretanto, ela exige a definição de uma vizinhança, que se mal escolhida pode ser ocasionalmente instável. A instabilidade da explicação do LIME vem do fato de que ela depende do número de instâncias geradas e da largura do kernel escolhido, o que determina o tamanho da vizinhança. Essa vizinhança define o nível de localidade da explicação. Uma vizinhança significativa precisa ser pequena o suficiente para atingir a linearidade local, mas grande o suficiente para evitar tendências erradas na explicação global.

Portanto, o LIME pode apresentar instabilidade devido à amostragem aleatória de sua vizinhança, o que o torna menos consistente entre as execuções. Se uma interpretação estável e consistente for crucial, especialmente em aplicações sensíveis, é preferível usar o SHAP.

Então, para modelos mais simples em que a interpretabilidade localizada é suficiente, o LIME é mais adequado. Aplicações como detecção de fraudes, classificações incorretas de imagens e classificação de textos são perfeitas para que o LIME fornecendo insights claros. O SHAP é útil para modelos complexos de ML, pois ele oferece uma perspectiva mais ampla da contribuição dos recursos. É ideal para aplicações como pontuação de crédito, redes neurais complexas para reconhecimento de imagens, oferecendo uma visão geral abrangente da importância dos recursos em nível global (FATHIMA, 2024).

Principalmente pelo maior entendimento oferecido em suas explicações pelo Algoritmo LIME de forma mais simples e pelo fato do modelo black-box escolhido não se tratar de uma IA que realiza Deep Learning, o LIME foi priorizado no desenvolvimento deste trabalho.

3 Revisão Bibliográfica

Para o avanço deste trabalho, uma etapa fundamental é a revisão da literatura sobre o tema abordado. Essa seção contém os detalhes e resultados encontrados.

3.1 Inteligência artificial explicável (XAI)

A área de IA Explicável tem-se desenvolvido nas últimas décadas e nela destacam-se trabalhos como (MUELLER et al., 2019) que definiu como avaliar a explicabilidade e publicou uma análise histórica dos métodos de XAI. Estes foram classificados em três gerações:

- A primeira geração buscou descrever explicitamente o processo de funcionamento interno dos modelos, transformando as regras em expressões de linguagem natural, em meados da década de 70.
- A segunda geração enfatizou o desenvolvimento de sistemas humano-computador que forneciam assistência cognitiva, centrando-se no conhecimento humano a partir do início dos anos 2000.
- Os sistemas de terceira geração voltaram a esclarecer o funcionamento interno dos sistemas, da mesma forma que a primeira geração. No entanto, os sistemas de terceira geração abordaram marjoritariamente sistemas black-box, aproximadamente a partir de 2012.

O trabalho realizado por (MUELLER et al., 2019) foi importante para contextualizar a evolução do campo de IA Explicável e auxiliar na localização no tempo e nos desafios que já foram superados nas últimas décadas. A força do artigo está em seu amplo escopo e abordagem interdisciplinar, que integra perspectivas da ciência cognitiva, interação humano-computador e inteligência artificial. Ao categorizar e analisar uma ampla gama de literatura sobre XAI, os autores oferecem uma visão geral estruturada que é particularmente útil para os recém-chegados ao campo.

No entanto, o artigo tem algumas limitações como a falta de ponderação crítica das metodologias discutidas. Os autores não abordam as possíveis compensações entre a precisão do modelo e a explicabilidade, nem avaliam a escalabilidade de várias técnicas de XAI em aplicações do mundo real.

Em (ZHANG, 2018) foi mostrado como as explicações podem ser utilizadas para detectar problemas de representação em aprendizados, especialmente em Deep Learning,

devido aos viéses induzidos nos dados de treinamento. Tais estudos tiveram importância para reforçar a necessidade de direcionar esforços para aumentar a explicabilidade de modelos de IA. Uma das principais contribuições desta obra é sua ênfase em aplicações práticas de interpretabilidade visual. O autor não apenas analisa abordagens teóricas, mas também discute como essas técnicas podem ser aplicadas a problemas na realidade, como classificação de imagens médicas e detecção de objetos.

Um problema do artigo é o foco restrito na interpretabilidade visual. Embora esse seja um aspecto importante da XAI, o autor não explora como as técnicas visuais podem ser integradas a outras formas de explicação, como explicações textuais ou simbólicas.

Já em (MESKE; BUNDE, 2020) há um trabalho interessante de abordagem centrada no usuário, enfatizando a importância de projetar explicações que não sejam apenas tecnicamente sólidas, mas também significativas e acionáveis para os usuários finais. Outro ponto forte deste trabalho é a realização de um estudo de usuário para avaliar a eficácia das explicações agnósticas de modelo. Essa abordagem agrega um valor significativo ao artigo, pois vai além das discussões teóricas para fornecer evidências concretas de como as explicações afetam o comportamento do usuário. Os resultados sugerem que as explicações agnósticas em relação ao modelo podem, de fato, aumentar a transparência e a confiança, o que é um insight valioso para este trabalho que busca fazer um experimento semelhante de esclarecimento de modelos black-box.

Uma limitação do artigo porém é que o escopo é restrito no estudo de usuários. Há uma limitação a um contexto específico (suporte à decisão baseado em visão computacional) e a um tamanho de amostra relativamente pequeno. O que levanta dúvidas sobre a generalização das descobertas para outros domínios.

Há um desenvolvimento interessante sobre o escopo das técnicas de XAI em (GUIDOTTI, 2018), sendo elas categorizadas em técnicas específicas e técnicas agnósticas. Também existe uma investigação de uma variedade de abordagens para explicar modelos black-box de maior complexidade, incluindo técnicas de mineração de dados.

Os autores focaram ainda na análise de técnicas como o LIME e o SHAP, mas também discutem métodos menos conhecidos. A comparação dos primeiros citados foi importante para este trabalho considerando que se tratou da decisão final de qual método escolher.

No entanto, o artigo tem algumas limitações como a falta da apresentação de perspectiva para o futuro. Por mais que haja um excelente resumo dos métodos existentes, ele não discute as tendências emergentes ou as questões de pesquisa em aberto na área.

O algoritmo LIME é proposto em (RIBEIRO; SINGH; GUESTRIN, 2016) como um método para explicar modelos, o LIME, ao apresentar previsões individuais representativas e as suas explicações de uma forma não redundante e demonstrando a flexibilidade destes

métodos ao explicar diferentes modelos para texto (por exemplo, florestas aleatórias) e classificação de imagens (por exemplo, redes neuronais). Este é o principal trabalho de referência para desenvolver projetos relacionados ao Algoritmo LIME, apresenta todas as demonstrações e exemplos necessários pra o total entendimento do modelo.

O ponto interessante de (ROSS; HUGHES; DOSHI-VELEZ, 2017) foi que o trabalho aproveita explicações do LIME para orientar o treinamento de outros modelos de IA. Algo que de certa forma também foi feito neste trabalho na seção dos experimentos. Os autores também propõem uma estrutura que penaliza os desvios das justificativas restringindo efetivamente o modelo para produzir explicações que sejam precisas e interpretáveis. Houveram avaliações empíricas e utilização do Algoritmo LIME como base para avaliar a consistência e o custo computacional de cada explicação proposta.

Enquanto o ponto fraco é a complexidade computacional da abordagem proposta. A restrição de explicações durante o treinamento adiciona uma sobrecarga em termos de execução, o que pode tornar o método menos prático para aplicações de maior escala ou em tempo real. Fica claro que o LIME é computacionalmente caro quando a abordagem dos autores agrava o problema integrando as restrições de explicação no loop de treinamento.

No trabalho de (LUNDBERG; LEE, 2017) obteve-se a definição e o entendimento do SHAP (Shapley Additive Explanations). Há uma extensa explicação sobre como são calculados os valores de Shapley, que indicam a influência de cada atributo na classificação do modelo a ser explicado e também uma diversificação de implementações como Kernel SHAP e Tree SHAP para abranger modelos particularmente desafiadores computacionalmente. Porém o artigo poderia ter feito uma avaliação mais abrangente da robustez do SHAP em diferentes arquiteturas de modelos. Embora exista a aplicação do SHAP em várias bases de dados, os autores não o comparam tanto com outros métodos de explicação em termos de interpretabilidade, eficiência computacional ou satisfação do usuário.

Em (MOLNAR, 2020) há a explicação mais didática de como o SHAP designa a cada atributo um valor de importância para uma previsão específica. Os valores de Shapley são a única solução que satisfaz as propriedades de Eficiência, Simetria, Dummy e Aditividade. O SHAP satisfaz essas propriedades calculando os valores únicos de Shapley. O trabalho também fornece explicações detalhadas sobre importância de atributos, gráficos de dependência parcial (PDPs) e gráficos de expectativa condicional individual (ICE) juntamente com exemplos práticos e trechos de código. Essa abordagem didática torna o livro interessante para este trabalho e seu desenvolvimento.

Existe também uma discussão sobre a interpretabilidade intrínseca (por exemplo: modelos lineares, árvores de decisão) versus a interpretabilidade post-hoc (por exemplo: SHAP e LIME) fornecendo uma estrutura útil para pensar sobre as compensações entre a complexidade do modelo e a interpretabilidade. Um ponto não tão interessante do livro é que seu foco em oferecer um melhor entendimento dos conceitos tira espaço de observar os

métodos sendo aplicados de forma consistente.

Para a escolha entre o SHAP e o LIME, foi estudado principalmente o trabalho de (SALIH et al., 2024), que trouxe pontuações importantes acerca de ambos ao fornecer uma discussão detalhada sobre os fundamentos teóricos, os requisitos computacionais e os casos de uso prático do SHAP e do LIME. Os autores discutem como o SHAP e o LIME podem ser usados para interpretar modelos complexos em domínios como saúde, finanças e sistemas autônomos.

Outro ponto forte do artigo é sua avaliação crítica do SHAP e do LIME, os autores discutem as limitações desses métodos, como a complexidade computacional do SHAP e a possibilidade de o LIME produzir explicações instáveis ou inconsistentes. O que permite uma compreensão mais sutil das vantagens e desvantagens envolvidas no uso dessas técnicas. Há também discussão sobre avanços e extensões recentes, como o Kernel SHAP e o Tree SHAP, acrescentando profundidade ao artigo e destacando os esforços em andamento para abordar essas limitações.

Um ponto negativo porém, é que foco do artigo nos métodos de forma técnica prejudica uma discussão mais ampla sobre os fatores humanos envolvidos na XAI. Os autores não abordam, por exemplo, como as explicações de cada técnica podem ser adaptadas a diferentes grupos de usuários (especialistas no domínio versus usuários comuns) ou como os vieses cognitivos podem afetar a interpretação das explicações. Uma abordagem mais interdisciplinar das explicações poderia ter reforçado a relevância de cada método no mundo real.

3.2 kNN Ponderado (Weighted K-Nearest-Neighbor Classifier)

Enquanto em (NABABAN; SITOMPUL; TULUS, 2018) e há uma tentativa de aprimorar o algoritmo K-Nearest Neighbor (kNN) incorporando a ponderação de atributos com base no método Gain Ratio. Há a abordagem do principal problema do kNN: o tratamento igual de todos os recursos, o que pode levar a um desempenho abaixo do ideal em conjuntos de dados com atributos irrelevantes ou ruidosos. Ao ponderar os atributos de acordo com sua importância, os autores pretendem melhorar a precisão e a robustez do kNN. O artigo é interessante por sua simplicidade, já que o método proposto é fácil de implementar e não exige modificações complexas no algoritmo kNN. Existe um explicação clara dos autores sobre a metodologia e os resultados.

Um dos únicos problemas do artigo talvez seja não ter uma comparação com outras técnicas de ponderação de atributos. Por exemplo, os autores não comparam sua abordagem com outros métodos de seleção ou ponderação de recursos como Qui-Quadrado.

Outro artigo importante para estudar a ponderação de atributos no kNN foi (SYED, 2014) em que há a atribuição de pesos aos atributos com base em sua importância visando melhorar a capacidade do algoritmo de discernir padrões relevantes nos dados.

Neste artigo já há a exploração de diferentes estratégias de ponderação. A abordagem mais geral de atribuir pesos aos atributos com base em sua contribuição permitiu que este trabalho escolhesse em seu capítulo de Metodologia a forma mais simples citada em (NABABAN; SITOMPUL; TULUS, 2018).

3.3 Considerações Finais

Após o estudo e mapeamento do estado atual da área de IA Explicável, a importância de aumentar da explicabilidade em IAs e o possível aumento de escopo proporcionado pelo campo de XAI, foram buscadas referências de aplicações do Algoritmo LIME para realizar a análise de como desenvolver um trabalho que se relacionasse com os projetos existentes da área.

O primeiro e essencial passo foi dado debruçando-se sobre (RIBEIRO; SINGH; GUESTRIN, 2016) para compreender totalmente o algoritmo e visualizar seu funcionamento e suas aplicações básicas. Imprescindível para desenvolver este trabalho desde o início.

Em seguida foi visitado (ROSS; HUGHES; DOSHI-VELEZ, 2017) que compara o LIME com outras estratégias de XAI e discute sobretudo o desempenho do LIME e o quanto ele pode ser computacionalmente custoso para muitas instâncias, o que foi muito relevante para este trabalho já que buscou-se executar o LIME inúmeras vezes para extração de pesos para atributos. Também destaca-se nesse sentido (DIEBER; KIRRANE, 2020) em que foi possível observar em aplicações os pontos fortes e fracos do LIME.

Para a compreensão do SHAP, o trabalho de maior influência é o (LUNDBERG; LEE, 2017), em que há a sua definição detalhada. Mas igualmente importante foi (SALIH et al., 2024) que ajudou na comparação do SHAP com o LIME, para realizar a decisão de qual técnica de XAI utilizar.

Por último é válido mencionar que antes de utilizar o kNN Ponderado com pesos gerados neste trabalho foram visitados (NABABAN; SITOMPUL; TULUS, 2018) e (SYED, 2014), contribuindo para verificar que podia e como podia ser aplicado de forma correta.

4 Metodologia

A metodologia escolhida consiste em selecionar duas bases de dados pertinente, selecionar um modelo de IA black-box, empregar o algoritmo LIME de inteligência artificial explicável e visualizar os resultados. Busca-se proporcionar assim uma maior interpretação e entendimento sobre quais atributos mais interferem no resultado do modelo black-box.

Com o auxílio de duas versões do classificador k-vizinhos mais próximos (kNN), uma simples e outra ponderada ajustando o peso dos atributos de acordo com o resultado do LIME, deseja-se analisar e comparar as métricas de qualidade dos modelos utilizados nos experimentos para mensurar sua confiabilidade.

O objetivo desse capítulo é informar sobre as bases de dados, algoritmos, métodos e outros aspectos da metodologia utilizados durante o trabalho.

4.1 Ambiente de execução

Para a realização dos experimentos presentes neste trabalho, a plataforma Google Colab foi utilizada, que é um ambiente de desenvolvimento baseado na nuvem que oferece suporte para Jupyter Notebooks e permite a execução de códigos Python.

Os experimentos foram conduzidos utilizando o ambiente do Colab para que todos os métodos fossem analisados seguindo os mesmos parâmetros de configuração. A configuração específica geralmente incluiu o sistema operacional Linux baseado no Ubuntu (fornecido pelo Google Colab) e a versão 3.11.11 do Python.

Essa infraestrutura permitiu a execução dos modelos propostos, e que o custo computacional e a disponibilidade de recursos estivessem balanceados.

As principais bibliotecas utilizadas foram a sklearn (version: 1.6.1), a pandas (version: 2.2.2) a matplotlib (version: 3.10.0) e por fim a numpy (version: 1.26.4).

As principais utilidades dessas bibliotecas foram: a implementação do MLP, dos classificadores kNN e das medidas de interesse (sklearn); manipulação de dados (pandas) e a visualização de métricas (matplotlib). Também foi utilizada a implementação do algoritmo LIME em seu framework no Python (lime v0.2.0.1).

4.2 Bases de dados

Para a análise dos métodos e comparação foram escolhidas duas bases de dados que permitiram a interpretação dos resultados. As bases foram utilizada em diversos artigos

da área e estão disponibilizada para ser usada livremente.

4.2.1 Diabetes - NIDDK

A base de dados é fornecida pelo Instituto Nacional de Diabetes (NIDDK) e Doenças Digestivas e Renais dos Estados Unidos da América, estando em domínio público. A coleta de dados foi feita em pacientes do sexo feminino, com pelo menos 21 anos de idade e de origem indígena Pima, da região do Arizona. O objetivo é diagnosticar diabetes do tipo 2 (V, 2022).

Além da coluna do diagnóstico de diabetes (Sim ou Não) a base apresenta 8 atributos:

- Pressão: pressão arterial diastólica.
- Glicose: concentração de glicose no plasma sanguíneo.
- BMI: Body Mass Index ou Índice de massa corporal.
- Insulina: níveis de insulina pós-glicemia de 2 horas.
- Nº Gravidez: número de vezes que a paciente ficou grávida.
- Idade: idade da paciente no momento da coleta de dados.
- DPF: Diabetes Pedigree Function é uma função que pontua a probabilidade de diabetes com base no histórico familiar, com um intervalo de 0,08 a 2,42.
- Esp. Pele: espessura da dobra cutânea da pele no tríceps.

4.2.2 Rain in Australia - Commonwealth of Australia

Os dados foram coletados em várias estações meteorológicas pela Bureau de Meteorologia, uma Agência Executiva do Governo australiano responsável por proporcionar serviços de meteorologia, hidrologia e clima da Austrália e suas áreas circundantes (YOUNG; ADAMYOUNG, 2021).

Além da coluna da previsão de chuva para o dia seguinte (Sim ou Não) a base apresenta 12 atributos:

- Temperatura Mínima: a temperatura mínima do dia em graus Celsius.
- Temperatura Máxima: a temperatura máxima do dia em graus Celsius.
- Chuva: a quantidade de chuva registrada no dia em milímetros.

- Rajada de vento: a velocidade (km/h) da rajada de vento mais forte nas 24 horas até a meia-noite.
- Velocidade do Vento 9am: a velocidade média do vento (km/h) medida antes das 9 horas do período da manhã.
- Velocidade do Vento 3pm: a velocidade média do vento (km/h) medida antes das 15 horas do período da tarde.
- Umidade 9am: Umidade percentual medida às 9 horas do período da manhã.
- Umidade 3pm: Umidade percentual medida às 15 horas do período da tarde.
- Pressão 9am: Pressão atmosférica (hpa) reduzida ao nível médio do mar às 9 horas do período da manhã.
- Pressão 3pm: Pressão atmosférica (hpa) reduzida ao nível médio do mar às 15 horas do período da tarde.
- Temperatura 9am: a temperatura em graus Celsius medida às 9 horas do período da manhã.
- Temperatura 3pm: a temperatura em graus Celsius medida às 15 horas do período da tarde.

4.3 Algoritmos Utilizados

Utilizou-se três algoritmos diferentes: o Classificador MLP, o Algoritmo kNN simples e o Algoritmo kNN Ponderado. Em que para o último foi oportuno fazer a execução duas vezes: na primeira vez, apenas com os pesos gerados pelo LIME para uma instância e na segunda vez com a média aritmética de cada peso encontrado para os atributos em muitas instâncias.

Portanto inicialmente coletou-se métricas do Modelo MLP e pelo kNN simples para tê-los como referência. Depois estimou-se as medidas de desempenho para o kNN Ponderado 1, ajustado pelos pesos gerados pelo LIME para uma instância e por fim as métricas foram mensuradas para o kNN Ponderado 2, ajustado pela média aritmética simples dos pesos gerados pelo LIME para 100 instâncias.

Posteriormente, foram comparadas as métricas do kNN Simples com o kNN Ponderado 1, observando se a influência dos atributos reflete a importância deles na predição. Também foram avaliadas as mudanças na qualidade do kNN Ponderado 1 para o kNN Ponderado 2, considerando a mudança de levar em consideração uma instância para levar em consideração a média de muitas instâncias no cálculo dos pesos designados aos atributos.

4.3.1 MLP

A implementação do MLP da biblioteca sklearn conta com parâmetros como:

1. activation: utilizado para a seleção da função de ativação, sendo usado a ReLU na padronização dos experimentos.
2. learning_rate_init: usado para definir a taxa de aprendizado inicial.
3. hidden_layer_sizes: que define a quantidade camadas ocultas e também o número de neurônios por camada.
4. train_test_split: estabelece a proporção da divisão dos dados entre treinamento e testes.
5. random_state: que força uma divisão específica e possibilita uma comparação direta de resultados.

A função de ativação escolhida para fazer a conexão entre os neurônios de diferentes camadas foi a função sigmoide e a unidade linear retificada (ReLU). A função ReLU foi selecionada por ser a mais utilizada no campo devido a suas vantagens nos resultados dos treinamentos. A função é dada por:

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

Neste trabalho decidiu-se utilizar 4 camadas escondidas para as bases de dados, para um total de 6 camadas, considerando as camadas de entrada e saída. Enquanto a quantidade de neurônios em cada camada variou conforme as bases de dados, sendo igual à quantidade de atributos da base.

O principal critério de parada de execução é quando não há variação maior do que 0.0001 na métrica *training loss*, que é calculado a soma dos erros para os casos de teste. No capítulo de experimentos pôde-se observar que o modelo MLP atingiu essa métrica para todas as execuções. O que sugere que quando o *training loss* se estabiliza é improvável que continuar com o treinamento vá produzir melhorias significativas. Essa estabilização pode se dever à convergência, às configurações da taxa de aprendizado, ao *overfitting* ou às limitações de precisão numérica.

4.3.2 Algoritmo LIME

Para utilização do Algoritmo LIME foi necessário definir qual explicador da classe será utilizado (Tabular, Text ou Image), utilizando como parâmetros: o vetor de dados de treinamento, os nomes dos atributos da base de dados e os nomes das classes. Bem como

definir o modo utilizado na classificação, sendo os principais classificação ou regressão. Neste trabalho utilizou-se o modo de classificação.

Para obter a explicação com a função correspondente, é necessário fornecer: uma instância, uma *prediction function* (uma função em que a entrada é um vetor numpy e o retorno são as *prediction probabilities*) e o número máximo de atributos presentes na explicação.

4.3.3 Algoritmo kNN

Foram utilizadas duas versões do algoritmo kNN, sendo ambas baseadas na biblioteca sklearn. A primeira versão é o kNN Simples oferecido pela biblioteca, que tem como parâmetro apenas o número k de vizinhos mais próximos e a fórmula para calcular a proximidade das instâncias. Neste trabalho a escolhida foi a distância Euclideana, dada por:

$$\text{distância}(x_1, x_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$$

A segunda versão foi a utilizada no kNN Ponderado 1 e no kNN Ponderado 2, tem como parâmetros de destaque o número k de vizinhos e a função desejada para calcular a proximidade das instâncias. A função foi reescrita para receber os pesos designados aos atributos sendo calculada pela distância ponderada Euclideana:

$$\text{distância}(x_1, x_2) = \sqrt{\sum_{i=1}^{n} w_i(x_{1i} - x_{2i})^2}$$

Sendo ω cada peso designado ao atributo da base de dados.

4.4 Medidas de Avaliação

Para a análise comparativa dos métodos computacionais foram utilizadas as seguintes métricas: tempo de execução, acurácia, precisão, recall, F1 Score e Jaccard.

4.4.1 Tempo de execução

O tempo de execução é uma métrica de eficiência computacional que determina o tempo necessário para a execução de um algoritmo com base na quantidade de entradas geradas. Nessa métrica, é mais interessante um tempo menor de execução.

Avaliar o tempo de execução é muito importante para a métrica da eficiência, pois um tempo de execução de algoritmos menor resulta em uma maior eficácia e a possibilidade

de obtenção de resultados de forma mais ágil, utilizando menos recursos computacionais (PAPADIMITRIOU, 2003).

4.4.2 Acurácia

Se trata da proporção de acertos positivos e negativos em relação ao total de previsões. É calculada por:

$$\text{Acurácia} = \frac{\text{AcertosPos} + \text{AcertosNeg}}{\text{AcertosPos} + \text{AcertosNeg} + \text{FalsosPos} + \text{FalsosNeg}}$$

4.4.3 Precisão

É equivalente a proporção de acertos positivos em relação ao número total de previsões positivas. Apresenta a seguinte fórmula:

$$\text{Precisão} = \frac{\text{AcertosPos}}{\text{AcertosPos} + \text{FalsosPos}}$$

4.4.4 Recall

É dado pela taxa de acertos positivos em relação ao total de instâncias que eram para ser positivas, portanto, os acertos positivos e os falsos negativos. Calculado por:

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

4.4.5 F1 Score

A Pontuação F1, ou F1 Score, é considerada como a média harmônica entre a precisão e o recall. É dado pela fórmula:

$$\text{F1Score} = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}}$$

4.4.6 Jaccard

O índice de Jaccard demonstra a similaridade dos conjuntos de previsões e dos resultados esperados, estimado pela divisão da interseção pela união desses conjuntos:

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

4.5 Divisão de Dados e Avaliação

A base de dados Diabetes Dataset apresenta 768 entradas e a base de dados Rain in Australia apresenta 11959 instâncias. Ambas foram separada na proporção 70% para treinamento e 30% para testes de validação.

A divisão 70/30 foi considerada suficiente para a avaliação dos modelo pois o desempenho do modelo se mostrou estável.

4.6 Considerações Finais

Um ponto relevante é que tanto o kNN Simples quanto o kNN Ponderado, apresentam o cálculo direto de suas distâncias para cada instância e seus atributos, portanto, quando as quantidades de ambos os parâmetros sobem, o custo computacional e o tempo de execução sobem consideravelmente.

Uma possível melhoria na validação poderia ser utilizar a validação cruzada para cobrir a possibilidade dos modelos apresentarem alta variação, hiperparâmetros ou conjuntos de dados desequilibrados.

Após a execução dos experimentos deste trabalho, a metodologia aqui descrita pôde ser considerada como satisfatória para a geração de resultados.

5 Experimentos e resultados

O experimento visou, com seus resultados, responder à quatro perguntas:

a) Qual é a influência dos atributos em um modelo black-box, no caso o MLP? A influência dos atributos no modelo black-box pode ser utilizada para aumentar a explicabilidade do modelo?

b) A influência dos atributos extraída pelo Algoritmo LIME reflete a importância do atributo na predição?

c) Existe variação da influência dos atributos fornecida pelo Algoritmo LIME em diferentes instâncias da base?

d) A média do desempenho do Algoritmo LIME melhora a expressividade da influência global dos atributos no modelo de classificação?

O experimento conduzido foi feito em um conjunto de etapas listadas abaixo.

1. Primeira etapa do experimento:
   - Classificação da base de dados pelo Modelo MLP e pelo kNN simples coletando métricas de ambos para tê-los como referência de desempenho.
2. Segunda etapa do experimento:
   - Aplicou-se o Algoritmo LIME no MLP para obter a influência dos atributos em uma única instância, respondendo o item a).
   - Os pesos encontrados pelo Algoritmo LIME para uma instância foram utilizados para ajustar o kNN Ponderado-1.
   - Comparou-se o desempenho nas métricas de avaliação do kNN simples com o kNN Ponderado-1 para auxiliar a responder ao item b).
3. Terceira etapa do experimento:
   - Em seguida utilizou-se o Algoritmo LIME no MLP para encontrar a influência dos atributos em diferentes instâncias analisando se há variação do peso dos atributos a medida que mudam as instâncias e suas respectivas as classes.
   - Agora, os pesos encontrados pelo Algoritmo LIME para 100 instâncias foram utilizados para ajustar o kNN Ponderado 2.
   - Finalmente comparou-se o desempenho nas métricas de avaliação do kNN Ponderado-1 com o kNN Ponderado-2 para auxiliar a responder aos itens c) e d).

5.1 Base de dados Diabetes - NIDDK

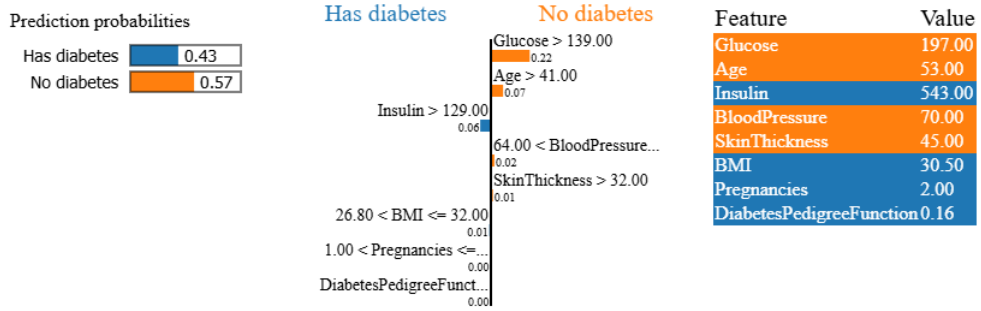
A base de dados de pacientes possivelmente diagnosticados com diabetes (V, 2022) conta com 8 atributos: pressão arterial, taxa de glicose, índice de massa corporal, nível de insulina, número de gravidez, idade, Diabetes Pedigree Function e Espessura da Pele. Apresentando 768 instâncias, foi a escolhida para ser utilizada primeiro.

Na primeira etapa do experimento, foram coletadas as métricas de desempenho para o MLP e para o kNN simples presentes na Tabela 3. No início da segunda etapa obteve-se, para a instância 8 da base de dados, os pesos dos atributos da Tabela 1 e a explicação presente na Figura 7.

Tabela 1 – Influência dos atributos gerada pelo LIME na instância 8 da base de dados Diabetes.

Instância	Pressão Art.	Glicose	BMI	Insulina	Nº Gravidez	Idade	DPF	Esp. Pele	Predição
8	-0.024	-0.222	0.014	-0.061	0.001	-0.077	-0.004	-0.016	Não

Figura 7 – Explicação gerada pelo LIME na instância 8 da base de dados Diabetes.



Fonte: elaborado pelo autor.

Para interpretar os resultados é preciso entender que os valores de cada atributo na Tabela 1 são a proporção da importância daquele atributo para a decisão do modelo. Portanto, quanto maior o valor absoluto, maior a importância para a predição. Valores positivos exercem influência para diagnóstico de diabetes (Sim) e valores negativos para a não existência da doença no paciente (Não).

Como é possível observar na Figura 7 a explicação fornecida pelo Algoritmo LIME oferece à esquerda da imagem o quão certo o algoritmo está da sua previsão (quanto maior o desequilíbrio dos valores, mais confiante o modelo está) ao centro da imagem, estão os atributos exercendo sua influência numérica e, por fim, à direita da imagem, há a lista dos valores brutos reais dos atributos para a instância.

Então, fica visualmente claro no centro da Figura 7 com a explicação oferecida pelo Algoritmo LIME mostrando com barras azuis e laranjas a disputa dos atributos para a tomada de decisão do classificador.

Enquanto o que a Tabela 1 demonstra é que a quantificação da importância dos atributos, assim como citado no capítulo de Fundamentação Teórica, é feita de forma a separá-los em valores positivos e negativos, para posteriormente verificar se a previsão será para uma classe ou para a outra.

Seguindo com o desenvolvimento da segunda etapa do experimento, foram utilizados os pesos encontrados para os atributos pelo Algoritmo LIME para a instância 8 (Tabela 1) para ajustar o kNN Ponderado 1, gerando as métricas de sua coluna na Tabela 3.

Entrando na terceira etapa, o Algoritmo LIME foi usado no MLP para encontrar a influência dos atributos para um recorte de 100 instâncias. Então, foi feita uma média aritmética simples utilizando os pesos dos atributos das 100 instâncias, em valores absolutos, para ajustar o kNN Ponderado 2. A Tabela 2 apresenta as 15 primeiras instâncias para serem exibidas e assim oferecer uma melhor visualização da importância dos atributos das instâncias originais.

Tabela 2 – Pesos designados aos atributos - LIME explicando MLP para a base de dados Diabetes.

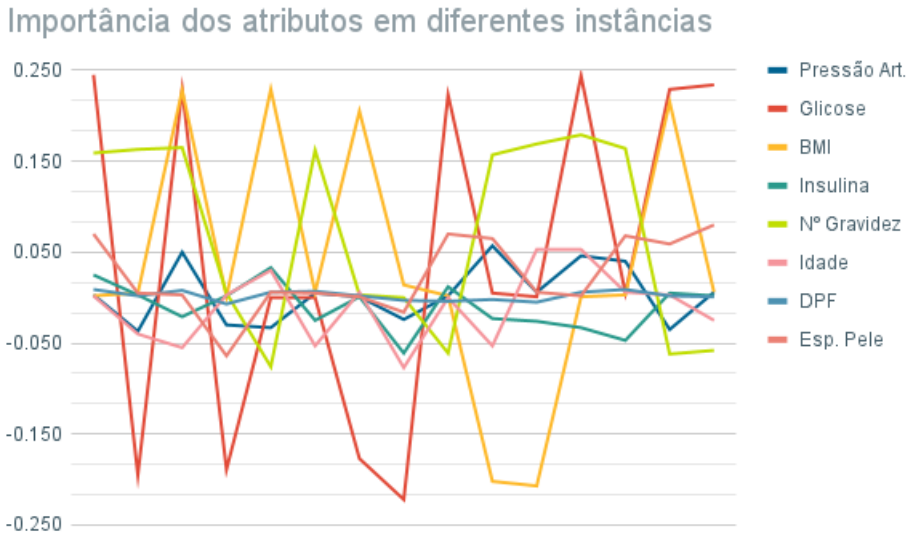
Instância	**Pressão Art.**	**Glicose**	**BMI**	**Insulina**	**Nº Gravidez**	**Idade**	**DPF**	**Esp. Pele**	**Predição**
1	0.003	0.245	0.002	0.025	0.159	0.002	0.009	0.070	Sim
2	-0.037	-0.195	0.005	0.003	0.163	-0.040	0.002	0.005	Não
3	0.050	0.229	0.228	-0.021	0.165	-0.055	0.008	0.003	Sim
4	-0.030	-0.189	0.001	0.002	0.005	0.003	-0.007	-0.064	Não
5	-0.033	0.000	0.229	0.033	-0.076	0.030	0.006	0.006	Sim
6	0.005	0.000	0.006	-0.025	0.162	-0.053	0.007	0.005	Sim
7	0.001	-0.177	0.205	0.001	0.003	0.006	0.002	0.001	Sim
8	-0.024	-0.222	0.014	-0.061	0.000	-0.077	-0.003	-0.016	Não
9	0.002	0.223	0.002	0.012	-0.061	0.000	-0.004	0.070	Sim
10	0.057	0.005	-0.202	-0.023	0.157	-0.053	-0.002	0.065	Não
11	0.006	0.001	-0.207	-0.026	0.169	0.053	-0.005	0.006	Não
12	0.046	0.244	0.001	-0.033	0.179	0.053	0.006	0.002	Sim
13	0.040	0.005	0.003	-0.047	0.164	0.006	0.009	0.068	Sim
14	-0.035	0.229	0.215	0.005	-0.062	0.003	0.002	0.059	Sim
15	0.006	0.234	0.006	0.002	-0.058	-0.025	0.001	0.080	Sim
Média	0.110	0.103	0.028	0.103	0.025	0.030	0.003	0.013	-

Quanto a Tabela 2, é notável que os atributos de Glicose e Número de Gravidez são os que exercem a maior influência em valores absolutos para as predições, enquanto a variável DPF é a que menos interfere nos resultados do classificador com os valores absolutos mais baixos na maioria das instâncias.

A Figura 8 apresenta um gráfico de linhas demonstrando visualmente a evolução dos valores dos pesos designados aos atributos para a base de dados de Diabetes oferecidos na Tabela 2.

Com o kNN Ponderado 2 devidamente ajustado com a média de cada peso, obteve-

Figura 8 – Gráfico de linhas dos pesos designados aos atributos de 100 instâncias.



Fonte: elaborado pelo autor.

se os valores das métricas presentes na Tabela 3. Além disso destaca-se que o tempo de execução do algoritmo kNN Ponderado 2 contém o tempo de treinamento do MLP (1.425s), o tempo do LIME para a retirada dos pesos de atributos de 100 instâncias (5s), somados ao tempo de treinamento do próprio kNN Ponderado (0.0962s).

Tabela 3 – Medidas de avaliação - Base de dados Diabetes.

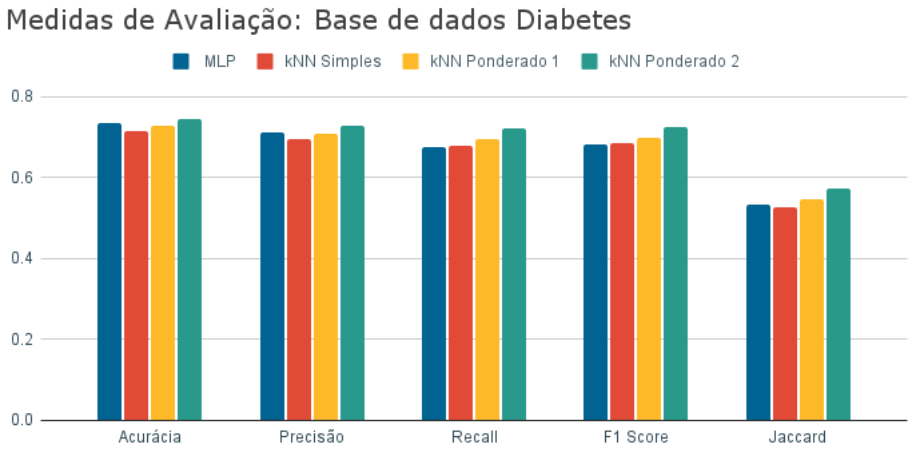
Métrica	**MLP**	**kNN Simples**	**kNN Ponderado 1**	**kNN Ponderado 2**
Tempo de Exec.	1.479s	0.003s	1.444s	6.521s
Acurácia	0.735	0.714	0.727	0.744
Precisão	0.712	0.695	0.710	0.727
Recall	0.674	0.679	0.694	0.722
F1 Score	0.683	0.684	0.699	0.724
Jaccard	0.534	0.528	0.545	0.573

É importante também retomar que o tempo de execução do algoritmo kNN ponderado 1 inclui o tempo de treinamento do MLP (0.454s) para a retirada dos pesos de atributos pelo LIME a uma única instância (0.112s), somado ao tempo de treinamento do próprio kNN Ponderado (0.065s).

Também traçou-se um gráfico de barras presente na Figura 9 para melhor visualização das métricas de desempenho da Tabela 3:

Vemos no gráfico da Figura 9 e na Tabela 3 que o kNN Ponderado 2 atingiu os melhores valores para as métricas de Tempo de Execução, Acurácia, Precisão, Recall e

Figura 9 – Gráfico de barras para as medidas de avaliação.



Fonte: elaborado pelo autor.

Jaccard e F1 Score.

No tocante a comparação proposta ao final da etapa 2, pôde-se observar que, nestes dados, a influência dos atributos extraída pelo Algoritmo LIME dos classificadores explicados melhorou o desempenho nas métricas do kNN ponderado 1 em relação ao kNN simples.

A respeito da comparação proposta ao final da etapa 3, consolidou-se que no Algoritmo LIME há variação da influência dos atributos quando também se variam as instâncias. Sendo visível na Tabela 3 que ao comparar o kNN Ponderado 1 com o kNN ponderado 2, que o segundo apresentou métricas melhores, atestando que realizar a média da influência dos pesos dos atributos em uma grande quantidade de instâncias causa uma melhora na expressividade da influência global no modelo de classificação.

5.2 Rain in Australia - Commonwealth of Australia

A base de dados preenchida com informações meteorológicas de cidades australianas (YOUNG; ADAMYOUNG, 2021) buscando prever se irá chover no dia seguinte ou não, conta com 12 atributos: Temperatura Mínima, Temperatura Máxima, Chuva no dia atual (em milímetros), Rajada de vento mais forte, Velocidade do Vento 9am, Velocidade do Vento 3pm, Umidade 9am, Umidade 3pm, Pressão 9am, Pressão 3pm, Temperatura 9am, Temperatura 3pm. Em que 9am e 3pm se referem respectivamente à coleta do dado no horário das 9h da manhã e às 15h da tarde. Apresenta 11959 instâncias.

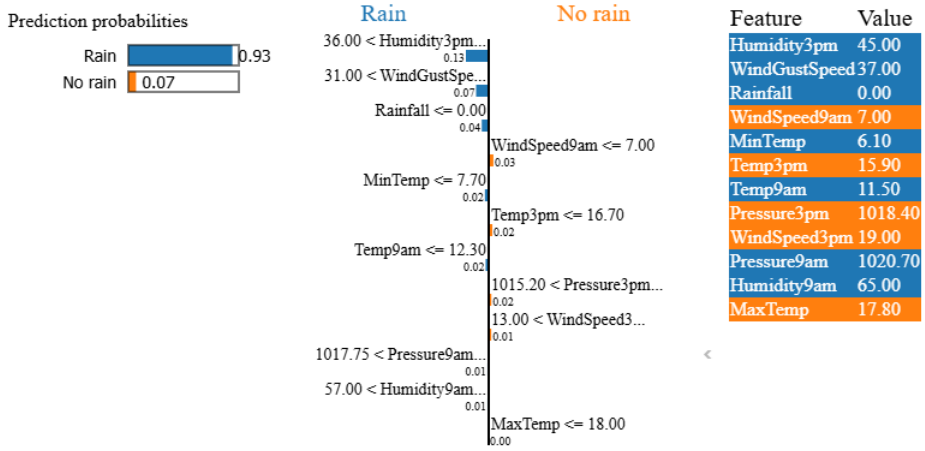
Novamente, para executar a primeira etapa do experimento, coletou-se as métricas de desempenho para o MLP e para o kNN simples presentes na Tabela 6. Para iniciar a segunda etapa, buscou-se obter, para a instância 3 da base de dados, os pesos dos atributos

da Tabela 4 e a explicação presente na Figura 10.

Tabela 4 – Influência dos atributos gerada pelo LIME na instância 3 da base Rain in Australia.

Instância	Temp. Min.	Temp. Max.	Chuva	Rajada Vento	Vel. Vento 9am	Vel. Vento 3pm	Umidade 9am	Umidade 3pm	Pressão 9am	Pressão 3pm	Temp. 9am	Temp. 3pm	Predição
3	-0.022	0.003	-0.041	-0.076	0.035	0.011	-0.001	-0.136	-0.014	0.028	-0.027	0.021	Sim

Figura 10 – Explicação gerada pelo LIME na instância 3 da base Rain in Australia.



Fonte: elaborado pelo autor.

Na Figura 10 a explicação oferecida pelo Algoritmo LIME mostra uma disputa entre os atributos mais equilibrada do que na instância da primeira base de dados deste capítulo. As barras azuis e laranjas são menores e, portanto, os pesos definidos, para a tomada de decisão do classificador, são menores e mais sensíveis.

A Tabela 4 oferece a quantificação da importância dos atributos, em valores positivos influenciando para a predição de não chover no dia seguinte e negativos pesando em favor da predição ser chover no dia seguinte, para que eles se somem posteriormente verifiquem se a previsão será para uma classe ou para o outra.

Seguindo com o desenvolvimento da segunda etapa do experimento, foram utilizados os pesos encontrados para os atributos pelo Algoritmo LIME para a instância 3 (Tabela 4) para ajustar o kNN Ponderado 1, gerando as métricas de sua coluna na Tabela 6.

A terceira etapa começou aplicando o Algoritmo LIME no MLP para encontrar a influência dos atributos para o mesmo recorte de 100 instâncias. Também foi realizada uma média aritmética simples utilizando os pesos dos atributos das 100 instâncias, em valores absolutos, para ajustar o kNN Ponderado 2. Assim como para a base anterior, a Tabela 5 apresenta as 15 primeiras instâncias para serem exibidas e assim oferecerem uma melhor visualização da importância dos atributos das instâncias originais.

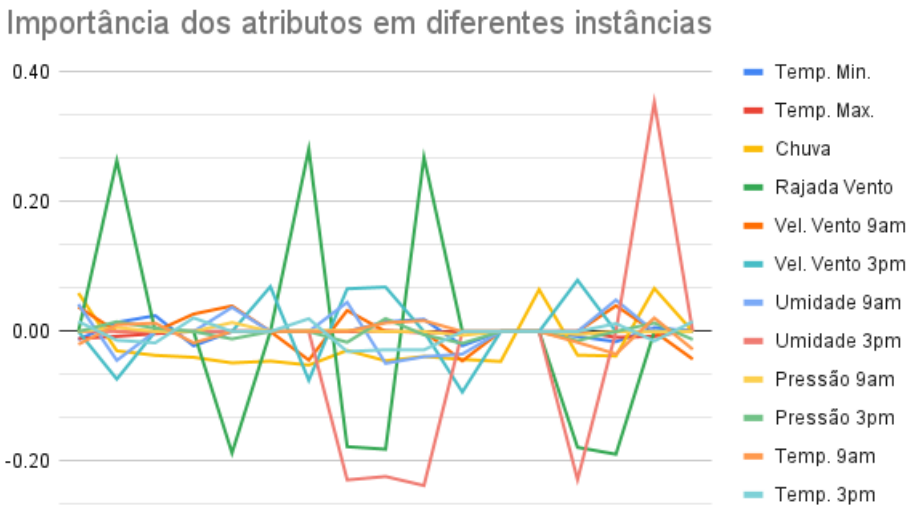
Tabela 5 – Pesos designados aos atributos - LIME explicando MLP para a base Rain in Australia.

Instância	Temp. Min.	Temp. Max.	Chuva	Rajada Vento	Vel. Vento 9am	Vel. Vento 3pm	Umidade 9am	Umidade 3pm	Pressão 9am	Pressão 3pm	Temp. 9am	Temp. 3pm	Predição
1	-0.012	-0.011	0.059	0.000	0.038	0.000	0.042	0.000	0.000	0.000	-0.020	0.015	Sim
2	0.014	-0.007	-0.030	0.263	0.000	-0.073	-0.044	0.000	0.006	0.015	0.010	-0.013	Sim
3	-0.022	0.003	-0.041	-0.076	0.035	0.011	-0.001	-0.136	-0.014	0.028	-0.027	0.021	Sim
4	0.024	-0.003	-0.037	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.013	-0.017	Não
5	0.000	0.000	-0.048	-0.187	0.039	0.000	0.037	0.000	0.014	-0.011	0.000	0.000	Não
6	0.000	0.000	-0.046	0.000	0.000	0.069	0.000	0.000	0.000	0.000	0.000	0.000	Sim
7	0.000	0.000	-0.052	0.281	-0.044	-0.075	0.000	0.000	0.001	0.000	0.000	0.020	Sim
8	0.000	0.000	-0.029	-0.178	0.032	0.066	0.045	-0.229	0.002	-0.017	0.000	-0.031	Não
9	0.015	0.001	-0.045	-0.182	0.000	0.069	-0.049	-0.224	0.000	0.020	0.013	-0.028	Não
10	0.019	-0.002	-0.038	0.269	0.000	0.000	-0.039	-0.237	0.000	-0.005	0.017	-0.028	Não
11	-0.022	0.000	-0.043	0.000	-0.045	-0.093	-0.035	0.000	-0.006	-0.018	0.000	0.000	Não
12	0.000	0.000	-0.046	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Não
13	0.000	0.000	0.065	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Sim
14	-0.007	0.001	-0.036	-0.179	0.000	0.079	0.000	-0.228	-0.004	-0.014	-0.017	0.000	Não
15	-0.016	-0.009	-0.038	-0.189	0.040	0.000	0.049	0.000	0.000	0.000	-0.035	0.012	Não
Média	0.006	0.003	0.041	0.096	0.016	0.038	0.023	0.145	0.003	0.006	0.010	0.011	-

Observamos na Tabela 5 que de fato os atributos apresentam valores mais equilibrados pela maioria das instâncias e que apenas em casos específicos características como Rajada de Vento e Umidade 3pm exercem a maior influência em valores absolutos para as predições.

Um gráfico de linhas pode ser visto na Figura 11 para visualizar a evolução dos valores dos pesos designados aos atributos para a base de dados de (YOUNG; ADAMYOUNG, 2021) oferecidos na Tabela 5.

Figura 11 – Gráfico de linhas dos pesos designados aos atributos de 100 instâncias da base Rain in Australia.



Fonte: elaborado pelo autor.

Após a utilização da média dos pesos presente na última linha da Tabela 5 para ajustar o kNN Ponderado 2, aplicou-se o algortimo na base de dados e foram encontrados os valores das métricas presentes na Tabela 6. Também destaca-se que o tempo de execução do algoritmo kNN Ponderado 2 contém o tempo de treinamento do MLP (2.551s), o tempo

do LIME para a retirada dos pesos de atributos de 100 instâncias (5s), somados ao tempo de treinamento do próprio kNN Ponderado (0.081s).

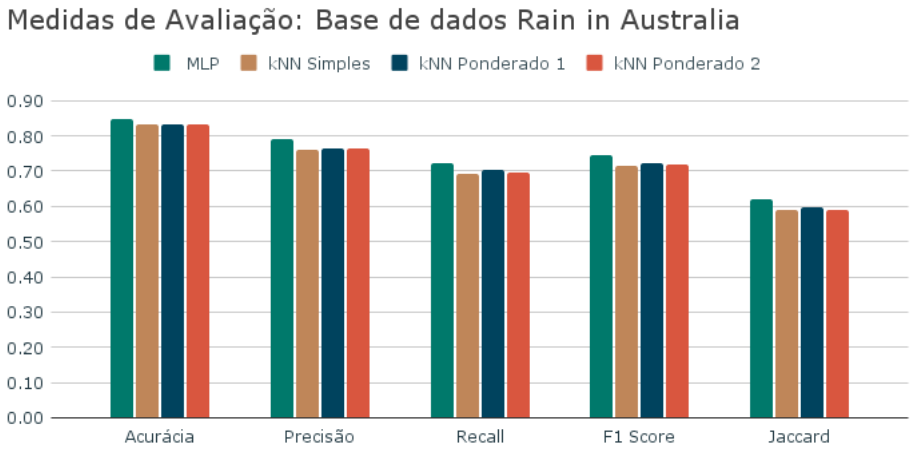
Tabela 6 – Medidas de avaliação - Base de dados Rain in Australia.

Métrica	MLP	kNN Simples	kNN Ponderado 1	kNN Ponderado 2
Tempo de Exec.	2.237s	0.003s	2.612s	7.632s
Acurácia	0.849	0.831	0.833	0.832
Precisão	0.790	0.761	0.764	0.763
Recall	0.721	0.694	0.702	0.695
F1 Score	0.746	0.717	0.724	0.719
Jaccard	0.622	0.590	0.598	0.591

Reitera-se que o tempo de execução do algoritmo kNN ponderado 1 inclui o tempo de treinamento do MLP (2.488s) para a retirada dos pesos de atributos pelo LIME a uma única instância (0.167s), somado ao tempo de treinamento do próprio kNN Ponderado (0.078s).

Na Figura 12 há um gráfico de barras traçado para oferecer melhor visualização das métricas de desempenho da Tabela 6:

Figura 12 – Gráfico de barras para as medidas de avaliação - Base de dados Rain in Australia.



Fonte: elaborado pelo autor.

Ao observarmos a Figura 12 e a Tabela 6 vemos que numa base de dados com mais instâncias o kNN Ponderado 1 e o kNN Ponderado 2 atingiram resultados muito semelhantes. Com os melhores valores para as métricas de Tempo de Execução, Acurácia, Precisão, Recall e Jaccard e F1 Score sendo obtidos pelo kNN Ponderado 1. O que, retomando a comparação proposta ao final da etapa 2, indica que nestes dados a influência

dos atributos extraída pelo Algoritmo LIME dos classificadores explicados melhora pouco, mas ainda melhora o desempenho nas métricas do kNN ponderado 1 em relação ao kNN simples.

Enquanto a comparação proposta ao final da etapa 3, em uma base de dados mais robusta o Algoritmo LIME continuou apresentando variação da influência dos atributos quando se variam as instâncias.

A Tabela 6 mostra que ao comparar o kNN Ponderado 1 com o kNN ponderado 2, em uma base com mais instâncias, que o primeiro apresentou métricas melhores, e portanto, realizar a média da influência dos pesos dos atributos em uma grande quantidade de instâncias não causou uma melhora significativa na expressividade da influência global no modelo de classificação.

5.3 Considerações Finais

É relevante citar que possíveis limitações do experimento são: ter utilizado uma amostra de 100 instâncias para gerar os pesos dos atributos para o kNN Ponderado 2 e existir a possibilidade, inerente a escolha do Algoritmo LIME, da definição da vizinhança feita pelo o algoritmo nessa base de dados não tenha sido ideal, o que pode ocasionar em resultados instáveis.

No tocante das perguntas estabelecidas no começo deste capítulo, os resultados mostraram, com as Tabelas 1 e 4 e a Figuras 7 e 10, qual a influência de cada atributo nas decisões do Multilayer Perceptron, um modelo black-box. Os experimentos também sugerem que os pesos dos atributos gerados pelo Algoritmo LIME podem ser utilizados para aumentar a explicabilidade do modelo black-box. Já que torna explícita qual característica foi mais determinante para tal classificação.

Então, na Tabela 3 e na Figura 9 vemos que para as métricas de Tempo de Execução, Acurácia, Precisão, Recall e Jaccard e F1 Score o kNN Ponderado 2 desempenhou melhor, o que nos sugere que existem casos em que modelos mais simples são capazes de substituir modelos complexos sem muitas perdas, como conclui (SARKAR et al., 2016).

Já na Tabela 6 e na Figura 12 observa-se que houve melhor desempenho do kNN Ponderado 1 para as métricas de Tempo de Execução, Acurácia, Precisão, Recall e Jaccard e F1 Score. O que indica que nem sempre a média do desempenho do Algoritmo LIME irá melhorar significativamente a expressividade da influência global dos atributos no modelo de classificação.

Destaca-se também nas Tabelas 3 e 6 as diferenças de custo computacional, representado pelo tempo de execução, em que o kNN Ponderado 2, por contar com a existência do tempo de geração dos pesos dos atributos pelo Algoritmo LIME, exige um custo maior.

Ainda foi possível visualizar nas Figuras 9 e 12 que a influência dos atributos extraída pelo Algoritmo LIME dos classificadores explicados foi positiva, pois em ambas as bases de dados, houve melhora do desempenho nas métricas do kNN ponderado 1 em relação ao kNN simples.

Também pôde-se observar com o auxílio das Tabelas 2 e 5 e na Figuras 8 e 11 que os valores numéricos da influência de cada atributo, quando extraída pelo Algoritmo LIME, variam em diferentes instâncias da base.

Por fim, ao comparar as Tabelas 3 e 6 vemos que quando comparamos o desempenho do kNN Ponderado 1, ajustado com os pesos gerados para uma instância, com o desempenho do kNN Ponderado 2, ajustado com a média dos pesos gerados para 100 instâncias, para a primeira base de dados houve melhora a expressividade da influência global dos atributos no modelo de classificação com o uso da média do desempenho do Algoritmo LIME, enquanto para a segunda base de dados, não houve tal melhora na expressividade. O que pode indicar que o recorte de 100 instâncias seja um recorte pequeno para bases de dados maiores.

6 Conclusão

Ao retomarmos os objetivos deste trabalho, é possível estabelecer que foi realizado um estudo exploratório de técnicas de IA Explicável, abordando principalmente o Algoritmo LIME. Então, desenvolveu-se um experimento que inicialmente buscou detalhar quais os critérios de maior importância eleitos por um modelo black-box, o Multilayer Perceptron, em sua tomada de decisão.

Possibilitou-se visualizar que, sem as explicações geradas pelas técnicas de XAI, surgem dificuldades para o cérebro humano com a opacidade no funcionamento de algoritmos black-box, afinal conforme o aumento do número de camadas e neurônios do modelo, e o consequente aumento em sua complexidade, prejudica-se o entendimento de como uma classificação é realizada.

Então, foi possível observar nos primeiros resultados experimentais que, com a ajuda do Algoritmo LIME esclarece-se o quanto cada atributo é importante ao Modelo MLP em seu funcionamento, aumentando sua interpretabilidade e confiabilidade.

Com o desenvolvimento do experimento, pôde-se explorar possibilidades de classificação com o Algoritmo kNN Simples e com uma variação, o kNN Ponderado, que viabilizou aplicar pesos para cada atributo da base de dados. Sendo ambos classificadores white-box, que apresentam funcionamento claro em seus processos de previsão, eles permitiram fazer uma comparação direta nas medidas de desempenho com o MLP, um modelo black-box.

Os resultados indicaram que nas métricas de Acurácia, de Precisão, de Recall, de F1 Score e de Índice de Jaccard, com exceção do kNN simples, os modelos white-box ponderados obtiveram resultados melhores. Enquanto para o custo computacional mensurado pelo Tempo de Execução, o modelo MLP foi melhor do que o kNN Ponderado mas não supera o kNN simples. Sugerindo que a substituição de modelos black-box por modelos mais simples é uma abordagem viável.

Porém, é importante salientar que o Modelo MLP utilizado não foi totalmente otimizado e que os resultados são particulares as bases de dados estudadas. A literatura indica que, para casos de maior complexidade, é provável que o MLP apresente melhores resultados.

Ao dar enfoque aos pesos extraídos pelo Algoritmo LIME para os atributos nas previsões do MLP, o trabalho encontrou que os valores descobertos melhoraram, para ambas as bases de dados, o desempenho nas métricas do kNN Ponderado (ajustado com os pesos dos atributos retirados de uma instância) em relação ao kNN simples.

Entretanto, ao realizar uma média dos pesos extraídos pelo LIME para 100 instân-

cias, notou-se que a média do desempenho do Algoritmo LIME melhorou a expressividade da influência global dos atributos no modelo de classificação para a primeira base de dados, mas não melhorou tal expressividade para a segunda base de dados.

Isto é, a segunda versão do kNN Ponderado, ao ser ajustada com valores da média dos pesos, se saiu melhor do que a primeira versão que havia sido ajustada com apenas os valores dos pesos de apenas uma instância somente na base com menos instâncias. O que sugere que o recorte de 100 instâncias possa ser insuficiente para abordar a influência global de bases de dados maiores.

Portanto, os resultados aqui vistos são interessantes para o entendimento das técnicas de IA Explicável, afinal, ficou notável o aumento da explicabilidade que métodos como o LIME oferecem ao mensurar a importância de cada característica da base de dados e produzir uma explicação local para modelos black-box visualmente simples de ser lida.

Foi possível, então, indicar a viabilidade das técnicas de XAI, o que sugere um caminho promissor para a área, que deve continuar crescendo com o aumento da importância de modelos de IA na sociedade.

6.1 Pesquisas Futuras

Poderia ser uma abordagem interessante de estudo realizar a aplicação das técnicas de XAI citadas neste trabalho como SHAP e LIME em bases de dados comparando as explicações geradas por ambas as técnicas e julgando qual a melhor utilização para cada contexto.

Também seria viável fazer com que no experimento, o LIME ao extrair os pesos dos atributos de cada instância no MLP, imediatamente já utilize tais pesos no kNN para ponderar a classificação da mesma instância. Verificando se existem instâncias que maximizam o ganho.

Seria possível ainda realizar um novo trabalho aumentando o número de instâncias utilizadas para mensurar os pesos dos atributos, para verificar se há um limite de melhora para a extração do Algoritmo LIME, apesar de ser um experimento que seria computacionalmente muito custoso.

Finalmente, utilizar outro algoritmo white-box no lugar do kNN para realizar a substituição e comparação com um modelo black-box. Visto que a técnica de substituição de modelo se mostrou promissora.

Referências

ADADI, A.; BERRADA, M. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, v. 6, p. 52138–52160, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:52965836>. Citado 2 vezes nas páginas 16 e 22.

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, Association for Computing Machinery, v. 22, n. 2, p. 207–216, 1993. Disponível em: <https://doi.org/10.1145/170036.170072>. Citado na página 19.

AWAN, A. A. *An Introduction to SHAP Values and Machine Learning Interpretability*. 2023. Disponível em: <https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability>. Citado 2 vezes nas páginas 27 e 28.

AYTEKIN, C. *Neural Networks are Decision Trees*. 2022. Disponível em: <https://arxiv.org/abs/2210.05189>. Citado na página 23.

BAEHRENS, D. et al. How to explain individual classification decisions. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 1803–1831, ago. 2010. ISSN 1532-4435. Citado na página 22.

BURL, M. C.; FOWLKES, C.; RODEN, J. Mining for image content. *Systemics, cybernetics, and informatics/information systems: analysis and synthesis*, 1999. Citado na página 19.

DIEBER, J.; KIRRANE, S. *Why model why? Assessing the strengths and limitations of LIME*. 2020. Disponível em: <https://arxiv.org/abs/2012.00093>. Citado na página 35.

FATHIMA, S. *LIME vs SHAP: A Comparative Analysis of Interpretability Tools*. 2024. Disponível em: <https://www.markovml.com/blog/lime-vs-shap#:~:text=If%20you%20have%20a%20simpler,both%20local%20and%20global%20interpretability.> Citado na página 29.

FREITAS, P. N. e P. M. *Inteligência Artificial e regulação de algoritmos*. 2018. Available at: <https://www.academia.edu/39044468/InteligÃłncia_Artificial_e_RegulaÃğÃčo_de_algoritmos>. Citado na página 16.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Machine Learning*, v. 29, p. 131–163, 11 1997. Citado na página 19.

GUIDOTTI, R. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, v. 51, n. 5, p. 1–42, 2018. Citado 2 vezes nas páginas 15 e 32.

HECKERMAN, D. Bayesian networks for data mining. *Data Min. Knowl. Discov.*, v. 1, p. 79–119, 03 1997. Citado na página 19.

HOFFMAN, R. R. et al. *Metrics for Explainable AI: Challenges and Prospects*. 2019. Disponível em: <https://arxiv.org/abs/1812.04608>. Citado na página 16.

LI, B.; YU, S.; LU, Q. *An Improved k-Nearest Neighbor Algorithm for Text Categorization*. 2003. Disponível em: <https://arxiv.org/abs/cs/0306099>. Citado na página 20.

LIU, S. et al. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, v. 1, n. 1, p. 48–56, 2017. ISSN 2468-502X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2468502X17300086>. Citado na página 29.

LODWICH, A.; FRASCH, J.; BREUEL, T. *Report on Practical Bayes-True Data Generators For Evaluation of Machine Learning, Pattern Recognition and Data Mining Methods*. DFKI, 2007. Disponível em: <http://rgdoi.net/10.13140/RG.2.1.5045.4649>. Citado na página 20.

LUNDBERG, S.; LEE, S.-I. *A Unified Approach to Interpreting Model Predictions*. 2017. Disponível em: <https://arxiv.org/abs/1705.07874>. Citado 5 vezes nas páginas 16, 25, 26, 33 e 35.

MATSUMURA, R. *Perceptrons*. 2020. Disponível em: <https://ricardomatsumura.medium.com/perceptrons-f18935009a61>. Citado na página 21.

MCCULLOCH, W. S. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, p. 115–133, 1943. Citado na página 21.

MENA, J. *Data Mining Your Website*. [S.l.]: Digital Press, 1999. Citado na página 19.

MESKE, C.; BUNDE, E. Transparency and trust in human-ai-interaction: The role of model-agnostic explanations in computer vision-based decision support. In: DEGEN, H.; REINERMAN-JONES, L. (Ed.). *Artificial Intelligence in HCI*. Cham: Springer International Publishing, 2020. p. 54–69. Citado na página 32.

MOLNAR, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. [s.n.], 2020. Online book available at <https://christophm.github.io/interpretable-ml-book/>. Disponível em: <https://christophm.github.io/interpretable-ml-book/>. Citado 5 vezes nas páginas 23, 25, 26, 28 e 33.

MORALES, E. F.; ESCALANTE, H. J. A brief introduction to supervised, unsupervised, and reinforcement learning. In: TORRES-GARCíA, A. A. et al. (Ed.). *Biosignal Processing and Classification Using Computational Learning and Intelligence*. Academic Press, 2022. p. 111–129. ISBN 9780128201251. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780128201251000178>. Citado na página 15.

MUELLER, S. T. et al. *Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI*. 2019. Disponível em: <https://arxiv.org/abs/1902.01876>. Citado na página 31.

NABABAN, A. A.; SITOMPUL, O. S.; TULUS. Attribute weighting based k-nearest neighbor using gain ratio. *Journal of Physics: Conference Series*, IOP Publishing, v. 1007, n. 1, p. 012007, apr 2018. Disponível em: <https://dx.doi.org/10.1088/1742-6596/1007/1/012007>. Citado 2 vezes nas páginas 34 e 35.

NORVIG, S. R. P. *Artificial Intelligence: A Modern Approach*. [S.l.]: Prentice Hall, 2009. Citado na página 15.

PAPADIMITRIOU, C. H. Computational complexity. In: *Encyclopedia of Computer Science*. [S.l.]: John Wiley & Sons, 2003. p. 260–265. Citado na página 42.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. Disponível em: <https://arxiv.org/abs/1602.04938>. Citado 6 vezes nas páginas 16, 23, 24, 25, 32 e 35.

ROSS, A. S.; HUGHES, M. C.; DOSHI-VELEZ, F. *Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations*. 2017. Disponível em: <https://arxiv.org/abs/1703.03717>. Citado 2 vezes nas páginas 33 e 35.

SALIH, A. M. et al. A perspective on explainable artificial intelligence methods: Shap and lime. *Advanced Intelligent Systems*, Wiley, v. 7, n. 1, jun. 2024. ISSN 2640-4567. Disponível em: <http://dx.doi.org/10.1002/aisy.202400304>. Citado 2 vezes nas páginas 34 e 35.

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 53.

SYED, M. E. Attribute weighting in k-nearest neighbor classification. In: . [s.n.], 2014. Disponível em: <https://api.semanticscholar.org/CorpusID:118505539>. Citado na página 35.

TECH, D. *O que é e como funciona o algoritmo KNN?* 2024. Disponível em: <https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-knn/>. Citado na página 20.

V, B. J. C. *Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms*. 2022. Kaggle. DOI: https://doi.org/10.1007/s00521-022-07049-z. Citado 2 vezes nas páginas 38 e 46.

WANG, Z.; HAMZA, W.; SONG, L. *k-Nearest Neighbor Augmented Neural Networks for Text Classification*. 2017. Disponível em: <https://arxiv.org/abs/1708.07863>. Citado na página 20.

WU, Y. chen; FENG, J. wen. Development and application of artificial neural network. *Wireless Personal Communications*, Springer, v. 102, p. 1645–1656, 2018. Citado na página 15.

YOUNG, J.; ADAMYOUNG. *Rain in Australia*. 2021. Kaggle Hub. Http://www.bom.gov.au/climate/data. Citado 3 vezes nas páginas 38, 49 e 51.

ZHANG, Q.-s. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology And Electronic Engineering*, 2018. Citado na página 31.