UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

HENRIQUE DANTAS PIGHINI
Orientador: Prof. Dr. Rodrigo Cesar Pedrosa Silva

UM ESTUDO SOBRE MÉTRICAS PARA A AVALIAÇÃO DE ATRIBUTOS

Ouro Preto, MG
2025

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

HENRIQUE DANTAS PIGHINI

UM ESTUDO SOBRE MÉTRICAS PARA A AVALIAÇÃO DE ATRIBUTOS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Rodrigo Cesar Pedrosa Silva

Ouro Preto, MG
2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Henrique Dantas Pighini

Um estudo sobre métricas para a avaliação de atributos

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 4 de Abril de 2025.

Membros da banca

Rodrigo César Pedrosa Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Lauro Angelo Gonçalves de Moraes (Examinador) - Mestre - Programa de Pós Graduação em Ciência da Computação (UFOP)
Daniel Bortot de Salles (Examinador) - Bacharel - Programa de Pós Graduação em Ciência da Computação (UFOP)

Rodrigo César Pedrosa Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 4/04/2025

Documento assinado eletronicamente por **Rodrigo Cesar Pedrosa Silva**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 04/04/2025, às 15:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0886316** e o código CRC **62A57B4B**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Resumo

Esta monografia busca encontrar métricas de avaliação para extração de atributos adaptadas para problemas de classificação binária, visando melhorar o desempenho de algoritmos de aprendizado de máquina. O projeto explora métodos amplamente utilizados, como algoritmos genéticos e redes neurais MLPs, para criar *features* aleatórias e testar novas métricas de avaliação. Além de aprimorar o desempenho, o foco está na melhoria da interpretabilidade e robustez dos modelos. Experimentos nos principais modelos de *machine learning* foram realizados em diversos conjuntos de dados para analisar profundamente os métodos propostos, com o objetivo de fornecer *insights* valiosos sobre padrões em dados tabulares e contribuir para o avanço da ciência de dados e aprendizado de máquina.

Palavras-chave: Engenharia de Atributos. Extração de Características. Aprendizado de Máquina. Inteligência Artificial.

Abstract

This dissertation aims to identify evaluation metrics for feature extraction tailored to binary classification problems, with the goal of improving the performance of machine learning algorithms. The project explores widely used methods, such as genetic algorithms and MLP neural networks, to generate random features and test new evaluation metrics. In addition to enhancing performance, the focus is on improving model interpretability and robustness. Experiments on leading machine learning models were conducted across various datasets to thoroughly analyze the proposed methods, aiming to provide valuable insights into patterns in tabular data and contribute to the advancement of data science and machine learning.

Keywords: Feature Engineering. Feature Extraction. Machine Learning. Artificial Intelligence.

Lista de Ilustrações

Lista de Tabelas

Lista de Algoritmos

Sumário

1 Introdução

Nos últimos anos, vivemos em uma era de explosão de dados, marcada por um aumento exponencial na quantidade de informações geradas e coletadas. Sites registram cada clique dos usuários, nossos smartphones capturam conversas, fotos e preferências continuamente, dispositivos monitoram nossos sinais vitais e carros inteligentes rastreiam os hábitos dos motoristas em cada percurso. Essa vasta rede de dados está interconectada por meio da globalização, da internet e das redes sociais, abrangendo desde artigos científicos até os memes que circulam online (GRUS, 2019).

Nesse contexto, a inteligência artificial tem ganhado destaque, impulsionada pelo avanço das técnicas de mineração de dados e pelo crescente interesse de grandes corporações. O resultado disso é um grande crescimento nos grandes bancos de dados e o desenvolvimento de inúmeros novos algoritmos de previsão.

Assim sendo, tornou-se cada vez mais evidente o papel crucial desempenhado pela qualidade e interpretação dos dados na tomada de decisões informadas e na descoberta de insights valiosos. Como consequência, tanto o campo da aprendizagem de máquina quanto outros domínios têm adotado, em algum grau, uma abordagem centrada nos dados (GOLDSCHMIDT; PASSOS; BEZERRA, 2015). Essa abordagem destaca a importância de uma exploração profunda, análise, transformação e visualização dos dados por meio de diversas técnicas, ferramentas e metodologias.

Melhorar o desempenho de modelos aprendizado de máquina requer a extração eficiente de características (atributos) (OZDEMIR, 2022). Esse processo envolve identificar os melhores atributos para um problema específico, o que demanda uma análise intensiva e sistemática dos dados para identificar padrões, relações e características relevantes. Muitas vezes, essa etapa exige a contratação de especialistas para obter informações cruciais. Mesmo com os esforços dos cientistas de dados, as características extraídas nem sempre agregam informações substanciais ao problema, resultando em um desempenho final insatisfatório.

Essa dificuldade decorre do fato de que o processo de geração de características está intrinsecamente ligado à compreensão dos dados e à habilidade de programação do analista responsável, o que, por sua vez, influencia diretamente na qualidade das características produzidas (OZDEMIR, 2022).

A fim de simplificar o processo de engenharia de atributos, foram desenvolvidos algoritmos de AutoFE (*Auto Feature Engineering*) (SONG, 2018). Essas técnicas automatizadas facilitam a criação de variáveis relevantes para modelos de aprendizado de máquina, eliminando a necessidade da intervenção manual de cientistas de dados. O AutoFE gera automaticamente novas *features* a partir dos dados brutos, acelerando o processo de modelagem e potencialmente

melhorando o desempenho preditivo.

Entretanto, o uso de algoritmos automatizados de extração de atributos apresenta desafios, uma vez que cada conjunto de dados possui características únicas (OZDEMIR, 2022). A singularidade de cada projeto implica que, muitas vezes, não será possível descobrir *insights* relevantes apenas com ferramentas automatizadas.

Como alternativa, alguns estudos exploraram o aprimoramento de algoritmos de otimização, como os algoritmos genéticos (TAN, 2007), ou focaram em métricas específicas de otimização (NEMBRINI; KÖNIG; WRIGHT, 2018). Este último, em particular, demonstrou que o índice de Gini, apesar de ser uma métrica mais simples, pode ser utilizado em algoritmos de otimização de gradiente para alcançar resultados superiores, em comparação com modelos mais complexos e suas métricas associadas.

Essas pesquisas serviram como inspiração para o desenvolvimento deste trabalho. Nesse projeto será realizado um estudo aprofundado sobre como os modelos de inteligência artificial respondem à redimensionalização de seus bancos de dados, utilizando técnicas que serão exploradas ao longo do trabalho. A proposta central é demonstrar se, a partir de uma métrica de seleção de *features*, é possível redimensionar bancos de dados e obter resultados positivos.

Se a hipótese estiver correta, será possível determinar uma métrica que consegue, a partir de técnicas engenharia de atributos, melhorar os resultados obtidos pelos modelos de aprendizado de máquina.

Este projeto de pesquisa é relevante para o campo da ciência de dados e aprendizado de máquina, especialmente na extração de atributos informativos de dados tabulares, que são essenciais para gerar previsões e insights precisos e confiáveis. Os resultados deste trabalho terão implicações importantes no:

a) Aprimoramento do desempenho dos modelos: Ao aplicar técnicas de extração de atributos, o projeto visa melhorar o desempenho preditivo de modelos de aprendizado de máquina em dados tabulares. Isso resultará em previsões mais precisas para tarefas de classificação.

b) Aumento da interpretabilidade: A interpretabilidade dos atributos é fundamental para compreender as relações entre as variáveis de entrada e as previsões do modelo. Desenvolver métodos que extraem atributos interpretáveis de dados tabulares contribuirá para a transparência e confiabilidade dos modelos de aprendizado de máquina.

1.1 Objetivos

Este trabalho tem como objetivo investigar se, a partir de uma métrica de seleção de *features*, é possível criar novas características que resultem em modelos de melhor desempenho. A premissa é que atributos que anteriormente não apresentavam uma relação aparente possam, após a aplicação do algoritmo proposto, revelar essas relações, facilitando o trabalho dos modelos de predição.

Assim sendo, neste projeto foi realizado:

1. Desenvolvimento de um algoritmo de geração de *features*.
2. Estudo profundo sobre as métricas utilizadas por algoritmos de seleção de atributos.
3. Teste em diversos bancos de dados utilizando métodos estatísticos para encontrar as métricas mais significativas.
4. Análise dos resultados gerados e comparação das métricas.

2 Revisão Bibliográfica

2.1 Fundamentação Teórica

2.1.1 Dados tabulares

Dados tabulares são informações organizadas em uma tabela, onde as linhas representam registros ou entradas individuais, e as colunas correspondem a diferentes atributos ou variáveis desses registros, como nome, data, valor, ou categoria. Esse formato estruturado facilita a leitura e manipulação dos dados, permitindo que sejam facilmente filtrados, ordenados ou agrupados para análise.

Devido à sua consistência e organização, dados tabulares são amplamente utilizados em bancos de dados relacionais, planilhas e ferramentas de análise de dados. Cada coluna geralmente contém um tipo específico de dado, como números, textos ou datas, o que torna possível realizar operações complexas de análise, como cálculos estatísticos e visualizações gráficas, com maior precisão e eficiência.

Dados tabulares são amplamente utilizados em diversas áreas práticas, como medicina (ULMER; MEIJERINK; CINÀ, 2020), finanças (CLEMENTS et al., 2020), ciência do clima (VAGHEFI et al., 2023) e muitas outras aplicações baseadas em bancos de dados relacionais. Na última década, métodos tradicionais de aprendizado de máquina, como árvores de decisão com aumento de gradiente (GBDT), ainda dominam a modelagem de dados tabulares, demonstrando desempenho superior em relação ao aprendizado profundo (SHWARTZ-ZIV; ARMON, 2022). Isso ocorre pois em dados tabulares, as correlações entre as características tendem a ser mais dispersas e menos evidentes. Isso significa que uma determinada característica pode estar relacionada apenas a um subconjunto de outras características, em vez de apresentar correlação com todas elas (ARIK; PFISTER, 2021).

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S

Figura 2.1 – Um exemplo simples de um dado tabular.

2.1.2 Aprendizado de máquina

Aprendizado de máquina (*Machine Learning*) é uma disciplina dentro da inteligência artificial, que possibilita sistemas computacionais a aprender padrões e tomar decisões sem serem

explicitamente programados para tal (ZHOU, 2021). Essa abordagem tem aplicações em uma variedade de setores, desde diagnósticos médicos (ULMER; MEIJERINK; CINÀ, 2020), finanças (CLEMENTS et al., 2020) até reconhecimento de voz (ALI; ABDULLAH; FADHIL, 2021) e veículos autônomos (SONI et al., 2021).

Machine Learning se concentra no desenvolvimento de algoritmos capazes de aprender e melhorar sua performance ao longo do tempo (ZHOU, 2021). Ele se baseia na ideia de que os sistemas podem aprender com dados, identificar padrões e tomar decisões sem intervenção humana direta (HAHNE et al., 2008). Os principais conceitos incluem (ZHOU, 2021):

- **Modelos de aprendizado:** representações matemáticas que refletem o relacionamento entre variáveis em um conjunto de dados.
- **Treinamento e teste:** divisão do conjunto de dados em partes para treinar o modelo e avaliar sua performance.

Existem três tipos principais de aprendizado de máquina. O aprendizado supervisionado (NASTESKI, 2017), onde o modelo é treinado em um conjunto de dados rotulado, onde a entrada e a saída desejada são conhecidas. O objetivo é fazer previsões ou classificações. O aprendizado não supervisionado (HAHNE et al., 2008), onde o modelo é treinado em um conjunto de dados sem rótulos, buscando padrões e estruturas subjacentes. Agrupamento e redução de dimensionalidade são exemplos. E, por fim, o aprendizado por reforço (ZHANG et al., 2021), onde o modelo aprende através da interação com um ambiente, recebendo recompensas ou penalidades com base em suas ações.

Neste trabalho, adotaremos o aprendizado supervisionado, uma vez que nosso objetivo é construir modelos capazes de realizar previsões a partir de um conjunto de dados rotulado. A utilização desse paradigma permite avaliar o impacto das métricas de seleção de *features* na performance dos modelos de forma objetiva, comparando os resultados obtidos em diferentes experimentos.

Com essa abordagem definida, o próximo passo é apresentar os modelos de aprendizado de máquina que serão empregados na avaliação dos métodos propostos.

2.1.2.1 Árvore de Decisão

Árvore de decisão (SUTHAHARAN; SUTHAHARAN, 2016) é um método de aprendizado de máquina utilizado para tarefas de classificação e regressão. Elas são chamadas assim porque a tomada de decisão no modelo segue uma estrutura hierárquica semelhante a uma árvore, onde cada nó interno representa uma característica ou atributo dos dados, cada aresta corresponde a uma regra de decisão, e cada nó folha representa uma decisão ou resultado final.

O processo de construção de uma árvore de decisão começa com o nó raiz, que contém todos os dados de entrada. O algoritmo seleciona o atributo que melhor divide os dados com base

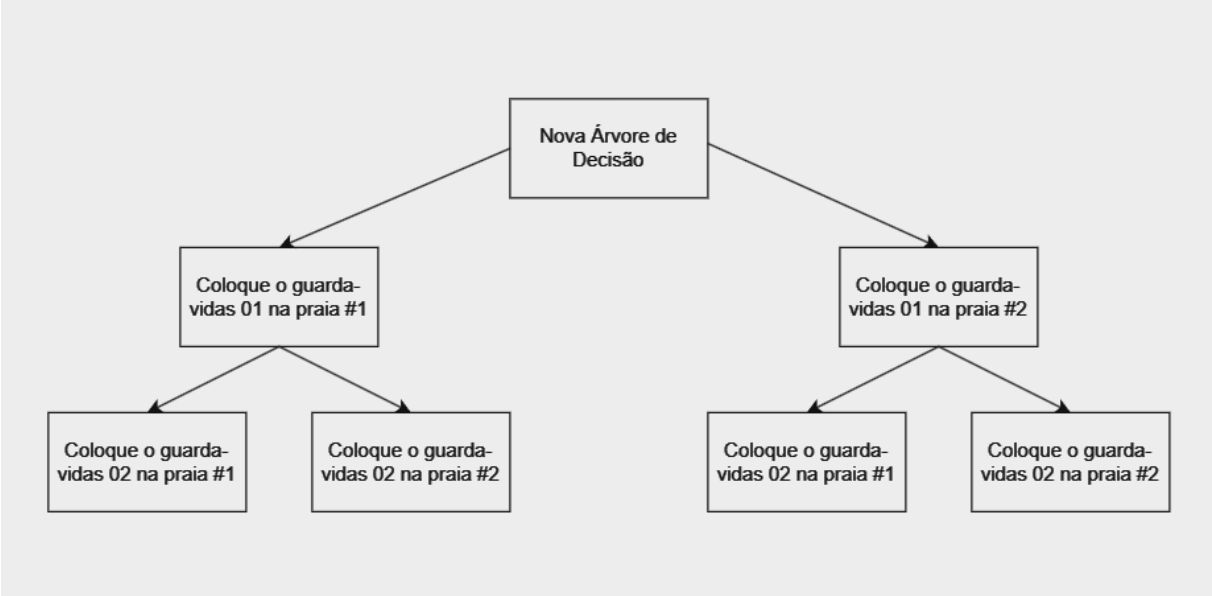

Figura 2.2 – Visualização simples de uma Árvore de Decisão, inspirado em (MAKOWER, 1976)

em um critério de divisão, como o índice de Gini ou a entropia. Os dados são então divididos em subconjuntos com base nesse atributo, e o processo é repetido recursivamente para cada subconjunto, criando ramos da árvore. Esse procedimento continua até que todos os dados sejam perfeitamente classificados ou até que um critério de parada, como a profundidade máxima da árvore ou o número mínimo de amostras por nó, seja atingido.

Um dos principais benefícios das árvores de decisão é sua interpretabilidade. As regras de decisão são facilmente compreendidas e podem ser visualizadas, o que torna esse modelo particularmente útil em situações onde a transparência é importante. No entanto, árvores de decisão simples podem sofrer de *overfitting*, ou seja, podem se ajustar muito aos dados de treinamento, capturando ruído em vez de padrões reais. Para mitigar esse problema, técnicas como poda de árvores, que remove partes da árvore que contribuem pouco para a generalização, são comumente usadas.

Além disso, árvores de decisão podem ser combinadas em métodos de aprendizado conjunto, como *Random Forest* e *Gradient Boosting*, para melhorar a precisão e a robustez do modelo. Esses métodos constroem múltiplas árvores de decisão e combinam suas previsões, resultando em um modelo mais poderoso e menos propenso a *overfitting*.

Árvores de decisão são amplamente utilizadas em diversas áreas, no entanto, sua performance pode ser limitada em casos onde os dados são altamente complexos e não lineares, exigindo a consideração de modelos mais sofisticados.

2.1.2.2 SVM

Máquina de Vetores de Suporte, ou SVM (*Support Vector Machines*) (XUE; YANG; CHEN, 2009), é uma poderosa técnica de aprendizado de máquina utilizada tanto para tarefas de classificação quanto para regressão. Essa abordagem é especialmente eficaz em espaços de alta dimensionalidade e é fundamentada na busca por um hiperplano de decisão otimizado para separar classes ou realizar previsões.

O principal objetivo das SVM é encontrar o hiperplano de decisão que melhor separa as instâncias pertencentes a diferentes classes em um espaço de características. Para isso, as SVM procuram maximizar a margem entre as instâncias das classes, definida como a distância entre o hiperplano e as instâncias mais próximas de cada classe, conhecidas como vetores de suporte. O hiperplano é um espaço de dimensão $n - 1$ em um espaço de n dimensões, onde n é o número de características do conjunto de dados. Em um problema de classificação binária, o hiperplano separa duas classes e é escolhido de maneira a maximizar a margem. A Figura 2.3 apresenta um exemplo simplificado de uma SVM.

A margem é a distância entre o hiperplano de decisão e os pontos mais próximos de cada classe. A SVM busca o hiperplano que maximiza essa margem, proporcionando uma maior robustez ao modelo. Os vetores de suporte são as instâncias do conjunto de dados que estão mais próximas do hiperplano de decisão. Eles desempenham um papel crucial na definição do hiperplano e na determinação da margem. A alteração ou remoção de vetores de suporte pode impactar significativamente a posição do hiperplano.

Em casos em que os dados não são linearmente separáveis, a SVM utiliza funções de *kernel* (HOFMANN; SCHÖLKOPF; SMOLA, 2008) para mapear os dados para um espaço de maior dimensão, onde a separação linear é possível. Isso permite que as SVM lidem eficazmente com problemas complexos. O processo de aplicar uma função de *kernel* aos dados é chamado de *kernel*ização. Os tipos comuns de funções de *kernel* incluem o *kernel* linear, o *kernel* polinomial e o *kernel* radial (RBF/Gaussiano).

As SVM são amplamente utilizadas em problemas de classificação, como reconhecimento de imagem (CHAGANTI et al., 2020), detecção de spam (KUMAR; BISWAS, 2017), diagnóstico médico (ROJAS-DOMÍNGUEZ et al., 2017), entre outros. Além disso, podem ser aplicadas em tarefas de regressão para prever valores contínuos (YANG; CHAN; KING, 2002). Elas são eficazes em espaços de alta dimensionalidade, podem lidar com conjuntos de dados complexos e são robustas em relação a *overfitting*, especialmente quando a margem é maximizada. No entanto, as SVM também têm desafios, como a sensibilidade à escala dos dados e a necessidade de ajustar cuidadosamente parâmetros, como a constante de regularização (C) e o tipo de *kernel*, para obter o melhor desempenho.

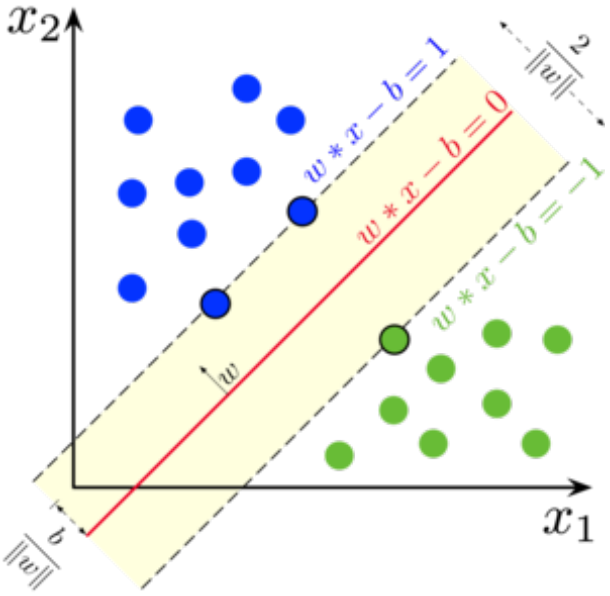

Figura 2.3 – Visualização simples de uma SVM, retirada de (LARHMAM, 2018)

2.1.2.3 Random Forest

Floresta aleatória (*Random Forest*) (RIGATTI, 2017) é um método de aprendizado em conjunto que se aplica a tarefas de classificação e regressão, construindo diversas árvores de decisão durante o treinamento. No contexto de classificação, a saída do modelo é a classe mais escolhida entre as árvores, enquanto em tarefas de regressão, a previsão é a média das previsões das árvores individuais. Essa técnica corrige a propensão das árvores de decisão em se ajustarem demais ao conjunto de treinamento.

Especificamente, quando as árvores são construídas com muita profundidade, elas podem aprender padrões excessivamente irregulares, resultando em baixo viés e alta variância. A floresta aleatório surge como uma solução, promovendo a média de várias árvores profundas treinadas em diferentes partes do mesmo conjunto de treinamento, visando a redução da variância. Essa abordagem implica em um ligeiro aumento no viés e alguma perda de interpretabilidade, mas, em geral, melhora significativamente o desempenho do modelo final. A Figura 2.4 apresenta um exemplo simplificado de uma random forest.

Embora as florestas aleatórias frequentemente superem uma única árvore de decisão em precisão, elas abrem mão da interpretabilidade inerente dessas árvores. As árvores de Decisão estão entre os modelos de aprendizado de máquina mais facilmente interpretáveis, juntamente com modelos lineares, baseados em regras e em atenção. Essa capacidade de interpretação é altamente valorizada, permitindo que os desenvolvedores confirmem que o modelo aprendeu informações realistas dos dados e permitindo que os usuários finais confiem nas decisões do modelo. Embora seja trivial seguir o caminho de decisão de uma única árvore, essa tarefa torna-se consideravelmente mais desafiadora com dezenas ou centenas de árvores. Para conciliar desempenho e interpretabilidade, algumas técnicas de compressão de modelo possibilitam transformar

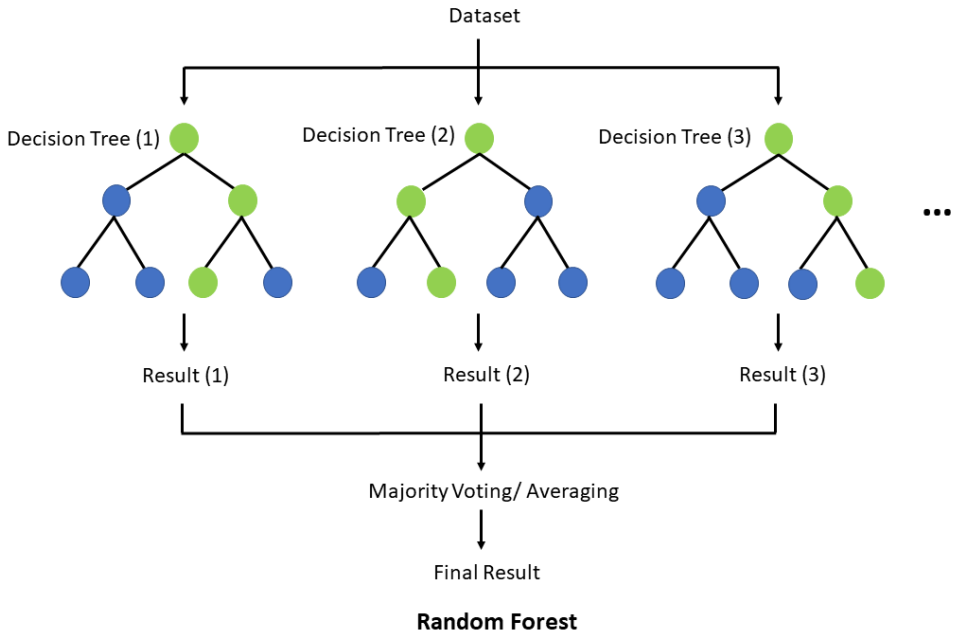

Figura 2.4 – Visualização simples de uma *Random Forest*, retirada de (TSEKICHUN, 2021)

uma floresta aleatória em uma árvore de Decisão "renascida", mantendo a mesma função de decisão. Em situações onde os atributos preditivos estão linearmente correlacionados com a variável-alvo, o uso de uma floresta aleatória pode não resultar em ganhos significativos de precisão para o modelo base. Adicionalmente, em problemas que envolvem múltiplas variáveis categóricas, a floresta aleatória pode não conseguir aumentar a precisão do modelo base.

2.1.2.4 Redes neurais

Redes neurais (PICTON; PICTON, 1994) são um paradigma de aprendizado de máquina inspirado na estrutura e no funcionamento do cérebro humano. Elas consistem em um conjunto interconectado de unidades básicas, chamadas neurônios artificiais, organizados em camadas. Cada neurônio recebe entradas, realiza cálculos ponderados e gera uma saída que pode ser usada como entrada para outros neurônios.

Cada neurônio artificial funciona como uma unidade de processamento. Eles recebem várias entradas, aplicam pesos a essas entradas, realizam uma soma ponderada e, em seguida, passam o resultado por uma função de ativação para gerar a saída e são organizados em camadas. A camada de entrada recebe os dados originais, a camada de saída produz as previsões ou resultados desejados, e entre elas, existem camadas intermediárias chamadas camadas ocultas. A profundidade da rede é determinada pelo número de camadas ocultas. A Figura 2.5 apresenta um exemplo simplificado de uma rede neural.

Cada conexão entre neurônios tem um peso associado. Esses pesos representam a força e

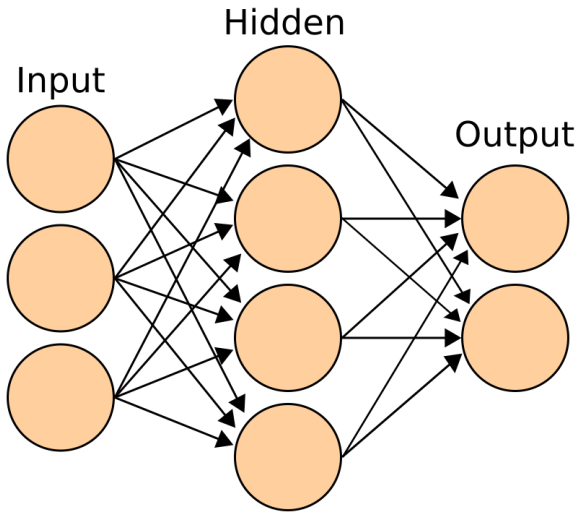

Figura 2.5 – Visualização simples de uma rede neural, retirada de (CBURNETT, 2006)

a direção da influência que uma unidade exerce sobre a outra. O processo de treinamento da rede envolve ajustar esses pesos para otimizar o desempenho. Após a soma ponderada das entradas, é aplicada uma função de ativação para introduzir não linearidades na rede. Isso permite que a rede aprenda padrões complexos e relações não lineares nos dados. O treinamento de uma rede neural envolve a apresentação de dados de treinamento à rede, ajustando os pesos com base nas diferenças entre as previsões da rede e os resultados reais. O algoritmo de otimização, como o gradiente descendente, é frequentemente usado nesse processo.

Redes neurais são utilizadas em diversas aplicações, incluindo reconhecimento de imagem (TRAORE; KAMSU-FOGUEM; TANGARA, 2018), processamento de linguagem natural (GOLDBERG, 2017), reconhecimento de voz (LYASHENKO et al., 2021), previsão de séries temporais (AZOFF, 1994), entre outros. Seu poder reside na capacidade de aprender representações automaticamente a partir dos dados. Porém, elas podem ser suscetíveis a *overfitting*, requerem grandes quantidades de dados para treinamento eficaz e podem ser computacionalmente intensivas. Além disso, questões éticas relacionadas à transparência e interpretabilidade são relevantes ao usar redes neurais em decisões críticas.

2.1.2.5 Regressão Logística

A regressão logística (LAVALLEY, 2008) é uma técnica estatística de aprendizado de máquina usada principalmente para resolver problemas de classificação binária, onde o objetivo é prever uma das duas classes possíveis. Diferente da regressão linear, que é usada para prever valores contínuos, a regressão logística prevê a probabilidade de um dado ponto pertencer a uma classe específica.

A regressão logística funciona aplicando a função logística, também conhecida como função sigmoide, à combinação linear dos atributos de entrada. A função sigmoide mapeia qual-

quer valor real para um intervalo entre 0 e 1, transformando assim a saída em uma probabilidade. Se a probabilidade prevista for maior que um determinado limiar (geralmente 0,5), o modelo atribui a classe positiva; caso contrário, atribui a classe negativa.

Matematicamente, a regressão logística é expressa como:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}$$

Aqui, $P(Y = 1|X)$ é a probabilidade de o resultado ser a classe 1 dado o vetor de atributos X, β_0 é o intercepto, e $\beta_1, \beta_2, \ldots, \beta_n$ são os coeficientes que precisam ser ajustados durante o treinamento do modelo. O objetivo do treinamento é encontrar os valores desses coeficientes que maximizem a função de verossimilhança, tornando as previsões do modelo o mais precisas possível.

É especialmente útil quando se lida com problemas em que a saída é categórica, mas é importante notar que ela só pode ser usada para problemas de classificação binária diretamente. Para problemas com mais de duas classes, a regressão logística pode ser estendida usando técnicas como a regressão logística multinomial ou estratégias como "um contra todos"(*One-vs-All*).

Embora a regressão logística seja fácil de implementar e interpretar, ela assume uma relação linear entre as variáveis independentes e o logit da variável dependente. Isso pode limitar sua eficácia em casos onde as relações entre as variáveis são mais complexas. No entanto, sua simplicidade e eficácia em resolver problemas de classificação binária tornam a regressão logística uma ferramenta fundamental em ciência de dados e estatística.

2.1.2.6 XGBoost

XGBoost, abreviação de *Extreme Gradient Boosting*, é uma poderosa biblioteca de aprendizado de máquina que implementa a técnica de *boosting* de gradiente, otimizada para desempenho, velocidade e eficiência de recursos. Ela é amplamente utilizada em competições de ciência de dados, como as do Kaggle, devido à sua capacidade de produzir modelos altamente precisos.

O XGBoost funciona construindo um conjunto de árvores de decisão, onde cada nova árvore tenta corrigir os erros cometidos pelas árvores anteriores. A ideia central do *boosting* é adicionar novos modelos de maneira sequencial, de modo que cada modelo adicione o máximo de ganho possível à função de custo. Diferente de outras implementações de *boosting*, o XGBoost incorpora várias otimizações, como o tratamento inteligente de valores ausentes, a paralelização da construção das árvores e a regularização L1 e L2, que ajuda a evitar o *overfitting*.

Matematicamente, o XGBoost minimiza a seguinte função de custo:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

onde l é a função de perda que mede a diferença entre a previsão $\hat{y}_i$ e o valor real y_i, e $\Omega(f_k)$ é um termo de regularização que penaliza a complexidade do modelo. Este termo de regularização ajuda a controlar o *overfitting*, garantindo que as árvores de decisão criadas sejam generalizáveis.

O XGBoost oferece suporte a várias funcionalidades, como:

- **Paralelização** durante a fase de construção das árvores, aumentando a eficiência computacional.
- **Amostragem por coluna** (*Column Subsampling*), que permite a construção de árvores usando apenas uma parte dos atributos disponíveis, ajudando na redução do *overfitting*.
- **Suporte a diferentes funções de perda**, como erro quadrático médio (MSE) para regressão e *log-loss* para classificação.
- **Regularização** L1 (Lasso) e L2 (Ridge) para controlar a complexidade do modelo.

O XGBoost é amplamente aplicado em várias áreas, incluindo finanças, marketing, medicina e até mesmo em bioinformática, devido à sua capacidade de lidar com dados complexos e produzir modelos precisos. No entanto, apesar de sua eficácia, a configuração dos hiperparâmetros do XGBoost pode ser desafiadora e requer experimentação cuidadosa para obter o melhor desempenho.

2.1.3 Engenharia de Atributos

Nos últimos tempos, quando pensamos em inteligência artificial, o foco tem sido predominantemente nos modelos que geram resultados, o que muitas vezes nos faz negligenciar a importância da manutenção adequada dos dados para o desempenho desses algoritmos.

Em (OZDEMIR, 2022), é ressaltado que grandes figuras da IA (Inteligência artificial) têm alertado os cientistas de dados sobre a necessidade de maior cuidado e atenção com os dados, incentivando uma abordagem mais centrada neles. Um exemplo disso pode ser visto na Figura 2.6.

Quando adotamos uma abordagem mais centrada nos dados, o foco deixa de ser a manipulação ou modificação do modelo em si. Em vez disso, concentramos nossos esforços em transformar os dados, permitindo que os algoritmos de aprendizado de máquina processem essas informações de forma mais eficaz, resultando em *insights* mais valiosos.

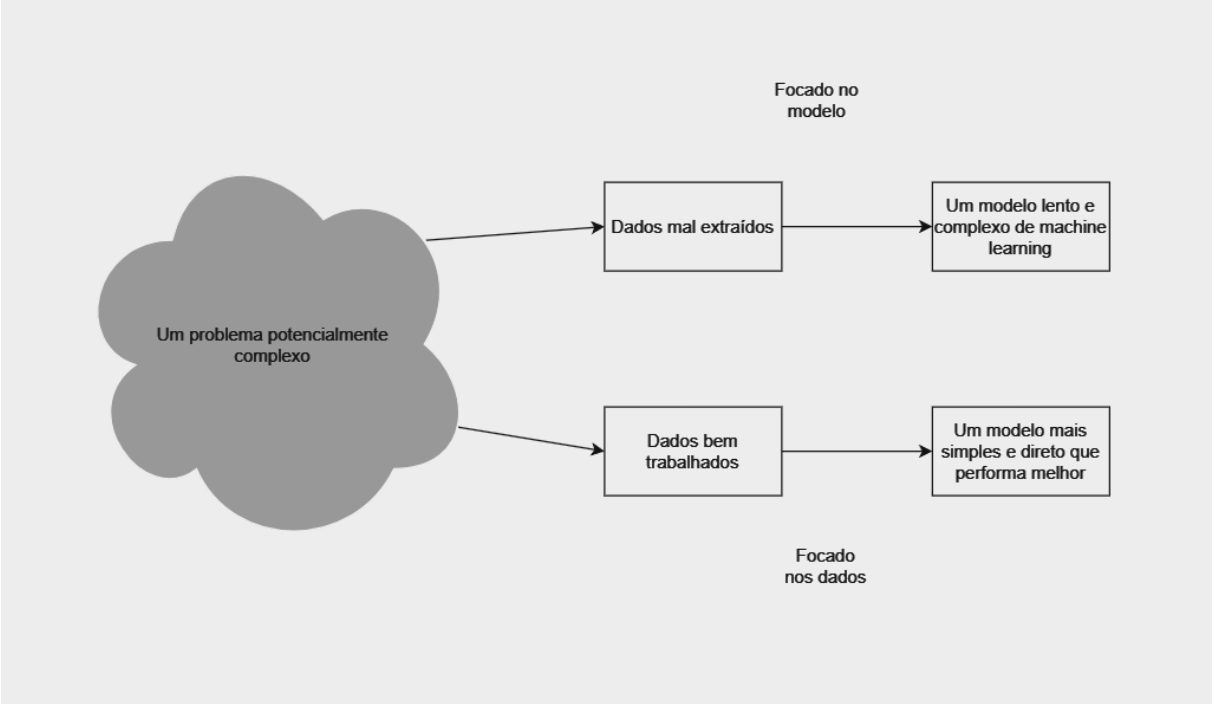

Figura 2.6 – Uma figura representando as diferenças entre uma visão mais modelo-centrada contra uma data-centrada, inspirado em (OZDEMIR, 2022)

Este campo que vai se preocupar mais em trabalhar com os dados em modelos de IA é chamado de engenharia de atributos (*Feature Engeniring*). O processo de *Feature Engineering* geralmente envolve as seguintes etapas descritas no livro (OZDEMIR, 2022):

1. **Compreensão de Atributos(*Feature Understanding*):** envolve a compreensão profunda dos dados e dos atributos disponíveis. Nesta etapa, é importante analisar a distribuição dos dados, identificar correlações entre os atributos e compreender a relevância de cada atributo para o problema em questão. Esse entendimento é fundamental para guiar as próximas etapas do processo.

2. **Estruturação de Atributos(*Feature Structuring*):** refere-se à criação e transformação de atributos. Novos atributos podem ser criados a partir dos dados existentes, utilizando operações matemáticas simples, como somas e produtos, ou técnicas mais avançadas, como transformações logarítmicas ou operações de agrupamento. Além disso, os atributos existentes podem ser transformados para melhor se adequar ao modelo, através de técnicas como normalização, padronização ou *one-hot encoding* para variáveis categóricas.

3. **Otimização de Atributos(*Feature Optimization*):** é a etapa em que se realiza a seleção de atributos. Nem todos os atributos disponíveis são necessariamente úteis para o modelo. A seleção de atributos envolve identificar e reter os atributos que mais contribuem para o desempenho do modelo, enquanto se eliminam os irrelevantes ou redundantes. Técnicas

como análise de correlação, seleção baseada em importância de atributos (*feature importance*), ou algoritmos de seleção automática como *Recursive Feature Elimination* (RFE) são comumente usadas nesta etapa.

4. **Avaliação de Atributos(*Feature Evaluation*):** é a última etapa, que envolve a avaliação do impacto dos atributos selecionados no desempenho do modelo. Isso pode incluir a análise do desempenho do modelo em um conjunto de validação e o ajuste de atributos com base nos resultados obtidos. O objetivo é garantir que os atributos selecionados realmente contribuam para a melhoria do modelo e que o modelo não esteja superajustado aos dados de treinamento.

Uma forma mais representativa e visual pode ser vista na Figura 2.7.

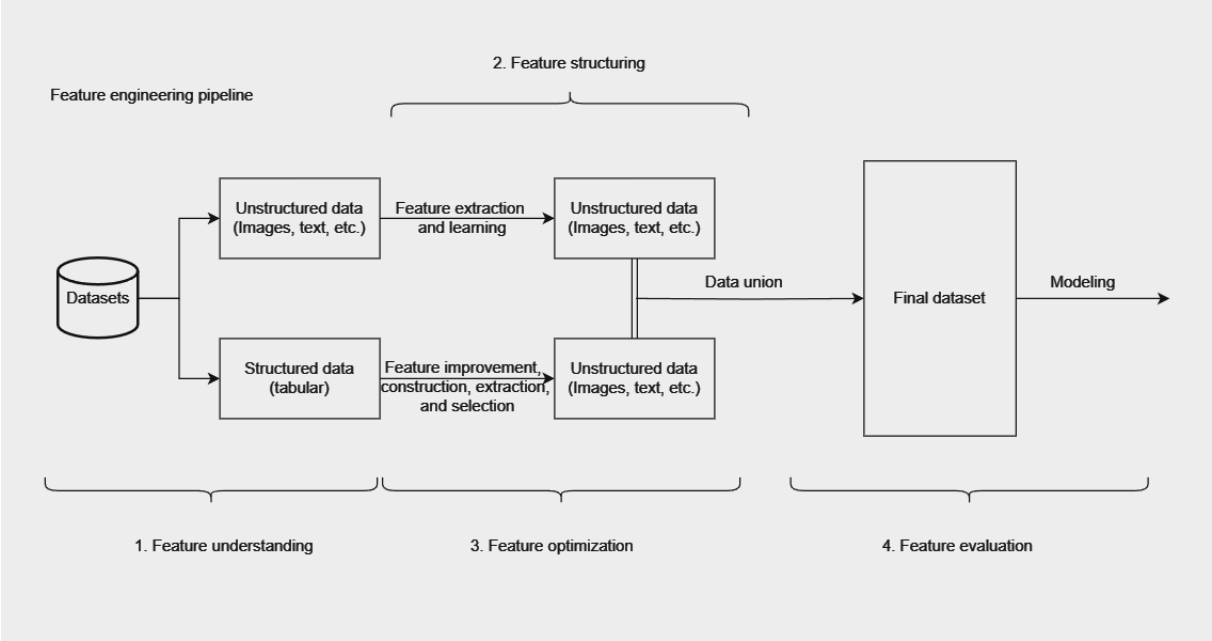

Figura 2.7 – Uma figura representando os processos que acontecem na engenharia de atributos, inpirado em (OZDEMIR, 2022)

O sucesso do *Feature Engineering* depende de um bom entendimento tanto dos dados quanto do domínio do problema. A escolha dos atributos certos pode simplificar a estrutura do modelo, melhorar a precisão das previsões e reduzir o tempo de treinamento.

Em alguns casos, o *Feature Engineering* pode ser realizado de forma automática utilizando técnicas de aprendizado de máquina, como em abordagens de *AutoML* (BOSCH et al., 2021). No entanto, o toque manual de especialistas em dados continua sendo insubstituível em muitas situações, especialmente quando se trata de dados complexos ou específicos de um domínio.

Com base no que foi discutido acima, é fundamental entender que a Engenharia de Atributos é um campo extenso, repleto de desafios, e que abrange diversas técnicas que, embora

façam parte do mesmo domínio, seguem caminhos distintos. Existem cinco principais tipos de *Feature Engineering* (OZDEMIR, 2022):

- **Criação de Atributos (*Feature Creation*):** Este tipo envolve a geração de novos atributos a partir dos dados existentes. A criação de atributos pode ser feita combinando, transformando ou decompondo atributos já disponíveis. Exemplos incluem a criação de interações entre variáveis, a extração de atributos temporais (como dia da semana ou mês) e a geração de novas características a partir de métodos matemáticos, como transformações logarítmicas.

- **Transformação de Atributos (*Feature Transformation*):** A transformação de atributos é essencial para preparar os dados de forma que o modelo de aprendizado de máquina possa interpretá-los corretamente. Isso pode incluir a normalização ou padronização dos dados, o *one-hot encoding* de variáveis categóricas, a aplicação de transformações polinomiais ou mesmo a aplicação de técnicas como PCA (Análise de Componentes Principais) para reduzir a dimensionalidade dos dados.

- **Seleção de Atributos (*Feature Selection*):** A seleção de atributos é o processo de identificar e manter apenas os atributos mais relevantes para o modelo. Este tipo de *Feature Engineering* ajuda a reduzir a complexidade do modelo, melhorar sua interpretabilidade e, muitas vezes, seu desempenho. Métodos comuns incluem a análise de correlação, técnicas baseadas em importância de atributos (como a importância dos atributos em florestas aleatórias), e métodos automatizados como *Recursive Feature Elimination* (RFE).

- **Imputação de Atributos (*Feature Imputation*):** Este tipo envolve lidar com dados faltantes, um problema comum em conjuntos de dados do mundo real. A imputação de atributos pode ser feita substituindo valores faltantes por estatísticas simples (como a média ou mediana), por métodos baseados em algoritmos (como *KNN Imputer*), ou até mesmo por técnicas avançadas como imputação múltipla, que considera a variabilidade dos dados.

- **Extração de Atributos (*Feature Extraction*):** A extração de atributos é o processo de derivar novos atributos de alto valor a partir de dados brutos ou complexos, como texto, imagens ou sinais. Métodos como TF-IDF (*Term Frequency-Inverse Document Frequency*) para dados textuais, a extração de bordas em imagens, ou a transformação de sinais em séries temporais são exemplos desse tipo de *Feature Engineering*.

Cada uma dessas técnicas de *Feature Engineering* tem seu lugar e aplicação dependendo do problema em questão e dos dados disponíveis. Um projeto bem-sucedido de aprendizado de máquina muitas vezes envolve uma combinação dessas abordagens para extrair o máximo de informações úteis dos dados. Embora algumas etapas possam ser automatizadas, a intuição e o conhecimento do domínio desempenham um papel vital na escolha e aplicação adequadas dessas técnicas, destacando a importância da expertise humana no processo.

2.1.4 Possíveis Métricas de Avaliação de Atributos

As métricas de avaliação desempenham um papel crucial na análise do desempenho dos algoritmos de aprendizado de máquina. Elas fornecem insights sobre a eficácia de um modelo na realização de tarefas específicas, como classificação e regressão. A escolha das métricas apropriadas depende do problema em questão, do tipo de dados utilizados e dos objetivos da aplicação.

Métricas boas ajudam os cientistas de dados e engenheiros a quantificar o desempenho dos modelos. Elas permitem comparar diferentes algoritmos, identificar áreas de melhoria e otimizar os parâmetros do modelo. Além disso, as métricas oferecem uma maneira objetiva de avaliar como um modelo se comporta em cenários do mundo real, em vez de depender apenas de suposições ou intuições.

Um dos desafios na escolha das métricas é que algumas podem não refletir adequadamente a qualidade do modelo em determinados contextos. Por exemplo, em problemas de classificação onde uma das classes é muito menos frequente, a acurácia pode ser alta mesmo que o modelo não esteja performando bem para a classe minoritária. Portanto, é essencial considerar o contexto do problema e as implicações de diferentes métricas.

Além disso, as métricas podem ser influenciadas por fatores como o desbalanceamento dos dados e a escolha dos limiares de decisão. Por isso, é comum utilizar várias métricas em conjunto para obter uma avaliação mais robusta do desempenho do modelo.

2.1.4.1 Índice de Gini

O Índice de Gini (MEDINA, 2011) é uma métrica amplamente utilizada em algoritmos de aprendizado de máquina, especialmente em árvores de decisão, para medir a impureza ou a desigualdade de uma amostra de dados. O principal objetivo ao utilizar o Índice de Gini é identificar a melhor divisão dos dados, resultando em subconjuntos mais puros em relação à classe-alvo.

O Índice de Gini é definido pela seguinte fórmula:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2$$

onde:

- p_i representa a proporção de amostras da classe i em relação ao total de amostras.
- C é o número total de classes.

Interpretando os valores do Gini:

- **Gini = 0**: O nó é puro, ou seja, todas as instâncias pertencem à mesma classe.
- **Gini > 0**: O nó contém instâncias de diferentes classes, e quanto maior o Índice de Gini, maior a impureza do nó.

Durante a construção da árvore de decisão, o algoritmo avalia diversas divisões possíveis dos dados com base em diferentes atributos. Para cada divisão, o Índice de Gini é calculado para os subconjuntos resultantes. O objetivo é escolher a divisão que minimiza o Índice de Gini dos subconjuntos, resultando em maior pureza.

Um Índice de Gini menor indica que os subconjuntos resultantes são mais puros em relação à classe-alvo. Portanto, durante o treinamento da árvore de decisão, o algoritmo busca divisões que levam a índices de Gini mais baixos, resultando em uma árvore mais eficaz na classificação de novos dados.

2.1.4.2 Índice de Fisher

O Índice de Fisher (SANTOS, 2007) é uma métrica amplamente utilizada em estatísticas e aprendizado de máquina, principalmente em problemas de classificação, como na Análise Discriminante de Fisher. Seu objetivo é avaliar o grau de separação entre diferentes classes em um conjunto de dados, ajudando a identificar características que melhor discriminam essas classes.

A fórmula do Fisher Score para a k-ésima característica é, retirado de (GAN; ZHANG, 2021) :

$$F(k) = \frac{\sum_{i=1}^{c} n_i(\mu_i^k - \mu_k)^2}{\sum_{i=1}^{c} n_i(\sigma_i^k)^2}$$

onde:

- n_i é o número de amostras na i-ésima classe.
- μ_i^k é a média da k-ésima característica na i-ésima classe.
- μ_k é a média global da k-ésima característica em todas as classes.
- σ_i^k é a variância da k-ésima característica na i-ésima classe.

No numerador, $\sum_{i=1}^{c} n_i(\mu_i^k - \mu_k)^2$ representa a **dispersão entre as classes**, que mede o quão distantes as médias das classes estão em relação à média global. No denominador, $\sum_{i=1}^{c} n_i(\sigma_i^k)^2$ representa a **dispersão dentro das classes**, ou seja, a variabilidade dos dados dentro de cada classe.

A interpretação do Índice de Fisher envolve os seguintes pontos principais:

1. **Separação das Classes**: Um valor alto do Índice de Fisher indica uma grande separação entre as classes, ou seja, a variância entre as classes é maior do que a variância dentro das classes, o que facilita a discriminação entre elas.

2. **Identificação de Características Relevantes**: Ao calcular o Índice de Fisher para diferentes características, é possível determinar quais atributos são mais eficazes para discriminar entre as classes, auxiliando na seleção de variáveis relevantes para modelos de aprendizado de máquina.

3. **Otimização de Modelos**: O Índice de Fisher é frequentemente usado como critério de otimização em algoritmos de redução de dimensionalidade e seleção de características, permitindo a construção de modelos mais eficientes e com melhor desempenho. Ele é utilizado, por exemplo, na Análise Discriminante Linear (LDA), onde buscamos maximizar a separação entre classes em um espaço de menor dimensão.

2.1.4.3 F-Ratio

O F-Ratio (WITSIL et al., 2022) é uma métrica estatística utilizada para comparar a variabilidade entre diferentes grupos em relação à variabilidade dentro dos grupos. É frequentemente utilizado em testes de hipóteses, especialmente na análise de variância (ANOVA), para determinar se existem diferenças significativas entre as médias de várias populações.

A fórmula para o F-Ratio é definida como:

$$F = \frac{S_B}{S_W}$$

onde:

- S_B é a variância entre as classes, representando o quão distantes as médias das classes estão umas das outras. A fórmula é dada por:

$$S_B = \sum_{i=1}^{C} n_i(\mu_i - \mu)^2$$

onde n_i é o número de amostras na classe i, μ_i é a média da classe i, e μ é a média global.

- S_W é a variância dentro das classes, medindo a dispersão das amostras em torno da média de suas respectivas classes. A fórmula é:

$$S_W = \sum_{i=1}^{C} \sum_{x \in C_i} (x - \mu_i)^2$$

onde x são os dados da classe i.

A interpretação do F-Ratio envolve os seguintes pontos principais:

1. **Comparação de Variâncias**: Um valor alto do F-Ratio indica que a variabilidade entre as médias dos grupos é maior do que a variabilidade dentro dos grupos, sugerindo a presença de diferenças significativas entre as populações.

2. **Testes de Hipóteses**: O F-Ratio é utilizado para testar a hipótese nula de que todas as médias dos grupos são iguais. Um valor de F maior do que um determinado valor crítico (obtido a partir da distribuição F) leva à rejeição da hipótese nula, indicando que pelo menos uma média é significativamente diferente.

3. **Aplicações em Modelos Estatísticos**: O F-Ratio é amplamente utilizado em diversos modelos estatísticos, incluindo ANOVA de um fator, ANOVA multifatorial e regressão linear, para avaliar a qualidade do ajuste do modelo e a significância dos efeitos dos fatores.

2.1.4.4 Correlação de Spearman

A correlação de Spearman (GUIMARÃES, 2017) é uma medida não paramétrica que avalia a força e a direção da associação entre duas variáveis. Ao contrário da correlação de Pearson, que assume que as variáveis são normalmente distribuídas e mede a relação linear, a correlação de Spearman avalia a relação monotônica entre as variáveis, sem fazer suposições sobre a distribuição dos dados.

A correlação de Spearman é calculada classificando as observações em vez de usar os valores reais. Cada valor é substituído por sua posição na ordem dos dados, e, em seguida, a correlação é calculada com base nessas classificações. O coeficiente de correlação de Spearman, denotado como ρ (rho), varia de -1 a +1, onde:

- $\rho = +1$ indica uma correlação perfeita positiva, ou seja, à medida que uma variável aumenta, a outra também aumenta.
- $\rho = -1$ indica uma correlação perfeita negativa, ou seja, à medida que uma variável aumenta, a outra diminui.
- $\rho = 0$ sugere que não há correlação entre as variáveis.

A fórmula para calcular o coeficiente de correlação de Spearman é a seguinte:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

onde d_i é a diferença entre as classes de cada par de observações e n é o número total de observações.

A correlação de Spearman é especialmente útil em situações onde os dados podem não seguir uma distribuição normal ou onde a relação entre as variáveis não é linear. Além disso,

ela pode ser aplicada em dados ordinais, permitindo que pesquisadores analisem relações entre variáveis categóricas que possuem uma ordem definida.

2.1.4.5 Correlação de Pearson

A correlação de Pearson (FILHO; JÚNIOR, 2009) é uma medida estatística que quantifica a força e a direção da relação linear entre duas variáveis contínuas. Essa métrica é amplamente utilizada em diversas áreas, incluindo ciências sociais, biologia, economia e engenharia, para entender como as variáveis estão relacionadas entre si.

O coeficiente de correlação de Pearson, denotado como r, varia de -1 a +1, onde:

- $r = +1$ indica uma correlação perfeita positiva, significando que, à medida que uma variável aumenta, a outra também aumenta de maneira proporcional.
- $r = -1$ indica uma correlação perfeita negativa, significando que, à medida que uma variável aumenta, a outra diminui de maneira proporcional.
- $r = 0$ sugere que não há correlação linear entre as variáveis.

A fórmula para calcular o coeficiente de correlação de Pearson retirada de (BENESTY et al., 2009) é dada por:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

onde x_i e y_i representam os valores das variáveis, e $\bar{x}$ e $\bar{y}$ são as médias das respectivas variáveis.

A correlação de Pearson assume que os dados seguem uma distribuição normal e mede apenas a relação linear entre as variáveis. Isso significa que, mesmo que duas variáveis estejam relacionadas, se essa relação não for linear, o coeficiente de correlação de Pearson pode não refletir adequadamente a força da relação. Por esse motivo, é importante visualmente inspecionar os dados por meio de gráficos de dispersão para verificar a linearidade antes de confiar no valor do coeficiente.

Além disso, a correlação de Pearson é sensível a valores extremos (outliers), que podem distorcer a interpretação do coeficiente. Portanto, ao usar essa métrica, é essencial considerar a presença de outliers e, se necessário, aplicar técnicas de transformação ou remoção desses pontos.

2.1.4.6 Análise de Componentes Principais (PCA)

A Análise de Componentes Principais (*Principal Component Analysis* – PCA) é uma técnica estatística amplamente utilizada para redução de dimensionalidade (ARAUJO; COELHO,

2009) (WOLD; ESBENSEN; GELADI, 1987), sendo aplicada em problemas de aprendizado de máquina e análise exploratória de dados. O PCA transforma um conjunto de variáveis possivelmente correlacionadas em um novo conjunto de variáveis não correlacionadas, chamadas de componentes principais, ordenadas de acordo com a quantidade de variância que explicam nos dados originais.

O principal objetivo do PCA é encontrar uma representação mais compacta dos dados, preservando a maior quantidade possível de informação. Isso é feito através da decomposição da matriz de covariância dos dados e da projeção das amostras em um novo espaço de menor dimensão. Esse processo ajuda a mitigar o efeito da multicolinearidade entre as variáveis e pode melhorar o desempenho de modelos de aprendizado de máquina ao remover redundâncias nos dados.

Matematicamente, o PCA encontra uma matriz de componentes principais W a partir da decomposição em valores singulares (*Singular Value Decomposition* – SVD) ou da decomposição espectral da matriz de covariância Σ, tal que:

$$Z = XW$$

onde X representa a matriz original de dados, W é a matriz cujas colunas são os autovetores correspondentes aos maiores autovalores de Σ, e Z representa os dados transformados no novo espaço de componentes principais.

O PCA possui diversas aplicações em ciência de dados, incluindo:

- **Redução de dimensionalidade**: ao reduzir o número de variáveis explicativas sem perder muita informação, facilitando a visualização e o processamento dos dados.
- **Pré-processamento para aprendizado de máquina**: ajudando a remover redundâncias e melhorar o desempenho de algoritmos sensíveis a multicolinearidade.
- **Compressão de dados**: reduzindo o custo computacional ao armazenar apenas as componentes principais mais significativas.
- **Remoção de ruído**: filtrando componentes menos relevantes que podem estar associados a variações aleatórias nos dados.

Apesar de sua eficiência, o PCA assume que a variância captura a informação mais relevante nos dados, o que pode não ser adequado em alguns casos. Além disso, a interpretação dos componentes principais pode ser desafiadora, uma vez que são combinações lineares das variáveis originais.

Para avaliar a relevância do novo atributo gerado, utilizamos a Análise de Componentes Principais (PCA). O objetivo é verificar o impacto desse atributo na variabilidade dos dados, analisando sua contribuição para os componentes principais.

Primeiramente, aplicamos o PCA aos dados transformados, ajustando um modelo sobre a matriz de atributos expandidos X', que inclui o novo atributo. A decomposição do PCA retorna:

- $\mathbf{W}$ – Matriz de autovetores (componentes principais), onde cada coluna representa uma combinação linear dos atributos originais.
- λ – Vetor de autovalores normalizados, representando a variância explicada por cada componente principal.

A importância do novo atributo é então calculada por meio da seguinte equação:

$$I_{\text{novo}} = |\mathbf{w}_{\text{novo}}| \cdot \lambda$$

onde $\mathbf{w}_{\text{novo}}$ corresponde à última linha da matriz de componentes principais $\mathbf{W}$, que indica a contribuição do novo atributo em cada componente principal, e λ representa a variância explicada associada a cada componente.

Esse método permite quantificar a influência do novo atributo em termos de sua contribuição para a variância total dos dados. Um valor alto de I_{novo} sugere que o novo atributo introduz informações relevantes, enquanto um valor baixo indica que ele pode ser redundante ou ter pouca importância estatística.

2.2 Trabalhos Relacionados

2.2.1 Métricas utilizadas e sua superioridade

Inteligência artificial e algoritmos de otimização estão intrinsecamente ligados, pois ambos pertencem a campos que compartilham princípios fundamentais, principalmente no que tange à busca de soluções eficientes para problemas complexos. Um aspecto central que une esses dois domínios é o uso de métricas que orientam os modelos e algoritmos durante o processo de aprendizado ou de tomada de decisões. As métricas são essenciais para avaliar o desempenho, ajustar parâmetros, e garantir que os modelos estejam caminhando em direção a uma solução ótima.

Na literatura científica, destacam-se algumas métricas amplamente utilizadas, como o coeficiente de correlação de Pearson e a razão f (*f-ratio*), que são frequentemente aplicadas em uma variedade de contextos. O coeficiente de Pearson mede a correlação linear entre duas variáveis, sendo particularmente útil quando se busca entender a relação direta entre características

numéricas em um conjunto de dados ((BENESTY et al., 2009); (RODGERS; NICEWANDER, 1988)). A razão f, por sua vez, é fundamental em testes estatísticos, especialmente na análise de variância (ANOVA), onde é usada para determinar se existem diferenças significativas entre grupos ((ELSSIED; IBRAHIM; OSMAN, 2014)(ST; WOLD et al., 1989); (KHLAIEF et al., 2019)).

Além dessas métricas principais, é igualmente relevante considerar métricas derivadas ou relacionadas, que podem oferecer insights complementares. Um exemplo é a correlação de Spearman, que, embora esteja relacionada ao coeficiente de Pearson, é mais apropriada quando se trabalha com dados não paramétricos ou quando se busca capturar relações monotônicas, em vez de lineares ((FITNI; RAMLI, 2020); (WANG et al., 2024)). Já a métrica de Fisher, vinculada à razão f, desempenha um papel importante na análise de variância multivariada, fornecendo uma visão mais ampla sobre a distribuição e variabilidade dos dados ((RUSTAM; HIDAYAT et al., 2019); (AKSU et al., 2018)).

Outra métrica comumente explorada é o índice de Gini, amplamente conhecido por sua aplicação em problemas de desigualdade de renda (MEDINA, 2011), mas também utilizado em *machine learning* (SUGUIURA, 2022), especialmente em algoritmos de árvores de decisão. Apesar de ser considerada menos poderosa ou complexa em relação a métricas como Pearson ou f-ratio, o índice de Gini tem a vantagem de ser facilmente interpretável, o que o torna uma excelente métrica de controle em muitos cenários. Sua simplicidade permite uma análise rápida e intuitiva, servindo como uma referência sólida, especialmente quando o objetivo é balancear precisão com facilidade de entendimento.

Quando analisamos a superioridade dessas métricas, é importante reconhecer que cada caso apresenta suas próprias particularidades, o que pode influenciar diretamente a escolha da métrica mais adequada. A forma como o problema é estruturado ou a natureza dos dados pode fazer com que uma métrica se destaque em relação às outras.

Por exemplo, o coeficiente de correlação de Pearson é altamente eficaz quando estamos lidando com variáveis que têm uma relação linear (FILHO; JÚNIOR, 2009), sendo amplamente utilizado em situações onde a linearidade é uma suposição razoável, como em problemas de regressão simples. No entanto, quando os dados não seguem uma distribuição normal ou apresentam relações não lineares, a correlação de Spearman se torna mais adequada, já que esta métrica captura associações monotônicas, sendo ideal para cenários onde a ordem dos dados é mais importante do que a magnitude das diferenças entre eles (SOUSA, 2019).

A razão f (*f-ratio*), por sua vez, se destaca em análises de variância (ANOVA) (CONNELLY, 2021), sendo utilizada principalmente para comparar médias entre múltiplos grupos e identificar se as diferenças observadas são estatisticamente significativas. Isso a torna uma escolha superior em experimentos controlados ou estudos com múltiplas variáveis, como testes de eficácia de diferentes tratamentos. Já a métrica de Fisher, relacionada, é preferida em cenários onde a análise multivariada é necessária, especialmente quando se está lidando com a análise

conjunta de várias variáveis dependentes (GU; LI; HAN, 2012).

O índice de Gini, amplamente utilizado em *machine learning*, brilha quando aplicado em problemas de classificação, especialmente em algoritmos de árvores de decisão, como o CART (*Classification and Regression Trees*). Sua simplicidade e facilidade de interpretação o tornam ideal para problemas que exigem resultados rápidos e compreensíveis, mesmo que ele não capture nuances tão complexas quanto outras métricas mais elaboradas.

A escolha da métrica ideal sempre dependerá do contexto e do tipo de problema a ser resolvido. Não há uma métrica que seja universalmente superior; cada uma delas tem seu ponto forte e, aplicada ao problema certo, pode fornecer resultados mais precisos e significativos. Assim, para cada caso específico, a métrica mais apropriada brilhará de maneira única, oferecendo a melhor forma de avaliar o desempenho e os resultados.

2.2.2 Ideias e Referências na Literatura

Ao realizar a pesquisa para embasar os objetivos propostos neste trabalho, foram identificados diversos estudos que ressaltam a importância das métricas e seu impacto direto no desempenho de modelos de *machine learning*. Um exemplo é o trabalho de (NEMBRINI; KÖNIG; WRIGHT, 2018), que discute o "*revival*"das técnicas de otimização de métricas para melhorar a precisão dos modelos. Os autores argumentam que a escolha adequada de métricas não apenas refina o desempenho dos modelos, mas também assegura que os resultados sejam consistentes e aplicáveis em contextos reais, fortalecendo a base para a tomada de decisões em sistemas automatizados. Isso sublinha a necessidade de uma abordagem meticulosa ao selecionar indicadores de avaliação, especialmente em cenários onde a robustez dos resultados é crucial.

Além disso, o estudo de (NIU, 2020) exemplifica como os algoritmos de aprendizado de máquina podem ser aplicados em situações do mundo real para resolver problemas complexos, especificamente no contexto de redes de supermercado. O trabalho demonstra como a integração de técnicas de engenharia de software com metodologias de *machine learning* pode otimizar processos internos, aumentar a eficiência e criar soluções. Este estudo é uma demonstração clara da importância de se alinhar o aprendizado de máquina com princípios sólidos de *feature engineering*, ressaltando os benefícios da colaboração entre essas duas áreas.

Assim, tanto o trabalho de (NEMBRINI; KÖNIG; WRIGHT, 2018) quanto o de (NIU, 2020) reforçam a ideia de que a combinação de métricas otimizadas com algoritmos de aprendizado eficazes pode gerar soluções robustas e eficientes. Essas pesquisas fornecem uma base sólida para a aplicação das abordagens discutidas neste trabalho, oferecendo exemplos práticos de metodologias que podem ser replicadas ou adaptadas conforme necessário. Essa integração entre métricas bem definidas e modelos eficientes é essencial para alcançar melhores resultados em diferentes tipos de problemas.

Por outro lado, uma revisão mais ampla da literatura revelou que muitos autores tendem

a concentrar-se mais nos algoritmos prontos, ao invés de nas métricas utilizadas para avaliar esses modelos, o que dificulta uma avaliação mais detalhada e comparativa sobre qual métrica oferece um desempenho superior de forma consistente (SHWARTZ-ZIV; ARMON, 2022). Esse foco desproporcional nos algoritmos, em detrimento das métricas, limita o entendimento da real eficácia dos modelos.

Visando preencher essa lacuna, o presente trabalho propõe uma abordagem detalhada que parte da identificação do tipo de problema presente em um banco de dados específico. A partir dessa análise inicial, será possível avaliar quais métricas exercem maior influência nos resultados obtidos e, consequentemente, determinar a métrica mais adequada para cada tipo de problema. Essa estratégia permitirá uma compreensão mais clara da relação entre a métrica escolhida e o desempenho final do modelo, facilitando decisões mais fundamentadas e eficazes.

Essa proposta contribui para o campo ao trazer uma nova perspectiva sobre a importância das métricas no aprendizado de máquina, além de auxiliar pesquisadores e desenvolvedores a selecionar as métricas mais apropriadas para seus problemas específicos, otimizando o desempenho dos modelos e promovendo resultados mais robustos e consistentes.

3 Materiais e Métodos

3.1 Gerador de Atributos

Para realizar os testes mencionados no capítulo anterior, foi necessário utilizar algoritmos capazes de produzir características (*features*) de forma aleatória. Para isso, utilizamos duas abordagens principais: uma rede neural do tipo MLP (*Multilayer Perceptron*) e um gerador baseado em árvores de decisão. Ambas as abordagens têm como objetivo criar atributos processando os dados de entrada. A rede MLP introduz transformações não lineares, enquanto a árvore de decisão particiona os dados de forma hierárquica, destacando relações entre os atributos.

A seguir, descrevemos os algoritmos utilizados para essa tarefa.

3.1.1 Geração de Atributos com MLP

O algoritmo de geração da MLP cria uma rede com um número variável de camadas, onde o tamanho de cada camada também é gerado aleatoriamente. O objetivo principal é gerar uma arquitetura que produza saídas com novas características a partir dos dados de entrada fornecidos. A rede é executada utilizando o otimizador Adam e a função de perda do erro quadrático médio (MSE).

O algoritmo segue os seguintes passos:

1. Definir o número de camadas ocultas da rede (num_layers).
2. Gerar o tamanho de cada camada oculta de forma aleatória.
3. Construir a rede MLP:
    - A primeira camada recebe a dimensão dos dados de entrada.
    - Cada camada oculta subsequente é definida com a função de ativação ReLU.
    - A última camada é a camada de saída, que não possui função de ativação (linear).
4. Compilar o modelo com o otimizador Adam e a função de perda MSE.

3.1.2 Geração de Atributos com Árvores de Decisão

A segunda abordagem utilizada para a geração de atributos aleatórios consiste no uso de árvores de decisão. As árvores de decisão podem identificar padrões estruturais nos dados ao realizar divisões hierárquicas baseadas em critérios como ganho de informação ou redução de impureza. Isso pode resultar em atributos que capturam interações entre as variáveis.

O processo de geração de atributos com árvores de decisão segue os seguintes passos:

1. Criar uma árvore de decisão com profundidade, critério de divisão e número mínimo de amostras aleatórios.
2. Treinar a árvore com os dados de entrada e o atributo alvo.
3. Utilizar a previsão da árvore como um novo atributo derivado.

O algoritmo utilizado está descrito em 1.

Algoritmo 1 Geração de atributos com árvores de decisão

```
def generate_tree_features(df, target, is_classifier=True):
    tree = generate_random_decision_tree(is_classifier)
    tree.fit(df['data'], target)
    new_feature = tree.predict(df['data']).reshape(-1, 1)
    return new_feature
```

3.1.3 Função de Avaliação

Uma vez que os geradores de atributos criam novas características, é necessário avaliar sua qualidade. Para isso, utilizamos uma função de *fitness* que aplica métricas selecionadas sobre os novos atributos gerados.

O processo de avaliação segue os seguintes passos:

1. Gerar um novo atributo utilizando a MLP ou árvore de decisão.
2. Aplicar as métricas de avaliação.
3. Retornar os escores calculados para cada métrica.

O algoritmo correspondente é apresentado em 2.

Algoritmo 2 Avaliação de métrica para features geradas

```
def fitness_function(df, target, metrics, method='mlp'):
    if method == 'mlp':
        new_feature = generate_mlp(input_dim=df['data'].shape[1]).predict(df['data'])
    else:
        new_feature = generate_tree_features(df, target)
    metric_scores = []
    for metric in metrics:
        metric_score = compute_metrics(new_feature.flatten(), new_feature, target, metric)
        metric_scores.append(metric_score)
    return metric_scores, new_feature
```

Com isso, temos duas abordagens para gerar atributos aleatórios: MLP e árvores de decisão, permitindo analisar a efetividade das métricas sobre diferentes tipos de transformação dos dados.

3.2 Métricas de Avaliação

As métricas de avaliação que foram utilizadas são :

1. Índice de Gini
2. Fisher Score
3. F-Ratio
4. Correlação de Pearson
5. Correlação de Spearman
6. P-Valor derivado de Pearson
7. PCA

3.3 Projeto Experimental

3.3.1 Bases de dados

Os bancos de dados selecionados para realizar esse trabalho foram escolhidos de maneira a diversificar as situações em que as métricas seriam analisadas. Todos são de problemas de classificação e seus atributos *target* (o atributo que desejamos classificar) são binários, afim de simplificar a interpretabilidade dos resultados a serem alcançados.

Esses bancos de dados podem ser encontrados e ter suas origens identificadas no código disponível em [1] no arquivo *clean-metric-analyzer.ipynb*.

1. QSAR Biodegradation Data Set
2. Heart Disease Classification Dataset
3. Breast Cancer Wisconsin (Diagnostic)
4. Pima Indians Diabetes Database
5. Statlog (German Credit Data)

[1] Disponível em<https://github.com/henriquedpighini/Testes-Metricas>

Banco de dados	N.º de Instâncias	N.º Atributos
QSAR	1055	41
Heart-Disease	303	13
Breast-Cancer	569	30
Diabetes-DB	768	8
Credit-Data	1000	20
Tic-Tac-Toe	958	10
Climate-Crashes	540	21

Tabela 3.1 – Bancos de dados e suas características

6. Tic-Tac-Toe Endgame database

7. Climate Model Simulation Crashes

3.3.2 Algoritmo de teste

Com as bases de dados, métricas e modelos de aprendizado definidos, foi desenvolvido o algoritmo que vai realizar os testes 3.

Algoritmo 3 Algoritmo de teste desenvolvido

```
1: Para cada Banco de dados
2:     Gerar atributos usando MLP
3:     Para cada Métrica
4:         Para cada Modelo
5:             Registrar os valores de F1-Score e da Métrica
6: Exibir esses resultados para que a análise seja feita
```

4 Resultados

Esta seção apresenta os resultados do experimento estruturado no capítulo anterior. A saída do algoritmo consiste em arquivos .csv que, para cada *feature* gerada, contêm os valores das métricas correspondentes e os *F1-Scores* de cada modelo testado.

Um problema encontrado foi simplificar a visualização dos dados obtidos graças a grande quantidade de informações que o algoritmo retornou. Para solucionar esse problema, é necessário calcular a correlação do valor da métrica com o valor do *F1-Score* que foi retornado, já que dessa forma é possível eliminar os casos em que a métrica não foi relevante para o problema. Com isso, utilizamos a Correlação de Pearson para simplificar os casos em que a métrica foi impactante.

Um exemplo de como isso é aplicado pode ser observado nas Figuras 4.1 e 4.2. A Figura 4.1 mostra uma correlação indicando o impacto da métrica, enquanto a Figura 4.2 apresenta um caso em que a métrica não teve influência significativa.

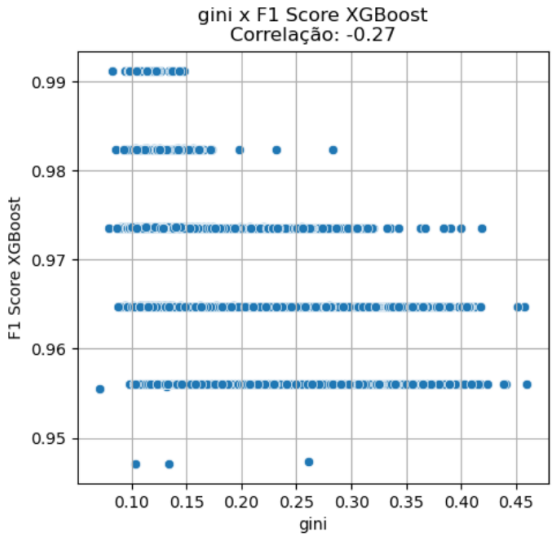

Figura 4.1 – Exemplo de um caso onde é possível ver uma correlação entre os valores da métrica Gini e o *F1-Score* do modelo

Assim, para focar em resultados que podem gerar *insights* significativos, vamos filtrar esses gráficos e realizar a análise desses resultados mostrando as reações que cada base de dados teve.

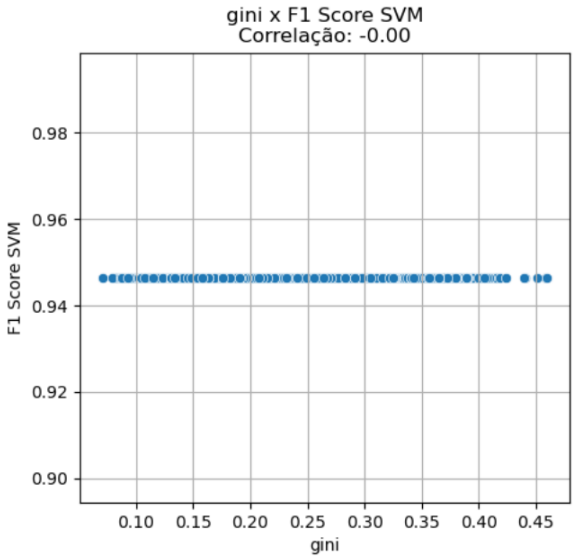

Figura 4.2 – Exemplo de um caso onde é possível que a alteração do valor da métrica não impacta no desempenho do modelo

4.1 Testes usando MLP como gerador

Os testes foram divididos em dois grupos: um utilizando uma Rede Neural Multicamadas (*Multi-Layer Perceptron* – MLP) como gerador de atributos e outro utilizando uma Árvore de Decisão. No primeiro grupo, foram aplicadas todas as métricas de avaliação definidas, enquanto no segundo, a avaliação foi realizada apenas no estratégia usando PCA.

Para facilitar a leitura das figuras, utilizamos as seguintes abreviações ao longo deste trabalho: DT refere-se a *Decision Tree* (Árvore de Decisão); NN, a *Neural Network* (Rede Neural); RF, a *Random Forest*; e RL, a *Regressão Logística*.

4.1.1 Breast Cancer Wisconsin (Diagnostic)

Os resultados gerados podem ser vistos na Tabela 4.1.

X	DT	NN	RF	SVM	RL	XGBoost
Gini	0.10	-0.03	-0.03	0	0.01	-0.27
Fisher	-0.16	0.08	0.01	0	-0.01	0.03
F-Ratio	-0.16	0.08	0.02	0	-0.01	0.05
Pearson	0.01	0.15	0	0	0	0
Spearman	0.01	0.15	0	0	0	0
P-Valor	0.03	-0.03	0	0	0	-0.02
PCA	-0.01	0.01	0	0	-0.07	0.01

Tabela 4.1 – Resultados das correlações no banco de dados Breast Cancer Wisconsin (Diagnostic).

Primeiramente é possível identificar que o SVM não sofreu nenhuma influência das *features* e os gráficos de resultado se assemelham muito com os dados da Figura 4.2. Isso pode

indicar que o modelo pode não ser apropriado para os testes que foram realizados.

Podemos ver impacto nos seguintes casos:

1. Gini, Fisher e *F-Ratio* em modelos de Árvore de Decisão;
2. Pearson e Spearman em Redes Neurais;
3. Gini em XGBoost;

O caso mais visivel é na métrica Gini com o modelo XGBoost que foi mostrado na Figura 4.1, quanto mais próximo de 0 o Gini está, melhor os resultados de *F1-Score* do XGBoost tendem a ser. Já nos outros casos é possível ver uma relação, porém ela não é tão expressiva, um exemplo disso pode ser visto na Figura 4.3.

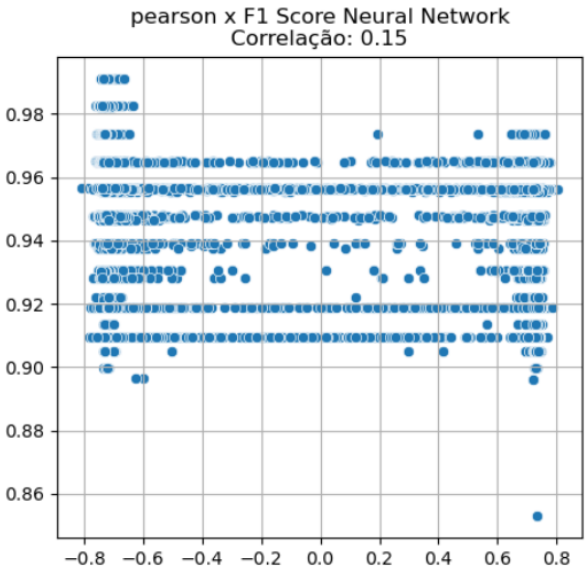

Figura 4.3 – Exemplo de um caso onde é possível que a alteração do valor da métrica não impacta no desempenho do modelo

4.1.2 Climate Model Simulation Crashes

Os resultados gerados podem ser vistos na Tabela 4.2.

X	DT	NN	RF	SVM	RL	XGBoost
Gini	-0.20	0	0.02	0	0	-0.04
Fisher	0.04	0.01	0.02	0	0	0.01
F-Ratio	0	0.01	0.02	0	0	0.01
Pearson	-0.03	0.02	0.01	0	0	0.01
Spearman	-0.03	0.02	0.01	0	0	0.01
P-Valor	-0.02	0	0	0	0	-0.01
PCA	-0.01	0.01	0	0	-0.01	0.01

Tabela 4.2 – Resultados das correlações no banco de dados Climate Model Simulation Crashes.

Este banco de dados se demonstrou mais resistente aos impactos das novas *features* e somente um caso mostrou uma correlação maior:

1. Gini em modelos de árvore de decisão como na Figura (4.4);

O interessante no gráfico da Figura 4.4 é que os valores vão melhorando quando o Gini vai se aproximando de 0, porém, depois de um limiar de 0.15, os valores se tornam mais escassos e se observa uma queda no desempenho dos modelos.

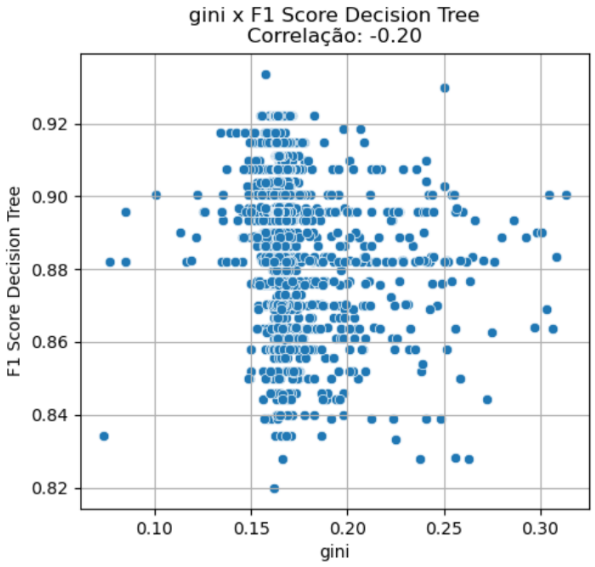

Figura 4.4 – Resultado usando a métrica Gini no modelo de árvore de decisão no banco Climate Model Simulation Crashes

4.1.3 Statlog (German Credit Data)

Os resultados gerados podem ser vistos na Tabela 4.3.

X	DT	NN	RF	SVM	RL	XGBoost
Gini	-0.10	-0.01	-0.06	0	-0.01	0.04
Fisher	0	0.01	-0.08	0	0.01	0.03
F-Ratio	0	0.01	-0.09	0	-0.01	0.03
Pearson	0.15	0.01	0.10	0	0.02	0.01
Spearman	0.14	0.01	0.10	0	0.02	0.01
P-Valor	-0.02	0	-0.01	0	0	-0.02
PCA	-0.02	0.01	0	0	-0.01	-0.02

Tabela 4.3 – Resultados das correlações no banco de dados Climate Model Simulation Crashes.

Nessa base existem casos em que a correlação possuiu um valor diferente de 0, mas que não foram significativos para serem analisados. Em resumo esse banco não demonstrou uma boa reação às *features* que foram geradas.

4.1.4 Pima Indians Diabetes Database

Os resultados gerados podem ser vistos na Tabela 4.4.

X	DT	NN	RF	SVM	RL	XGBoost
Gini	0.01	0.05	-0.07	-0.02	-0.03	-0.01
Fisher	-0.02	-0.05	0.15	0.01	-0.01	0.01
F-Ratio	-0.02	-0.06	0.15	0	-0.01	0.01
Pearson	0	0.09	0.02	0.07	0.01	0
Spearman	0	0.09	0.02	0.08	0.01	0
P-Valor	-0.01	0	0.0	0.01	0.01	0.01

Tabela 4.4 – Resultados das correlações no banco de dados Pima Indians Diabetes Database.

Os valores que mais chamaram a atenção foram com as métricas Fisher e F-Ratio nos modelos *Random Forest* (Gráfico do Fisher 4.5).

É possível observar no gráfico uma tendência de que os valores se tornam mais altos à medida que a métrica de Fisher aumenta. Curiosamente, embora valores baixos se tornem mais raros com o aumento da métrica, ainda é possível encontrar valores altos mesmo quando a métrica está baixa.

4.1.5 Heart Disease Classification Dataset

Os resultados gerados podem ser vistos na Tabela 4.5.

É possivel, a partir da Tabela 4.5 observar que nas métricas Fisher e F-Ratio no modelo de Árvore de decisão acontece uma correlação negativa (Quanto mais próximo de 0 melhor o resultado dos F1-Scores). Esse resultado é curioso já que a proposta dessas métricas é que quanto maior o valor melhor a separação que ele identifica entre as classes.

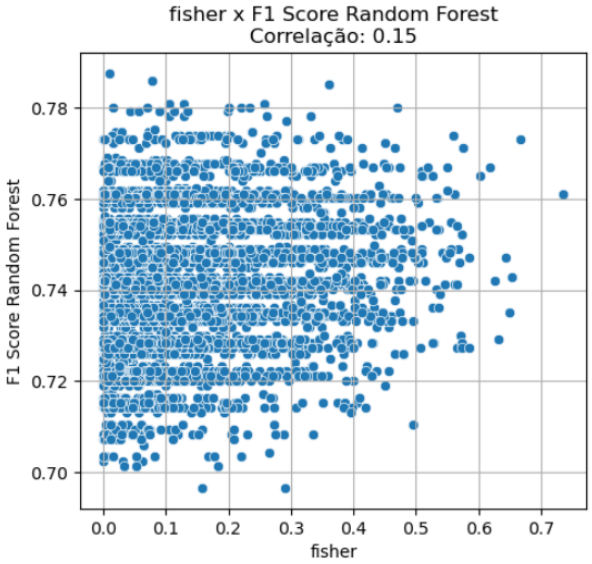

Figura 4.5 – Resultado usando a métrica Fisher no modelo de *Random Forest* no banco Pima Indians Diabetes Database

X	DT	NN	RF	SVM	RL	XGBoost
Gini	0.10	0.04	-0.04	0.02	-0.01	0.02
Fisher	-0.26	-0.05	0	-0.02	0	-0.10
F-Ratio	-0.26	-0.05	-0.01	-0.03	0	0.10
Pearson	0.01	0.04	-0.05	-0.01	0	-0.02
Spearman	0.01	0.05	-0.05	-0.02	0	-0.02
P-Valor	0.22	0.04	0.04	- 0.01	-0.01	0.03
PCA	-0.01	0.01	0	0	-0.04	0.01

Tabela 4.5 – Resultados das correlações no banco de dados Heart Disease Classification Dataset.

4.1.6 QSAR Biodegradation Data Set

Os resultados gerados podem ser vistos na Tabela 4.6.

X	DT	NN	RF	SVM	RL	XGBoost
Gini	-0.02	-0.01	0.08	-0.02	0.01	-0.01
Fisher	-0.02	0.01	-0.01	-0.06	0.01	0.05
F-Ratio	-0.02	0.01	0	-0.06	0.01	0.05
Pearson	0.03	0.06	-0.02	-0.04	0	-0.01
Spearman	0.03	0.05	-0.01	-0.03	0	-0.01
P-Valor	0.01	-0.01	-0.02	0.02	0	-0.03
PCA	-0.01	0.01	0	0	-0.02	0.01

Tabela 4.6 – Resultados das correlações no banco de dados QSAR Biodegradation Data Set.

Nessa base, existem casos em que a correlação possuiu um valor diferente de 0, mas que não foram significativos para serem analisados. Em resumo, esse banco não demonstrou uma boa reação às *features* que foram geradas.

4.1.7 Tic-Tac-Toe Endgame database

Os resultados gerados podem ser vistos na Tabela 4.7.

X	DT	NN	RF	SVM	RL	XGBoost
Gini	-0.05	0	0.08	0	0.01	0.01
Fisher	-0.07	-0.02	-0.22	0.01	-0.04	-0.04
F-Ratio	-0.07	-0.02	-0.22	0.01	-0.03	-0.04
Pearson	0.01	-0.01	0.01	0.08	0.01	-0.01
Spearman	0.01	-0.01	0.01	0.08	0.02	-0.01
P-Valor	0.03	0.01	0.13	0	0.04	0.02
PCA	0.03	-0.09	0.03	-0.26	-0.23	0.01

Tabela 4.7 – Resultados das correlações no banco de dados Tic-Tac-Toe Endgame database.

Esse banco mostrou um comportamente similar ao banco Heart Disease Classification Dataset, onde é visto uma correlação negativa nas métricas Fisher e F-Ratio, porém agora no modelo de *Random Forest* apresentado na Figura 4.6. Interessante também ressaltar que foi o primeiro banco em que a estratégia usando o PCA mostrou uma reação, porém, as correlações foram negativas, o que é curioso pensando que o valor de importância deseja ser maximizado.

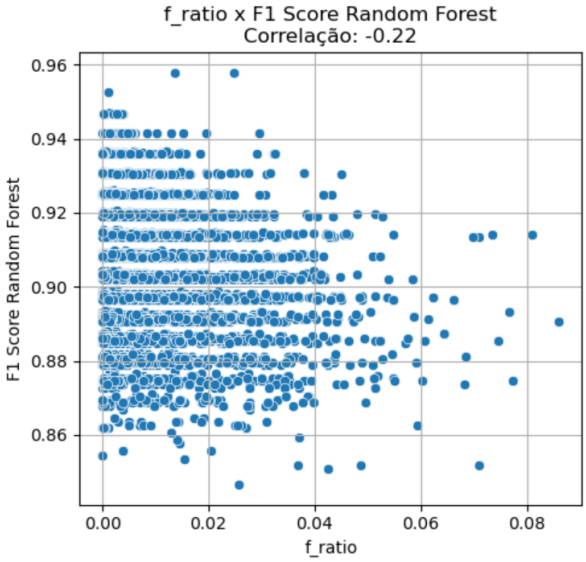

Figura 4.6 – Resultado usando a métrica *F-Ratio* no modelo *Random Forest* no banco Tic-Tac-Toe Endgame database

4.2 Testes usando Árvore de Decisão como gerador

Como observado nos testes anteriores, a estratégia baseada no PCA não se mostrou eficaz, pois não foi possível determinar sua qualidade de forma conclusiva. As *features* geradas não permitiram uma análise consistente, dificultando a avaliação do método. Diante disso, também realizamos testes utilizando árvores de decisão como geradoras de *features*, com o objetivo de forçar, de maneira mais consistente, a criação de atributos explicitamente relevantes. Como a nova métrica será baseada na classificação de uma árvore de decisão treinada diretamente no conjunto de dados, é importante analisar a diferença entre os resultados obtidos antes e depois da inserção desse elemento. Isso permitirá avaliar o impacto da nova métrica na performance das *features* geradas.

4.2.1 Breast Cancer Wisconsin (Diagnostic)

Os resultados gerados podem ser vistos na Tabela 4.8.

	DT	NN	RF	SVM	RL	XGBoost
PCA	-0.17	-0.04	-0.12	0	-0.04	-0.04

Tabela 4.8 – Resultados das correlações no banco de dados Breast Cancer Wisconsin (Diagnostic).

O banco de dados Breast Cancer Wisconsin (Diagnostic) é um dos mais simples do nosso conjunto de testes. Mesmo ao utilizar um gerador mais direcionado, os resultados obtidos na Figura 4.7 em termos de correlação não se mostraram significativos. No entanto, ao analisar a métrica de *F1-score*, a abordagem apresentou desempenhos mais promissores.

Antes da adição da nova métrica, o *F1-score* do modelo estava consistentemente em torno de 0.95. Com a inclusão da métrica, alguns modelos conseguiram atingir um *F1-score* de até 1, o que demonstra a capacidade do gerador de produzir atributos relevantes. No entanto, a estratégia baseada em PCA não foi eficaz na identificação dessa melhoria de forma consistente.

Figura 4.7 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão.

4.2.2 Climate Model Simulation Crashes

Os resultados gerados podem ser vistos na Tabela 4.9.

	DT	NN	RF	SVM	RL	XGBoost
PCA	0.02	0.03	0.12	0	0.13	0.08

Tabela 4.9 – Resultados das correlações no banco de dados Climate Model Simulation Crashes.

O banco de dados Climate Model Simulation Crashes apresentou uma correlação insatisfatória. No entanto, o gráfico da Figura 4.8 resultante revelou um padrão peculiar: a importância calculada por PCA permaneceu constante em um valor fixo, enquanto o *F1-Score* oscilava, indicando que a métrica gerava modelos diferentes, apesar de não conseguir diferenciá-los de forma eficaz. Esse comportamento sugere que a variabilidade no *F1-Score* pode ter sido influenciada por outros fatores não capturados pela métrica em si.

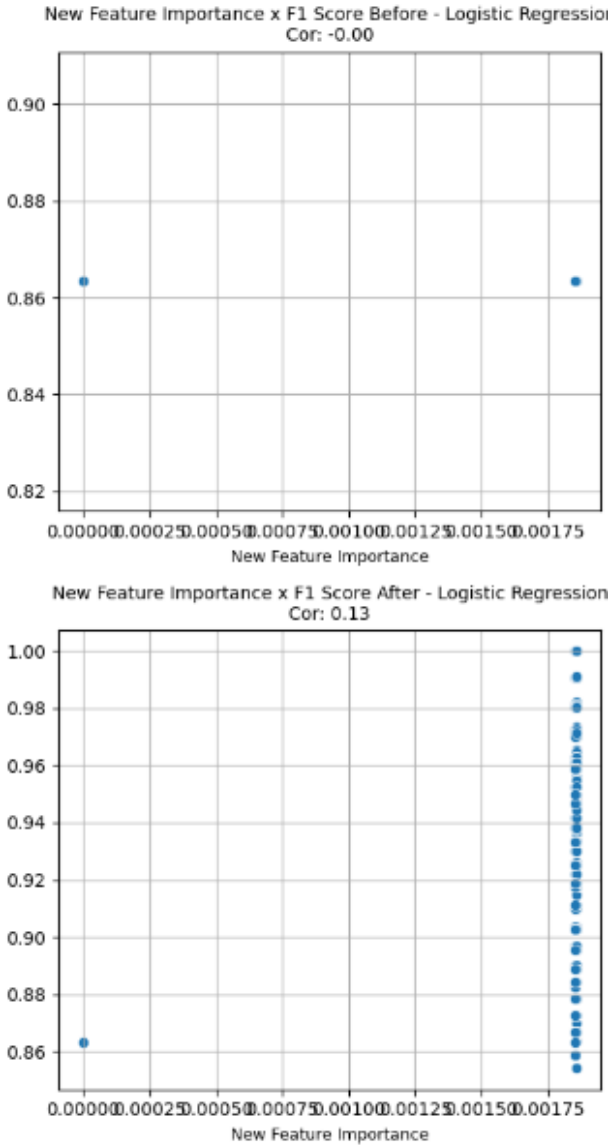

Figura 4.8 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão na base de dados Climate Model Simulation Crashes

4.2.3 Statlog (German Credit Data)

Os resultados gerados podem ser vistos na Tabela 4.10.

	DT	NN	RF	SVM	RL	XGBoost
PCA	-0.36	0.03	-0.42	0	-0.46	-0.34

Tabela 4.10 – Resultados das correlações no banco de dados Statlog (German Credit Data).

O banco de dados Statlog (German Credit Data) demonstrou uma correlação extremamente alta. Observou-se uma correlação negativa significativa nos modelos, onde quanto menor a importância calculada por PCA, melhor foi o desempenho do modelo. Esse comportamento negativo é interessante, pois indica que a redução da importância por PCA pode estar associada a uma melhor performance. Além disso, é relevante analisar a diferença no *F1-Score* dos modelos

antes e depois da Figura 4.9, já que os modelos cujos valores da métrica se aproximam de 0 tendem a apresentar um desempenho superior em comparação àqueles cujos valores se distanciam dessa referência.

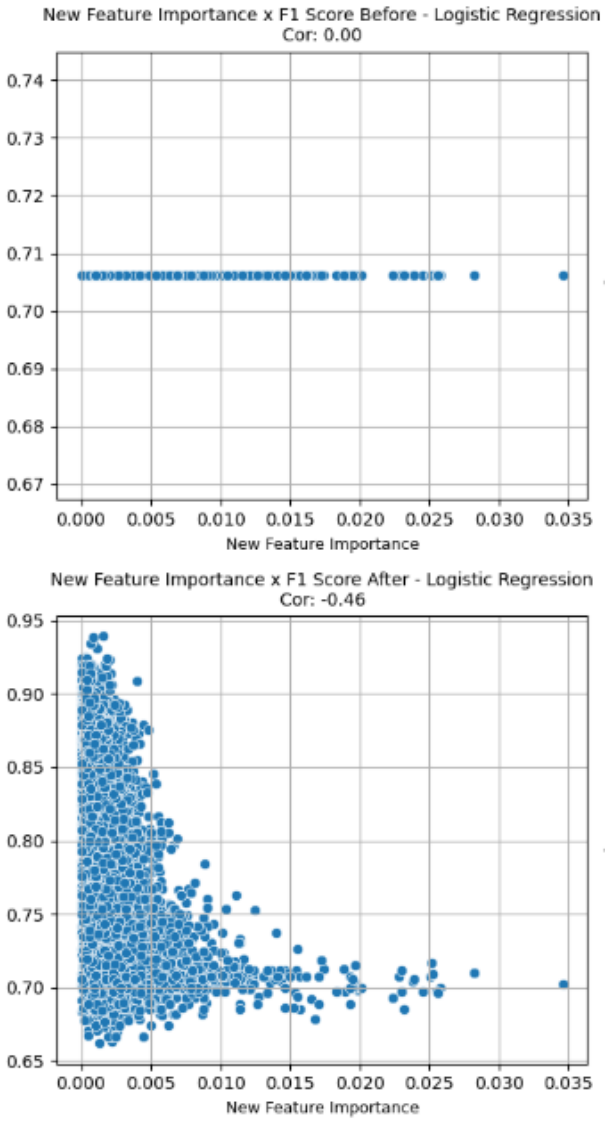

Figura 4.9 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão na base de dados Statlog (German Credit Data)

4.2.4 Pima Indians Diabetes Database

Os resultados gerados podem ser vistos na tabela 4.11

	DT	NN	RF	SVM	RL	XGBoost
PCA	-0.01	0.02	-0.02	0	-0.03	-0.06

Tabela 4.11 – Resultados das correlações no banco de dados Pima Indians Diabetes Database.

O banco de dados Pima Indians Diabetes Database apresentou uma correlação baixa. No entanto, foi observado um pico interessante nos resultados da Figura 4.10, onde quanto mais próximo de um determinado valor a métrica de importância estava, melhor o *F1-Score* do modelo. Esse comportamento sugere que, ao utilizar o gerador baseados em árvore, os modelos parecem identificar padrões específicos que proporcionam um aumento significativo no desempenho, levando a um "boost"nos resultados.

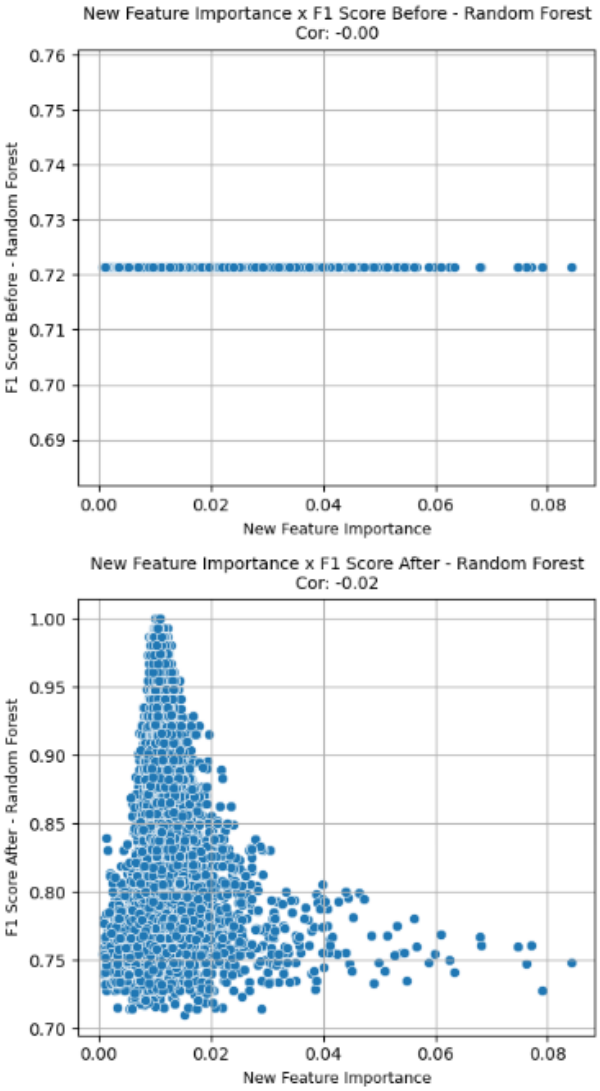

Figura 4.10 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão na base de dados Pima Indians Diabetes Database

4.2.5 Heart Disease Classification Dataset

Os resultados gerados podem ser vistos na tabela 4.12

	DT	NN	RF	SVM	RL	XGBoost
PCA	0.46	0.37	0.53	0	0.51	0.49

Tabela 4.12 – Resultados das correlações no banco de dados Heart Disease Classification Dataset.

O banco de dados Heart Disease Classification Dataset apresentou uma correlação alta, com valores chegando a 0.50 em alguns casos. Observou-se uma correlação positiva moderada entre a importância calculada por PCA e o desempenho dos modelos, sugerindo que a maior importância atribuída por PCA pode estar relacionada a melhores resultados. Além disso, é importante observar a variação no *F1-Score* dos modelos antes e depois da Figura 4.11, já que os

modelos que mantiveram valores mais distantes de 0 na métrica apresentaram um desempenho superior, em contraste com os modelos da Figura 4.10 que ficaram mais próximos de 0.

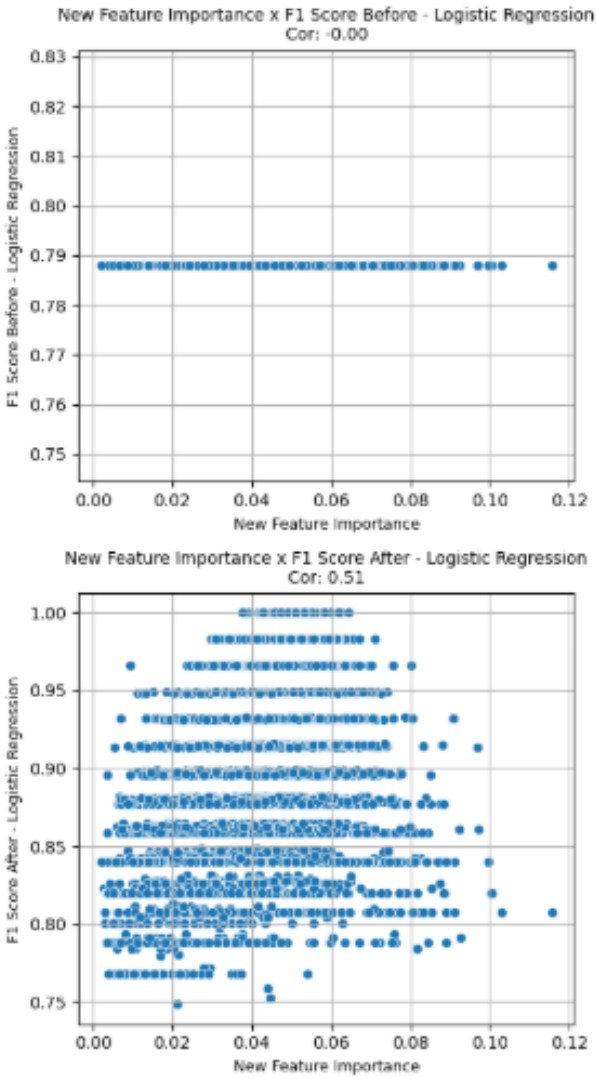

Figura 4.11 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão na base de dados Heart Disease Classification Dataset

4.2.6 QSAR Biodegradation Data Set

Os resultados gerados podem ser vistos na tabela 4.13

	DT	NN	RF	SVM	RL	XGBoost
PCA	0.19	0.18	0.13	0.17	0.15	0.16

Tabela 4.13 – Resultados das correlações no banco de dados QSAR Biodegradation Data Set.

O conjunto de dados QSAR Biodegradation Data Set apresentou, de modo geral, uma baixa correlação. Entretanto, conforme ilustrado na Figura 4.12, foi identificado um comportamento interessante: o *F1-Score* do modelo tende a ser mais alto quando a métrica de importância se aproxima de um determinado valor. Esse padrão indica que, ao utilizar geradores baseados em árvore, os modelos conseguem capturar características específicas dos dados que resultam em um aumento significativo no desempenho, produzindo um verdadeiro "boost"nos resultados.

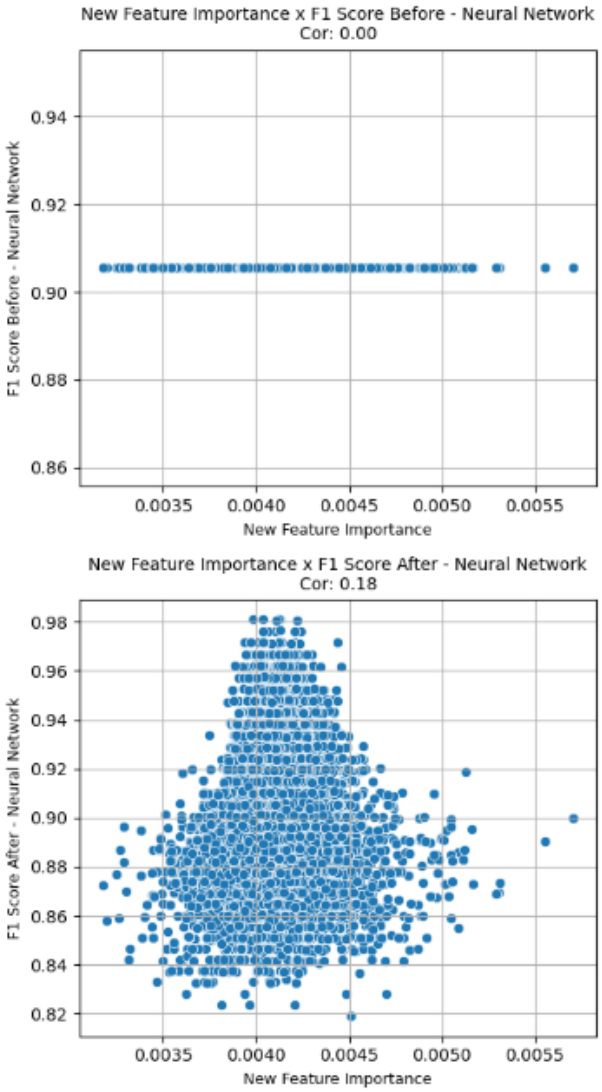

Figura 4.12 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão na base de dados QSAR Biodegradation Data Set

4.2.7 Tic-Tac-Toe Endgame database

Os resultados gerados podem ser vistos na tabela 4.14

	DT	NN	RF	SVM	RL	XGBoost
PCA	0.14	-0.12	-0.24	0.04	0.29	-0.38

Tabela 4.14 – Resultados das correlações no banco de dados Tic-Tac-Toe Endgame database.

O banco de dados Tic-Tac-Toe Endgame database consiste em disposições de jogos da velha. Modelos mais complexos conseguem classificar facilmente se o jogo acabou ou não. No entanto, ao adicionar uma nova *feature* baseada na classificação de uma árvore de decisão, observou-se uma oscilação no comportamento da base de dados. Em alguns casos, modelos como *XGBoost* e *Random Forest*, que já obtinham bons resultados, alcançaram desempenhos

melhores, mas também piores. Por outro lado, a regressão logística, que anteriormente apresentava dificuldades, conseguiu mostrar resultados mais consistentes como mostrado na Figura 4.13.

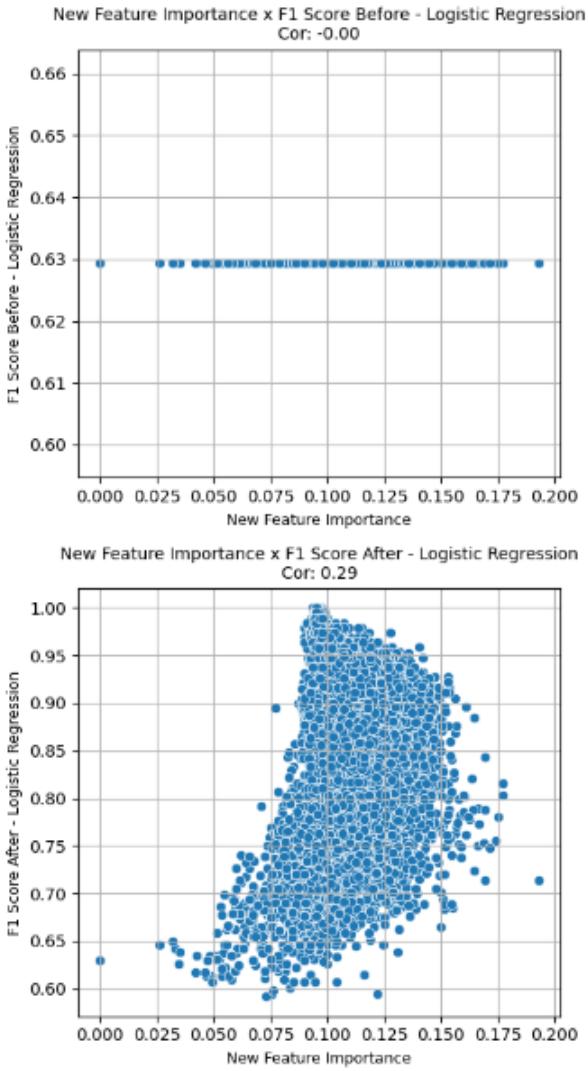

Figura 4.13 – Resultado antes e depois da inserção de um atributo usando Árvore de Decisão na base de dados Tic-Tac-Toe Endgame database

5 Considerações Finais

5.1 Conclusão

Após a análise dos dados gerados, foi possível observar que nem todos os modelos reagiram positivamente à adição de *features* nas bases de dados. Um exemplo disso foi o desempenho do SVM e da Regressão Logística. Ao utilizar o gerador com MLP, os modelos não mostraram ganho significativo de informação. No entanto, ao aplicar o gerador baseado em árvores, esses modelos foram os principais beneficiados, sugerindo que a nova *feature* (uma classificação da base de dados gerada por uma árvore aleatória) foi interpretada como um atributo relevante, resultando em uma melhoria nos seus *scores*.

Em comparação, todas as métricas reagiram bem quando a base de dados era favorável a mesma. Isso significa que a premissa do trabalho está no caminho certo. A partir da composição do banco de dados (se os atributos são linearmente relacionáveis, por exemplo), é possível identificar qual dessas métricas vai impactar e criar *features* positivas para o problema. É importante destacar o uso da estratégia baseada em PCA, que se revelou a métrica mais complexa de ser analisada. Sua reação variou de forma peculiar em cada banco de dados: em alguns casos, não apresentou correlação, enquanto em outros, revelou relações interessantes e significativas.

No tópico de banco de dados, é visto que eles são diversos e representativos, no sentido que compõem uma grande margem dos problemas de classificação que encontramos. Um caso que vale a pena ser citado é a Tic-Tac-Toe Endgame database, que representa os estados finais de um jogo da velha onde seus atributos são posições de um tabuleiro do jogo, e que, mesmo sendo um caso incomum, foi capaz de mostrar um resultado de correlação significativo.

Um dos principais obstáculos deste trabalho era o algoritmo de geração de *features* aleatórias, que foi superado com o uso do gerador baseado em árvores. Esse gerador não só conseguiu criar uma ampla variedade de *features* aleatórias, mas também gerou *features* diretamente positivas, facilitando significativamente o processo de análise das métricas.

O objetivo deste trabalho era demonstrar se, ao utilizar uma métrica de seleção de *features*, seria possível redimensionar um banco de dados e obter resultados positivos. Embora esse objetivo não tenha sido totalmente alcançado, foi possível identificar algumas tendências e apontar um caminho para alcançar essa meta. A métrica baseada em PCA se destacou por sua abordagem única, analisando os dados de forma conjunta, ao contrário das métricas tradicionais, que fazem essa análise de maneira separada. Realizar testes com outras métricas semelhantes pode gerar *insights* valiosos e contribuir para a evolução dessa pesquisa.

Referências

AKSU, D.; ÜSTEBAY, S.; AYDIN, M. A.; ATMACA, T. Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. In: SPRINGER. *Computer and Information Sciences: 32nd International Symposium, ISCIS 2018, Held at the 24th IFIP World Computer Congress, WCC 2018, Poznan, Poland, September 20-21, 2018, Proceedings 32*. [S.l.], 2018. p. 141–149.

ALI, A. T.; ABDULLAH, H. S.; FADHIL, M. N. Voice recognition system using machine learning techniques. *Materials Today: Proceedings*, Elsevier, p. 1–7, 2021.

ARAUJO, W. O. de; COELHO, C. J. Análise de componentes principais (pca). *University Center of Anápolis, Annapolis*, 2009.

ARIK, S. Ö.; PFISTER, T. Tabnet: Attentive interpretable tabular learning. In: *Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2021. v. 35, n. 8, p. 6679–6687.

AZOFF, E. M. *Neural network time series forecasting of financial markets*. [S.l.]: John Wiley & Sons, Inc., 1994.

BENESTY, J.; CHEN, J.; HUANG, Y.; COHEN, I. Pearson correlation coefficient. In: _____. *Noise Reduction in Speech Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1–4. ISBN 978-3-642-00296-0. Disponível em: <https://doi.org/10.1007/978-3-642-00296-0_5>.

BOSCH, N. et al. Automl feature engineering for student modeling yields high accuracy, but limited interpretability. *Journal of Educational Data Mining*, v. 13, n. 2, p. 55–79, 2021.

CBURNETT. *An example artificial neural network with a hidden layer*. 2006. [Online; accessed 30-Setember-2024]. Disponível em: <https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg>.

CHAGANTI, S. Y.; NANDA, I.; PANDI, K. R.; PRUDHVITH, T. G.; KUMAR, N. Image classification using svm and cnn. In: IEEE. *2020 International conference on computer science, engineering and applications (ICCSEA)*. [S.l.], 2020. p. 1–5.

CLEMENTS, J. M.; XU, D.; YOUSEFI, N.; EFIMOV, D. Sequential deep learning for credit risk monitoring with tabular financial data. *arXiv preprint arXiv:2012.15330*, 2020.

CONNELLY, L. M. Introduction to analysis of variance (anova). *Medsurg Nursing*, v. 30, n. 3, 2021.

ELSSIED, N. O. F.; IBRAHIM, O.; OSMAN, A. H. A novel feature selection based on one-way anova f-test for e-mail spam classification. *Research Journal of Applied Sciences, Engineering and Technology*, Maxwell Scientific Publications, v. 7, n. 3, p. 625–638, 2014.

FILHO, D. B. F.; JÚNIOR, J. A. S. Desvendando os mistérios do coeficiente de correlação de pearson (r). *Revista política hoje*, Recife, v. 18, n. 1, p. 115–146, 2009.

FITNI, Q. R. S.; RAMLI, K. Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In: IEEE. *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. [S.l.], 2020. p. 118–124.

GAN, M.; ZHANG, L. Iteratively local fisher score for feature selection. *Applied Intelligence*, Springer, v. 51, p. 6167–6181, 2021.

GOLDBERG, Y. *Neural network methods in natural language processing*. [S.l.]: Morgan & Claypool Publishers, 2017.

GOLDSCHMIDT, R.; PASSOS, E.; BEZERRA, E. *Data mining*. [S.l.]: Elsevier Brasil, 2015.

GRUS, J. *Data Science from Scratch: First Principles with Python*. 2st. ed. [S.l.]: O'Reilly Media, Inc., 2019.

GU, Q.; LI, Z.; HAN, J. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

GUIMARÃES, P. R. B. Análise de correlação e medidas de associação. *Curitiba: universidade federal do paraná*, p. 1–26, 2017.

HAHNE, F.; HUBER, W.; GENTLEMAN, R.; FALCON, S.; GENTLEMAN, R.; CAREY, V. Unsupervised machine learning. *Bioconductor case studies*, Springer, p. 137–157, 2008.

HOFMANN, T.; SCHÖLKOPF, B.; SMOLA, A. J. Kernel methods in machine learning. 2008.

KHLAIEF, A.; NGUYEN, K.; MEDJAHER, K.; PICOT, A.; MAUSSION, P.; TOBON, D.; CHAUCHAT, B.; CHERON, R. Feature engineering for ball bearing combined-fault detection and diagnostic. In: IEEE. *2019 IEEE 12th international symposium on diagnostics for electrical machines, power electronics and drives (SDEMPED)*. [S.l.], 2019. p. 384–390.

KUMAR, P.; BISWAS, M. Svm with gaussian kernel-based image spam detection on textual features. In: IEEE. *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*. [S.l.], 2017. p. 1–6.

LARHMAM. *Maximum-margin hyperplane and margin for an SVM trained on two classes. Samples on margins are called support vectors.* 2018. [Online; accessed 3-February-2024]. Disponível em: <https://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=1198920933>.

LAVALLEY, M. P. Logistic regression. *Circulation*, Am Heart Assoc, v. 117, n. 18, p. 2395–2399, 2008.

LYASHENKO, V.; LAARIEDH, F.; SOTNIK, S.; AYAZ, A. M. Recognition of voice commands based on neural network. TEM Journal, 2021.

MAKOWER, M. *Principles of Operations Research–with Applications to Managerial Decisions*. [S.l.]: JSTOR, 1976.

MEDINA, F. *Consideraciones sobre el índice de Gini para medir la concentración del ingreso*. [S.l.]: CEPAL-División de Estadística y Proyecciones Económicas, 2011.

NASTESKI, V. An overview of the supervised machine learning methods. *Horizons. b*, v. 4, n. 51-62, p. 56, 2017.

NEMBRINI, S.; KÖNIG, I. R.; WRIGHT, M. N. The revival of the gini importance? *Bioinformatics*, Oxford University Press, v. 34, n. 21, p. 3711–3718, 2018.

NIU, Y. Walmart sales forecasting using xgboost algorithm and feature engineering. In: IEEE. *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. [S.l.], 2020. p. 458–461.

OZDEMIR, S. *Feature Engineering Bookcamp*. Manning, 2022. ISBN 9781617299797. Disponível em: <https://books.google.com.br/books?id=3n6HEAAAQBAJ>.

PICTON, P.; PICTON, P. *What is a neural network?* [S.l.]: Springer, 1994.

RIGATTI, S. J. Random forest. *Journal of Insurance Medicine*, American Academy of Insurance Medicine 1700 Magnavox Way, Fort Wayne, IN 46804, v. 47, n. 1, p. 31–39, 2017.

RODGERS, J. L.; NICEWANDER, W. A. Thirteen ways to look at the correlation coefficient. *The American Statistician*, Taylor & Francis, v. 42, n. 1, p. 59–66, 1988.

ROJAS-DOMÍNGUEZ, A.; PADIERNA, L. C.; VALADEZ, J. M. C.; PUGA-SOBERANES, H. J.; FRAIRE, H. J. Optimal hyper-parameter tuning of svm classifiers with application to medical diagnosis. *Ieee Access*, IEEE, v. 6, p. 7164–7176, 2017.

RUSTAM, Z.; HIDAYAT, R. et al. Indonesia composite index prediction using fuzzy support vector regression with fisher score feature selection. *International Journal on Advanced Science, Engineering and Information Technology*, INSIGHT-Indonesian Society for Knowledge and Human Development, v. 9, n. 1, p. 121–128, 2019.

SANTOS, C. Estatística descritiva. *Manual de auto-aprendizagem*, v. 2, p. 3, 2007.

SHWARTZ-ZIV, R.; ARMON, A. Tabular data: Deep learning is not all you need. *Information Fusion*, v. 81, p. 84–90, 2022. ISSN 1566-2535. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1566253521002360>.

SONG, H. *AutoFE: efficient and robust automated feature engineering*. Tese (Doutorado) — Massachusetts Institute of Technology, 2018.

SONI, A.; DHARMACHARYA, D.; PAL, A.; SRIVASTAVA, V. K.; SHAW, R. N.; GHOSH, A. Design of a machine learning-based self-driving car. *Machine Learning for Robotics Applications*, Springer, p. 139–151, 2021.

SOUSA, Á. Coeficiente de correlação de pearson e coeficiente de correlação de spearman: o que medem e em que situações devem ser utilizados? *Correio dos Açores*, Gráfica Açoreana, Lda, p. 19–19, 2019.

ST, L.; WOLD, S. et al. Analysis of variance (anova). *Chemometrics and intelligent laboratory systems*, Elsevier, v. 6, n. 4, p. 259–272, 1989.

SUGUIURA, F. O. R. Decision tree in machine learning. *Revista Varianza*, p. 38, 2022.

SUTHAHARAN, S.; SUTHAHARAN, S. Decision tree learning. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, Springer, p. 237–269, 2016.

TAN, F. Improving feature selection techniques for machine learning. 2007.

TRAORE, B. B.; KAMSU-FOGUEM, B.; TANGARA, F. Deep convolution neural network for image recognition. *Ecological informatics*, Elsevier, v. 48, p. 257–268, 2018.

TSEKICHUN. *Random forest explain*. 2021. [Online; accessed 30-Setember-2024]. Disponível em: <https://commons.wikimedia.org/wiki/File:Random_forest_explain.png>.

ULMER, D.; MEIJERINK, L.; CINÀ, G. Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In: PMLR. *Machine Learning for Health*. [S.l.], 2020. p. 341–354.

VAGHEFI, S. A.; STAMMBACH, D.; MUCCIONE, V.; BINGLER, J.; NI, J.; KRAUS, M.; ALLEN, S.; COLESANTI-SENNI, C.; WEKHOF, T.; SCHIMANSKI, T. et al. Chatclimate: Grounding conversational ai in climate science. *Communications Earth & Environment*, Nature Publishing Group UK London, v. 4, n. 1, p. 480, 2023.

WANG, Z.; VELIČKOVIĆ, P.; HENNES, D.; TOMAŠEV, N.; PRINCE, L.; KAISERS, M.; BACHRACH, Y.; ELIE, R.; WENLIANG, L. K.; PICCININI, F. et al. Tacticai: an ai assistant for football tactics. *Nature communications*, Nature Publishing Group UK London, v. 15, n. 1, p. 1906, 2024.

WITSIL, A.; FEE, D.; DICKEY, J.; PEÑA, R.; WAXLER, R.; BLOM, P. Detecting large explosions with machine learning models trained on synthetic infrasound data. *Geophysical Research Letters*, Wiley Online Library, v. 49, n. 11, p. e2022GL097785, 2022.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, v. 2, n. 1, p. 37–52, 1987. ISSN 0169-7439. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. Disponível em: <https://www.sciencedirect.com/science/article/pii/0169743987800849>.

XUE, H.; YANG, Q.; CHEN, S. Svm: Support vector machines. In: *The top ten algorithms in data mining*. [S.l.]: Chapman and Hall/CRC, 2009. p. 51–74.

YANG, H.; CHAN, L.; KING, I. Support vector machine regression for volatile stock market prediction. In: SPRINGER. *International conference on intelligent data engineering and automated learning*. [S.l.], 2002. p. 391–396.

ZHANG, K.; WANG, J.; LIU, T.; LUO, Y.; LOH, X. J.; CHEN, X. Machine learning-reinforced noninvasive biosensors for healthcare. *Advanced Healthcare Materials*, Wiley Online Library, v. 10, n. 17, p. 2100734, 2021.

ZHOU, Z.-H. *Machine learning*. [S.l.]: Springer nature, 2021.