UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA QUÍMICA

JULIA CARATTA OLIVA

APRENDIZADO DE MÁQUINA APLICADO A UM ESTUDO DE CASO DE CLASSIFICAÇÃO DE QUALIDADE DE MAÇÃS

Uberlândia
2024

JULIA CARATTA OLIVA

APRENDIZADO DE MÁQUINA APLICADO A UM ESTUDO DE CASO DE CLASSIFICAÇÃO DE QUALIDADE DE MAÇÃS

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia Química da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de Engenheiro Químico.

Área de concentração: Núcleo de Modelagem, Controle e Otimização de Processos (NUCOP)

Orientadora: Prof.ª Dr.ª Sarah Arvelos Altino

Uberlândia

2024

RESUMO

O uso de aprendizado de máquina, do inglês Machine Learning (ML), tem se destacado por suas contribuições significativas em diversas áreas, incluindo a Engenharia Química, ao possibilitar a análise de grandes volumes de dados e a extração de padrões complexos. Modelos de classificação, em particular, têm sido amplamente aplicados para prever resultados em processos industriais, permitindo a automação de decisões e a melhoria da eficiência. Este estudo teve como foco a aplicação de diferentes técnicas *ensemble* em modelos de árvores de decisão, com o objetivo de investigar o desempenho dessas abordagens em problemas de classificação. Técnicas *ensemble*, como *bagging* e *boosting*, combinam múltiplos modelos para melhorar a performance geral, especialmente em cenários onde modelos individuais podem apresentar limitações em termos de precisão e robustez. A investigação visou avaliar o desempenho das técnicas de *ensemble* em comparação com modelos isolados, considerando métricas de performance como acurácia, precisão, recall e F1-score. Além disso, foram utilizadas técnicas de interpretabilidade de modelos, como o *SHAP* (*Shapley Additive Explanations*), para entender o impacto de cada variável nas previsões e garantir que os resultados fossem interpretáveis e aplicáveis a contextos industriais. A interpretabilidade é uma questão crítica, especialmente em áreas como a Engenharia Química, onde a compreensão das decisões do modelo pode ser tão importante quanto sua precisão. Os resultados demonstraram que ambas as técnicas de *ensemble*, *bagging* e *boosting*, tiveram bom desempenho, com alta acurácia e robustez na maioria das tarefas. No entanto, o *boosting* apresentou uma capacidade superior de ajuste dos erros, ao minimizar de forma mais eficaz os vieses presentes nos dados. Essa técnica mostrou-se particularmente eficiente em melhorar a performance em casos onde os modelos de árvore de decisão isolados enfrentaram dificuldades em lidar com outliers ou padrões complexos nos dados. As conclusões deste trabalho reforçam a importância de explorar diferentes técnicas de aprendizado de máquina e avaliar seu desempenho de forma criteriosa, considerando não apenas as métricas tradicionais, mas também a interpretabilidade dos modelos. Técnicas como *boosting* oferecem soluções promissoras para problemas complexos, garantindo que os modelos não apenas tenham boa performance, mas também possam ser compreendidos e aplicados de forma prática em ambientes reais.

Palavras-chave: Aprendizado de Máquina; Algoritmo; Engenharia Química; Modelo.

ABSTRACT

The use of machine learning (ML) has become prominent for its significant contributions across various fields, including Chemical Engineering, by enabling the analysis of large data volumes and extracting complex patterns. Classification models, in particular, have been widely applied to predict outcomes in industrial processes, allowing decision automation and efficiency improvements. This study focused on applying different ensemble techniques to decision tree models, aiming to investigate the performance of these approaches in classification problems. ensemble techniques, such as bagging and boosting, combine multiple models to improve overall performance, especially in scenarios where individual models may have limitations in terms of accuracy and robustness. The research aimed to evaluate the performance of ensemble techniques compared to standalone models, considering performance metrics such as accuracy, precision, recall, and F1-score. Additionally, model interpretability techniques, such as SHAP (Shapley Additive Explanations), were used to understand the impact of each variable on predictions and ensure that the results were interpretable and applicable to industrial contexts. Interpretability is a critical issue, especially in fields like Chemical Engineering, where understanding the model's decisions can be as important as its accuracy. The results showed that both ensemble techniques, bagging and boosting, performed well, with high accuracy and robustness in most tasks. However, boosting exhibited a superior ability to adjust errors, effectively minimizing biases in the data. This technique proved particularly efficient in improving performance in cases where standalone decision tree models struggled with outliers or complex data patterns. The conclusions of this study reinforce the importance of exploring different machine learning techniques and evaluating their performance carefully, considering not only traditional metrics but also the interpretability of the models. Techniques like boosting offer promising solutions for complex problems, ensuring that models not only perform well but can also be understood and practically applied in real-world environments.

Keywords: Machine Learning; Algorithm; Chemical Engineering; Model.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
LGBM	*Light Gradient boosting Machine*
ML	*Machine Learning* (Aprendizado de Máquina)
UFU	Universidade Federal de Uberlândia
TPE	*Tree-Structured Parzen Estimator*

SUMÁRIO

1. INTRODUÇÃO

Durante as últimas décadas, as novas tecnologias possibilitaram progressos significativos na ciência, permeando todas as áreas correlatas à Engenharia Química. As leis clássicas que regem fenômenos físico-químicos tinham inicialmente suas bases desenvolvidas a partir de métodos determinísticos. Esses algoritmos apresentam estratégias de soluções mais simplificadas, em que resoluções gráficas e algébricas são utilizadas para encontrar a relação entre as variáveis de entrada e saída (SPROVIERI; COMELLI, 2021).

No entanto, a mensuração de todos os elementos envolvidos no fenômeno pode apresentar-se como um problema na indústria, seja pela inconveniente interrupção do processo produtivo ou pela dificuldade de mensuração de todos seus elementos. Isso implica incerteza na solução modelada, o que gera resultados imprecisos e limitados (GARCIA, 2009).

Como forma alternativa às soluções numéricas, os modelos empíricos se apresentam como opção de maior viabilidade (HIMMELBLAU; BISCHOFF, 1968). Esses podem apresentar modelos mais simplificados em relação aos modelos físico-químicos. No entanto, eles garantem soluções relativamente ágeis com bom nível de confiabilidade para determinado grau de incerteza.

Para lidar com os riscos inerentes relacionados à limitação de recursos, utiliza-se técnicas de simulação, que são pautadas em conceitos probabilísticos. As noções da distribuição de probabilidade dada uma amostra viabiliza que, a partir de um nível de incerteza, seja possível encontrar valores que permitam a compreensão de um evento (RODRIGUES, 2003).

Para a utilização das técnicas empíricas, há necessidade de aplicação da estatística multivariada (VICINI, 2005). Como parte do desenvolvimento, deve-se definir quais são os parâmetros que mais influenciam o processo, bem como estudar o efeito da alteração de uma variável de entrada no efeito estudado (PANNELL, 1997).

Uma forma de evitar a alta complexidade matemática relativa às resoluções algébricas, cria-se premissas que regem essas resoluções, implicando em simplificações da realidade. Por isso, Seborg (2009) diz que “Um modelo nada mais é do que uma abstração

matemática de um processo real".

Uma das maiores vantagens em relação às técnicas tradicionais está relacionada à alta agilidade em realizar diferentes simulações, o que possibilita que processos de maior complexidade sejam estudados. Isso gera impacto positivo na tomada de decisão industrial, uma vez que garante maior embasamento técnico. O grande desafio do engenheiro passa a ser equilibrar o nível de detalhamento do sistema em detrimento de esforço computacional (FERNANDES, 1998).

Arthur Samuel, pioneiro na área de *Machine Learning* (ML), definiu em 1959 que o Aprendizado de Máquina é o campo de estudo que possibilita aos computadores a habilidade de aprender sem explicitamente programá-los (BHEEMAIAH *et al.*, 2017).

Machine learning pode ser considerado uma técnica de modelagem matemática. Ele envolve a criação de modelos matemáticos que aprendem padrões a partir de dados, de forma que possam fazer previsões ou classificações sobre novos dados. Esses modelos utilizam conceitos de estatística, álgebra linear, cálculo e teoria da probabilidade para ajustar os parâmetros a partir dos dados de entrada e fornecer uma saída que corresponda a determinado comportamento ou padrão. No contexto da modelagem matemática, o aprendizado de máquina é amplamente utilizado para representar fenômenos complexos, como predição de séries temporais, reconhecimento de padrões, classificação e regressão, entre outros. A diferença é que, ao invés de o modelo ser completamente definido por equações explícitas, ele "aprende" essas relações com base em dados e otimiza seus parâmetros através de técnicas como gradiente descendente, regularização e outras ferramentas de otimização (GERON, 2017).

O presente trabalho teve como objetivo geral explorar essas técnicas e validar sua aplicabilidade de diferentes técnicas de ML no contexto industrial apresentando um estudo de caso, em que os modelos probabilísticos podem se apresentar como alternativa interessante para tomada de decisão. Além disso, os objetivos específicos foram:

- Estudar técnicas estatísticas de modelagem de problemas reais, a partir da distribuição dos dados até a construção de modelos preditivos;
- Comparar a eficiência de treinamento e generalização de diferentes modelos de *Machine Learning*;
- Analisar a eficácia de algoritmos como *Random Forest* (*bagging*) e *Gradient boosting*

(*boosting*) utilizando métricas de performance: precisão, recall e F1-score, para determinar suas capacidades preditivas em diferentes cenários.

2. DESENVOLVIMENTO

2.1. PROCESSO ANALISADO

O objetivo do trabalho foi determinar a qualidade de lotes de maçãs a partir de suas características sensoriais e físicas, sendo elas: grau de doçura, crocância, suculência, acidez, tamanho, peso e maturação.

Essas características são comumente acessíveis a produtores e vendedores rurais, podendo ser utilizadas em processos de distribuição de alimentos em larga escala (FONTES, 2007). Em virtude disso, o modelo facilita a seleção e distribuição dos alimentos produzidos, de forma que garanta distribuição estratégica desses alimentos.

Como resultado, o projeto de um modelo de aprendizado de máquina auxilia na determinação do valor dos produtos de acordo com a qualidade. A utilização dos resultados pode se estender de acordo com o interesse comercial, de forma que possibilite ao produtor mapear quais condições produtivas podem alavancar a qualidade de suas colheitas.

Os dados utilizados no trabalho foram obtidos a partir de fonte pública. Este conjunto de dados foi disponibilizado por Nidula Elgiriyewithana na comunidade *Kaggle* (ELGIRIYEWITHANA, 2024), sendo originários de uma empresa de agricultura americana. A base de dados foi selecionada por sua relevância e adequação ao escopo da pesquisa.

2.2. DETALHES COMPUTACIONAIS

A parte programável do projeto desde trabalho abrangeu a linguagem Python (versão 3.12.4), com utilização das suas principais bibliotecas científicas: *NumPy, Pandas* e *Matplotlib* para visualização e tratamento inicial dos dados (GRUS, 2015).

Para a parte de modelagem do problema, utilizou-se a biblioteca de código aberto de aprendizado de máquina *scikit-learn* (PEDREGOSA et al., 2011). Ela disponibiliza passos e boas práticas de criação de modelos de predição, bem como exemplos de seus resultados (BSD License, 2024).

Os modelos de *LightGBM* e *Random Forest* foram testados para a implementação e comparação dos métodos de *boosting* e *bagging*. Como apoio, a biblioteca *Optuna* (AKIBA et al., 2019) foi utilizada para a otimização dos hiperparâmetros dos modelos de aprendizado de máquina.

Todos os conceitos podem ser desenvolvidos de forma aplicável por meio do Projeto Jupyter Notebook (*https://jupyter.org/*), uma ferramenta poderosa de desenvolvimento de computação interativa. Os detalhes de cada método serão citados em momentos oportunos no presente trabalho.

2.3. ANÁLISE EXPLORATÓRIA DOS DADOS

Os princípios fundamentais para o desenvolvimento dos modelos de interesse indicam que é necessário garantir a qualidade dos dados utilizados. No entanto, os dados brutos nem sempre são fornecidos já preparados para sua utilização. Por isso, é fundamental que o desenvolvimento do modelo abranja etapas de pré-processamento dos dados (DE ALMEIDA PRADO ALVES BATISTA, 2003).

> Seus primeiros passos envolvem a remoção e tratamento de informações nulas ou inconsistentes; bem como a seleção e manipulação estratégica dos dados originais (BRUCE *et al.*, 2017). A segunda parte – referente à combinação de características relevantes – é nomeada de "*Feature Engineering*". Essa fase garante a efetividade e confiabilidade dos dados que serão utilizados para a aprendizagem do algoritmo, sendo fundamental para atingir boa precisão preditiva (ALICE ZHENG, 2016).

Para isso, utilizou-se de técnicas estatísticas de validação e visualização das informações. Essa etapa é comumente conhecida como Análise Exploratória. O estágio foi descrito por Mukhiya e Ahmed (2020) como o primeiro passo após a coleta, em que os dados são trabalhados de forma a extrair informações relevantes ao projeto. Essa etapa pode ser classificada como "definição do problema", uma vez que as oportunidades, custos e benefícios

podem ser extraídos a partir do que há disponível.

Ainda na Análise Exploratória, Mukhiya e Ahmed (2020) descrevem a preparação das informações incluindo a limpeza dos dados como etapa antecedente da análise descritiva. O procedimento é fundamental para que a etapa de sumarização das informações, uma vez que as correlações e causalidades do processo podem ser encontradas para definição das variáveis preditoras mais relevantes.

Para visualização completa dos dados, utiliza-se ferramentas como gráficos de dispersão, histogramas, boxplots, entre outras técnicas de representações gráficas e numéricas (BRUCE *et al.*, 2017).

Toda a implementação da análise foi realizada na linguagem *Python* no ambiente de desenvolvimento interativo *Jupyter Notebook*. As ferramentas comportam bibliotecas importantes para a análise e modelagem dos dados, sendo elas (GRUS, 2015):

- *Pandas*: a biblioteca tem como objetivo facilitar a manipulação de tabelas — nomeadas Dataframes —, uma vez que concentra funcionalidades de leitura e escrita dos dados. Ela facilita a agregação e modificação de dados de acordo com a necessidade do projeto de forma eficiente e otimizada.

- *Numpy*: o pacote possibilita a computação numérica em *Python*, útil para trabalhos, pesquisas e empresas que necessitam de apoio em operações matemáticas com arrays e matrizes. A utilização de dados matriciais melhora a agilidade na simulação de combinações numéricas entre os dados, que serve como base para as operações estatísticas que sustentam os princípios da modelagem.

- *Matplotlib*: a biblioteca apoia a visualização gráfica dos dados. Ela facilita a visualização das correlações de forma interativa de forma elaborada. O pacote é especialmente útil para aplicações de média complexidade, como a produção de gráficos que facilitam a relação entre variáveis de entrada e sobreposição de dados.

- *Seaborn*: é uma biblioteca integrada aos pacotes *Matplotlib* e *Pandas*, que auxilia na plotagem dos dados contidos nos *Dataframes* e *arrays*. O pacote tem como objetivo produzir gráficos informativos, apresentando alta flexibilidade em seu uso.

- *scikit-learn*: biblioteca de aprendizado de máquina com função principal de permitir a modelagem preditiva. Oferece ferramentas para tarefas de classificação, regressão,

clusterização, redução de dimensionalidade, entre outros.

2.4. APRENDIZAGEM SUPERVISIONADA

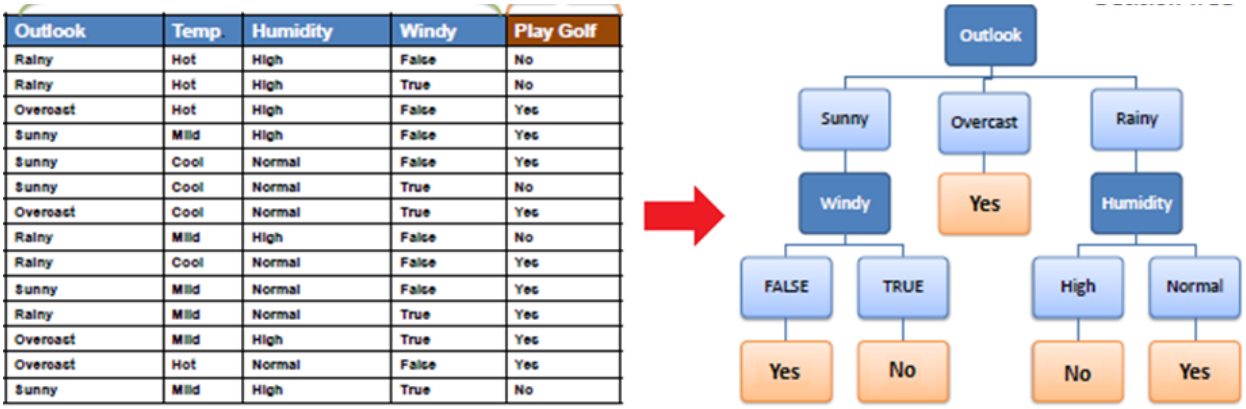
Nesta seção, destacam-se os conceitos do tipo de aprendizagem utilizada, que é caracterizada pelo tipo de supervisão durante o treinamento. Para o objetivo de classificação, esse é o modelo mais adequado. É importante ressaltar que existem outros tipos de aprendizagem, como os modelos não supervisionados ou baseados em aprendizado por reforço (GÉRON, 2017).

No aprendizado supervisionado, os dados são fornecidos ao algoritmo com marcações do tipo de resposta que é desejada. Eles são utilizados principalmente para previsão de uma variável de saída, baseado em conjuntos de dados já definidos e classificados (GÉRON, 2017).

A técnica foi escolhida por se ajustar bem aos dados para diferentes contextos. O algoritmo de "Árvore de Decisão" cria estruturas condicionais fixas para prever um resultado. MORETTIN (2021) explicam que os modelos baseados em árvore envolvem uma segmentação dos dados, gerado pelas variáveis preditoras em algumas regiões definidas estatisticamente. A definição dessas áreas apoia-se em alguma medida de erro de classificação de forma iterativa, até identificação da separação ótima.

Na Figura 1, é possível visualizar, mesmo que de forma simplificada e genérica, os princípios de funcionamento desse algoritmo. Este Figura mostra como um dado tabulado pode ser apresentado através de uma árvore de decisão. Ela é um exemplo didático de um algoritmo baseado em árvore, que tem como objetivo classificar o clima, com o intuito de determinar a viabilidade de realizar um jogo de golfe.

Figura . Exemplo de modelo de Árvore de Decisão.



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Fonte: Árvores de Decisão do Zero usando *Python* e *NumPy* com visualização (2024)

A técnica pode ser ajustada conforme a necessidade. Seus vieses serão definidos na etapa de treinamento do modelo, a partir de um conjunto de dados selecionados. É possível compreender seu funcionamento a partir da visão geral:

> O modelo de Árvore de Decisão particiona recursivamente os dados em subconjuntos com base nos valores das características selecionadas, criando "nós" de decisão. Em cada nó, o algoritmo identifica a característica que oferece a divisão ótima dos dados, e esse processo continua até que um critério de parada seja atendido. A árvore resultante pode então ser usada para fazer previsões em novos dados (MATTOS, 2024).

Entre seus principais benefícios, destacam-se a fácil interpretação de resultados. No contexto em que é importante entender como o modelo tomou sua decisão do modelo, essa técnica se apresenta como altamente eficaz. Além disso, os modelos baseados em árvore de decisão são computacionalmente mais simples, o que gera menor custo. Por isso, a técnica abrange alta flexibilidade em sua aplicação (FERREIRA, 2021).

2.5. ESCOLHA DOS ALGORITMOS

2.5.1. MÉTODOS ENSEMBLE

Quando se trata de previsão, aproveitar os resultados de múltiplas árvores costuma ser mais potente do que simplesmente usar uma única árvore (BRUCE, 2017). Por isso os métodos de agrupamento ganharam ampla aplicação, pela maior acurácia de seus resultados para um custo viável.

Dessa forma, os modelos de agrupamento podem ser classificados em grandes grupos. Dentre eles, temos interesse nos métodos de *bagging* e *boosting*. O trabalho tem como objetivo comparar os métodos, utilizando os exemplos de *RandomForestClassifier* (Floresta Aleatória) que se baseia na aplicação de *bagging* em árvores de decisão, e *LightGradientBoostingMachine* (LGBM).

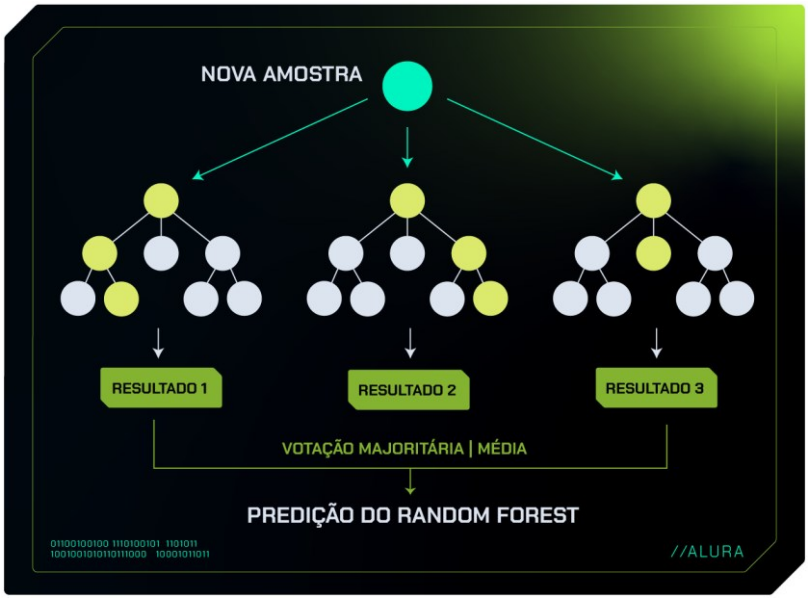
2.5.2. BAGGING

De acordo com Breiman (1996), a palavra *bagging* é originária do termo "*bootstrap agregador*", que consiste em subdividir o conjunto de dados (população) em pequenas amostras randômicas e com reposição. Dessa forma, os dados são rearranjados em amostras menores de maneira iterativa, até que os dados subamostrados tenham distribuição que se aproxima da população inteira.

A técnica parte de premissas de tendência do limite central e ajuda a entender o comportamento da média e do desvio populacional. Dessa forma, com maior diversificação dos dados de treinamento e teste, um modelo *bagging* é capaz de gerar múltiplas predições, reduzindo a variância do modelo resultante (CHA ZHANG, 2012). No final do treinamento, a melhor predição é obtida a partir do voto majoritário (BREIMAN, 1996).

Um exemplo de classificador desse tipo é o *RandomForestClassifier*, em que pode-se ajustar o número de árvores a serem treinadas, conforme indica a Figura 2. Esta Figura ilustra a criação das árvores geradas por amostras aleatórias a partir do conjunto populacional de dados. Cada amostra é gerada a partir de um subconjunto aleatório (SAHOUR, H., 2021). As marcações em amarelo ilustram os caminhos de tomada de decisão de cada árvore para conclusão da classe predita.

Figura 2. Exemplo de funcionamento de modelos *bagging*.



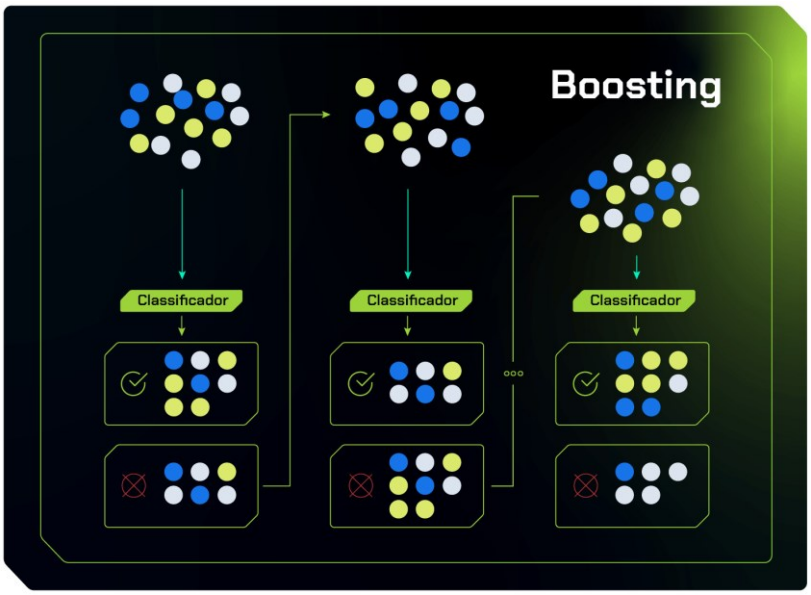
Fonte: Alura (2024)

2.5.3. BOOSTING

O método *boosting* tem similaridades com o *bagging*, mas realiza a otimização das métricas de forma sequencial (BRUCE, 2017). Dessa forma, o ajuste do modelo é realizado a partir dos erros obtidos no modelo anterior. O aprendizado é realizado dando mais pesos aos registros com alto erro de predição, gerando posteriores treinamentos ponderados pelo desempenho das piores classificações anteriores.

A Figura 3 apresenta um esquema representativo do processo de sequenciamento, em que ilustra-se o processo iterativo de treino. Nesse processo, utiliza-se os erros de predição da árvore anterior - ilustrados com marcações vermelhas - para gerar novas amostras de treino. Os treinos sequenciais ocorrem de forma a gerar classificadores mais fortes e robustos, ilustrados com marcações verdes. Entre os modelos de *boosting*, o LGBM ganha destaque devido sua alta velocidade de treinamento e baixo uso de memória (MICROSOFT, 2019).

Figura 3. Exemplo de funcionamento de modelos *boosting*.



Fonte: Alura (2024)

2.6. OTIMIZAÇÃO DOS HIPERPARÂMETROS

Para treinar um modelo de aprendizado de máquina, é fundamental entender a importância dos hiperparâmetros. Eles podem ser entendidos como variáveis que integram o sistema de predição, que devem ser ajustados de forma experimental (AMAZON, 2023).

Encontrar o conjunto ideal de hiperparâmetros é fundamental para que o modelo tenha a performance, uma vez que ele auxilia que o modelo consiga generalizar seu potencial de predição para dados nunca vistos anteriormente. Esse processo é chamado de regularização (GÉRON, 2017) e evita que o modelo sobreajuste seus resultados a partir dos dados de treinamento.

Ao utilizar bases de dados relativamente pequenas, técnicas que exploram todas as combinações possíveis são úteis e não exigem alto custo computacional (GRUS, 2015). Para isso, é possível utilizar a função *Grid Search*, que realiza uma busca exaustiva para todas as combinações possíveis. Uma vez que ele realiza métodos de força bruta para encontrar o mínimo global da função, ele é altamente eficiente na busca do melhor resultado.

No entanto, o método apresenta a desvantagem de ineficiência para dados de alta

dimensionalidade (PINHEIRO, 2023), dado que o número de iterações cresce exponencialmente em função do volume de dados e da quantidade de hiperparâmetros a serem otimizados. Por isso, à medida que o volume de dados aumenta, os custos computacionais se tornam inviáveis, exigindo técnicas alternativas.

Os algoritmos de otimização Bayesiana resolvem essa limitação. Para implantação, utiliza-se o *Optuna*, uma biblioteca de otimização de hiperparâmetros de aprendizado de máquina. Seu princípio de funcionamento é simples, partindo dos termos de "estudo" — que define a função objetivo a ser otimizada — e a "tentativa" — que configura cada execução da função objetivo e armazena as informações de combinação realizadas (RISHABH, 2022).

Dessa forma, por meio de múltiplas tentativas, a função de estudo atinge os hiperparâmetros que possuem ajuste ótimo da função objetivo, que auxiliará o agente a chegar ao modelo de predição ideal.

O algoritmo utilizado para otimização dos hiperparâmetros foi o *Tree-Structured Parzen Estimator* (TPE). O método TPE é baseado no Teorema de *Bayes*, que faz parte da teoria probabilística.

O Teorema de *Bayes* é utilizado para calcular a probabilidade (*P*) de um evento acontecer dado uma condição específica (BIAGGI, 2023), que pode ser definido pela equação 3. Esse teorema é útil quando interpretado como uma regra para indução: os dados e o evento A são considerados como sucessores de B, o grau de crença anterior à realização do experimento. Assim, P(B) é chamado de probabilidade *a priori* a qual será modificada pela experiência. A experiência é determinada pela verossimilhança P(A|B). Finalmente, P(B|A) é a probabilidade posterior, ou o nível de crença após a realização do experimento.

Equação 3 - Teorema de *Bayes*

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)}$$

Onde:

- P(A): É a probabilidade de A ocorrer;
- P(B): É a probabilidade de B ocorrer;

- $P(A|B)$: indica a probabilidade de A ocorrer dado que B ocorreu;
- $P(B|A)$: indica a probabilidade de B ocorrer dado que A ocorreu.

A cada tentativa, o TPE ajusta um modelo Gaussiano ao conjunto de valores de parâmetros, de forma a minimizar a função de perda a cada iteração realizada. Dessa forma, ele modela a probabilidade de encontrar um hiperparâmetro ótimo em função das tentativas anteriores (BERGSTRA, 2011).

O *Optuna* pode ser integrado a qualquer *framework* de aprendizado de máquina, sendo útil para treinamento de modelos de classificação, como o LightGBM e *Random Forest*.

Os hiperparâmetros otimizados nos treinamentos dos modelos são descritos abaixo. Todos podem ser encontrados com detalhes na documentação do *scikit-learn* (PEDREGOSA et al., 2011), que define seus valores padrões e exemplifica o uso. Na Tabela 1, define-se os principais hiperparâmetros utilizados, bem como sua definição aplicada ao modelo.

Tabela 1. Definição de hiperparâmetros.

Hiperparâmetro	Descrição
random_state	Controle de aleatoriedade e reprodutibilidade do modelo. Ao defini-lo de forma fixa, há garantia que o modelo gera os mesmos resultados de treinamento ao ser reexecutado. O parâmetro define a aleatoriedade de escolha de amostras e inicialização de pesos.
n_estimators	Especifica o número de árvores que serão criadas no método *ensemble*. Valores maiores geram maior variabilidade de resultados e aumenta o potencial de predição. No entanto, seu aumento também gera maior tempo de treinamento.
max_depth	Determina a profundidade máxima da árvore de decisão. O sobreajuste do modelo está diretamente ligado a altas profundidades, uma vez que o aumento da árvore gera maior complexidade dos critérios de predição. Em contrapartida, baixas profundidades geram árvores muito simplificadas, que podem gerar subsajuste do modelo.
n_jobs	Define o número de núcleos de centrais de processamento a serem utilizadas no treinamento do modelo. O parâmetro auxilia na agilidade de treinamento.
criterion	Coeficiente utilizado para divisão das árvores. O hiperparâmetro surge do conceito de entropia, que define a desordem dos dados. Quanto maior a entropia dos dados, menor a ordenação.
max_features	Define o máximo de variáveis a serem utilizadas para divisão dos nós. De acordo com a documentação, "as opções incluem 'sqrt' (raiz quadrada do número de características), 'log2' (logaritmo em base 2 do número de características), ou None (considera todas as características)."
learning_rate	Controla o tamanho do passo que o modelo realiza a cada iteração no processo de treinamento para encontrar o ajuste ótimo global de hiperparâmetros. Quanto menor o parâmetro, melhor a robustez do resultado. No entanto, a diminuição da variável também gera processos de treinamento mais lentos.
reg_alpha	Termo de regularização, responsável por ajuste para evitar sobreajuste do modelo aos dados de treino. O parâmetro auxilia na prevenção de sobreajuste, uma vez que penaliza variáveis com maior taxa de erro de predição e diminui ou zera o peso, caso necessário. Gera modelos mais simples após finalização das iterações.
reg_lambda	Termo de regularização, responsável por ajuste para evitar sobreajuste do modelo aos dados de treino. O parâmetro auxilia na prevenção de sobreajuste, uma vez que penaliza variáveis com maior taxa de erro de predição e diminui seu peso, caso necessário. Gera modelos mais simples após finalização das iterações.

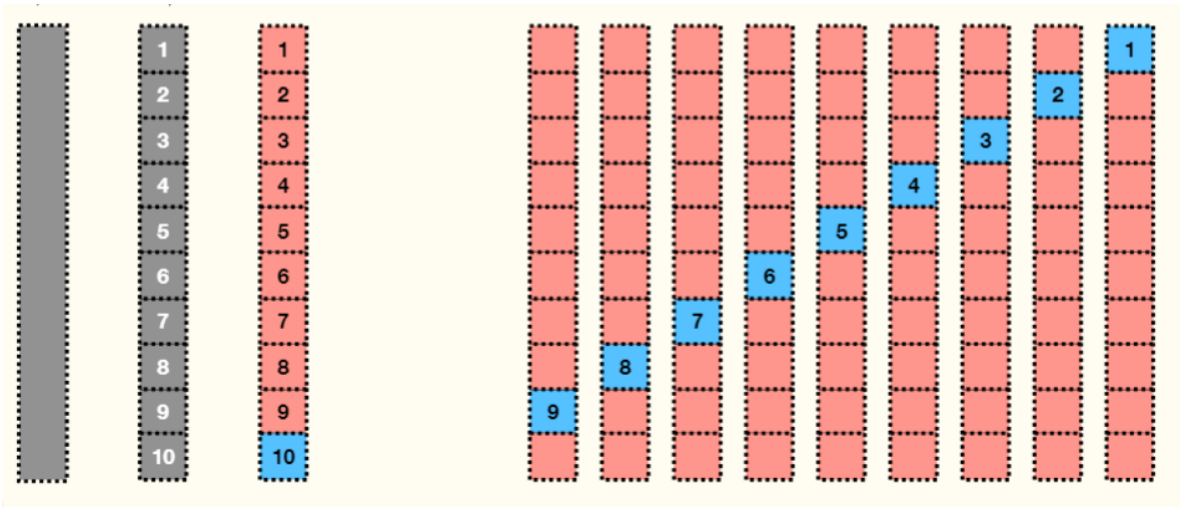
2.6.1 VALIDAÇÃO CRUZADA

Ao treinar um modelo de *machine learning* com dados relativamente escassos, o erro de teste pode não ser estimado de forma confiável. Dessa forma, a capacidade que o modelo de generalizar a população a partir da amostra pode ser afetada (ZUBEN, 1996). Isso ocorre pois as predições são geradas a partir da função de densidade de probabilidade, que depende diretamente da amostra utilizada para treinamento do modelo. Usualmente, define-se que 80% dos dados são utilizados para treinamento, sendo 20% deles separados exclusivamente para validação dos resultados (FERREIRA, 2021).

O procedimento de validação cruzada *(K-fold*) é utilizado para resolver o problema da alta variância de resultados de treinamento, que é gerada naturalmente, a depender da amostra que foi aleatoriamente segmentada do todo (EFRON, 1983).

Para realizá-lo, utiliza-se uma parcela dos dados para treinamento. Starmer (2022) descreve detalhadamente seu procedimento. Para cada repetição de treino, o algoritmo calcula o erro entre as predições geradas nos dados de teste em relação às marcações reais. A quantidade de repetições é definida pelo K, que define a quantidade de subconjuntos utilizados para teste (YADAV, 2016). A Figura 4 apresenta um esquema do processo de validação cruzada.

Figura 4. Esquematização da validação cruzada com K=10



Fonte: STARMER (2022)

2.7. MÉTRICAS DE AVALIAÇÃO E INTERPRETAÇÃO DO MODELO

As métricas de avaliação são fundamentais em todo o processo de modelagem, sendo úteis para etapas como a definição das *features* na análise exploratória ou na interpretação dos resultados de cada modelo na etapa de validação.

2.7.1. INFORMATION VALUE

De acordo com BIAGGI (2024), o método *Information Value* (IV) tem como objetivo identificar as variáveis contínuas com maior poder de predição. Ela mede a relação entre boas distribuições em relação às más distribuições, conforme Equação 4. Define-se:

- Boa distribuição: porcentagem de eventos em que a variável alvo é igual a 1;
- Má distribuição: porcentagem de eventos em que a variável alvo é igual a 0.

Equação 4 - Information Value

$$IV = \sum_{k=1}^{K} (G_k - B_k) * ln(\frac{G_k}{B_k})$$

Onde:

- G_k: nomeado de "*good distribution*", representa o percentual de eventos em que a classe positiva atende ao valor específico em relação ao total de observações;
- B_k: nomeado de "*bad distribution*", representa o percentual de eventos em que a classe positiva não atende ao valor específico em relação ao total de observações.

No final do cálculo do IV para todas as variáveis contínuas, é possível entender a importância relativa de cada variável para a predição desejada (Cao, J., 2023).

2.7.2. CORRELAÇÃO DE PEARSON

O coeficiente de correlação mede a correlação linear entre duas variáveis (FIGUEIREDO, 2009). A métrica é baseada na associação da distribuição das frequências da variável ou pelo comportamento de suas variâncias, conforme mostra a Equação 5.

Equação 5 - Coeficiente de correlação de Pearson

$$r = \frac{1}{n-1} * \sum (\frac{x_i - \underline{x}}{s_x}) * \left(\frac{y_i - \underline{y}}{s_y}\right)$$

Onde:

- r: coeficiente de correlação de Pearson, que mede a força e a direção da relação linear entre duas variáveis;
- n: número de pares de observações (ou dados);
- x_i: representa o valor individual da variável x_i na observação i;
- y_i: representa o valor individual da variável y_i na observação i;
- $\underline{x}$: média da variável x;
- $\underline{y}$: média da variável y;
- s_x: desvio padrão de x.

O coeficiente de correlação de Pearson considera a premissa de normalidade dos dados. Define-se arbitrariamente que uma alta correlação terá coeficiente r >= 0.7. Caso essa condição seja satisfeita, apenas uma das variáveis correlacionadas deve ser selecionada, a fim de evitar redundância nos dados de treinamento.

2.7.3. IMPORTÂNCIA DAS VARIÁVEIS

O "*feature importance*" é um atributo disponível dentro da biblioteca *scikit-learn*, que auxilia na avaliação das predições geradas por uma árvore de classificação. De acordo com a documentação, a importância de cada variável é calculada a partir do ganho gerado a cada nó. Nó pode ser definido como o conjunto de regras que gera uma ramificação na árvore, sendo essa uma representação gráfica ou de regra de um valor a ser dividido (BRUCE *et al.*, 2017).

Essa divisão é baseada na soma da diminuição da impureza a cada tomada de decisão, que é a medição da heterogeneidade de cada sub partição dos dados. Dessa forma, entende-se que quanto mais misturadas as classes estão mais impuro é o nó.

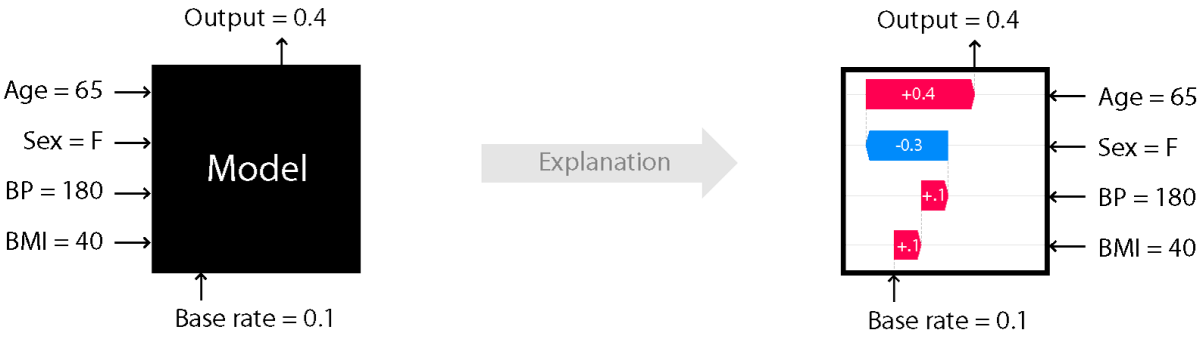
Quanto maior o coeficiente de importância, maior a relação entre a predição do modelo. É importante que as *features* tenham importância balanceada, com o intuito de não gerar vieses na predição do modelo. Essa técnica pode ser combinada com outras para uma boa avaliação do resultado (FERREIRA, 2021).

2.7.4. SHAPLEY ADDITIVE EXPLANATIONS (SHAP)

O *Shapley Additive exPlanations* (SHAP) é uma abordagem baseada na teoria da probabilidade que auxilia no entendimento da saída do modelo. A técnica é fundamental no contexto de modelos probabilísticos, em que nem sempre é possível explicar com exatidão o motivo da predição gerada (MENGNAN, 2018).

Para resolver o problema, a biblioteca SHAP fornece uma solução gráfica para interpretação da correlação entre as variáveis de entrada (*input*) com a saída do modelo (*output*). De forma intuitiva, a contribuição de cada variável é a média de sua importância para predição em relação a todas as iterações de predição que não a incluem, conforme indica a Figura 5. Esta Figura indica seu funcionamento que parte dos parâmetros utilizados no treinamento do modelo e a explicação gerada pela biblioteca SHAP.

Figura 5. Exemplificação do fundamento do SHAP



Fonte: Big Data Brasil (2020)

De acordo com Big Data Brasil (2020), é possível interpretar o gráfico SHAP da seguinte forma:

> Em cada linha onde se encontra cada uma das variáveis, cada ponto representa um exemplo do dataset. Se o ponto está azul, significa que para aquele ponto, aquela variável tem um valor baixo (quando considerando todos os valores que aquela variável assume no dataset). Quanto mais vermelho o ponto é, mais alto é o valor. A posição do ponto no eixo horizontal indica o efeito, o SHAP Value, daquele ponto. Quanto mais à direita, mais positiva é a contribuição daquela variável naquele ponto.

A técnica SHAP é especialmente útil para avaliar se a direção das features é importante, uma vez que ela indica se a variável gerou impacto positivo ou negativo na predição (REIS, 2021). Dessa forma, é possível avaliar a coerência do modelo com a realidade, sendo essa uma etapa fundamental da sua validação.

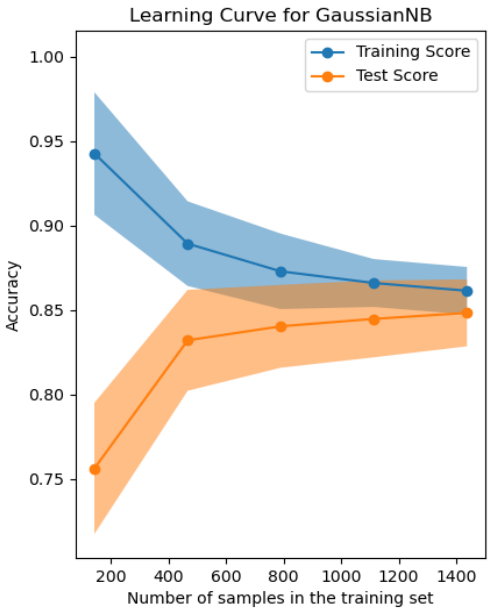
2.7.5. CURVA DE APRENDIZADO

A curva de aprendizagem é útil para validação do tamanho da amostra de treinamento. Dessa forma, ela indica visualmente o efeito de adicionar mais amostras durante a etapa de treinamento, conforme ilustra a Figura 6.

Altas distâncias entre as pontuais de dados entre as bases de treinamento e teste podem indicar que o modelo sofreu sobreajuste, uma vez que haveria queda nas métricas de performance no momento que o modelo encontra novos conjuntos de dados (KAMPAKIS, 2024).

O método pode auxiliar na redução de custos e gestão de desempenho no treinamento. Isso ocorre pois as curvas de aprendizado tendem a atingir um platô — indicativo de que não há alteração dos resultados independente do volume de dados disponíveis. Dessa forma, poderíamos concluir que o desempenho do modelo não melhoraria com maior disponibilidade de dados. O inverso também funciona, em que a inexistência de platô poderia indicar que o modelo não possui dados suficientes para gerar predições satisfatórias (KAMPAKIS, 2024).

Figura 6. Exemplificação da curva de aprendizado



Fonte: KAMPAKIS (2024)

2.7.6. MÉTRICAS DE PERFORMANCE DO MODELO

As métricas de avaliação utilizadas para avaliar o modelo na fase de validação foram acurácia, sensibilidade (recall), precisão e F1-score. Elas utilizam da comparação entre os rótulos binários originais em relação aos rótulos gerados no modelo de classificação. Dessa forma, é possível encontrar 4 classes (MARIANO, 2021):

- Verdadeiro positivo (VP): a predição gerada é correta, sendo de classe positiva;
- Verdadeiro negativo (VN): a predição gerada é correta, sendo de classe negativa;
- Falso positivo (FP): a predição gerada é incorreta, sendo de classe positiva;
- Falso negativo (FN): a predição gerada é incorreta, sendo de classe negativa.

A partir das classes acima, calcula-se as métricas de desempenho as quais estão indicadas na Tabela 2.

Tabela 2. Métricas de avaliação dos modelos.

Métrica	Equação
Acurácia	$\frac{VP + VN}{VP + VN + FP + FN}$
Precisão	$\frac{VP}{VP + VN}$
Sensibilidade (Recall)	$\frac{VP}{VP + FP}$
F1 - Score	$2 * \frac{precisão * sensibilidade}{precisão + sensibilidade}$

2.7.7. TESTE DE HIPÓTESE

O teste de hipótese é uma ferramenta fundamental na validação de resultados a partir de experimentos. Integrante fundamental da estatística inferencial, os testes nos auxiliam a tomar decisões a partir de números obtidos experimentalmente e generalizá-los a partir das informações obtidas em uma amostra (ZIBETTI, 2020).

De acordo com Beiguelman (1996), o teste qui-quadrado objetiva comparar proporções entre 2 grupos independentes. Dessa forma, é possível validar se a divergências entre as frequências encontradas dada uma variável são relevantes, considerando a variação aleatória e natural de eventos.

Para realizar um teste de hipótese, deve-se definir uma premissa a ser validada. Essa premissa, que pode ser validada ou rejeitada, atende a um nível de confiança (FERREIRA, 2016). Por convenção, define-se que a tolerância ao erro pode variar entre 1% a 5%, sendo essa a variável relacionada ao nível de significância α (alfa). Os resultados atingem os níveis de confiança de, respectivamente, 99% a 95%.

Usualmente, define-se:

- Hipótese Nula (H0): não há diferença entre os grupos. A variação observada ocorre provavelmente ao acaso.

- Hipótese Alternativa (Ha): há diferença entre os grupos estudados.

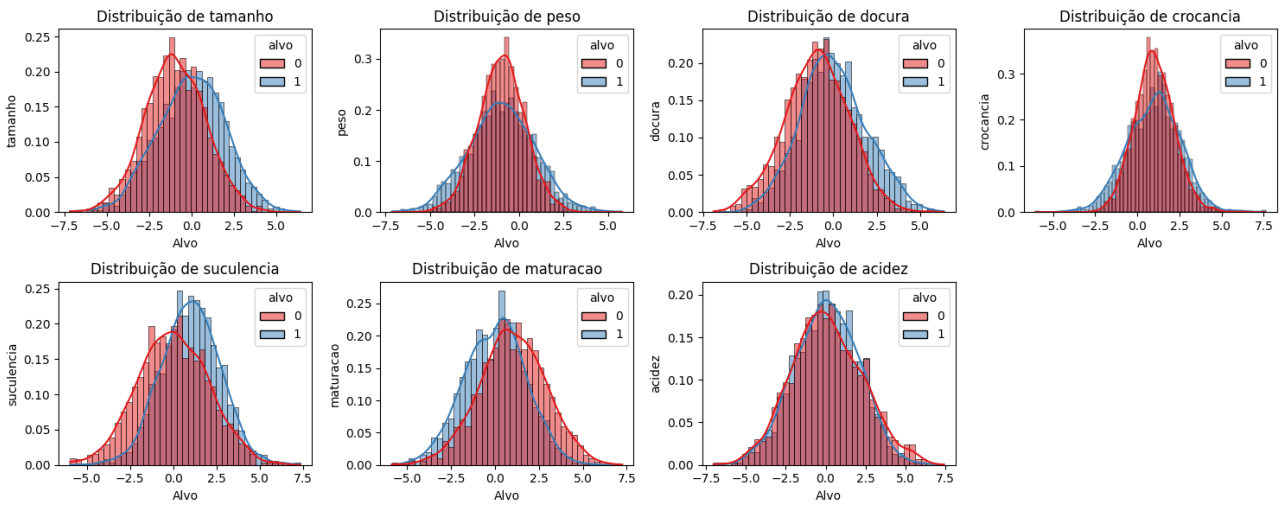
Independente do teste estatístico, define-se que é possível rejeitar H0 caso o p-valor tenha resultado menor ou igual ao nível de significância definido arbitrariamente (FERREIRA, 2016).

3. RESULTADOS E DISCUSSÃO

3.1. ANÁLISE EXPLORATÓRIA DOS DADOS

É possível verificar na Figura 7 que as variáveis disponíveis possuem distribuição aproximadamente normal, tendo variabilidade notável de valores entre classes. Ter *features* com distribuição aproximadamente normal pode ser benéfico para muitos algoritmos de aprendizado de máquina, pois alguns deles costumam ter melhor desempenho quando os dados são aproximadamente normais. Isso ocorre porque esses algoritmos assumem, implicitamente ou explicitamente, que os dados são distribuídos de forma simétrica ou seguem uma distribuição específica. Além disso, Se os dados forem aproximadamente normais, a probabilidade de *outliers* extremos é menor, o que pode ajudar a melhorar a robustez do modelo (HASTIE, 2009).

Figura 7. Histograma de variáveis



Em relação ao poder preditivo, vemos que as melhores variáveis são suculência e maturação. Temos o IV de cada informação na Tabela 3.

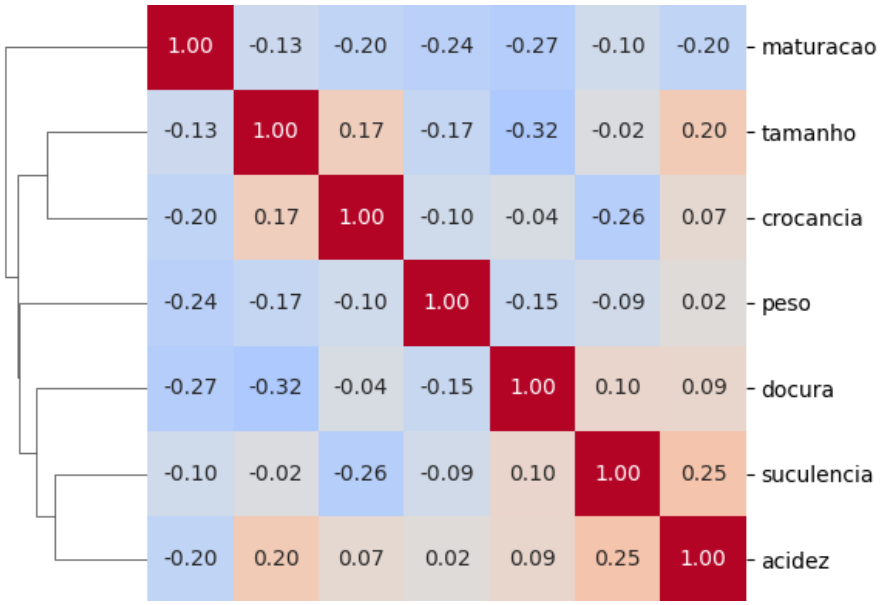
Tabela 3. Resultado do *Information Value* (IV).

Variável	Suculência	Maturação	Tamanho	Doçura	Peso	Crocância	Acidez
IV	0.347576	0.306164	0.276001	0.261441	0.184268	0.153418	0.025452

A suculência de um fruto está fortemente ligada à quantidade de água e ao estado das células da polpa, o que pode ser um indicador direto de sua qualidade e frescor. Como diferentes níveis de suculência são percebidos de forma distinta pelas pessoas (ou por sensores, dependendo do contexto), ela pode ser uma característica marcante para distinguir entre classes (como tipos de frutos, estágios de maturação, etc.) (MOURA, 2011). A maturação está associada a mudanças físicas e químicas substanciais no fruto, como a alteração na cor, textura, acúmulo de açúcares, e mudanças na acidez. Como o estágio de maturação afeta muitos aspectos visíveis e tangíveis, é natural que seja uma característica com forte poder discriminativo entre classes (FERREIRA, 2010). Embora a acidez também mude durante o processo de maturação, essas variações podem ser menos perceptíveis ou menos consistentes para diferentes tipos de frutos ou entre diferentes estágios. A acidez pode não ser tão fácil de medir ou discernir de maneira clara, e sua contribuição para a distinção entre classes pode ser menos expressiva em comparação com características mais visuais ou táteis, como suculência e maturação (MARTINS, 2010).

Em relação à correlação linear das variáveis, na Figura 8, percebe-se que não há alta correlação entre elas. Isso indica que o descarte de *features* não é necessário. Quando as características não apresentam correlação linear entre si, isso indica que cada uma traz informações distintas e complementares ao modelo. Em outras palavras, os dados não possuem redundância significativa. Isso pode ajudar o modelo a aprender padrões mais variados e melhorar sua capacidade de generalização (BERK, 2008).

Figura 8. Mapa de correlação linear de variáveis.



3.2. TREINAMENTO E TESTE

Após treinamento, a avaliação da performance dos modelos a partir das métricas de desempenho foi realizada através dos resultados apresentados nas Tabelas 4 e 5, bem como a avaliação da interpretabilidade dos resultados. A Tabela 4 apresenta as métricas para o modelo RF e a Tabela 5 as métricas para o modelo GBM.

Tabela 4. Métricas de avaliação do modelo *bagging*.

	Precisão	**Recall**	**F1**
Treinamento	0,8748 ± 0,0147	0,8973 ± 0,0115	0,8858 ± 0,0049
Teste	0,8644	0,8992	0,8815

O conjunto de hiperparâmetros para o modelo *bagging* foi: *max_depth*=15, *n_estimators*=280; *max_features*='log2').

Tabela 5. Métricas de avaliação do modelo *boosting.*

	Precisão	**Recall**	**F1**
Treinamento	0,8907 ± 0,0173	0,9004 ± 0,0066	0,8955 ± 0,0108
Teste	0,8802	0,9068	0,8933

O conjunto de hiperparâmetros para o modelo *boosting* foi: *learning_rate*=0.069; *max_depth*=13, *n_estimators*=150; *reg_alpha*=0.8).

Analisando os resultados apresentados nas Tabelas 4 e 5, nota-se que ambos modelos atingiram resultados satisfatórios e poderiam ser utilizados para a tarefa de classificação. O modelo *boosting* apresentou métricas ligeiramente melhores.

Para validar se os resultados são estatisticamente significativos, executou-se o teste de hipótese Qui-quadrado.

- Hipótese nula (H0): a diferença observada nos resultados dos modelos de *bagging* e *boosting* não é estatisticamente significativa.
- Hipótese alternativa (Ha): a diferença observada nos resultados dos modelos de *bagging* e *boosting* é estatisticamente significativa.

O resultado para o teste, a 95% de confiança é de P-valor 0,6358. Dado que o p-valor é maior que o $\alpha = 0.05$, rejeita-se H0.

Por isso, podemos concluir que há diferença entre os modelos treinados, sendo o modelo de *Gradient boosting*, superior ao modelo *bagging*.

As Figuras 9 e 10 apresentam as escalas de importâncias relativas entre as variáveis estudadas pelos modelos. A Figura 9 apresenta o ajuste para o RF e a Figura 10 para o GBM.

Figura 9. Importância das *features* do modelo *bagging*.

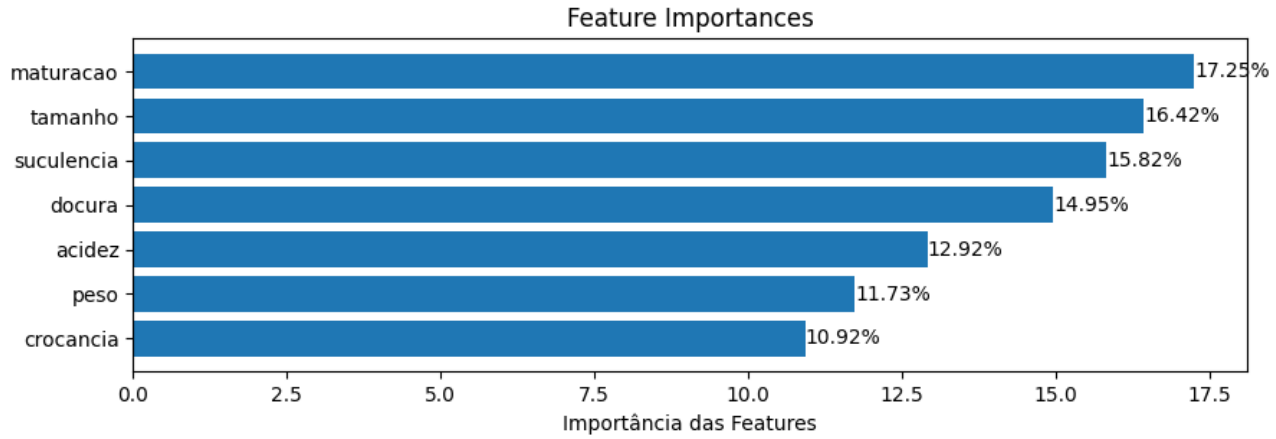
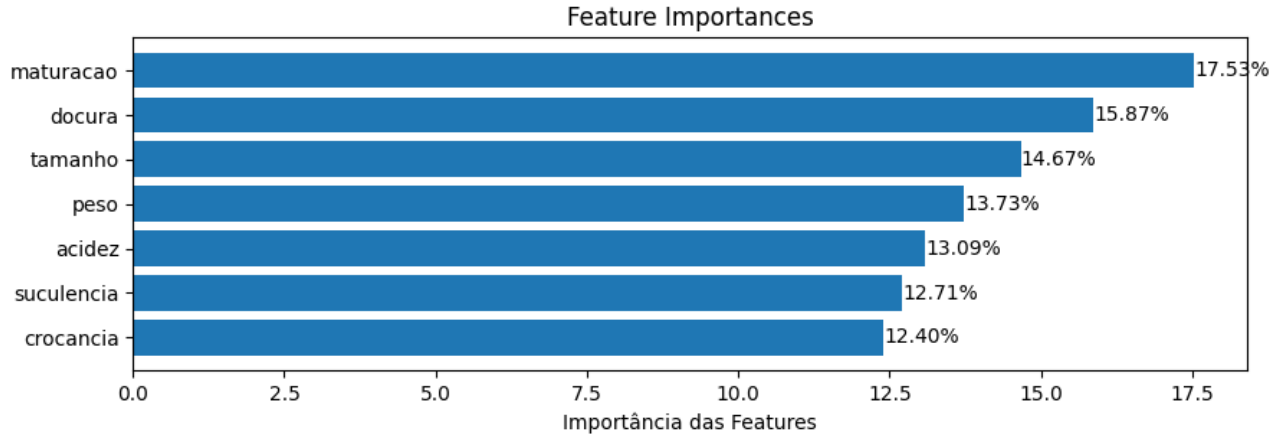


Figura 10. Importância das *features* do modelo *boosting*.



O modelo *boosting* apresentou maior porcentagem de importância para a informação de maturação, enquanto o modelo *bagging* apresentou maior equilíbrio na distribuição de importância ao verificarmos a distribuição dos pontos percentis.

A distribuição igualitária é fundamental para um modelo de predição, pois evita que o algoritmo sofra brusca redução de performance — e consequentemente se torne obsoleto — caso alguma informação sofra desvio em relação aos dados de treinamento. Esse contexto é comum em modelos de *machine learning*, chamados de "*Model Drift*" e "*Data Drift*". De

acordo com Matos (2024):

> "*Model Drift* refere-se à degradação da performance do modelo ao longo do tempo. Isso pode ocorrer quando os padrões nos dados mudam. [...] *Data Drift*, por outro lado, refere-se a mudanças na distribuição dos dados de entrada que alimentam o modelo. Isso pode acontecer devido a mudanças sazonais, tendências de mercado, mudanças no comportamento do usuário ou qualquer outro fator que altere a natureza dos dados." .

Além disso, é evidente que cada modelo gera um ranqueamento diferente de *features*. Por isso, a escolha do algoritmo preditor deve passar não só por validação de métricas de performance como também uma validação manual do agente, uma vez que o modelo deve estar coerente com o contexto de utilização.

A escolha pode passar por critérios como nível de confiabilidade de cada variável como também algum interesse comercial. Por exemplo, caso seja de interesse gerar predições de maior qualidade para maçãs com maior grau de doçura é interessante que o modelo *bagging* seja o escolhido, dado que sua importância para predição é maior, em relação ao outro treinado.

Nas Figuras 11 e 12, percebe-se que o gráfico SHAP de ambos modelos é bem similar, apesar do ranqueamento de variáveis serem distintas. Essa tendência de colaboração positiva ou negativa das *features* é natural, uma vez que os dados de treinamento partem da mesma amostra. No entanto, a ordem de importância está intrinsecamente ligada à forma que as predições são construídas.

Figura 11. SHAP do modelo *bagging*.

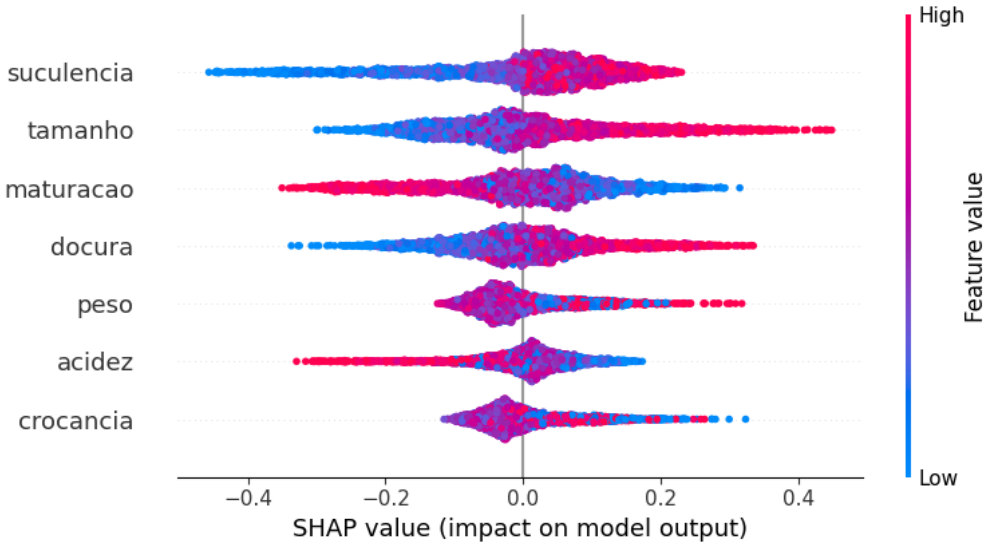
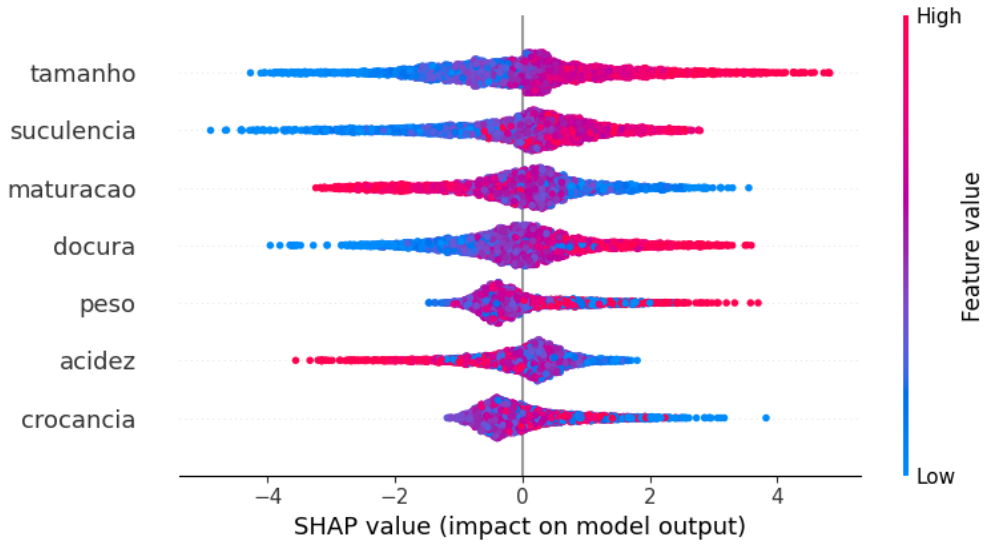


Figura 12. SHAP do modelo *boosting*.



No *bagging*, como as subamostras são treinadas de forma independente, cada modelo treinado terá importâncias distintas de variáveis. Assim, o modelo finalista depende diretamente da média dos treinamentos realizados em paralelo, por combinação dos modelos individuais.

Em comparação, o *boosting* constroi modelos sequenciais, de forma a corrigir os erros anteriores. Dessa forma, as *features* de maior importância são aquelas que geraram os menores erros após as iterações. A importância de características nesse método tende a ser mais adaptativa, refletindo as características que mais contribuíram para reduzir os erros de predição ao longo das iterações.

Enquanto o modelo *boosting* gera um resultado mais robusto ao contexto, o *bagging* gera predições com maior uniformidade e, consequentemente, mais estável. Assim, o modelo de *bagging* pode ser mais resistente ao sobreajuste de dados. Por isso, é fundamental avaliar se o modelo resultante apresenta alta divergência entre os dados de treinamento e teste, como avaliado nas Figuras 13 e 14.

Figura 13. Curva de aprendizado do modelo *bagging*.

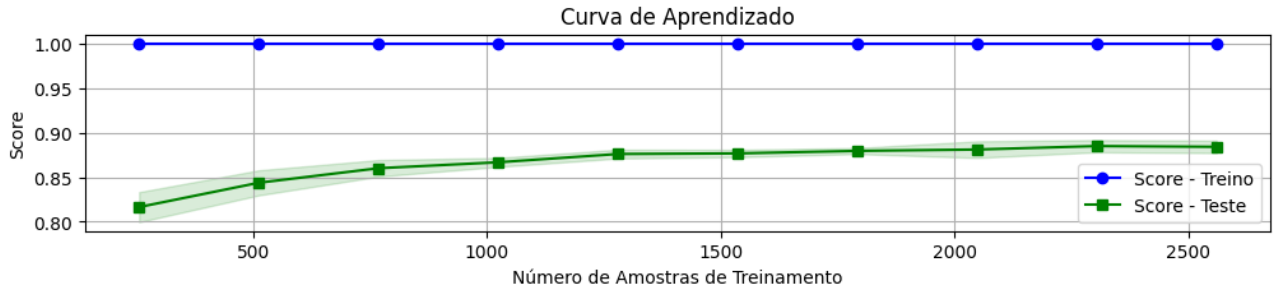
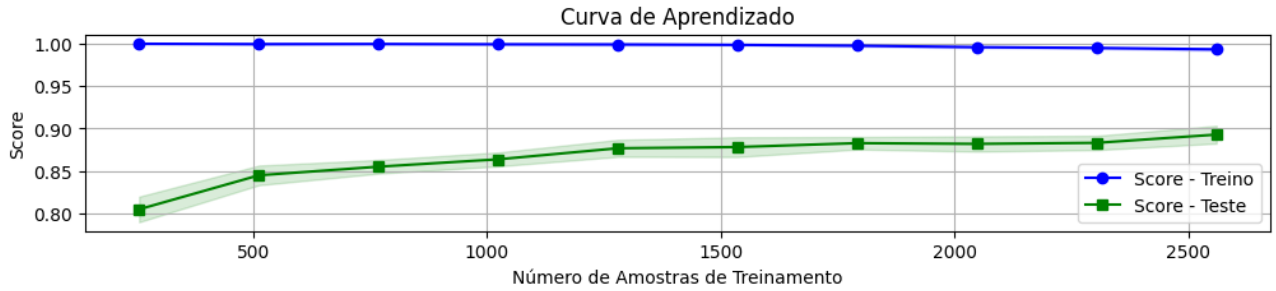


Figura 14. Curva de aprendizado do modelo *boosting*.



A análise das Figuras 13 e 14 indica que a curva de aprendizado de ambos modelos é bem similar, ambos necessitando de poucos dados para atingir o platô e possuindo tendência de convergência entre as acurácias de treinamento e teste.

Dessa forma, como visto nas análises anteriores, o modelo de *boosting* não tem tendência ao sobreajuste, uma vez que não sofre queda das métricas na fase de treinamento. Além disso, técnicas de regularização nos auxiliam na confiabilidade das predições, como os hiperparâmetros de *learning_rate*, *reg_alpha* e *reg_lambda*.

4. CONCLUSÃO

Neste trabalho, analisou-se e comparou-se duas das técnicas mais populares de *ensemble learning*: *bagging* e *boosting*. As abordagens, embora compartilhem a ideia central de combinar múltiplos modelos de árvore de decisão para melhorar a performance preditiva, diferem significativamente em suas estratégias e resultados.

Bagging, representado pelo algoritmo *Random Forest*, demonstrou ser eficaz em reduzir a variância dos modelos, proporcionando estabilidade e robustez. A técnica mostra-se vantajosa em situações onde o modelo base, como a árvore de decisão, é suscetível a variações nos dados de treinamento, beneficiando-se da aleatoriedade na seleção de amostras e características.

Por outro lado, *boosting*, exemplificado pelo algoritmo *Gradient boosting*, mostrou-se eficiente e com performance superior ao modelo *bagging*. *Boosting* tem a capacidade de ajustar-se de forma mais fina aos dados, corrigindo os erros cometidos pelos modelos anteriores na sequência.

A escolha entre *bagging* e *boosting* deve, portanto, ser guiada pelo contexto específico do problema. Se a estabilidade e a simplicidade do modelo são prioridades, *bagging* é a escolha mais apropriada.

Finalmente, a experimentação e a validação cruzada são essenciais para determinar qual técnica oferece o melhor desempenho no contexto do problema estudado, assegurando que a escolha do método de *ensemble* esteja alinhada com as necessidades e os desafios específicos dos dados em questão.

5. REFERÊNCIAS

AMAZON WEB SERVICES, INC. O que é ajuste de hiperparâmetros? Disponível em: <https://aws.amazon.com/pt/what-is/hyperparameter-tuning/>. Acesso em: 9 ago. 2024.

ALICE ZHENG, A. C. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. Sebastopol, CA, USA: O'Reilly Media, 2016.

AKIBA, T. et al. Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). 2019. Disponível em: <https://optuna.org/>. Acesso em: 27 jul. 2024.

BEIGUELMAN, B. 1996. Curso de Bioestatística Básica. 4ed. Ribeirão Preto: Sociedade Brasileira de Genética.

BERGSTRA, JAMES, BARDENET, R. KÉGL, BALÁZS, BENGIO, Y. Algorithms for Hyper-Parameter Optimization. Advances in Neural Information Processing Systems, 2011.

BERK, R. A. Statistical learning from a regression perspective. 1. ed. New York, NY: Springer-Verlag, 2008. (Springer series in statistics). ISBN 9780387775005.

BHEEMAIAH, K.; ESPOSITO, M.; TSE, T. What is machine learning? The Conversation, 3 maio 2017.

BIAGGI, R. Vantangens e desvantagens do Information Value (IV) e Weight of Evidence (WoE). Disponível em: <https://medium.com/@renata-biaggi/vantangens-e-desvantagens-do-information-value-iv-e-weight-of-evidence-woe-47d2cf2362eb>. Acesso em: 29 ago. 2024.

BIAGGI, R. (2022). E.B.A - Estatística do Básico ao Avançado Disponível em: <https://renatabiaggi.com/eba/>. Acesso em: 01 out. 2024.

BREIMAN, L. bagging predictors. Machine Learning, 24, 123-140 (1996), 1996. doi: https://link.springer.com/content/pdf/10.1007/BF00058655.pdf.

BRUCE, A.; BRUCE, P.; GEDEK, P. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python. Sebastopol, CA, USA: O'Reilly Media, 2017.

BSD License, S.-L. sklearn.tree.decisionTreeClassifier. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>. Acesso em: 27

fev. 2024.

CAO, J., QIN, S., YAO, J., ZHANG, C., LIU, G., ZHAO, Y., & ZHANG, R. (2023). Debris flow susceptibility assessment based on information value and machine learning coupling method: from the perspective of sustainable development. Environmental Science and Pollution Research International, 30(37), 87500–87516. https://doi.org/10.1007/s11356-023-28575-w

DE ALMEIDA PRADO ALVES BATISTA, G. E. Pré-processamento de Dados em Aprendizado de Máquina Supervisionado. [s.l.] USP - São Carlos, 3 nov. 2003.

EFRON, B. Estimating the error rate of a prediction rule: Improvement on cross-validation. Journal of the American Statistical Association, v. 78, n. 382, p. 316, 1983.

ELGIRIYEWITHANA, N. Apple Quality. , 11 jan. 2024. Disponível em: <https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality>. Acesso em: 8 out. 2024

FERNANDES, F. M. S. Introdução à Simulação Computacional em Termodinâmica Estatística. 1998. Disponível em: <https://webpages.ciencias.ulisboa.pt/~fmfernandes/Teses/DC4_Curso_Agrega_1998.pdf>.

FERREIRA J. C., PATINO C. M. O que realmente significa o valor-p? J Bras Pneumol. 2016.

FERREIRA, L. de; CARVALHO, A. C. P. Inteligência Artificial: uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2021.

FERREIRA, S. M. R.; QUADROS, D. A. de; KARKLE, E. N. L.; LIMA, J. J. de; TULLIO, L. T.; FREITAS, R. J. S. de. Qualidade pós-colheita do tomate de mesa convencional e orgânico. Food Science and Technology, v. 30, n. 4, p. 858–869, 2010. Disponível em: https://doi.org/10.1590/s0101-20612010000400004.

FIGUEIREDO FILHO, D. B.; SILVA JÚNIOR, J. A. Desvendando os mistérios do coeficiente de correlação de Pearson. Revista Política Hoje, Pernambuco, 2009.

FONTES, L. C. B.; SARMENTO, S. B. S.; SPOTO, M. H. F. Características sensoriais e microbiológicas de maçãs minimamente processadas recobertas com películas. Food Science and Technology, v. 27, n. 1, p. 91–98, 2007. DOI: 10.1590/s0101-20612007000100016. Disponível em: https://doi.org/10.1590/s0101-20612007000100016. Acesso em: 20 out. 2024.

GARCIA, C. Modelagem e Simulação de Processos Industriais e de Sistemas Eletromecânicos. Editora da Universidade de São Paulo, 2009.

GERON, A. Mãos à Obra: Aprendizado de Máquina com scikit-learn, Keras & TensorFlow: Conceitos, Ferramentas e Técnicas Para a Construção de Sistemas Inteligentes. Sebastopol, CA, USA: O'Reilly Media, 2017.

GRUS, J. Data Science from Scratch: First Principles with Python. Sebastopol, CA, USA: O'Reilly Media, 2015.

HASTIE, T., TIBSHIRANI, R., & FRIEDMAN, J. (2009). The elements of statistical learning. Springer New York.

HIMMELBLAU, D. M., & BISCHOFF, K. B. (1968). Process analysis and simulation - deterministic systems. First edition edn. John Wiley & Sons.

KAMPAKIS, S. (STELIOS). How to use learning curves in scikit-learn. Disponível em: <https://thedatascientist.com/learning-curves-scikit-learn/>. Acesso em: 8 out. 2024.

LJUNG, L. System Identification: Theory for the User. NJ, USA: Prentice Hall, 1987.

Machine Learning: conhecendo as técnicas de bagging e boosting. Disponível em: <https://www.alura.com.br/artigos/machine-learning-tecnicas-bagging-boosting?srsltid=AfmBOopDy5QRRcySG5MaHOOyMS8EuOJbzA1Jgkd_bTohunj6J_1NdJ0H>. Acesso em: 29 out. 2024.

MARIANO, D. Métricas de avaliação em machine learning: acurácia, sensibilidade, precisão, especificidade e F-score. Revista Brasileira de Bioinformática e Biologia Computacional, 2021.

MARTINS, C. R.; FARIA, J. C.; ROMBALDI, C. V.; FARIAS, R. M. Qualidade sensorial de maçãs produzidas em diferentes sistemas de produção. Scientia Agraria, v. 11, n. 2, p. 91-99, 2010. Disponível em: https://doi.org/10.1590/s0101-20612010000400004.

MATOS, D. Importância de Model e Data Drift no Contexto de Machine Learning. Disponível em:<https://www.cienciaedados.com/importancia-de-model-e-data-drift-no-contexto-de-machine-learning/>. Acesso em: 3 set. 2024.

MATTOS, M. Árvores de Decisão do Zero usando Python e NumPy com visualização

interativa. Disponível em: <https://github.com/matheuscamposmt/decision-trees?tab=readme- ov-file>. Acesso em: 4 mar. 2024.

MOURA, R. S.; STORCH, T. T.; CERO, J. D.; GIRARDI, C. L.; ROMBALDI, C. V. Avaliação da suculência de maçãs, cv. Gala, tratadas com 1-MCP e armazenadas sob atmosfera refrigerada. Embrapa, 2011. Disponível em: https://www.alice.cnptia.embrapa.br/alice/bitstream/doc/906704/1/P363.pdf. Acesso em: 25 out. 2024.

MICROSOFT. LGBM Parameters. 2019. Microsoft Open Source Code of Conduct. Disponível em: <https://lightgbm.readthedocs.io/en/latest/>. Acesso em: ago. 2024.

MENGNAN, D.; NINGHAO, L.; XIA, H. Techniques for Interpretable Machine Learning. arXiv, 2018. Disponível em: <https://arxiv.org/abs/1808.00033>.

MUKHIYA, Suresh Kumar; AHMED, Usman. Hands-On Exploratory Data Analysis with Python. 1th. ed. BIRMINGHAM: Packt, 2020.

PANNELL, D. (1997). Sensitivity analysis of normative economic models: Theoretical framework and practical strategies. Agricultural economics, 16, 139–152.

PEDREGOSA, F. et al. scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 27 jul. 2024.

PEDRO A. MORETTIN, J. M. S. Estatística e Ciência de Dados Versão preliminar. 2021. Disponível em: <https://www.ime.usp.br/~pam/cdadosf3.pdf>.

PINHEIRO, J. M. H. Um estudo sobre Algoritmos de boosting e a Otimização de Hiperparâmetros Utilizando Optuna. São Carlos, SP., 2023.

REIS, K. Como eu gostaria que alguém me explicasse SHAP values. BIX tecnologia, 2021. Disponível em: <https://bixtecnologia.com.br/como-eu-gostaria-que-alguem-me-explicasse-shap-values/>. Acesso em: 2 set. 2024.

RISHABH MANOJ, AI TEAM. Optuna — An Automatic Hyperparameter Optimization Framework. Medium, 2022. Disponível em: <https://medium.com/@publiciscommerce/optuna-an-automatic-hyperparameter-optimization-framework-f64638621ff7>. Acesso em: 2 set. 2024

RODRIGUES, RAIMUNDO NONATO. Avaliação de empresas sob condições de Risco. 2003. Disponível em: <http://www.academia.edu/7142866/AVALIA%C3%87%C3%83O_SOB_ RISCO>. Acesso em: 27 jul. 2024.

SAHOUR, H., GHOLAMI, V., TORKAMAN, J., VAZIFEDAN, M., & SAEEDI, S. (2021). Random forest and extreme gradient boosting algorithms for streamflow modeling using vessel features and tree-rings. Environmental Earth Sciences, 80(22). https://doi.org/10.1007/s12665-021-10054-5

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. Process dynamics and control. 2. ed. [s.l.] John Wiley & Sons, 2009.

SPROVIERI, P. F.; COMELLI, C. F. Algoritmo determinístico: uma base para a implementação de aplicativos educacionais que auxiliem o processo ensino-aprendizagem na resolução de problemas de programação linear de duas variáveis. Revista Interface Tecnológica, [S. l.], v. 18, n. 2, p. 290–303, 2021. DOI: 10.31510/infa.v18i2.1303. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/article/view/1303. Acesso em: 27 jul. 2024.

STARMER, J. The StatQuest Illustrated Guide To Machine Learning. Birmingham, England: Packt Publishing, 2022.

VICINI, L. ANÁLISE MULTIVARIADA DA TEORIA À PRÁTICA. Santa Maria, RS, Brasil: Biblioteca Central da UFSM, 2005.

YADAV, S.; SHUKLA, S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. 2016 IEEE 6th International Conference on Advanced Computing (IACC). Anais...IEEE, 2016.

Y. M. E. CHA ZHANG. Ensemble machine learning: methods and applications. Springer Science & Business Media, 2012.

ZIBETTI, A. Teste de Hipóteses. Florianópolis: Departamento de Informática e Estatística, UFSC, 2020. Disponível em:. Inf. Ufsc. Br/~andre. Zibetti/probabilidade/teste-de-hipoteses. Html. Acesso em: 02 set. 2024.

ZUBEN, C. J. et al. Theoretical approaches to forensic entomology: I. Mathematical model of

postfeeding larval dispersal. Zeitschrift für angewandte Entomologie [Journal of applied entomology], v. 120, n. 1–5, p. 379–382, 1996.