UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL

NICOLAS DE OLIVEIRA LOPES BRAGA
JOÃO VITOR NANTES DA SILVA MATOS

ANÁLISE DE ALGORITMOS DE MACHINE LEARNING UTILIZANDO A BIBLIOTECA PYCARET PARA DETECÇÃO DE FRAUDES EM TRANSAÇÕES ONLINE DE CARTÃO DE CRÉDITO

CAMPO GRANDE
2023

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL

NICOLAS DE OLIVEIRA LOPES BRAGA
JOÃO VITOR NANTES DA SILVA MATOS

ANÁLISE DE ALGORITMOS DE MACHINE LEARNING UTILIZANDO A BIBLIOTECA PYCARET PARA DETECÇÃO DE FRAUDES EM TRANSAÇÕES ONLINE DE CARTÃO DE CRÉDITO

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Mato Grosso do Sul como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação, na área de Análise de Dados.

Orientador (a): Dionisio Machado Leite Filho

CAMPO GRANDE
2023

AGRADECIMENTOS

A conclusão deste trabalho representa o término de uma jornada intensa e desafiadora, e é com profunda gratidão que reconhecemos todos aqueles que contribuíram para o êxito deste projeto.

Em primeiro lugar, expressamos nossa sincera gratidão à nossa família, cujo apoio inabalável foi uma âncora vital ao longo desta jornada acadêmica. Seu incentivo e compreensão foram a força motriz que nos impulsionou nos momentos mais difíceis.

Ao nosso orientador, Dionísio Machado Leite Filho, dirigimos nossos agradecimentos pela orientação valiosa, apoio incansável e insights que foram fundamentais para o desenvolvimento deste trabalho. Estendemos nossos agradecimentos aos professores e profissionais que compartilharam conosco esta trajetória, enriquecendo nossa jornada acadêmica com conhecimentos valiosos.

À Universidade Federal de Mato Grosso do Sul, expressamos nossa gratidão pela infraestrutura fornecida e apoio ao longo destes anos. Sua contribuição transcendeu os limites da sala de aula, proporcionando um ambiente propício ao crescimento e aprendizado.

Por fim, agradecemos um ao outro, pois reconhecemos que nossa parceria foi essencial para a conclusão bem-sucedida deste trabalho. Ao longo de todos esses anos, não poderíamos imaginar um encerramento mais significativo.

RESUMO

A análise de algoritmos de *machine learning* para detecção de fraudes em transações online de cartão de crédito é um campo crucial na segurança financeira. A detecção de fraudes em transações online de cartão de crédito é desafiadora devido à evolução constante das técnicas fraudulentas. Assim, como metodologia, foi utilizada a biblioteca PyCaret. Essa biblioteca permitiu a experimentação com diferentes modelos de *machine learning*, como *Random Forest*, *Gradient Boosting* e *Support Vector Machines*, otimizando a precisão na detecção de atividades fraudulentas. Além disso, a biblioteca permitiu a interpretação dos resultados e visualizações, auxiliando na compreensão do desempenho dos algoritmos.

Palavras-chave: Machine Learning. PyCaret. Detecção de Fraudes. Transações Online. Cartão de Crédito.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

Segundo o dicionário Oxford, fraude é o crime de enganar alguém para obter dinheiro ou bens ilegalmente. Existem diversos tipos de fraude; conforme o estudo do Serasa, os crimes que mais preocupam atualmente são: fraude de identidade, fraude financeira, vazamento de dados e o uso de documentos falsos (ROCHA, 2023).

A análise de algoritmos de *machine learning* é uma parte fundamental do desenvolvimento de soluções eficientes para a detecção de fraudes em transações online de cartão de crédito. A utilização da biblioteca PyCaret simplifica e acelera esse processo, permitindo aos cientistas de dados e desenvolvedores explorar diversas opções de algoritmos e ajustar parâmetros com facilidade (VELOSO, 2023).

A detecção de fraudes em transações online de cartão de crédito é uma preocupação crescente, dada a constante evolução das técnicas utilizadas pelos fraudadores. Nesse cenário, a análise de algoritmos de *machine learning* desempenha um papel crucial, uma vez que pode identificar padrões complexos e anomalias que escapam à detecção humana (DUARTE, 2022).

Dessa forma, visando a utilização de várias técnicas de *machine learning*, foi utilizado, neste trabalho, a biblioteca PyCaret. Trata-se de uma biblioteca de código aberto em Python que simplifica o fluxo de trabalho de *machine learning*. Ela oferece uma interface simples e fácil de usar, permitindo aos profissionais de dados realizar tarefas complexas com poucas linhas de código. Uma das principais vantagens do PyCaret é a automação de diversas etapas do processo, como pré-processamento de dados, seleção de modelos, ajuste de hiperparâmetros e avaliação de desempenho (MALTA, 2021).

A preparação dos dados foi realizada utilizando a biblioteca PyCaret. A biblioteca fornece ferramentas visuais que auxiliam na compreensão da distribuição dos dados e na identificação de possíveis outliers. A seleção de modelos foi realizada via biblioteca PyCaret, uma vez que ela disponibiliza todo o processo de comparação dos modelos de *machine learning* com uma única linha de código.

Os modelos treinados foram avaliados usando métricas específicas para detecção de anomalias, como precisão, recall e F1-score. A biblioteca também forneceu visualizações intuitivas, como matrizes de confusão e curvas ROC, para auxiliar na interpretação dos resultados.

1.1. PROBLEMA

A detecção eficaz de fraudes bancárias tornou-se uma tarefa complexa, uma vez que os métodos tradicionais muitas vezes se mostram insuficientes diante da sofisticação crescente dos fraudadores. Sendo assim, este trabalho visou abordar o problema das fraudes bancárias em transações com cartão de crédito por meio da análise de diversos algoritmos de *machine learning*. Esta pesquisa buscou identificar quais modelos apresentam melhor desempenho na detecção de padrões suspeitos nas transações, considerando diversos fatores.

1.2. OBJETIVOS

1.2.1. Objetivo geral

Este trabalho tem como objetivo geral realizar a análise dos algoritmos de *machine learning* disponíveis na biblioteca PyCaret e desenvolver um modelo para combater fraudes bancárias nas operações de cartão de crédito.

1.2.2. Objetivos específicos

- Realizar uma revisão abrangente da bibliografia existente sobre detecção de fraudes bancárias, a fim de compreender as tendências e avanços na área;
- Identificar qual algoritmo de *Machine Learning* mais se adequa ao problema apresentado;
- Realizar o tratamento de dados no *dataset* utilizado e o balanceamento;
- Aplicar técnicas de avaliação dos modelos testados;
- Desenvolver um modelo baseado no estudo realizado.

2. METODOLOGIA

Este estudo explorou a abordagem das fraudes bancárias na literatura, onde em seguida, foram pesquisadas bases de dados que refletissem esse cenário. Uma biblioteca foi empregada no desenvolvimento de vários modelos simultâneos, usando configurações específicas. Para validar os resultados, foram realizados testes, e validações cruzadas foram aplicadas para garantir a robustez das técnicas utilizadas.

A necessidade de garantir a representatividade em análises de detecção de fraudes em transações de cartão de crédito é crucial para o desenvolvimento de modelos robustos e eficazes. O ponto inicial desse processo é a cuidadosa seleção de um conjunto de dados relevante e significativo. No entanto, dada a natureza sensível dessas informações, é imperativo adotar medidas rigorosas para preservar a privacidade dos dados. Optamos, portanto, pela utilização da técnica de Análise de Componentes Principais (PCA), uma abordagem que não apenas preserva a representatividade estatística, mas também remove informações originais, garantindo confidencialidade.

A qualidade dos dados desempenha um papel central na eficácia do modelo. Para assegurar isso, implementamos técnicas avançadas de pré-processamento. Isso inclui não apenas o tratamento de valores ausentes, mas também a remoção de duplicatas, o tratamento de outliers e o balanceamento adequado do conjunto de dados. Essas etapas são fundamentais para garantir que o modelo seja treinado com informações confiáveis e representativas da complexa realidade das transações de cartão de crédito.

A escolha da biblioteca PyCaret representa uma abordagem estratégica para a implementação eficiente de uma variedade de algoritmos de *Machine Learning*. Essa ferramenta não apenas simplifica o processo, mas também oferece flexibilidade para explorar modelos como Logistic Regression, Ridge Classifier, Naive Bayes, K Neighbors, Decision Tree e Extreme Gradient Boosting. A agilidade proporcionada pelo PyCaret é crucial, otimizando o tempo e os recursos dedicados ao desenvolvimento do modelo.

No processo de treinamento e avaliação dos modelos, a análise será conduzida utilizando diversas métricas de desempenho. Matrizes de confusão, precisão, recall, F1-score, área sob a curva ROC e outras métricas fornecerão uma visão abrangente sobre como os modelos se comportam em relação à detecção de fraudes. Essa fase crítica permitirá a identificação de pontos fortes e áreas de melhoria em cada modelo.

A análise detalhada dos modelos selecionados será a última etapa do processo. Compreender a eficácia de cada algoritmo na detecção de fraudes exigirá uma abordagem holística, considerando não apenas as métricas de desempenho, mas também fatores como a interpretabilidade do modelo e capacidade de generalização. Essa análise aprofundada será crucial para a seleção do modelo ideal, que equilibra precisão, confiabilidade e aplicabilidade prática na detecção de fraudes em transações online de cartão de crédito.

Em resumo, a abordagem proposta, que combina a seleção cuidadosa do conjunto de dados, um pré-processamento robusto, a utilização eficiente da biblioteca PyCaret e uma análise detalhada dos resultados, visa desenvolver um modelo de detecção de fraudes sólido e eficaz para transações online de cartão de crédito.

3. REVISÃO BIBLIOGRÁFICA

3.1. PYCARET

PyCaret é uma biblioteca de *machine learning* em Python que se destaca por ser de código aberto e de baixo código, automatizando fluxos de trabalho e acelerando significativamente os ciclos de experimentação. Funcionando como uma ferramenta abrangente para *machine learning* e gestão de modelos, PyCaret simplifica a implementação de modelos complexos, substituindo centenas de linhas de código por apenas algumas (WHIG, 2023).

Em comparação com outras bibliotecas de *machine learning* de código aberto, PyCaret se destaca como uma opção de baixo código, tornando os experimentos não apenas mais rápidos, mas também mais eficientes. PyCaret atua como um wrapper em torno de diversas bibliotecas e frameworks de *machine learning*, incluindo scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, entre outros (IQBAL, 2021).

O design e a simplicidade do PyCaret são influenciados pelo surgimento dos "cientistas de dados cidadãos", um termo cunhado pelo Gartner. Esses são usuários avançados capazes de realizar tarefas analíticas que, anteriormente, exigiriam um conhecimento técnico mais aprofundado. Em essência, PyCaret busca democratizar o acesso ao *machine learning*, facilitando a vida tanto de especialistas quanto de usuários menos técnicos (ARAÚJO, 2022).

3.2. TÉCNICAS DE PRÉ-PROCESSAMENTO E TRATAMENTO DE DADOS SENSÍVEIS

A sensibilidade dos dados de transações de cartão de crédito demanda precauções específicas no tratamento e pré-processamento, visando preservar a privacidade e a confidencialidade. Dentre as técnicas frequentemente empregadas, a Análise de Componentes Principais (PCA) destaca-se como uma abordagem comum para reduzir a dimensionalidade dos dados, mantendo, ao mesmo tempo, sua estrutura estatística (SILVA, 2023).

A utilização da PCA é crucial para proteger informações sensíveis, uma vez que ela permite a transformação dos dados originais em um conjunto reduzido de componentes, mantendo a maior parte da variância dos dados. Isso não apenas preserva

a integridade estatística dos dados, mas também minimiza o risco de exposição de informações sensíveis durante a análise e treinamento de modelos (HOSOUME, 2020).

Além da PCA, a aplicação de técnicas avançadas de pré-processamento é essencial para garantir a qualidade dos dados antes de serem submetidos aos algoritmos de *Machine Learning*. O tratamento de valores ausentes é uma etapa crítica, visando evitar distorções nos resultados e garantir que as análises se baseiem em dados completos e confiáveis. A remoção de duplicatas é outra prática importante, contribuindo para a consistência e integridade dos dados (FRANCO, 2020).

O cuidado com a privacidade não se limita apenas ao processamento dos dados, estendendo-se também à manipulação de informações ausentes. Estratégias como a imputação de dados faltantes podem ser empregadas de forma ponderada, garantindo que a integridade das informações seja mantida sem comprometer a segurança dos dados sensíveis (REAL; NICOLETTI, 2014).

A combinação de técnicas como PCA, tratamento de valores ausentes e remoção de duplicatas não apenas assegura a privacidade dos dados, mas também cria um ambiente propício para a aplicação eficaz de algoritmos de *Machine Learning*. O equilíbrio entre a preservação da sensibilidade dos dados e a eficácia da análise é crucial para o desenvolvimento de modelos robustos de detecção de fraudes em transações de cartão de crédito, garantindo a confiabilidade e ética no manuseio dessas informações sensíveis (VIACAVA et al., 2012).

3.3. AVALIAÇÃO DE DESEMPENHO E SELEÇÃO DO MODELO

A etapa de avaliação de desempenho dos modelos representa um ponto crucial no desenvolvimento de sistemas de detecção de fraudes em transações de cartão de crédito. Diversas métricas são empregadas para fornecer uma visão abrangente do quão bem um modelo está executando suas predições. Entre essas métricas, destacam-se a matriz de confusão, precisão, recall, F1-score e a área sob a curva ROC (JUNIOR; CARPINETTI, 2019).

A matriz de confusão é uma ferramenta valiosa para avaliar o desempenho do modelo, fornecendo uma visão detalhada dos resultados, destacando verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. A precisão é uma medida geral da acurácia do modelo, enquanto o recall destaca a capacidade do modelo em identificar corretamente as instâncias positivas. O F1-score combina precisão e recall,

proporcionando uma métrica balanceada entre esses dois aspectos (DOMINGUES; PEDROSA; BERNARDINO, 2020).

A área sob a curva ROC (Receiver Operating Characteristic) é uma métrica especialmente útil para avaliar o desempenho de modelos de detecção de fraudes, uma vez que analisa a taxa de verdadeiros positivos em relação à taxa de falsos positivos em diferentes pontos de corte. Quanto maior a área sob a curva ROC, melhor o desempenho do modelo (CASTRO; BRAGA, 2011).

Através de gráficos e relatórios gerados pelo PyCaret, os desenvolvedores e cientistas de dados podem rapidamente comparar o desempenho de diferentes modelos, facilitando a identificação daquele que melhor atende aos requisitos específicos do problema em questão. Essa abordagem simplificada agiliza o ciclo de desenvolvimento e permite uma tomada de decisão informada, resultando na seleção do modelo mais eficaz para a detecção de fraudes em transações online de cartão de crédito (GARCIA et al., 2015).

3.4. A TÉCNICA SMOTE

A técnica SMOTE (Synthetic Minority Over-sampling Technique), empregada no *machine learning*, aborda o problema do desequilíbrio de classes em tarefas de classificação. Esse desequilíbrio surge quando uma classe possui muito menos instâncias do que outra, levando a modelos tendenciosos e com desempenho deficiente na classe minoritária (Alves et al., 2021).

Segundo Moreno et al. (2009), o SMOTE soluciona esse problema gerando exemplos sintéticos da classe minoritária para equilibrar a distribuição de classes. A técnica opera da seguinte forma:

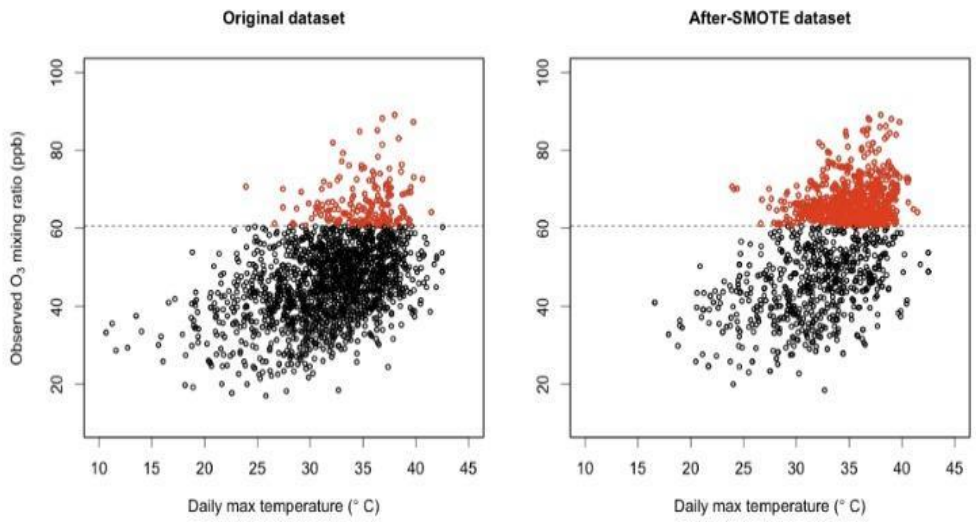
- Identificação da Classe Minoritária: Determine qual classe possui menos instâncias no seu conjunto de dados.
- Seleção de uma Instância da Classe Minoritária: Escolha aleatoriamente uma instância dessa classe.
- Encontre os k-Vizinhos Mais Próximos: Identifique os k-vizinhos mais próximos para a instância selecionada, geralmente utilizando a distância euclidiana como métrica.
- Criação de Instâncias Sintéticas: Para cada um dos k-vizinhos mais próximos, crie instâncias sintéticas, interpolando entre a instância selecionada e seus vizinhos. Isso é feito multiplicando a diferença entre os

valores das características da instância e seu vizinho por um número aleatório entre 0 e 1, e depois adicionando à instância selecionada.

- Repetição dos Passos: Repita os passos anteriores até alcançar um equilíbrio desejado entre as classes minoritária e majoritária.

O SMOTE visa evitar o viés em direção à classe majoritária, melhorando a generalização do modelo. É essencial ressaltar que a eficácia do SMOTE varia conforme a figura 1, que mostra o resultado da aplicação da técnica SMOTE.

Figura 1. Exemplo de aplicação da técnica Smote



3.5. EXTRA TREES CLASSIFIER

O *Extra Trees Classifier* representa uma evolução na abordagem de *machine learning*, inserindo-se na família de métodos *ensemble*, notadamente como uma variação dos *Random Forests*. O cerne desse algoritmo reside na construção de árvores de decisão, adotando uma estratégia única que se diferencia do seu predecessor (BABY et al., 2021).

Ao contrário do *Random Forest*, o *Extra Trees* utiliza todas as características disponíveis para a divisão em cada nó da árvore, eliminando a aleatoriedade na escolha de subconjuntos de características. Além disso, ao determinar os umbrais para cada característica, o *Extra Trees* introduz uma camada adicional de aleatoriedade, conferindo uma abordagem distinta na construção das árvores de decisão (SHARAFF; GUPTA, 2018).

Uma das características notáveis do *Extra Trees* é a eficiência computacional decorrente do uso abrangente de características em cada decisão de nó. Essa eficiência se

traduz em vantagens práticas, especialmente em conjuntos de dados extensos, onde a construção de modelos pode ser computacionalmente intensiva (BHATI; RAI, 2020).

A busca pela redução de variância, uma característica central dos métodos ensemble, é uma forte vantagem do Extra Trees. Ao combinar diversas árvores de decisão, o modelo resultante torna-se mais robusto, mitigando riscos de overfitting e proporcionando uma generalização mais sólida para novos dados (SHARMA; KUMAR; JAIN, 2022).

A inclusão de aleatoriedade na escolha de características e umbrais não apenas confere ao modelo uma estabilidade adicional, mas também contribui para torná-lo menos suscetível a pequenas variações nos dados de treinamento. Essa estabilidade é crucial para garantir que o modelo mantenha um desempenho consistente em diferentes conjuntos de dados (DÉSIR et al., 2012).

Com sua eficácia em tarefas de classificação e a relativa facilidade de uso, o Extra Trees Classifier oferece uma abordagem sólida e eficiente para uma variedade de problemas em *machine learning*. Seja na análise de dados ou em contextos mais complexos, a implementação criteriosa desse algoritmo, ajustando hiperparâmetros conforme necessário, pode resultar em modelos confiáveis e generalizáveis (ROCHA et al., 2023).

3.6. RANDOM FOREST CLASSIFIER

O *Random Forest Classifier* é um poderoso algoritmo de *machine learning* que faz parte da família de métodos *ensemble*. Desenvolvido com base em árvores de decisão, esse algoritmo se destaca por sua capacidade de construir múltiplas árvores e combinar seus resultados para obter uma previsão mais robusta e precisa (PAL, 2005).

A característica fundamental do *Random Forest* reside na criação de uma "floresta" composta por um conjunto de árvores de decisão independentes. Cada árvore é treinada em um subconjunto aleatório dos dados de treinamento, e suas decisões são combinadas através de um processo de votação (no caso de classificação) ou média (no caso de regressão) para produzir a saída final do modelo (RODRIGUEZ-GALIANO, 2012).

Uma das vantagens proeminentes do *Random Forest* é a capacidade de lidar com conjuntos de dados grandes e complexos, mantendo uma boa generalização. Ao treinar árvores em subconjuntos aleatórios dos dados, o modelo evita o *overfitting*, tornando-se mais robusto e resistente a variações nos dados (AZAR et al., 2014).

Além disso, o *Random Forest* é eficaz na identificação de características importantes. Durante o treinamento, o algoritmo avalia a importância de cada característica, fornecendo *insights* valiosos sobre quais variáveis têm maior influência nas decisões do modelo (DEVETYAROV; NOURETDINOV, 2010).

Outra característica relevante é a facilidade de ajuste de hiperparâmetros. Parâmetros como o número de árvores na floresta, a profundidade máxima das árvores e o número de características consideradas em cada divisão podem ser ajustados para otimizar o desempenho do modelo para diferentes conjuntos de dados (KULKARNI; SINHA, 2023).

O Random Forest tem aplicações em uma variedade de domínios, incluindo classificação, regressão e detecção de anomalias. Sua versatilidade, capacidade de lidar com grandes volumes de dados e habilidade para reduzir o *overfitting* tornam-no uma escolha popular em muitos cenários de aprendizado de máquina (CHAUDHARY; KOLHE; KAMAL, 2016).

3.7. EXTREME GRADIENT BOOSTING

O *Extreme Gradient Boosting* (XGBoost) representa um avanço notável na arena do *machine learning*, sendo uma técnica de *gradient boosting* conhecida por sua eficiência, desempenho superior e aplicabilidade versátil em diversas tarefas. Desenvolvido para superar limitações de implementações anteriores de *gradient boosting*, o XGBoost ganhou destaque por sua capacidade de lidar com grandes conjuntos de dados e sua eficácia em termos de velocidade de treinamento e generalização (CHEN, 2016).

Uma característica central do XGBoost é sua abordagem extrema para lidar com *overfitting* e melhorar a eficiência do treinamento. Ao treinar uma sequência de modelos de aprendizado fraco, frequentemente árvores de decisão, o XGBoost busca corrigir os erros do modelo anterior, aprimorando gradualmente a precisão e robustez do modelo final (CARMONA; CLIMENT; MOMPARLER, 2019).

A inclusão de técnicas avançadas de regularização, como termos de penalização, destaca-se como uma estratégia fundamental do XGBoost para controlar a complexidade do modelo e mitigar o *overfitting*. Além disso, a personalização da função objetivo, adaptada para diferentes tipos de problemas, proporciona uma flexibilidade que abrange desde classificação até regressão e tarefas mais complexas (CHANG; WU, 2018).

A eficiência computacional do XGBoost é reconhecida, otimizada para paralelismo eficaz, o que o torna escalável e capaz de lidar com volumes substanciais de

dados de maneira eficiente. Sua capacidade de lidar diretamente com valores ausentes e fornecer pontuações de importância para cada característica contribui para a interpretabilidade do modelo (FAN et al., 2018).

A aplicação prática do XGBoost é diversificada, estendendo-se desde competições de ciência de dados até ambientes de produção, onde sua confiabilidade e desempenho superior são reconhecidos. Sua posição proeminente no campo do aprendizado de máquina reflete um equilíbrio notável entre robustez, eficiência e flexibilidade, consolidando o XGBoost como uma ferramenta essencial para enfrentar desafios complexos em análise de dados e modelagem preditiva (OSMAN et al., 2021).

3.8. LIGHT GRADIENT BOOSTING MACHINE

A *Light Gradient Boosting Machine* (LightGBM) representa uma evolução significativa na paisagem do *machine learning*, integrando-se à categoria de algoritmos de *gradient boosting*. Reconhecido por sua abordagem inovadora e eficiência computacional, o *LightGBM* oferece uma alternativa poderosa e flexível para a construção de modelos preditivos em uma variedade de contextos (FAN et al., 2019).

Um dos traços distintivos do *LightGBM* é a adoção do particionamento em nível de folha, em contraste com a abordagem tradicional de crescimento em largura. Essa estratégia concentra a construção da árvore nas folhas que mais contribuem para a redução da função objetivo, resultando em um processo de treinamento mais eficiente.

A utilização de histogramas de gradiente para a seleção de características é outra característica notável do *LightGBM*. Essa técnica acelera a tomada de decisões sobre divisões nas árvores, proporcionando ganhos significativos em termos de velocidade de treinamento, especialmente em conjuntos de dados grandes (TAHA; MALEBARY, 2020).

A eficiência do *LightGBM* na manipulação de dados esparsos é um diferencial, tornando-o uma escolha adequada para cenários nos quais os conjuntos de dados apresentam muitos valores ausentes. Sua capacidade de lidar com dados esparsos contribui para a aplicabilidade do *LightGBM* em uma variedade de problemas do mundo real (GUO et al., 2023).

Assim como o XGBoost, o *LightGBM* é otimizado para paralelismo eficaz, tornando-o escalável e capaz de lidar com volumes substanciais de dados. A flexibilidade proporcionada por uma variedade de hiperparâmetros configuráveis permite a adaptação do modelo para diferentes tipos de dados e problemas (LI et al., 2018).

A aplicação do *LightGBM* abrange diversas tarefas, desde classificação até regressão e ranking. Sua performance competitiva em competições de ciência de dados e sua eficiência em ambientes práticos consolidam o LightGBM como uma escolha robusta e eficaz no cenário do *machine learning* moderno. Seu design inovador e abordagem eficiente continuam a posicionar o LightGBM como uma ferramenta valiosa para desafios complexos em modelagem preditiva e análise de dados (DENG et al., 2018).

3.9. RIDGE CLASSIFIER

O *Ridge Classifier* é um modelo utilizado em *machine learning*, especialmente em problemas de classificação. Ele é uma variação do algoritmo de regressão linear regularizado chamado *Ridge Regression*. O *Ridge Classifier* é frequentemente empregado quando se lida com problemas de classificação linear, e seu principal propósito é evitar problemas de multicolinearidade e *overfitting* (SINGH; PRAKASH; CHANDRASEKARAN et al., 2016). A Ridge Regression introduz uma penalização nos coeficientes da regressão para evitar que eles assumam valores extremamente altos. Isso é feito adicionando um termo de regularização à função de custo durante o treinamento do modelo. O mesmo conceito é aplicado ao *Ridge Classifie*r (DEEPA et al., 2021).

A principal característica do *Ridge Classifier* é a capacidade de lidar com conjuntos de dados em que as variáveis independentes (características) estão correlacionadas. Quando as características são altamente correlacionadas, a matriz de covariância pode se tornar singular ou próxima disso, causando instabilidade nos coeficientes estimados pelo modelo de regressão linear padrão. O termo de regularização adicionado pelo Ridge ajuda a resolver esse problema. Quanto maior o valor de alpha, maior será a penalização, levando a coeficientes mais próximos de zero. Isso ajuda a evitar overfitting, tornando o modelo mais generalizável para novos dados (STAAL et al., 2004).

4. TRABALHOS RELACIONADOS

4.1. DETECÇÃO DE FRAUDES EM TRANSAÇÕES DE CARTÃO DE CRÉDITO

A detecção de fraudes em transações online de cartão de crédito representa um desafio significativo devido à constante evolução das técnicas empregadas por fraudadores. As estratégias tradicionais de segurança muitas vezes se mostram insuficientes para lidar com a crescente complexidade desses ataques, o que justifica a

necessidade de recorrer a métodos mais avançados, entre eles, os algoritmos de *Machine Learning* (SILVA, 2023).

A aplicação de algoritmos de *Machine Learning* nesse contexto se destaca pela sua capacidade de analisar padrões e comportamentos não lineares nos dados. Ao contrário de abordagens mais convencionais, esses algoritmos conseguem identificar anomalias sutis que podem indicar atividade fraudulenta. A natureza dinâmica e em constante mutação dos métodos utilizados por fraudadores exige uma resposta igualmente adaptável, algo que os algoritmos de *Machine Learning* proporcionam de maneira eficaz (DE SOUZA; JÚNIOR, 2023).

Esses algoritmos têm a capacidade de aprender com dados históricos, reconhecendo padrões complexos e adaptando-se a novas ameaças à medida que surgem. A detecção de fraudes em transações de cartão de crédito é um problema intrinsecamente desafiador, envolvendo a identificação de comportamentos anômalos em meio a uma grande quantidade de dados legítimos. Os algoritmos de *Machine Learning*, ao operarem em grande escala e em tempo real, conseguem oferecer uma abordagem mais proativa para lidar com essas situações (DE SOUZA, 2023).

Além disso, a capacidade de análise não linear desses algoritmos permite a detecção de padrões complexos que podem não ser evidentes por meio de métodos tradicionais. A dinâmica das transações online exige uma abordagem que vá além de simples regras estáticas, e os algoritmos de *Machine Learning* se destacam ao se adaptarem à natureza mutável das fraudes em transações de cartão de crédito (CRISTOVÃO, 2023).

Em resumo, a utilização de algoritmos de *Machine Learning* na detecção de fraudes em transações online de cartão de crédito representa uma resposta eficaz e adaptável a um desafio em constante evolução. A capacidade desses algoritmos de aprender com dados históricos e identificar padrões complexos os torna uma ferramenta valiosa na proteção contra atividades fraudulentas, proporcionando uma camada adicional de segurança em um ambiente cada vez mais sofisticado e dinâmico (ARAUJO, 2022).

4.2. ANÁLISE DE ALGORITMOS DE MACHINE LEARNING

A análise de algoritmos de *Machine Learning* é uma etapa fundamental no desenvolvimento de sistemas eficazes para a detecção de fraudes. A variedade de algoritmos disponíveis, como Logistic Regression, Naive Bayes, Decision Trees e

Gradient Boosting, oferece abordagens distintas para lidar com a complexidade inerente a esse problema. A utilização desses métodos exige uma avaliação cuidadosa para determinar qual se adequa melhor às características específicas dos dados e aos padrões de comportamento associados às transações fraudulentas (VELOSO, 2023).

A biblioteca PyCaret emerge como uma ferramenta valiosa nesse processo de análise, pois proporciona uma interface unificada que simplifica a experimentação com diversos modelos de *Machine Learning*. Essa abordagem simplificada acelera o desenvolvimento, permitindo que cientistas de dados e desenvolvedores avaliem rapidamente a performance de diferentes algoritmos em um conjunto de dados específico (BLUVOL, 2022).

Segundo Martins et al (2019), a escolha do algoritmo adequado é crucial, considerando que diferentes métodos têm vantagens e limitações específicas. A Logistic Regression, por exemplo, é eficaz em problemas de classificação binária, enquanto o Gradient Boosting é conhecido por sua capacidade de lidar com conjuntos de dados complexos e não lineares. A capacidade da biblioteca PyCaret de oferecer uma visão abrangente e comparativa desses algoritmos facilita a tomada de decisões informadas sobre quais modelos prosseguir no processo de desenvolvimento.

Além disso, a interface simplificada do PyCaret também agrega valor ao possibilitar a exploração rápida de diferentes configurações de hiperparâmetros, acelerando a otimização do desempenho dos modelos. A automação de tarefas como pré-processamento de dados, seleção de características e ajuste de hiperparâmetros torna o processo mais eficiente, permitindo que os desenvolvedores foquem na análise interpretativa dos resultados.

Em síntese, a análise de algoritmos de *Machine Learning* é uma etapa crucial na construção de sistemas de detecção de fraudes robustos. A diversidade de algoritmos exige uma avaliação criteriosa, e a biblioteca PyCaret emerge como uma ferramenta valiosa ao proporcionar uma abordagem unificada e eficiente para essa análise, contribuindo para a rápida seleção e implementação de modelos eficazes (SOUSA, 2023).

Analisando os trabalhos apresentados, observa-se que cada autor introduz uma técnica distinta, sugerindo a relevância de avaliações comparativas entre elas. Nesse contexto, esta pesquisa propõe a aplicação de múltiplas técnicas em um mesmo conjunto de dados. O objetivo é realizar uma análise comparativa abrangente, visando entender e destacar as nuances, vantagens e limitações de cada abordagem.

5. RESULTADOS E DISCUSSÃO

Após a aplicação do processo de pré-processamento à base de dados de transações de cartão de crédito, foi gerado o conjunto de dados utilizado neste estudo (disponível em: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud). Ao empregar esse conjunto de dados nas técnicas disponíveis no Pycaret obtivemos os seguintes parâmetros para configuração do ambiente descritos na figura 2.

Figura 2. Setup Pycaret

	Description	Value
0	Session id	1
1	Target	Class
2	Target type	Binary
3	Original data shape	(284807, 31)
4	Transformed data shape	(464123, 31)
5	Transformed train set shape	(378680, 31)
6	Transformed test set shape	(85443, 31)
7	Numeric features	30
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Remove outliers	True
13	Outliers threshold	0.050000
14	Fix imbalance	True
15	Fix imbalance method	SMOTE
16	Fold Generator	StratifiedKFold
17	Fold Number	10
18	CPU Jobs	-1
19	Use GPU	False
20	Log Experiment	False
21	Experiment Name	clf-default-name
22	USI	39a0

Com base na Figura 3, destaca-se o processo de execução dos modelos aplicados pela biblioteca, onde temos também algumas métricas para a análise. A execução teve duração de três horas e meia. Notavelmente, as técnicas *Extra Trees Classifier* e a *Random Forest Classifier* (dada que a *Dummy* é utilizada apenas para comparação a técnicas mais avançadas) se destacaram como as mais promissoras, pois tiveram os

melhores valores para *Accuracy* e AUC. Os modelos baselines, em comparação com os modelos apresentados, estão ilustrados na Figura 3.

Figura 3. Modelos Comparados

```
# compare baseline models
best = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9983	0.9550	0.0233	0.4833	0.0442	0.0440	0.1040	50.2580
dummy	Dummy Classifier	0.9983	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	5.8930
rf	Random Forest Classifier	0.9982	0.9524	0.0087	0.1000	0.0158	0.0156	0.0285	363.5680
xgboost	Extreme Gradient Boosting	0.9981	0.7883	0.0724	0.3041	0.1124	0.1118	0.1427	12.2400
catboost	CatBoost Classifier	0.9975	0.6188	0.0553	0.1135	0.0730	0.0719	0.0773	77.2210
lightgbm	Light Gradient Boosting Machine	0.9974	0.9252	0.1824	0.2038	0.1908	0.1895	0.1907	26.0230
dt	Decision Tree Classifier	0.9965	0.5236	0.0490	0.0389	0.0427	0.0410	0.0416	43.1470
gbc	Gradient Boosting Classifier	0.9727	0.7332	0.2384	0.0153	0.0288	0.0257	0.0544	439.4170
ada	Ada Boost Classifier	0.9487	0.4062	0.1890	0.0065	0.0125	0.0092	0.0263	89.7800
qda	Quadratic Discriminant Analysis	0.8964	0.5965	0.4635	0.0142	0.0275	0.0242	0.0686	6.9140
nb	Naive Bayes	0.8899	0.8649	0.8318	0.0132	0.0260	0.0227	0.0962	6.1940
ridge	Ridge Classifier	0.8825	0.0000	0.8789	0.0128	0.0252	0.0218	0.0976	6.4450
lda	Linear Discriminant Analysis	0.8825	0.9271	0.8761	0.0127	0.0251	0.0218	0.0972	8.0770
lr	Logistic Regression	0.8367	0.9081	0.8592	0.0094	0.0186	0.0152	0.0795	7.5840
knn	K Neighbors Classifier	0.8255	0.5891	0.3233	0.0032	0.0064	0.0029	0.0164	77.7570
svm	SVM - Linear Kernel	0.7391	0.0000	0.4740	0.0063	0.0122	0.0089	0.0315	62.7360

De acordo com a Figura 4, nota-se que a matriz de confusão (Figura 4), é uma ferramenta fundamental na avaliação de modelos de classificação em *machine learning*, fornecendo uma representação detalhada do desempenho do modelo ao comparar suas previsões com os valores reais do conjunto de dados. Ao lidar com problemas de classificação binária, a matriz de confusão organiza os resultados em quatro quadrantes: Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN).

Os Verdadeiros Positivos representam instâncias corretamente classificadas como positivas, enquanto os Falsos Positivos indicam instâncias negativas erroneamente classificadas como positivas. Por sua vez, os Verdadeiros Negativos são instâncias corretamente classificadas como negativas, e os Falsos Negativos representam instâncias positivas erroneamente classificadas como negativas. Esses elementos da matriz de confusão são cruciais para a avaliação quantitativa do desempenho do modelo.

Diversas métricas são derivadas da matriz de confusão para oferecer uma compreensão mais abrangente do desempenho do modelo. A Acurácia fornece uma visão geral da proporção de previsões corretas, enquanto a Precisão destaca a qualidade das

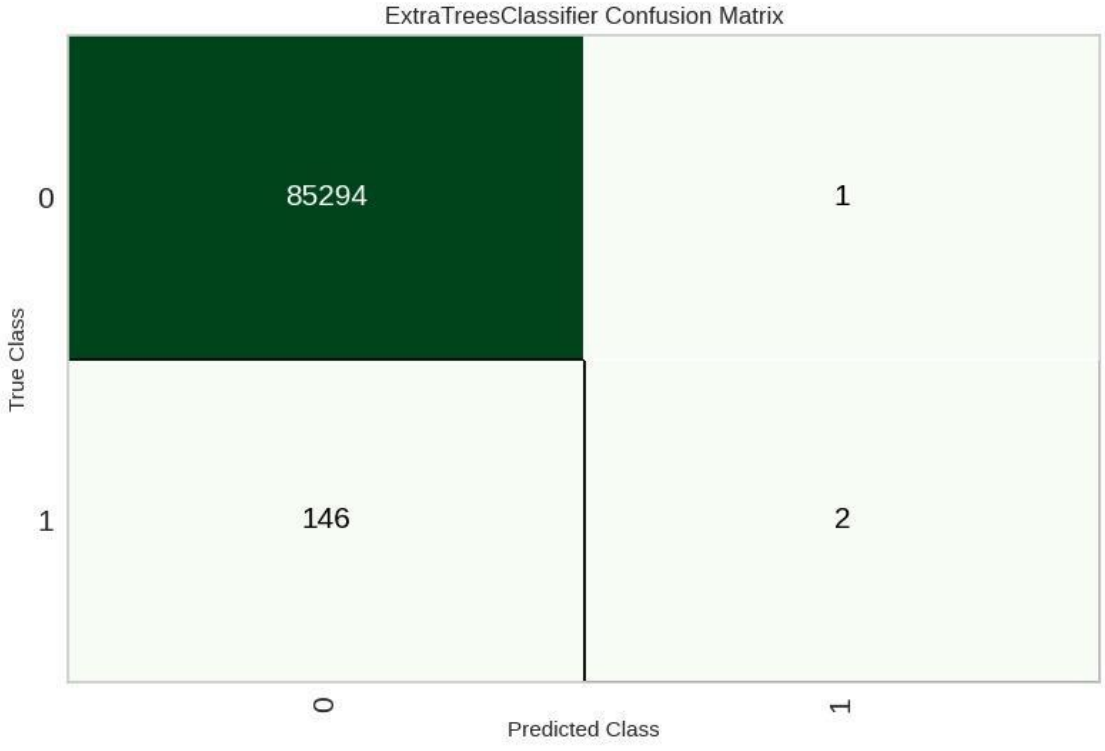
previsões positivas. O Recall, também conhecido como Sensibilidade, destaca a capacidade do modelo de identificar corretamente instâncias positivas.

Além disso, a Especificidade destaca a habilidade do modelo em identificar corretamente instâncias negativas. A métrica F1-Score, uma média harmônica de Precisão e Recall, oferece uma medida combinada de precisão e abrangência.

A interpretação da matriz de confusão e suas métricas associadas é intrinsecamente vinculada ao contexto específico do problema. Em aplicações médicas, por exemplo, minimizar Falsos Negativos pode ser crucial para evitar diagnósticos perdidos. Em contraste, em cenários financeiros, a minimização de Falsos Positivos pode ser mais crucial para evitar alarmes falsos em detecção de fraudes. Essa análise aprofundada da matriz de confusão é essencial para uma avaliação informada e eficaz do desempenho do modelo de *machine learning* em situações do mundo real.

Figura 4. Matriz de confusão

```
# plot confusion matrix
plot_model(best, plot = 'confusion_matrix')
```



As Curvas ROC (Receiver Operating Characteristic) são uma ferramenta gráfica essencial utilizada na avaliação de modelos de classificação em *machine learning*. Essas curvas fornecem uma representação visual da relação entre a taxa de verdadeiros positivos

(TPR) e a taxa de falsos positivos (FPR) em diferentes pontos de corte para a probabilidade de classificação.

A análise das Curvas ROC é particularmente útil em contextos onde o desequilíbrio entre as classes é uma consideração significativa. A curva é construída variando o limiar de decisão do modelo, e, para cada limiar, são calculadas a TPR e a FPR. A TPR é a proporção de instâncias positivas corretamente classificadas em relação ao total de instâncias positivas, enquanto a FPR é a proporção de instâncias negativas erroneamente classificadas como positivas em relação ao total de instâncias negativas.

Um modelo perfeito teria uma Curva ROC que subisse verticalmente até o canto superior esquerdo (TPR=1, FPR=0), indicando alta sensibilidade (capacidade de identificar verdadeiros positivos) e baixa taxa de falsos positivos. Uma curva diagonal, por outro lado, representaria um modelo que não tem habilidade discriminativa.

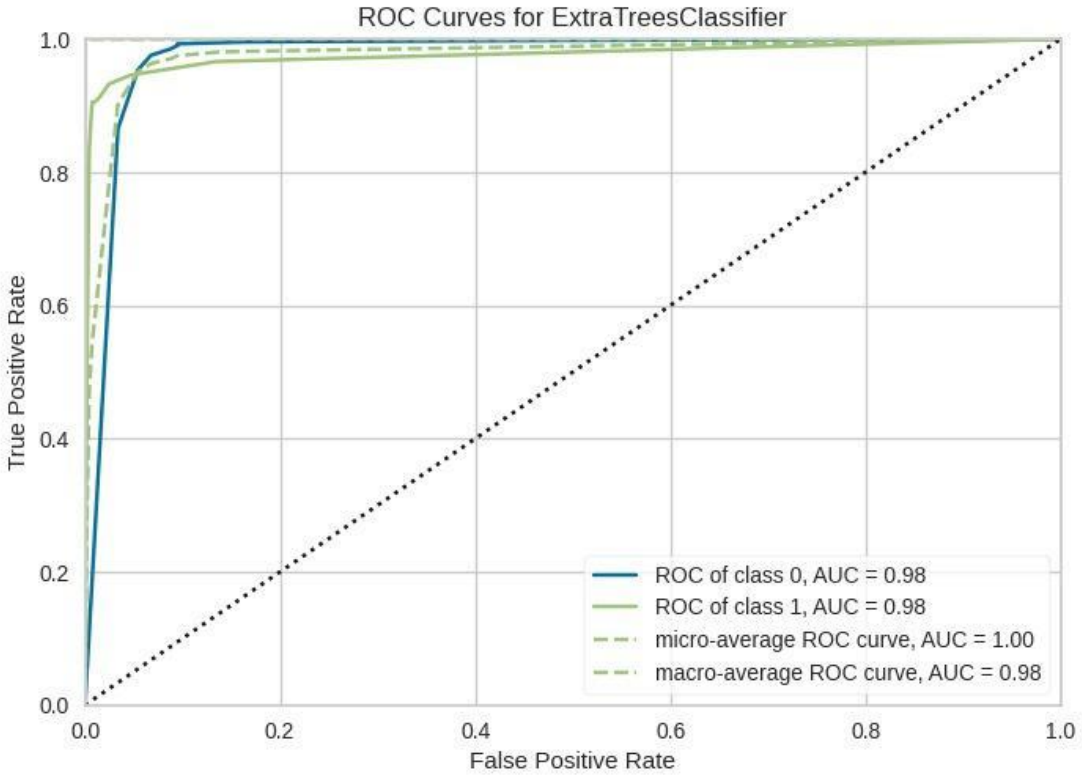
A Área sob a Curva ROC (AUC-ROC) é uma métrica comum derivada dessas curvas e fornece uma medida quantitativa do desempenho global do modelo. Um AUC-ROC de 1.0 indica um modelo perfeito, enquanto um valor de 0.5 sugere que o modelo é equivalente a uma escolha aleatória.

A interpretação das Curvas ROC e do AUC-ROC é crucial para entender a capacidade do modelo de distinguir entre classes, especialmente quando há desequilíbrio. Modelos com curvas ROC mais próximas do canto superior esquerdo e AUC-ROC mais próximos de 1.0 são geralmente considerados mais eficazes na discriminação entre classes.

Essa análise gráfica fornece uma visão holística da performance do modelo, ajudando a selecionar o ponto de operação ideal com base no equilíbrio desejado entre sensibilidade e especificidade, dependendo do contexto da aplicação. Em resumo, as Curvas ROC são uma ferramenta valiosa para avaliação e ajuste fino de modelos de classificação, especialmente em cenários onde a escolha do ponto de corte é uma consideração crítica, onde na Figura 5 é apresentada a curva ROC obtida.

Figura 5. Curvas ROC

```
# plot AUC
plot_model(best, plot = 'auc')
```



Observando a Figura 6, é possível perceber que a análise de Importância de Features representa uma faceta crucial no domínio do *machine learning*, proporcionando insights valiosos sobre o papel relativo de diferentes características no processo de tomada de decisões do modelo. Este exame detalhado visa revelar a contribuição relativa de cada feature, oferecendo uma compreensão mais profunda do funcionamento interno do modelo.

Uma abordagem comum para avaliar a importância das features é a utilização de modelos baseados em árvores, como *Decision Trees*, *Random Forests* e *Gradient Boosting Machines*. Nestes modelos, a importância é muitas vezes medida pela contribuição de cada feature para a redução da impureza nos nós da árvore. Essa análise fornece uma pontuação que indica a relevância relativa das features, permitindo uma classificação hierárquica de sua influência.

Outra técnica amplamente empregada é a Permutação de *Features,* que consiste na aleatorização dos valores de uma feature para observar o impacto resultante no desempenho do modelo. A diferença no desempenho antes e depois da permutação revela a importância da *feature*, uma abordagem particularmente eficaz em modelos como *Random Forests* e *XGBoost*.

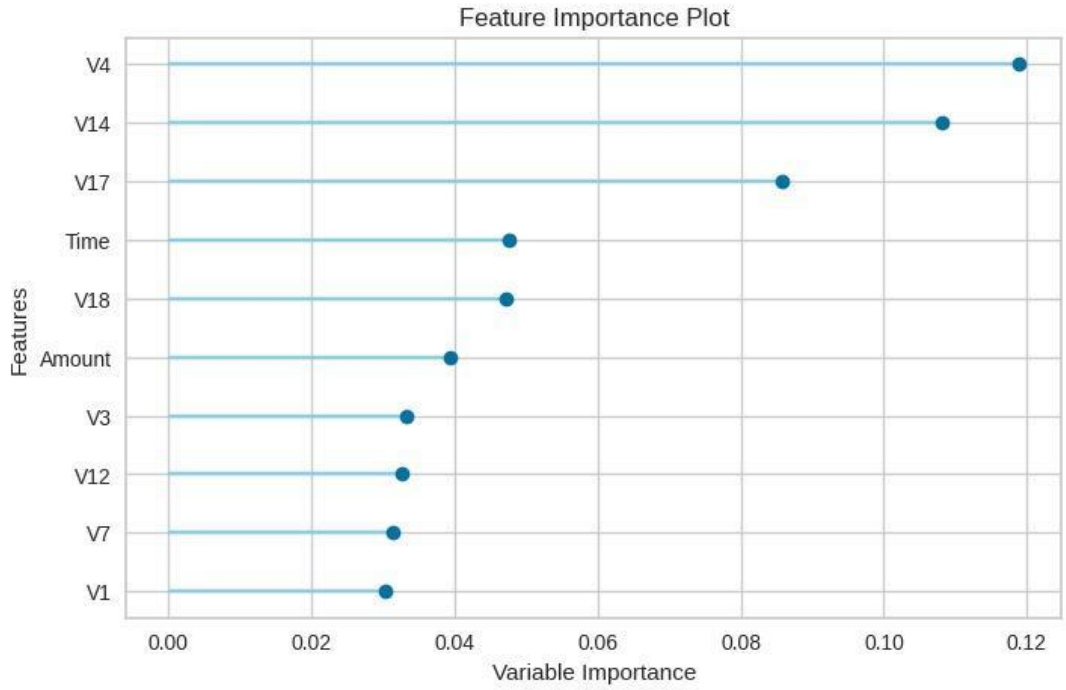
Em modelos lineares, como Regressão Linear, a importância das features é refletida diretamente pelos coeficientes atribuídos a cada *feature*. Coeficientes mais elevados indicam uma influência mais significativa na predição do modelo.

A análise de SHAP (SHapley Additive exPlanations) é uma abordagem teórica que aplica os princípios da teoria dos jogos à interpretação de modelos de *machine learning*. Proporciona explicações globais ou locais sobre a contribuição de cada feature para a saída do modelo, oferecendo uma perspectiva mais abrangente da importância das features.

Compreender a importância das features é essencial não apenas para aprimorar o desempenho do modelo, mas também para selecionar *features* mais relevantes, identificar potenciais problemas de multicolinearidade e comunicar de maneira eficaz o funcionamento do modelo a partes interessadas não técnicas. Além disso, essa análise contribui para a detecção de features que podem ser redundantes ou até mesmo prejudiciais ao desempenho global do modelo, promovendo uma abordagem refinada e eficaz no desenvolvimento de modelos de *machine learning*, corresponde a figura 6.

Figura 6. Feature Importance

```
# plot feature importance
plot_model(best, plot = 'feature')
```



Na Figura 7 temos a Aprendizagem de Máquina Automática, ou AutoML, que representa uma abordagem inovadora que visa automatizar o processo de construção, treinamento e otimização de modelos de *machine learning*. Essa metodologia emergiu como uma resposta à crescente complexidade associada à implementação de algoritmos de *machine learning* e à necessidade de facilitar o acesso a essas poderosas ferramentas para uma gama mais ampla de usuários, incluindo aqueles sem conhecimento especializado em *machine learning*.

O AutoML opera através da aplicação de técnicas automatizadas para várias etapas críticas do ciclo de vida do modelo, incluindo a seleção de features, a escolha do algoritmo, a otimização de hiperparâmetros e até mesmo a interpretação dos resultados. Ao utilizar métodos de busca eficientes, como busca aleatória, busca em grade ou otimização bayesiana, o AutoML pode explorar o espaço de hiperparâmetros de forma mais inteligente, identificando configurações que otimizam o desempenho do modelo.

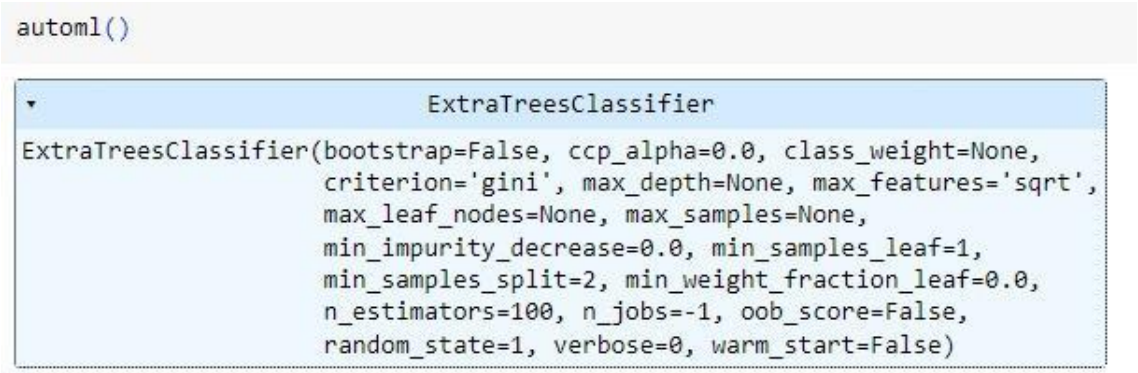
Uma das vantagens notáveis do AutoML é a redução da necessidade de intervenção humana em tarefas que normalmente exigiriam um conhecimento especializado extenso. Isso democratiza o uso de técnicas de *machine learning*, permitindo que cientistas de dados, desenvolvedores e profissionais de domínio

específico aproveitem os benefícios do *machine learning* sem a curva de aprendizado íngreme.

Essas ferramentas automatizadas também podem lidar com problemas práticos, como seleção de features, normalização de dados e tratamento de valores ausentes, proporcionando uma abordagem mais holística ao desenvolvimento de modelos. No entanto, é crucial notar que embora o AutoML simplifique muitos aspectos do processo, a compreensão dos fundamentos da aprendizagem de máquina continua sendo valiosa para interpretar e contextualizar os resultados obtidos.

Em resumo, o AutoML representa uma evolução significativa no campo de *machine learning*, tornando as técnicas avançadas mais acessíveis e eficientes. Essa abordagem automatizada promete acelerar o desenvolvimento de modelos, reduzir a dependência de conhecimento especializado e permitir que uma gama mais ampla de profissionais incorpore técnicas de *machine learning* em seus projetos, como mostra a figura 7.

Figura 7. AutoML



O texto oferece uma explicação clara e abrangente sobre o conceito de matriz de confusão, abordando suas quatro categorias (Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Negativos) e as métricas derivadas, como acurácia, precisão, recall, especificidade e F1-Score. A contextualização dos exemplos médico e financeiro contribui para uma compreensão prática da importância da matriz de confusão na avaliação de modelos de classificação.

A explicação detalhada sobre as Curvas ROC, destacando sua utilidade na avaliação de modelos de classificação, especialmente em cenários de desequilíbrio entre as classes. A introdução da Área sob a Curva ROC (AUC-ROC) como métrica quantitativa é relevante, e a analogia visual da curva diagonal versus a curva desejada

adiciona clareza à interpretação. A contextualização da escolha do ponto de operação ideal destaca a flexibilidade dessa ferramenta na adaptação aos requisitos específicos do problema.

A visão abrangente sobre a análise de importância de features, destacando diversas técnicas, como importância baseada em árvores, permutação de features, coeficientes em modelos lineares e análise de SHAP. A abordagem é informativa, abrangendo desde modelos lineares até técnicas mais avançadas, como SHAP. A explicação da importância prática dessa análise na seleção de features e na interpretabilidade do modelo é valiosa.

A maneira clara e abrangente o conceito de Aprendizagem de Máquina Automática (AutoML), destacando seu papel na automação de várias etapas do ciclo de vida do modelo. A discussão sobre a redução da intervenção humana, a democratização do acesso ao *machine learning* e a abordagem holística para problemas práticos enriquecem a compreensão. A ênfase na importância contínua do entendimento dos fundamentos do *machine learning* equilibra a narrativa ao reconhecer que, apesar da automação, o conhecimento técnico é crucial.

Cada texto demonstra uma abordagem clara e informativa, fornecendo uma compreensão aprofundada dos temas apresentados. A contextualização prática e as analogias visuais contribuem para tornar os conceitos mais acessíveis.

6. CONCLUSÃO

A análise aprofundada de algoritmos de *machine learning*, empregando a biblioteca PyCaret para a detecção de fraudes em transações online de cartão de crédito, revela conclusões fundamentais que podem direcionar estratégias cruciais para fortalecer a segurança nas operações financeiras digitais. Dentro do contexto dinâmico e desafiador das transações online, onde as fraudes se reinventam continuamente, a escolha da abordagem oferecida por uma biblioteca de baixo código, como PyCaret, emergiu como uma estratégia eficiente e de alto impacto.

A eficiência inerente à aplicação da biblioteca PyCaret no processo de análise de algoritmos de *machine learning* é evidente na simplificação e aceleração do ciclo de avaliação. Essa abordagem de baixo código proporciona não apenas economia de tempo, mas também recursos valiosos, permitindo uma comparação ágil e abrangente de diversos algoritmos. Os resultados obtidos não são apenas academicamente significativos, mas também têm implicações práticas substanciais. A superioridade do algoritmo Extra Trees Classifier contribui diretamente para elevar a confiança dos clientes e salvaguardar a integridade dos sistemas financeiros caso seja escolhido como classificador de transações.

A análise conduzida evidencia não apenas a eficácia imediata dos algoritmos selecionados, mas também destaca a necessidade contínua de adaptação. Dada a natureza mutável das estratégias de fraudes, a flexibilidade proporcionada pela abordagem de baixo código, aliada à capacidade de incorporar novos dados e reavaliar os modelos periodicamente, torna-se imperativa para a manutenção da eficácia ao longo do tempo.

Perspectivas futuras podem expandir esses horizontes, explorando a combinação de abordagens de *machine learning* com métodos avançados de processamento de linguagem natural e aprendizado profundo. Essa sinergia pode oferecer uma resposta mais abrangente e adaptável aos desafios em constante evolução apresentados pelas fraudes em transações online de cartão de crédito.

Em síntese, a análise realizada destaca não apenas a eficiência da biblioteca PyCaret e a superioridade do algoritmo Extra Trees Classifier, mas também enfatiza a necessidade de constante adaptação para enfrentar os desafios dinâmicos e complexos inerentes à detecção de fraudes em transações financeiras online.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Elton et al. Performance analysis among predictive models of lightning occurrence using artificial neural networks and SMOTE. **IEEE Latin America Transactions**, v. 19, n. 5, p. 755-762, 2021.

ARAUJO, Danilo. **Estudo comparativo entre algoritmos de aprendizagem de máquina aplicados à detecção de fraudes de cartão de crédito**. 2022. Universidade Federal de Pernambuco, Recife.

ARAÚJO, José Leonardo Gomes de. Modelagem de reforma catalítica seca de metano a gás de síntese, utilizando machine learning e redes neurais. 2022. Universidade Federal de Pernambuco, Recife.

AZAR, Ahmad Taher et al. A random forest classifier for lymph diseases. Computer methods and programs in biomedicine, v. 113, n. 2, p. 465-473, 2014.

BABY, Diana et al. Leukocyte classification based on feature selection using extra trees classifier: Atransfer learning approach. Turkish Journal of Electrical Engineering and Computer Sciences, v. 29, n. 8, p. 2742-2757, 2021.

BHATI, Bhoopesh Singh; RAI, C. S. Ensemble based approach for intrusion detection using extra tree classifier. In: Intelligent Computing in Engineering: Select Proceedings of RICE 2019. Springer Singapore, 2020. p. 213-220.

BLUVOL, Leonardo. **Análise de algoritmos de machine learning para previsão de preços de IBOVESPA**. 2022. Fundação Getúlio Vargas, Rio de Janeiro.

CARMONA, Pedro; CLIMENT, Francisco; MOMPARLER, Alexandre. Predicting failure in the US banking sector: An extreme gradient boosting approach. International Review of Economics & Finance, v. 61, p. 304-323, 2019.

CASTRO, Cristiano Leite de; BRAGA, Antônio Pádua. Aprendizado supervisionado com conjuntos de dados desbalanceados. Sba: Controle & Automação Sociedade Brasileira de Automatica, v. 22, p. 441-466, 2011.

CHAUDHARY, Archana; KOLHE, Savita; KAMAL, Raj. An improved random forest classifier for multi-class classification. Information Processing in Agriculture, v. 3, n. 4, p. 215-222, 2016.

CHEN, Tianqi et al. Xgboost: extreme gradient boosting. R package version 0.4-2, v. 1, n. 4, p. 1-4, 2015.

CRISTOVÃO, Rafael Belmiro. **Detecção de fraudes em cartão de crédito: um caso de uso de modelos supervisionados no e-commerce brasileiro**. 2023. Tese de Doutorado. Universidade de São Paulo.

DE SOUZA, Daniel Henrique Miguel; BORDIN JR, Claudio J. Detecção de fraude de cartão de crédito por meio de algoritmos de aprendizado de máquina. **Revista Brasileira de Computação Aplicada**, v. 15, n. 1, p. 1-11, 2023.

DE SOUZA, Daniel Henrique Miguel; JÚNIOR, Claudio José Bordin. Novo algoritmo ensemble para detecção de fraude em transações de cartão de crédito. **Revista Tecnologia e Sociedade**, v. 19, n. 56, p. 128-145, 2023.

DEEPA, Natarajan et al. An AI-based intelligent system for healthcare analysis using Ridge-Adaline Stochastic Gradient Descent Classifier. The Journal of Supercomputing, v. 77, p. 1998-2017, 2021.

DENG, Lei et al. PDRLGB: precise DNA-binding residue prediction using a light gradient boosting machine. BMC bioinformatics, v. 19, p. 135-145, 2018.

DÉSIR, Chesner et al. Classification of endomicroscopic images of the lung based on random subwindows and extra-trees. IEEE transactions on biomedical engineering, v. 59, n. 9, p. 2677-2683, 2012.

DEVETYAROV, Dmitry; NOURETDINOV, Ilia. Prediction with confidence based on a random forest classifier. In: Artificial Intelligence Applications and Innovations: 6th IFIP WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010. Proceedings 6. Springer Berlin Heidelberg, 2010. p. 37-44.

DOMINGUES, Ricardo; PEDROSA, Isabel; BERNARDINO, Jorge. Indicadores chave de desempenho em marketing. **Indicadores chave de desempenho em marketing**, n. E35, p. 128-140, 2020.

FAN, Junliang et al. Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China. Energy conversion and management, v. 164, p. 102-111, 2018.

FAN, Junliang et al. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. Agricultural water management, v. 225, p. 105758, 2019.

FRANCO, Igor Tedeschi. Manutenção preditiva utilizando técnicas de machine learning em um sistema embarcado. 2020.

GARCIA, Merisandra Côrtes de Mattos et al. Avaliação de métodos de data mining e regressão logística aplicados na análise de traumatismo cranioencefálico grave. 2015.

GUO, Jiaxin et al. Prediction of heating and cooling loads based on light gradient boosting machine algorithms. Building and Environment, v. 236, p. 110252, 2023.

HOSOUME, Juliana Mayumi. Recomendação de técnicas de pré-processamento por meta-aprendizado no contexto de AutoML. 2020.

IQBAL, Faisal Bin et al. Performance analysis of intrusion detection systems using the PyCaret machine learning library on the UNSW-NB15 dataset. 2021. Tese de Doutorado. Brac University.

JUNIOR, Francisco Rodrigues Lima; CARPINETTI, Luiz Cesar Ribeiro. Modelos de decisão para avaliação de desempenho de cadeias de suprimentos baseados no Scor®: uma revisão da literatura. **Brazilian Journal of Development**, v. 5, n. 7, p. 10301-10318, 2019.

KULKARNI, Vrushali Y.; SINHA, Pradeep K. Pruning of random forest classifiers: A survey and future directions. In: 2012 International Conference on Data Science & Engineering (ICDSE). IEEE, 2012. p. 64-68.2023.

LI, Fei et al. A light gradient boosting machine for remainning useful life estimation of aircraft engines. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018. p. 3562-3567.

MARTINS, Joao A.; ROLIM, Carlos O.; LEITHARDT, Valderi RQ. Uma Proposta de Análise de Algoritmos de Machine Learning para o Envio e Recebimento de Notificaçoes em Ambientes Inteligentes. In: **Anais da XIX Escola Regional de Alto Desempenho da Região Sul**. SBC, 2019.

OSMAN, Ahmedbahaaaldin Ibrahem Ahmed et al. Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia. Ain Shams Engineering Journal, v. 12, n. 2, p. 1545-1556, 2021.

PAL, Mahesh. Random forest classifier for remote sensing classification. International journal of remote sensing, v. 26, n. 1, p. 217-222, 2005.

REAL, Eduardo Machado; NICOLETTI, M. do C. Investigação de algoritmos sequenciais de agrupamento com pré-processamento de dados em aprendizado de máquina. Faculdade Campo Limpo Paulista. Campo Limpo Paulista, 2014.

ROCHA, Dárcio Santos et al. Identificação de tipos de relações temporais event-time em português: uma abordagem baseada em regras com classificação associativa. 2023.

RODRIGUEZ-GALIANO, Victor Francisco et al. An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS journal of photogrammetry and remote sensing, v. 67, p. 93-104, 2012.

SHARMA, Deepti; KUMAR, Rajneesh; JAIN, Anurag. Breast cancer prediction based on neural networks and extra tree classifier using feature ensemble learning. Measurement: Sensors, v. 24, p. 100560, 2022.

SILVA, Clodes Fernando de Morais. **Aplicação do PageRank na detecção de fraude em transações com cartão de crédito**. 2023. Universidade Federal de Pernambuco, Recife.

SILVA, Gustavo de Faria. Aplicação de técnicas de pré-processamento e agrupamento na base de dados do aplicativo michelzinho. 2023.

SINGH, Anagh; PRAKASH, B. Shiva; CHANDRASEKARAN, K. A comparison of linear discriminant analysis and ridge classifier on Twitter data. In: 2016 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2016. p. 133-138.

SOUSA, Maria Cristina Cordeiro. Uma análise do algoritmo K-means como introdução ao Aprendizado de Máquinas. 2023.

SOUSA, Thiago R. et al. LGPD: Levantamento de Técnicas Criptográficas e de Anonimização para Proteção de Bases de Dados. In: **Anais do XX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**. SBC, 2020. p. 55-68.

STAAL, Joes et al. Ridge-based vessel segmentation in color images of the retina. IEEE transactions on medical imaging, v. 23, n. 4, p. 501-509, 2004.

TAHA, Altyeb Altaher; MALEBARY, Sharaf Jameel. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access, v. 8, p. 25579-25587, 2020.

VELOSO, Bruno Cruz. **Análise de algoritmos de machine learning para deteção de violência em áudio**. 2023. Universidade do Minho, Braga.

VIACAVA, Francisco et al. Avaliação de desempenho de sistemas de saúde: um modelo de análise. Ciência & Saúde Coletiva, v. 17, p. 921-934, 2012.

WHIG, Pawan et al. A novel method for diabetes classification and prediction with Pycaret. Microsystem Technologies, p. 1-9, 2023.

Moreno, J & Rodriguez, Daniel & Sicilia, M. & Riquelme, José & Ruiz, Y. (2009). SMOTE-I: mejora del algoritmo SMOTE para balanceo de clases minoritarias.

Sharaff, Aakanksha & Gupta, Harshil. (2019). Extra-Tree Classifier with Metaheuristics Approach for Email Classification. 10.1007/978-981-13-6861-5_17.

Chang, Y.-C., Chang, K.-H., & Wu, G.-J. (2018). Application of eXtreme Gradient Boosting Trees in the Construction of Credit Risk Assessment Models for Financial Institutions. Applied Soft Computing, 73, 914-920.