O impacto da seleção de atributos e da separação dos dados de detecção de fraudes na tarefa de classificação

Giovanna R. Mendes[1], **Matheus Felipe A. Durães**[1], **Jonathan de A. Silva**[1]

[1] Curso de Ciência da Computação - Faculdade de Computação
Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 79.070-900 – Campo Grande – MS – Brasil

{giovanna.mendes, m.felipe, jonathan.andrade}@ufms.br

Abstract. *In the context of credit card transaction fraud, it is necessary to enhance anti-fraud systems by considering the data splitting strategy and feature selection. This study evaluates how feature selection and data separation affect the performance of machine learning models. Three feature selection strategies and five training approaches were applied using three tree-based models. Techniques such as K-Means, undersampling, and nested cross-validation were combined to create different experimental scenarios. Feature selection II achieved good and consistent results across all experiments.*

Resumo. *No contexto de fraudes em transações de cartão de crédito, é preciso aprimorar os sistemas anti-fraudes, levando em consideração o modo de divisão dos dados e a seleção dos atributos. Assim, este trabalho avalia como a seleção de atributos e a separação dos dados impactam o desempenho de modelos de aprendizado de máquina. Três estratégias de seleção de atributos e cinco abordagens de treinamento foram aplicadas, usando três modelos baseados em árvores. Técnicas como K-Means, undersampling e validação cruzada aninhada foram combinadas para formar diferentes cenários experimentais. A seleção de atributos II obteve resultados bons e consistentes em todos os experimentos.*

1. Introdução

Com o advento da tecnologia, muitas áreas têm se desenvolvido com o intuito de facilitar atividades humanas, sendo as máquinas uma das ferramentas mais valorizadas atualmente. Outro aspecto incluído com a modernização são as transações financeiras de modo *online* que trazem mais conforto e facilidade aos donos de contas bancárias. Contudo, a forma de transferência *online* trouxe um novo modelo mais sofisticado para golpistas utilizarem para criar fraudes [Sreekala et al. 2024]. Os casos de golpes não se limitam a determinados países, é um problema global [Sreekala et al. 2024].

As fraudes, no contexto do estudo, são transações bancárias que não são feitas nem autorizadas pelos proprietário das contas, mas sim por outra pessoa a fim de ganhar alguma vantagem financeira. Com isso, fraudadores têm feito transferências cada vez mais verossímeis com transações legítimas, usando valores em produtos com maior liquidez, maior valor e maior custo, como noticiado pela [CNN 2024]. Isso acarreta em prejuízos financeiros aos clientes e socioeconômicos às empresas, uma vez que perdem dinheiro e confiança dos consumidores de seus produtos [Sreekala et al. 2024]. Assim, a

detecção das operações pelos sistemas antifraudes é um desafio para instituições financeiras [Ahmad et al. 2023].

Segundo a [Unico 2023], empresa que gerencia identidades digitais, é importante haver medidas de proteção contra fraudes por parte das empresas. A falta da segurança provoca a reputação negativa delas, como indicado na pesquisa Relatório de Fraude 2024 feita pelo Serasa Experian em que "[...] *62% das pessoas dizem estar dispostas a pagar mais caro em uma marca que lhes ofereça menor possibilidade de serem vítimas de fraude online. 86% dos usuários têm expectativas de que as instituições adotem as medidas necessárias para protegê-los dos golpes*."[Experian 2024]. Estratégias de proteção contra fraudes a se citar a autenticação de identidade por biometria facial e a segurança no Wi-Fi devem ser priorizadas, conforme recomendado pela IDTech Unico [Unico 2023]. Tais abordagens e outras providências iniciadas pelos clientes auxiliam na redução dos casos de golpes, deixando as demais a cargo das empresas incumbidas de detectá-los. Portanto, é necessário desenvolver métodos para identificar as fraudes de transações de cartão de crédito para evitar perdas aos usuários [Ileberi et al. 2022].

Por conseguinte, para combater os golpes globais é preciso identificar os fatores que dificultam o desempenho dos sistemas de detecção de fraudes, aprimorar a análise dos dados e otimizar os modelos, não somente sob o viés de técnicas de aprendizado de máquina corretas [Ahmad et al. 2023]. Deve-se, em adição, considerar o modo de divisão dos dados e a escolha de atributos para uma melhor aprendizagem e, por consequência, aperfeiçoar a identificação de fraudes [Ahmad et al. 2023, Ileberi et al. 2022]. Dessa forma, busca-se proporcionar maior segurança aos usuários no que se refere a suas informações.

1.1. Objetivo

Como ocorre neste trabalho e nas pesquisas relacionadas, estas são apontadas na Seção 3, a adoção de abordagens diferentes e complementares na tarefa de detecção acarreta distintos aprendizados pelos modelos. Diante disso, o corrente trabalho tem como objetivo avaliar o impacto da separação dos dados e da seleção de atributos de forma sistemática no desempenho de modelos de aprendizado de máquina aplicados à detecção de fraudes em transações financeiras. Além dessa abordagem, busca-se evidenciar discrepâncias de resultados em comparação com implementações análogas às de [Ahmad et al. 2023] e [Sreekala et al. 2024].

Ademais, busca-se responder as seguintes perguntas:

a) O quanto a separação dos dados impacta no modelo final?
b) Como a seleção de atributos pode impactar no algoritmo?
c) Os modelos são consistentes nos cenários propostos?
d) Qual o desempenho do Validação Cruzada Aninhada [Varma and Simon 2006, Berrar et al. 2019] em relação aos outros experimentos?

Para essa análise, diferentes abordagens de pré-processamento e validação são testadas, incluindo:

a) Três conjuntos distintos de atributos;
b) Cinco estratégias de separação de dados: Validação Cruzada Aninhada, *K-Means* [MacQueen 1967], *K-Means* com RUS [Mishra 2017] manual e 70/30;

c) Três algoritmos de classificação amplamente utilizados: Árvore de Decisão [Quinlan 1986], Floresta Aleatória [Breiman 2001] e XGBoost [Chen and Guestrin 2016];
d) Três formas de visualização: PCA [Pearson 1901], tabelas e gráficos.

Este trabalho também visa contribuir com pesquisas na área, demonstrar, por meio de experimentações empíricas, como diferentes escolhas metodológicas - principalmente a seleção de atributos e o modo de separação dos dados - afetam significativamente o desempenho dos modelos em tarefas com forte desbalanceamento de classes, como é o caso da detecção de fraudes de cartão de crédito.

Além disso, o estudo apresenta um comparativo detalhado de métricas de desempenho em múltiplos cenários, reforça a importância de estratégias de balanceamento de dados (como o *undersampling*) e fornece um protocolo replicável para testes com distintos conjuntos de atributos e abordagens de separação, podendo ser úteis para pesquisadores e profissionais da área.

2. Fundamentação teórica

No processo de treinamento e de teste de algoritmos de aprendizado de máquina supervisionado, diversas etapas são realizadas até obter resultados satisfatórios. Durante o pré-processamento dos dados, técnicas de normalização, de autocodificação e de balanceamentos dos dados podem ser aplicadas. A respeito do balanceamento dos dados, a Seção 2.3 aborda métodos para equilibrar as proporções de dados entre as classes majoritárias e minoritárias. Para lidar com isso, existem diferentes estratégias, dentre elas as abordagens de subamostragem (*undersampling*) e de sobreamostragem (*oversampling*). Tal abordagem é necessária, pois algoritmos de classificação tendem a se tornar enviesados em favor da classe majoritária, o que pode resultar em acurácia alta, porém baixo desempenho em identificar a classe minoritária.

Em um primeiro momento, modelos de aprendizado de máquina não-supervisionados, que são responsáveis por identificar padrões em dados não rotulados, podem ser operadas no pré-processamento dos dados, discutidos na Seção 2.1, a se referir o *K-Means* ou o PCA. Para a etapa de treinamento, algoritmos de aprendizado de máquina supervisionados, responsáveis por realizar previsões a partir de dados rotulados, exposto na Seção 2.2, são selecionados. Conforme exibido na Seção 2.4, técnicas para avaliar a execução do algoritmo de aprendizado de máquina podem ser utilizadas com o propósito de testar o modelo de uma forma justa e coesa para atingir resultados similares ao especificar uma semente para a aleatoriedade. Para avaliar o desempenho dos modelos, leva-se em conta os resultados originários de métricas expostas na Seção 2.5.

2.1. Modelo de aprendizado de máquina não-supervisionado

2.1.1. K-*Means*

O *K-Means* é um algoritmo iterativo que organiza os dados em grupos com base em sua similaridade com outros. Cada *cluster* é representado por um ponto central denominado centroide. O número de agrupamentos, representado por k, é previamente definido pelo usuário. O algoritmo busca minimizar a distância entre os dados e seus respectivos centroides e finaliza sua execução quando as atualizações dos centroides se tornem insignificantes ou quando uma condição de parada é atingida.

2.1.2. PCA

O *Principal Component Analysis* (PCA) é uma técnica de redução de dimensionalidade que transforma variáveis possivelmente correlacionadas em um subconjunto menor de variáveis, chamadas de principais componentes. O objetivo é reter a maior variância possível dos dados para facilitar a visualização e o processamento, além de minimizar o surgimento de *overfitting*. O PCA normalmente é utilizado como etapa de pré-processamento antes da aplicação de algoritmos de aprendizado de máquina.

2.2. Modelos de aprendizado de máquina supervisionados

2.2.1. Árvore de Decisão

A Árvore de Decisão (DT ou *Decision Tree* em inglês) é um algoritmo de aprendizado de máquina supervisionado que utiliza uma estrutura hierárquica baseada em decisões. O algoritmo cria um diagrama em formato de árvore, por meio de perguntas binárias, onde cada nó representa uma condição sobre um atributo. A ramificação finaliza nos nós folha que indicam as classes atribuídas.

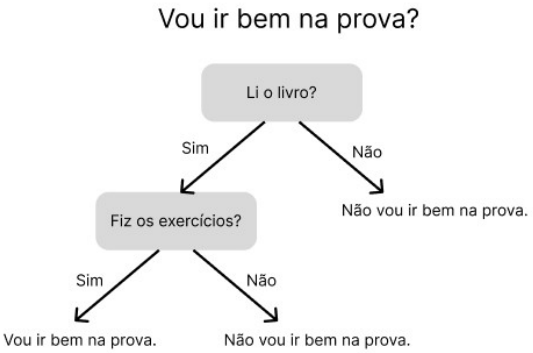


Figura 1. Árvore de decisão

Para gerar as ramificações, a árvore cria pontos de decisão, em que funciona como perguntas hierárquicas para dividir os dados em subconjuntos do modo mais eficiente possível, como apresentada pela Figura 1. Para isso, são usados os seguintes cálculos:

a) Gini [IBM 2022]: a medição serve para medir a impureza de um nó, que retrata o grau de mistura das classes. Caso o valor resultante da equação seja 0, o nó é considerado puro, já que possui valores de uma única classe. Entretanto, se resultar em 1, significa que o nó é impuro formado por mescla de classes. Este é o critério mais comum para a construção da árvore.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \tag{1}$$

Em que $p_i = \frac{|S_i|}{|S_{total}|}$, sendo S_i o número de amostras da classe *i* e o S_{total}, o número total de amostras no nó.

b) Entropia [IBM 2022]: a medição analisa a desordem dos dados. Quanto mais próximo o valor de 0, maior é a pureza. Porém, quanto mais próximo de 1, maior é a desorganização do *dataset* e mais equilibrado é em relação à quantidade de dados de cada classe.

$$Entropia = -\sum_{i=1}^{n}(p_i * log_2 p_i) \tag{2}$$

c) Ganho de informação [IBM 2022]: a medição quantifica a redução da entropia após a divisão, representa o quanto se ganha ao dividir um conjunto de dados conforme um atributo em relação à pureza do dado. Quanto maior o ganho, melhor é o atributo para a divisão dos dados.

$$Ganho\ de\ informação = Entropia_{no\,pai} - \Sigma peso_{no\,filho} * Entropia_{no\,filho} \tag{3}$$

Em que $peso_{no\,filho} = \frac{|S_u|}{|S_{total}|}$, sendo S_u o número de amostras da classe *u* e o S_{total}, o número total de amostras no nó.

Vale salientar que, as Árvores de Decisão não são utilizadas somente para a tarefa de classificação - objeto do presente estudo -, mas também para a de regressão. Em suma, são capazes de prever tanto variáveis categóricas (como ”sim”ou ”não”) quanto numéricas. No entanto, a técnica pode resultar em *overfitting*, uma vez que pode ajustar-se excessivamente aos dados de treinamento com a inclusão de dados ruidosos e padrões específicos [IBM 2021].

2.2.2. Floresta Aleatória

A Floresta Aleatória (RF ou *Random Forest*), é um algoritmo de aprendizado de máquina que combina múltiplas saídas das Árvores de Decisão. Une a simplicidade das árvores com uma carga de aleatoriedade na seleção de atributos (*feature bagging*) e dados para melhorar a precisão, o que resulta em um modelo mais robusto e com maior capacidade de generalização. Sendo assim, um método que traz uma solução mitigadora para o problema de *overfitting* das Árvores de Decisão, mencionado na Seção 2.2.1 [Ileberi et al. 2022].

A execução supervisionada do algoritmo é construída por diversas etapas. Na primeira fase, uma amostragem aleatória com reposição, chamado *bootstrap*, é criada para formar cada árvore. Da amostra de treinamento, um terço dos dados é escolhido para formar o conjunto de teste, os demais dados são destinamos para o treino (o *out-of-bag* ou oob). Em seguida, características são escolhidas de modo aleatório para cada árvore, seguindo o mesmo princípio da Árvore de Decisão. Durante a execução, identifica-se qual atributo proporciona a melhor divisão dos dados com a posterior adesão dos demais atributos nas árvores.

Para suceder a classificação, cada árvore percorre seus nós percorridos até os nós folha para atribuir a cada dado uma classe. A decisão final é feita por intermédio de votação com a consideração da classe majoritária resultante da análise das árvores, quando

o problema for de classificação. Em vista disso, a Floresta Aleatória é considerada um método de *ensemble* por combinar múltiplos resultados de modelos para produzir uma previsão agregada. É importante destacar que, é indispensável definir inicialmente hiperparâmetros, como: tamanho dos nós, número de árvores e número de características amostradas [IBM 2024].

O modelo possui várias vantagens, como robustez, poucas iterações de ajustes e não é tão propenso a sofrer *overfitting*. Todavia, exige um poder grande de processamento, ao ser comparado às Árvores de Decisão, visto que a decisão é pautada em uma quantidade significante de árvores e de nós para serem criados e percorridos.

2.2.3. XGBoost

O *eXtreme Gradient Boosting* ou XGBoost é um algoritmo de aprendizado de máquina baseado na Árvore de Decisão que baseia-se no *Gradient Boosting* (GB) [Friedman 2001]. Cria uma sequência de árvores, adicionando ao conjunto modelos fracos (*weak learners*), em que cada nova árvore é treinada para corrigir os erros cometidos pelas anteriores. Desta forma, o modelo aprimora de modo progressivo sua performance.

2.3. Técnicas de balanceamento dos dados

2.3.1. RUS

O *Random Undersampling* (RUS) é um método de subamostragem do qual é responsável por reduzir a quantidade de dados da classe majoritária de forma aleatória. Dessa forma, auxilia na redução do tempo de processamento, mas perde-se informações da classe majoritária.

2.3.2. SMOTE

O SMOTE [Chawla et al. 2002] é uma técnica de sobreamostragem em que cria-se mais dados à amostra da classe minoritária. A escolha dos dados para serem a base da formação dos novos é feito por meio do cálculo da distância euclidiana dos dados da classe minoritária com seus *k* vizinhos mais próximos (KNN ou *K-Nearest Neighbors* em inglês [Cover and Hart 1967]). Assim, cria-se os *k* dados sintéticos. O *Adaptive Synthetic Minority Sampling Technique* (ADASYN) [He et al. 2008] é uma variante do SMOTE [Chawla et al. 2002] que ajusta a criação de dados sintéticos com base na dificuldade de aprendizado.

2.4. Técnicas para avaliação de modelos de aprendizado de máquina

2.4.1. K-Fold Cross-Validation

A Validação Cruzada (em inglês, *Cross-Validation*) [Stone 1978, Berrar et al. 2019] é uma técnica estatística para avaliar e comparar modelos de aprendizado de máquina. O objetivo é particionar os dados de um conjunto em *k* subconjuntos ou dobras (*folds*). Em cada iteração, uma porção dos dados é selecionada para o teste e os 'k-1' subconjuntos restantes são destinados para o treinamento. O processo é repetido *k* vezes de modo que cada subconjunto faça parte do conjunto de teste.

2.4.2. Validação Cruzada Aninhada

A Validação Cruzada Aninhada (Nested Cross-Validation em inglês) é uma abordagem utilizada para avaliar os modelos de aprendizado de máquina. É usada para treinar um modelo de modo a otimizar seus hiperparâmetros. O método incorpora dois níveis de validação cruzada, um interno e outro, externo. O laço externo tem como objetivo avaliar o desempenho do modelo de modo geral, ao passo que o interno, selecionar e otimizar os hiperparâmetros.

O processo de execução da abordagem ocorre da seguinte forma:

a) No *loop* externo, os dados são particionados em treino e teste. O modelo é avaliado no conjunto de teste externo;
b) A parte de treino do *loop* externo é então submetida ao *loop* interno, que realiza uma nova divisão em treino e validação. Nesse ciclo interno, são executadas buscas exaustivas ou algoritmos de otimização sobre os hiperparâmetros;
c) Ao término do *loop* interno, seleciona-se o conjunto de hiperparâmetros que apresenta melhor desempenho, treinando-se o modelo com esses hiperparâmetros otimizados na totalidade dos dados do treino externo;
d) Esse modelo otimizado é então avaliado no conjunto de teste externo.

2.5. Métricas de classificação

A fim de os experimentos poderem ser comparados com de outros estudos, o presente trabalho utiliza cinco métricas de classificação, sendo as mais relevantes para o contexto de fraudes a 1 e a 1. Isso porque ambas indicam se o modelo está acertando o que são as transações legítimas e as fraudulentas e não confundi entre as classes, podendo, em um pior caso, considerar uma fraude como transação verídica.

Assim, serão apresentadas cada uma das métricas após a explicação de como funciona a matriz de confusão.

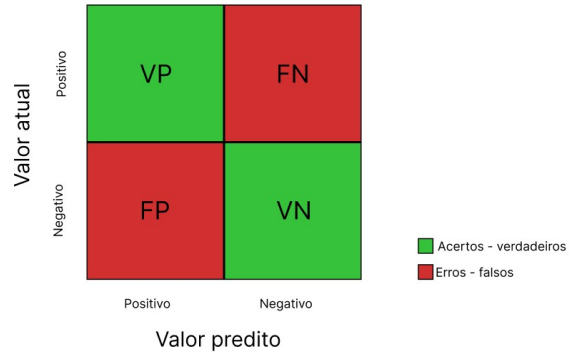


Figura 2. Matriz de confusão

Os resultados obtidos de um algoritmo são expostos, de modo visual, por meio de uma matriz de confusão, como indicada na Figura 2. Cálculos são executados para medir o desempenho de um algoritmo de aprendizado de máquina de um modelo de classificação, revelado nas subseções a diante.

Sobre as nomenclaturas e sua definições:

a) Verdadeiros positivos (VP): corresponde às previsões corretas da classe positiva;
b) Verdadeiros negativos (VN): corresponde às previsões corretas da classe negativa;
c) Falsos positivos (FP): corresponde às previsões incorretas da classe negativa, classificados como positiva;
d) Falsos negativos (FN): corresponde às previsões incorretas da classe positiva, classificados como negativa.

No contexto das classes do *dataset* de detecção de fraudes de cartão de crédito do [Kaggle 2018], as nomenclaturas possuem as seguintes especificações:

a) Verdadeiros positivos (VP): corresponde às previsões corretas da classe correspondente a uma transação legítima;
b) Verdadeiros negativos (VN): corresponde às previsões corretas da classe correspondente a uma transação fraudulenta;
c) Falsos positivos (FP): corresponde às previsões incorretas da classe fraudulenta, classificados como legítima;
d) Falsos negativos (FN): corresponde às previsões incorretas da classe legítima, classificados como fraudulenta.

Tabela 1. Métricas

Tipo	Descrição	Cálculo	Finalidade
Acurácia	Quão correto está o treinamento.	$Acurácia = \frac{VP+VN}{VP+FP+VN+FN}$	Para ser usado de comparação com outros artigos.
Precisão	Taxa de dados realmente verdadeiros entre todos os positivos identificados.	$Precisão = \frac{VP}{VP+FP}$	Analisar quantas transações legítimas o modelo está encontrando.
Revocação	Taxa de dados positivos reais foram classificados corretamente.	$Revocação = \frac{VP}{VP+FN}$	Analisar quantas transações fraudulentas o modelo está encontrando.
Especificidade	Taxa de verdadeiros negativos em relação ao total de negativos reais existentes.	$Especificidade = \frac{VN}{VN+FP}$	Explicar métrica usada por artigos com outros artigos.
F1-Score	Média harmônica entre Revocação e Precisão.	$F1\text{-}Score = \frac{2*Precisão*Revocação}{Precisão+Revocação}$	Para ser usado de comparação com outros artigos.
AUC	Probabilidade do modelo classificar uma instância positiva maior do que uma negativa.	$TPR = \frac{VP}{VP+FN}$ e $FPR = \frac{FP}{FP+VN}$	Analisar se o modelo consegue distinguir entre as classes.

Diante das aplicações das métricas de classificação, as mais utilizadas na literatura são: acurácia, precisão, revocação e F1-*Score*. Contudo, no cenário de detecção de fraudes de cartão de crédito, as mais relevantes são: precisão, revocação e AUC, sendo as demais, no presente estudo, usadas para comparação com outros experimentos de outras pesquisas na área.

3. Revisão da literatura

Diante da relevância da temática e dos impactos causados em fraudes de transações financeiras, diversas pesquisas foram realizadas nessa área com intuito de aprimorar métodos de detecção. Para esse propósito, uma ampla variedade de tecnologias foram desenvolvidas. Devido à diversidade de estratégias propostas na literatura, a seção de revisão literária foi estruturada em três vertentes - separação de dados, seleção de atributos e tratamento do desbalanceamento de dados -, selecionando artigos com base em temáticas relevantes para os experimentos que servem como base para esses, presentes na Seção 4.

A seleção dos artigos foi realizada com base na análise das temáticas abordadas, verificando-se a correspondência com os seguintes campos de estudo: *Credit Card Fraud Detection*, impacto da separação em treino e teste, seleção ou exclusão de atributos e técnicas de amostragem dos dados. Foram considerados trabalhos publicados a partir de 2021 com o idioma inglês e disponíveis em uma das seguintes bases: *IEEE*, *Taylor & Francis* e *Springer*. Além desses, também foram selecionados estudos das bases de dados da *Nature* e do *Seventh Sense Research Group* por atenderam os critérios estabelecidos.

3.1. Separação de dados

[Singh et al. 2021] apontaram o impacto da escolha de métodos para a validação do treinamento e do teste. Para os experimentos, dois conjuntos de dados distintos foram usados, um sobre pacientes que foram sujeitos a imagens de perfusão miocárdica de estresse nuclear proporcionado pelo *Cedars-Sinai Medical Center* (CSMC) do *Artificial Intelligence in Medicine Program* [Arsanjani et al. 2015] e o outro, sobre pacientes recrutados sem doença arterial coronariana, infarto do miocárdio ou revascularização coronariana prévia anteriormente conhecidos oferecido pelo centro na coorte prognóstica do registro REFINE SPECT [Slomka et al. 2020], ambos aprovados para o uso pelo Cedars Sinai e no primeiro conjunto ser aplicado a seleção de atributos. Regressão Logística (LR) [Hosmer Jr et al. 2013], *Gaussian Naïve Bayes* (GNB) [Rish et al. 2001], Floresta Aleatória e *Linear Discriminant Analysis* (LDA) [FISHER 1936] são os algoritmos de *Machine Learning* (ML) usados em conjunto com a validação estratificada 70/30 e 50/50 (treinamento/teste), a validação cruzada estratificada com 10-*fold*, a validação cruzada estratificada com 10-*fold* repetida 10 vezes, o *bootstrapping* (repete 500 vezes) e a validação deixando um para fora (ou, em inglês, *leave one out*, LOO) com a utilização da biblioteca Sklearn [Pedregosa et al. 2011]. Para cada algoritmo, exceto o LOO, 100 sementes diferentes foram utilizadas para implementar divisões dos dados distintas. Com divisão estratificada 70/30, os modelos conquistaram resultados mais significativos, sendo a Regressão Logística e a Floresta Aleatória os destaques. Porém, aponta a validação cruzada estratificada com 10-*fold*, a validação cruzada estratificada com 10-*fold* repetida 10 vezes e o *bootstrapping* como fornecedores de maior precisão e de estabilidade devido às estimativas de desempenho mesmo que mais custosos.

[Ahmad et al. 2023] expuseram um modo diferente de detectar as fraudes nas transações de cartão de crédito do [Kaggle 2018] baseado no *similarity-based selection* (SBS) com Fuzzy C-means [Dunn 1973]. Os dados são separados em cinco grupos contendo, pelo menos, 10% dos dados fraudulentos enquanto é aplicado o Fuzzy C-means para efetuar uma subamostragem dos dados de forma não aleatória. Para fazer o balanceamento, instâncias legítimas são escolhidas com base na sua maior semelhança com as fraudulentas dentro de cada um dos *clusters*. Vale ressaltar que, três conjuntos são criados por meio das proporções de dados fraudulentos e legítimos, a se citar 1:1, 1:2 e 1:3, para compor a amostra. Desse modo, agrupa os dados que posteriormente serão selecionados para formar uma amostra em que 70% é destinada para o treino e o restante, para o teste. *Artificial Neural Networks* (ANN) [McCulloch and Pitts 1943], Regressão Logística, KNN e Naïve Bayes (NB) [Rish et al. 2001] são os algoritmos de aprendizado de máquina selecionados. Por fim, os resultados obtidos com o SBS são comparados com os alcançados pelo RUS e CBUFN [Rekha and Tyagi 2021] diante da métricas usadas. Um dos melhores resultados obtidos foi utilizando SBS e o ANN com o conjunto de

dados constituído por 25% de fraudulentos e 75%, verídicos.

[Sreekala et al. 2024] executaram uma abordagem de separação de dados do conjunto de dados de detecção de fraudes de cartão de crédito [Kaggle 2018] com o uso do K-*Means*. O algoritmo é utilizado também na etapa posterior de pseudo-rotulagem. Na fase seguinte, diversas MLs foram usadas para comparar os resultados da técnica aplicada, como o KNN, a Árvore de Decisão, a Floresta Aleatória, o Naïve Bayes e o *Support Vector Classification* (SVC) [Boser et al. 1992]. Deve-se enfatizar que, o trabalho aplicou o método do cotovelo (ou *Elbow Method*) [Bholowalia and Kumar 2014] para seletar o melhor valor para *k*, o número de agrupamentos, que resultou em valor ótimo de dois ou três. O procedimento consiste de iterar de um até um valor *k* de *clusters* que se deseja (no caso pontual, foi atribuído o valor 10), calcular o WCSS (indica o quão bem os pontos estão agrupados ao redor de seus respectivos centroides guardando os resultados das somas dos quadrados das distâncias dos pontos até o centroide mais próximo) e representar por meio de um gráfico. Se um valor está antes do "cotovelo", há um aumento no número de grupos que captura de modo mais eficaz a variabilidade dos dados. Após o "cotovelo", existe a possibilidade de ocorrer *overfitting* uma vez que há *clusters* adicionais que não são necessários, o que pode causar equívocos no aprendizado. O melhor resultado obtido por meio da análise das métricas usadas foi com o uso da Floresta Aleatória.

[Rinku et al. 2024] apresentaram testes com a base de dados de detecção de fraudes de cartão de crédito [Kaggle 2018]. Para a seleção de algoritmos de ML, Regressão Logística, KNN, *Support Vector Machines* (SVM) [Boser et al. 1992], Árvore de Decisão, Floresta Aleatória, AdaBoost [Freund et al. 1996], *Gradient Boosting*, *Multi-Layer Perceptron* (MLP) [Kruse et al. 2022] e *Gaussian Naïve Bayes* (GNB) [Rish et al. 2001] foram escolhidos. Para seleção de atributos, o LDA foi selecionado. Como a base de dados [Kaggle 2018] é desbalanceada, o SMOTE e o ADASYN foram as técnicas de pré-processamento dos dados escolhidas. Para o treinamento, além da etapa de pré-processamento dos dados e da escolha dos algoritmos, três porções dos dados foram determinadas para compor os experimentos a se citar 70:30, 75:25 e 80:20. Após o treinamento do modelo e da etapa de *tuning* dos hiperparâmetros, técnicas de *ensemble* de aprendizado são realizadas para combinar previsões de múltiplos modelos. O destaque levantado foi para o uso do Floresta Aleatória com ADASYN com a separação 80:20.

3.2. Seleção de atributos

[Ileberi et al. 2022] indicaram uma análise com o uso de algoritmo genético (GA) usando Floresta Aleatória inspirado em [Davis 1991] para fazer a seleção de atributos para, logo após, aplicar os seguintes classificadores de *Machine Learning* para comparação: Árvore de Decisão, Floresta Aleatória, Regressão Logística, Naïve Bayes e ANN na base de dados de detecção de fraudes de cartão de crédito do [Kaggle 2018]. A pesquisa foi voltada para desenvolver um algoritmo de seleção de atributos por intermédio do GA. Para a separação dos dados, foi separado 20% da base de dados original que, em sequência, foi divido em treino e teste. Após a separação, uma ML foi treinada. Para a seleção dos melhores atributos, entra-se em um laço de repetição. Se os resultados não fossem bons o suficiente, conforme o valor da acurácia, entraria em um *pipeline* que representa o GA. Nele, a escolha dos atributos mais importantes ocorreu. Caso a divisão não seja ótima, há a execução de um *crossover*, uma mutação e um *fitness* para uma modificação nos dados. A finalização do *pipeline* retorna as colunas mais relevantes que são devolvidas para

um novo treinamento com os algoritmos de aprendizado de máquina em um novo ciclo. Ademais, para avaliar se a seleção de dados foi eficiente, um experimento é executado com o conjunto de dados sintéticos proveniente da IBM [Padhi 2021]. Um dos melhores resultados obtidos com os dados reais do [Kaggle 2018] foi com a utilização da Floresta Aleatória com a seleção dos atributos V1, V5, V7, V8, V11, V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24 e *Amount* selecionados.

[Mniai et al. 2023] evidenciaram um *framework*, produzido por outros autores e referenciado no artigo, que utiliza técnicas de classificação baseadas em descrição de dados vetoriais de suporte. Em um primeiro momento, o conjunto de dados de detecção de fraudes de cartão de crédito [Kaggle 2018] passa por uma etapa de pré-processamento e reamostragem com *imblearn* (mantém a quantidade de dados das classes minoritárias, mas diminui a das majoritárias) para a posterior seleção dos melhores recursos. Em seguida, há a otimização de hiperparâmetros com *Polynomial Self Learning PSO* (PSLPSO), a escolha dos melhores atributos e a divisão dos dados. O PSLPSO é proposto baseado no PSO [Marini and Walczak 2015]. Para a seleção de atributos, três métodos são utilizadas com a obtenção de conjunto de atributos distintos, são eles: *filter*, *wrapper* e *embedded*. Para a fase subsequente, o *Support Vector Data Description* (SVDD) [Tax and Duin 2004] é usado para a classificação que identifica se os dados estão dentro ou fora de um limite esférico correspondido a uma região de recursos com tamanho mínimo ao redor de amostras positivas. O resultado, então, é comparado com outros algoritmos de Aprendizado de Máquina, a se considerar a Regressão Logística, a Árvore de Decisão, o SVM, o KNN e a Floresta Aleatória. O melhor resultado apresentado foi com a seleção de atributos incorporados (*embedded feature selection technique*).

[Salekshahrezaee et al. 2023] mostraram um estudo comparativo com relação aos métodos de balanceamento de dados e modelos de aprendizagem de máquina. A base de dados [Kaggle 2018] utilizada é a mesma do presente trabalho. Desse modo, foi preciso tomar medidas para mitigar as desvantagens do desbalanceamento no aprendizado. SMOTE, SMOTE Tomek [Tomek 1976] e RUS foram as técnicas usadas para tratar o desequilíbrio nas classes. Para extrair *features*, o PCA e o *Convolutional Autoencoder* (CAE) [Masci et al. 2011] foram as técnicas escolhidas. Cinco porções diferentes de dados foram usados em união ao RUS para escolher uma quantidade de fraudulentos e legítimos, na devida ordem, a se referir 1:1, 1:5, 1:10, 1:20 e 1:50. O parâmetro *k neighbors* é definido como cinco para a utilização junto com o SMOTE e o SMOTE Tomek. Após a implementação do PCA, a quantidade de atributos obtidos é de 15 componentes principais, o que corresponde à metade das características originais. O CAE é implementado usando Keras [Chollet et al. 2015] e TensorFlow [Abadi et al. 2015] com parâmetros otimizados selecionados durante experimentos preliminares. Além disso, os modelos aplicados no estudo, foram: Floresta Aleatória, LightGBM [Ke et al. 2017] e CatBoost [Prokhorenkova et al. 2018]. O destaque para melhor resultado geral foi com a utilização da Floresta Aleatória com RUS e CAE e o conjunto de dados foi composto da proporção um dado dos fraudulentos para 50, dos legítimos.

[Flondor et al. 2024] explicitaram o motivo de se utilizar a Árvore de Decisão para escolher os melhores atributos para o treinamento de dois *datasets*. Leva-se em consideração que os valores presentes em cada atributo podem ser muito distantes, por isso usou método para normalizar os dados. Essas bases de dados usadas foram geradas,

uma pela ferramenta Sparkov [Shenoy 2020] e a outra, pela IBM [Kaggle 2021]. Cada uma tem suas próprias características e, por meio da Árvore de Decisão, os atributos mais importantes para a detecção de fraudes de cartão de crédito foram escolhidos, visto que auxiliam na distinção dos dados. O resultado obtido para o conjunto de dados da IBM [Kaggle 2021] foi 99% para acurácia, F1-*Score*, precisão e revocação. Já para o outro [Shenoy 2020], 96%, 97%, 99% e 94% para a mesma ordem.

[Ileberi and Sun 2024] revelaram um meio mais eficiente de equilibrar a distribuição dos dados baseado no processo de ADASYN. Fornece uma representação proporcional da distribuição de dados conforme o nível de equilíbrio especificado por um coeficiente escolhido. Além disso, conta com o *Recursive Feature Elimination with Cross-Validation* (RFECV) [Misra and Yadav 2020], que elimina atributos menos significantes de modo recursivo em conjunto com a Validação Cruzada [Berrar et al. 2019]. O que implica muitos benefícios a se considerar: fornece tratamento dos dados de forma recursiva até obter um subconjunto ideal, evita *overfitting*, generaliza para dados não visualizados de forma confiável, traz robustez em relação ao conjunto de dados diferentes, dentre outros. Ademais, comparou seus resultados com outros trabalhos e com seu outro experimento sem o ADASYN com os resultados provenientes das métricas de classificação utilizadas. O uso da Floresta Aleatória e do XGBoost tiveram destaque dentre outros MLs, como Regressão Logística, Árvore de Decisão e *Light Gradient Boosting Model* (LGBM) [Ke et al. 2017]. Vale notabilizar que, a base de dados aplicada foi o *dataset* de detecção de fraudes de cartão de crédito do [Kaggle 2018].

3.3. Tratamento do desbalanceamento de dados

[Vihurskyi 2024] evidenciou que o *dataset* de detecção de fraudes de cartão de crédito [Kaggle 2018] possui dados duplicados. Além disso, adotou um processo de SMOTE nos dados e utilizou técnicas de *Machine Learning*, como Floresta Aleatória, Regressão Logística, Árvores de Decisão, KNN e *Naïve Bayes*. Somado a isso, foi descrita a metodologia *Local Interpretable Model-agnostic Explanations* (LIME) [Ribeiro et al. 2016], que está dentro da *Explainable Artificial Intelligence* (XAI) [Adadi and Berrada 2018], que consiste em explicar os processos de decisão em uma aprendizagem de máquina, esclarecer o comportamento do modelo. Consoante às medidas de desempenho e às análises empregadas, a Árvores de Decisão e a Floresta Aleatória foram consideradas os melhores algoritmos de aprendizado de máquina.

Tabela 2. Melhores resultados dos artigos relacionados

Autor	ML	Método de balanceamento	Seleção de atributos	Acurácia	Precisão	Revocação	AUC	F1-Score	Especificidade
[Ileberi et al. 2022]	RF	-	Sim	0,9994	0,7699	0,8969	0,96	0,8285	-
[Ahmad et al. 2023]	ANN	SBS	Sim	0,966	0,964	0,894	0.944	0.93	0.989
[Mniai et al. 2023]	SVDD	*imblearn*	Sim	0,93	0,9	0,97	-	0,93	-
[Salekshahrezaee et al. 2023]	RF	RUS	Sim	-	-	-	0,984	0,902	-
[Ileberi and Sun 2024]	XGB	ADASYN	Sim	0,9998	0,9992	0,9999	-	0,9995	-
[Rinku et al. 2024]	RF	ADASYN	Sim	0,9995	0,8485	0,8571	0,97	0,8528	0,9997
[Sreekala et al. 2024]	RF	-	Não	0,997	0,995	0,953	-	0,974	-
[Vihurskyi 2024]	DT	SMOTE	Não	1	1	1	0,99	1	-

Diante dos estudos, o corrente trabalho vem com o intuito de analisar o impacto da separação dos dados e da seleção de atributos tanto abordadas as metodologias nos artigos quanto introduzindo experimentos novos. Além disso, comparar experimentos semelhantes e como mudanças de abordagens podem impactar o aprendizado do modelo.

4. Metodologia

A seção tem como finalidade apresentar os recursos, as técnicas e os procedimentos utilizados ao longo do desenvolvimento do trabalho. Descreve os artefatos utilizados e produzidos a se citar o conjunto de dados [Kaggle 2018], o delineamento experimental, os testes realizados e os métodos optados para manipulação e avaliação da base. Há também a discussão do nível de relevância de cada atributo para o aprendizado, bem como o uso de métricas e métodos para comparação e avaliação dos algoritmos selecionados.

A Figura 3 mostra as etapas dos experimentos feitos.

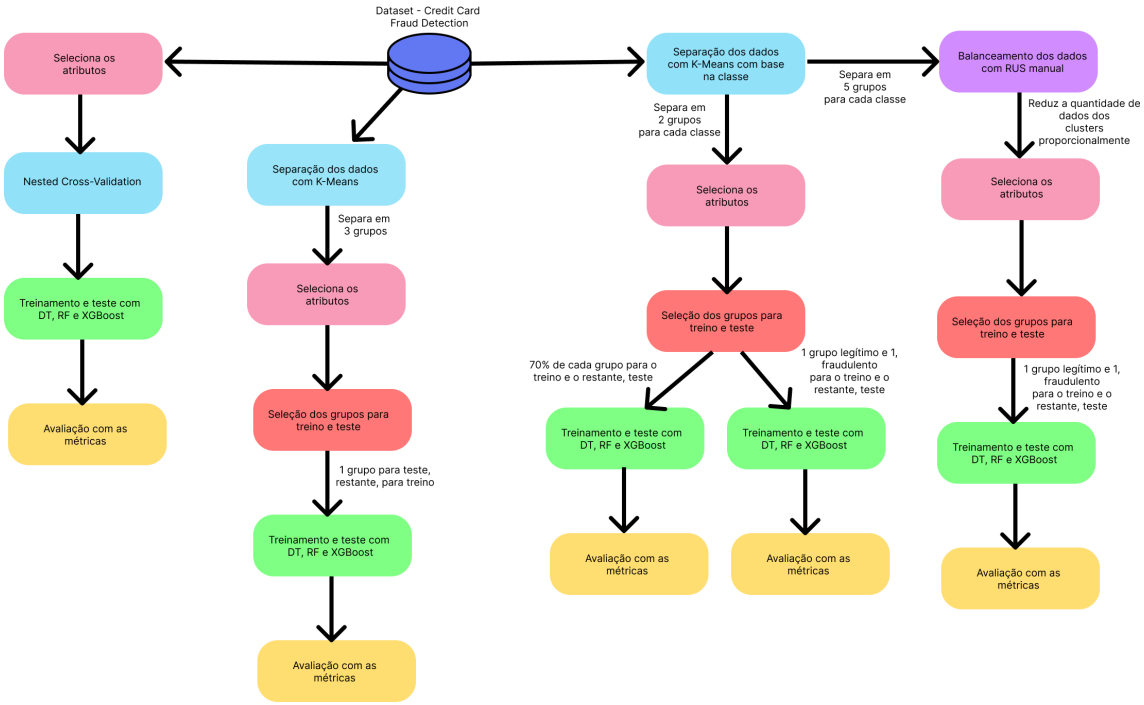


Figura 3. Etapas dos experimentos

4.1. Conjunto de dados

O presente trabalho utiliza o *dataset* de detecção de fraudes de cartão de crédito (em inglês, Credit Card Fraud Detection) disponibilizado pelo [Kaggle 2018] que possui 473 registros de transações classificadas como fraudulentas e 283253, como legítimas. Ressalta-se que, o conjunto original apresenta dados duplicados, o que justifica essa menção em relação a de outros estudos que citam 284807 transações, das quais 492 são fraudulentas. A base de dados conta com 30 atributos: dois atributos originais - o *Amount* (valor da transação) e o *Time* (tempo decorrido desde a primeira transação) - e os 28 mascarados para manter a privacidade, nomeados do V1 ao V28. Além desses, há o atributo classe, denominada '*Class*', em que atribui valor 0 para as transações legítimas e 1, para as fraudulentas.

A escolha desse conjunto de dados se deve ao fato de possuir dados reais, ser usado de forma ampla na literatura atualmente e ainda estar disponível para uso.

4.2. Escolha dos atributos

Considerando os benefícios descritos por [Flondor et al. 2024] em relação ao uso da Árvore de Decisão, esse método foi escolhido para constituir uma das abordagens para a escolha dos atributos. Conforme mencionado na Seção 2.2.1, as Árvores de Decisão são capazes de lidar com dados categóricos e contínuos, além de serem interpretáveis. Essas características as tornam adequadas para a diversidade de dados presentes em transações de cartão de crédito ao passo que permite o rastreio do processo da tomada de decisão e a compreensão da lógica da classificação pelos analistas.

O treinamento com a Árvore de Decisão com a base inteira revelou atributos que contribuem para o modelo, enquanto outros não apresentam essa importância. Em particular, os atributos V5, V9, V21 e V28 não se mostram úteis para o processo de aprendizado do modelo.

Os atributos que corroboram com o aprendizado em diferentes cenários são:

a) V1, V10, V13, V15, V19, V27 - para os dados verídicos em caso de não retirada de nenhum atributo;
b) V1, V7, V11, V13, V23 - para os dados verídicos em caso de retirada dos quatro atributos citados mais acima;
c) V19, V25 - para os dados fraudulentos em caso de não retirada de nenhum atributo;
d) V4, V19, V15, V20 - para os dados fraudulentos em caso de retirada dos quatro atributos citados mais acima.

Além dessa abordagem, outra foi adotada para seleção de atributos com base em análise visual dos dados. A seleção envolveu os atributos V3, V4, V5, V7, V9, V10, V11, V12, V14, V16, V17, V19 e *Amount*, decididos a partir da inspeção do gráfico do tipo *Violin plot* e de representações gráficas simples com o uso da biblioteca Matplotlib [Hunter 2007]. Combinações de atributos com *Parallel line plot* da mesma biblioteca também foram exploradas.

A seleção foi guiada pela observação individual dos atributos com a finalidade de encontrar distribuições distintas quando comparados aos gráficos dos violinos dos dados verídicos e dos fraudulentos, sendo os limites diferentes. Por conseguinte, os atributos cujos violinos evidenciavam separabilidade entre as classes foram mantidos, visto que, após treino e teste, demonstraram impacto positivo para o desempenho do modelo.

Para fins de organização e referência nas tabelas e figuras, foram adotadas as seguintes classificações para os conjuntos de atributos selecionados:

a) Conjunto I: Seleção dos atributos excluindo V5, V9, V21 e V28;
b) Conjunto II: Seleção dos atributos excluindo *Time*, V1, V2, V6, V8, V13, V15, V18, V20, V21, V22, V23, V24, V25, V26, V27, V28;
c) Conjunto III: Seleção de todos os atributos, nenhum é excluído.

4.3. Testes realizados

Os testes realizados consideram diferentes combinações de modelos, métodos e métricas previamente apresentados. Vale lembrar que, todos os experimentos usam os algoritmos de aprendizado de máquina: Árvore de Decisão e Floresta Aleatória da biblioteca Sklearn e XGBoost, da biblioteca XGBoost. As abordagens experimentais aplicadas foram:

a) Aplicação da técnica de Validação Cruzada Aninhada com os algoritmos;
b) Emprego do *K-Means* para segmentação dos dados, seguido do treinamento com os algoritmos;
c) Utilização do *K-Means* com separação prévia dos dados com fundamento nas classes (legítima ou fraudulenta), seguida do treinamento com os algoritmos;
d) Uso dos *clusters* do experimento anterior para selecionar 70% dos dados de cada grupo para o treino, o restante, para o teste;
e) Treinamento com os algoritmos com o uso dos dados agrupados com *K-Means* e balanceados por meio do método RUS.

Conforme citado por [Ahmad et al. 2023], algoritmos de aprendizado de máquina tendem a favorecer o grupo majoritário durante o treinamento, em caso do conjunto de dados ser desbalanceado, o que pode ocasionar em previsões equívocas, uma vez que compromete a detecção de instâncias da classe minoritária. Portanto, para mitigar esse problema, um dos testes inclui a aplicação do método RUS de forma manual. Assim, reduz-se o número de amostras da classe majoritária, tornando o conjunto de dados mais balanceado. Já que essa técnica é muito utilizada na literatura e apresentou bons resultados para os experimentos do [Salekshahrezaee et al. 2023].

Ademais, a técnica de Validação Cruzada Aninhada é utilizada para avaliar o desempenho de modelos submetidos ao ajuste de hiperparâmetros com a separação adequada dos dados em treino, teste e validação. Durante sua execução nos experimentos, foram otimizados alguns hiperparâmetros, como: *gini*, *entropy*, *log_loss*.

5. Experimentos e resultados

A seção aborda os resultados dos experimentos realizados a partir das diferentes combinações de modelos e técnicas analisados com base nas métricas mostradas anteriormente para uma visão mais ampla dos experimentos, visando responder os questionamentos levantados na Introdução, Seção 1.

5.1. Configuração experimental

A configuração do computador utilizado é AMD Ryzen 5 5600 com 32,0 GB de RAM e processador de 3,50 GHz com sistema operacional Windows 10 de 64 bits.

5.2. Resultados e discussões

5.2.1. Separação com Validação Cruzada Aninhada

A Tabela 3 refere-se aos melhores e piores resultados obtidos com a Árvore de Decisão, a Floresta Aleatória e o XGBoost com o uso do Validação Cruzada Aninhada e com ou sem a exclusão dos atributos.

Os resultados, em sua maioria, obtiveram valores bons, sendo as seleções de atributos com maior destaque a I e a III. Vale destacar que, a melhor média do experimento foi obtida tanto com a utilização da seleção I quanto da III com a Floresta Aleatória.

Tabela 3. Média dos resultados com Validação Cruzada Aninhada

ML	Seleção de atributos	Precisão	Revocação	F1-Score	AUC
DT	I	0,81	0,83	0,82	0,91
RF	**I**	**0,96**	**0,87**	**0,91**	**0,93**
XGB	I	0,96	0,82	0,88	0,91
DT	II	0,79	0,83	0,81	0,91
RF	**II**	**0,97**	**0,84**	**0,9**	**0,92**
XGB	II	0,93	0,83	0,88	0,91
DT	III	0,81	0,84	0,82	0,92
RF	**III**	**0,96**	**0,87**	**0,91**	**0,93**
XGB	III	0,94	0,84	0,89	0,92

5.2.2. Separação em três agrupamentos com K-Means

A Tabela 4 indica os resultados provenientes dos mesmos algoritmos de aprendizado de máquina supervisionado após a aplicação do *K-Means* para separar o conjunto de dados em três agrupamentos, valor proveniente tanto do método do cotovelo quanto citado pelo artigo [Sreekala et al. 2024]. Os grupos de treino e de teste são formados por permutações, onde dois grupos compõem o treino e um, o teste. Destaca-se que o número dos grupos estão nas colunas 'Grupo treino' e 'Grupo teste' indicados na Tabela 4. Os agrupamentos podem ser visualizados na Figura 4, onde é feito a *clusterização* com todos os dados e atributos antes de selecioná-los.

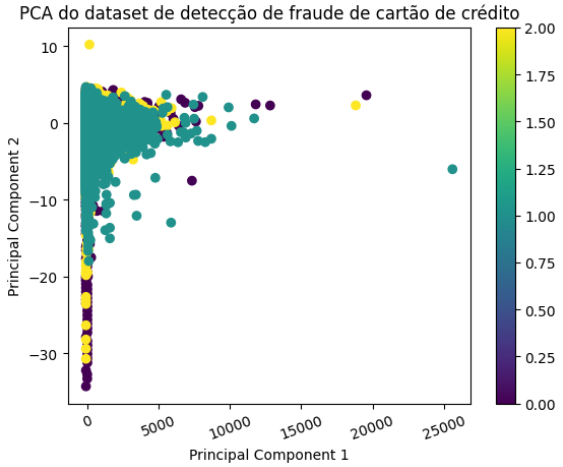


Figura 4. Dados agrupados com visualização com PCA

Diante dos cenários com exclusão de atributos, as seleções I e II se destacaram de um modo consistente em relação aos resultados de todas as métricas com os grupos 0 e 2 para compor o conjunto de treino. Sem a exclusão de qualquer atributo, a configuração que obteve destaque foi a com os grupos 1 e 2 para o treino. Apesar dos valores do AUC serem bons, as demais métricas não tiveram resultados satisfatórios.

Desse modo, entende-se que a estrutura do experimento não é adequada, visto que

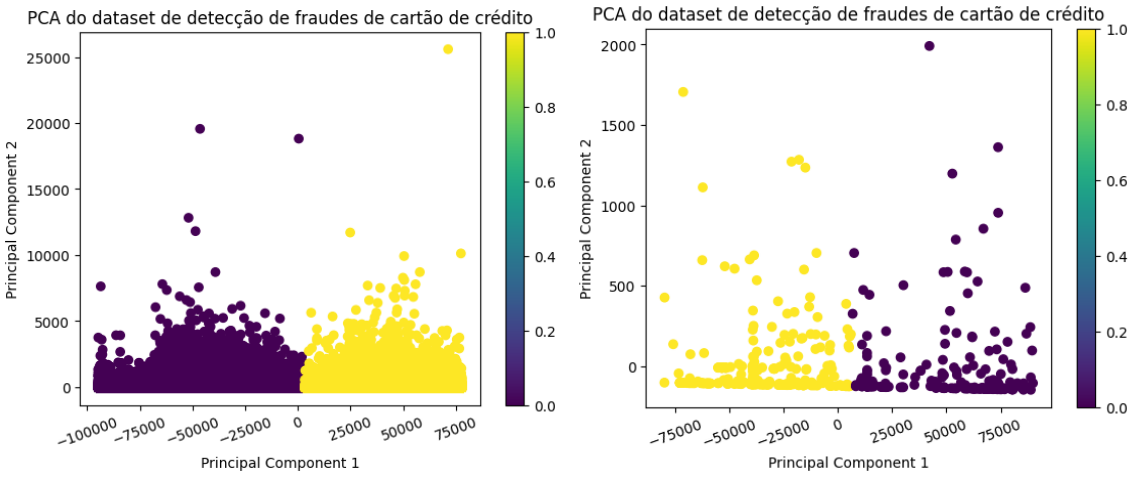
Tabela 4. Média dos resultados com *K-Means* dividindo o conjunto de dados

Seleção de atributos	Grupo treino	Grupo teste	Precisão	Revocação	F1-Score	AUC
I	1 e 2	0	0,59	0,62	0,6	0,81
I	**0 e 2**	**1**	**0,69**	**0,65**	**0,66**	**0,82**
I	0 e 1	2	0,66	0,66	0,65	0,83
II	1 e 2	0	0,61	0,62	0,61	0,81
II	**0 e 2**	**1**	**0,71**	**0,65**	**0,67**	**0,82**
II	0 e 1	2	0,69	0,65	0,67	0,82
III	**1 e 2**	**0**	**0,71**	**0,76**	**0,72**	**0,88**
III	0 e 2	1	0,75	0,73	0,73	0,86
III	0 e 1	2	0,73	0,71	0,71	0,85

une-se amostras dos dados parecidos para testar o treinamento com outra que não possui tantas semelhanças com os dois grupos. Isso torna o cenário de teste muito difícil, o que explica os resultados.

5.2.3. Separação em dois agrupamentos por classe com K-Means

A Tabela 5 mostra os resultados da divisão dos dados em dois *clusters* para cada classe - legítima e fraudulenta - com a utilização do *K-Means*. O número de *clusters* obtidos para cada classe é proveniente da Silhueta com o Sklearn e do método do cotovelo. A permutação entre os grupos de cada classe constituem os conjuntos de treino e teste. Vale destacar que, os identificadores numéricos referem-se aos dados legítimos seguidos pelos fraudulentos. As Figuras 5(a) e 5(b), obtidas por intermédio do PCA da biblioteca Sklearn, correspondem a separação dos dados legítimos e fraudulentos na devida sequência.



(a) Dados legítimos

(b) Dados fraudulentos

Figura 5. Dados agrupados com visualização com PCA

Nesse experimento, em que os dados foram agrupados de acordo com suas classes, o grupo 1 dos legítimos com o 0 dos fraudulentos foi a composição que obteve melhor resultado, em questão de AUC, ao ser comparada com as demais configurações dentro

Tabela 5. Média dos resultados com *K-Means* subdividindo as classes

Seleção de atributos	Grupo treino	Grupo teste	Acurácia	Precisão	Revocação	F1-Score	AUC
I	1 e 1	0 e 0	0,43	0,21	0,39	0,21	0,41
I	**1 e 0**	**0 e 1**	**0,57**	**0,31**	**0,34**	**0,29**	**0,45**
I	0 e 1	1 e 0	0,24	0,05	0,5	0,1	0,37
I	0 e 0	1 e 1	0,38	0,17	0,45	0,18	0,41
II	1 e 1	0 e 0	0,99	0,48	0,49	0,48	0,74
II	**1 e 0**	**0 e 1**	**0,99**	**0,53**	**0,53**	**0,52**	**0,76**
II	0 e 1	1 e 0	0,99	0,39	0,46	0,41	0,73
II	0 e 0	1 e 1	0,99	0,43	0,49	0,44	0,74
III	1 e 1	0 e 0	0,42	0,23	0,38	0,21	0,4
III	**1 e 0**	**0 e 1**	**0,57**	**0,32**	**0,34**	**0,3**	**0,46**
III	0 e 1	1 e 0	0,25	0,09	0,44	0,07	0,35
III	0 e 0	1 e 1	0,39	0,18	0,42	0,16	0,41

da própria seleção de atributos. Contudo, nenhuma média apresentada mostra o impacto positivo para esse experimento.

Esses valores das métricas inferiores à 0,5 e próximos de zero podem ser justificados pela falta de representatividade dos dados no conjunto de treinamento.

Vale destacar que, nesse cenário, a seleção que obteve melhores resultados em todas as métricas se comparados com os de configuração igual é a II.

5.2.4. Separação em dois agrupamentos por classe com K-Means com 70/30

A fim de compreender se os valores próximos a zero são provenientes da falta de representatividade dos dados no conjunto de treinamento, esse experimento é feito. A Tabela 6 mostra os resultados da escolha de 70% dos dados de cada *cluster* do experimento anterior para compor o conjunto de treino e o restante, para teste.

Tabela 6. Média dos resultados com *K-Means* subdividindo as classes separando 70/30

ML	Seleção de atributos	Precisão	Revocação	F1-*Score*	AUC
DT	I	0,71	0,73	0,72	0,86
RF	**I**	**0,93**	**0,76**	**0,84**	**0,88**
XGB	I	0,87	0,73	0,8	0,86
DT	II	0,73	0,75	0,74	0,87
RF	**II**	**0,93**	**0,76**	**0,84**	**0,88**
XGB	II	0,87	0,73	0,8	0,86
DT	III	0,73	0,74	0,73	0,87
RF	**III**	**0,93**	**0,76**	**0,84**	**0,88**
XGB	III	0,87	0,73	0,8	0,86

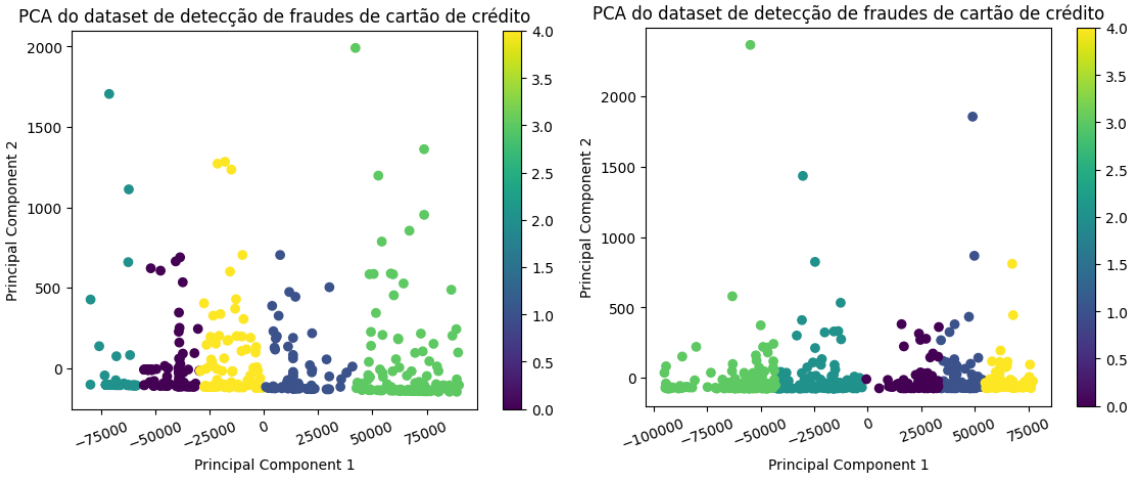
Com as médias dos resultados, nota-se que houve pouca variação dos modelos em relação ao critério de seleção de atributos. A seleção II obteve resultados timidamente mais altos em relação aos demais, sendo sua união com a Floresta Aleatória, o melhor

resultado. Contudo, o valor da revocação não é um valor aceitável diante do cenário de detecção de fraudes de cartão de crédito onde se deseja validar as transações legítimas.

Os resultados das métricas foram grandes em relação ao Experimento 5.2.3, indicando que o modo de separação dos dados desse para os conjuntos de treino e teste não foi adequado. É provável que muitos dados relevantes tenham sido excluídos do treinamento, o que comprometeu a capacidade de aprendizado do modelo.

5.2.5. Separação com K-Means e balanceamento com RUS

A Tabela 7 evidencia os resultados obtidos com a técnica RUS, feita de modo manual. De início, o *K-Means* da biblioteca Scipy [Virtanen et al. 2020] é usado para encontrar cinco *clusters* de cada conjunto de dados legítimos e fraudulentos. Em sequência, a quantidade de instâncias nos agrupamentos dos dados verídicos é reduzida para igualar à dos fraudulentos. Logo após, uma permutação é feita para selecionar um grupo de cada classe para formar o conjunto de teste e os demais, o conjunto de treino. Vale destacar que, os valores informados na coluna de 'Grupo teste' referem-se aos *clusters* legítimos e fraudulentos respectivamente. A Figura 6 designa as separações feitas via PCA da biblioteca Sklearn referentes aos dados sem a exclusão dos atributos, que posteriormente serão submetidos a escolha dos atributos - I, II ou III.



(a) Dados legítimos (b) Dados fraudulentos

Figura 6. Dados agrupados com visualização com PCA

Tabela 7. Média dos resultados com *K-Means* subdividindo as classes e utilizando RUS

Seleção de atributos	Grupo teste	Acurácia	Precisão	Revocação	F1-Score	AUC
I	0 e 0	0,83	0,79	0,97	0,86	0,84
I	0 e 1	0,85	0,85	0,89	0,86	0,85
I	0 e 2	0,72	0,79	0,76	0,76	0,73
I	0 e 3	0,79	0,78	0,92	0,83	0,8
I	0 e 4	0,74	0,77	0,82	0,77	0,75
I	1 e 0	0,82	0,78	0,97	0,85	0,83
I	1 e 1	0,84	0,84	0,88	0,86	0,84
I	1 e 2	0,7	0,77	0,75	0,75	0,71
I	1 e 3	0,77	0,76	0,92	0,81	0,78
I	1 e 4	0,73	0,75	0,82	0,76	0,74
I	2 e 0	0,83	0,78	0,97	0,85	0,84
I	**2 e 1**	**0,85**	**0,86**	**0,88**	**0,87**	**0,86**
I	2 e 2	0,68	0,74	0,74	0,73	0,68
I	2 e 3	0,78	0,76	0,92	0,81	0,79
I	2 e 4	0,74	0,76	0,82	0,77	0,75
I	3 e 0	0,81	0,78	0,97	0,85	0,83
I	3 e 1	0,85	0,86	0,88	0,86	0,85
I	3 e 2	0,68	0,75	0,73	0,73	0,69
I	3 e 3	0,76	0,74	0,92	0,8	0,77
I	3 e 4	0,72	0,74	0,82	0,76	0,73
I	4 e 0	0,79	0,74	0,97	0,83	0,8
I	4 e 1	0,83	0,84	0,88	0,85	0,84
I	4 e 2	0,65	0,72	0,72	0,7	0,65
I	4 e 3	0,73	0,71	0,93	0,78	0,75
I	4 e 4	0,69	0,7	0,82	0,73	0,70
II	0 e 0	0,95	0,92	0,98	0,94	0,95
II	0 e 1	0,91	0,93	0,88	0,9	0,91
II	0 e 2	0,89	0,96	0,84	0,9	0,90
II	0 e 3	0,93	0,92	0,94	0,93	0,93
II	0 e 4	0,88	0,92	0,82	0,87	0,88
II	1 e 0	0,95	0,92	0,98	0,94	0,95
II	1 e 1	0,91	0,93	0,88	0,9	0,91
II	1 e 2	0,89	0,96	0,84	0,9	0,90
II	1 e 3	0,93	0,92	0,94	0,93	0,93
II	1 e 4	0,88	0,92	0,82	0,87	0,88
II	2 e 0	0,94	0,91	0,98	0,94	0,95
II	2 e 1	0,91	0,93	0,88	0,90	0,91
II	2 e 2	0,89	0,95	0,84	0,90	0,90
II	2 e 3	0,93	0,91	0,94	0,93	0,93
II	2 e 4	0,88	0,92	0,82	0,87	0,88
II	**3 e 0**	**0,95**	**0,92**	**0,98**	**0,95**	**0,95**
II	3 e 1	0,91	0,93	0,88	0,90	0,91
II	3 e 2	0,89	0,95	0,85	0,90	0,90
II	3 e 3	0,93	0,92	0,94	0,93	0,93
II	3 e 4	0,88	0,93	0,82	0,87	0,88
II	4 e 0	0,94	0,91	0,98	0,94	0,95
II	4 e 1	0,90	0,92	0,88	0,90	0,90
II	4 e 2	0,89	0,95	0,85	0,89	0,89
II	4 e 3	0,93	0,91	0,94	0,92	0,93
II	4 e 4	0,88	0,92	0,82	0,87	0,88
III	**0 e 0**	**0,84**	**0,81**	**0,97**	**0,87**	**0,85**
III	0 e 1	0,85	0,85	0,88	0,86	0,85
III	0 e 2	0,72	0,79	0,76	0,76	0,73
III	0 e 3	0,79	0,77	0,92	0,82	0,80
III	0 e 4	0,75	0,77	0,82	0,78	0,76
III	1 e 0	0,83	0,79	0,97	0,86	0,84
III	1 e 1	0,84	0,85	0,88	0,86	0,84
III	1 e 2	0,70	0,77	0,75	0,75	0,71
III	1 e 3	0,77	0,75	0,91	0,81	0,78
III	1 e 4	0,74	0,76	0,82	0,77	0,75
III	2 e 0	0,83	0,78	0,97	0,85	0,84
III	2 e 1	0,85	0,85	0,88	0,86	0,85
III	2 e 2	0,68	0,74	0,74	0,73	0,69
III	2 e 3	0,78	0,77	0,92	0,82	0,80
III	2 e 4	0,75	0,77	0,82	0,77	0,76
III	3 e 0	0,81	0,78	0,97	0,85	0,83
III	3 e 1	0,84	0,85	0,88	0,86	0,84
III	3 e 2	0,68	0,76	0,74	0,73	0,69
III	3 e 3	0,76	0,75	0,91	0,81	0,78
III	3 e 4	0,73	0,75	0,82	0,76	0,74
III	4 e 0	0,79	0,74	0,97	0,83	0,80
III	4 e 1	0,82	0,83	0,88	0,84	0,83
III	4 e 2	0,65	0,72	0,72	0,70	0,66
III	4 e 3	0,74	0,72	0,92	0,79	0,76
III	4 e 4	0,70	0,71	0,82	0,74	0,71

Vale ressaltar que, os melhores resultados dentre as seleções de atributos têm como critério para essa eleição o maior AUC, posteriormente F1-*Score*, revocação e precisão.

Diante disso, apontamentos relevantes podem ser feitos pelo uso do K-*Means* e *undersampling* nos experimentos. O grupo 0 dos fraudulentos no grupo teste foi essencial para um bom resultado, o que indica que os grupos 1, 2, 3 e 4 possuem dados relevantes para o processo de treinamento. Já para os legítimos, é difícil elencar um melhor, está atrelado tanto aos grupos fraudulentos quanto a seleção de atributos. Além disso, a seleção II foi a que alcançou resultados satisfatórios, principalmente em relação ao AUC.

Logo, a escolha dos agrupamentos para compor os conjuntos de treino e teste é crucial visto que uma mínima mudança pode acarretar em variações de desempenho.

5.2.6. Análise dos resultados

É importante ressaltar que, em relação aos grupos mencionados anteriormente, os dados agrupados em um mesmo *cluster* são equivalentes, porém, após a criação dos agrupamentos com o *K-Means*, informações são retiradas. É nessa fase posterior à *clusterização* que é feito a seleção dos atributos. Ademais, quando não aparece a coluna da acurácia, significa que todos os resultados foram 0,99.

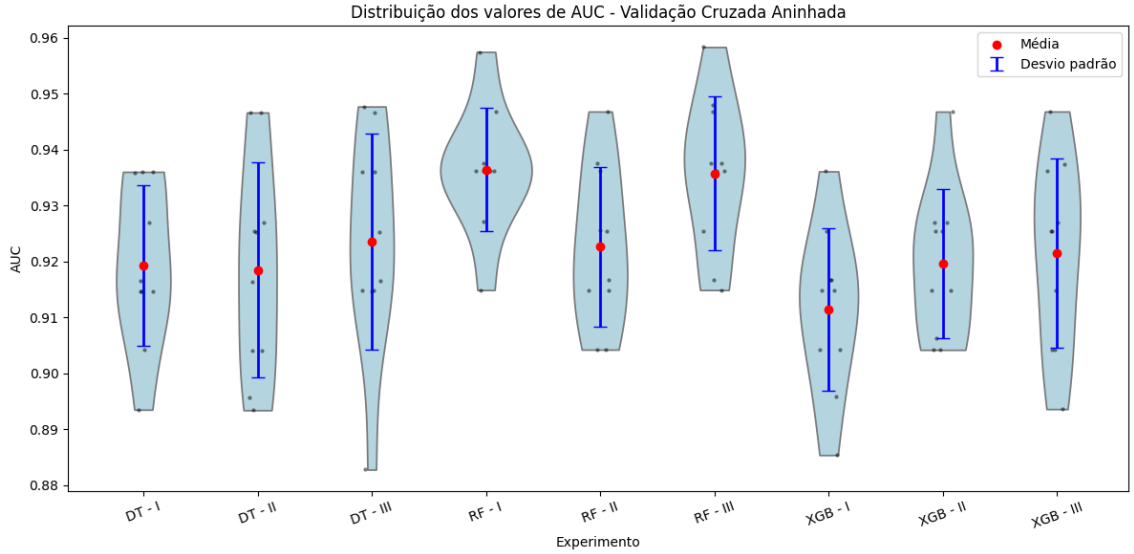


Figura 7. Desvio padrão dos resultados com Validação Cruzada Aninhada

A Validação Cruzada Aninhada apresenta bons resultados, como apresentada pela Figura 7, tendo o desvio padrão com valores baixo com todas as formas de seleção de atributos e modelos.

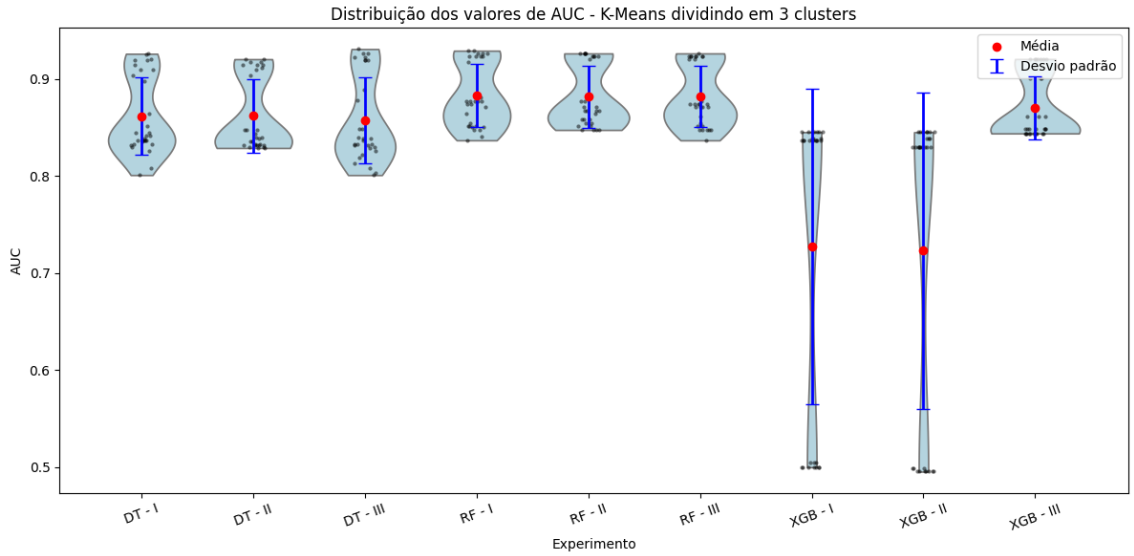


Figura 8. Desvio padrão dos resultados com *K-Means* dividindo o conjunto de dados

Com o uso do *K-Means* para separar os dados em três *clusters*, o XGBoost foi o menos consistente nas separações I e II, mostrado na Figura 8. No entanto, na seleção III, o desempenho desse modelo se assemelha aos demais, indicando que uma maior quantidade de atributos foi benéfica.

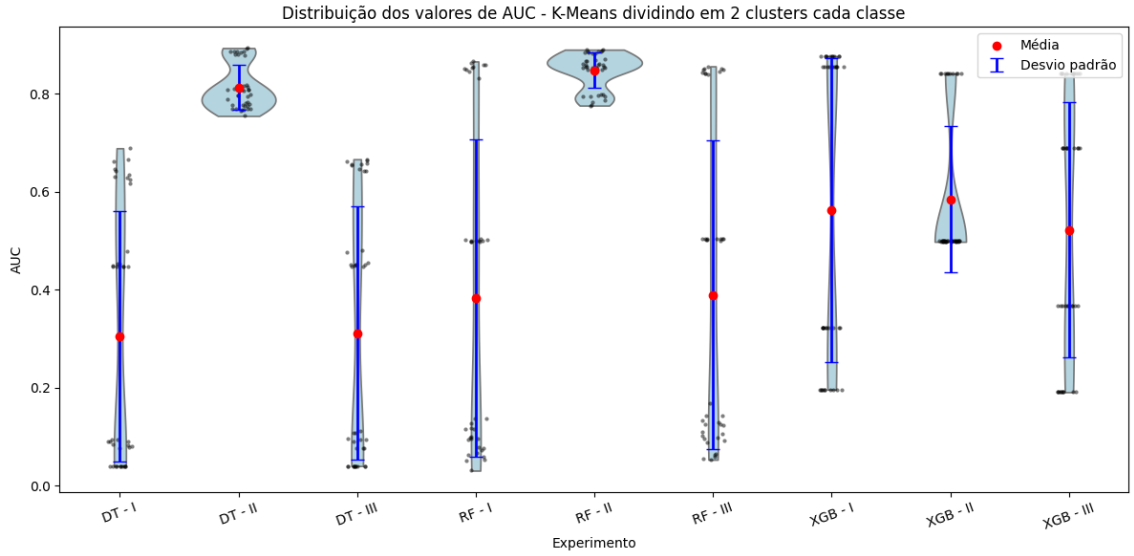


Figura 9. Desvio padrão dos resultados com *K-Means* subdividindo as classes

Com o uso do *K-Means* para separar os dados em dois *clusters* para cada classe, as seleções I e III foram muito inconsistentes com seus modelos, como indicado na Figura 9. Já a II, teve desempenho consistente com altos resultados, exceto com o XGBoost.

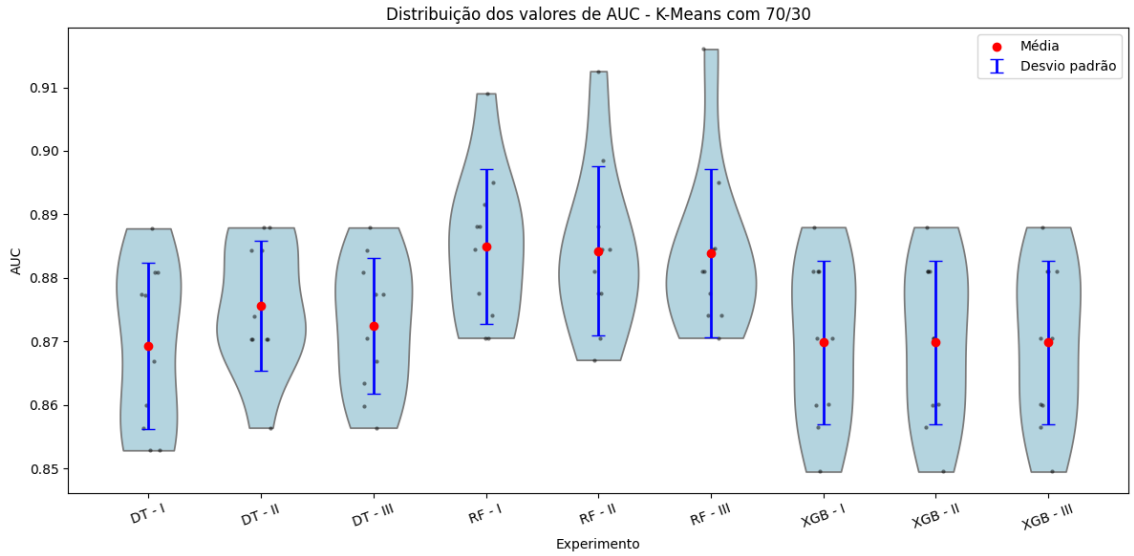


Figura 10. Desvio padrão dos resultados com *K-Means* com 70/30

O uso da divisão dos *clusters* em 70% dos dados para o treinamento e 30%, para o teste ocasionou em uma melhoria substancial nos resultados e consistência nas seleções. Isso pode ser visualizado na Figura 10.

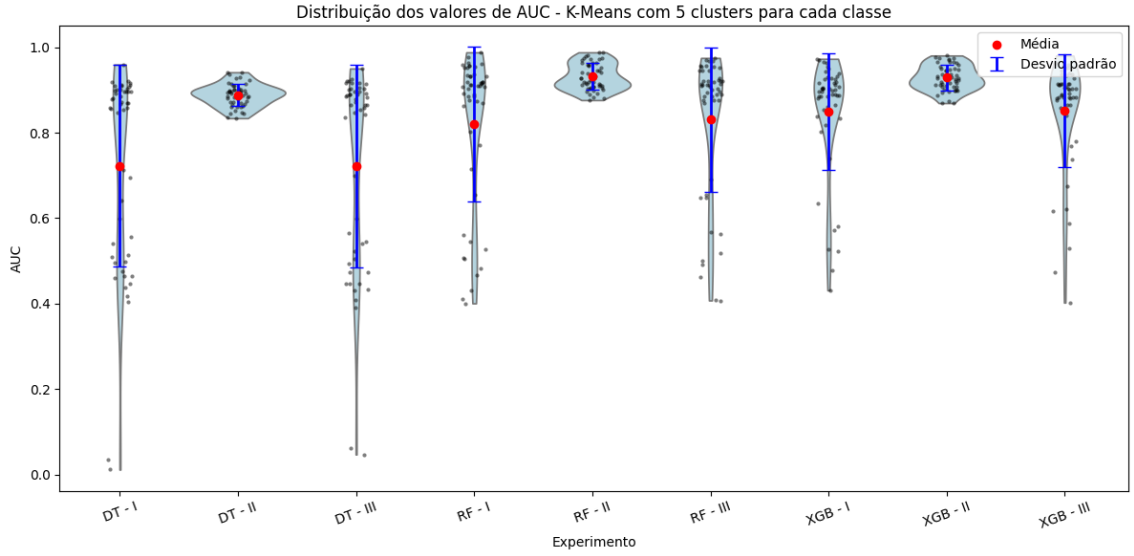


Figura 11. Desvio padrão dos resultados com *K-Means* com RUS

Com o uso do RUS, observa-se que uma maior quantidade de atributos, provenientes das seleções I e III, atrapalhou nos aprendizados. Todavia, a seleção II demonstrou consistência em seus resultados nos três modelos, visualizado na Figura 11.

6. Conclusão

O processo de separação de dados é fundamental durante o treinamento de algoritmos de aprendizado de máquina, assim como a seleção de atributos, conforme discutido em seis dos dez artigos apresentados na Seção 3. A combinação das abordagens resultou

em desempenhos comparáveis aos trabalhos que utilizam o *dataset* do [Kaggle 2018], demonstrado na Tabela 2.

Ademais, experimentos que utilizaram o Validação Cruzada Aninhada atingiram resultados expressivos em conjunto com o classificador Floresta Aleatória e o XGBoost. No que se refere ao balanceamento de dados, a aplicação manual do RUS em cada agrupamento produzido pelo *K-Means* também forneceu bons resultados com a seleção II, sendo o maior AUC com os grupos 0 e 0 para teste. Não obstante, os demais experimentos que utilizaram o *K-Means* não apresentaram desempenho satisfatório, assim como alguns presentes no experimento com RUS, o que pode ser atribuído à escolha dos grupos que compuseram os conjuntos de treino e teste. Para essa caso, o experimento com 70/30 afirmou que há necessidade de ter um conjunto de treino e teste mais representativo com o uso das informações de todos os *clusters*.

Os diferentes modos de execução com aprendizado de máquina evidenciam a importância das etapas de pré-processamento, dado que a maioria dos resultados apresentados, que correspondem a processos semelhantes aos estudos [Sreekala et al. 2024] e [Ahmad et al. 2023], foram inferiores. Vale lembrar que, as principais diferenças entre o presente trabalho e o do [Ahmad et al. 2023] residem nos algoritmos de aprendizado de máquina utilizados, nas porções dos dados empregados tanto na etapa do balanceamento quanto na etapa de seleção para os conjuntos de treino e teste e no método de balanceamento. Em relação ao estudo de [Sreekala et al. 2024], o processo foi semelhante, no entanto, utilizou uma pseudo-rotulagem durante o treinamento, técnica não incorporada no presente trabalho.

Vale ressaltar que, o melhor resultado obtido por este trabalho foi proveniente do *K-Means* com o RUS e a seleção II, em que os valores são 0,95 para a acurácia, 0,92 para a precisão, 0,98 para a revocação, 0,95 para o F1-*Score* e 0,95 para o AUC. Isso se deve a representatividade significante dos dados, por separar o conjunto em cinco, e o balanceamento desses, além de utilizar um dos algoritmos mais robustos e usados atualmente, a Floresta Aleatória. Nesse caso, a não exclusão de atributos foi importante para o melhor desempenho. Esses valores são superiores aos resultados dos seguintes estudos: [Ahmad et al. 2023], [Sreekala et al. 2024], [Rinku et al. 2024], [Salekshahrezaee et al. 2023], [Ileberi et al. 2022] e do [Mniai et al. 2023]. Contudo, obteve resultados inferiores ao ser comparado aos valores dos estudos dos autores [Ileberi and Sun 2024] , [Vihurskyi 2024] e [Sreekala et al. 2024].

Sobre o questionamento para introdução do trabalho, é possível constar que a separação dos dados impacta no aprendizado, visto que os resultados mais consistentes são provenientes da Validação Cruzada Aninhada e do 70/30. Em relação às seleções de atributos, o uso da II ajudou na aprendizagem dos modelos, reduzindo tempo e proporcionando a exclusão de atributos que não são tão relevantes, processo apoiado pelo estudo de [Ileberi et al. 2022].

Sobre os modelos, não há uma consistência independente do cenário proposto, como apontados das Figuras 7, 10 e 9, em que a Floresta Aleatória, a Árvore de Decisão e o XGBoost são apresentam consistência nos dois primeiros experimentos, mas não no último.

Com isso, nota-se que o modo de execução da Validação Cruzada Aninhada, além

de otimizar os hiperparâmetros, foi capaz de ajudar no aprendizado obtendo excelentes resultados. Entretanto, seu desempenho é insatisfatório, levando em consideração que seus resultados são próximos ao do experimento com RUS gastando menos recursos e tempo com a seleção de atributos II. Isso se deve ao fato que quando associado a *datasets* como o de detecção de fraudes de cartão de crédito do [Kaggle 2018], utiliza-se um tempo substancial, visto que para a execução de cada modelo com suas especificações foram, aproximadamente, cinco dias. Enquanto a execução completa das outras quatro metodologias somadas durou aproximadamente oito horas, sendo o RUS a metodologia mais rápida tendo demorado cerca de 8 minutos.

Em síntese, este trabalho tem como contribuição no aspecto de mostrar o impacto das seleções de atributos e da separação dos dados, não apenas analisando um modelo e conjunto de técnicas para encontrar um melhor método para aprendizagem de um modelo. Como apontado, a seleção II, apesar de excluir 17 atributos, foi a configuração que mais obteve resultados bons e foi mais consistente em relação às métricas para o cenário de detecção de fraudes de cartão de crédito. Além disso, agrupamentos específicos de *clusters* para o treinamento foram responsáveis por resultados bons e ruins, o que pode ser inferido a questão dos dados presentes neles serem relevantes para um aprendizado excelente.

Em vista dos experimentos feitos, trabalhos futuros podem aprofundar-se na aplicação de subamostragem guiada por agrupamentos provenientes do *K-Means* em combinação com a Validação Cruzada Aninhada aplicando-se uma técnica de balanceamento de dados por conta que essa validação é mais utilizada com conjuntos de dados pequenos. Além de utilizar técnicas de *undersampling*, um trabalho futuro pode trazer técnicas de *oversampling*, como o ADASYN. Uma investigação mais detalhada pode ser feita para analisar os dados que mais influenciam o desempenho entre os conjuntos de treino e teste. Além disso, abordar outros algoritmos de aprendizado de máquina mais utilizados no momento pode contribuir para resultados mais robustos e generalizáveis.

7. Agradecimentos

Agradecimentos a Deus por nos manter firmes e persistentes até o presente momento, ao nosso orientador pelos auxílios, pelas opiniões, correções e por toda orientação dada durante a execução do trabalho, à UFMS por todo o suporte dado durante a nossa formação no curso de Ciência da Computação e, não menos importante, aos nossos pais, familiares e amigos pelo apoio e companheirismo.

Referências

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

Ahmad, H., Kasasbeh, B., Aldabaybah, B., and Rawashdeh, E. (2023). Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (sbs). *International Journal of Information Technology*, 15(1):325–333.

Arsanjani, R., Dey, D., Khachatryan, T., Shalev, A., Hayes, S. W., Fish, M., Nakanishi, R., Germano, G., Berman, D. S., and Slomka, P. (2015). Prediction of revascularization after myocardial perfusion spect by machine learning in a large population. *Journal of Nuclear Cardiology*, 22(5):877–884.

Berrar, D. et al. (2019). Cross-validation.

Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9).

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

CNN (2024). Tentativas de fraude já somam r$ 1,2 bi em 2024; veja data, hora e alvos preferidos dos golpistas. Disponível em: `https://www.cnnbrasil.com.br/economia/financas/tentativas-de-fraude-ja-somam-r-12-bi-em-2024-veja-data-hora-e-alvos-preferidos-de-golpistas/`. Acessado em 08 de outubro de 2024.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhol, New York.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Experian, S. (2024). Prevenção e detecção: como garantir a proteção contra fraudes. Disponível em: `https://www.serasaexperian.com.br/conteudos/prevencao-a-fraude/relatorio-de-fraude-2024-consumidores-estao-dispostos-a-pagar-mais-caro-por-seguranca/`. Acessado em 15 de janeiro de 2025.

FISHER, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Flondor, E., Donath, L., and Neamtu, M. (2024). Automatic card fraud detection based on decision tree algorithm. *Applied Artificial Intelligence*, 38(1):2385249.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. Ieee.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

IBM (2021). O que é overfitting? Disponível em: `https://www.ibm.com/br-pt/think/topics/overfitting`. Acessado em 20 de maio de 2025.

IBM (2022). What is a decision tree? Disponível em: `https://www.ibm.com/think/topics/decision-trees`. Acessado em 20 de maio de 2025.

IBM (2024). What is random forest? Disponível em: `https://www.ibm.com/think/topics/random-forest`. Acessado em 20 de maio de 2025.

Ileberi, E. and Sun, Y. (2024). Advancing model performance with adasyn and recurrent feature elimination and cross-validation in machine learning-assisted credit card fraud detection: A comparative analysis. *IEEE Access*.

Ileberi, E., Sun, Y., and Wang, Z. (2022). A machine learning based credit card fraud detection using the ga algorithm for feature selection. *Journal of Big Data*, 9(1):24.

Kaggle (2018). Credit card fraud detection. Disponível em: `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`. Acessado em 08 de outubro de 2024.

Kaggle (2021). Credit card transactions. Disponível em: `https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions(openinanewwindow)`.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., and Steinbrecher, M. (2022). Multilayer perceptrons. In *Computational intelligence: a methodological introduction*, pages 53–124. Springer.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press.

Marini, F. and Walczak, B. (2015). Particle swarm optimization (pso). a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149:153–165.

Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial neural networks and machine learning–ICANN 2011: 21st international conference on artificial neural networks, espoo, Finland, June 14-17, 2011, proceedings, part i 21*, pages 52–59. Springer.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

Mishra, S. (2017). Handling imbalanced data: Smote vs. random undersampling. *Int. Res. J. Eng. Technol*, 4(8):317–320.

Misra, P. and Yadav, A. S. (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *Int. J. Emerg. Technol*, 11(3):659–665.

Mniai, A., Tarik, M., and Jebari, K. (2023). A novel framework for credit card fraud detection. *IEEE Access*.

Padhi, I. (2021). Data for tabformer — fornecido pelo box. Disponível em: `https://ibm.ent.box.com/v/tabformer-data/folder/130747715605`. Acessado em 12 de maio de 2025.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1:81–106.

Rekha, G. and Tyagi, A. K. (2021). Cluster-based under-sampling using farthest neighbour technique for imbalanced datasets. In *Innovations in Bio-Inspired Computing and Applications: Proceedings of the 10th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2019) held in Gunupur, Odisha, India during December 16-18, 2019 10*, pages 35–44. Springer.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?"explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rinku, Dubey, A. K., Narang, S. K., and Kishore, N. (2024). A machine learning based approach for the fraud detection in imbalanced credit card transaction dataset. *SSRG International Journal of Electronics and Communication Engineering*, 11(8):244 – 259.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. Seattle, USA.

Salekshahrezaee, Z., Leevy, J. L., and Khoshgoftaar, T. M. (2023). The effect of feature extraction and data sampling on credit card fraud detection. *Journal of Big Data*, 10(1):6.

Shenoy, K. (2020). Credit card transactions fraud detection dataset. *Kaggle. Available online: https://www. kaggle. com/datasets/kartik2112/fraud-detection (accessed on 29 July 2023)*.

Singh, V., Pencina, M., Einstein, A. J., Liang, J. X., Berman, D. S., and Slomka, P. (2021). Impact of train/test sample regimen on performance estimate stability of machine learning in cardiovascular imaging. *Scientific reports*, 11(1):14490.

Slomka, P. J., Betancur, J., Liang, J. X., Otaki, Y., Hu, L.-H., Sharir, T., Dorbala, S., Di Carli, M., Fish, M. B., Ruddy, T. D., et al. (2020). Rationale and design of the registry of fast myocardial perfusion imaging with next generation spect (refine spect). *Journal of Nuclear Cardiology*, 27(3):1010–1021.

Sreekala, K., Sridivya, R., Rao, N. K. K., Mandal, R. K., Moses, G. J., and Lakshmanarao, A. (2024). A hybrid kmeans and ml classification approach for credit card fraud detection. In *2024 3rd International Conference for Innovation in Technology (INOCON)*, pages 1–5.

Stone, M. (1978). Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics*, 9(1):127–139.

Tax, D. M. and Duin, R. P. (2004). Support vector data description. *Machine learning*, 54:45–66.

Tomek, I. (1976). Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772.

Unico (2023). Prevenção e detecção: como garantir a proteção contra fraudes. Disponível em: `https://unico.io/unicocheck/protecao-contra-fraudes/`. Acessado em 15 de janeiro de 2025.

Varma, S. and Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7:1–8.

Vihurskyi, B. (2024). Credit card fraud detection with xai: Improving interpretability and trust. In *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, pages 1–6.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.