UNIVERSIDADE ESTADUAL PAULISTA - UNESP

Faculdade de Engenharia Mecânica de Ilha Solteira

LUCAS COSTA FERREIRA ASSUMPÇÃO

Detecção de Fraudes Bancárias:

Aplicação de Machine Learning e Engenharia de Características em uma Base Simulada de Transações com Cartão de Crédito

Ilha Solteira

2025

LUCAS COSTA FERREIRA ASSUMPÇÃO

Detecção de Fraudes Bancárias:

Aplicação de Machine Learning e Engenharia de Características em uma Base Simulada de Transações com Cartão de Crédito

Trabalho de Conclusão de Curso apresentada à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia de Ilha Solteira (FEIS), para obtenção do título de Grau acadêmico Bacharel em Engenharia Mecânica

Orientador: Prof. Dr. Aparecido Carlos Gonçalves

Ilha Solteira

2025

e Transações com Cartão de Crédito

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Assumpção, Lucas Costa Ferreira.
A851d Detecção de fraudes bancárias: aplicação de Machine Learning e engenharia de características em uma base simulada de transações com cartão de crédito / Lucas Costa Ferreira Assumpção. -- Ilha Solteira: [s.n.], 2025
58 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Mecânica) - Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Ilha Solteira, 2025

Orientador: Aparecido Carlos Gonçalves

Inclui bibliografia

1. Detecção de fraudes. 2. Aprendizado de máquina. 3. Engenharia de características. 4. Modelagem estatística. 5. Análise temporal.

Sandra Montibeller - CRB-8/060

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Ilha Solteira

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA – CÂMPUS DE ILHA SOLTEIRA

CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA
ATA DA DEFESA – TRABALHO DE GRADUAÇÃO

TÍTULO: Aplicação de Machine Learning e Engenharia de Características em uma Base Simulada de Transações com Cartão de Crédito

ALUNO: Lucas Costa Ferreira Assumpção RA: 172053889

Orientador: Prof. Aparecido Carlos Gonçalves

Aprovado (X) - Reprovado ( ) pela Comissão Examinadora
Nota obtida: dez (10,0)

Comissão Examinadora:

gov.br Documento assinado digitalmente
APARECIDO CARLOS GONCALVES
Data: 19/05/2025 14:28:47-0300
Verifique em https://validar.iti.gov.br

Prof. Aparecido Carlos Gonçalves
Presidente (Orientador)

gov.br Documento assinado digitalmente
ANDERSON INACIO JUNQUEIRA JUNIOR
Data: 19/05/2025 10:34:31-0300
Verifique em https://validar.iti.gov.br

M. Sc.: Anderson Inácio Junqueira Júnior

gov.br Documento assinado digitalmente
ALEJANDRO JOSUE VARGAS FIGUEROA
Data: 19/05/2025 10:45:10-0300
Verifique em https://validar.iti.gov.br

Engenheiro: Alejandro Josué Vargas Figueroa

gov.br Documento assinado digitalmente
LUCAS COSTA FERREIRA ASSUMPCAO
Data: 19/05/2025 12:09:11-0300
Verifique em https://validar.iti.gov.br

Assinatura do Aluno

Ilha Solteira (SP) 19 de maio de 2025

Dedico este trabalho aos meus pais, minhas irmãs, minhas tias, minha namorada e meus amigos, pelo apoio, incentivo e presença ao longo desta jornada.

AGRADECIMENTOS

A jornada até a conclusão deste trabalho foi desafiadora, mas repleta de aprendizados e conquistas. Nenhum caminho é trilhado sozinho, e por isso, expresso aqui minha mais sincera gratidão àqueles que me apoiaram ao longo dessa trajetória.

Agradeço, em primeiro lugar, aos meus pais, pelo amor incondicional, pelo incentivo constante e por acreditarem em mim em cada passo do caminho. Suas palavras de apoio e sua dedicação foram fundamentais para que eu chegasse até aqui.

Às minhas irmãs, pelo companheirismo, pelas conversas e pelo suporte em todos os momentos. Suas palavras de encorajamento e confiança sempre me deram forças para continuar.

Às minhas tias, que sempre estiveram presentes, demonstrando carinho e apoio inestimáveis. Seus gestos de cuidado e motivação foram essenciais ao longo desse processo.

À minha namorada, pelo incentivo diário, pela paciência e pela compreensão nos momentos de dedicação intensa. Seu apoio incondicional tornou esse percurso mais leve e significativo.

Aos meus amigos, por cada conversa, pelas trocas de conhecimento e pelos momentos de descontração que ajudaram a aliviar a pressão ao longo dessa jornada. A amizade e o apoio de vocês foram indispensáveis.

A todos que, direta ou indiretamente, contribuíram para minha formação e para a realização deste trabalho, meu mais sincero agradecimento.

"A informação é a resolução da incerteza."
Claude Shannon

RESUMO

A detecção de fraudes financeiras tornou-se um dos desafios mais críticos para instituições bancárias e empresas do setor de pagamentos, devido ao crescente volume de transações digitais e à sofisticação dos ataques fraudulentos. Este trabalho foi inspirado no estudo de Le Borgne et al. (Borgne *et al.*, 2022) e apresenta o desenvolvimento de um sistema de detecção de fraudes utilizando técnicas de aprendizado de máquina, com ênfase na transformação de características e análise temporal.

Inicialmente, foi realizada uma **simulação de dados de transações financeiras**, utilizando distribuições estatísticas como normal, Poisson e uniforme para modelar variáveis como valores das transações, frequência de operações e localização de terminais. A partir dessa simulação, foi conduzido um processo de **geração e enriquecimento de dados**, no qual variáveis adicionais foram criadas para capturar padrões comportamentais e características de risco associadas a clientes e terminais.

Os modelos de aprendizado de máquina aplicados incluíram **Regressão Logística, Árvores de Decisão, Random Forest e XGBoost**, os quais foram treinados e avaliados com base em métricas como **AUC-ROC, Average Precision e Precisão do Cartão@100**. Os resultados demonstraram que o **Random Forest** obteve o melhor desempenho no conjunto de teste, atingindo um equilíbrio entre acurácia e eficiência computacional. No entanto, o tempo de treinamento do modelo foi significativamente maior em comparação a técnicas mais simples, como a **Regressão Logística**.

Os experimentos evidenciaram a importância de uma **engenharia de dados robusta**, destacando que a inclusão de variáveis temporais e agregações comportamentais melhora a performance dos modelos. Além disso, observou-se que modelos mais complexos, como **Redes Neurais**, podem ser explorados em trabalhos futuros para aprimorar a detecção de fraudes. Conclui-se que a combinação de aprendizado de máquina com uma estrutura de dados bem definida pode fornecer um sistema eficaz para a mitigação de fraudes no setor financeiro.

Palavras-chave: Detecção de fraudes. Aprendizado de máquina. Engenharia de características. Modelagem estatística. Análise temporal.

ABSTRACT

The detection of financial fraud has become one of the most critical challenges for banking institutions and payment service providers due to the increasing volume of digital transactions and the growing sophistication of fraudulent schemes. Inspired by the work of Le Borgne et al. (Borgne *et al.*, 2022), this study presents the development of a fraud detection system using machine learning techniques, with a focus on feature engineering and temporal analysis.

Initially, a **simulation of financial transaction data** was conducted, utilizing statistical distributions such as normal, Poisson, and uniform to model transaction values, transaction frequency, and terminal locations. Based on this simulation, a **data generation and enrichment process** was carried out, creating additional features to capture behavioral patterns and risk characteristics associated with clients and terminals.

The applied machine learning models included **Logistic Regression, Decision Trees, Random Forest, and XGBoost**, which were trained and evaluated using metrics such as **AUC-ROC, Average Precision, and Card Precision@100**. The results showed that **Random Forest** achieved the best overall performance on the test set, balancing accuracy and computational efficiency. However, its training time was significantly longer compared to simpler models, such as **Logistic Regression**.

The experiments highlighted the importance of a **robust feature engineering process**, demonstrating that incorporating temporal variables and behavioral aggregations improves model performance. Additionally, more complex models such as **Deep Learning and Gradient Boosting** could be explored in future research to enhance fraud detection. It is concluded that the combination of machine learning with a well-structured data processing pipeline can provide an effective system for mitigating fraud in the financial sector.

Keywords: Fraud detection. Machine learning. Feature engineering. Statistical modeling. Temporal analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O desenvolvimento das economias globais tem sido impulsionado pelo contínuo aumento da disponibilidade e aceitação de crédito (Silva, 2011). Com a facilitação do acesso ao crédito pela população, o cartão de crédito se tornou um dos principais instrumentos financeiros de pagamento. Este instrumento permite ao titular efetuar o pagamento de bens e serviços ao comerciante, com a premissa de que o titular honrará essa transação futuramente para o banco emissor (Arthur; Sheffrin, 2003).

De acordo com estudos conduzidos pela Associação Brasileira das Empresas de Cartão de Crédito e Serviços (ABECS), em 2022, foram realizadas 18,2 bilhões de transações por meio de cartões de crédito, totalizando um volume de R$2,1 trilhões de reais. Esses números evidenciam a significativa presença do cartão de crédito na cultura brasileira.

A adesão nacional e internacional dos cartões de crédito trouxe consigo uma série de atividades fraudulentas, resultando em perdas financeiras expressivas por todo o globo. Segundo o relatório do Banco Central Europeu (BCE), de um total de 4,84 trilhões de euros transacionados em 2018, 1,80 bilhão foi associado a atividades fraudulentas (Banco Central Europeu, 2018), destacando a importância de combater essas práticas.

É evidente, dada a magnitude do problema, que as atividades fraudulentas impactam não apenas as instituições financeiras, mas também a sociedade em geral, gerando efeitos negativos que reverberam por todo o sistema financeiro, afetando investimentos, a confiança do consumidor e a atividade econômica como um todo. Com o avanço tecnológico, as práticas fraudulentas tornaram-se cada vez mais sofisticadas; no entanto, as práticas de combate também evoluíram. Isso inclui abordagens educacionais e conscientizadoras promovidas por órgãos privados e públicos, bem como o uso de tecnologias antifraude, como sistemas de inteligência artificial e aprendizado de máquina, que visam reconhecer padrões de transações em tempo real.

Este trabalho, buscou apresentar o desenvolvimento de um sistema para a detecção de fraudes em transações financeiras utilizando técnicas de aprendizado de máquina, com ênfase na transformação de características, análise temporal e avaliação de modelos preditivos. Inspirado por estudos conduzidos pelo grupo Machine Learning Group (MLG) da Université Libre de Bruxelles e pela Worldline (Borgne *et al.*, 2022), integraram-se princípios teóricos e práticos, culminando em uma solução robusta e adaptável para um problema amplamente presente no setor financeiro.

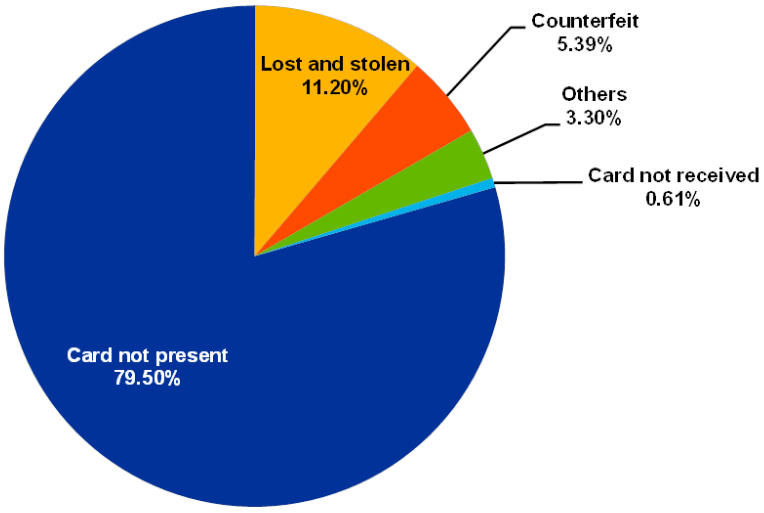
2 FUNDAMENTAÇÃO TEÓRICA

As fraudes envolvendo cartões de pagamento são, em geral, classificadas em duas modalidades principais: transações com cartão não presente e transações com cartão presente. De acordo com o sexto relatório do Banco Central Europeu (ECB), em 2018, 79% do valor total de fraudes na Área Única de Pagamentos em Euros (SEPA) ocorreram na modalidade de cartão não presente, enquanto 21% foram observadas em transações com cartão presente, sendo 15% registradas em terminais de ponto de venda e 6% em caixas eletrônicos (Banco Central Europeu, 2018).

2.1 Fraudes com cartão não presente

Nessa modalidade, a fraude acontece de forma remota, utilizando-se os dados do cartão por meio de correio, telefone ou internet. Na maioria dos casos, as informações são obtidas a partir da violação de dados pessoais do titular, por meio de técnicas como phishing, que envolve o envio de e-mails, mensagens de texto ou ligações telefônicas fraudulentas, muitas vezes simulando contatos de instituições financeiras. Esses comunicados normalmente contêm links maliciosos que levam a vítima a inserir suas credenciais, expondo informações sensíveis do cartão. Outra estratégia recorrente para a coleta de dados é a disseminação de vírus do tipo malware, cavalos de Troia ou keyloggers, que infectam computadores e capturam dados bancários e outras credenciais.

Figura 1 – Valor dos tipos de fraude como percentagem do total de fraudes com cartões utilizando cartões emitidos no âmbito da SEPA



Fonte: (Banco Central Europeu, 2018)

O Gráfico 1 mostra que a maior parte das fraudes corresponde à categoria de cartão

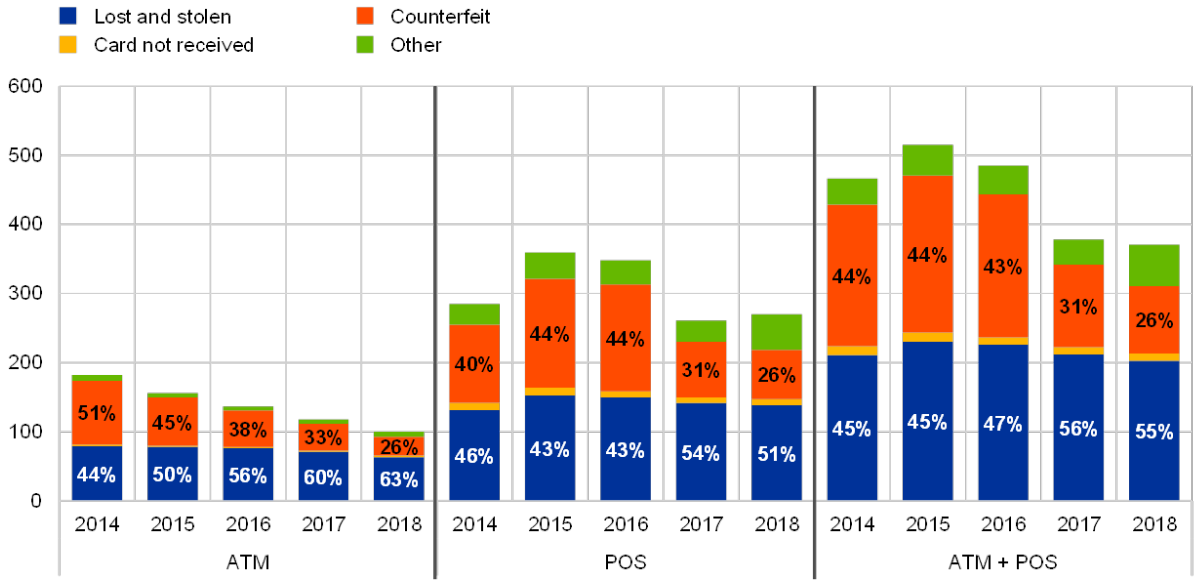
não presente, somando 79% do total de ocorrências na Área Única de Pagamentos em Euros. Em seguida, estão as fraudes envolvendo cartões perdidos ou roubados, com 11,20%, enquanto os cartões clonados correspondem a 5,38% do valor total de fraudes (Banco Central Europeu, 2018).

2.2 Fraudes com cartão físico

Esse tipo de fraude ocorre quando a transação é realizada presencialmente, seja em um caixa eletrônico ou em um terminal de ponto de venda. As práticas criminosas variam em grau de complexidade. Em alguns casos, o cartão é simplesmente roubado ou perdido, caso não se efetue o bloqueio imediato, o fraudador pode efetuar várias compras em curto intervalo de tempo, antes que a vítima perceba a irregularidade. Também existe a possibilidade de o cartão ser interceptado durante o envio ao titular, podendo até mesmo ser encomendado diretamente pelo criminoso, caso ele tenha acesso à conta bancária da vítima.

Outra forma recorrente de obtenção de dados é por meio do skimming, que se vale de dispositivos eletrônicos para capturar informações magnéticas do cartão, incluindo número e data de validade, viabilizando a clonagem do cartão original. Esse processo dificulta a identificação da origem da fraude, pois a vítima muitas vezes não faz ideia de que seu cartão foi copiado, e o criminoso pode aguardar um longo período antes de realizar a primeira transação fraudulenta.

Figura 2 – Evolução e distribuição do valor da fraude com cartão presente por categoria



Fonte: (Banco Central Europeu, 2018)

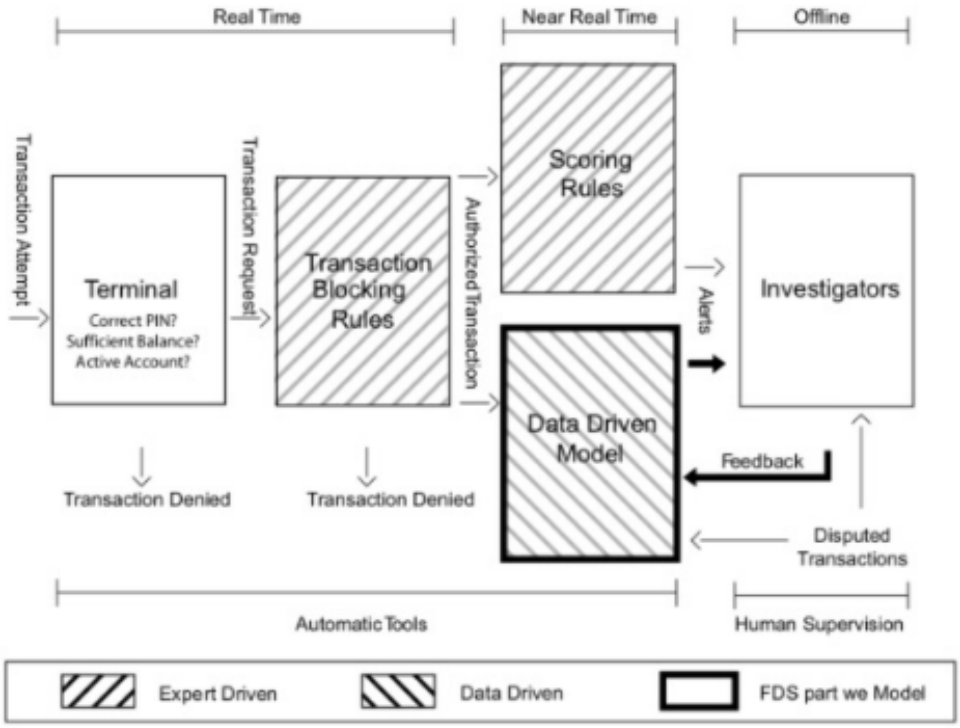
Na figura 2, são apresentadas as principais fraudes que utilizam o cartão físico, abrangendo as categorias de cartões roubados e clonados. Já os cartões que não chegam a ser recebidos pelo titular representam uma fração menor, tanto em transações realizadas

em caixas eletrônicos quanto em terminais de ponto de venda. Nesse tipo de crime, os titulares podem permanecer desinformados sobre a existência de um clone, permitindo que o criminoso aguarde um período prolongado antes de realizar a primeira transação fraudulenta.

2.3 Sistema de detecção de fraude em cartão de crédito

Um sistema de detecção de fraude em cartão de crédito é composto por cinco camadas principais: terminal, regras de bloqueio de transações, regras de pontuação, modelo orientado a dados e, por fim, investigadores.

Figura 3 – Diagrama das camadas de controle em um Sistema de Detecção de Fraudes



Fonte:(Borgne *et al.*, 2022)

2.3.0.1 Terminal

O terminal corresponde à primeira camada de controle, realizando verificações convencionais de segurança em toda solicitação de pagamento, como quantidade de tentativas, uso do código PIN, verificação de saldo disponível e limite do cartão. Esses procedimentos são consultados em milissegundos em um servidor da empresa emissora do cartão. Caso a transação não atenda a alguma dessas validações, ela é bloqueada; do contrário, prossegue para as próximas camadas.

2.3.0.2 Regras de Bloqueio de Transações e Pontuação

As regras de bloqueio de transações compõem a segunda camada de controle e envolvem um conjunto de instruções condicionais baseadas nas informações pontuais de cada operação, sem considerar históricos ou perfis dos titulares. Definidas por equipes especializadas, essas regras requerem precisão e rapidez na execução (SSC), 2019). Caso a transação não atenda a alguma dessas condições, o processo é interrompido; do contrário, avança para as etapas seguintes.

Com o enriquecimento de dados — por meio de características agregadas que contextualizam a transação, como gasto médio, localização das últimas compras, número médio de transações no mesmo dia em relação às anteriores e o histórico do titular —, passam a existir informações adicionais que não eram consideradas pelas camadas primárias. No campo da ciência de dados, o processamento desses atributos é denominado "feature engineering" (Zheng, 2018), e, neste contexto, são incorporados aos dados da transação para fortalecer a identificação de fraudes.

Semelhante às regras de bloqueio, as regras de pontuação são validações condicionais que atribuem pontuações às características agregadas de cada transação autorizada. Essas pontuações são definidas manualmente por equipes especializadas, as quais estabelecem de forma arbitrária os valores associados a cada condição (Pozzolo, 2015). Contudo, para que essas estruturas condicionais sejam efetivas, é necessário conhecer as estratégias de fraude em vigor. Além disso, a atribuição de pontuações pode variar de um investigador para outro, o que pode tornar essa camada relativamente incompleta e com menor capacidade de adaptação no longo prazo.

2.3.0.3 Modelo Orientado a Dados

Nesta camada, que constitui o foco principal deste estudo, são adotados modelos estatísticos para estimar a probabilidade de cada atributo da transação ser fraudulento ou não. Dessa forma, a pontuação de fraude passa a ser definida com base na probabilidade calculada pelo modelo, que, por sua vez, é treinado a partir de um amplo conjunto de transações rotuladas como fraude ou não fraude. Essa abordagem caracteriza o uso de aprendizado de máquina, que reduz a necessidade de interação humana direta. Desse modo, espera-se que um modelo orientado a dados seja capaz de detectar padrões de fraude em grande escala, muitas vezes invisíveis para equipes especializadas e que não se limitam a regras interpretáveis (Pozzolo, 2015).

2.3.0.4 Investigadores

A camada final é composta pelo time de especialistas que elabora as regras de bloqueio e define as pontuações, além de monitorar os alertas gerados pelas camadas anteriores e pelo modelo orientado a dados. Na prática, esses profissionais validam as infor-

mações vinculadas às 6 transações suspeitas, incluindo as pontuações e as probabilidades atribuídas. Ao confirmar a natureza de cada transação, eles fornecem feedback às camadas responsáveis, aprimorando as regras e ajustando continuamente o modelo estatístico.

O desafio de identificar fraudes em meio a milhões de transações processadas diariamente é imenso do ponto de vista humano, pois a vasta quantidade de dados dificulta a análise manual de cada operação. Consequentemente, o uso de algoritmos de aprendizado de máquina, capazes de processar grandes volumes de informação, tornou-se amplamente difundido na área de detecção de fraudes (Banco Central Europeu, 2018).

2.4 Modelagem Estatística e Distribuições Probabilísticas

A geração dos dados de transações financeiras neste estudo foi baseada em modelos estatísticos que permitem simular de forma realista o comportamento dos clientes e das operações realizadas. Dentre as principais distribuições probabilísticas utilizadas, destacam-se a distribuição normal, a distribuição de Poisson e a distribuição uniforme, cada uma aplicada conforme as características dos fenômenos modelados.

2.4.1 Distribuição Normal

A *distribuição normal* é amplamente utilizada em modelagem estatística devido à sua propriedade de descrever fenômenos naturais em que há uma concentração de valores em torno da média, com uma dispersão gradual para os extremos (Montgomery; Runger, 2019). No contexto da simulação das transações financeiras, a distribuição normal foi aplicada para:

- Gerar os horários das transações ao longo do dia, considerando que a maioria das operações ocorre durante o período comercial. O tempo das transações foi modelado com uma média centralizada em 43.200 segundos (meio do dia) e um desvio padrão de 20.000 segundos, permitindo uma variação natural dos horários (Ross, 2017).

- Determinar os valores das transações, utilizando como média o valor médio de transação do cliente e um desvio padrão proporcional a esse valor médio. Esse modelo reflete a realidade de sistemas de pagamento, nos quais transações de baixo valor são predominantes, enquanto operações de alto valor ocorrem com menor frequência (James *et al.*, 2013).

Figura 4 – Gráfico ilustrativo da distribuição Normal

Fonte: Zibetti (2024)

2.4.2 Distribuição de Poisson

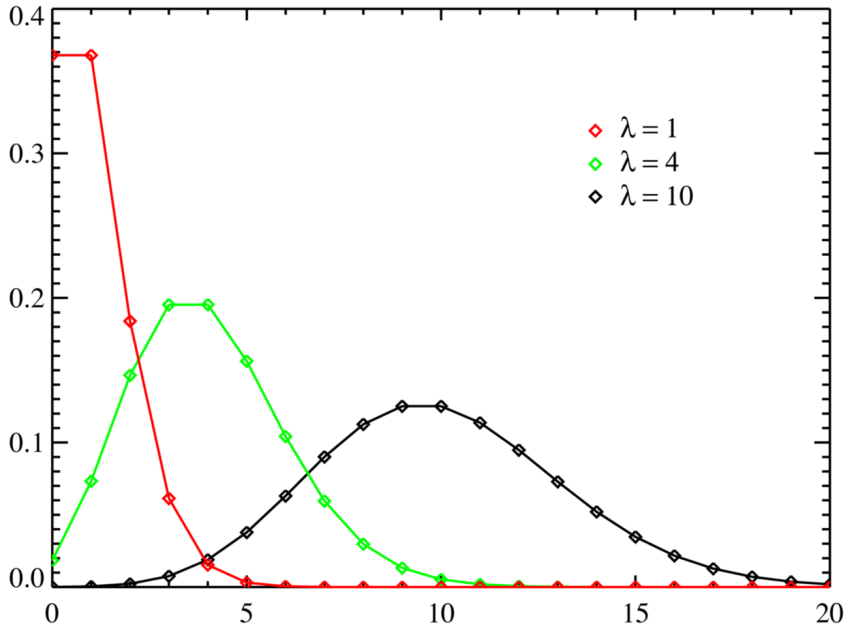
A *distribuição de Poisson* é apropriada para modelar eventos que ocorrem em intervalos de tempo fixos, mas de maneira aleatória (Casella; Berger, 2021). No presente estudo, essa distribuição foi empregada para determinar a quantidade de transações diárias realizadas por cada cliente, considerando que:

- O número de transações por cliente segue uma média pré-determinada, mas pode variar diariamente dentro de uma faixa de valores aceitável.

- Cada transação é tratada como um evento independente, mantendo a aleatoriedade natural observada no comportamento dos consumidores (DeGroot; Schervish, 2012).

A escolha da distribuição de Poisson garante que os dados gerados sejam realistas e coerentes com padrões financeiros observados em transações reais.

Figura 5 – Gráfico ilustrativo da distribuição de Poisson



Fonte: (Wikipedia, 2025)

2.4.3 Distribuição Uniforme

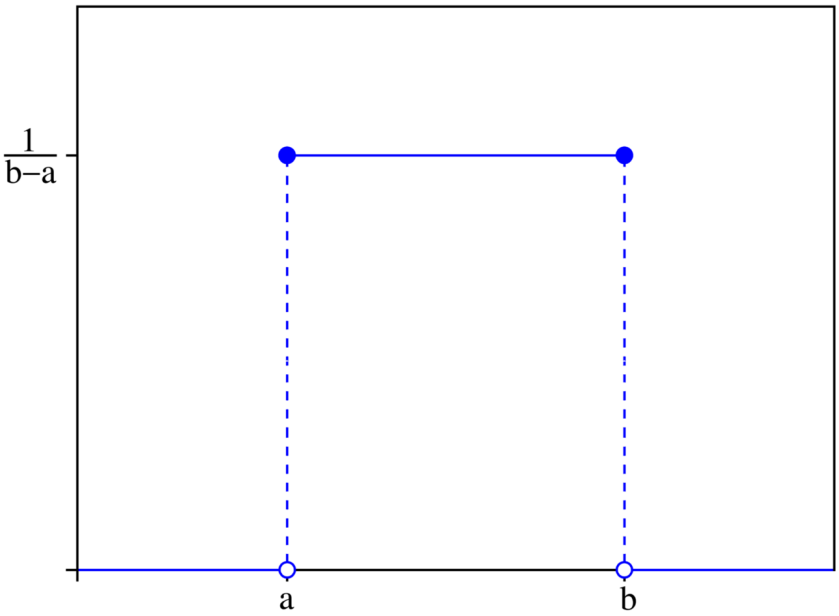
A *distribuição uniforme* é caracterizada pela igualdade de probabilidade entre todos os valores dentro de um intervalo especificado. Essa propriedade a torna adequada para situações em que se deseja evitar viés na geração dos dados (Mood; Graybill; Boes, 1974). No estudo em questão, essa distribuição foi aplicada para:

- Gerar as coordenadas geográficas dos clientes e dos terminais de pagamento, assegurando que estejam distribuídos homogeneamente dentro do espaço simulado.

- Determinar os valores médios das transações e a média de transações diárias por cliente, garantindo uma distribuição equilibrada dos atributos financeiros (Devore, 2011).

A utilização da distribuição uniforme foi fundamental para assegurar que o conjunto de dados fosse estatisticamente balanceado, evitando vieses que poderiam comprometer a avaliação dos modelos de detecção de fraudes.

Figura 6 – Gráfico ilustrativo da distribuição Uniforme



Fonte: (Wikipédia, 2024)

2.4.4 Impacto das Distribuições na Simulação de Dados

A escolha criteriosa das distribuições estatísticas permitiu a criação de um ambiente simulado que reflete padrões reais de transações financeiras. A distribuição normal garantiu a variabilidade natural dos valores e horários das operações, enquanto a distribuição de Poisson capturou a aleatoriedade do volume diário de transações. Por fim, a distribuição uniforme assegurou a homogeneidade espacial dos clientes e terminais, além de balancear os valores médios das transações. A integração dessas abordagens estatísticas na modelagem dos dados forneceu uma base robusta para o treinamento e teste dos modelos de aprendizado de máquina.

2.5 Aprendizado de máquina

O aprendizado de máquina, também conhecido pelo termo machine learning, difere dos métodos tradicionais de programação, pois permite ao sistema aprender com os dados e melhorar a partir da experiência. Esse ramo de estudo, advindo da ciência da computação e da inteligência artificial, relaciona-se estreitamente à estatística, ao reconhecimento de padrões e à mineração de dados, focando no desenvolvimento de algoritmos para a extração de conhecimento (Mitchell, 1997) .

Seu funcionamento envolve alimentar com dados o modelo durante o treinamento, enquanto o modelo é uma representação matemática que relaciona entradas e saídas desejadas. O algoritmo de aprendizado define como o modelo é treinado com base nesses dados, e a avaliação é essencial para mensurar o desempenho sobre novos conjuntos de

dados (Zheng, 2018).

Os sistemas de aprendizado de máquina podem ser categorizados conforme o nível e o tipo de supervisão recebidos durante o treinamento, como aprendizado supervisionado, não supervisionado, semissupervisionado ou por reforço (Goodfellow; Bengio; Courville, 2016). Cada categoria apresenta aplicações específicas, oferecendo diferentes maneiras de solucionar problemas complexos, a exemplo da detecção de fraudes em transações financeiras.

2.5.1 Aprendizado supervisionado

O aprendizado supervisionado consiste em treinar modelos de aprendizado de máquina a partir de um conjunto de dados rotulados com as categorias ou valores, de forma que cada instância apresente uma entrada (características) e uma saída (rótulo). Nesse contexto, o objetivo é fazer com que o modelo aprenda uma função ou regra de mapeamento entre a entrada e a saída, de modo a generalizar, o melhor possível, para um conjunto não visto durante o treinamento. O objetivo consiste em reduzir a diferença entre as previsões do modelo e os valores reais, utilizando medidas de erro como parâmetro de atualização. Dessa forma, o algoritmo ajusta seus parâmetros para minimizar erros de classificação ou de regressão, dependendo do tipo de tarefa.

A aplicação do aprendizado supervisionado é bastante ampla, abrangendo problemas de classificação (por exemplo, detecção de fraude em transações financeiras) e de regressão (como previsões de valores de imóveis). Em ambos os casos, a rotulação prévia dos dados permite que o modelo calcule métricas de desempenho, como acurácia, precisão ou erro quadrático médio, servindo como feedback para otimização do modelo (Mitchell, 1997). Além disso, o aprendizado supervisionado também possibilita estratégias de validação, em que o conjunto de dados é subdividido em treinamento e teste, visando avaliar a capacidade de generalização do modelo de forma consistente (Goodfellow; Bengio; Courville, 2016).

2.5.2 Aprendizado não supervisionado

Ao contrário do aprendizado supervisionado, o aprendizado não supervisionado não conta com rótulos ou valores de saída previamente definidos. Nesta abordagem, o algoritmo recebe apenas os dados, buscando encontrar padrões, agrupamentos ou estruturas subjacentes de forma autônoma. Geralmente, métodos de aprendizado não supervisionado são aplicados em tarefas de agrupamento (clustering) e de redução de dimensionalidade (principal component analysis – PCA), bem como na detecção de anomalias, quando se procura identificar pontos fora de um padrão comum.

Por não depender de rótulos, o aprendizado não supervisionado torna-se especialmente útil em cenários com grandes volumes de dados, cujas características são pouco conhecidas. Nesses casos, agrupar as instâncias de acordo com similaridades ou projetá-las

em subespaços de menor dimensão pode facilitar a análise exploratória. Na detecção de fraudes, o aprendizado não supervisionado serve, por exemplo, para encontrar transações com comportamento atípico, auxiliando na identificação de atividades suspeitas, mesmo que não se possua um histórico rotulado de fraudes. A efetividade desse método, contudo, depende do critério de similaridade adotado e do conhecimento prévio que se tem sobre os dados, pois, sem rótulos, pode ocorrer a descoberta de padrões sem relevância prática (Bishop, 2006).

2.5.3 Engenharia de Características

A engenharia de características é uma etapa fundamental para potencializar a eficácia dos métodos de aprendizado de máquina, sejam eles supervisionados ou não supervisionados, pois visa transformar dados brutos em atributos mais informativos e relevantes para a tarefa de detecção de fraudes. Em cenários bancários, como na análise de transações com cartões, as variáveis originais podem ser limitadas ou pouco esclarecedoras, o que exige um processo de enriquecimento para que padrões suspeitos sejam devidamente evidenciados.

No contexto de fraudes, esse enriquecimento pode envolver a criação de atributos temporais (como o dia da semana ou o horário de uma transação), a identificação de *outliers* ou a elaboração de indicadores financeiros capazes de refletir o perfil de gasto de cada cliente. Por exemplo, é possível adicionar colunas que mostrem a frequência de transações em intervalos específicos, bem como a variação nos valores transacionados ao longo do tempo. Além disso, quando se lida com variáveis categóricas (tais como o tipo de estabelecimento ou a localização do terminal), métodos como *one-hot encoding* podem converter essas variáveis em formatos numéricos, facilitando o processamento pelos algoritmos de classificação (Mitchell, 1997).

A qualidade dessas transformações costuma influenciar mais o desempenho final do modelo do que a complexidade do próprio algoritmo, pois atributos bem formulados podem evidenciar correlações e padrões nos dados (Zheng, 2018). Além disso, em instituições financeiras, a criação de bons atributos também pode favorecer a interpretabilidade do modelo, o que é crucial para justificar aprovações ou bloqueios de transações a clientes e autoridades.

Mediante uma estratégia bem planejada de engenharia de características, os modelos podem detectar sinais sutis de anomalia, mesmo em situações em que as fraudes representam uma fração minúscula do total de transações. Isso reduz não só as perdas financeiras, mas também reforça a confiança de todas as partes envolvidas, ao evidenciar padrões atípicos em tempo hábil e de forma explicável.

2.5.4 Regressão Logística

A Regressão Logística é um modelo de aprendizado supervisionado frequentemente empregado em problemas de classificação binária, nos quais se busca prever a probabilidade de ocorrência de um determinado evento (por exemplo, fraude ou não fraude). Diferentemente das abordagens tradicionais de regressão linear, que produzem valores contínuos como saída, a regressão logística utiliza a função logística (também conhecida como função sigmoide) para converter a combinação linear das variáveis de entrada em um valor entre 0 e 1, interpretado como uma probabilidade (Pregibon, 1981) .

Matematicamente, para um conjunto de variáveis independentes $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, o modelo de Regressão Logística estima a probabilidade de um evento $y = 1$ (fraude, por exemplo) ocorrer da seguinte forma:

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) \tag{2.1}$$

onde $\mathbf{w}$ é o vetor de parâmetros (coeficientes) associados às variáveis de entrada, b é o intercepto e $\sigma(z)$ é a função sigmoide, dada por:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{2.2}$$

Esse formato assegura que o resultado fique limitado ao intervalo [0,1], permitindo a interpretação direta como probabilidade. Para determinar os valores de w e b, utiliza-se o método de máxima verossimilhança, que visa encontrar os parâmetros que maximizam a probabilidade de o modelo reproduzir os rótulos observados no conjunto de treinamento (Agresti, 2007).

A Regressão Logística pode ser interpretada como uma forma de linearidade no espaço das razões de chances (log-odds). Em outras palavras, o logaritmo do quociente entre a probabilidade de um evento ocorrer e de não ocorrer é modelado como uma função linear dos preditores:

$$\ln\left(\frac{P(y = 1 \mid \mathbf{x})}{1 - P(y = 1 \mid \mathbf{x})}\right) = \mathbf{w} \cdot \mathbf{x} + b \tag{2.3}$$

Algumas de suas vantagens deve-se a ampla aplicabilidade e implementação simples, exigindo menos recursos computacionais em comparação com modelos mais complexos, sendo eficaz quando o número de variáveis é moderado e as relações entre preditores e resposta são majoritariamente lineares. No entanto, sua performance pode ser limitada em cenários com relações não lineares entre variáveis ou quando há um grande número de características em relação às amostras, aumentando o risco de sobreajuste.

2.5.5 Árvores de Decisão

As árvores de decisão são modelos que operam por meio de uma estrutura hierárquica em formato de nós e ramos, em que cada nó interno realiza uma partição dos dados com base em um critério de seleção de atributos, conduzindo a caminhos que eventualmente levam a nós folha, os quais indicam a classe ou valor de saída (Quinlan, 1993). Esse formato confere ao modelo uma estrutura facilmente interpretável, que pode ser visualizada de maneira semelhante à aplicação de uma série de "perguntas" sequenciais sobre as variáveis de entrada.

Do ponto de vista matemático, as árvores de decisão buscam dividir o espaço das variáveis preditoras de maneira recursiva, de modo a maximizar a pureza dos nós folha. Em problemas de classificação, métricas como entropia e índice Gini são comumente utilizadas para determinar a melhor divisão nos nós (Safavian; Landgrebe, 1991). Cada divisão sucessiva particiona o conjunto de dados em subconjuntos cada vez mais homogêneos em relação à variável-alvo.

Uma das principais vantagens das árvores de decisão reside em sua interpretabilidade. A sequência de divisões pode ser lida e compreendida de modo relativamente simples, em contraste com outros modelos tidos como "caixa-preta", a exemplo de redes neurais profundas. Além disso, as árvores de decisão são capazes de lidar com variáveis tanto numéricas quanto categóricas, além de resistir a valores ausentes por meio de estratégias como surrogate splits ou imputação (SSC), 2019).

2.5.6 Random Forest

O método Random Forest consiste em treinar várias várias Árvores de Decisão de forma paralela a partir de diferentes amostras do conjunto de dados e de subconjuntos aleatórios de variáveis. Cada árvore funciona como um "especialista" que emite seu parecer, e a decisão final é tomada a partir da votação majoritária das respostas fornecidas por esses 'especialistas' (Zheng, 2018). Esse processo, conhecido também como bagging, reforça a capacidade de generalização do modelo, pois agrega perspectivas diferentes sobre o mesmo problema.

A diversidade dentro de uma Random Forest é gerada pela aplicação de técnicas de amostragem, como o bootstrap, que consiste em selecionar aleatoriamente exemplos do conjunto de treinamento (com reposição) para cada árvore, além de aleatorizar as variáveis usadas em cada divisão. Essa abordagem reduz a correlação entre as árvores, evitando o sobreajuste (overfitting). Como resultado, surgem modelos robustos, especialmente úteis em cenários com grande volume de dados ou características complexas (Dietterich, 2000).

Outro aspecto relevante está na maneira como cada árvore vota. Para problemas de classificação, cada árvore aponta uma classe e a predição final decorre do voto majoritário.

Já em tarefas de regressão, as saídas das árvores podem ser agregadas por média. Embora essa combinação de diversos modelos torne a Random Forest menos interpretável do que uma única Árvore de Decisão, a robustez e o bom desempenho costumam compensar essa desvantagem, sobretudo em aplicações que exigem alta precisão (Liaw; Wiener, 2002).

2.5.7 XGBoost

O algoritmo XGBoost (Extreme Gradient Boosting) é um algoritmo de aprendizado ensemble, uma técnica que combina múltiplos modelos simples, como por exemplo árvores de decisão e regressão logística, de forma sequencial a fim de reduzir erros do conjunto anterior e formar um modelo forte. Enquanto o Random Forest combina árvores em paralelo (método conhecido como bagging), o boosting foca em cada árvore subsequente para corrigir as falhas da árvore anterior [20]. No caso do XGBoost, esse processo é otimizado por uma série de melhorias, como paralelismo eficiente, reguladores que diminuem o risco de sobreajuste e estratégias de amostragem que visam equilibrar desempenho e custo computacional.

A característica mais marcante do XGBoost é a adoção de um procedimento de gradiente descendente para ajustar cada árvore de maneira que o erro residual seja reduzido ao longo das iterações. Essa forma de aprendizado contínuo confere ao modelo uma alta capacidade de generalização, principalmente em problemas de classificação e regressão com dados de alta dimensão (Friedman, 2001). Além disso, a implementação inclui recursos de shrinkage, em que uma taxa de aprendizado controla a contribuição de cada árvore, e parâmetros de regularização (como lambda e alpha) para penalizar valores extremos nos pesos das árvores (Chen; Guestrin, 2016).

2.5.8 Métricas de Avaliação de Desempenho: AUC ROC e Average Precision

Para avaliar o desempenho dos modelos de aprendizado de máquina aplicados, foram utilizadas as métricas AUC ROC (Area Under the ROC Curve) e Average Precision (AP). Essas métricas são especialmente relevantes em cenários como a detecção de fraudes, onde a quantidade de ocorrências positivas (fraudes) é significativamente menor do que a de negativas (transações legítimas). Diante desse desequilíbrio de classes, torna-se essencial adotar métricas que consigam refletir com precisão a capacidade do modelo de distinguir corretamente entre eventos fraudulentos e legítimos (Han; Kamber; Pei, 2012a).

2.5.8.1 AUC ROC

A sigla AUC ROC representa a *Área Sob a Curva ROC* (do inglês *Receiver Operating Characteristic*). A curva ROC é traçada a partir da variação de um limiar sobre a probabilidade prevista pelo modelo, gerando pontos que relacionam a taxa de verdadeiros positivos (*True Positive Rate*, TPR) com a taxa de falsos positivos (*False Positive Rate*,

FPR). A AUC, por sua vez, corresponde à área sob essa curva, fornecendo um valor entre 0 e 1. Quanto mais próximo de 1, maior o poder de discriminação do modelo (Fawcett, 2006).

A **taxa de verdadeiros positivos** (TPR) é definida como:

$$TPR = \frac{TP}{TP + FN} \tag{2.4}$$

onde TP representa os verdadeiros positivos e FN os falsos negativos.

Já a **taxa de falsos positivos** (FPR) é dada por:

$$FPR = \frac{FP}{FP + TN} \tag{2.5}$$

onde FP representa os falsos positivos e TN os verdadeiros negativos.

A AUC pode ser interpretada como a integral da curva ROC, representada por:

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \tag{2.6}$$

Em termos práticos, um modelo com AUC ROC = 0.5 se assemelha a uma escolha aleatória entre as classes positiva e negativa. Já valores próximos de 1 indicam alta capacidade de distinguir corretamente exemplos de fraude dos exemplos legítimos. No entanto, a AUC ROC pode ser menos informativa em bases de dados muito desbalanceadas, porque a FPR pode permanecer baixa mesmo quando o modelo classifica equivocadamente várias fraudes como transações legítimas, desde que o número de verdadeiros negativos seja muito grande (Hernández-Orallo; Flach; Ferri, 2012).

2.5.8.2 Average Precision (AP)

O cálculo do AP ocorre por meio da média ponderada dos valores de precisão medidos em diferentes níveis de revocação, definindo uma área sob a curva de precisão-revocação (*Precision-Recall Curve*). Essa métrica é calculada como:

$$AP = \sum_{i=1}^{n} (R_i - R_{i-1}) P_i \tag{2.7}$$

onde:

- P_i é a precisão no i-ésimo ponto da curva.
- R_i é a revocação correspondente ao i-ésimo ponto.
- A soma percorre todos os pontos da curva, considerando as variações na revocação $(R_i - R_{i-1})$.

Em problemas com classes desequilibradas, essa representação tende a ser mais esclarecedora do que a curva ROC, pois enfatiza a capacidade do modelo em identificar exemplos positivos sem incluir muitos alarmes falsos. Uma pontuação de AP mais elevada reflete melhor equilíbrio entre precisão e revocação em diversos limiares de classificação, apontando para um maior desempenho na detecção de fraudes (Saito; Rehmsmeier, 2015).

3 METODOLOGIA

3.1 Simulador de dados de transação

As transações financeiras a nível de cliente são extremamente sensíveis e contêm informações sigilosas, como dados pessoais, hábitos de consumo, padrões de gastos e detalhes financeiros. Dessa forma, garantir a confidencialidade dessas informações é essencial, tanto para a privacidade dos clientes quanto para a segurança da própria instituição financeira, que precisa proteger seus dados por questões de competitividade.

No entanto, a escassez de fontes de dados públicos sobre transações financeiras dificulta a realização de estudos aprofundados sobre o tema. Atualmente, o site Kaggle disponibiliza um conjunto de dados contendo transações realizadas com cartão de crédito por titulares europeus em setembro de 2013. Entretanto, essa base de dados apresenta apenas variáveis numéricas e não inclui informações detalhadas sobre os atributos originais das transações. Dessa forma, surge a necessidade de modelar e simular um conjunto de transações, permitindo a exploração e o estudo de maneira mais didática.

A base gerada busca representar, de forma aproximada, a dinâmica subjacente aos dados reais de transações com cartões de pagamento. Apesar de ser uma abordagem simplificada, essa escolha favorece a interpretação dos padrões que diferentes técnicas de detecção de fraudes podem identificar (Han; Kamber; Pei, 2012b). Mesmo com um design simplificado, o simulador de dados gera um conjunto desbalanceado entre transações legítimas e fraudulentas, um desafio que também se apresenta em cenários reais de monitoramento financeiro.

O foco do design são os recursos mais essenciais de uma transação. Uma transação com cartão de pagamento, em essência, consiste em qualquer quantia paga a uma ponta final por uma ponta inicial em um determinado momento.

3.2 Geração da Tabela de Clientes

Inicialmente foi gerada a tabela de clientes, por meio, será criada por meio de uma função denominada **gera_tabela_cliente**, que recebe como parâmetro de entrada um número n de clientes. A partir disso, realiza-se uma iteração que percorre do primeiro até o $(n-1)$-ésimo cliente, gerando um conjunto de características, tais como: identificação, coordenadas de localização, valores médios das transações, desvio padrão das transações e média do número de transações diárias.

3.2.1 Geração das Coordenadas de Localização

As coordenadas de localização, representadas pelas variáveis `x_cliente_id` e `y_cliente_id`, são geradas aleatoriamente com distribuição uniforme dentro de um intervalo arbitrário de 0 a 100. Esse intervalo pode ser ajustado conforme necessário. A escolha da distribuição uniforme é apropriada, pois não há uma faixa específica a ser favorecida. No contexto das coordenadas de localização, essa distribuição garante que os clientes sejam distribuídos uniformemente dentro do espaço definido, evitando viés em direção a uma região específica.

3.2.2 Geração dos Valores Médios das Transações

Os valores médios das transações, representados pela variável `valor_medio`, também são gerados aleatoriamente por meio de uma distribuição uniforme dentro do intervalo [5, 100]. Essa abordagem garante que nenhum valor médio seja mais provável do que outro dentro do intervalo especificado.

3.2.3 Cálculo do Desvio Padrão das Transações

O desvio padrão das transações, representado pela variável `std_valor`, é calculado como a metade do valor médio gerado. Esse valor é utilizado para introduzir variabilidade nos valores médios das transações. A escolha de definir o desvio padrão como metade do valor médio é arbitrária, mas visa manter uma relação coerente entre ambos, evitando que a variabilidade dos dados seja excessivamente grande em relação à média.

3.2.4 Média de Transações Diárias

A média de transações diárias por cliente, representada pela variável `media_nb_tx_por_dia`, é gerada aleatoriamente por meio de uma distribuição uniforme dentro do intervalo [0, 4]. Mais uma vez, a distribuição uniforme é utilizada para garantir que o conjunto de dados não apresente viés em relação à quantidade de transações.

3.3 Geração da Tabela de Terminais

Assim como a tabela de clientes, a tabela de terminais será gerada por meio de uma função denominada **gera_tabela_terminal**, que recebe como parâmetro de entrada um número n de terminais. A partir disso, realiza-se uma iteração que percorre do primeiro até o $(n-1)$-ésimo terminal, gerando apenas a identificação do terminal e suas coordenadas geográficas.

3.3.1 Geração das Coordenadas Geográficas

As coordenadas dos terminais são geradas da mesma forma que as coordenadas dos clientes, ou seja, por meio de uma distribuição uniforme no intervalo $[0, 100]$. Esse processo

garante que não haja desbalanceamento ou viés em uma região específica, assegurando uma distribuição homogênea dos terminais.

3.4 Gerador de Associação de Terminais aos Perfis de Clientes

A tabela que associa os terminais aos perfis de clientes, denominada **lista_terminais_disponiveis**, possui como parâmetros de entrada a tabela de clientes, as coordenadas dos terminais e o raio r desejado. A partir desses parâmetros, a função tem por objetivo retornar uma lista de terminais que estão contidos dentro do raio r em relação à localização de um cliente específico.

3.4.1 Cálculo da Distância Euclidiana

Inicialmente, a função coleta em um vetor as localizações dos clientes. Em seguida, calcula-se a diferença euclidiana entre as coordenadas dos clientes e as coordenadas dos terminais, determinando a distância entre os dois pontos em um espaço cartesiano para cada par cliente-terminal.

Sejam as coordenadas $(x_{\text{cliente}}, y_{\text{cliente}})$ do cliente e as coordenadas do terminal i, a distância euclidiana d_i entre o cliente e o terminal i é dada por:

$$d_i = \sqrt{(x_{\text{cliente}} - x_{\text{terminal}_i})^2 + (y_{\text{cliente}} - y_{\text{terminal}_i})^2}$$

São computadas as diferenças quadráticas das coordenadas ao longo dos eixos das abcissas e das ordenadas entre o cliente e cada terminal. Essas diferenças são somadas e, em seguida, calcula-se a raiz quadrada da soma, obtendo-se a distância euclidiana entre o cliente e cada terminal.

3.4.2 Filtragem dos Terminais Disponíveis

Após calcular as distâncias entre os pontos, são selecionados apenas os terminais cuja distância em relação aos clientes satisfaz a condição de ser menor que o raio r estabelecido como parâmetro de entrada. Dessa forma, é possível construir uma nova coluna na tabela de clientes contendo uma lista de terminais disponíveis para cada cliente.

3.5 Gerador de Transações

A geração das transações inicia-se com a criação de uma lista vazia, denominada **transacoes_clientes**, que armazenará as informações geradas ao longo do processo. Em seguida, a função percorre cada dia do período estipulado por **n_dias**, determinando, para cada um deles, o número de transações (**n_tx**) por meio de uma distribuição de Poisson, cuja média é definida pelo atributo **media_nb_tx_por_dia** da **tabela_cliente**.

A escolha da distribuição de Poisson é apropriada, pois assume que as transações ocorrem em torno de uma média diária, ao mesmo tempo em que permite variações naturais, capturando a aleatoriedade do comportamento dos clientes. Cada transação é tratada como um evento independente, refletindo o caráter probabilístico do número de operações diárias.

Para garantir que apenas valores positivos sejam considerados, impõe-se a condição de que **n_tx** seja maior que zero. Após essa verificação, a função itera sobre todas as transações do dia, gerando aleatoriamente, por meio de uma distribuição normal, o horário da transação, representado pela variável **tempo_tx**. Considerando que um dia possui 86.400 segundos, e que a maioria das transações ocorre no período diurno, os tempos das transações são gerados com distribuição normal centrada em $\frac{86400}{2}$, com um desvio padrão de 20.000 segundos. Esse desvio capta a variação natural dos horários de transação.

Para evitar a geração de valores inválidos, apenas os tempos de transação situados no intervalo de 0 a 86.400 segundos são considerados. Em seguida, o valor da transação, representado pela variável **valor**, é extraído de uma distribuição normal com média igual ao valor médio das transações do cliente (**tabela_cliente.valor_medio**) e desvio padrão igual a **tabela_cliente.std_valor**.

3.6 Tratamento de Valores Negativos e Associação com Terminais

Caso algum valor de transação gerado seja negativo, a função substitui esse valor por um novo número aleatório, extraído de uma distribuição uniforme no intervalo $[0, 2 \times \textbf{tabela\_cliente.valor\_medio}]$. Esse procedimento garante que todas as transações apresentem valores positivos. Para padronização, os valores das transações são arredondados para duas casas decimais.

A etapa final do processo consiste em associar as transações a terminais de pagamento disponíveis. Caso o cliente possua terminais vinculados em sua base de dados, a função executa os seguintes passos:

1. Seleção aleatória de um terminal, armazenado na variável **terminal_id**, a partir do conjunto **tabela_cliente.terminais_disponiveis**.

2. Armazenamento das transações geradas na lista **transacoes_clientes**, incluindo os seguintes atributos:

   - **tempo_tx + dia × 86400**: Tempo absoluto da transação, em segundos, obtido somando o tempo gerado ao número do dia multiplicado por 86.400, garantindo que cada transação possua um horário único.
   - **dia**: Número do dia em que a transação ocorre dentro do período definido por **n_dias**.

- **tabela_cliente.cliente_id**: Identificação do cliente.
- **terminal_id**: Identificação do terminal onde a transação foi realizada.
- **valor**: Valor da transação.

3.7 Geração da Tabela de Transações

Após o término da iteração, os dados acumulados são organizados em um objeto *pandas DataFrame*, cujas colunas são nomeadas da seguinte forma: **TX_TEMPO_SEGUNDOS, TX_TEMPO_DIAS, CLIENTE_ID, TERMINAL_ID, TX_VALOR**.

Caso o *DataFrame* não esteja vazio, a função converte a coluna **TX_TEMPO_SEGUNDOS** para o formato de data e hora, criando uma nova coluna denominada **TX_DATA**. Por fim, a função retorna o objeto *DataFrame*, com as informações geradas.

3.8 Geração do Conjunto de Dados de Transações

Com a implementação das funções responsáveis pela geração de dados necessários, torna-se possível escalonar a base de transações para um conjunto de dados maior. Para esse propósito, foi desenvolvida a função **gera_dataset**, cuja finalidade é criar as três tabelas principais mencionadas anteriormente: a tabela de clientes, a tabela de terminais e a tabela de transações.

3.8.1 Parâmetros de Entrada

A função **gera_dataset** recebe os seguintes parâmetros:

- **n_clientes**: Quantidade de clientes a serem gerados no conjunto de dados (padrão: 10.000).
- **n_terminais**: Número de terminais que serão gerados (padrão: 100.000).
- **n_dias**: Período, em dias, no qual as transações serão geradas (padrão: 90 dias).
- **data_inicio**: Data inicial para a geração das transações (padrão: "2024-01-01").
- **r**: Raio de proximidade que define a associação entre clientes e terminais, com valor padrão de 5.

3.8.2 Processo de Geração dos Dados

A função inicia-se com a criação da tabela de clientes, utilizando a função **gera_tabela_cliente**. Em seguida, a tabela de terminais é gerada a partir da função **gera_tabela_terminal**. Com ambas as tabelas construídas, aplica-se a lógica de proximidade

dos terminais, determinando quais terminais estão dentro do raio r especificado e, portanto, acessíveis a cada cliente.

Os terminais disponíveis para cada cliente são armazenados em uma lista dentro da tabela de clientes, e a quantidade total de terminais acessíveis também é registrada. Esse processo assegura que cada cliente possua um conjunto definido de terminais nos quais pode realizar transações.

3.8.3 Geração e Organização das Transações

A etapa subsequente corresponde à geração das transações para cada cliente, executada por meio da função **gera_tabela_transacoes**. O número de transações geradas é baseado na frequência de transações definida para cada cliente. As transações são então agregadas por cliente, levando em consideração o período de tempo especificado no parâmetro **n_dias**. O resultado desse processo é um *DataFrame* consolidado contendo todas as transações simuladas. Durante essa etapa, o tempo de execução do processamento é impresso, permitindo monitorar o desempenho da função.

Uma vez geradas, as transações são organizadas cronologicamente, ordenadas pela data da transação. Além disso, os índices do *DataFrame* são reinicializados para garantir uma sequência adequada. Uma nova coluna, denominada **TRANSACAO_ID**, é adicionada ao conjunto de dados, atribuindo um identificador único para cada transação.

3.9 Geração de Cenários Fraudulentos

Nesta etapa da simulação, o conjunto de dados foi enriquecido com padrões de comportamento fraudulentos por meio da função **add_fraudes**. Essa função incorpora três cenários distintos de fraude ao conjunto de dados, permitindo a análise e o aprimoramento de técnicas de detecção.

A função **add_fraudes** recebe como parâmetros as tabelas de clientes (**tabela_clientes**), terminais (**tabela_terminal**) e transações (**df_transacoes**). Inicialmente, todas as transações são consideradas genuínas. Para identificar as fraudes, são adicionadas duas novas colunas ao *DataFrame*:

- **TX_FRAUDE**: Indica se a transação é fraudulenta (1) ou legítima (0).
- **TX_FRAUDE_CENARIO**: Especifica o tipo de fraude associado à transação (0 para transações genuínas).

3.9.1 Cenários de Fraude

3.9.1.1 Cenário 1: Fraudes por Valor Elevado

No primeiro cenário, transações cujo valor exceda R$ 220 são automaticamente classificadas como fraudulentas. Embora essa abordagem seja simplificada e não represente integralmente a complexidade das fraudes no mundo real, ela serve como um padrão inicial para validar a eficácia dos métodos de detecção de anomalias.

Implementação:

- Seleção de Transações: Identificação de todas as transações com **TX_VALOR** superior a 220.
- Marcação: Atualização das colunas **TX_FRAUDE** e **TX_FRAUDE_CENARIO**, atribuindo o valor 1 para fraude e 1 para o cenário de fraude correspondente.

Esse cenário avalia a capacidade do modelo de detectar anomalias evidentes, uma vez que se trata de um critério explícito e de fácil identificação.

3.9.2 Cenário 2: Comprometimento de Terminais

O segundo cenário simula fraudes decorrentes do uso de terminais comprometidos. A cada dia da simulação, dois terminais são aleatoriamente selecionados e todas as transações realizadas nesses terminais pelos 28 dias subsequentes são marcadas como fraudulentas. Esse padrão reflete casos em que criminosos obtêm controle sobre terminais físicos, como em ataques de *phishing* ou clonagem de máquinas de pagamento.

Implementação:

- **Seleção Diária**: Escolha aleatória de dois terminais por dia durante o período da simulação.
- **Identificação de Transações**: Filtragem de todas as transações realizadas nesses terminais nos 28 dias seguintes.
- **Marcação**: Definição da coluna **TX_FRAUDE** como 1 e **TX_FRAUDE_CENARIO** como 2 para indicar fraude associada ao terminal comprometido.

Esse cenário possibilita a modelagem de técnicas voltadas à detecção de padrões de fraudes recorrentes em pontos de pagamento específicos.

3.9.3 Cenário 3: Comprometimento de Clientes

O terceiro cenário representa fraudes cometidas por meio de clientes cujas credenciais foram comprometidas. A cada dia da simulação, três clientes são escolhidos aleatoriamente,

e, durante os 14 dias seguintes, um terço de suas transações é marcado como fraudulento, além de ter seus valores multiplicados por cinco. Esse padrão simula fraudes em que o criminoso, ao obter acesso às credenciais do cliente, realiza transações de alto valor enquanto a vítima continua operando normalmente.

Implementação:

- **Seleção Diária**: Escolha aleatória de três clientes por dia.
- **Identificação de Transações**: Filtragem das transações realizadas por esses clientes nos próximos 14 dias.
- **Amostragem Parcial**: Seleção de um terço dessas transações para serem marcadas como fraudulentas.
- **Multiplicação de Valor**: Multiplicação do valor das transações fraudulentas por cinco, simulando um padrão anômalo típico de fraudes com cartões clonados.
- **Marcação**: Definição das colunas **TX_FRAUDE** como 1 e **TX_FRAUDE_CENARIO** como 3.

Esse cenário reflete um dos tipos mais comuns de fraude, onde criminosos utilizam os dados do cliente para realizar transações significativamente maiores do que o padrão usual.

3.10 Transformação de Características do Conjunto de Dados

Para melhorar a capacidade preditiva dos modelos de aprendizado de máquina na detecção de fraudes, realizou-se transformações nas características do conjunto de dados. Embora os dados originais forneçam informações básicas sobre as transações, eles não estão no formato ideal para a maioria dos algoritmos de *machine learning*. A transformação e o enriquecimento dessas características são essenciais para capturar padrões relevantes e, consequentemente, aumentar o desempenho dos modelos preditivos.

Neste contexto, foram implementadas três categorias de transformações de características, amplamente reconhecidas por sua eficácia na detecção de fraudes em transações financeiras:

- Codificação Temporal;
- Características de Comportamento de Gasto (modelo RFM);
- Transformação baseada em Pontuação de Risco de Terminais.

3.11 Codificação Temporal

O comportamento fraudulento pode variar de acordo com o momento em que as transações são realizadas. Com base nisso, duas transformações temporais foram aplicadas à variável de data e hora das transações (**TX_DATA**): identificação de transações durante finais de semana e em horários noturnos.

3.11.1 Dia Útil ou Final de Semana (TX_FIM_SEMANA)

Para capturar variações comportamentais associadas aos dias da semana, foi criada a função **fim_semana**, que transforma a variável **TX_DATA** em um atributo binário denominado **TX_FIM_SEMANA**. A função converte a data em um número entre 0 e 6, representando de segunda-feira a domingo, respectivamente. Caso a transação ocorra em um sábado ou domingo, o valor retornado é 1; caso contrário, é 0.

Essa transformação é relevante porque estudos indicam uma maior incidência de fraudes durante finais de semana e feriados, períodos em que o monitoramento de transações costuma ser mais relaxado.

3.11.2 Dia ou Noite (TX_DURANTE_NOITE)

De maneira similar, a função **valida_noite** foi criada para identificar transações realizadas em horários noturnos. A variável **TX_DATA** é convertida em um valor de 0 a 23, representando as horas do dia. Transações realizadas entre meia-noite e 6h da manhã são marcadas com 1 na nova coluna **TX_DURANTE_NOITE**; as demais recebem 0.

Fraudes tendem a ocorrer com mais frequência durante a madrugada, dado o menor volume de transações legítimas e, consequentemente, uma supervisão reduzida.

3.12 Características de Comportamento de Gasto (Modelo RFM)

A segunda categoria de transformação explora o comportamento de gasto dos clientes com base no modelo RFM (*Recência, Frequência e Valor Monetário*), amplamente utilizado em análises de comportamento de consumo. Nesta implementação, foram considerados dois componentes: frequência e valor monetário, calculados em janelas de 1, 7 e 30 dias, totalizando seis novos atributos.

A função **obter_caracteristicas_comportamento_gasto_cliente** foi desenvolvida para realizar essa transformação. Como entrada, a função recebe o *DataFrame* de transações de um cliente e as janelas de tempo predefinidas.

Inicialmente, as transações são ordenadas cronologicamente com base na coluna **TX_DATA**, garantindo que os cálculos sejam realizados em sequência temporal correta. Para cada janela de tempo, a função realiza dois cálculos principais:

- **Frequência (N_TX_JANELA)**: Contagem de transações realizadas no período especificado.
- **Valor Monetário Médio (VALOR_MED_TX_JANELA)**: Média dos valores das transações no período.

Os resultados dessas transformações são incorporados ao *DataFrame* original. Após a adição dos novos atributos, o índice é redefinido para a coluna **TRANSACAO_ID**, preservando a estrutura original do conjunto de dados.

3.13 Transformação de ID de Terminal com Pontuação de Risco

O terceiro conjunto de transformações foca na análise de risco associada aos terminais de pagamento. O objetivo é identificar padrões de uso suspeitos, estimando a exposição de cada terminal a transações fraudulentas ao longo do tempo.

A função **obter_pontuacao_risco_janela_temporal** foi criada para calcular a pontuação de risco com base no histórico de transações de cada terminal. Diferentemente das transformações aplicadas aos clientes, esta considera um **período de atraso** de 7 dias, simulando o tempo necessário para que uma transação seja identificada como fraudulenta após investigações.

O processo ocorre em duas etapas:

1. **Cálculo com Período de Atraso**: Para cada terminal, a função calcula o número total de transações e de fraudes nos 7 dias anteriores ao período analisado. Esses valores são armazenados nas variáveis **N_FRAUDE_DELAY** e **N_TX_DELAY**.
2. **Cálculo com Janelas de Tempo**: As mesmas métricas são calculadas para janelas de 1, 7 e 30 dias, somadas ao período de atraso. A pontuação de risco é obtida pela razão entre o número de transações fraudulentas e o total de transações em cada janela.

Caso não haja transações em determinada janela, a pontuação de risco é automaticamente definida como zero, evitando divisões por zero. Além disso, o número de transações em cada janela é registrado, resultando em novos atributos que refletem o risco e a atividade do terminal ao longo do tempo.

Os valores calculados são incorporados ao *DataFrame* original, com a criação de novas colunas que indicam tanto a quantidade de transações quanto a pontuação de risco para cada uma das janelas temporais analisadas. Por fim, o índice do *DataFrame* é redefinido para **TRANSACAO_ID**, e eventuais valores ausentes são substituídos por zero, garantindo a consistência e integridade do conjunto de dados.

3.14 Sistema de Detecção de Fraudes

O sistema de detecção de fraudes abordado neste trabalho está dividido em três fases principais: (i) a preparação dos dados, (ii) o treinamento do modelo preditivo e (iii) a avaliação de desempenho.

Na primeira etapa, definem-se dois conjuntos de dados: um conjunto de treinamento, formado por dados históricos, e um conjunto de teste, que contém dados recentes. O objetivo dessa segmentação é garantir que o modelo seja capaz de generalizar para novos cenários, sem se restringir unicamente ao contexto dos dados de treinamento.

A segunda etapa consiste no treinamento de um modelo de classificação capaz de diferenciar transações lícitas das fraudulentas. Para tanto, utiliza-se a biblioteca *scikit-learn* do Python, reconhecida por sua variedade de algoritmos de aprendizado de máquina e facilidade de uso, permitindo a criação rápida de soluções preditivas.

Por fim, a terceira etapa foca na avaliação do modelo, em que o conjunto de teste serve para mensurar a eficácia do algoritmo frente a dados que ele ainda não viu. Essa análise garante aplicabilidade prática e assegura que o sistema detecte fraudes em um cenário real de produção.

Esse processo sistemático assegura a construção de um sistema de detecção de fraudes consistente, voltado a grandes volumes de transações financeiras e apto a reconhecer padrões anômalos de maneira eficiente.

3.14.1 Conjunto de Treinamento e Teste

O treinamento do modelo de detecção de fraudes exige a criação de dois conjuntos de dados: o primeiro para a etapa de treinamento e o segundo para a etapa de teste. O conjunto de treinamento permite que o modelo aprenda a identificar transações fraudulentas, enquanto o de teste mede a eficiência do modelo em detectar novas ocorrências de fraude.

Em aplicações reais, recomenda-se que as transações no conjunto de teste ocorram em um período posterior ao conjunto de treinamento, reproduzindo o cenário em que o sistema será efetivamente utilizado. Neste estudo, as transações de 01/04/2024 a 31/05/2024 formam o conjunto de treinamento, enquanto as de 08/06/2024 a 14/06/2024 compõem o conjunto de teste. Embora apenas uma semana de dados seja suficiente para um protótipo inicial e para avaliar o desempenho do modelo, em fases futuras poderão ser usados intervalos maiores, a fim de investigar como volumes de dados mais extensos afetam a performance.

No mundo real, o rótulo (fraudulento ou legítimo) de uma transação costuma surgir apenas após a devida investigação ou reclamação do cliente, o que justifica a inclusão de um período de atraso (*feedback delay*). Aqui, adotou-se uma semana. Apesar de simplificado,

esse intervalo fornece uma aproximação plausível, visto que muitas fraudes são descobertas nesse prazo. Ainda assim, esse período pode ser menor, se a instituição financeira identificar rapidamente as fraudes, ou maior, quando algumas delas demoram semanas ou meses para virem à tona.

Para operacionalizar esse fluxo, foi criada a função **obter_conjunto_treino_teste**, que recebe:

- **df_transacoes**: conjunto completo das transações;
- **data_inicio_treinamento**: data inicial do período de treinamento;
- **delta_train, delta_delay, delta_test**: intervalos que definem o período de treinamento, o atraso (feedback) e o período de teste, respectivamente.

Primeiro, selecionam-se as transações cujo tempo (**TX_DATA**) está entre *data_inicio_treinamento* e *data_inicio_treinamento + delta_train*. Esse subconjunto serve de base para o aprendizado do modelo, pois contém registros históricos com seus respectivos rótulos de fraude ou não.

Em seguida, é construído o conjunto de teste, assegurando que eventuais fraudes identificadas até o fim do *delta_delay* sejam excluídas dessa etapa, simulando o ambiente em que o modelo não conhece antecipadamente as ocorrências de fraude. Para isso, cria-se um conjunto de clientes reconhecidamente fraudulentos, cujas transações são removidas do conjunto de teste, de modo que a avaliação seja feita somente sobre dados "desconhecidos". Ao final, ordena-se o conjunto de teste pelo **TRANSACAO_ID**, garantindo consistência e coerência ao retornar o par (*treino*, *teste*).

3.14.2 Treinamento de Modelos de Aprendizado de Máquina

A etapa seguinte é o treinamento dos modelos de aprendizado de máquina, com o objetivo de gerar uma função preditiva que relacione as características de entrada — variáveis descritivas das transações — à saída binária de interesse (**TX_FRAUDE**), a qual assume valor 1 para fraudes e 0 para transações legítimas. Nesse cenário, as variáveis de entrada podem incluir desde o valor de cada transação (**TX_VALOR**) até atributos mais avançados, derivados das etapas de enriquecimento, como a frequência de transações e as pontuações de risco associadas a cada terminal.

Para organizar esse processo, foi desenvolvida a função **obter_modelo_e_predicoes**, que se baseia em quatro componentes principais:

- **classificador**: Um modelo de aprendizado de máquina (por exemplo, árvore de decisão, regressão logística ou *random forest*);

- **df_treinamento** e **df_teste**: Conjuntos de dados que fornecem exemplos rotulados (treinamento) e amostras recentes (teste), permitindo medir a eficácia do modelo em dados não vistos;

- **caracteristicas_input** e **caracteristicas_output**: Conjunto de variáveis de entrada (como **TX_VALOR**, frequência de transações e pontuações de risco) e a saída binária **TX_FRAUDE**;

- **scala**: Parâmetro que determina se as variáveis de entrada devem ser normalizadas para manter a mesma escala, recorrendo a métodos como *StandardScaler* ou *MinMaxScaler*.

Inicialmente a função valida se o usuário optou por normalizar as variáveis de entrada pelo parâmetro **scala**,caso sim, as variáveis de entrada dos conjuntos de treinamento e teste são normalizadas, assegurando uniformidade nos valores numéricos e evitando que atributos em escalas muito diferentes enviesem o ajuste do modelo. Em seguida, chama-se o método *fit* do classificador, que adapta o modelo aos dados de treinamento, registrando o tempo de execução dessa operação para análise de desempenho.

Uma vez concluído o treinamento, o modelo é avaliado por meio da função *predict_proba*, a qual retorna as probabilidades de fraude para cada amostra do conjunto de teste. O tempo de inferência também é cronometrado, possibilitando a avaliação de sua viabilidade em cenários de produção. Além disso, são obtidas previsões de probabilidade para o próprio conjunto de treinamento, permitindo comparar o desempenho em dados vistos e não vistos.

Por fim, todas as informações relevantes — tais como o classificador ajustado, as previsões de probabilidade para treinamento e teste e os tempos de execução — são agregadas em um dicionário chamado **dicionario_modelo_e_predicoes**. Essa estrutura facilita a consulta e o uso dos resultados em análises posteriores, contemplando desde o cálculo de métricas de desempenho (AUC, *precision*, *recall*, etc.) até a comparação entre diferentes algoritmos de detecção de fraudes.

3.14.3 Avaliação de Desempenho

A análise de precisão na detecção de fraudes, sobretudo no monitoramento de cartões comprometidos, recorre às funções **precisao_cartao_top_k_dia**, **precisao_cartao_top_k** e **performance_avaliacao**. Elas permitem avaliar o desempenho do classificador de diferentes perspectivas, destacando a precisão no *top-k* clientes mais prováveis de fraude e métricas gerais como **AUC-ROC** e **Average Precision**.

3.14.3.0.1 Precisão Diária no Top-k Cartões.

A função **precisao_cartao_top_k_dia** calcula a precisão em cada dia, baseada no grupo de cartões com maior probabilidade de fraude (*top-k*). Agrupam-se as transações por cliente, tomando os valores máximos de *predict_proba* e os rótulos reais (**TX_FRAUDE**). Depois, ordenam-se esses clientes em ordem decrescente de probabilidade de fraude, selecionando o *top-k*. A precisão decorre da fração de cartões efetivamente comprometidos dentre esses mais suspeitos.

3.14.3.0.2 Precisão Diária ao Longo do Período (Top-k).

A função **precisao_cartao_top_k** estende esse raciocínio a todo o intervalo de teste, percorrendo cada dia em ordem cronológica e removendo os cartões já identificados como fraudulentos. Em seguida, **precisao_cartao_top_k_dia** obtém a precisão específica de cada dia, gerando uma lista cujos valores são posteriormente combinados para se obter a precisão média do *top-k* ao longo de todo o período.

3.14.3.0.3 Desempenho Global.

Por fim, a função **performance_avaliacao** consolida o desempenho do modelo, calculando métricas como **AUC-ROC** e **Average Precision**, além da já mencionada precisão no *top-k* cartões. Esses resultados são reunidos em um *DataFrame* que pode ser arredondado, facilitando a interpretação. A flexibilidade dessas funções permite ainda a inclusão de novas métricas ou ajustes nos limiares, adaptando a solução de detecção de fraudes a vários cenários do setor financeiro.

4 RESULTADOS E DISCUSSÃO

O sistema criado para detectar fraudes em transações financeiras com cartões de crédito foi desenvolvido e avaliado utilizando dados simulados, que, embora provenientes de um ambiente controlado, reproduzem características de situações reais. Os resultados apresentados neste capítulo derivam da aplicação de técnicas de aprendizado de máquina em dados enriquecidos por transformações de atributos e pela inserção de cenários de fraude em um conjunto de transações simulado, previamente descrito na seção 3 da metodologia. As métricas de desempenho adotadas permitem analisar a eficácia do modelo na identificação de padrões fraudulentos.

4.1 Execução do simulador de transações

Para compor a base de dados utilizada neste trabalho, foi executado o simulador de transações com os parâmetros apresentados a seguir:

- **n_clientes** = 5000
- **n_terminais** = 10000
- **n_dias** = 365
- **data_inicio** = "2024-01-01"
- **r** = 5

Como resultado da configuração adotada, o simulador gerou um total de **3.492.276 transações**, distribuídas entre os **5000 clientes** e os **10.000 terminais**. Esse volume de dados é expressivo e fundamental para análises robustas de detecção de fraudes, uma vez que representa um cenário de transações distribuídas ao longo de um ano completo (**365 dias**). Esse horizonte temporal permite capturar variações sazonais e flutuações no comportamento de consumo, contribuindo para a avaliação de diferentes estratégias de modelagem preditiva.

Tabela 1 – Exemplo das primeiras transações simuladas

TRANSACAO_ID	TX_DATA	CLIENTE_ID	TERMINAL_ID	TX_VALOR	TX_TEMPO_SEGUNDOS	TX_TEMPO_DIAS
0	2024-01-01 00:00:31	596	298	57,16	31	0
1	2024-01-01 00:02:10	4961	441	81,51	130	0
2	2024-01-01 00:07:56	2	316	146,00	476	0
3	2024-01-01 00:09:29	4128	370	64,49	569	0
4	2024-01-01 00:10:34	927	415	50,99	634	0

Fonte: Autor

A Tabela 1 ilustra um exemplo das primeiras transações geradas pelo simulador, detalhando informações como o identificador da transação, data e hora, cliente e terminal associados, valor transacionado e o tempo em segundos e dias desde o início da simulação.

O tempo de processamento das etapas para geração da base de dados está descrito na Tabela 2. Nota-se que, mesmo com a complexidade envolvida e o elevado volume de dados, o simulador foi capaz de produzir a base de forma rápida e eficiente. A geração das tabelas de clientes e terminais consumiu um tempo ínfimo, enquanto a etapa de associação — que exige a verificação da proximidade entre clientes e terminais —, embora mais demorada, ainda foi concluída em um tempo bastante reduzido.

Tabela 2 – Desempenho das etapas de geração da base de dados

Etapa	**Tempo (segundos)**
Geração da tabela de clientes	0,041
Geração da tabela de terminais	0,004
Associação entre clientes e terminais	1,4

Fonte: Autor

4.1.1 Análise da Qualidade dos Dados Gerados

Para assegurar a qualidade dos dados gerados, analisou-se a distribuição dos valores e tempos das transações por meio de histogramas, considerando os primeiros 10 dias da simulação. A avaliação foi realizada sobre um subconjunto de 10.000 amostras, permitindo a análise separada dos valores transacionados e dos tempos das transações.

4.1.2 Distribuição dos Valores das Transações

Inicialmente, os valores das transações foram extraídos e representados graficamente. O histograma gerado, apresentado no primeiro gráfico da Figura 3, exibe a distribuição da variável **TX_VALOR**, evidenciando que a maioria das transações ocorre com valores menores. Esse comportamento está alinhado com os parâmetros definidos no simulador, refletindo uma realidade em que transações de pequeno valor são mais frequentes. Tal distribuição é consistente com padrões observados em sistemas financeiros reais, onde há predominância de transações de baixo valor, enquanto valores elevados ocorrem com menor frequência.

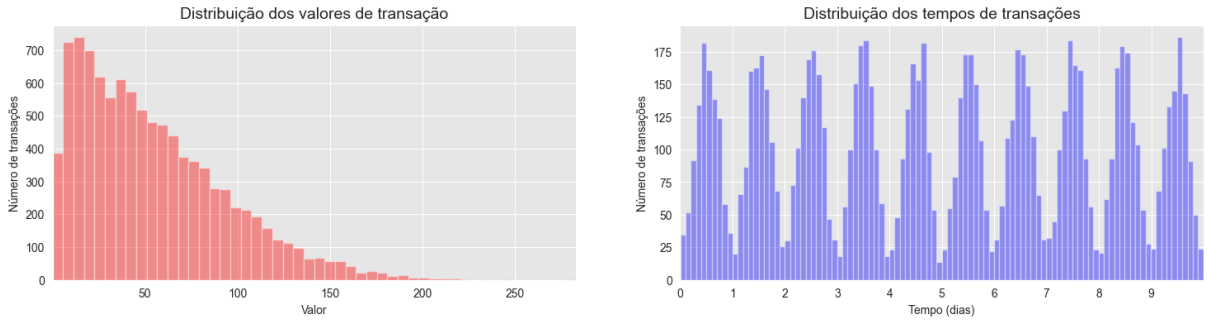
4.1.3 Distribuição dos Tempos das Transações

O histograma, gráfico da direita da Figura 3, representa a variável **TX_TEMPO_SEGUNDOS**, que indica o instante do dia em que cada transação ocorreu. Para facilitar a interpretação, os valores de tempo foram convertidos para dias, dividindo-os por 86.400 segundos. O histograma resultante revelou uma distribuição aproximadamente gaussiana

dos tempos de transação ao longo do dia, com concentração em torno do meio-dia. Essa distribuição reflete o comportamento esperado, dado que a maioria das transações em sistemas financeiros tende a ocorrer durante o horário comercial.

Os resultados obtidos por meio desses gráficos corroboram as expectativas estabelecidas na modelagem do simulador, validando que os padrões de valores e tempos das transações gerados seguem um comportamento realista. A predominância de transações de baixo valor e a distribuição centralizada dos tempos ao longo do dia reforçam a consistência dos dados simulados em relação ao funcionamento de sistemas de pagamentos reais.

Figura 7 – Análise das distribuições de valores e tepos das transações



Fonte: Autor

4.1.4 Gerador de cenários de fraude

A incorporação desses três cenários resultou na adição de um total de **196.000 transações fraudulentas**, representando **5,6%** de todas as transações presentes no conjunto de dados. Esse percentual permite avaliar a capacidade dos modelos de detecção ao enfrentar um cenário onde fraudes ocorrem em uma proporção realista dentro de sistemas financeiros.

Tabela 3 – Contagem de fraudes por cenário

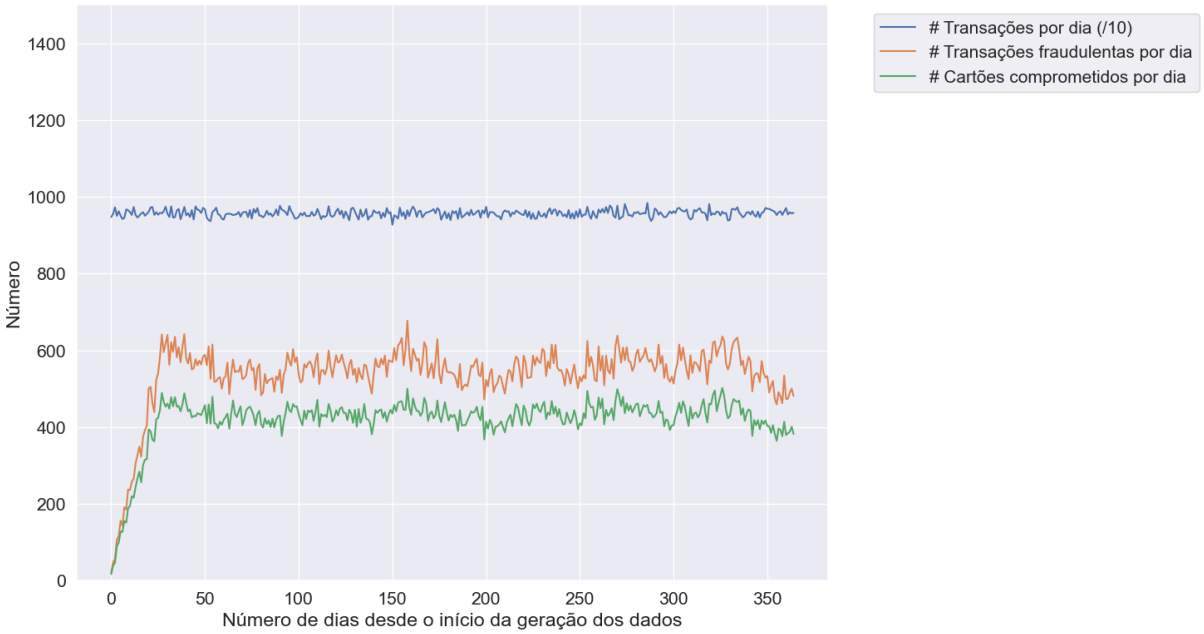
Cenário	**Número de Fraudes**
Cenário 1	1.930
Cenário 2	185.169
Cenário 3	8.901

Fonte: Autor

A simulação resultou em uma média de aproximadamente **9.500 transações por dia**. A Figura 8 ilustra a evolução do total de transações diárias, do número de transações fraudulentas e da quantidade de cartões comprometidos ao longo dos **365 dias** simulados.

A **linha azul** representa a série temporal do total de transações por dia, estabilizando-se em torno de **950 transações diárias**. Esse padrão constante reflete o comportamento

Figura 8 – Análise das distribuições de valores e tepos das transações



Fonte: Autor

previsível dos consumidores, com baixa variabilidade no volume de transações ao longo do tempo, comportamento comum em redes de pagamento com bases consolidadas de clientes.

A **linha laranja** demonstra a evolução do número de transações fraudulentas por dia, com média de aproximadamente **600 fraudes diárias**. Observa-se uma tendência de crescimento nos primeiros dias da simulação, seguida de estabilização. Essa variação inicial ocorre devido às características dos **cenários 2 e 3**, nos quais o comprometimento de terminais e cartões é temporário, com duração de **28 dias** e **14 dias**, respectivamente. Após esse período inicial de ativação dos cenários, o volume de fraudes estabiliza-se, ainda que com oscilações periódicas que refletem a reincidência desses eventos simulados.

A **linha verde** acompanha a evolução do número de cartões comprometidos ao longo do tempo e segue um padrão semelhante ao das fraudes, com média de aproximadamente **450 cartões comprometidos por dia**. A variação observada reforça a natureza cíclica dos ataques simulados, evidenciando períodos de maior atividade criminosa seguidos por momentos de normalidade. Tal comportamento destaca a importância de métodos preditivos que consigam capturar padrões temporais e variações sazonais.

4.1.5 Enriquecimento dos Dados

O processo de enriquecimento de dados foi fundamental para aumentar a capacidade preditiva do modelo de detecção de fraudes. Foram criadas variáveis adicionais baseadas em janelas temporais e no comportamento dos clientes e terminais, permitindo capturar padrões e anomalias que não seriam perceptíveis apenas com as variáveis originais.

4.1.5.1 Análise das Variáveis Enriquecidas

Foram geradas 14 novas variáveis enriquecidas, contemplando características temporais, de comportamento de clientes e de risco de terminais. Nenhuma dessas variáveis apresentou valores ausentes, o que assegura a integridade dos dados e evita tratamentos adicionais para imputação.

A Tabela 4 apresenta as estatísticas descritivas dessas variáveis, evidenciando variações significativas em termos de média, desvio padrão e amplitude dos dados.

Tabela 4 – Estatísticas descritivas das variáveis enriquecidas

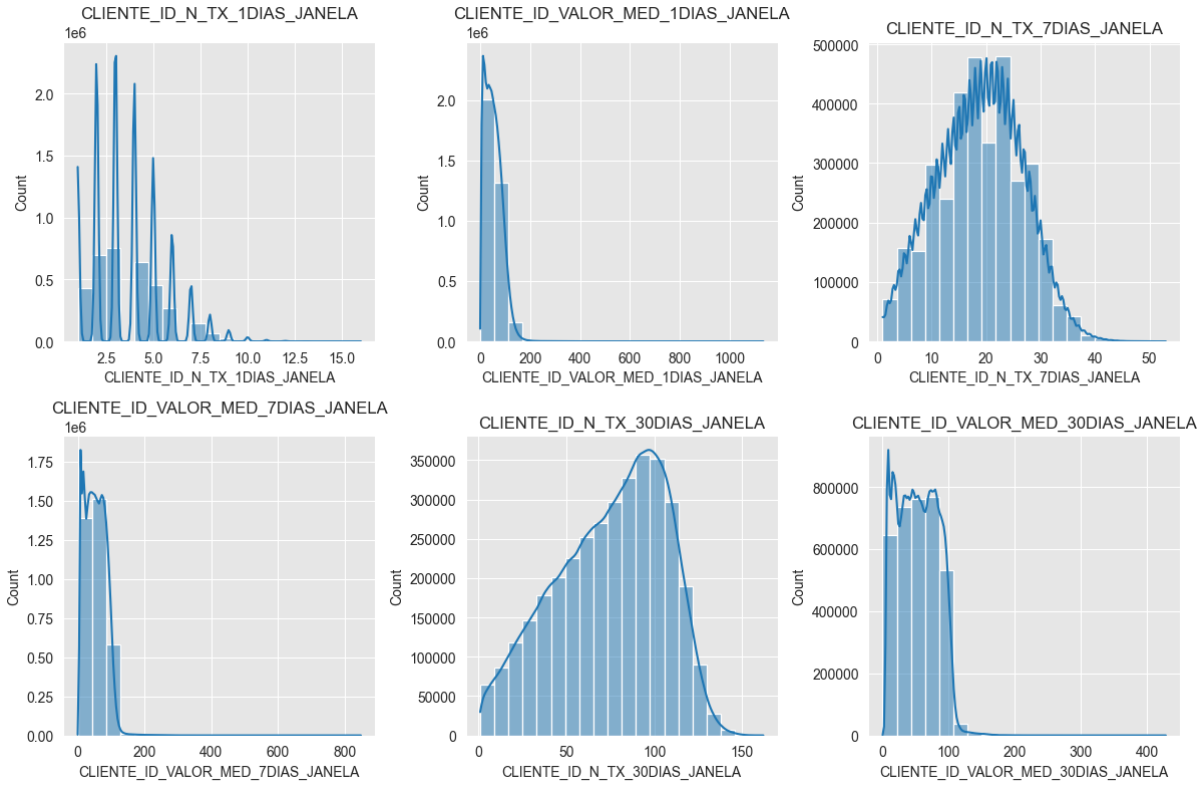
Variável	Média	Desvio Padrão	Mínimo	1º Quartil	Mediana	3º Quartil	Máximo
TX_FIM_SEMANA	0,28	0,45	0,00	0,00	0,00	1,00	1,00
TX_DURANTE_NOITE	0,17	0,38	0,00	0,00	0,00	0,00	1,00
CLIENTE_ID_N_TX_1DIAS_JANELA	3,57	1,85	1,00	2,00	3,00	5,00	16,00
CLIENTE_ID_VALOR_MED_1DIAS_JANELA	53,64	35,00	0,00	25,68	49,30	76,02	1134,50
CLIENTE_ID_N_TX_7DIAS_JANELA	18,83	7,81	1,00	13,00	19,00	24,00	53,00
CLIENTE_ID_VALOR_MED_7DIAS_JANELA	53,63	30,44	0,02	28,24	52,39	76,94	849,28
CLIENTE_ID_N_TX_30DIAS_JANELA	74,98	30,85	1,00	52,00	79,00	100,00	162,00
CLIENTE_ID_VALOR_MED_30DIAS_JANELA	53,63	29,27	0,21	28,72	53,06	77,68	428,06
ID_TERMINAL_N_TX_1DIAS_JANELA	10,53	5,53	0,00	7,00	10,00	13,00	65,00
ID_TERMINAL_RISCO_1DIAS_JANELA	0,05	0,22	0,00	0,00	0,00	0,00	1,00
ID_TERMINAL_N_TX_7DIAS_JANELA	73,10	32,75	0,00	54,00	69,00	87,00	317,00
ID_TERMINAL_RISCO_7DIAS_JANELA	0,05	0,21	0,00	0,00	0,00	0,00	1,00
ID_TERMINAL_N_TX_30DIAS_JANELA	303,05	140,96	0,00	226,00	289,00	366,00	1251,00
ID_TERMINAL_RISCO_30DIAS_JANELA	0,05	0,18	0,00	0,00	0,00	0,01	1,00

Fonte: Autor

4.1.5.2 Análise das Variáveis Relacionadas aos Clientes

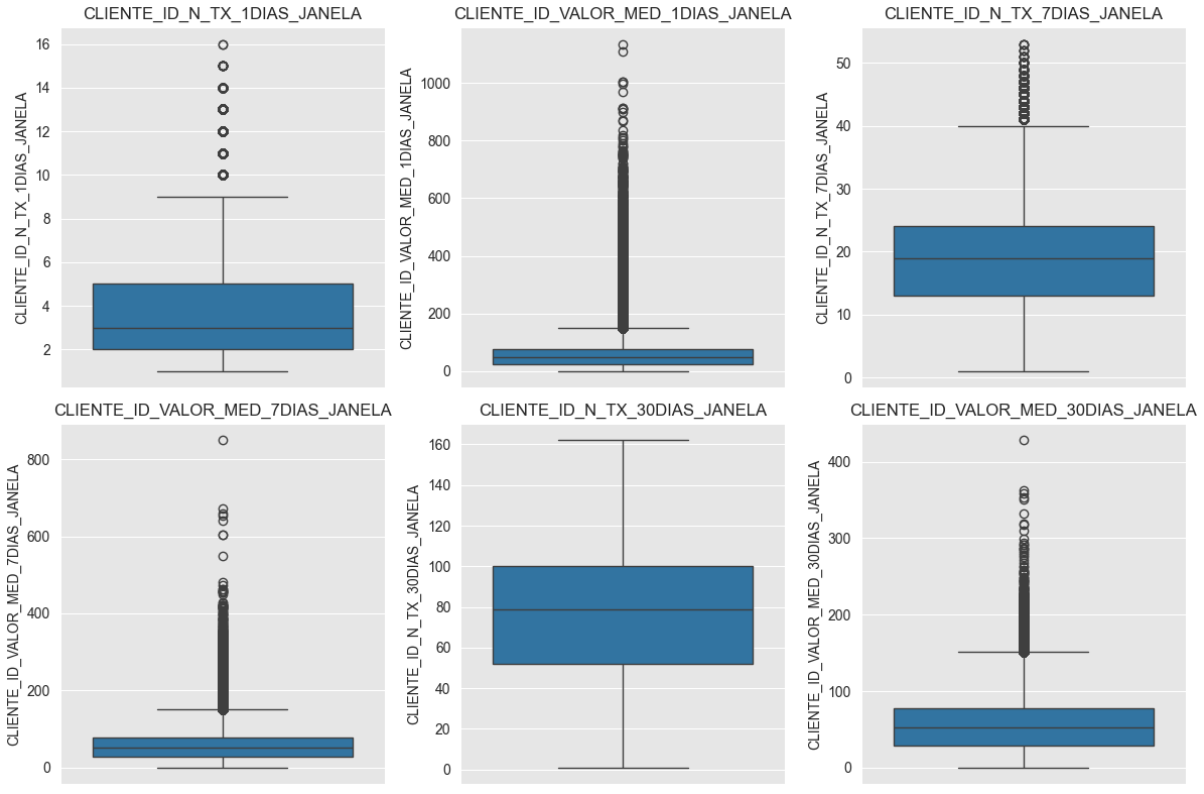
As variáveis relacionadas ao comportamento dos clientes ao longo das janelas de 1, 7 e 30 dias apresentaram distribuições com assimetria à direita, conforme observado nos histogramas e boxplots das Figuras 9 e 10. Isso indica que a maioria dos clientes realiza um número moderado de transações, mas há casos isolados com volumes elevados. Por exemplo, a variável **CLIENTE_ID_N_TX_30DIAS_JANELA** possui mediana de 79 transações, com valores que chegam a 162.

Figura 9 – Distribuição das variáveis enriquecidas relacionadas aos clientes



Fonte: Autor

Figura 10 – Boxplots das variáveis enriquecidas relacionadas aos clientes



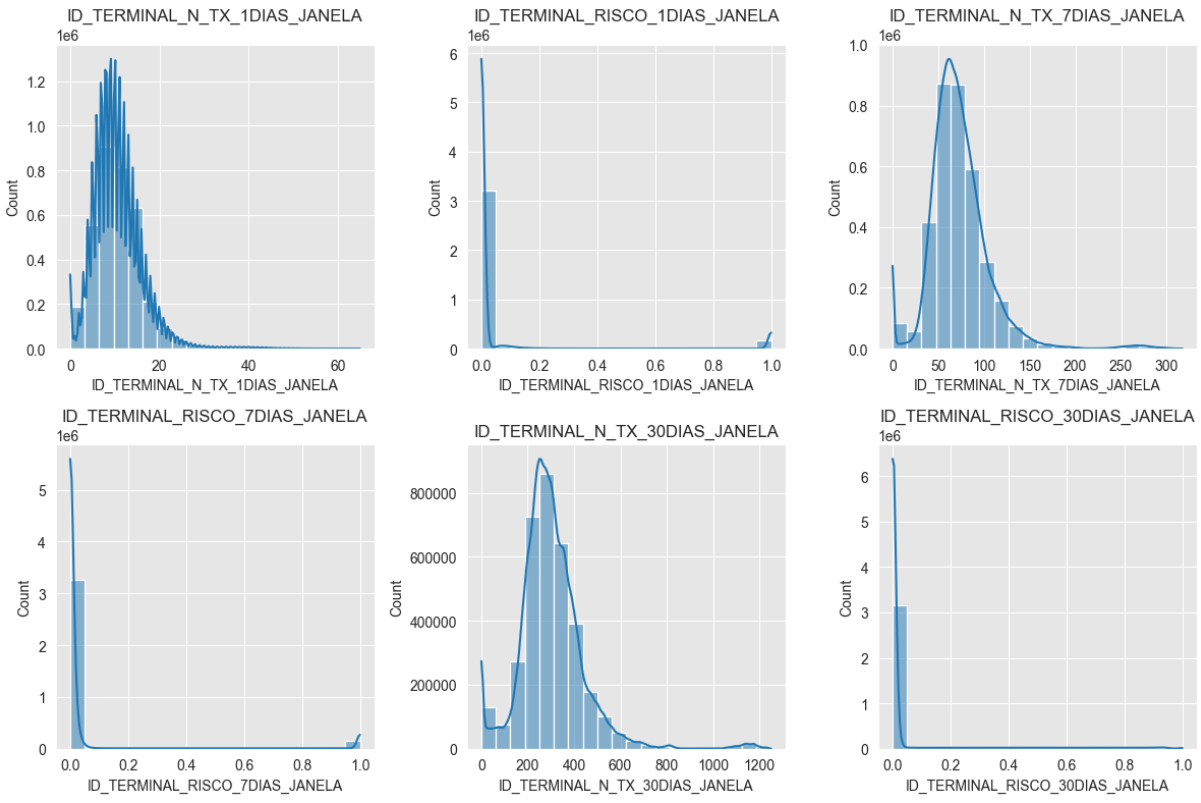
Fonte: Autor

As variáveis de valor médio das transações (**CLIENTE_ID_VALOR_MED**) apresentaram outliers significativos, sugerindo que alguns clientes realizam transações substancialmente superiores à média geral.

4.1.5.3 Análise das Variáveis Relacionadas aos Terminais

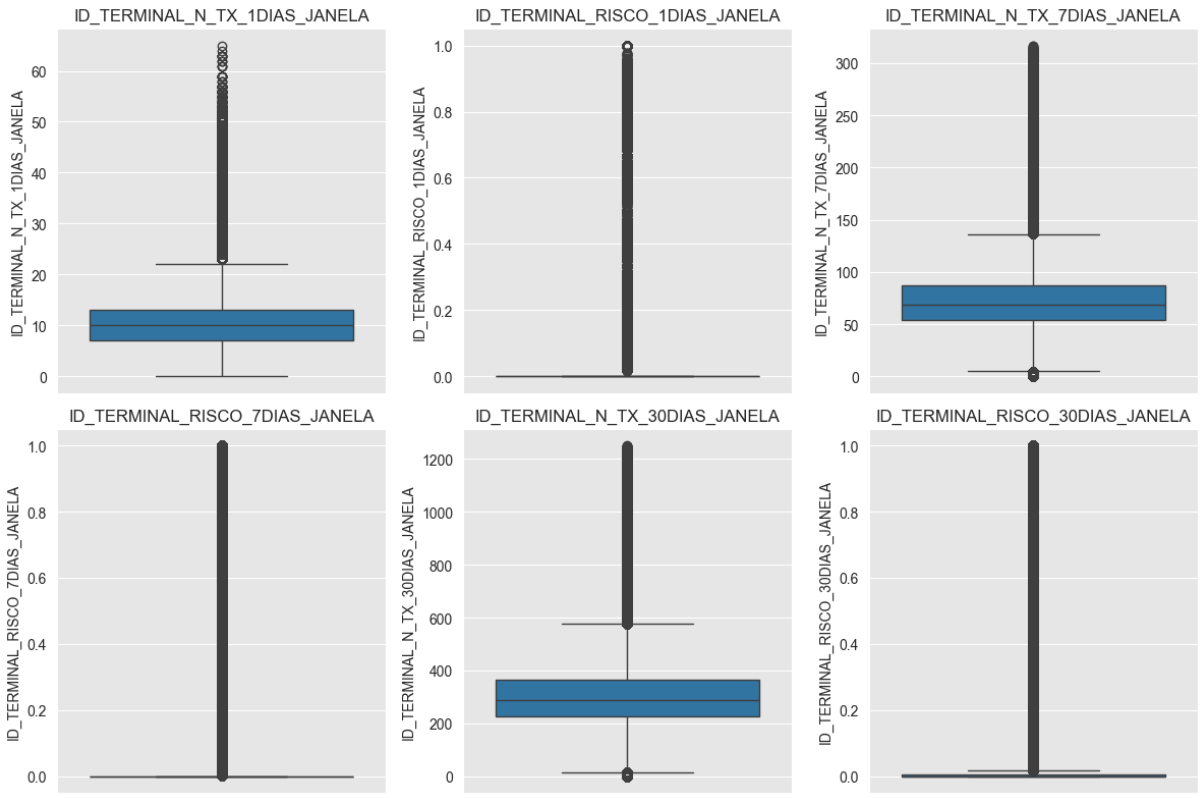
As variáveis associadas aos terminais de pagamento apresentaram um comportamento distinto em comparação às variáveis dos clientes. As Figuras 11 e 12 mostram que a maioria dos terminais processa um volume moderado de transações. A mediana da variável **ID_TERMINAL_N_TX_7DIAS_JANELA** foi de 69 transações, com valores máximos atingindo 317.

Figura 11 – Distribuição das variáveis enriquecidas relacionadas aos terminais



Fonte: Autor

Figura 12 – Boxplots das variáveis enriquecidas relacionadas aos terminais



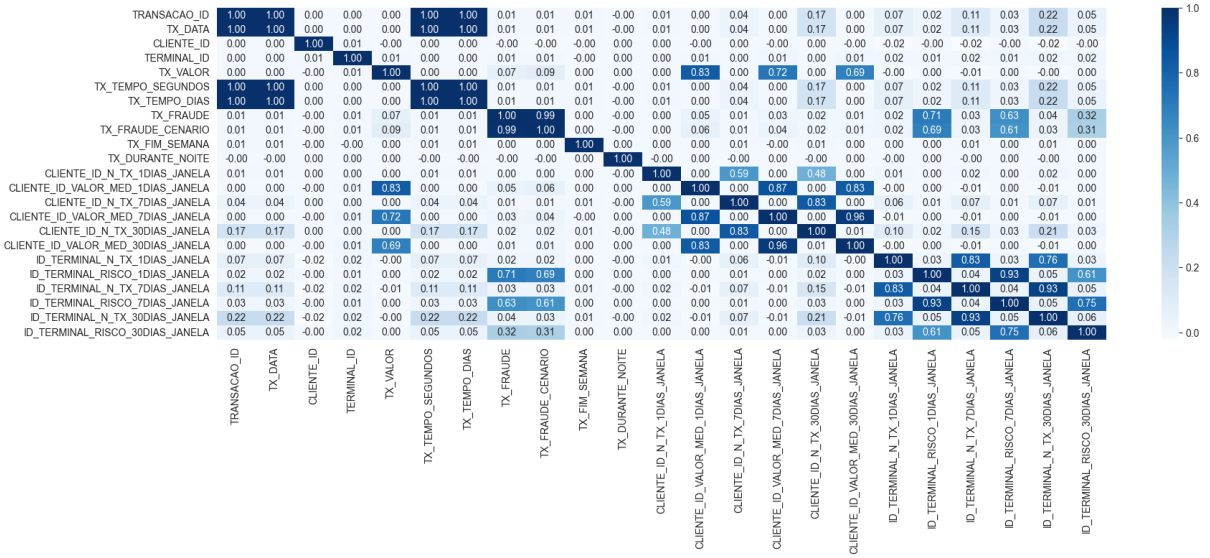
Fonte: Autor

As variáveis de risco dos terminais apresentaram um padrão em que a maioria dos valores esteve concentrada próximos a zero, com poucos terminais atingindo o valor máximo de 1, o que representa uma alta concentração de fraudes.

4.1.5.4 Correlação Entre Variáveis

A matriz de correlação, apresentada na Figura 13, revelou relações interessantes entre as variáveis enriquecidas. As variáveis de contagem de transações em janelas temporais mostraram alta correlação entre si, principalmente entre janelas próximas (1, 7 e 30 dias). Por outro lado, as variáveis de risco apresentaram correlações mais fracas com as demais variáveis, sendo especialmente úteis para identificar padrões específicos de comportamento fraudulento.

Figura 13 – Matriz de correlação entre as variáveis enriquecidas



Fonte: Autor

O enriquecimento dos dados proporcionou um conjunto robusto de características que capturam tanto aspectos temporais quanto comportamentais das transações. A ausência de valores ausentes e a variabilidade das variáveis reforçam a importância de abordagens de modelagem que considerem outliers e dados desbalanceados. As variáveis criadas, ao oferecer informações detalhadas sobre o comportamento dos clientes e terminais, aumentaram a capacidade de detecção de padrões anômalos e melhoraram a robustez do sistema de detecção de fraudes.

4.1.6 Resultados do Aprendizado de Máquina

Para avaliar a eficácia do modelo de detecção de fraudes, os dados foram divididos em conjuntos de treinamento e teste. O período de treinamento compreendeu transações realizadas entre 1° de abril e 31 de maio de 2024, enquanto o período de teste abrangeu os dias 8 a 14 de junho de 2024, com um intervalo de atraso entre 1° e 7 de junho de 2024. No total, foram carregadas 716.695 transações, das quais 42.159 foram classificadas como fraudulentas.

4.1.6.1 Divisão dos Conjuntos de Treinamento e Teste

A partir dos dados carregados, a função de separação gerou um conjunto de treinamento com 582.608 transações, contendo 33.863 registros fraudulentos, e um conjunto de teste com 18.538 transações, das quais 461 eram fraudes. Isso resultou em uma taxa de fraudes no conjunto de teste de aproximadamente 1,36% do total de transações.

4.1.6.2 Modelos de Classificação e Treinamento

Foram avaliados cinco algoritmos de aprendizado de máquina para a detecção de fraudes:

- **Regressão Logística**
- **Árvore de Decisão (profundidade 2)**
- **Árvore de Decisão (profundidade ilimitada)**
- **Random Forest**
- **XGBoost**

Cada um desses modelos foi ajustado com os dados de treinamento e testado no conjunto de validação. O tempo total de computação para ajuste e previsões foi de 1 minuto e 46 segundos.

4.1.6.3 Avaliação de Performance nos Dados de Teste

A avaliação dos modelos foi realizada com as métricas AUC-ROC, Average Precision e Precisão do Cartão@100. Os resultados estão descritos na Tabela 5.

Tabela 5 – Desempenho dos Modelos no Conjunto de Teste

Modelo	**AUC-ROC**	**Average Precision**	**Precisão do Cartão@100**
Regressão Logística	0,700	0,407	0,166
Árvore de Decisão (prof. 2)	0,690	0,381	0,150
Árvore de Decisão (ilimitada)	0,649	0,111	0,137
Random Forest	0,702	0,432	0,180
XGBoost	0,685	0,426	0,164

Fonte: Autor

Os resultados indicam que os modelos Random Forest e Regressão Logística apresentaram os melhores desempenhos, com AUC-ROC acima de 0,70 e precisão do Cartão@100 superior a 16%. A Árvore de Decisão com profundidade ilimitada teve o pior desempenho, com AUC-ROC de 0,649 e baixa precisão média.

4.1.6.4 Avaliação de Performance nos Dados de Treinamento

A mesma avaliação foi aplicada ao conjunto de treinamento, conforme a Tabela 6.

Os valores de AUC-ROC e Average Precision demonstram que os modelos Random Forest e Árvore de Decisão com profundidade ilimitada tiveram desempenho perfeito nos dados de treinamento, o que sugere possível overfitting.

Tabela 6 – Desempenho dos Modelos no Conjunto de Treinamento

Modelo	**AUC-ROC**	**Average Precision**	**Precisão do Cartão@100**
Regressão Logística	0,873	0,615	0,484
Árvore de Decisão (prof. 2)	0,859	0,580	0,486
Árvore de Decisão (ilimitada)	1,000	1,000	0,521
Random Forest	1,000	1,000	0,527
XGBoost	0,932	0,761	0,497

Fonte: Autor

4.1.6.5 Tempo de Execução dos Modelos

A Tabela 7 apresenta o tempo de execução do treinamento e das previsões para cada modelo.

Tabela 7 – Tempo de Execução dos Modelos

Modelo	**Treinamento (s)**	**Previsão (s)**
Regressão Logística	0,68	0,006
Árvore de Decisão (prof. 2)	0,87	0,004
Árvore de Decisão (ilimitada)	14,25	0,008
Random Forest	83,57	0,123
XGBoost	1,54	0,014

Fonte: Autor

Os tempos de execução indicam que modelos mais simples, como Regressão Logística e Árvore de Decisão com profundidade limitada, apresentam treinamento rápido e previsões eficientes. Já a Random Forest, embora tenha apresentado alta performance, demandou um tempo de treinamento significativamente maior.

4.1.6.6 Conclusão

Os resultados demonstram que o modelo Random Forest obteve a melhor performance geral, seguido da Regressão Logística e do XGBoost. No entanto, o tempo de execução do Random Forest pode ser um fator limitante em aplicações em tempo real. Além disso, a Árvore de Decisão ilimitada mostrou forte overfitting, indicando a necessidade de ajustes na complexidade do modelo. Assim, uma abordagem balanceada entre acurácia e eficiência computacional deve ser considerada para a implementação em ambiente produtivo.

5 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de um sistema de detecção de fraudes em transações financeiras utilizando aprendizado de máquina, com ênfase na transformação de características e análise temporal. A crescente digitalização dos serviços financeiros e o consequente aumento no volume de transações tornaram a detecção de fraudes um desafio crucial para instituições bancárias e empresas do setor. Diante desse cenário, buscou-se implementar um modelo preditivo capaz de identificar padrões suspeitos, minimizando perdas financeiras e garantindo maior segurança para clientes e instituições.

Durante o desenvolvimento deste estudo, foi realizada a simulação de dados de transações financeiras, permitindo a criação de um conjunto representativo das operações reais. Para isso, foram utilizadas distribuições estatísticas, como a distribuição normal para a modelagem dos valores das transações, a distribuição de Poisson para a geração do número de transações diárias por cliente e a distribuição uniforme para alocação de coordenadas geográficas de clientes e terminais. Esse processo foi fundamental para garantir a robustez dos dados e simular cenários realistas que pudessem ser utilizados no treinamento e validação dos modelos de detecção de fraudes.

A geração e enriquecimento dos dados desempenhou um papel essencial, sendo a transformação de características um dos pontos-chave para a obtenção de um modelo eficiente. Foram criadas variáveis temporais, comportamentais e de risco associadas a clientes e terminais, permitindo que os modelos de aprendizado de máquina captassem padrões mais sofisticados. Técnicas como normalização, padronização, agregação temporal e cálculo de métricas de risco foram aplicadas para aprimorar a qualidade dos dados de entrada e, consequentemente, melhorar o desempenho dos classificadores.

Na aplicação das técnicas de aprendizado de máquina, foram exploradas diversas abordagens supervisionadas, incluindo métodos como Regressão Logística, Árvores de Decisão, Random Forest e XGBoost. A análise dos resultados permitiu identificar que modelos baseados em algoritmos como Random Forest e Regressão Linear apresentaram os melhores desempenhos em termos de precisão e sensibilidade na identificação de transações fraudulentas. Além disso, a inclusão de variáveis temporais foi fundamental para captar padrões sazonais e tendências que diferenciam transações.

Ao longo do estudo, também foram discutidos os desafios enfrentados na implementação de um sistema de detecção de fraudes em um ambiente real. Entre os principais desafios, destacam-se a necessidade de processamento em tempo real, a adaptação a novos padrões de fraude e a explicabilidade dos modelos para garantir conformidade regulatória

e transparência nas decisões automatizadas.

Com base nos resultados obtidos, conclui-se que a aplicação de aprendizado de máquina na detecção de fraudes financeiras é uma abordagem promissora, desde que acompanhada de uma engenharia de dados robusta e estratégias eficazes de atualização e validação dos modelos. O impacto da implementação de tais sistemas não se restringe apenas à redução de perdas financeiras, mas também contribui para a confiança dos clientes e a integridade do sistema financeiro.

Por fim, este trabalho abre espaço para futuras pesquisas que possam aprimorar a detecção de fraudes, incluindo o uso de técnicas mais avançadas como deep learning, redes neurais recorrentes para análise temporal e modelos generativos para a criação de dados sintéticos que auxiliem na robustez do treinamento. Além disso, a integração com técnicas de explainable AI (XAI) pode ser explorada para aumentar a interpretabilidade dos modelos e facilitar sua adoção em ambientes regulatórios mais rígidos.

Dessa forma, reforça-se a importância do uso de aprendizado de máquina no combate a fraudes financeiras, destacando sua capacidade de identificar padrões anômalos com eficiência e precisão, e contribuindo para um ambiente transacional mais seguro e confiável.

REFERÊNCIAS

AGRESTI, A. **An Introduction to Categorical Data Analysis**. 2. ed. New York: Wiley, 2007.

ARTHUR, S.; SHEFFRIN, S. M. **Economics: principles in action**. [*S.l.: s.n.*]: Upper Saddle River, New Jersey, 2003. v. 7458. 173 p.

Banco Central Europeu. **Relatório sobre fraudes com cartões de crédito**. [*S.l.*], 2018. Acesso em: 27 jan. 2025. Disponível em: https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport202008~521edb602b.en.html.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006.

BORGNE, Y.-A. L. *et al.* **Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook**. Université Libre de Bruxelles, 2022. Disponível em: https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook.

CASELLA, G.; BERGER, R. L. **Statistical Inference**. 2. ed. Pacific Grove: Cengage Learning, 2021.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. *In*: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. San Francisco: ACM, 2016. p. 785–794.

DEGROOT, M. H.; SCHERVISH, M. J. **Probability and Statistics**. 4. ed. Boston: Pearson, 2012.

DEVORE, J. L. **Probability and Statistics for Engineering and the Sciences**. 8. ed. Boston: Cengage Learning, 2011.

DIETTERICH, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. **Machine Learning**, v. 40, n. 2, p. 139–158, 2000.

FAWCETT, T. An introduction to roc analysis. **Pattern Recognition Letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, v. 29, n. 5, p. 1189–1232, 2001.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. Waltham: Morgan Kaufmann, 2012.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3rd. ed. Waltham: Morgan Kaufmann, 2012.

HERNáNDEZ-ORALLO, J.; FLACH, P.; FERRI, C. A unified view of performance metrics: translating threshold choice into expected classification loss. **Journal of Machine Learning Research**, JMLR.org, v. 13, p. 2813–2869, 2012.

JAMES, G. *et al.* **An Introduction to Statistical Learning**. New York: Springer, 2013.

LIAW, A.; WIENER, M. Classification and regression by randomforest. **R News**, v. 2, n. 3, p. 18–22, 2002.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1997.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied Statistics and Probability for Engineers**. 7. ed. New York: Wiley, 2019.

MOOD, A. M.; GRAYBILL, F. A.; BOES, D. C. **Introduction to the Theory of Statistics**. 3. ed. New York: McGraw-Hill, 1974.

POZZOLO, A. D. **Adaptive Machine Learning for Credit Card Fraud Detection**. 2015. Tese (Tese (Doutorado em Ciências da Computação)) — Université libre de Bruxelles, Bélgica, 2015.

PREGIBON, D. Logistic regression. **Proceedings of the Workshop on Computationally Intensive Statistical Methods**, v. 9, n. 3, p. 3–14, 1981.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. San Mateo: Morgan Kaufmann, 1993.

ROSS, S. M. **Simulation**. 5. ed. [*S.l.: s.n.*]: Academic Press, 2017.

SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 21, n. 3, p. 660–674, 1991.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. **PLoS ONE**, v. 10, n. 3, p. e0118432, 2015.

SILVA, P. R. **Psicologia do risco de crédito: análise da contribuição de variáveis psicológicas em modelos de credit scoring**. 2011. Tese (Tese (Doutorado em Administração)) — Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, São Paulo, 2011.

SSC), P. C. I. S. S. C. P. **Payment Card Industry (PCI) Data Security Standard: Requirements and Security Assessment Procedures**. [*S.l.*], 2019. v. 3.2.1. Acesso em: 8 fev. 2025. Disponível em: https://www.pcisecuritystandards.org/.

WIKIPEDIA. **Distribuição de Poisson**. 2025. https://pt.wikipedia.org/wiki/Distribui%C3%A7%C3%A3o_de_Poisson. Acessado em: 28 de fevereiro de 2025.

Wikipédia. **Distribuição Normal**. 2024. https://en.wikipedia.org/wiki/Normal_distribution. Acessado em: 28 fev. 2024.

ZHENG, A. **Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists**. 1. ed. Sebastopol: O'Reilly Media, 2018.

ZIBETTI, A. **Distribuição Normal**. 2024. Acesso em: 28 fev. 2025. Disponível em: https://www.inf.ufsc.br/~andre.zibetti/probabilidade/normal.html.