WILLIAN ALVES BARBOZA

SOLUÇÃO DE BIG DATA PARA DETECÇÃO DE TRANSAÇÕES FRAUDULENTAS EM CARTÕES FÍSICOS DE CRÉDITO

Monografia apresentada ao Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo, para obtenção do título de Especialista, pelo Programa de Pós-Graduação em Big Data - Inteligência na Gestão dos Dados.

SÃO PAULO

2024

WILLIAN ALVES BARBOZA

SOLUÇÃO DE BIG DATA PARA DETECÇÃO DE TRANSAÇÕES FRAUDULENTAS EM CARTÕES FÍSICOS DE CRÉDITO

Monografia apresentada ao Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo, para obtenção do título de Especialista, pelo Programa de Pós-Graduação em Big Data - Inteligência na Gestão dos Dados.

Área de concentração: Tecnologia da Informação - Big Data

Orientador: Prof. MEE Jonas Santiago de Oliveira

SÃO PAULO

2024

FICHA CATALOGRÁFICA

Barboza, Willian
SOLUÇÃO DE BIG DATA PARA DETECÇÃO DE TRANSAÇÕES FRAUDULENTAS EM CARTÕES FÍSICOS DE CRÉDITO / W. Barboza -- São Paulo, 2023. 63p.

Monografia (Especialização em Engenharia de Dados & Big Data) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.Fraudes cartões de crédito 2.Big Data I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

AGRADECIMENTOS

Ao Prof. MEE Jonas Santiago, meu orientador, gostaria de expressar minha sincera gratidão pela dedicação incansável, paciência e valiosos ensinamentos que tiveram um impacto significativo em meu crescimento profissional e foram fundamentais para a elaboração deste trabalho.

Agradeço aos professores do Programa de Educação Continuada, com especial reconhecimento à Profa. Dra. Solange Nice Alves de Souza, pela generosa partilha de conhecimento e por abrir novos horizontes de aprendizado.

Aos meus pais que sempre me incentivaram a continuar com meus estudos e aos meus amigos de Pós-graduação, Erik Assunção Figueiredo e Paulo Henrique De Souza Pereira Prazeres. Em especial, não poderia deixar de mencionar meu profundo agradecimento à Rosa Daniele Ramos, cujo apoio e assistência foram pilares essenciais em grande parte do percurso deste curso.

CURSO Engenharia de Dados & Big Data

Coord.: Prof. Solange N. Alves de Souza
Vice-Coord.: Pedro Luiz Pizzigatti Corrêa

Perspectivas profissionais alcançadas com o curso:

O curso oferecido pela Escola Politécnica da Universidade de São Paulo, PECE-USP foi essencial para ampliar os conhecimentos em Big Data e engenharia de dados, pois foram abordados tópicos de arquitetura, tecnologias e metodologias capazes de formar um olhar mais crítico e assertivo.

RESUMO

O crédito é amplamente utilizado como meio de pagamento nos comércios atacadistas e varejistas, representando um valor significativo de transações anualmente. No entanto, devido ao volume financeiro envolvido, atrai a atenção de fraudadores em busca de vantagens financeiras. Apesar das medidas adotadas pelas empresas para combater fraudes, os fraudadores estão sempre buscando novas maneiras de obter sucesso em suas ações.

Os esforços para detectar fraudes estão em constante evolução, com profissionais buscando novas formas de identificar atividades fraudulentas. No entanto, os fraudadores também se adaptam e desenvolvem técnicas sofisticadas para contornar os sistemas de detecção existentes. Isso cria um desafio para os profissionais, que precisam estar atualizados e buscar constantemente novas maneiras de combater a fraude.

Nesse contexto, uma solução de Big Data é apresentada como uma abordagem para identificar transações suspeitas de fraude de forma mais rápida e eficiente. Em um cenário em que as fraudes financeiras estão em constante evolução, táticas como a clonagem de cartão se tornaram uma ameaça significativa. Além disso, outra prática comum entre fraudadores é realizar várias transações de pequeno valor em um curto espaço de tempo, na esperança de passarem despercebidas.

A solução proposta por este projeto implementa um conjunto de ferramentas de Big Data e Machine Learning para a análise e identificação próximo ao tempo real das transações com suspeitas de fraudes que foram realizadas através do uso de cartões de crédito físico.

Palavras-chave: Fraude, Big Data, crédito.

ABSTRACT

Credit is widely used as a means of payment in wholesale and retail trades, representing a significant value of transactions annually. However, due to the substantial financial volume involved, it attracts the attention of fraudsters seeking financial advantages. Despite the measures adopted by companies to combat fraud, perpetrators are constantly seeking new ways to succeed in their actions.

Efforts to detect fraud are continuously evolving, with professionals exploring new ways to identify fraudulent activities. However, fraudsters also adapt and develop sophisticated techniques to circumvent existing detection systems. This creates a challenge for professionals who need to stay updated and constantly seek new ways to combat fraud.

In this context, a Big Data solution is presented as an approach to identify suspicious fraud transactions more quickly and efficiently. In a scenario where financial frauds are constantly evolving, tactics such as card cloning have become a significant threat. Additionally, another common practice among fraudsters is to carry out multiple transactions of small amounts in a short period, hoping to go unnoticed.

The solution proposed by this project implements a set of Big Data and Machine Learning tools for the near real-time analysis and identification of transactions suspected of fraud, which were carried out through the use of physical credit cards.

Keywords: Fraud, Big Data, credit.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ABECS	Associação Brasileira de Cartões de Crédito e Servicos
CVC	Código de Verificação de Cartão
CVV	*Card Verification Value*
DW	*Data Warehouse*
ETL	*Extract, Transform e Load*
FEBRABAN	Federação dos Bancos Brasileiros
GMM	*Gaussian Mixture Model*
HDFS	*Hadoop Distributed File System*
LDA	*Latent Dirichlet Allocation*
ML	Machine Learning
NBD-PWG NIST	*Big Data Public Working Group*
NGA	Agência Nacional de Informação Geoespacial
NIST	*National Institute of Standards and Technology*
PDV	Ponto de Vendas
SMS	*Short Message Service*
SMOTE	*Synthetic Minority Over-sampling Technique*
Skimming	Captura de dados de cartões nas transações legítimas
SVM	Máquinas de Vetores de Suporte
Phishing	Captura fraudulenta de dados em pagamentos.
TI	Tecnologia da InformaçãoSumário

SUMÁRIO

1 INTRODUÇÃO

A utilização de cartões de crédito físicos como meio de pagamento nos terminais de vendas, é um meio de pagamento altamente utilizado tanto nos comercio de atacados quanto no varejo. Dessa forma, mesmo existindo outras opções de pagamento, a opção de pagamento por crédito pode ser considerada importante para à economia, pois segundo a Associação Brasileira de Cartões de Crédito e Servicos (ABECS, 2022) compras realizadas através de cartões pré-pagos, débitos e crédito entre janeiro e setembro de 2022 somam aproximadamente dois trilhoes de reais, outro número importante é a quantidade de transações que no mesmo período totalizam dez bilhões de transações.

Por ser um meio de pagamento que concentra uma parcela substancial das compras concentra um valor financeiro bastante elevado, despertando o interesses de fraudadores. Com o intuíto de obter vantagens financeiras que podem gerar prejuizo para as empresas conhecidas como adquirentes, ou credenciadoras, pois elas são empresas que oferecem essas transações em seu terminais de vendas.

Neste trabalho, é apresentada uma solução de *Big Data* utilzando algoritmos de Machine Learning para acelerar e aprimorar o processo de identificação de transações de cartão de crédito com potencial de serem fraudulentas. Através da aplicação de técnicas avançadas de *Big Data*, análise de transações realizadas com o mesmo cartão de crédito, porém em localidades distantes, mas com intervalo de tempo muito curto entre essas transações, geolocalização e correlação de transações em tempo quase real, a solução possibilita solução colaborará para identificar mais rapidamente transações de crédito que apresentem alto potencial para serem consideradas como fraudulentas.

1.1 Objetivo

O propósito deste projeto consiste no desenvolvimento de uma solução utilizando ferramentas de *Big Data* para auxiliar no processo de identificação e detecção transações fruadulentas suspeitas envolvendo cartões de crédito físicos em terminais de ponto de venda. Para atingir esse objetivo, o projeto se concentrará na detecção

de fraudes relacionadas com a divergência de geolocalização e com realização de múltiplas transações com um mesmo cartão em um curto intervalo de tempo.

1.5.1 Objetivos específicos

Para a elaboração deste trabalho, foram considerados os seguintes objetivos específicos:

1. Realizar pesquisa sobre transações com cartões de crédito físico realizadas nos Pontos de Vendas (PDV).
2. Apresentar uma solução de *Big Data* seguindo a arquitetura NIST para *Big Data*.
3. Apresentar uma solução de *Machine Learning* utilizando *Naive Bayes, Random Forest* ou K-NN
4. Desenvolver uma camada para visualização de dados para as áreas de fraudes das empresas ter acesso em *Dashboard* que permitam maior agilidade nas ações para identificar transações com suspeita de fraudes.
5. Desenvolver uma camada de notificação de mensagens flexível para que através dessa camada um grupo de analistas de fraudes recebam SMS relevantes alertando sobre transações com potencial de fraude.

1.2 Justificativa

As transações fraudulentas além de causarem prejuizos para as empresas, também influenciam na imagem ou reputação do nome da empesa perante o mercado de credenciadoras. A rapidez em identificar transações fraudulentas, permitirá a empresa detectar fraudes, e consequentemente, reduzir o prejuizo financeiro proveniente dessas transações.

Segundo a FEBRABAN (FEBRABAN, 2014), anualmente os bancos investem aproximadamente três bilhões de reais para tentar minimizar os prejuizos causados por fraudes.

O Brasil possui um mercado de cartões de crédito e débito em constante crescimento, com uma crescente preferência dos consumidores por essa forma de pagamento. De acordo com os dados da Associação Brasileira de Empresas de Cartões de Crédito e Serviços (ABECS) entre janeiro e setembro de 2022 as compras através do uso do cartão de crédito aumentaram em 25,6% em relação ao mesmo período do ano anterior, o que significa o registro de mais de R$ 527,6 bilhões de reais utilizando a opção de crédito (ABECS, 2022).

Em seguida, aparecem as transações com uso do cartão de débito, representando um crescimento de 1,2% e uma movimentação de aproximadamente R$ 240,5 bilhões de reais, e por último aparecem os cartões pré-pagos que apresentaram um crescimento de 84,7% e movimentaram em torno de R$ 59 bilhões de reais (ABECS, 2022).

Com a crescente dependência de transações eletrônicas e a conveniência dos pagamentos com cartão, os criminosos desenvolveram diversas técnicas sofisticadas para explorar vulnerabilidades no sistema, resultando em perdas financeiras substanciais e preocupações com a segurança dos dados. Nesta análise, exploraremos em detalhes os principais tipos de fraude de cartão de crédito em PDV, suas implicações e as estratégias de prevenção necessárias para proteger os consumidores e o comércio.

1.3 Metodologia

A pesquisa foi conduzida seguindo as principais etapas: leituras de artigos, dissertações e consultas na web para obter conhecimentos suficientes sobre os principais requisitos técnicos e de negócio relacionados aos Terminais de Pontos de Vendas, bem como os principais tipos de fraudes cometidas através de cartões de crédito físicos. Além disso, foram realizados estudos sobre Big Data, seguindo as determinações da arquitetura NIST para Big Data, e estudos de algoritmos de Machine Learning comparando três algoritmos para escolher o mais eficiente a ser utilizado nesta solução.

1.4 Organização do trabalho

A seguir, no Capítulo 2, serão explorados os principais conceitos e a fundamentação teórica do trabalho, incluíndo uma abordagem sobre Ponto de Vendas, principais tipos de fraudes realizadas através da utilização de cartões de crédito físico.

Continuando no Capítulo 2, há uma visão geral dos componentes de *Big Data*, seguindo as normas da arquitetura NIST, os algoritmos de *Machine Learning* e a fonte de dados utilizadas com base para o trabalho.

O Capítulo 3, relata a dificuldade em utilizar um *Data Warehouse* (DW) como base para identificar eventuais fraudes em cartões de crédito apresentando o processo atual de detecção de fraudes e seus principais desafios, em seguida apresenta o detalhamento do projeto, arquitetura, processos e etapas. Ainda no Capítulo 3, são apresentados a arquitetura proposta e os principais componentes em Big Data e o detalhamento das camadas envolvidas na solução.

Por fim, no Capítulo 4, serão apresentadas as contribuições desta monografias e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será fornecida uma visão geral dos conceitos fundamentais relacionados ao projeto, juntamente com as principais técnicas de *Big Data* empregadas na elaboração deste projeto.

2.1 Terminais de Pontos de Vendas

Conforme destacado por (JURGOVSKY et al., 2018), no contexto do setor de pagamentos, a fraude de cartão de crédito se materializa quando indivíduos obtêm informações de um cartão de forma ilícita, com o propósito de realizar compras sem a autorização explícita do titular. Entre os meios de pagamentos prejudicados por essas fraudes, aparecem os Pontos de Vendas.

Terminais de Ponto de Venda (PDV), também conhecidos como maquininhas de cartão, são dispositivos eletrônicos amplamente utilizados no comércio e em diversos setores para processar pagamentos. Segundo (SANTIAGO, 2014) esses terminais desempenham um papel fundamental nas operações comerciais modernas, oferecendo uma série de benefícios, como a conveniência de pagamento para os clientes e o registro eficiente de vendas para os comerciantes.

Os (PDVs) possibilitam que os clientes efetuem pagamentos utilizando cartões de crédito, débito, *vouchers* (tais como refeição e alimentação) e em determinados casos, dispositivos móveis. Esses dispositivos estabelecem conexão com uma rede de processamento de pagamentos para a autorização e conclusão das transações financeiras. Eles são projetados com a finalidade de simplificar e agilizar diversas operações relacionadas a transações financeiras e ao gerenciamento de vendas. Abaixo estão algumas das operações comuns permitidas em um PDV:

1. Pagamentos com Cartão de Crédito/Débito;
2. Pagamentos com Cartões Pré-pagos;
3. Pagamentos com Dispositivos Móveis;
4. Vales e Cupons;
5. Divisão de Contas;

6. Devoluções e Estornos;
7. Consultas de Saldo;
8. Impressão de Recibos;
9. Registro de Vendas.

Essas operações podem variar dependendo do tipo de estabelecimento comercial e do software específico usado no PDV. No entanto, em geral, os PDVs são projetados para simplificar e agilizar as transações financeiras dos clientes e permitir o gerenciamento de vendas pelos comerciantes.

2.2 Principais tipos de Fraudes em cartão de crédito físico

O cartão de crédito é uma forma de pagamento amplamente utilizada para efetuar compras e adquirir serviços (ABECS, 2022). Mensalmente, o titular do cartão recebe a fatura em seu endereço residencial, ou por email, que precisa ser quitada.

Nesse momento, o titular tem a opção de pagar o valor total da fatura, ou apenas o valor mínimo devido ou qualquer quantia intermediária, o que resultaria no adiamento do pagamento do saldo remanescente para o mês subsequente, sujeito a encargos financeiros (ABECS, 2022).

O banco emissor do cartão estabelece um limite de crédito para compras, sendo que a cada compra realizada, esse limite disponível é reduzido. Quando o limite disponível se esgota, novas tentativas de compras são negadas.

Ao efetuar o pagamento da fatura o limite de crédito é novamente liberado, tornando-o disponível para uso.

Diante das facilidades oferecidas por essa modalidade de pagamento, pessoas mal-intencionadas tentam realizar compras ou pagamentos de forma ilegal e fraudulentas.

2.2.1 Clonagem do cartão

A clonagem de cartão é uma das fraudes financeiras mais frequentes e lamentavelmente uma das mais sofisticadas (PARODI, 2008). Nesse tipo de fraude, os criminosos obtêm informações cruciais de um cartão de crédito ou débito, permitindo realizar compras não autorizadas ou realizar saques de contas das vítimas.

Essa prática ilegal está em constante evolução, impulsionada pela criatividade dos fraudadores e pelas vulnerabilidades tecnológicas.

A clonagem de um cartão é um processo que envolve etapas de coleta de dados, criação do cartão falso, utilização do cartão clonado e, por fim, a tentativa de evitar detecção, a seguir detalhados:

- Coleta de Dados: Os fraudadores geralmente adquirem as informações do cartão de crédito, como o número do cartão, a data de validade e o código de segurança (CVV/CVC).
- De acordo com (PARODI, 2008), isso pode ser realizado por meio de dispositivos de *skimming* instalados em caixas eletrônicos, terminais de pagamento ou até mesmo por meio de *phishing* online, em que os criminosos enganam as vítimas para que revelem seus detalhes de cartão.
- Criação do Cartão Falso: Após obter as informações do cartão, os fraudadores as gravam em um novo cartão. Isso pode ser feito com uma tarja magnética em branco ou até mesmo com tecnologia mais avançada, como a impressão 3D de cartões.
- Uso do Cartão Clonado: Com o cartão clonado em mãos, os criminosos o utilizam para realizar compras em lojas físicas, adquirir produtos online ou até mesmo para sacar dinheiro em caixas eletrônicos.
- Evasão da Detecção: Os fraudadores tentam gastar rapidamente o máximo possível do limite de crédito disponível antes que o titular do cartão perceba a fraude. Além disso, eles podem dividir as compras em pequenas transações para dificultar a identificação e detecção das operações fraudulentas.

2.2.2 Fraude Amigável

A fraude amigável segundo (MORAES, 2008), ocorre quando um titular de cartão, aparentemente legítimo, age de forma fraudulenta em relação às suas próprias transações. Nesse tipo de fraude, o titular do cartão realiza compras ou transações com o seu próprio cartão e, posteriormente, nega ter feito essas compras, alegando que o cartão foi usado sem sua autorização. Embora possa parecer paradoxal, a fraude amigável frequentemente envolve um processo intencional de desonestidade por parte do titular do cartão.

O processo da fraude amigável apresenta as seguintes etapas:

1) Compras Legítimas: Inicialmente, o titular do cartão efetua compras reais e autorizadas com seu cartão de crédito. Essas compras podem variar de bens físicos a serviços, geralmente envolvendo valores significativos.
2) Contestação das Transações: Após realizar essas compras, o titular do cartão contesta as transações junto ao emissor do cartão, alegando não as reconhecerem e que não autorizou as compras.
3) Processo de Investigação: A contestação das transações inicia um processo de investigação pela instituição que emitiu o cartão. Isso inclui uma análise detalhada da alegação de fraude amigável.

A motivação por trás da fraude amigável pode variar, mas geralmente, acontecem, porque o titular do cartão procura obter reembolsos para compras que ele realmente fez, buscando adquirir produtos ou serviços gratuitamente.

Em outros casos, algumas pessoas recorrem a esse tipo de fraude como uma maneira de enfrentar dificuldades financeiras ou pagar dívidas conforme (MORAES, 2008).

A fraude amigável impõe prejuízos financeiros ao emissor do cartão, que é obrigado a reembolsar o valor das transações contestadas, enquanto o titular do cartão retém os produtos ou serviços adquiridos.

Para combater a fraude amigável, as instituições financeiras e emissores de cartões de crédito implementam medidas rigorosas de detecção de fraudes e conduzem investigações detalhadas sobre alegações de transações não reconhecidas. Esses processos de análise de contestações geralmente envolvem uma revisão minuciosa

de evidências, como informações de localização, histórico de compras e autorizações prévias.

2.3 Visão geral dos componentes de *Big Data* utilizados na solução

Nesta sessão, será fornecida uma visão geral sobre as principais ferramentas do ecosistema de *Big Data* e ML (Machine Learning) utilizados neste projeto.

Big data refere-se a conjuntos massivos de dados que são tão grandes e complexos que as ferramentas tradicionais de processamento de dados têm dificuldade em lidar com eles de maneira eficiente. De acordo com (MARQUESONE, 2016), *Big Data* é constituído por Velocidade, Volume e Variedade.

Volume: Refere-se à enorme quantidade de dados que são gerados, coletados e processados diariamente. Em Big Data, a volumetria de dados atinge uma escala considerada alta, variando entre petabytes e exabytes. Essa massiva quantidade de dados é frequentemente gerada por diversos dispositivos, tais como sensores automotivos, registros de transações, logs de aplicações, mídias sociais e dispositivos móveis.

Velocidade: A geração e a ingestão de dados ocorrem a uma velocidade cada vez maior. Big data frequentemente pode lidar com dados em tempo real ou quase em tempo real, exigindo sistemas de processamento e análise rápidos para extrair informações úteis.

Variedade: Os dados podem ser de diferentes tipos e formatos, incluindo texto, áudio, vídeo, dados que podem ser estruturados, semiestruturados, ou não estruturados (SEGOOA; KALEMA, 2018). A variedade dos dados pode ser um desafio, pois requer a capacidade de lidar com informações de várias fontes.

Além dessas características, o *Big Data* também considera outras dimensões, como veracidade e valor dos dados.

As técnicas de *Big Data* podem ser aplicadas em diversas áreas, incluindo negócios, saúde, ciência, governo e finança, pois pode auxiliar a melhorar a tomada de decisões, otimizar processos, prever tendências e entender melhor o comportamento humano.

2.3.1 NiFi

O NiFi teve sua origem em 2006 (APACHE NIFI, 2023), quando foi concebido pela Agência Nacional de Inteligência Geoespacial dos Estados Unidos (NGA) como um projeto de código aberto denominado *Niagarafiles*. Seu propósito inicial era abordar as demandas crescentes por automação e gerenciamento eficiente de fluxos de dados em ambientes complexos e heterogêneos.

Em 2014, a *Apache Software Foundation* assumiu a responsabilidade do projeto, rebatizando-o como Apache NiFi. Desde então, o Apache NiFi tem sido divulgado e aprimorado como um projeto de código aberto pela comunidade Apache. Ao longo desse processo, transformou-se em uma ferramenta robusta e flexível para a integração de dados em tempo real, adaptando-se a diversos cenários, inclusive em ambientes de *Big Data*.

Destaca-se por algumas características que o tornam uma escolha eficiente para automatizar o fluxo de dados entre sistemas heterogêneos (APACHE NIFI, 2023). Além disso, possui uma interface gráfica intuitiva que permite aos usuários desenhar e visualizar o fluxo de dados, simplificando a configuração, gestão e monitoramento em tempo real.

Entre suas principais características, destacam-se:

1. Coleta e Distribuição de Dados: Facilita a movimentação de dados entre sistemas, sendo especialmente valioso em ambientes com diversas fontes e destinos de dados.
2. Gerenciamento de Fluxo de Dados em Tempo Real: Oferece a capacidade de gerenciar e rotear dados em tempo real, sendo essencial para situações em que a latência e a velocidade são críticas.

3. Extensibilidade e Integração: Altamente extensível, permite a integração com uma variedade de tecnologias e serviços, suportando diversos processadores para transformação, enriquecimento e roteamento de dados.
4. Segurança: Proporciona recursos robustos de segurança, incluindo autenticação, autorização e criptografia, garantindo a proteção dos dados em movimento.
5. Gestão de Fluxo e Monitoramento: Apresenta recursos avançados de monitoramento, possibilitando a visualização do desempenho do fluxo de dados, identificação de gargalos e monitoramento geral do sistema.
6. Fluxo Direcionado Baseado em Regras: Permite a configuração de regras para direcionar fluxos de dados com base em condições específicas, garantindo flexibilidade e automação.
7. Processamento de Eventos Complexos: Integra capacidades de processamento de eventos complexos para a detecção de padrões em tempo real nos fluxos de dados.

O NiFi é amplamente utilizado em diversos cenários, desde integração de ambientes corporativos até aplicações específicas como Internet das Coisas (IoT), análise de logs e ingestão de dados em *Big* Data (APACHE NIFI, 2023). Sua flexibilidade, facilidade de uso, tornando uma ferramenta valiosa para arquiteturas de dados distribuídas e complexas, atendendo às demandas variadas do mundo contemporâneo.

2.3.2 Apache Hadoop

O Apache Hadoop é uma plataforma de código aberto voltada para Big Data que possibilita o armazenamento e o processamento distribuído de extensos conjuntos de dados. Sua versatilidade permite lidar com diversos tipos de dados, abrangendo desde estruturados e semiestruturados até não estruturados. Projetado com a proposta de escalabilidade e tolerância a falhas, o Hadoop opera de maneira eficiente em ambientes que vão desde servidores únicos até complexas configurações

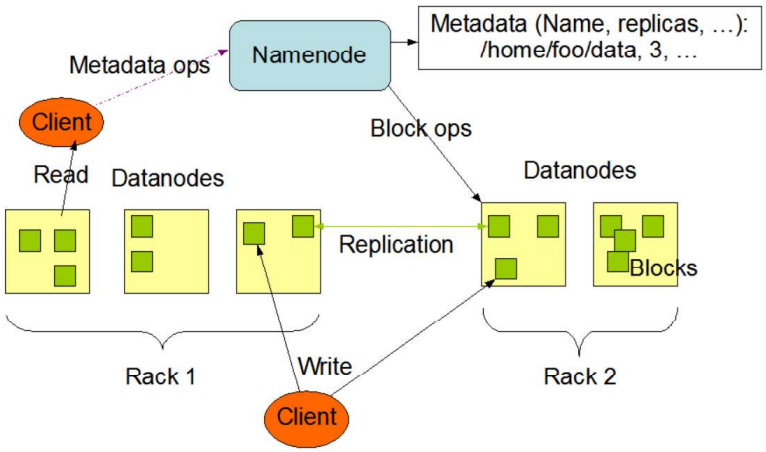
distribuídas envolvendo milhares de máquinas (PARMAR et al., 2018). Essa flexibilidade torna o Hadoop uma solução robusta para organizações que buscam gerenciar e processar grandes volumes de dados de maneira eficaz e resiliente.

O Hadoop é composto pelo framework MapReduce que é utilizado para processamento distribuído e pelo sistema de armazenamento de arquivos *Hadoop Distributed File System* (HDFS).

O HDFS tem sido uma parte central e essencial do ecossistema Hadoop, fornecendo uma solução de armazenamento distribuído para processamento de dados em larga escala, pois trata-se de um sistema de arquivos distribuídos, escalável e tolerante a falhas. Sua arquitetura é fundamentada em clusters e emprega o conceito de mestre e escravo.

A Figura 1 ilustra o servidor mestre, conhecido como *NameNode*, que tem como função gerenciar o acesso dos clientes e os metadados do sistema de arquivos. Os demais nós escravos, denominados *Datanodes*, são encarregados de armazenar os dados e atender às solicitações de leitura e gravação dos clientes.

Figura 1 – Arquitetura HDFS.



Fonte: reprodução (APACHE HADOOP, 2022)

2.3.3 MLlib

A MLlib, ou *Machine Learning Library*, é uma biblioteca de *Machine Learning* de código aberto que iniciou o seu desenvolvimento em 2012 e atualmente faz parte do projeto Apache Spark (MLLIB, 2023).

O Apache Spark é um framework de processamento de dados em larga escala que oferece suporte a computação distribuída. A MLlib fornece uma variedade de algoritmos de aprendizado de máquina e ferramentas para realizar tarefas como classificação, regressão, clustering e recomendação.

A biblioteca inclui algoritmos para processamento de dados estruturados e não estruturados, bem como ferramentas para pré-processamento e avaliação de modelos de *Machine Learning*. Sua integração com o ecossistema Spark torna a MLlib uma escolha popular para aplicações de *Big Data* e análise distribuída.

Entre os algoritmos disponibilizados na biblioteca MLlib, destacam-se os seguintes:

1. Algoritmo de classificação: Regressão Logística, Máquinas de Vetores de Suporte (SVM), Árvores de Decisão, *Random Forests* e *Gradient-Boosted Trees;*
2. Algoritmo de Regressão: Regressão Linear, Regressão de Mínimos Quadrados Generalizados (GLM), *Random Forest Regressor e Gradient-Boosted Trees Regressor;*
3. Algoritmos de Clustering: *K-Means, Gaussian Mixture Model (GMM), Latent Dirichlet Allocation (LDA).*

Além disso, a MLlib possibilita utilizar códigos Python através do PySpark (APACHE SPARK, 2023), dessa forma, torna-se uma ferramenta que atenda as programações dos algoritmos proposto neste projeto.

2.3.4 Python

De acordo com (NAGPAL; GABRINI, 2019), Python é considerado uma linguagem com um desempenho satisfatório, e de fácil entendimento, com código aberto que pode hospedar milhares de módulos de terceiros. Tanto as bibliotecas padrões do Python quanto os módulos contribuídos pela comunidade permitem infinitas possibilidades para seu uso.

Python é multiplataforma, o que significa que os programas escritos em Python podem ser executados em diferentes sistemas operacionais sem a necessidade de modificação. Essa portabilidade é particularmente valiosa em ambientes de desenvolvimento e implantação diversificados (PYTHON, 2023).

Com a ascensão de tecnologias emergentes, como inteligência artificial e análise de dados, Python se tornou uma escolha popular entre os cientistas de dados e engenheiros que os utilizam para desenvolver modelos de *Machine Learning* e redes neurais (PYTHON, 2023).

2.3.5 Balanceamento de classes

A desigualdade na distribuição das classes pode resultar em dificuldades para modelos de aprendizado computacional aprender corretamente as classes minoritárias, já que tende a favorecer a classe majoritária devido à quantidade maior de exemplos disponíveis para ela.

Uma vez que o desbalanceamento de classes pode prejudicar o desempenho dos algoritmos, foram desenvolvidas técnicas de balanceamento de classes para lidar com esses problemas.

Existem duas abordagens principais, sendo elas o *Oversampling*, que é uma técnica de aumento de amostragem, no qual é realizado o aumento da classe minoritária para se equiparar a classe majoritária. A outra abordagem é o *Undersampling* que ao

contrário do *Oversampling*, reduz a quantidade de exemplos da classe majoritária para equilibrar a distribuição das classes.

Segundo (HE; 2013), que abrange sobre conjuntos de dados desbalanceados e técnicas para lidar com esses desequilibrios de classes, os autores explicam que *SMOTE*, abreviação de *Synthetic Minority Over-sampling Technique* (Técnica de Aumento de Amostragem Sintética para Classes Minoritárias) é uma técnica avançada de *oversampling*, amplamente utilizada em uma variedade de algoritmos, pois enquanto o *oversampling* simplesmente replica exemplos existentes da classe minoritária, o SMOTE cria novos exemplos sintéticos da classe minoritária.

Esses exemplos ajudam a ampliar a representação da classe menos comum, melhorando a capacidade do algoritmo de aprender sobre as classes de forma mais equilibrada.

2.3.6 Tableau

O Tableau tem a capacidade de se conectar a uma ampla variedade de fontes de dados, como bancos de dados, planilhas e serviços (TABLEAU, 2023). Isso permite a integração com fontes de dados de Big Data, como Hadoop, Spark e outras tecnologias, permitindo análises avançadas em grandes conjuntos de dados.

Atualmente é uma ferramenta amplamente utilizada em organizações de diferentes setores para transformar dados em *insights* visuais significativos (TABLEAU, 2023), facilitando a tomada de decisões. Sua abordagem intuitiva a torna uma ferramenta popular para profissionais de análise de dados e negócios.

Através da utilização dessa ferramenta, os usuários, podem elaborar *Dashboar*ds, compartilhar relatórios entre a equipe, montar eventuais alarmes de aviso para um determinado grupo de pessoas.

2.4 Arquitetura NIST para Big Data

O *National Institute of Standards and Technology* (NIST), conhecido em português como Instituto Nacional de Padrões e Tecnologia, foi fundado nos Estados Unidos em 1901 com o objetivo de impulsionar a inovação tecnológica e aprimorar a competitividade industrial do país.

Para conduzir medições precisas e estabelecer padrões de referência, o NIST mantém colaborações com diversas organizações industriais, acadêmicas e agências governamentais ao redor do mundo. Essa colaboração internacional permite que o sistema de medições do NIST opere em uma escala global (NIST, 2022).

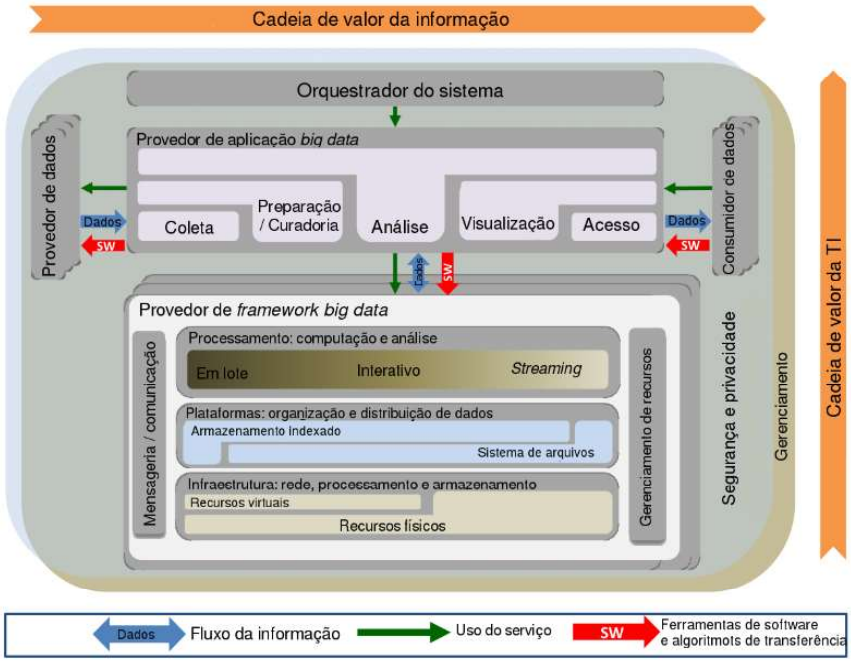
Com o intuito de impulsionar o avanço do *Big Data*, o NIST estabeleceu em 2013 o Grupo de Trabalho Público de Big Data (*NIST Big Data Public Working Group - NBD-PWG*), que contou com a participação abrangente de indústrias, universidades e órgãos governamentais (NBD-PWG, 2015).

Os resultados das atividades do NBD-PWG estão documentados em uma série de sete volumes denominada Estrutura de Interoperabilidade de *Big Data* do NIST. Cada volume se concentra em um tópico específico essencial, que inclui definições, taxonomia, casos de uso e requisitos gerais, segurança e privacidade, revisões de arquiteturas, arquitetura de referência e padrões a serem seguidos.

O NBD-PWG propõe um modelo conceitual de arquitetura de referência para o ambiente de *Big Data*, que é independente de fornecedor, tecnologia e infraestrutura. Esse modelo conceitual, conforme apresentado na Figura 2, descreve um sistema de *Big Data* composto pelos seguintes componentes: o orquestrador de sistema; o provedor de dados; o provedor de aplicações de Big Data; o provedor de estruturas de *Big Data;* e o consumidor de dados.

Os componentes se interconectam por meio de interfaces de interoperabilidade e são envolvidos por duas camadas que abrangem os requisitos de segurança, privacidade e gerenciamento (NBD-PWG, 2015).

Figura 2 – Arquitetura NIST.



Fonte:reprodução (NBD-PWG, 2015)

Conforme ilustrado na Figura 2, a arquitetura de referência do NIST é composta por dois eixos que representam as cadeias de valor no contexto do Big Data. O eixo horizontal representa a cadeia de valor da informação, englobando a coleta, preparação, integração, análise de dados e apresentação de resultados. No eixo vertical, temos a cadeia de valor da Tecnologia da Informação (TI), que fornece os recursos de infraestrutura de rede, plataformas de armazenamento e ferramentas para o processamento, organização e distribuição dos dados (NBD-PWG, 2015).

Dentro do diagrama da Figura 2, a seta com a palavra DADOS representa o movimento da informação entre os elementos funcionais do sistema de Big Data. A seta marcada como a sigla SW representa a realização de algoritmos de transferência e o emprego de ferramentas de software para o processamento dos dados. A seta denominada Uso do serviço indica a utilização de interfaces de software programáveis (NBD-PWG, 2015).

2.5 Algoritmos de Machine Learning

Machine Learning pode ser definida como a área, através da qual se faz pesquisa, estudos e se define um conjunto de técnicas para, automaticamente, detectar padrões em dados e utilizar esses padrões descobertos para predizer acontecimentos futuros, ou até mesmo, para realizar outros tipos de tomada de decisão relacionados a eventos não determinísticos, tipicamente os problemas tratados pela área são: classificação, regressão e agrupamento (OLIVEIRA, 2016).

Nos últimos anos, o *Machine Learning* emergiu como uma das tecnologias mais promissoras e impactantes do nosso tempo. Através da sua utilização, sistemas de computadores têm a capacidade de aprender e aprimorar seu desempenho automaticamente a partir de dados, sem a necessidade de programação explícita. Essa revolução na capacidade de processamento e análise de dados está redefinindo a maneira como enfrentamos desafios em áreas tão diversas como medicina, finanças, transporte, entretenimento e muito mais.

Para o desenvolvimento deste projeto, foram utilizados os algoritmos de *Machine Learning* (ML) abaixo descritos.

2.5.1 Naive Bayes

Naive Bayes, pode ser considerado como um método de classificação baseado no Teorema de Bayes, para calcular a probabilidade de um evento pertencer a uma determinada classe, dado um conjunto de características ou atributos.

O *Naive Bayes* é um algoritmo que pode ser utilizado em tarefas de classificação de spam de e-mails, diagnóstico médico, detecção de sentimentos em análise de texto e detecção de fraudes, devido à sua simplicidade e eficácia. Segundo (VAIRAM et al., 2020), o algoritmo é frequentemente usado para fazer previsões em tempo real, tornando-o mais adequado para detecção de fraudes em cartão de crédito.

O algoritmo é chamado de Naive devido a uma suposição simplificada de independência entre as características usadas na classificação. Segundo (IZBICKI, 2022), o método é especialmente útil em casos em que é necessário lidar com variáveis faltantes, uma vez que as redes modelam a distribuição conjunta.

O Teorema de Bayes é uma fórmula que descreve como calcular a probabilidade de um evento com base em informações prévias. A fórmula é a seguinte:

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

- P(A|B) é a probabilidade condicional de A dado B.
- P(B|A) é a probabilidade condicional de B dado A.
- P(A) é a probabilidade de A ocorrer.
- P(B) é a probabilidade de B ocorrer.

Neste projeto, o algoritmo de *machine learning* Naive Bayes, foi utilizado para identificar transações com suspeitas de fraudes, no qual as transações escolhidas foram as transações realizadas várias vezes com o mesmo cartão de crédito em um curto espaço de tempo, ou transações realizadas com o mesmo cartão de crédito, porém em lugares geográficos diferentes.

2.5.2 Random Forest

Random Forest, é um método de regressão e classificação, que consiste em um conjunto extenso de árvores individuais de decisão que operam como um conjunto.

Cada árvore individual dentro desse conjunto de árvores, gera uma predição de resultado e ao final a classe com a contagem mais alta de votos (no caso de classificações) ou a média dos resultados das árvores (no caso de regressão), torna-se o resultado de predição do modelo.

Segundo (BREIMAN, 2001), *Random Forest* trata-se de uma combinação de árvores de decisão, em que cada árvore depende dos valores de um vetor aleatório.

O processo envolve a amostragem aleatória de dados e a seleção aleatória de recursos, tornando as árvores diversificadas.

O *Random Forest* utiliza a votação majoritária como classificação ou a média com regressão das previsões das árvores individuais para chegar na previsão final. Trata-se de um algoritmo que pode ser utilizado em uma variedade de problemas de aprendizado de máquina, incluindo grandes conjuntos de dados

Neste projeto, o algoritmo de *Machine Learning* (ML) *Random Forest* foi utilizado para foi empregado na identificação de transações suspeitas de fraude. Foram selecionadas aquelas que ocorreram repetidamente com o mesmo cartão de crédito em um curto intervalo de tempo, bem como aquelas realizadas com o mesmo cartão de crédito, porém em localidades geográficas distintas.

2.5.3 K-NN

O *K-Nearest Neighbors* (K-NN), ou “Vizinhos mais Próximos”, é um algoritmo de aprendizado de máquina utilizado tanto para classificação quanto para regressão. De acordo com (MALINI; PUSHPA, 2017), o K-NN é um algoritmo de ML amplamente utilizado para detectar fraudes em cartões de crédito, os resultados do K-NN são dependentes dos seguintes fatores:

- A métrica de distância usada para decidir os vizinhos mais próximos;
- O número de vizinhos considerados para classificar a nova amostra;
- A regra de distância usada para a classificação do K-vizinho mais próximo.

Segundo (SINGHAI et al., 2023), o K-NN pode melhorar a acurácia nos processos desenvolvidos para detecção de transações de créditos consideradas fraudulentas além disso, é um algoritmo que permite ser combinado com outros algoritmos com o intuito de melhorar o desempenho nas detecções de fraude.

Em um contexto de classificação, quando um novo ponto de dados precisa ser classificado, o K-NN avalia os K pontos mais próximos a esse ponto no espaço de características. A classe predominante entre esses K vizinhos é atribuída ao ponto em questão.

A escolha de K é um parâmetro ajustável que influencia a sensibilidade do modelo à variação nos dados. Para problemas de regressão, o K-NN calcula a média (ou mediana) dos valores-alvo dos K vizinhos mais próximos para prever o valor de um novo ponto.

A principal ideia por trás do K-NN é que pontos semelhantes tendem a possuir rótulos ou valores semelhantes. Esse método é intuitivo e fácil de entender, mas sua eficácia pode depender da escolha adequada de K e da sensibilidade a outliers. Além disso, a performance do K-NN pode ser afetada pela dimensionalidade dos dados, sendo mais eficaz em conjuntos de dados com menor número de características.

Assim como os demais algoritmos, isto é, utilizado para identificar as transações que se repetem com o mesmo cartão de crédito em um curto intervalo de tempo, assim como aquelas realizadas com o mesmo cartão em diferentes locais geográficos.

2.6 Plataforma Kaggle

Kaggle é uma plataforma utilizada e reconhecida por cientistas de dados, entusiastas da análise de dados e profissionais de aprendizado de máquina. Fundada em 2010 (KAGGLE, 2019), ela se estabeleceu como um ponto central na comunidade global de cientistas de dados, oferecendo um conjunto diversificado de recursos, incluindo competições desafiadoras, conjuntos de dados, cursos educacionais e ferramentas essenciais para capacitar tanto profissionais experientes quanto iniciantes a aprimorar suas habilidades e colaborar em projetos de dados de classe mundial.

Trata-se de uma plataforma online que disponibiliza um vasto repositório de conjuntos de dados públicos abrangendo uma ampla variedade de tópicos. Esses conjuntos de dados são recursos valiosos que podem ser explorados para fins de prática,

desenvolvimento de modelos e realização de pesquisas. Além disso, os usuários têm a capacidade de contribuir com seus próprios conjuntos de dados para enriquecer ainda mais a comunidade de ciência de dados na plataforma.

Neste projeto, a plataforma Kaggle foi utilizada para encontrar um conjunto de dados que contivessem informações simulando transações de cartões de créditos físicos, dessa forma, pretendeu-se chegar o mais próximo possível das informações que uma transação de crédito é gerada em um PDV.

3 DESENVOLVIMENTO

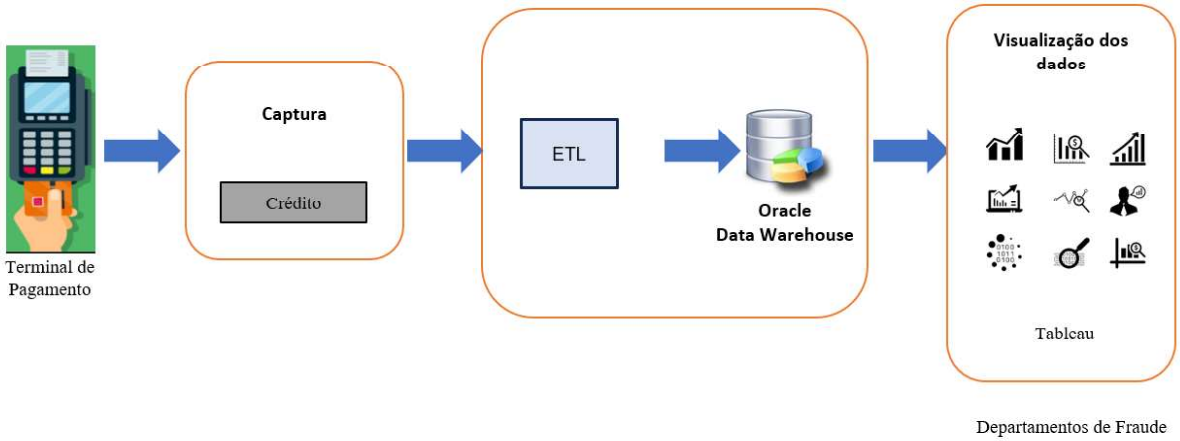
Este capítulo apresenta uma visão geral da solução de Big Data desenvolvida para a identificação e detecção de transações de cartões de créditu suspeitas de fraude. Inicialmente, será apresentado um exemplo de solução baseada em DW. Em seguida, será apresentada a solução de Big Data desenvolvida neste projeto, seguindo as normas da arquitetura NIST, e detalhando os processos desde a camada de ingestão até a camada de visualização e noticações.

3.1 Exemplo de solução tradicional baseada em DW e seus desafios

O projeto iniciou-se com a análise da solução atual, baseada em DW, para detecção de transações suspeitas de fraudes. Ao analisar a solução utilizada por algumas áreas de fraudes, encontrou-se as seguintes limitações:

- As informações das transações realizadas nos Pontos de Vendas, conforme demonstrado na Figura 3, são armazenadas na etapa de captura e a partir da 0:00hs de cada dia, são enviadas para serem processadas via ETL que em virtude do alto volume de informações e limitações de *Hardware,* o processamento e o armazenamento dos dados levam em torno de 24 horas para ficar disponível em um DW. Com isso as áreas de negócios só conseguem acessar as informações com um atraso considerável em relação aos acontecimentos, pois são informações do dia anterior (D-1).

Figura 3 – Solução utilizando DW para detecção de fraudes.



Fonte: O autor

Devido à restrição de acesso apenas às informações do dia anterior (D-1), as equipes de prevenção de fraudes estão limitadas a análises com dados desatualizados. Os dashboards não são atualizados automaticamente, resultando em informações obsoletas. Esse problema decorre do fato de que as equipes de prevenção de fraudes recebem os dados com um dia de atraso, o que não reflete as informações mais recentes.

Essas limitações trazem os seguintes impactos:

- Prejuízo financeiro, pois as transações fraudulentas só foram identificadas muito depois do ocorrido.
- No planejamento para verificar e atuar nos locais que apresentam maiores recorrências de transações com características de fraudes.

Ao analisar a solução atual de detecção de fraudes utilizando dados armazenados em um *Data Warehouse*, foram identificados os seguintes pontos de melhorias:

- Capturar e Analisar as transações de cartão de crédito o mais próximo do tempo real.
- Melhorar os algoritmos de detecção de fraudes, considerando geolocalização e várias tentativas de várias compras com valores pequenos e em curto espaço de tempo entre as compras.

- Centralizar o processo de identificação de transações que apresentam características de fraudes.
- Aumentar a velocidade na identificação das transações que possam ser consideradas fraudulentas.
- Facilitar a atualização de *Dashboard* e alarmes para as equipes de fraudes.

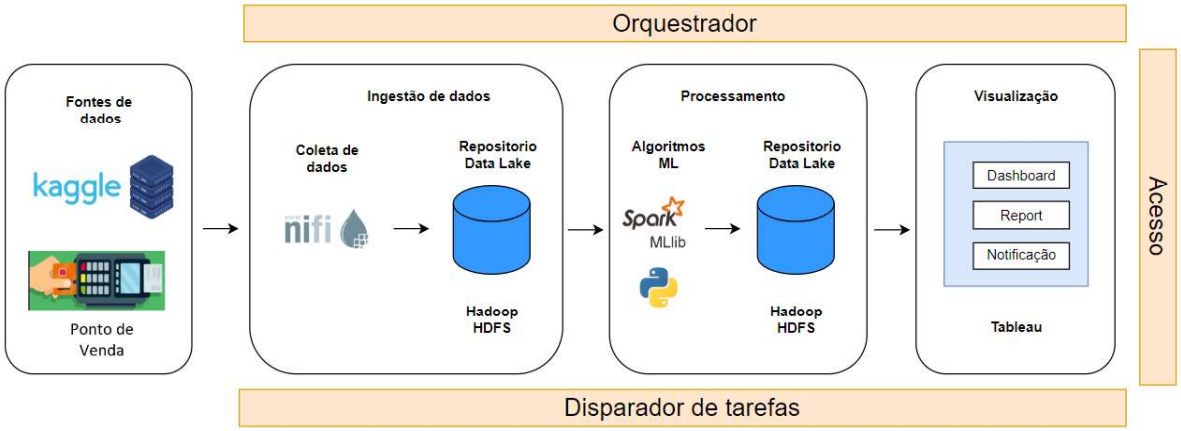
3.2 Solução proposta utilizando Big Data

A solução proposta foi baseada na arquitetura de *Big Data* NIST (NBD-PWG, 2015), com foco nas necessidades específicas para atender o projeto, essas necessidades foram:

1. Capacidade para processar dados estruturados, ou semiestruturados na mesma plataforma;
2. Necessidade de uma camada de armazenamento após os ciclos de processamento;
3. Necessidade de armazenar grandes volumes de dados.

Dessa forma, a Figura 4, representa resumidamente a estrutura das cinco camadas, as quais são: Fontes de dados, Ingestão de dados, Processamento, Bases de Consulta e Visualização.

Figura 4 – Arquitetura da solução.



Fonte: Próprio Autor

3.3 Fontes de dados

As Fontes de dados são provenientes das transações realizadas nos Pontos de Vendas (PDV) utilizando como meio de pagamento, cartões de créditos físicos, pois cada transação realizada no PDV gerará informações únicas.

Para essa solução, a priori, utilizou-se um conjunto de dados com informações de simulam transações realizadas com o uso de cartões de crédito semelhantes as informações geradas pelos PDV. Esse conjunto de dados foi gerado usando a ferramenta de geração de dados Sparkov (JOSE et al., 2023), . A captura de dados tem como objetivo simular os recebimentos das transações o mais próximo do tempo que elas ocorreram e foi executada com dados capturados entre janeiro de 2019 a dezembro de 2020, disponíveis na plataforma Kaggle (KAGGLE, 2019).

O conjunto de dados com as informações possui 1.852.394 registros, com tamanho de aproximadamente 350 MB e em formato .csv, contendo informações das transações realizadas com cartões de créditos como data e horário da transação, número fictício do cartão, assim como a geolocalização de cada transação em diversos comércios como restaurantes, shooping e mercados.

A figura 5, ilustra uma parcela das informações disponíveis no arquivo de transação.

Figura 5 – Dados brutos

```
WILLIAN > 2020 > POS_USP > 2022 > 2023 > MONOGRAFIA > BASE_DADOS > credicard.csv
trans_date_trans_time,cc_num,merchant,category,amt,first,last,gender,street,city,state,zip,lat,long,city_pop,job,dob,trans_num
0,2019-01-01 00:00:18,2703186189652095,"Rippin, Kub and Mann",misc_net,4.97,Jennifer,Banks,F,561 Perry Cove,Moravian Falls,NC,
1,2019-01-01 00:00:44,630423337322,"Heller, Gutmann and Zieme",grocery_pos,107.23,Stephanie,Gill,F,43039 Riley Greens Suite 39
2,2019-01-01 00:00:51,38859492057661,Lind-Buckridge,entertainment,220.11,Edward,Sanchez,M,594 White Dale Suite 530,Malad City,
3,2019-01-01 00:01:16,3534093764340240,"Kutch, Hermiston and Farrell",gas transport,45.0,Jeremy,White,M,9443 Cynthia Court Apt
4,2019-01-01 00:03:06,375534208663984,Keeling-Crist,misc_pos,41.96,Tyler,Garcia,M,408 Bradley Rest,Doe Hill,VA,24433,38.4207,-
5,2019-01-01 00:04:08,4767265376804500,"Stroman, Hudson and Erdman",gas_transport,94.63,Jennifer,Conner,F,4655 David Island,Du
6,2019-01-01 00:04:42,30074693890476,Rowe-Vandervort,grocery_net,44.54,Kelsey,Richards,F,889 Sarah Station Suite 624,Holcomb,K
7,2019-01-01 00:05:08,6011360759745864,Corwin-Collins,gas_transport,71.65,Steven,Williams,M,231 Flores Pass Suite 720,Edinburg
8,2019-01-01 00:05:18,4922710831011201,Herzog Ltd,misc_pos,4.27,Heather,Chase,F,6888 Hicks Stream Suite 954,Manor,PA,15665,40.
9,2019-01-01 00:06:01,2720830304681674,"Schoen, Kuphal and Nitzsche",grocery_pos,198.39,Melissa,Aguilar,F,21326 Taylor Squares
10,2019-01-01 00:06:23,4642894980163,Rutherford-Mertz,grocery_pos,24.74,Eddie,Mendez,M,1831 Faith View Suite 653,Clarinda,IA,5
11,2019-01-01 00:06:53,377234009633447,Kerluke-Abshire,shopping_net,7.77,Theresa,Blackwell,F,43576 Kristina Islands,Shenandoah
12,2019-01-01 00:06:56,180042946491150,Lockman Ltd,grocery_pos,71.22,Charles,Robles,M,3337 Lisa Divide,Saint Petersburg,FL,337
13,2019-01-01 00:07:27,5559857416065248,Kiehn Inc,grocery_pos,96.29,Jack,Hill,M,5916 Susan Bridge Apt. 939,Grenada,CA,96038,41
14,2019-01-01 00:09:03,3514865930894695,Beier-Hyatt,shopping_pos,7.77,Christopher,Castaneda,M,1632 Cohen Drive Suite 639,High
15,2019-01-01 00:09:20,6011999600625827,Schmidt and Sons,shopping_net,3.26,Ronald,Carson,M,870 Rocha Drive,Harrington Park,NJ,
16,2019-01-01 00:10:49,6011860238257910,Lebsack and Sons,misc_net,327.0,Lisa,Mendez,F,44259 Beth Station Suite 215,Lahoma,OK,7
17,2019-01-01 00:10:58,3565423334076143,Mayert Group,shopping_pos,341.67,Nathan,Thomas,M,4923 Campbell Pines Suite 717,Carlisl
```

Fonte: Próprio Autor

3.4 Camada de ingestão dos Dados

Nesta fase, realiza-se a coleta, limpeza, tratamento e transferência dos dados provenientes de fontes externas.

3.4.1 Coleta de dados

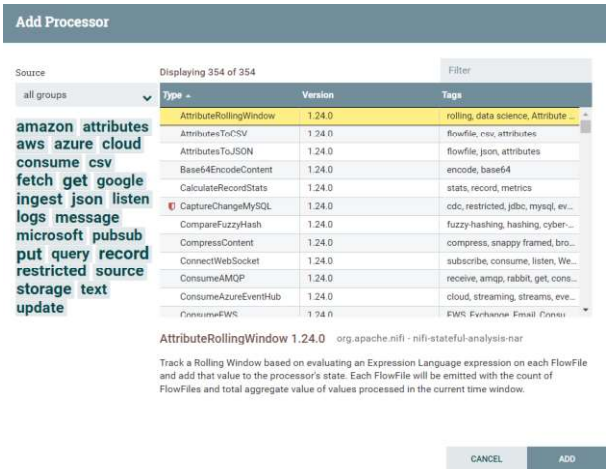
Para automatizar a coleta dos dados foi utilizada a ferramenta NiFi (Apache NiFi, 2023), dessa forma, o usuário poderá configurar o intervalo da coleta dos dados de acordo com suas necessidades. A Figura 6, representa o fluxo, no qual os dados coletados serão armazenados no HDFS.

Figura 6 – ingestão de dados

Fonte: Próprio Autor

Ao criar um fluxo de processamento no NiFi, a ferramenta oferece a opção para configurar a extensão e tipo de arquivo que pode ser de texto, ou de *log*, conforme ilustrado na Figura 7 a escolha será de acordo com necessidade do projeto.

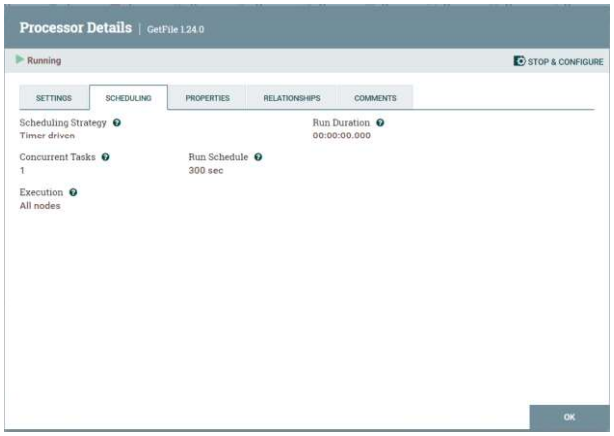
Figura 7 –Interface para configuração coleta dos dados



Fonte: Próprio Autor

A ferramenta também oferece a funcionalidade de agendamento, permitindo a programação do intervalo de coleta de dados. O agendamento é realizado em segundos, sendo necessário converter o tempo desejado para essa unidade. Neste projeto, o agendamento foi configurado para ser executado a cada 5 minutos, equivalente a 300 segundos, como indicado na Figura 8.

Figura 8 – Agendar coleta de dados

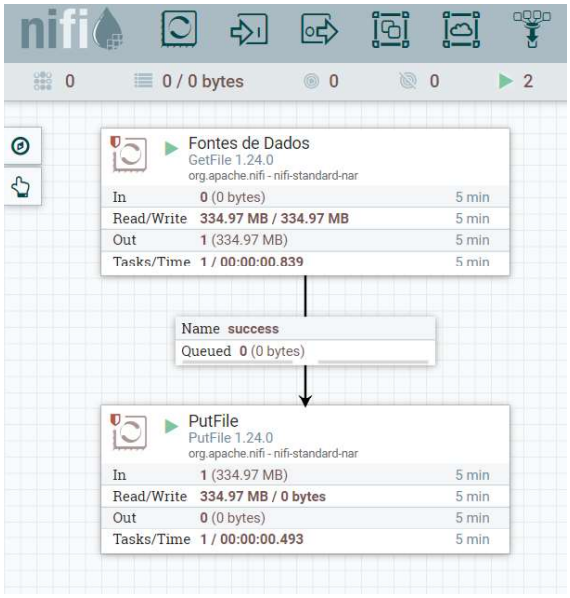


Fonte: Próprio Autor

A Figura 9 ilustra o processo configurado para coletar os dados, recepcionando as informações das transações de crédito e direcionando os arquivos para o diretório em que serão armazenados. Neste exemplo, há a recepção do arquivo *.csv com 350 MB que está disponível no Kaggle.

O processo de coleta de dados utilizará dois Processors, o *Getfile* e o *PutFile*. O *Getifile* é responsável por coletar o arquivo que será enviado para o destino, e o *PutFile*, permite especificar o local de armazenamento. A Figura 9, representa o processo de coleta de dados.

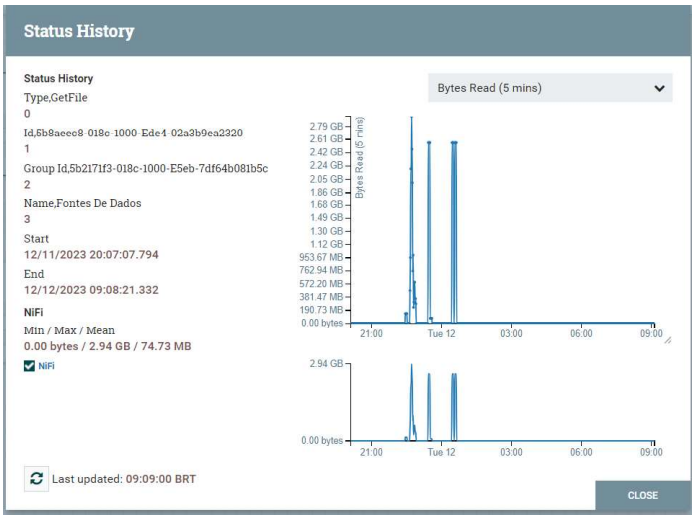
Figura 9 – Processo para coleta de dados



Fonte: Próprio Autor

A ferramenta também disponibiliza a opção de visualizar o histórico de processamento por meio de gráficos de linhas, que apresentam a quantidade de *bytes* processados. NA figura 10 é apresentado um exemplo de histórico de processamento de dados.

Figura 10 – histórico e coleta de dados



Fonte: Próprio Autor

3.4.2 Limpeza, Normalização e Validação dos dados

Frequentemente, os dados coletados precisam passar por processos de Limpeza, Normalização e Validação. Essas etapas envolvem a identificação e correção de valores ausentes, inconsistências ou erros nos dados. A padronização e normalização dos dados também podem ser necessárias para garantir que eles se encaixem em um formato consistente e sejam comparáveis. O mascaramento de dados é um componente importante, especialmente quando se trata de informações confidenciais ou sensíveis. O mascaramento é uma prática para embaralhar ou ocultar informações que podem identificar indivíduos ou expor dados sensíveis, garantindo, assim, a privacidade e a conformidade com regulamentos de proteção de dados.

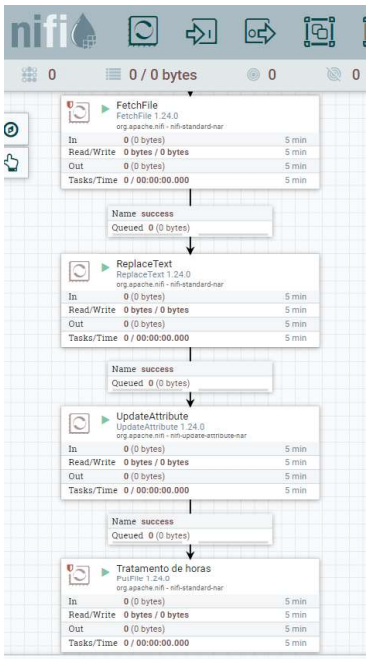
Conforme representado na Figura 11, os tratamentos serão conduzidos pela ferramenta NiFi (Apache NiFi, 2023), pois além de realizar coletas de dados sincronizadas a ferramenta também oferece algumas funcionalidades de Limpeza, Normalização e Validação. Após esse tratamento, os dados são direcionados para um diretório exclusivo para receber apenas informações provenientes de transações. Ao utilizar o NiFi nessa etapa, evita-se a necessidade de criar novos fluxos para a integração com outras ferramentas do Ecossistema Hadoop. Com isso, espera-se reduzir o tempo de eventuais manutenções futuras nesta etapa.

Figura 11 – Tratamento dos dados

Fonte: Próprio Autor

Para assegurar a qualidade da normalização dos dados, foram montados processos no NiFi dedicados a essa finalidade. Na Figura 12, é apresentado o processo responsável por realizar diversas normalizações, como, por exemplo, garantir que os campos de data e hora sigam um padrão consistente. Dessa forma, após a normalização, os campos de data e hora são uniformizados conforme o padrão estabelecido

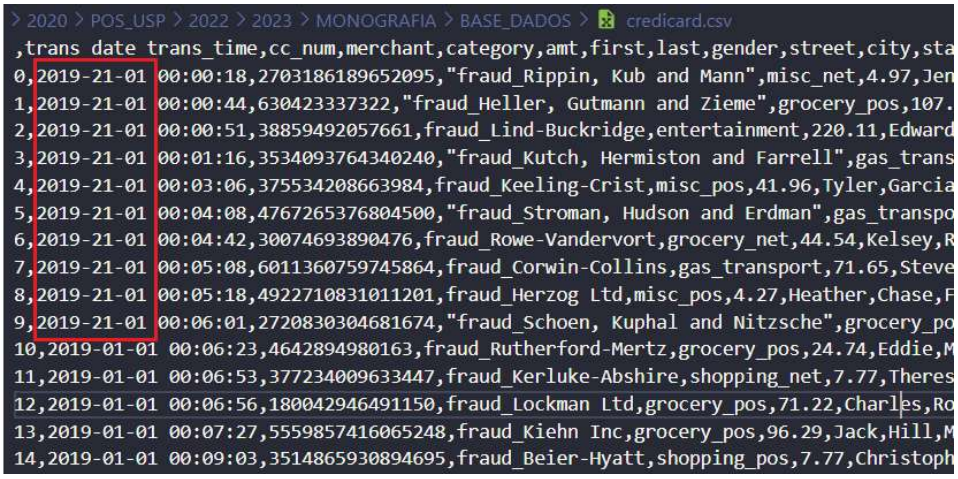
Figura 12 – Normalização de dados via NiFi



Fonte: Próprio Autor

A Figura 13, destaca um campo de data antes do processo de normalização. Neste exemplo, o foco é garantir que os campos de data e hora sigam um padrão consistente. As datas destacadas representam exemplos que precisaram passar pelo processo de ser normalizadas.

Figura 13 – Datas com padrões diferentes.

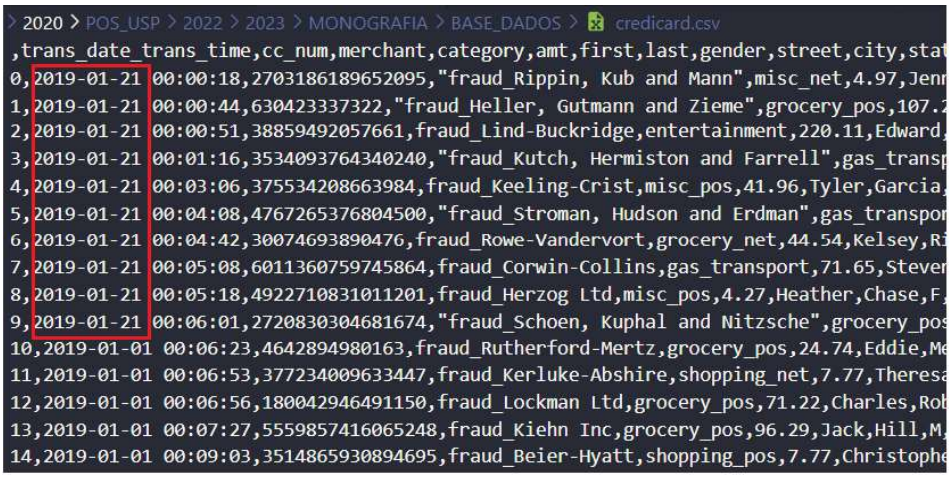


Fonte: Próprio Autor

A Figura 13 apresenta uma coluna de datas destacada no formato YYYY-DD-MM, onde YYYY representa o ano, DD o dia e MM o mês. Esses campos destacados diferem das demais datas que seguem o formato YYYY-MM-DD.

A Figura 14 mostra o campo após o processo de normalização, onde a coluna que contém informações de data agora está uniformizada como YYYY-MM-DD, assegurando uma única formatação em todos os registros.

Figura 14 – Normalização de Dados: Data dos Eventos



Fonte: Próprio Autor

A validação realizada nesta etapa, tem como objetivo verificar se há datas com valores incorretos, ou diferente do padrão escolhido para o trabalho. A Figura 15, apresenta o *script* para validações de data.

Figura 15 – Validação de datas

Verifica se há datas incorretas.

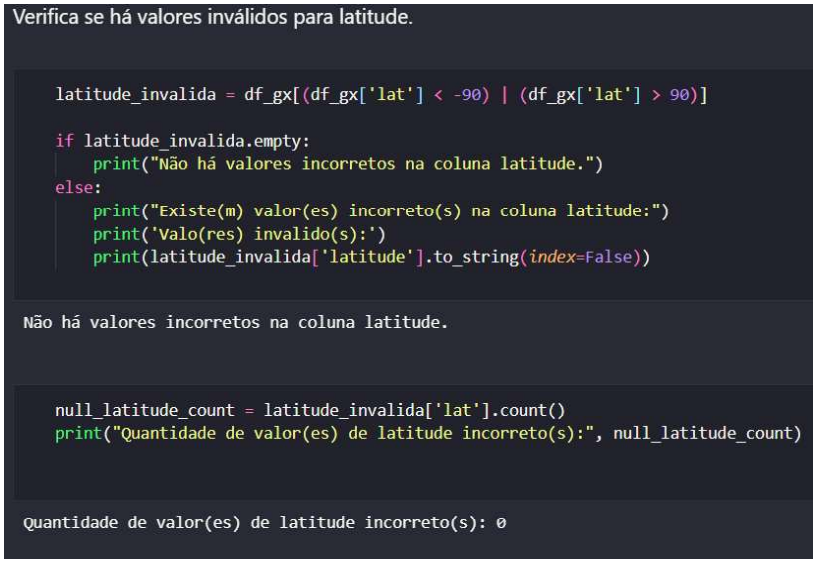
```
from datetime import datetime

data_invalida = []

for valor_data in df_gx['date']:
    try:
        datetime.strptime(valor_data, '%Y-%m-%d')
    except ValueError:
        data_invalida.append(valor_data)

if len(data_invalida) == 0:
    print("Não há valores incorretos na coluna de datas.")
else:
    # Criar DataFrame com os valores incorretos
    df_data_invalida = pd.DataFrame({'Data(s) Incorreta(s):': data_invalida})
    print("Existe(m) valor(es) incorreto(s) na coluna de datas:")
    print(df_data_invalida.to_string(index=False))
```

```
Não há valores incorretos na coluna de datas.
```

Fonte: Próprio Autor

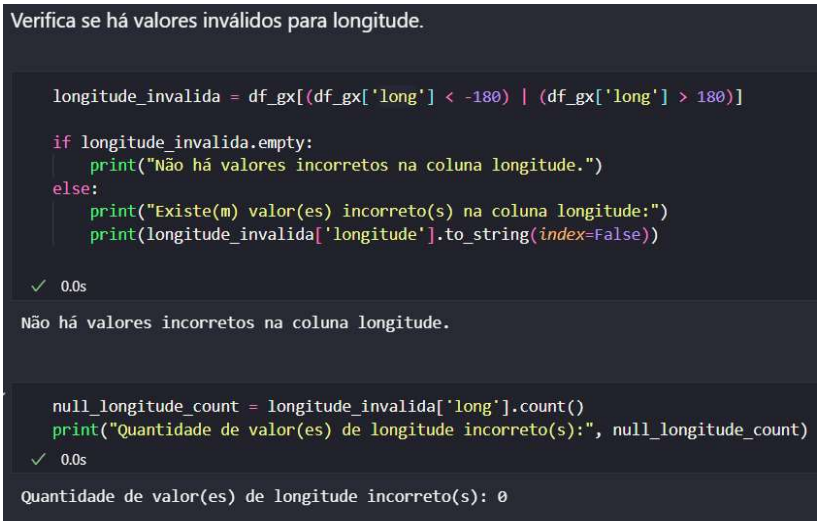
Outra validação realizada nesta etapa, consiste em garantir que a latitude e longitude (geolocalização) estejam consistentes . A Figura 16, apresenta o *script* para validações da latitude e a Figura 17 para longitude.

Figura 16 – Validação de latitude



Fonte: Próprio Auto

Figura 17 – Validação de longitude

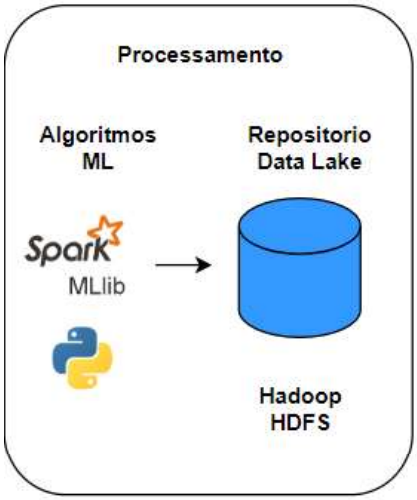


Fonte: Próprio Autor

3.5 Camada de processamento

Nesta etapa, será realizada a preparação dos dados para o aprendizado de *Machine Learnig* e depois processados por *scritps* desenvolvidos utilizando as bibliotecas da ferramenta MLlib (MLLIB, 2023) para os algoritmos de *Naive Bayes* e *Random Forest*. Para o algoritmo K-NN será utilizado a linguagem *Python* (PYTHON, 2023) em conjunto com *PySpark* (APACHE SPARK, 2023). A figura 18, destaca a etapa da camada em que ocorrerá esse processamento.

Figura 18 – Camada de Processamento.

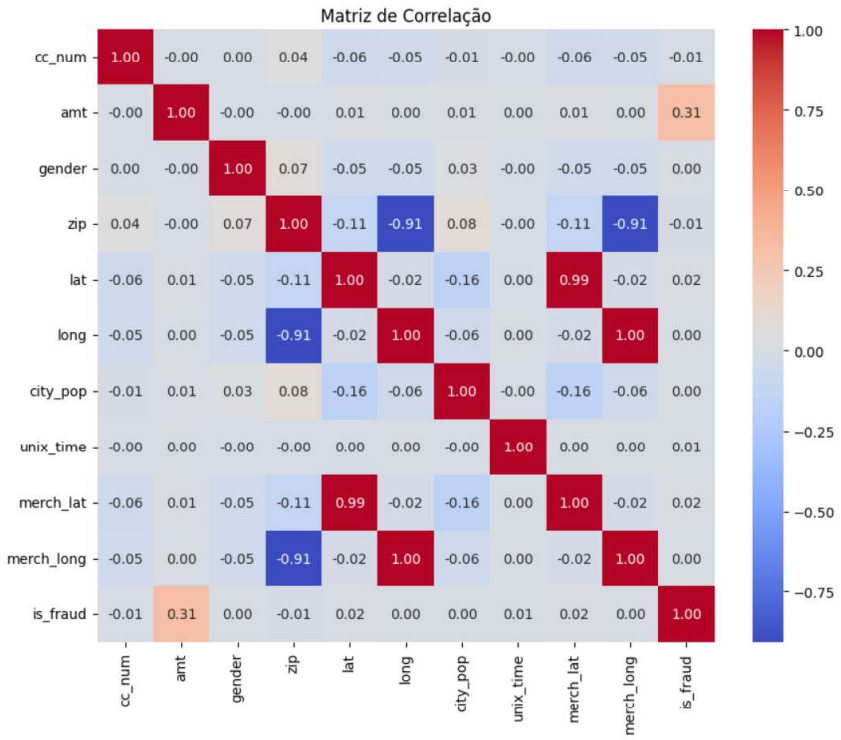


Fonte: Próprio Autor

3.5.1 Preparação dos Dados

Por meio da análise exploratória utilizando a matriz de correlação, observa-se que não existe uma correlação direta forte entre as variáveis que justifique deletar alguns dos atributos, conforme mostrado na Figura 19.

Figura 19 – Matriz de correlação.



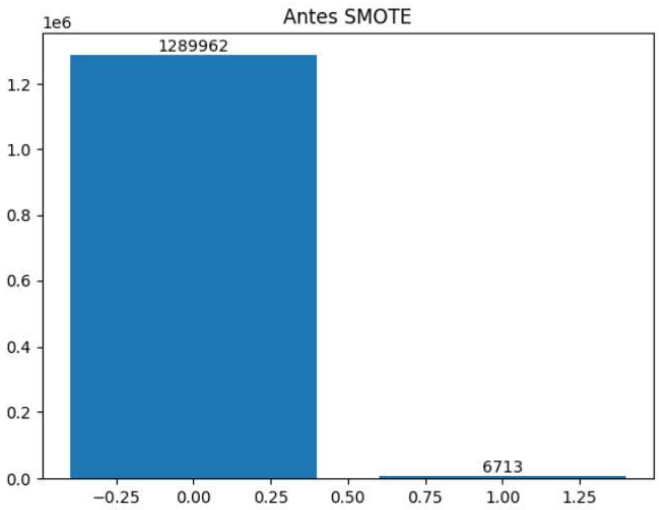
Fonte: Próprio Autor

Optou-se por adotar uma prática amplamente empregada em problemas de classificação, conhecida como divisão 70/30. Aplicando 70% dos dados para treinamento dos modelos e 30% para avaliação de seu respectivo desempenho conforme trabalhos de (PRANAVI et al., 2022) e (AHMED; SAINI ,2023).

Diante o desbalanceamento na distribuição dos dados, em que a presença desigual das classes pode introduzir um viés no aprendizado do algoritmo, especialmente favorecendo a compreensão de casos sem fraude, optou-se por empregar a técnica de balanceamento conhecida como *oversampling*. Durante os testes, essa abordagem demonstrou resultados mais promissores se comparado com a técnica de *undersampling*. Para solucionar o desbalanceamento, utilizou-se a biblioteca SMOTE em Python.

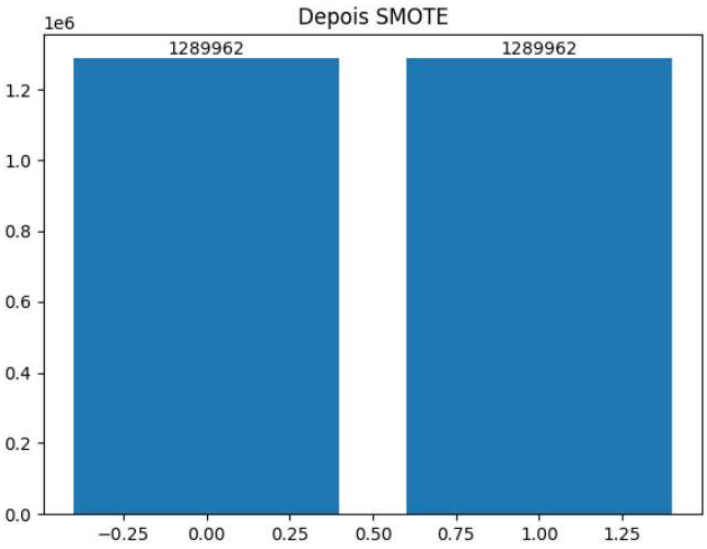
A Figura 20 mostra o resultado da base desbalanceada antes da execução do SMOTE, enquanto a Figura 21, ilustra o resultado com a base já balanceada.

Figura 20 – Base desbalanceada.



Fonte: Próprio Autor

Figura 21 – Base balanceada.



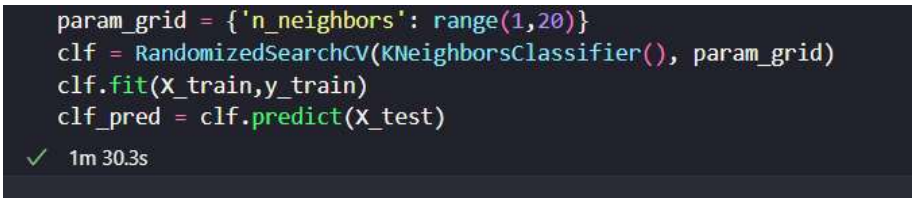
Fonte: Próprio Autor

3.5.2 Treinamento dos algoritmos de Machine Learning

Após concluir a etapa de preparação dos dados, a próxima etapa envolve a realização dos treinos dos algoritmos utilizando as funcionalidades e recursos disponíveis no MLlib em conjunto com *PySpark*. Inicia-se o treinamento para o K-NN, conforme apresentado na Figura 22.

Em virtude da limitação computacional, por se tratar de um computador considerado de uso pessoal, optou-se por utilizar um conjunto de dados de 30 mil registros para realização dos testes dos algoritmos de Machine Learning.

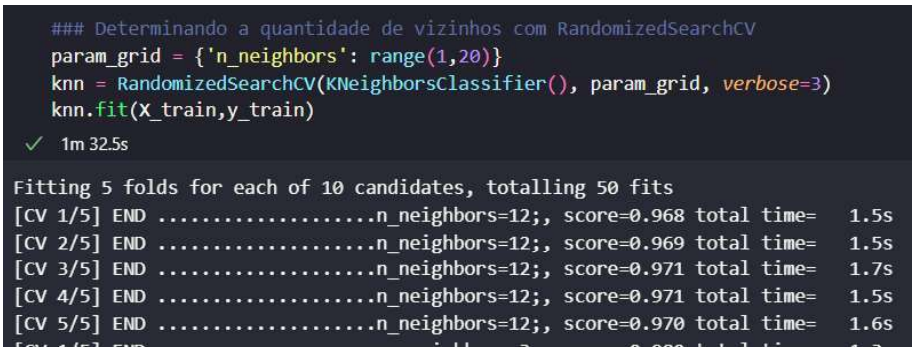
Figura 22 – Preparação para o treinamento K-NN



Fonte: Próprio Autor

A próxima etapa consiste em determinar a quantidade de vizinhos através da funcionalidade *RandomizedSearchCV,* A Figura 23, exibe parte dessa execução.

Figura 23 – definir a quantidade de vizinhos



Fonte: Próprio Autor

Após definir a quantidade de vizinhos, a próxima etapa é determinar qual é o melhor vizinho. Conforme ilustrado na Figura 24 encontra-se o melhor vizinho através da função *best_params_*.

Figura 24 – Encontrar o melhor parâmetro.

```
# Encontrar o melhor parametro
knn.best_params_
```

✓ 0.0s

```
{'n_neighbors': 1}
```

Fonte: Próprio Autor

Ao gerar a Matriz de confusão e calcular a acurácia para o algoritmo K-NN, obteve-se uma taxa de acurácia de 97%. No entanto, ao considerar que 0 representa uma transação não fraudulenta e 1 e uma transação com potencial de fraude, percebe-se que a precisão fica em 23% conforme demostrado na Figura 25.

Figura 25 – Matriz de confusão e acurácia K-NN

```
knn_pred = knn.predict(X_test)

print(confusion_matrix(y_test,knn_pred))
print('\n')
print(classification_report(y_test,knn_pred))
```

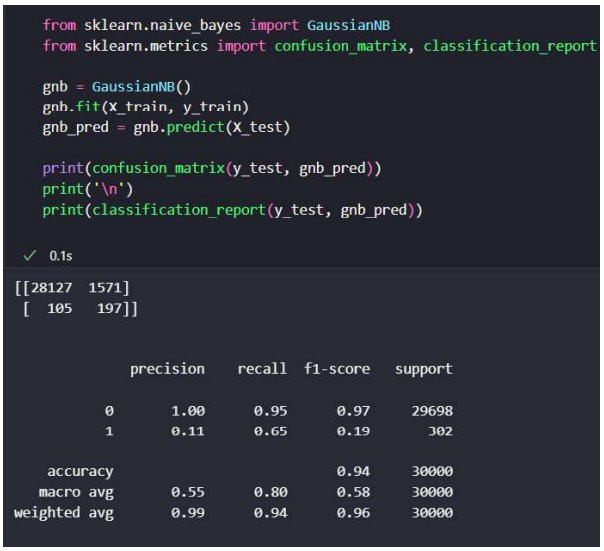
✓ 1.8s

```
[[28831   867]
 [   50   252]]


              precision    recall  f1-score   support

           0       1.00      0.97      0.98     29698
           1       0.23      0.83      0.35       302

    accuracy                           0.97     30000
   macro avg       0.61      0.90      0.67     30000
weighted avg       0.99      0.97      0.98     30000
```

Fonte: Próprio Autor

Para o *Naive Bayes*, após realizar o treinamento e gerar da matriz de confusão, conforme mostrado na Figura 26, obteve-se uma acurácia de 94% e precisão de 11% para identificar transação consideradas fraudulentas.

Figura 26 – Matriz de confusão e acurácia *Naive Bayes*



```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report

gnb = GaussianNB()
gnb.fit(X_train, y_train)
gnb_pred = gnb.predict(X_test)

print(confusion_matrix(y_test, gnb_pred))
print('\n')
print(classification_report(y_test, gnb_pred))
```

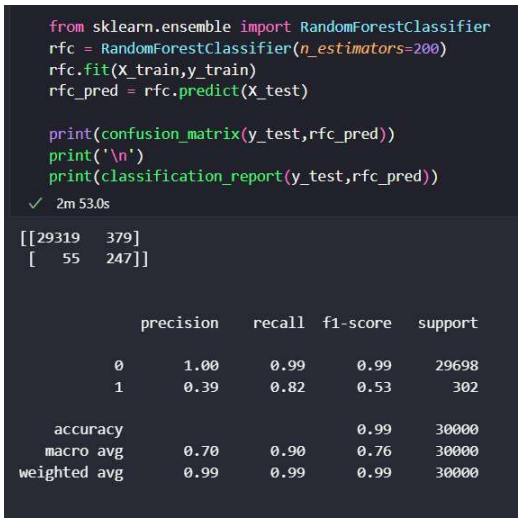
✓ 0.1s

```
[[28127  1571]
 [  105   197]]


              precision    recall  f1-score   support

           0       1.00      0.95      0.97     29698
           1       0.11      0.65      0.19       302

    accuracy                           0.94     30000
   macro avg       0.55      0.80      0.58     30000
weighted avg       0.99      0.94      0.96     30000
```

Fonte: Próprio Autor

Finalizando com o *Random Forest*, ao gerar a matriz de confusão conforme mostrado na Figura 27 obteve-se uma acurácia de 99% e uma precisão de 39% na identificação de transações fraudulentas.

Figura 27 – Matriz de confusão e acurácia *Random Forest*



```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train,y_train)
rfc_pred = rfc.predict(X_test)

print(confusion_matrix(y_test,rfc_pred))
print('\n')
print(classification_report(y_test,rfc_pred))
```

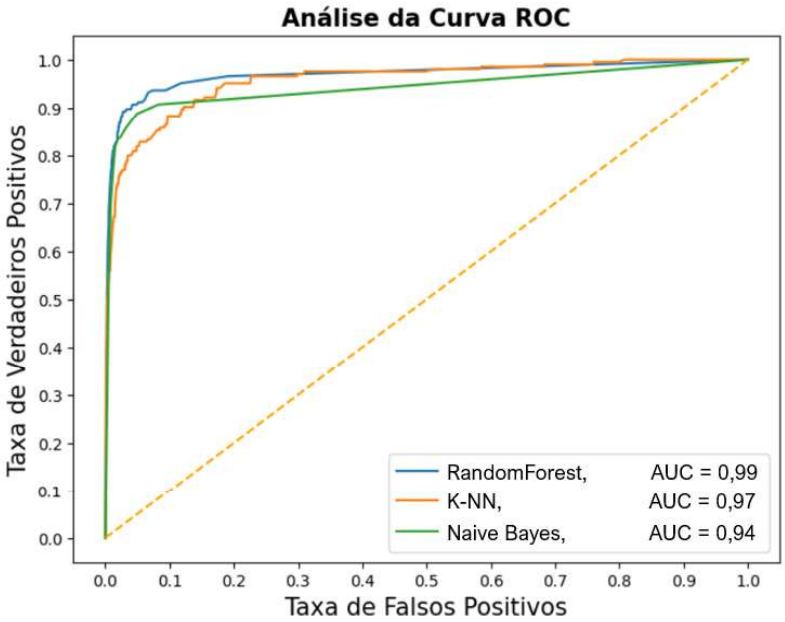
✓ 2m 53.0s

```
[[29319   379]
 [   55   247]]


              precision    recall  f1-score   support

           0       1.00      0.99      0.99     29698
           1       0.39      0.82      0.53       302

    accuracy                           0.99     30000
   macro avg       0.70      0.90      0.76     30000
weighted avg       0.99      0.99      0.99     30000
```

Fonte: Próprio Autor

Após a execução dos três algoritmos, elaborou-se o gráfico da Curva *Receiver Operating Characteristic* (ROC), conforme apresentado na Figura 28. Esta representação visual foi criada com o propósito de comparar o desempenho individual de cada algoritmo.

Figura 28 – Curva de ROC



Fonte: Próprio Autor

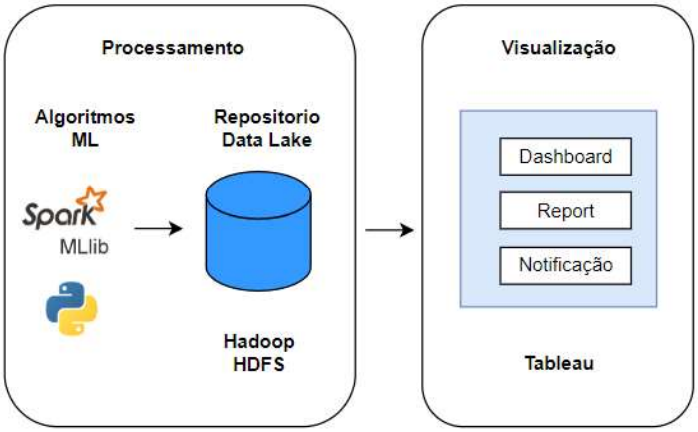
Após as execuções dos algoritmos e analisando seus respectivos desempenhos. preferiu-se por utilizar o *Random Forest* na solução deste projeto.

A princípio a solução foi pensada para utilizar computadores próprio e conforme for aceita pelas áreas implementar em uma solução que possa ser utilizada através de alguma solução em Nuvem.

Depois da execução do algoritmo escolhido o próximo passo será definir quais informações serão necessárias para serem disponibilizada para a camada de visualização, pois dessa forma, evita-se armazenamento de informações desnecessária. Conforme representado na Figura 29, no repositório HDFS ficará armazenadas informações sobre a localidade da transação, horário das transações assim como seus respectivos valores. A priori, selecionou-se essas informações,

porque são fundamentais para os Dashboards que foram gerados na camada de visualização.

Figura 29 – informações para camada de visualização



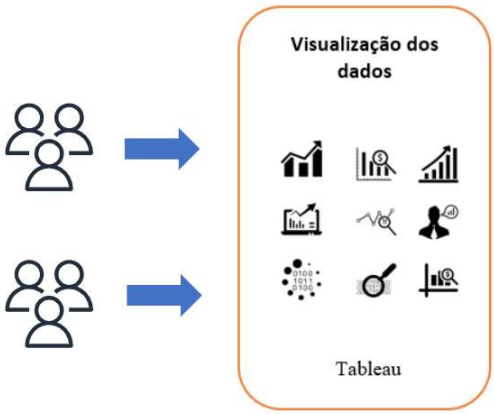
Fonte: Próprio Autor

3.6 Camada de visualização

A camada de visualização deste projeto tem como função criar uma camada intuitiva, flexível, de fácil configuração e amigável para os usuários do projeto. Para tanto, a ferramenta escolhida para implementar a camada de visualização foi o Tableau (TABLEAU, 2023). A decisão foi motivada pela sua interface de usuário intuitiva e amigável, o que facilita a criação de visualizações e relatórios de dados, tornando-os acessíveis a uma ampla gama de profissionais, independentemente de sua experiência em análise de dados. Além disso, a ferramenta permite configurar alertas via SMS para uma área ou usuário específico.

Dessa forma, a camada de visualização irá desempenhar um papel fundamental para auxiliar as áreas de prevenção de fraudes a rapidamente identificar, quase em tempo real, transações de cartão de crédito com alto suscetibilidade de fraude. Esta camada irá proporcionar uma interface dinâmica e intuitiva que permite a rápida e eficaz análise de dashboards, relatórios e informações, conforme representados na Figura 30.

Figura 30 – Camada de Visualização



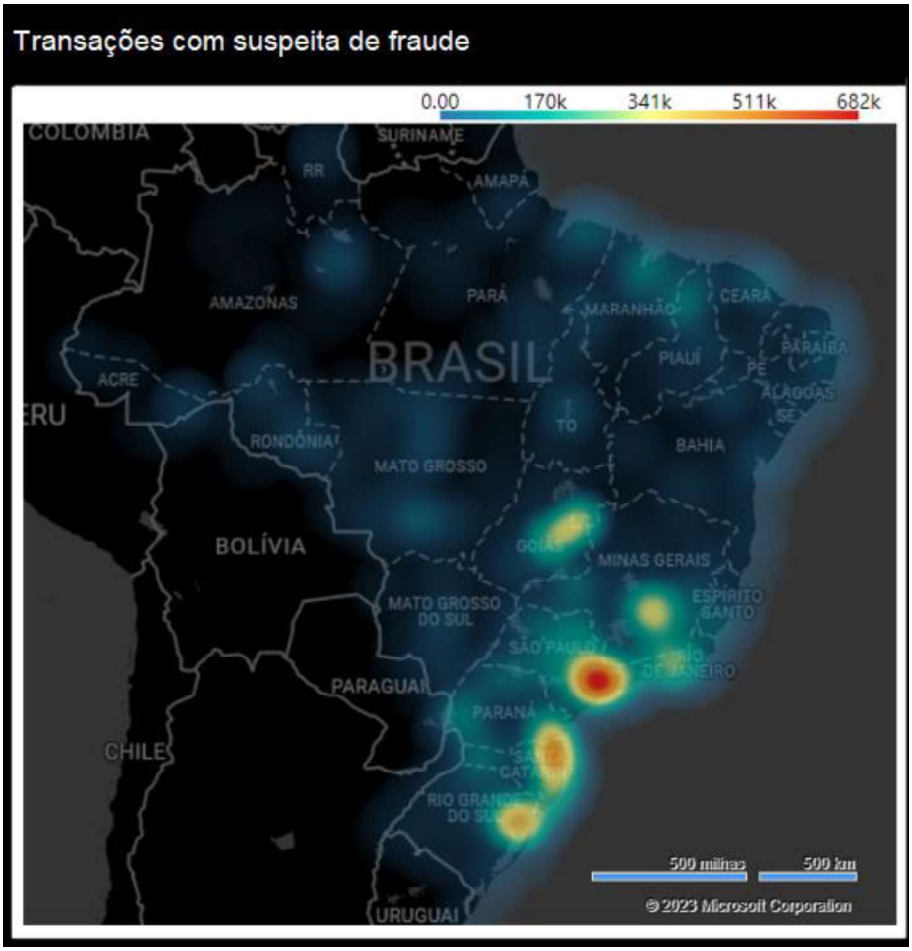
Fonte: Próprio Autor

Além de disponibilizar um *Dashboard* abrangente para a equipe responsável por lidar com questões relacionadas a fraudes, torna-se essencial desenvolver uma solução que permita o envio de notificações ou alertas direcionados a um grupo específico, composto por membros relevantes da equipe.

Essa abordagem visa não apenas fornecer um panorama visual por meio de *Dashboard*s, mas também garantir uma comunicação instantânea e eficaz. Na camada de visualização dos dados, o foco será em disponibilizar as informações de forma simplificada e acessível, proporcionando uma representação visual que simplifique o entendimento da complexidade dos dados por intermédio de gráficos, diagramas, mapas de calor, tabelas e outros elementos visuais modernos eficazes na comunicação de tendências e padrões.

A Figura 31, mostra um exemplo de gráfico gerada pelo Tableau, utilizando-se o mapa geopolítico do Brasil, em que através da utilização do mapa de calor é possível identificar a região que ocorrem as transações que apresentam características fraudulentas.

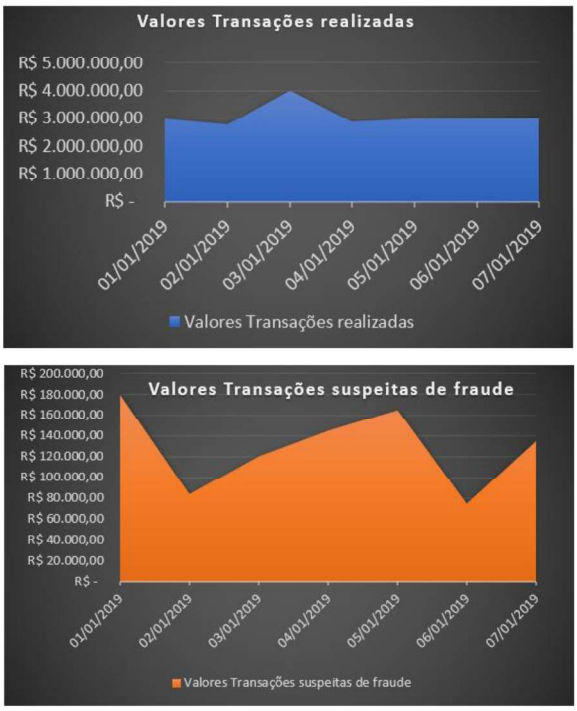
Figura 31 – Geolocalização das transações



Fonte: Próprio Autor

A Figura 32, mostra visualizações geradas através do Tableau, utilizando-se o mapa de área, ilustrando os valores das transações realizadas e os valores das fraudes diárias.

Figura 32 – Valores transacionados diariamente



Fonte: Próprio Autor

A Figura 33, ilustra uma visualização em gráfico de barras mostrando dia e os horários que ocorrem mais transações suspeitas de fraudes.

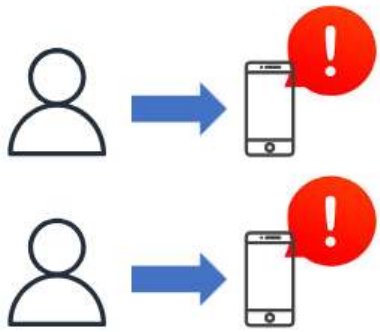
Figura 33 – Dias e horários com transações suspeitas de fraudes

Fonte: Próprio Autor

3.7 Camada de notificações

Ao estabelecer essa capacidade de notificação seletiva, aprimora não apenas a eficiência operacional, mas também melhora a capacidade de resposta diante de transações com caracteristicas de transações fraudulentas. A Figura 34 ilustra visualmente como esse sistema integrado de notificação se encaixa no contexto do *Dashboard* e sua contribuição para a detecção precoce de atividades fraudulentas.

Figura 34 – Envio de notificações e alarmes



Fonte: Próprio Autor

4 CONCLUSÃO

Os objetivos inicialmente propostos neste projeto foram concluídos e os resultados obtidos foram bastante satisfatórios. No período analisado, de janeiro de 2019 a dezembro de 2020, foram processadas um total de 1.852.394 transações de cartão de crédito. O montante financeiro dessas transações chega a R$ 129.785.332,00

Após a análise constatou-se que a quantidade de transações com suspeitas de fraudes foi de 10.651, totalizando aproximadamente R$ 5.121.500,00. Além disso, em alguns casos a quantidade de transações fraudulentas representaram até 6% do valor total das transações realizadas no dia.

Ao analisar os resultados dos algoritmos, observa-se que o K-NN exibe uma acurácia de 97%, mas com uma precisão relativamente baixa, alcançando apenas 23%. Por sua vez, o *Naive Bayes* apresenta uma acurácia de 94%, com uma precisão ainda mais modesta de 11% em comparação ao K-NN.

Por último, o *Random Forest* demonstrou ser mais eficiente do que seus predecessores, alcançando uma notável acurácia de 99% e uma precisão significativamente superior, atingindo 39%. Considerando essas métricas, o Random Forest surge como uma escolha mais apropriada para a detecção de transações fraudulentas.

Além disso, ao disponibilizar alguns Dashboard com informações próximas dos acontecimentos, possibilita aos usuários obterem uma visão próxima ao tempo real do processamento das transações de crédito e montarem suas estratégias para reduzir os prejuízos com transações fraudulentas.

4.1 Contribuições do trabalho

Neste trabalho a solução em Big Data proposta, mostrou-se satisfatória, pois ao processar as informações o mais próximo do ocorrido, proporciona a equipe de fraude atuar com maior eficiência.

4.2 Trabalhos futuros

Como trabalhos futuros, recomenda-se explorar a análise de outros tipos de transações efetuadas em pontos de venda, como é o caso das transações de débito. Devido à eficiência demonstrada no curto período de identificação de transações suspeitas de fraude, a solução mostra-se viável e promissora para lidar com transações de débito. Esta ampliação do escopo permitiria uma compreensão abrangente do desempenho da solução em diferentes modalidades de transações, contribuindo para sua aplicação em uma variedade de cenários.

REFERÊNCIAS BIBLIOGRÁFICAS

ABECS Pagamentos com cartões, 2022. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2022-11/pagamentos-com-cartoes-movimentam-r-827-bilhoes-no-3o-trimestre?_gl=1*n9n6iw*_ga*MTc1MTcyODk3NC4xNjk2MTgwMzk5*_ga_TGW7R30M20*MTY5NjE4MDM5OS4xLjEuMTY5NjE4MDU0NC41MC4wLjA>. Acesso em 10 de mai. de 2023.

AHMED, A.; SAINI, R., Detection of Credit Card Fraudulent Transactions Utilizing Machine Learning Algorithms, 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2023, pp. 1-5, doi: 10.1109/INOCON57975.2023.10101137

APACHE HADOOP introdução HDFS, 2022. Disponível em: <https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#Introduction> . Acesso em 20 jun. de 2023.

APACHE NIFI documentação e arquitetura, 2023. Disponível em: < https://nifi.apache.org/project-documentation.html>. Acesso em 21 jun. de 2023.

APACHE SPARK PYSPARK Visão geral, 2023. Disponível em: < https://spark.apache.org/docs/latest/api/python/index.html>. Acesso em 14 ago. de 2023.

Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

FEBRABAN Tentativas de fraudes, 2023. Disponivel em: <https://portal.febraban.org.br/noticia/3903/pt-br/>. Acesso em 12 de maio de 2023.

He, H.; Ma, Y. *Foundations of Imbalanced Learning,* in Imbalanced Learning: Foundations, Algorithms, and Applications, IEEE, 2013, pp.13-41, doi: 10.1002/9781118646106.ch2.

IZBICKI, R. Aprendizado de máquina: uma abordagem estatística. São Paulo: Editora UICLAP, 2022, 1ª edição. ISBN: 978-65-00-02410-4.

JURGOVSKY, J.; Granitzer, M; Ziegler, K.; Calabreto, S.; Portier, P.; Guelton, L.; Caelen, C. Sequence classification for credit-card fraud detection. ExpertSystems with Applications, v. 100, p. 234-245, 2018.

JOSE,S.; DEVASSY, D.; ANTONY, A. M., Detection of Credit Card Fraud Using Resampling and Boosting Technique, 2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), Ernakulam, India, 2023, pp. 1-8, doi: 10.1109/ACCTHPA57160.2023.10083376.

KAGGLE dataset de fraudes, 2019. Disponivel em: <https://www.kaggle.com/datasets/kartik2112/fraud-detection/code>. Acesso em 10 de jun. de 2023

MALINI, N.; PUSHPA, M. *Analysis on credit card fraud identification techniques based on KNN and outlier detection*, 2017 third international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB). IEEE, pp. 255-258, 2017.

MARQUESONE, R. Big Data: Técnicas e tecnologias para extração de valor de dados. São Paulo: Casa do Código, 2016. 245 p. ISBN 9788555192319.

MORAES, D. Modelagem de fraude em cartão de crédito. Universidade Federal de São Carlos, 2008.

MLlib Guia da ferramenta e arquitetura, 2023. Disponível em: <https://spark.apache.org/docs/latest/mllib-guide.html. Acesso em 21 jun. de 2023.

NAGPAL, A.; GABRINI, G., Python for Data Analytics, Scientific and Technical Applications, 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 140-145, doi: 10.1109/AICAI.2019.8701341.

NBD-PWG, N. NIST Big Data Interoperability Framework - Reference Architecture. v.6, September 2015.

NIST. About NIST. 2022. Acesso em 10 de set. de 2023. Disponível em: https://www.nist.gov/about-nist

OLIVEIRA, P. Detecção de fraudes em cartões: um classificador baseado em regras de associação e regressão logística. 2016. 103 f. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

PARMAR, A. et al. Performance Comparison of Hadoop Map Reduce and Apache Spark. International Journal of Advance Engineering and Research Development, v. 5, n. 03, p. 1323-1328, sep 2018. ISSN 23484470.

PARODI, L. Manual das fraudes. São Paulo: Brasport 2ª Edição, 2008. 440 p. ISBN 9788574523484

PRANAVI, N. et al., Credit Card Fraud Detection Using Minority Oversampling and Random Forest Technique, 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-6, doi: 10.1109/INCET54531.2022.9824146.

SANTIAGO, G. Um Processo para Modelagem e Aplicação de Técnicas Computacionais para Detecção de Fraudes em Transações Eletrônicas. 2014. 130 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2014.

SINGHAI, A.; AANJANKUMAR, S.; POONKUNTRAN, S. "A Novel Methodology for Credit Card Fraud Detection using KNN Dependent Machine Learning Methodology," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 878-884, doi: 10.1109/ICAAIC56838.2023.10141427.

SEGOOA, M. A.; KALEMA, B. M. Improve Decision Making towards Universities Performance through Big Data Analytics. In: 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems. [S.l.]: IEEE, 2018. ISBN 9781538630600.TABLEAU: O que é tableau, 2023. Disponível em <https://www.tableau.com/pt-br/why-tableau/what-is-tableau>. Acesso em 01 jul. de 2023

Python: Tutorial Python, 2023. Disponível em <https://docs.python.org/3/tutorial/index.html>. Acesso em 05 jun. de 2023

VAIRAM, T.; SARATHAMBEKAI, S.; BHAVADHARANI, S.; Kavi Dharshin, A. Evaluation of Naive Bayes and Voting Classifier Algorithm for Credit Card Fraud Detection. 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 602-608, doi: 10.1109/ICACCS54159.2022.9784968.