INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Machine Learning na identificação de operações de pagamento fraudulentas e atribuição de nível de risco de operações e clientes

NUNO FILIPE ROCHA COSTA
julho de 2023

POLITÉCNICO
DO PORTO

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Machine Learning na identificação de operações de pagamento fraudulentas e atribuição de nível de risco de operações e clientes

NUNO FILIPE ROCHA COSTA
Junho de 2023

POLITÉCNICO
DO PORTO

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Machine Learning na identificação de operações de pagamento fraudulentas e atribuição de nível de risco de operações e clientes

NUNO FILIPE ROCHA COSTA
Junho de 2023

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Payment Fraud Detection with Machine Learning

Identification of fraud operations and the corresponding level of risk per merchant

Nuno Filipe Rocha Costa

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática, Área de Especialização em

Sistemas de Informação e Conhecimento

Orientador: Goreti Marreiros

Co-orientador: Jorge Meira

Júri:

Presidente:

Bertil Marques, Doutorada, ISEP

Arguente:

Davide Carneiro, Doutorado, ISEP

Porto, Junho 2023

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 22 de julho de 2023

Dedicatória

Dedico esta dissertação a mim mesmo, como uma conquista pessoal e um marco na minha jornada académica. Este trabalho reflete o meu esforço, dedicação e perseverança.

Dedico à minha noiva, Ana Magalhães, pelo seu amor, apoio e compreensão ao longo desta jornada desafiadora. Sua presença constante e encorajamento foram fundamentais para que eu pudesse alcançar este objetivo. Dedico este trabalho a ti, com profundo carinho e gratidão.

Resumo

A utilização do comércio eletrónico*,* mais conhecido por *E-Commerce,* aumentou nos últimos anos, eventualmente devido à pandemia covid-19 bem como a outros fatores. Este aumento levou à criação de novas empresas, mas também estimulou diversas atividades de fraude on-line. As empresas cada vez mais têm dificuldades para combater e impedir tentativas de fraudes devido à sua constante mudança.

Esta dissertação visa apresentar todo o processo de estudo, experimentação, desenvolvimento e avaliação de uma solução de *Machine Learning* para a deteção de fraudes, desde a recolha dos dados ao seu processamento e apresentação de resultados.

Durante o desenvolvimento foram aplicados conhecimentos obtidos na fase de estudo e técnicas de *Machine Learning.* O objetivo é comparar os diferentes algoritmos [*Decision Tree* (DT), *Isolation Forest* (IF) e *K-nearest neighbors* (KNN)] e verificar quais oferecem melhor resultado a cada conjunto de dados [Clientes e Transferências], na deteção de fraude ou *outliers*. Um objetivo secundário, tirando partido do uso de *Machine Learning,* será efetuar previsões nomeadamente o nível de risco de um novo cliente por *Merchant.*

Nesta tese pode ser encontrado um estudo e comparação da aplicação de dois modelos, um para a previsão de fraude de um cliente e outro para a deteção de *outlier* nas transferências, assim como um estudo sobre as melhores configurações e parâmetros para cada modelo.

Palavras-chave: *Machine Learning*, *E-Commerce,* Deteção de Fraude, Deteção de *Outlier*

Abstract

The use of electronic commerce, commonly known as E-commerce, has increased in recent years, possibly due to the Covid-19 pandemic as well as other factors. This growth has led to the creation of new companies but has also stimulated various online fraud activities. Companies are increasingly facing challenges in combating and preventing fraud attempts due to their constantly evolving nature.

This dissertation aims to present the entire process of studying, experimenting, developing and evaluating a Machine Learning solution for fraud detection, from data collection to processing and presentation of results. During the development, knowledge gained in the study phase and Machine Learning techniques were applied. The goal is to compare different algorithms [Decision Tree (DT), Isolation Forest (IF), and K-nearest neighbors (KNN)] and determine which ones offer the best results for each data set [Customers and Transfers] in fraud or outlier detection. A secondary objective, taking advantage of the use of Machine Learning, is to make predictions, particularly the risk level of a new customer by Merchant.

This thesis includes a study and comparison of the application of two models, one for predicting customer fraud and another for outlier detection in transfers, as well as a study on the best configurations and parameters for each model.

Keywords: Machine Learning, E-Commerce, Fraud Detection, Outlier Detection

Agradecimentos

Gostaria de expressar meus sinceros agradecimentos a todas as pessoas que contribuíram para a realização desta dissertação:

Primeiramente, gostaria de agradecer à minha orientadora, Goreti Marreiros pela sua orientação, apoio e conhecimento ao longo deste processo. As suas sugestões valiosas e paciência foram fundamentais para o desenvolvimento deste trabalho.

Agradeço também ao professor Jorge Meira que dedicou o seu tempo para rever a minha dissertação e fornecer feedback construtivo. As suas observações e insights foram extremamente importantes para a melhoria do meu trabalho.

Gostaria de expressar a minha gratidão aos meus colegas de trabalho e amigos, que me acompanharam nesta jornada de pesquisa. As suas contribuições e discussões enriqueceram a minha compreensão sobre o tema e proporcionaram um ambiente de apoio e colaboração.

Não posso deixar de agradecer à Ifthenpay que tornou possível a realização deste estudo. A sua confiança e suporte foram essenciais para a condução da pesquisa bem como a partilha dos dados.

Por fim, quero expressar minha gratidão à minha família e amigos pelo seu amor, encorajamento e apoio incondicional ao longo desta jornada académica. As suas palavras de incentivo e compreensão foram fundamentais para me manter motivado e perseverante.

Novamente, agradeço a todos os que contribuíram de alguma forma para este trabalho. O seu apoio e envolvimento foram inestimáveis e sou profundamente grato por isso.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Equações

Lista de excertos de Código

Acrónimos e Símbolos

Lista de Acrónimos

AI *Artificial Intelligence/*Inteligência Artificial

ANN *Artificial Neural Networks/*Redes Neurais artificiais

BD Base de Dados

CAE Classificação Portuguesa das Atividades Económicas

CPU *Central Processing Unit*

CRISP-DM *Cross Industry Standard Process for Data Mining*

DT *Decision Tree/*Árvore de decisão

GB *Gigabyte*

HDD *Hard Disk Drive*

IF *Isolation Forest*

KNN *K-Nearest Neighbors*

MB Multibanco

ML *Machine Learning/*Aprendizagem de máquina

NIF Número de Identificação Fiscal

NN *Neural Networks/*Redes Neurais

NPPD *New Product and Process Development*/Desenvolvimento de Novos Produtos e Processos estruturados

RAM *Random Access Memory*

RF *Random Forest*

RGPD Regulamento Geral sobre a Proteção de Dados

SIBS Sociedade Interbancária de Serviços

SO Sistema Operativo

SSD *Solid State Drives*

SVM *Support Vector Machine/*Máquina de vetores de suporte

TPR *True Positive Rate*

UE União Europeia

UI *User Interface*

UX *User Experience*

VA *Value Analysis*/Análise de Valor

VPC *Value Proposition Canvas*

1 Introdução

Este documento sistematiza o trabalho desenvolvido ao longo desta Dissertação de Mestrado em Engenharia Informática no Instituto Superior de Engenharia do Porto.

Este trabalho foi desenvolvido em conjunto com a empresa Ifthenpay, sendo o seu objetivo principal a "deteção de fraude".

Neste Capítulo é apresentada a empresa na qual o projeto irá ser desenvolvido, o contexto, o problema a resolver, os objetivos propostos, a abordagem a seguir e por último, a estrutura do documento, sintetizando os assuntos abordados em cada capítulo.

Ao longo do documento irão ser abordados requisitos funcionais não funcionais que, devido à sua complexidade, não foram implementados. Serão apresentadas justificações para essa não implementação, bem como sugestões para trabalhos futuros relacionados com esses requisitos.

1.1 Apresentação da Empresa

A Ifthenpay é uma empresa portuguesa que atua na área de pagamentos digitais desde 2005. Foi criada pela empresa Ifthensoftware, sendo supervisionada pelo Banco de Portugal e cumprindo as normas impostas pelo mesmo, tornou-se uma empresa autónoma. A empresa Ifthenpay nasceu de uma necessidade especifica de um dos clientes da Ifthensoftware e desde esse momento continuou a crescer até ser o que é hoje.

A sua atividade é fornecer métodos de pagamentos a empresas. Até 2018, só disponibilizavam referências multibanco (MB), desde essa altura começaram a alargar o leque de métodos até à data. Neste momento disponibiliza referências MB, *MB Way*, *PayShop* e Cartão de Crédito (*Visa* e *MasterCard*) (*Ifthenpay | Pagamentos Digitais Para a Sua Empresa*, n.d.).

1.2 Contexto

O comércio eletrónico*,* mais conhecido por *E-Commerce,* está cada vez mais presente nos dias de hoje, facto que poderá estar associado aos efeitos da pandemia de covid-19, o que levou muitas pessoas a procurar e a optar por este tipo de comércio.

Segundo o estudo "O Comércio Eletrónico em Portugal e na União Europeia", 27% dos indivíduos nunca efetuaram compras pela internet e de cerca de 10% dos indivíduos efetuaram vendas online. O mesmo estudo revela também que o número de cibercompradores em Portugal aumentou mais em 2020 e 2021, subindo mais do que a média da UE. Nesse sentido, os resultados indicam que 13% dos portugueses fizeram compras online mais de cinco vezes nos últimos três meses (*Pandemia Marca o Ritmo No Comércio Eletrónico - E-Commerce - Jornal de Negócios*, 2022).

A Ifthenpay não foge a este fenómeno. Podemos ver pelos dados disponibilizados no seu site que desde 2012/2013, o valor transacionado pelos seus métodos tem aumentado de uma forma exponencial e o maior aumento foi em 2020 e 2021 como justifica o estudo. Juntando a todos estes fatores, também podemos verificar que o número de pessoas a subscrever os seus serviços aumentaram de forma linear e que teve uma disrupção em 2020 e 2021 com um aumento superior ao esperado e maior do que nos anos anteriores (*Ifthenpay | Pagamentos Digitais Para a Sua Empresa*, n.d.).

Por esta análise temporal consegue-se perceber que as pessoas estão a aderir cada vez mais aos métodos digitais e a gastar mais dinheiro neles e o maior "boom" foi em 2020 e 2021, associado aos efeitos da pandemia covid-19.

Ao utilizar estes meios de pagamentos online estamos mais sujeitos a fraudes. Segundo outro estudo, 84% das pessoas inquiridas confirmaram a legitimidade da empresa ou entidade antes de efetuar o pagamento pelo produto ou serviço adquirido (*Pandemia Marca o Ritmo No Comércio Eletrónico - E-Commerce - Jornal de Negócios*, 2022). Com o crescimento deste tipo de serviços e de plataformas on-line provocou um aumento de atividades de fraude on-line. Devido à rápida inovação tecnológica e comercial, que abre portas a um conjunto de produtos em constante expansão, os modelos de deteção de fraude supervisionados ou semi-supervisionados tornam-se insuficientes e ineficazes para estes serviços emergentes (Liu et al., 2022).

Com este constante aumento pela procura do comércio on-line e com o consequente aumento das fraudes cresce a preocupação por proteger os clientes e os interesses da empresa.

1.3 Problema

Com os milhares de transferências processados por dia, é necessário um mecanismo que auxilie na deteção de irregularidades e consiga descobrir padrões de risco através do histórico de dados.

Este trabalho no momento é feito recorrendo a padrões pré-estabelecidos e métodos estatísticos tais como média, mediana, máximos, frequências, entre outros. Alguns exemplos de alertas dos mecanismos já implementados:

- O cliente teve um pagamento superior a x;
- O cliente começou a ter muitos pagamentos em relação a sua média de pagamentos e de valores superior a x;
- O cliente esteve muito tempo sem ter pagamentos e de repente começou a ter pagamentos;
- A soma dos pagamentos é superior a x;
- Quantidade de reclamações do cliente;
- Elevados pagamentos ou montantes comparados com outros clientes do mesmo mercado/setor.
- Atribuição automática de grau de risco, atualizado diariamente, baseado em informações do cliente, do tipo de produto contratado, do tipo de mercado e estatísticas de pagamentos.

Os métodos já implementados poderão com o passar do tempo estar errados ou não corresponder à realidade, podendo não contemplar todas as situações de risco levando à necessidade de um trabalho excessivo da equipa de análise dos pagamentos antes de efetuar qualquer transferência dos fundos respetivos. A empresa efetua as transferências dois dias úteis após o pagamento (D+2), deixando o dia seguinte ao pagamento para analisar as transações a efetuar, o que torna o trabalho de deteção de fraude cada vez mais complexo e complicado com o aumento das transações.

1.4 Objetivos

O principal objetivo é o desenvolvimento de um software de *Machine Learning* para minimizar operações fraudulentas com base no histórico de pagamentos e dados dos clientes, detetando num universo de milhões de operações de pagamento as mais suscetíveis de serem operações fraudulentas.

Com a análise do histórico de pagamentos é pretendido atribuir o nível de risco de operações fraudulentas por *merchant*, com base no seu perfil, no histórico de operações do mesmo e aplicar ao *dataset* deteção de anomalias/outliers para melhor interpretação.

Um requisito importante é conseguir ter até ao dia útil seguinte os pagamentos analisados para poder transferi-los.

1.5 Abordagem e Processo de Desenvolvimento

Devido ao projeto se basear em exploração de um conjunto de dados, de forma analítica, mais conhecido por *Data Mining* ou Mineração de Dados, optou-se por seguir a metodologia CRIPS-DM (**CR**oss **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining). Criado em 1996, consistindo em conjuntos de boas práticas para um projeto de *Data Science*. Posteriormente em 1999, passou a ser padrão nos projetos de *Data Mining* de todos os setores, passando a ser a metodologia mais comum para projetos de mineração de dados, análises e ciência de dados (*CRISP-DM Help Overview - IBM Documentation*, n.d.; *What Is CRISP DM? - Data Science Process Alliance*, n.d.; Matheus Zambinati Rosa, 2020).

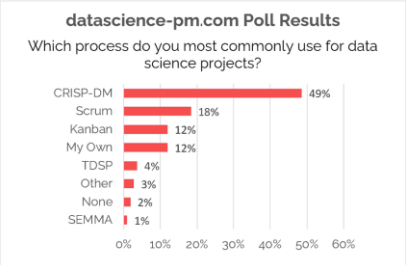


Figura 1 - Processos mais usados em projetos de *Data Science* (JEFF SALTZ, 2022)

A CRISP-DM consiste em seis etapas (Matheus Zambinati Rosa, 2020; Rodrigo Ribeiro G., 2020; *What Is CRISP DM? - Data Science Process Alliance*, n.d.):

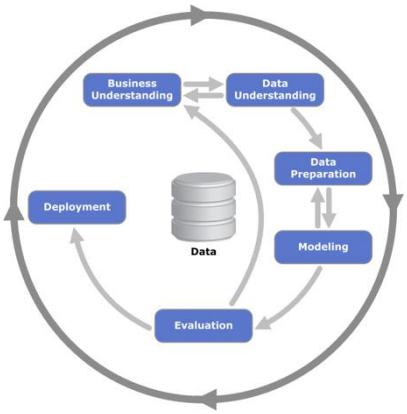


Figura 2 - Diagrama CRIPS-DM (Matheus Zambinati Rosa, 2020)

- *Business Understanding* (Compreensão do Negócio)
  A primeira etapa consiste em identificar o problema a ser resolvido podendo ser definido nas seguintes definições:
    - **Contexto** – Definição do contexto do problema e como o projeto pode resolvê-lo;
    - **Objetivos de negócios**– Descrição do objetivo do projeto;
    - **Critério de aceitação** – Estabelecer os critérios de aceitação para definir o projeto como entregue;
    - **Plano do projeto** – Definição de um plano e seleção de tecnologias e ferramentas a usar em cada fase do projeto.

  Começou-se por reunir com a empresa para perceber o problema a resolver e quais os objetivos do projeto. Traçou-se um plano começando por uma revisão da literatura sobre *Machine Learning* (ML), aplicado ao contexto de fraude para perceber o que já existe. O capítulo Estado da Arte é o resultado desse estudo, onde é possível retirar informações e melhorar a visão anterior da solução. Esta pesquisa vai permitir uma melhor compreensão das vantagens e desvantagens das abordagens existentes para o problema que este trabalho pretende abordar.

- *Data Understanding* (Compreensão dos Dados)
  Esta etapa é muito importante no desenvolvimento do projeto, sendo considerada como crítica e devendo ser realizada com bastante atenção.
  Pode ser definida em quatro tarefas, recolha de dados inicial, descrição dos dados, exploração dos dados e verificação da sua qualidade.

  Para uma melhor compreensão do projeto fez-se uma recolha inicial dos dados para perceber o que já existe disponível e bem como as relações entre eles.

- *Data Preparation* (Preparação dos Dados)
  Esta etapa tem como objetivo preparar os dados para a etapa seguinte, *Modelling*, de acordo com as necessidades do projeto. Segundo diferentes autores esta etapa pode significar 90% de todo o tempo dedicado ao projeto, sendo comum ocupar 80%.
  Também conhecida como *Data Mining* consiste em cinco fases:
    - **Seleção dados** – Determinar quais conjuntos de dados a usar bem como os motivos de inclusão e exclusão.
    - **Limpar dados** – Corrigir, imputar ou remover valores errados que causem entropia.
    - **Construir dados** – Criação de novos atributos necessários ou que poderão ser úteis mais a frente.
    - **Integrar dados** – Junção dos dados das diferentes fontes de dados.
    - **Formatar dados** – Formatar os dados finais de acordo com o necessário, como normalizando os dados para usar em operações matemáticas.

  Com a revisão da literatura concluída, e por ser importante, foi efetuada uma reunião novamente com a empresa e orientadores onde foi exposto o que foi encontrado e adquirido com o estudo. Desta forma são expostas novas informações que podem mudar as estratégias adotadas bem como os objetivos finais do projeto. Também nesta fase começou-se a selecionar os dados e explorar os *dataset* disponibilizados.

- *Modelling* (Modelagem dos dados)
  Definição do modelo de *Data Mining* mais adequado para os dados preparados nas etapas anteriores. Consiste na seleção da(s) tecnologia(s) a usar, desenho dos testes e construção do(s) modelo(s).

  Com a análise e exploração dos dados terminada é importante selecionar as melhores técnicas a utilizar para os diferentes conjuntos de dados. Com a exploração de dados chegou-se à conclusão de agrupar os dados por método de pagamento. Os *dataset* para cada método de pagamento eram constituídos por atributos e necessidades diferentes, o que pode dificultar e aumentar a complexidade de juntar tudo no mesmo modelo.

- *Evaluation* (Avaliação do modelo)
  Com a fase de modelação concluída está na altura de comparar o trabalho realizado com os critérios de aceitação. Se algum critério não se encontrar conforme volta para o início do processo para ser analisado novamente e reformulado caso necessário. Se o resultado for sucesso pode-se avançar para a próxima fase, *deployment*.

  Cada modelo terminado será confrontado com os requisitos e analisado o seu resultado junto com a empresa. Se este resultado for positivo é implementado o modelo em produção, caso contrário o modelo será revisto e discutidas as alterações necessárias.

- *Deployment* (Implantação do modelo)
  Esta etapa final consiste em colocar todo o desenvolvimento em produção, implementando o modelo, monitorizando-o, documentando e avaliando todo o projeto. Com os modelos implementados em produção é esperado no final que todos os requisitos sejam implementados bem como toda a documentação redigida neste documento como demonstração do trabalho.

1.6 Estrutura do documento

Este documento encontra-se dividido em 7 capítulos seguindo-se as referências utilizadas e por fim os anexos.

O primeiro (e atual) capítulo apresenta o problema que estamos a endereçar e a solução proposta. O segundo capítulo apresenta os antecedentes na área que este trabalho se insere. Em seguida no capítulo três efetua-se uma análise de valor à solução a ser desenvolvida. Aqui serão aplicados vários modelos de análise e construído um modelo de negócio. No capítulo quatro apresenta-se uma análise e desenho da solução a desenvolver através de requisitos, alternativas e um desenho detalhado. No capítulo quinto são evidenciadas as metodologias utilizadas para o desenvolvimento e pontos principais de implementação como lógica e código utilizado. No capítulo seis são apresentados os resultados obtidos bem como a avaliação de cada modelo. Finalmente, no capítulo sete, as limitações deste trabalho e direções futuras são discutidas e é realizada uma apreciação global do trabalho efetuado.

2 Estado da Arte

Neste capítulo é efetuado um estudo da arte das áreas tecnológicas/científicas relativas à dissertação, através da descrição de alguns conceitos, métodos e abordagens existentes mais úteis para a resolução do problema apresentado.

O capítulo é composto por uma análise de *Machine Learning*, uma análise aos conceitos, abordagens e ferramentas existentes aplicadas à deteção de fraude e como podem ser aplicadas no desenvolvimento da solução ou adaptadas. Por fim, é efetuado um sumário de toda a informação recolhida nos tópicos anteriores.

2.1 Machine Learning

Aprendizagem de máquina mais conhecida como *Machine Learning* (ML) é uma forma aplicada de tecnologias de *Artificial Intelligence* (AI). AI trata de imitar a capacidade humana de resolução de problemas e tomada de decisão. Para John McCarthy num artigo em 2004 "É a ciência e a engenharia de fazer máquinas inteligentes, especialmente programas de computador inteligentes, semelhante à tarefa de usar computadores para entender a inteligência humana, mas a AI não precisa de se limitar a métodos que são biologicamente observáveis.", mas já muitos anos antes, em 1950, Alan Turing, já investigava sobre o assunto e um dos artigos mais conhecido é o ”Computing Machinery and Intelligence” (Turing, 1950; *What Is Artificial Intelligence (AI)? | IBM*, n.d.)

Uma das questões importantes em ML é quais os algoritmos que funcionam melhor para quais problemas. As questões-chave são:

- O problema precisa de uma abordagem supervisionada ou não supervisionada?
- É um problema preceptivo ou numérico?
- A saída requer uma regressão ou classificação?

- Quantos dados de treino estão disponíveis?

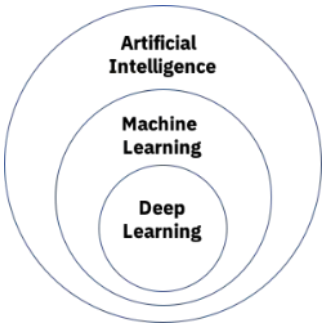


Figura 3 - Artificial intelligence vs Machine Learning (*What Is Artificial Intelligence (AI)? | IBM*, n.d.)

Os algoritmos de ML são modelos matemáticos construídos a partir de dados de amostra, conhecidos como dados de treino, geralmente representando até 30% de todos os dados disponíveis, para fazer previsões ou decisões sem serem programados.

É muito comum ML ser chamada de "análise preditiva", mas este conceito é só aplicado quando usado em algoritmos de *Neural Networks.* Mas ML é muito mais, inclui muitos mais algoritmos e métodos, resolvemos o mesmo problema de diferentes maneiras e o mesmo problema com diferentes algoritmos. Pode ser definido pelo uso da máquina para detetar os diferentes tipos de dados que possuem diferentes padrões de forma automatizada por sistemas e aplicativos de computador, consistindo em sistemas automatizados que facilitam o processo de previsão em vez dos esforços humanos de previsão. Os sistemas de ML são baseados num processo de análise de *big data* para executar a tarefa de previsão (Seify et al., 2022).

Quando optamos por usar ML existem logo ao início duas abordagens que se dividem entre supervisionadas e não supervisionadas. No entanto, alguns artigos incluem um terceiro tipo, ML de reforço e semi-supervisionado (Seify et al., 2022).

2.1.1 Aprendizagem Supervisionada

É o tipo mais comum e difere dos restantes pela necessidade de rotular os dados. Os dados de *input*(entrada) incluem uma *Label*(rótulo) correspondente ao *outpout*(saída) desejada. O principal aspeto positivo desta é que a saída do sistema automatizado é útil para o uso humano e classificação. No entanto, este pode ser difícil de implementar onde a quantidade de dados não pode ser rotulada devido ao seu alto volume. O algoritmo ML compara os resultados obtidos com os resultados reais e esperados para identificar erros. Esta abordagem ainda pode ser separada em regressão ou classificação (Brandon Dan, 2020; Seify et al., 2022).

Alguns exemplos de algoritmos supervisionados são: *Linear Regression* (MLR), *Logistic Regression*, *Regressions Naive Bayes*, *Linear Discriminant Analysis*, *Support Vector Machines* (SVM), *Trees and Forests*, *Neural Networks(NN)* (Brandon Dan, 2020).

2.1.2 Aprendizagem Não Supervisionada

No que diz respeito à abordagem não supervisionada o algoritmo utiliza a informação disponibilizada que não é classificada nem rotulada e começa a agrupar a informação de acordo com semelhanças, padrões e diferenças. O sistema não identifica a saída adequada, mas descobre os dados e grava observações do conjunto de dados para encontrar padrões ocultos a partir de dados não marcados. É normal esta abordagem também ser chamada de "mineração de dados".

Alguns exemplos de algoritmos não supervisionados são: *K-Mean Clustering, K-Nearest Neighbor Clustering Hierarchical Cluster Analysis*, *Neural Networks* (Taher et al., 2019).

2.1.3 Aprendizagem Semi-Supervisionada

Os algoritmos semi-supervisionados são particularmente relevantes para cenários onde os dados rotulados são escassos. Nesses casos, pode ser difícil construir um classificador supervisionado confiável. Esta situação ocorre em domínios de aplicação em que os dados rotulados são caros ou difíceis de obter. Se não existirem dados rotulados suficientes e sob certas suposições sobre a distribuição dos dados, os dados não rotulados podem ajudar na construção de um classificador melhor. Na prática, os semi-supervisionados também são aplicados a cenários onde não existe falta significativa de dados rotulados: se os pontos de dados não rotulados fornecerem informações adicionais relevantes para a previsão, eles podem ser usados para obter um melhor desempenho de classificação (van Engelen et al., 2020).

2.1.4 Aprendizagem por reforço

Esta aprendizagem interage com o ambiente por ações e localiza erros ou recompensas. Ele permite que os sistemas e programas de software identifiquem o comportamento ideal num contexto específico para aumentar o desempenho, ou seja, obtém feedback das ações no ambiente e identifica o comportamento ideal a ter (Saravanan & Sujatha, 2019).

2.2 Trabalhos Relacionados

É fundamental comunicar de uma forma clara e organizada todo o processo de identificação e seleção de evidências para uma revisão sistemática de qualidade. Para tal optou-se por recorrer ao método PRISMA e adaptá-lo às necessidades. Utilizou-se o *template* "PRISMA 2020 flow diagram for new systematic reviews which included searches of databases and registers only" do site oficial e adaptou-se (*PRISMA*, n.d.).

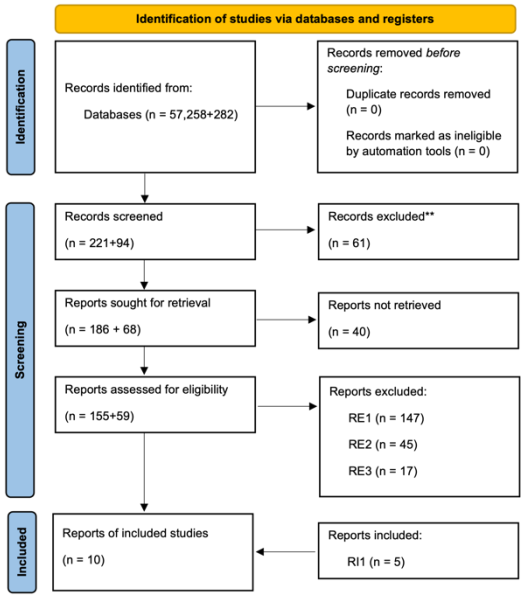


Figura 4 - Método PRISMA

Recorreu-se a dois repositórios de artigos científicos, *ScienceDirect* e ACM, resultando no final da revisão em 10 artigos científicos.

Na revisão usou-se a *query* abaixo, seguindo os critérios de inclusão:

- CI1 – O título inclui *Fraud*.
- CI2 – O Resumo inclui *Machine Learning*.
- CI3 – O artigo é de 2018 até ao presente.
- CI4 – Só artigos de pesquisa.

"query": { Title:(Fraud) AND Abstract:(Machine Learning) }
"filter": { Article Type: Research Article, E-Publication Date: (01/01/2018 TO 12/31/2023), ACM Content: DL }

Depois de reunir os artigos elegíveis usaram-se três critérios de exclusão:

- RE1 - Título do projeto incompatível
- RE2 - Resumo não incluído na pesquisa
- RE3 - Leitura rápida do projeto

Contudo, ao longo da revisão alguns artigos encontrados foram considerados interessantes, mas com a adição dos critérios de inclusão e restrição não foram encontrados na pesquisa. Neste caso optou-se por incluir no resultado da revisão. Esta adição encontra-se identificada como RI1 na última parte do prisma onde foram adicionados cinco artigos. Uma apatação efetuada ao templeta do PRISMA.

Ao longo dos artigos são mencionados diferentes algoritmos e técnicas de ML para a deteção de fraude. Para ter uma visão geral de todos mencionados foi construída a Tabela 1.

Tabela 1 - Algoritmos e técnicas mencionados nos artigos

Algoritmos	Artigo
Supervisionado	
Bayesian network	(van Belle et al., 2023) (Huang & Huang, 2019) (Sadgali et al., 2019) (Ryman-Tubb et al., 2018)
Bivariate Probit Model (BP)	(Sadgali et al., 2019)
CATCHM	(van Belle et al., 2023)
Classification and Regression Tree (CART)	(Madhurya et al., 2022) (Sadgali et al., 2019)
Convolutional Neural Network (CNN)	(Yuan et al., 2017)
Cost-sensitive Decision Tree (CSDT)	(Sadgali et al., 2019)
Decision Tree (DT)	(van Belle et al., 2023) (Madhurya et al., 2022) (Craja et al., 2020) (Huang & Huang, 2019) (Sadgali et al., 2019)
Evolutionary Computing	(van Belle et al., 2023)
Group Method of Data Handling (GMDH)	(Sadgali et al., 2019)
Hidden Markov Model	(Ryman-Tubb et al., 2018)
KNN and Other Clustering	(Aslam et al., 2022) (Madhurya et al., 2022) (Seify et al., 2022) (Huang & Huang, 2019) (Sadgali et al., 2019) (Ryman-Tubb et al., 2018) (Yuan et al., 2017)
Logistic Regression (LR)	(Aslam et al., 2022) (Madhurya et al., 2022) (Sadgali et al., 2019)
Naive Bayes Algorithm	(Aslam et al., 2022)

	(Madhurya et al., 2022)
	(Seify et al., 2022)
Probabilistic neural network (PNN)	(Sadgali et al., 2019)
Random Forest	(Aslam et al., 2022)
	(Madhurya et al., 2022)
	(Ryman-Tubb et al., 2018)
Support Vector Machine (SVM)	(van Belle et al., 2023)
	(Aslam et al., 2022)
	(Madhurya et al., 2022)
	(Seify et al., 2022)
	(Craja et al., 2020)
	(Sadgali et al., 2019)
	(Ryman-Tubb et al., 2018)
	(Yuan et al., 2017)
XGBOOST	(Aslam et al., 2022)
	(Madhurya et al., 2022)
Não supervisionado	
Association Rule Analysis (AR)	(Sadgali et al., 2019)
Case-based Reasoning (CBR)	(Ryman-Tubb et al., 2018)
Deep Autoencoder (DAE)	(Yuan et al., 2017)
Density Based Spatial Clustering of Applications with Noise (DBSCAN)	(Sadgali et al., 2019)
Neural Networks (ANNs/NNs)1	(van Belle et al., 2023)
	(Aslam et al., 2022)
	(Liu et al., 2022)
	(Madhurya et al., 2022)
	(Seify et al., 2022)
	(Craja et al., 2020)
	(Huang & Huang, 2019)
	(Sadgali et al., 2019)
	(Ryman-Tubb et al., 2018)
Outlier Detection Methods (OD)	(Sadgali et al., 2019)
Personalized PageRank (PPR)	(van Belle et al., 2023)
Self-organizing Map (SOM)	(Sadgali et al., 2019)
	(Ryman-Tubb et al., 2018)

[1] As *Neural Netwoks* podem também ser classificadas como algoritmos supervisionados, dependendo do tipo de tarefa a que se destinam.

Através da Tabela 1, conseguimos perceber que existem muitos algoritmos usados neste contexto, sendo a maior parte algoritmos supervisionados. Alguns dos algoritmos apresentados são derivados de outros, como é o exemplo das *Convolutional Neural Network*, que é uma classe de *Neural Network* e o *Deep Autoencoder* também um tipo de *Neural Network.*

Os modelos de ML podem ajudar a prever a alavancagem melhor do que os modelos lineares e a identificar um conjunto mais amplo de determinantes de alavancagem. Mostram uma precisão significativamente maior do que o modelo de risco discreto e que podem ser usados para resolver uma variedade de problemas de classificação diferentes na área das finanças (Aslam et al., 2022).

Alguns artigos mencionam modelos de *Neural Netwoks* (NNs) e KNN que são os mais precisos. As NNs foram usadas para identificar o padrão, as tendências dos mercados financeiros e usadas para detetar quanto o padrão do mercado mudou. Outros estudos apontam o *Random Forest* (RF) com melhor desempenho em comparação a *Logistic Regression* (LR) e *Support Vector Machine* (SVM) no âmbito de deteção de fraudes de cartão de crédito. O RF demonstrou ser excelente para deteção de fraudes, não apenas pelo desempenho, mas também pela sua facilidade de implementação e rápido tempo de computação, mesmo com grandes volumes de dados (Aslam et al., 2022).

Noutros casos, a *Decision Tree* também foi utilizada para detetar as fraudes de seguros prediais e notícias falsas. No caso de fraudes de cartões de crédito, o KNN demonstrou ter melhor desempenho em comparação com Naive Bayes, LR, RF, DT, SVM (Aslam et al., 2022).

Para o objetivo pretendido de deteção de fraude, os algoritmos geralmente usados são ANN, LR, DT e SVM (Madhurya et al., 2022).

Com uma ampla possibilidade de algoritmos de ML possíveis de usar na resolução do problema, depois de rever todas as comparações mencionadas anteriormente e os artigos foram escolhidos quatro algoritmos possíveis de usar-se no desenvolvimento (DT, SVM, NN, RF).

2.2.1 Decision Tree (DT)

Decision Tree (DT) é uma estrutura em árvore, onde cada nó representa um teste num atributo e cada ramificação representa um resultado do teste. Desta forma, a árvore tenta dividir as observações em subgrupos mutuamente exclusivos (Sadgali et al., 2019).

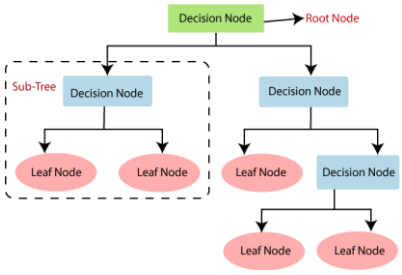


Figura 5 - Decision Tree (Viraj Lakshitha, 2021)

Em *Data Mining,* a DT é um modelo preditivo que pode ser usado para representar classificadores e modelos de regressão, podendo adotar outros nomes (*Classification Trees* e *Regression Tree* respetivamente) perante o uso destinado. Por outro lado, na pesquisa operacional, a DT representa um modelo hierárquico de decisões e consequências para identificar a estratégia com maior probabilidade de alcançar o objetivo pretendido (Rokach & Maimon, 2007).

As *Classification Trees* são úteis como técnica exploratória dos dados, para classificar um objeto ou uma instância num conjunto predefinido de classes com base nos seus valores de atributos. Frequentemente são usadas nas áreas das finanças, marketing, engenharia e medicina (Rokach & Maimon, 2007).

2.2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) usa um modelo linear para implementar limites de classe não lineares mapeando vetores de entrada não linearmente num espaço de recurso de alta dimensão. No novo espaço, um hiperplano ótimo de separação é construído (Sadgali et al., 2019).

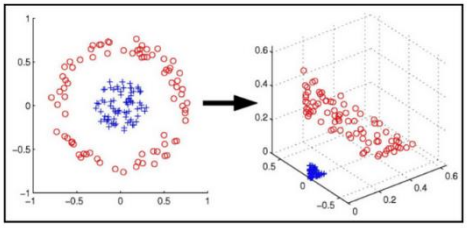


Figura 6 - Support Vector Machine (Zach Bedell, 2018)

SVMs foram desenvolvidos na ordem inversa ao desenvolvimento de redes neurais (NNs). SVMs evoluíram da teoria sólida para a implementação e experimentação, enquanto as NNs seguiram

um caminho mais heurístico, de aplicações e experimentação extensiva para a teoria (Wang, 2005) .

Inicialmente passaram despercebidas até 1992, quando obtiveram excelentes resultados em *benchmarks,* em reconhecimento de dígitos, visão computacional e categorização de texto. Na atualidade o SVM tem apresentado melhores resultados do que as NNs e outros modelos estatísticos nos problemas mais populares (Wang, 2005).

Para tarefas de classificação, função de decisão é representado pela fórmula (Lee & Verri, 2002):

$$\mathrm{y} = \mathrm{sing}\left(\sum_{i=1}^{T} y_i \alpha_i K(x, x_i) + b\right) \tag{1}$$

onde $x \in \mathbb{R}^d$ é o vetor de entrada d-dimensional de um exemplo de teste, $y \in \{-1, 1\}$ é um rótulo de classe, x_i é o vetor de entrada para o exemplo de treino i, y_i é seu rótulo de classe associado, T é o número de exemplos de treino, $K\ (x, x_i)$ é uma função *kernel* definida positiva e $\alpha = \{\alpha 1, \ldots, \alpha \mathrm{T}\}$ e b são os parâmetros do modelo.

Treinar um SVM consiste em encontrar α que minimize a função objetivo (Lee & Verri, 2002)

$$\mathrm{Q}(\alpha) = -\sum_{i=1}^{T} \alpha_i + \frac{1}{2}\sum_{i=1}^{T}\sum_{j=1}^{T} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{2}$$

sujeito às restrições

$$\sum_{i=1}^{T} \alpha_i y_i = 0 \tag{3}$$

e

$$0 \leq \alpha_i \leq \mathrm{C}\ \forall i \tag{4}$$

O *kernel* $K(x_i, x_j)$ pode ter diferentes formas, como a Função de Base Radial (RBF) (Lee & Verri, 2002)

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right) \tag{5}$$

com parâmetro σ.

Portanto, para treinar um SVM, deve-se resolver um problema de otimização quadrática, onde o número de parâmetros é T . Isto torna o uso de SVM para grandes conjuntos de dados difíceis de calcular $K(x_i, x_j)$ para cada par de treino exigiria cálculo de $O(T^2)$ e a resolução pode levar até $O(T^3)$. Observando que o estado da arte atual os algoritmos parecem ter escalonamento de complexidade de tempo de treino muito mais próximo de $O(T^2)$ do que $O(T^3)$ (Lee & Verri, 2002).

2.2.3 Neural Networks (ANN/NN)

As *Artificial Neural Networks* (ANNs), normalmente conhecidas como *Neural Networks* (NNs), são uma tecnologia madura com uma teoria estabelecida em áreas de aplicação reconhecidas. Uma NN consiste num número de neurónios, ou seja, unidades de processamento interconectadas. Associado a cada conexão está um valor numérico, chamado "peso" (Sadgali et al., 2019).

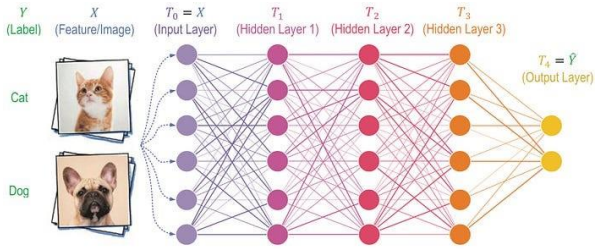


Figura 7 - Neural Networks

Uma NN simula o funcionamento dos neurónios de seres vivos, através de um conjunto interconectado de vários elementos de processamento. Os neurónios comunicam por impulsos (sinais elétricos) através de regiões responsáveis pela conexão e comunicação dos neurónios, chamada de sinapse. No caso de neurónios artificiais, são utilizados pesos para imitar as sinapses. Desta forma, cada valor de entrada representa a comunicação a passar e os pesos as regiões de sinapse, o que implica a multiplicação de cada valor de entrada pelo respetivo peso (Gurney, 2018).

Posteriormente, os valores obtidos nesta fase são somados e fornecidos a um nó com a função de ativação. Todas estas fases podem ser representadas pela seguinte fórmula (Gurney, 2018):

$$\alpha = \sum_{t=1}^{n} w_i x_i \tag{6}$$

Onde ∑ representa o somatório de 1 a n, com índice i, da multiplicação dos valores dos pesos w com os respetivos *inputs* x. O resultado desta fórmula, α, é o valor fornecido à função de ativação que de seguida é utilizado na comparação com o *threshold*.

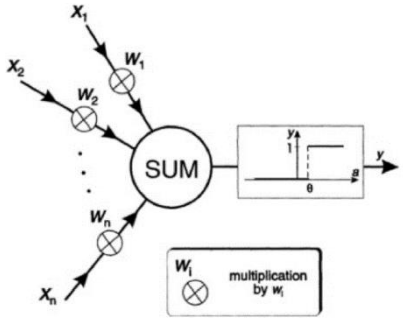


Figura 8 - Threshold logic unit (TLU) (Gurney, 2018)

Nesta comparação, conhecida como *threshold logic unit* (TLU), se o valor de ativação for maior ou igual ao valor de *threshold* o resultado é o valor 1, mas se o valor de ativação for menor que o *threshold* o resultado é o valor 0. Esta comparação pode ser representada da seguinte forma (Gurney, 2018):

$$y = \begin{cases} 1, & \alpha > \theta \\ 0, & \alpha < \theta \end{cases} \tag{7}$$

Onde y exprime o resultado, α o valor de ativação previamente obtido e θ o valor de *threshold* definido. A lógica aqui representada é depois usada em sistemas de neurónios que compõem as redes neuronais. Estas têm uma grande capacidade de processamento e essa capacidade é armazenada nos pesos obtidos durante o processo de aprendizagem (Gurney, 2018).

2.2.4 Random Forest (RF)

Random Forest (RF) representa várias DT aleatórias. Existem dois tipos de aleatoriedades na construção das árvores. Inicialmente, cada árvore é construída a partir de uma amostra aleatória dos dados e depois cada nó das árvores é selecionado um subconjunto de recursos aleatoriamente para gerar a melhor divisão (Houtao Deng, 2018).

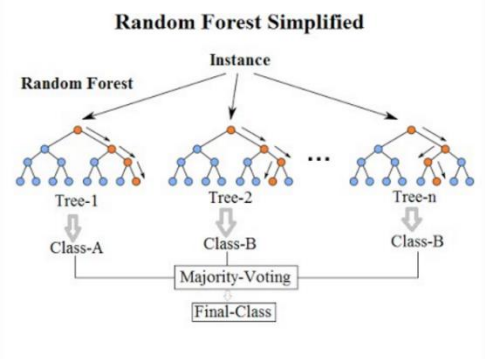


Figura 9 - Random Forest (Will Koehrsen, 2017)

Formalmente o RF é representado como uma coleção de classificadores estruturados em árvore $\{f(x, \theta k), k = 1, \dots\}$, onde x representa um vetor de entrada de valores preditores P usados para atribuir uma classe e k um índice para uma determinada "*tree*". Cada θk trata-se de um vetor aleatório construído para a k-ésima de modo que seja independente dos vetores aleatórios $\theta_1, \dots, \theta_{k-1}$ e é gerado a partir da mesma distribuição (Berk, 2008).

As DT geradas, geralmente são treinadas com o método "*bagging*". É por este método que são selecionadas aleatoriamente as observações e subconjuntos de preditores são apresentados sem substituição para cada divisão. Θk contém uma coleção de inteiros. A saída de um determinado classificador é uma classe atribuída para cada observação, determinada através dos valores de entrada x (Berk, 2008).

É usado um sistema de votação sobre o conjunto de classificadores para atribuir a cada observação uma classe, mas é importante distinguir uma classe atribuída pelo classificador k de uma classe finalmente atribuída pela votação (Berk, 2008).

3 Análise de valor

Neste capítulo irá ser abordada a análise de valor que contém métodos de avaliação, abrangendo o processo de desenvolvimento, a análise do valor do cliente, a avaliação das exigências do cliente e por fim a utilização de uma ferramenta de decisão para determinar a solução ideal a adotar. São apresentadas respostas às perguntas "O que é o produto?", "Quem são os clientes pretendidos?", "Que valor traz à empresa?" e "O que distingue o produto?".

3.1 Análise de valor

Análise de valor (VA) é uma metodologia concebida por Lawrence Miles, em 1945, com base na aplicação de técnicas de melhoria de processos de um produto com o objetivo de reduzir os custos, mas mantendo a qualidade.

"A redução de custos de componentes era uma maneira eficaz e popular de melhorar o "valor" quando a mão de obra direta e o custo do material determinavam o sucesso de um produto."(DRM Associates, 2016).

$$\text{Value} = (\text{Performance} + \text{Capability})/\text{Cost} = \text{Function}/\text{Cost} \quad (8)$$

Lawrence baseia-se em duas questões:

- "O que quer dizer quando diz que um produto tem boa qualidade?"
- "O que quer dizer quando diz que um produto tem um bom valor?"

Com o objetivo de desenvolver técnicas que mantivessem a qualidade, os fatores de segurança e as características do cliente, mas de forma mais eficiente e eficaz, promovendo custos adequados (L. D. Miles, 1962).

O mesmo refere análise de valor como “uma filosofia implementada pelo uso de um conjunto específico de técnicas, um corpo de conhecimento e um grupo de habilidades adquiridas, sendo que tem como principal objetivo a identificação eficiente de custos desnecessários” (L. D. Miles, 1962).

3.2 Fuzzy Front of Innovation e Modelo NCD

Todo o processo de inovação pode ser dividido em três partes: *Front End of Innovation* (FEI), *New Product and Process Development* (NPPD) e as Fases de Comercialização, todas influenciadas pelos mesmos fatores.

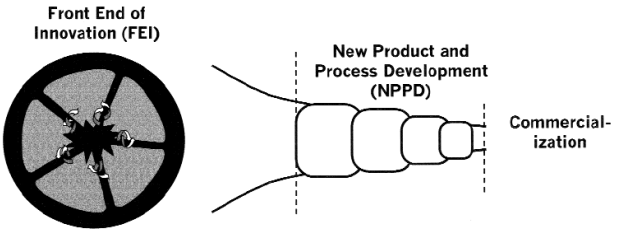


Figura 10 - Modelo de inovação

O FEI, surgiu da reformulação do *Fuzzy Front End* (FFE), é definido como atividades que antecedem o processo formal e bem estruturado do Desenvolvimento de Novos Produtos e Processos estruturados (NPPD). A forma circular do Desenvolvimento de Novos Conceitos (NCD) pretende sugerir que as ideias devem fluir e interagir entre todos os cinco elementos. Em contraste, a parte NPPD é ilustrada como uma série de etapas sequenciais, bem estruturadas e ordenadas cronologicamente (P. Koen et al., 2001).

Estabelecer um processo comum para descrever o FEI e melhorar o processo geral de inovação, modelo de NCD foi definido para fornecer uma visão adequada e também uma linguagem comum (P. Koen et al., 2001).

O FEI é definido como as atividades que precedem o NPPD.

Sendo as atividades do FEI muito diferentes das do NPPD, já que o FEI tende a ser imprevisível e desestruturado e o NPPD é um processo bem definido e estruturado, o NCD fornece uma abordagem cíclica que permite o alinhamento entre ambos processos.

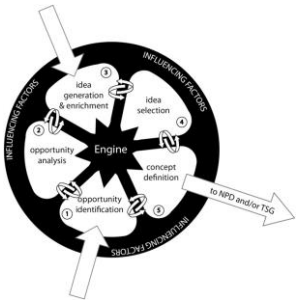


Figura 11 - O Novo Modelo de Desenvolvimento de Conceitos (NCD)(P. A. Koen et al., 2014)

O modelo NCD é dividido em três partes principais (P. A. Koen et al., n.d.):

- A parte interior define os cinco elementos-chave (identificação de oportunidade, análise de oportunidade, geração e enriquecimento de ideias, seleção de ideias e definição de conceito) do FEI.
- O elemento central denominado como motor que impulsiona os cinco elementos do FEI, através da cultura e liderança presente dentro da organização.
- Por último o elemento que consiste em fatores externos que influenciam o motor e por sua vez altera os cinco elementos do FEI, o NPPD e também a comercialização.

As subsecções que se seguem apresentam a definição dos cinco elementos do modelo NCD no contexto do projeto.

3.2.1 Identificação de oportunidade

Como referido anteriormente, no estudo "O Comércio Eletrónico em Portugal e na União Europeia" o número de *cibercompradores* aumentou mais do que a média da EU a partir de 2020 (*Pandemia Marca o Ritmo No Comércio Eletrónico - E-Commerce - Jornal de Negócios*, 2022).

A Ifthenpay não foi exceção a esse fenómeno, como podemos verificar na Figura 12, a nível de volume de pagamentos aumentou 24.3%, 26.7%,24.1% e o número de entidades aderentes aumentou 22.1%,19,1%,14,3 nos anos 2020,2021 e 2022 respetivamente. Também podemos observar na parte das entidades aderentes que a partir de 2020 houve um aumento significativo em relação aos anos anteriores (*Ifthenpay | Pagamentos Digitais Para a Sua Empresa*, n.d.).

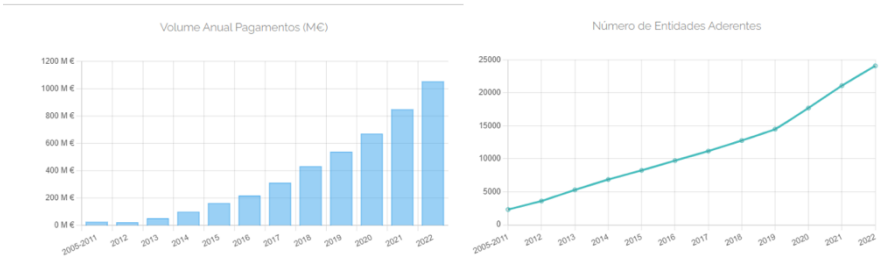


Figura 12 - Volume Anual Pagamentos e Entidades Aderentes

3.2.2 Análise de oportunidade

Com o aumento dos volumes de pagamentos e de métodos de pagamentos (*MB Way*, *Payshop* e Cartões de Crédito) para além das referências multibanco a análise dos pagamentos é cada vez mais exigente e complexa, devido a existir diferentes métodos e serem diferentes conjuntos de dados a analisar.

Uma análise feita pela empresa Feedzai, comparando os gastos e consumos antes e depois da pandemia covid-19, entre 2019 e 2021, em 18 mil milhões de transações bancárias globais, que correspondem a um aumento de 65%, os ataques fraudulentos deram um salto de 233%, um aumento substancial de risco (*Ataques Financeiros Online Dispararam 233% Durante Pandemia*, 2022).

Em Portugal o E-commerce bateu recordes de vendas em 2022, com uma previsão de crescimento entre 15% a 25%, atingindo aproximadamente um volume de 130 mil milhões de euros. De acordo com o banco de Portugal as tentativas de fraude desde 2019 aumentaram em 65%, e segundo a SIBS em 10 mil euros 14,50 euros são alvo de fraude (Paulo Marmé, 2022).

Com milhares de novas empresas de E-commerce criadas durante a pandemia foram criadas oportunidades para os fraudadores.

Numa pesquisa realizada pela Stripe, para 64% dos líderes empresariais globais, desde a pandemia ficou mais difícil combater fraudes devido aos tipos e volumes de fraudes. Para mais de 75% de empresas de B2C (*business to consumer*) implicou uma revisão manual e a necessidade de direcionar mais recursos para combater fraudes (*Stripe: Panorama Da Fraude Online.*, n.d.).

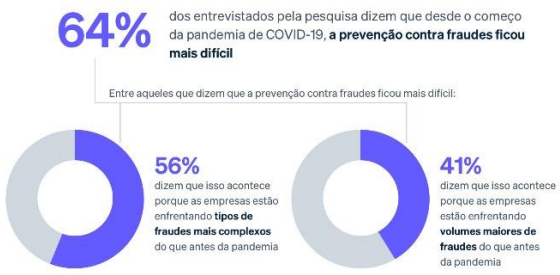


Figura 13 - Motivos de dificuldade de prevenção contra fraudes

A fraude tem impacto para além do prejuízo financeiro, obrigando a ampliar equipas de deteção de fraude ou a direcionar recursos de produtos ou engenharia para gerenciar as despesas gerais operacionais, removendo recursos valiosos do produto principal (*Stripe: Panorama Da Fraude Online.*, n.d.).

Figura 14 - Impacto das fraudes para além das perdas financeiras

As tentativas de fraude continuam cada vez mais sofisticadas, explorando novas formas de afetar empresas através da criação de grupos e interações entre eles, compartilhando os ataques e práticas comuns. Os ataques são a nível global, o que contribui ainda mais para novas formas de ataque e aumento dos grupos de partilha de informação dos ataques (*Stripe: Panorama Da Fraude Online.*, n.d.).

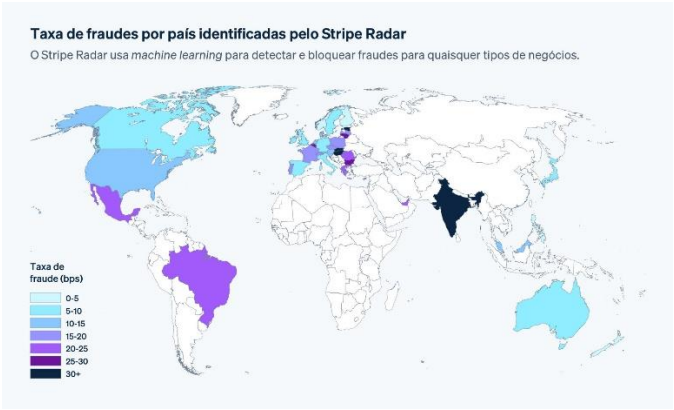


Figura 15 - Mapa de fraudes global

A Checkout.com destaca a importância do E-commerce utilizar portais de pagamento que estejam preparados para evitar fraudes online, na qual conseguiu economizar 1,95 bilião de dólares em possíveis perdas fraudulentas, em 2022 (Paulo Marmé, 2022).

Consequentemente também com a procura por serviços de E-commerce, as intenções maliciosas começaram a focar cada vez mais neste setor, levando a que os mecanismos já implementados tenham de ser revistos cada vez mais para estar a par de todas as novas tentativas fraudulentas para proteger os clientes e a empresa.

A implementação de mecanismos mais recentes, como ML, traz mais segurança e diminui o risco de fraudes, ao contrário dos mecanismos estatísticos.

Tabela 2 - Mecanismos Estatísticos VS Mecanismos ML

Mecanismos Estatísticos	Mecanismos com ML
Rápidos de implementar	Tempo de desenvolvimento elevado
Necessita de alterações face a novos padrões	Adaptação e deteção de novos padrões sem necessidade de refazer o modelo. Podendo treiná-lo novamente caso exija.
Falsos-positivos estáticos e repetidos ao longo do tempo	Capacidade de aprender com dados históricos e ajustar os falsos-positivos ao longo do tempo
Deteção unicamente do previsto	Deteção de irregularidades não previstas, anomalias, tendência e reconhecimentos padrões. Capacidade de gerar previsões precisas sobre eventos futuros.
Novos dados/métodos de pagamentos necessitam de revisão do modelo	Aprende com dados/métodos de pagamentos ao longo do tempo
Maiores custos e tempo com a análise de fraude	Redução de custos e tempo na análise de fraudes. Maior escalabilidade sem aumento das equipas de deteção de fraude
Tomada de decisão necessita de analise mais demorada e trabalhosa.	Tomadas de decisão otimizadas Automação de tarefas complexas que anteriormente exigiriam intervenção humana Ideal para grandes volumes de dados

3.2.3 Geração e Enriquecimento de Ideia

Após análise de todos os processos já implementados e discussão das necessidades, juntamente com os requerimentos do Banco de Portugal efetuou-se uma análise primaria dos dados disponibilizados para poder elaborar a possível abordagem.

Também se tentou procurar abordagens já realizadas para o efeito, mas grande parte da informação não se encontra disponível fora das empresas por isso não surtiu resultados desejados dessa pesquisa.

Durante a geração de ideia surgiu a possibilidade de poder também classificar os clientes por *merchant* e o seu nível de risco, através do histórico de pagamento. Isto é, verificar se para uma área específica de um cliente da Ifthenpay, é normal esse cliente cometer fraude aos seus próprios clientes ou existir reclamações/devoluções desse cliente à Ifthenpay.

Com o processo de geração de ideias ficaram as seguintes duvidas:

- Qual a melhor tecnologia a usar? (C#, R, Python)
- Será melhor agregar os métodos de pagamentos com a mesma estrutura de dados e utilizar técnicas para cada grupo ou juntar todos e aplicar a mesma técnica?
- O nível de risco por *merchant* encontra-se definido igualmente para todos os métodos ou só se aplica algum específico? Ou é diferente para cada um dos métodos?

3.2.4 Seleção de Ideias

Depois de geradas as ideias foi preciso selecionar aquelas que melhor se adequam aos objetivos definidos e responderam às questões levantadas. Para escolher a melhor tecnologia para se utilizar na aplicação foi utilizada a metodologia de decisão multicritério discreta. A metodologia escolhida foi o método de análise hierárquica (AHP- *Analytic Hierarchy Process*). Esta análise foi efetuada de forma a entender qual das tecnologias iria trazer mais valor à empresa. Nesse sentido, foram deliberados com a empresa um conjunto de critérios para utilizar nesta metodologia de forma a efetuar a melhor escolha.

Os critérios selecionados foram a funcionalidade, facilidade de utilização e a inovação.

Às restantes questões optou-se por serem respondidas à medida que se irá implementar as tecnologias e análise dos resultados obtidos.

3.2.5 Definição de Conceito

O objetivo do projeto será no final a empresa contar com o software de ML para poder obter informação de deteção de fraude que evoluirá com o tempo, em vez dos métodos tradicionais estatísticos. O mecanismo também poderá detetar padrões irregulares e com base no histórico atribuir o nível de risco de um novo cliente, como também a nível de uma transação ser fraude ou não. O software poderá analisar as transações em *real-time*, mas a prioridade será antes de a empresa efetuar as transações no dia seguinte ter todas as transações analisadas.

3.3 Proposta de valor

A proposta de valor representa os benefícios que uma organização consegue entregar aos elementos externos. Segundo Osterwalder e Pigneur é a forma como são agrupados um conjunto de itens de valor (como produtos ou serviços), de maneira que possa ser possível satisfazer as necessidades dos clientes. Uma proposta de valor serve para oferecer uma visão aos clientes de como os mesmo irão conseguir retirar valor de um produto ou serviço disponibilizado (*Perceived Value*), esta é efetuada de forma diferente para cada um dos segmentos de clientes e é o fator de diferenciação entre os competidores existentes no mercado (Osterwalder & Pigneur, 2003) .

Para a representação da proposta de valor foi utilizado o *Value Proposition Canvas* (VPC) idealizado por Osterwalder. VPC é uma feramente para esquematizar as ideias principais de um projeto, para entender as necessidades do cliente e como as satisfazer. É usado para comparar o que a solução proposta tem para oferecer e como ela contribui para a solução do problema.

O VPC permite uma visão geral dos "ganhos" e "dores" atuais que o cliente experimenta e os compara com os geradores de "ganhos" e "analgésicos" oferecidos pela solução. É constituído por 3 partes diferentes, primeiro temos o *Customer Profile* (perfil do cliente), onde é clarificado aquilo que é entendido ou percebido do cliente. A segunda parte, o *Value Map* (mapa de valor) é usado para descrever como se pretender criar valor para o cliente. Na parte central temos o *Fit* (encaixe), que simboliza o encontro das ambas as partes anteriores. Aqui conseguimos também obter informações mais especificas sobre os segmentos de clientes escolhidos (Osterwalder et al., 2014).

O ***Customer Profile*** é constituído por os seguintes componentes:

- ***Customer Jobs*** (Trabalhos dos clientes) – Informar os pagamentos fraudulentos. Atribuição de nível de riscos a novos clientes com base dos já existentes.
- ***Customer Pains*** (Dores dos clientes) – Diferentes estruturas de dados para diferentes métodos de pagamentos. Custos elevados em desenvolvimento e manutenção do modelo.
- ***Customer Gains*** (Ganhos dos clientes) – Maior nível de deteção de fraude e padrões irregulares. Resultados mais detalhados e certos. Diminuição de custos de equipa e análise de fraudes.

O ***Value Map*** é constituído por os seguintes componentes:

- ***Products and Services*** (Produtos e Serviços) – ML para Identificação de operações de fraude e nível de riscos por Merchant.
- ***Pain Relievers*** (Analgésicos) - Uso de ML para deteção de padrões não pré-concebidos.
- ***Gain Creators*** (Ganhos criados) - Deteção de padrões novos. Dados analíticos

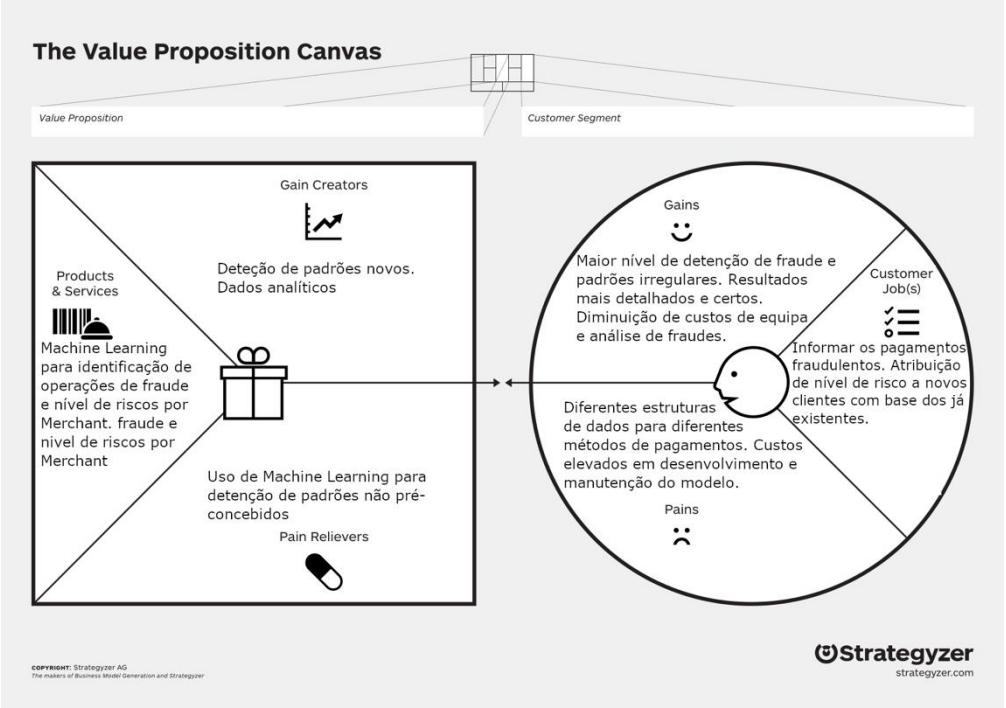


Figura 16 - Value Proposition Canvas

3.4 Analytic Hierarchy Process (AHP)

Para efetuar uma decisão primeiro é necessário estabelecer um conjunto de prioridades e organização das mesmas. Este raciocínio prévio é necessário para termos a certeza que realizamos a melhor escolha com a informação que despomos. Uma forma de organizar a ideia é por uma representação hierárquica, devido a isso foi decidido usar o método AHP, criado por Thomas Saaty. Este método baseia-se em álgebra linear, requerendo que os valores numéricos utilizados estejam de acordo com medições científicas. Segundo o autor o AHP é apropriado quando os critérios de decisão são algo subjetivos, abstratos e não quantificáveis (Saaty, 1988a, 1991).

O AHP pode ser dividido em sete ou oito fases dependendo dos autores, mas foi optado seguir as sete fases usadas por Saaty (Ammarapala et al., 2018; Nicola, n.d.).

3.4.1 Fase 1 - Construção da árvore hierárquica de decisão

É necessário definir o problema a resolver para de seguida se estruturar um diagrama hierárquico, isto é, decompor o problema/decisão numa hierarquia composta por no mínimo um objetivo, critérios e alternativas.

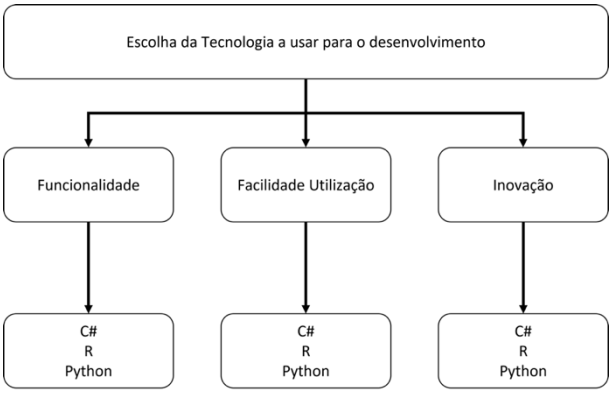


Figura 17 - Árvore hierárquica de decisão

Através da Figura 17 é possível perceber que o objetivo deste diagrama desenvolvido é escolher a tecnologia a usar no desenvolvimento do projeto.

Foram escolhidos 3 critérios para a comparação das mesmas, de forma que seja possível quantificar e tomar a decisão correta.

- Funcionalidade - Permitir que no desenvolvimento possa ser elaborado um software robusto e completo com todas funcionalidades possíveis sem limitações por parte da Linguagem de programação ou necessidade de uso e integração de softwares ou *Frameworks* externas.
- Facilidade de utilização – Entender a facilidade de utilização da Linguagem de programação, neste caso aquela que permite ao programador desenvolver de forma mais facilitada. Alguns dos fatores que indicação a facilidade são por exemplo: o tempo de aprendizagem e a existência de boa documentação ou falta da mesma e conhecimentos por parte da empresa bem como os conhecimentos disponibilizados.
- Inovação – Nos dias de hoje existem muitas librarias/Framework para as diferentes linguagens que ajudam no desenvolvimento e acrescem mais valor ao projeto.

3.4.2 Fase 2 - Comparação das alternativas e critérios

Consiste em estabelecer um nível de importância/prioridades entre os elementos para cada nível da hierarquia, por meio de uma matriz de comparação. Thomas Saaty definiu que a mesma deve comparar cada critério selecionada comparando as mesmas em grau de importância. Para tal, elaborou uma escala fundamental com atribuição de prioridade em forma numérica.

Desta atribuição deve resultar então uma matriz de comparação onde são comparados os elementos a esquerda desta matriz com os respetivos elementos superiores, sendo que numero de "julgamentos" de uma matriz de ordem n, calcula-se da seguinte maneira, n(n-1) /2, sendo n o número de elementos a comparar. De referir que existem situações especiais em que pode ser menor. O autor afirma ainda que é importante fazer a pergunta certa de modo que seja obtida a resposta certa (Saaty, 1988b).

Tabela 3 - Escala fundamental de AHP (Saaty & Katz, 1990)

Nivel de importância	Definição	Explicação
1	Igual Importância	Duas atividades contribuem de maneira igual para o objetivo
3	Moderada importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito Forte importância	Uma atividade é fortemente favorecida e a sua dominância é demonstrada na prática
9	Extrema importância	Uma atividade esta acima da outra com o maior grau de afirmação possível
2,4,6,8	Valores intermédios	Quando há necessidade de compromisso

Agora que definimos todos os parâmetros necessários e apresentamos os valores possíveis da escala Fundamental, iremos então proceder a realização da matriz de comparação com os critérios Funcionalidade, Facilidade de Utilização e Inovação em combinação com os valores da escala de Saaty.

Tabela 4 – Matriz de comparação com os critérios Funcionalidade

Critérios	Funcionalidade	Facilidade de Utilização	Inovação
Funcionalidade	1	3	2
Facilidade de Utilização	1/3	1	1/2
Inovação	1/2	2	1

3.4.3 Fase 3 - Prioridade relativa de cada critério

Consiste na atribuição de uma prioridade relativa a cada critério, sendo necessário normalizar os valores da matriz de comparações, colocando todos os critérios na mesma unidade. Para tal cada valor da matriz é dividido pelo total da respetiva coluna.

Tabela 5 - Matriz de comparação dos critérios do Segundo Nível

Critérios	Funcionalidade	Facilidade de Utilização	Inovação
Funcionalidade	1	3	2
Facilidade de Utilização	1/3	1	1/2
Inovação	1/2	2	1
Soma	11/6	6	7/2

Com o resultado da soma passamos a efetuar a matriz normalizada.

Tabela 6 - Matriz normalizada dos critérios do Segundo Nível

Critérios	Funcionalidade	Facilidade de Utilização	Inovação
Funcionalidade	6/11	1/2	4/7
Facilidade de Utilização	2/11	1/6	1/7
Inovação	3/11	1/3	2/7

Para identificar a importância de cada um dos critérios selecionados é necessário obter o vetor de prioridades, através do cálculo da média aritmética dos valores de cada linha da matriz normalizada.

Critérios	Funcionalidade	Facilidade de Utilização	Inovação	Prioridade Relativa
Funcionalidade	6/11	1/2	4/7	(∑=249/154) / 3 ≈ 0.54
Facilidade de Utilização	2/11	1/6	1/7	(∑=227/462) / 3 ≈ 0.16
Inovação	3/11	1/3	2/7	(∑=206/231) / 3 ≈ 0.30

Com a prioridade relativa calculada temos a ordem de importância de cada critério, Funcionalidade, Facilidade de Utilização e por último a Inovação.

3.4.4 Fase 4 - Avaliar a consistência das prioridades relativas

Para avaliar a consistência das prioridades relativas, é necessário calcular a Razão de Consistência (RC) para medir o quanto os julgamentos foram consistentes em relação a grandes amostras de juízos completamente aleatórios.

$$RC = \frac{IC}{IR} \tag{9}$$

Antes de obter o valor de RC é necessário calcular o índice de consistência (IC) e o índice de consistência aleatório (IR).

$$IC = \frac{\lambda_{max} - n}{n - 1} \tag{10}$$

No cálculo do IC, n representa o número de critérios a ser utilizados, neste caso 3, e o λ_{max} o valor próprio obtido através do uso do vetor próprio.

$$\begin{bmatrix} 1 & 3 & 2 \\ 1/3 & 1 & 1/2 \\ 1/2 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.54 \\ 0.16 \\ 0.30 \end{bmatrix} = \lambda_{max} \begin{bmatrix} 1.62 \\ 0.49 \\ 0.89 \end{bmatrix} \tag{11}$$

O λ_{max} é obtido pela média da divisão dos valores resultantes da equação (10) com o vetor próprio.

$$y_{max} = \overline{\times} \left\{ \frac{1.62}{0.54}, \frac{0.49}{0.16}, \frac{0.89}{0.30} \right\} = 3.01 \tag{12}$$

Com o λ_{max} calculado podemos achar o índice de consistência.

$$IC = \frac{3.01 - 3}{3 - 1} = 0.005 \tag{13}$$

Com o IC obtido, falta o IR, este valor é referente a um número de comparações par a par efetuadas. Este índice aleatório é calculado para matrizes quadradas de ordem n, onde existem algumas investigações que apontam diferentes tabelas. Saaty utiliza a tabela representada na Tabela 7, elaborada pelo Laboratório Nacional de Oak Ridge, nos EUA.

Tabela 7 - Indices de Consistencies (Ammarapala et al., 2018)

Tamanho da matriz	1	2	3	4	5	6	7	8	9	10
Índice de consistência aleatório (IR)	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Como as matrizes são de ordem 3 (n=3) o valor do IR é 0.58.

$$RC = \frac{0.005}{0.58} \approx 0.01 \tag{14}$$

Como o RC ≈ 0.01, podemos concluir então que 0.01 < 0.1, neste caso os valores das prioridades relativas atribuídas inicialmente estão consistentes.

3.4.5 Fase 5 - Construção da matriz de comparação paritária para cada critério, considerando cada uma das alternativas selecionadas

A matriz de comparação paritária consiste na determinação da prioridade relativa de cada critério, observando a importância relativa de cada uma das alternativas que compõem a estrutura hierarquia desenvolvida na primeira fase deste processo.

Tabela 8 - Matriz de comparação paritária do critério Funcionalidade

Funcionalidade	C#	R	Python
C#	1	1/3	1/5
R	3	1	1
Python	5	1	1
Soma	9	7/3	11/5

Tabela 9 - Matriz de comparação paritária do critério Funcionalidade normalizada

Funcionalidade	C#	R	Python	Prioridade Relativa
C#	1	1/3	1/5	(∑=23/15) / 3 ≈ 0.51
R	3	1	1	(∑=5) / 3 ≈ 1.8
Python	5	1	1	(∑=7) / 3 ≈ 2.34

Tabela 10 - Matriz de comparação paritária do critério Facilidade de Utilização

Facilidade de Utilização	C#	R	Python
C#	1	3	2
R	1/3	1	1/2
Python	1/2	2	1
Soma	11/6	6	9/2

Tabela 11 - Matriz de comparação paritária do critério Facilidade de Utilização normalizada

Facilidade de Utilização	C#	R	Python	Prioridade Relativa
C#	1	3	2	(∑=6) / 3 = 2
R	1/3	1	1/2	(∑=11/6) / 3 ≈ 0.61
Python	1/2	2	1	(∑=9/2) / 3 = 1.5

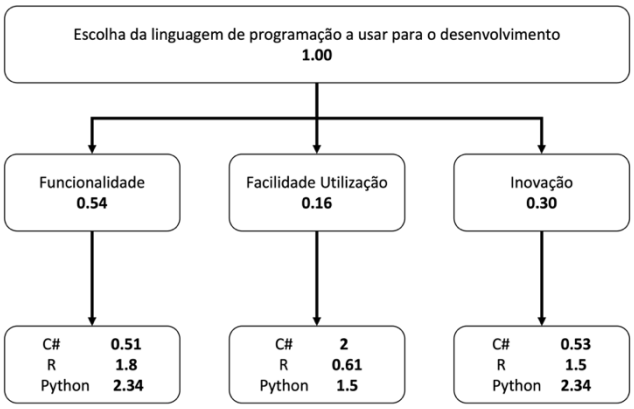
Tabela 12 - Matriz de comparação paritária do critério Inovação

Inovação	C#	R	Python
C#	1	1/3	1/4
R	3	1	1/2
Python	4	2	1
Soma	8	10/3	7/4

Tabela 13 - Matriz de comparação paritária do critério Inovação

Inovação	C#	R	Python	Prioridade Relativa
C#	1	1/3	1/4	(∑=19/12) / 3 ≈ 0.53
R	3	1	1/2	(∑=9/2) / 3 = 1.5
Python	4	2	1	(∑=7) / 3 ≈ 2.34

Com as matrizes calculadas, elabora-se novamente o diagrama em árvore hierárquica com os valores obtidos nas diferentes fases para melhor compreensão dos valores e conclusões sobre os mesmos.



3.4.6 Fase 6 - Obter a prioridade composta para as alternativas

Através dos valores obtidos na fase 5 no diagrama em árvore hierárquica, multiplicandos os valores pelas prioridades relativas obtemos as prioridades compostas das alternativas, resultando na tomada de uma decisão de qual alternativa escolher.

$$\begin{bmatrix} 0.51 & 2 & 0.53 \\ 1.8 & 0.61 & 1.5 \\ 2.34 & 1.5 & 2.34 \end{bmatrix} \times \begin{bmatrix} 0.54 \\ 0.16 \\ 0.30 \end{bmatrix} = \begin{bmatrix} 0.75 \\ 1.52 \\ 2.21 \end{bmatrix} \quad (15)$$

3.4.7 Fase 7 - Escolha da alternativa

$$\begin{bmatrix} 0.75 \\ 1.52 \\ \mathbf{2.21} \end{bmatrix} \quad (16)$$

Com os valores obtidos chegamos a conclusão de que a melhor linguagem de programação indicada para o desenvolvimento é Python. Em função dos critérios definidos (Funcionalidade, Facilidade de Utilização e Inovação) e das suas respetivas importâncias, esta foi a que obteve o valor mais alto de 2.21.

4 Análise e Desenho

Neste capítulo é apresentada uma análise e desenho da solução a conceber através da recolha de requisitos e identificação de possíveis alternativas de desenho. O capítulo é composto por uma identificação dos requisitos funcionais e não funcionais, a apresentação e descrição da arquitetura de cada modelo e alternativas de possíveis arquiteturas.

4.1 Requisitos Funcionais

O passo inicial da análise é definir os requisitos funcionais necessários para o software. Estes requisitos são parte das etapas anteriores.

Tabela 14 - Requisitos Funcionais

Requisitos funcionais	Descrição
R01 - Recolha de dados	Definir quais dados serão recolhidos, como serão armazenados e como serão pré-processados.
R02 - Preparação de dados	Limpar, normalizar e transformar os dados brutos num formato adequado para treinar um modelo.
R03 – Treino do modelo	Selecionar um algoritmo de *Machine Learning* adequado, definir parâmetros, treinar o modelo com dados de treino e avaliar a precisão do modelo.
R04 - Implementação do modelo	Integrar o modelo num sistema maior, como uma aplicação ou plataforma, e fornecer suporte para a manutenção e atualizações contínuas do modelo.
R05 - Monitorização do modelo	Monitorizar o desempenho do modelo em produção, detetar desvios de desempenho e atualizar o modelo ou os dados, se necessário.

R06 - Interpretação do modelo	Entender como o modelo toma decisões, quais recursos são importantes e como é que pode ser melhorado.
R07 - Nível de Risco do Cliente	Consultar o nível de risco de um determinado cliente.
R08 – Relatório de Resultados	Criação de um relatório, possível de impressão dos resultados de um determinado dia.
R09 – Histórico de Análises	Consultar dados processados num determinado dia e reprocessá-los novamente.
R10 – API	Comunicar com o software através de API para pedir dados (ex: Nível de Risco).
R11 – Autenticação e Perfil	Criação de Autenticação e secção só acessíveis a determinados perfis.

4.2 Requisitos Não Funcionais

Juntamente com os requisitos funcionais também é necessário definir os requisitos não funcionais. Eles são importantes para avaliar a qualidade do sistema. Para tal foi utilizado o modelo FURPS+, um modelo para classificação de atributos de qualidade de software, contendo as categorias de funcionalidade, usabilidade, confiabilidade, desempenho e suportabilidade. O sinal "+" permite ainda identificar requisitos extra como: desenho, implementação, interface e físicos.

Tabela 15 - Requisitos não funcionais (modelo FURPS+)

Classificação FURPS+	Requisitos não funcionais
RN01 – Funcionalidade (segurança)	Garantir que a informação confidencial e privada só é acessível através de autenticação específica.
RN02 - Usabilidade	Garantir que o sistema satisfaz os seus utilizadores e é eficaz (cumpre os objetivos) e eficiente (cumpre os objetivos com o mínimo esforço possível)
RN03 - Confiabilidade	Garantir que o sistema está pronto para falhas e consegue recuperar dos mesmos (tentar novamente obter um recurso que no momento se encontrava indisponível).
RN04 – Desempenho	Análise de todas transações até ao dia seguinte às 8h. Tempo de resposta da API no máximo de 30 segundos.
RN05 - Suportabilidade	Suporte para os browsers mais utilizados (Google Chrome, Microsoft Edge, Firefox).

4.3 Arquitetura do Modelo

Após descritos os requisitos e casos de uso fundamentais da solução é importante pensar nas possíveis arquiteturas que o modelo pode adotar perante aquilo que foi identificado no estado da arte.

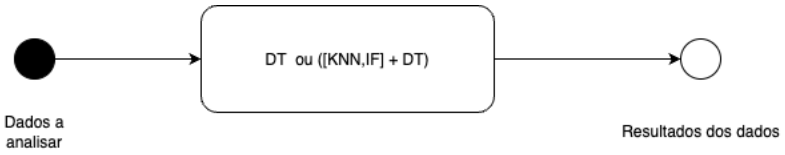


Figura 18 - Abordagem da arquitetura do modelo

O *input* representa os dados que se pretendem analisar, por exemplo os pagamentos do dia anterior. O resultado, *output*, contém a probabilidade de cada pagamento ser *outlier*. O modelo conta com três algoritmos, mas para cada tipo de dados (transferências e clientes) é usado um conjunto específico em vez de passar por todos os algoritmos.

4.3.1 Decision Tree (DT)

Como mencionado anteriormente na secção 2.2.1, *Decision Tree* (DT) é uma estrutura em árvore, onde cada nó representa um teste num atributo e cada ramificação representa um resultado do teste. Desta forma, a árvore tenta dividir as observações em subgrupos mutuamente exclusivos (Sadgali et al., 2019).

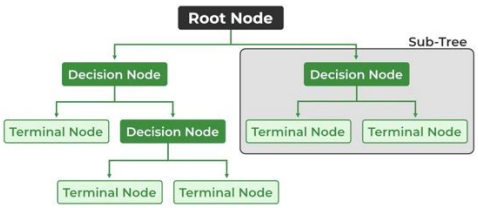


Figura 19 - Decision Tree (*Decision Tree - GeeksforGeeks*, n.d.)

Em *Data Mining,* a DT é um modelo preditivo que pode ser usado para representar classificadores e modelos de regressão, podendo adotar outros nomes (*Classification Trees* e *Regression Tree* respetivamente) perante o uso destinado. (Rokach & Maimon, 2007).

As *Classification Trees* são úteis como técnica exploratória dos dados, para classificar um objeto ou uma instância num conjunto predefinido de classes com base nos seus valores de atributos. (Rokach & Maimon, 2007).

Uma DT é uma das ferramentas mais poderosas de algoritmos de aprendizagem supervisionada usados para classificação e regressão. Representa uma estrutura de árvore semelhante a um fluxograma onde cada nó interno denota um teste em um atributo, cada ramo representa um

resultado do teste e cada nó de folha (nó terminal) contém um rótulo de classe. Ele é construído dividindo recursivamente os dados em subconjuntos com base nos valores dos atributos até que um critério de paragem seja atendido, como a profundidade máxima da árvore ou o número mínimo de amostras necessárias para dividir um nó. Durante o treino, o algoritmo DT seleciona o melhor atributo para dividir os dados com base em duas métricas, Entropia ou Impureza de Gini, que medem a aleatoriedade nos subconjuntos ou nível de impureza respetivamente. O objetivo é encontrar o atributo que maximiza o ganho de informação ou a redução da impureza após a divisão (*Decision Tree - GeeksforGeeks*, n.d.).

A entropia para um do conjunto de dados com número K de classes para o iésimo nó pode ser definido como:

$$H_i = -\sum_{k \in K}^{n} p(i,k) \log_2 p(i,k) \tag{17}$$

Onde:

- S é o exemplo de conjunto de dados.
- k é a classe particular das classes K
- p(k) é a proporção dos pontos de dados que pertencem à classe k em relação ao número total de pontos de dados na amostra S do conjunto de dados.
- p(i,k) não deve ser igual a zero.

Vantagem da Entropia (*Decision Tree - GeeksforGeeks*, n.d.):

- 0 quando o conjunto de dados é completamente homogéneo, o que significa que cada instância pertence à mesma classe. Quanto mais baixa for a entropia menor é a incerteza na amostra do conjunto de dados.
- Quando o conjunto de dados é dividido igualmente entre várias classes, a entropia está no seu valor máximo. Portanto, a entropia é maior quando a distribuição de rótulos de classe é uniforme, indicando incerteza máxima na amostra do conjunto de dados.
- A entropia é usada para avaliar a qualidade de uma divisão. O objetivo da entropia é selecionar o atributo que minimiza a entropia dos subconjuntos resultantes, dividindo o conjunto de dados em subconjuntos mais homogéneos em relação aos rótulos de classe.
- O atributo de maior ganho de informação é escolhido como critério de divisão (ou seja, a redução da entropia após a divisão desse atributo), e o processo é repetido recursivamente para construir a árvore de decisão.

A impureza ou índice de Gini é um *score* que avalia a precisão de uma divisão entre os grupos classificados. O *score* é representado no intervalo entre 0 e 1, onde 0 é quando todas as observações pertencem a uma classe, e 1 é uma distribuição aleatória dos elementos dentro das classes. Neste caso, queremos ter uma pontuação de índice de Gini o mais baixa possível.

$$Gini\ Impurity = 1 - \sum p_i^2 \tag{18}$$

Tabela 16 - Vantagens vs Desvantagens da *Decision Tree* (*Decision Tree - GeeksforGeeks*, n.d.)

Vantagens	Desvantagens
É simples de entender, pois segue o mesmo processo que um ser humano segue ao tomar algumas decisões na vida real. Pode ser muito útil para resolver problemas relacionados com a decisão. Ajuda a pensar em todos os resultados possíveis para um problema. Há menos exigência de limpeza de dados em comparação com outros algoritmos.	Pode ter um problema de sobre ajuste (*overfitting*), que pode ser resolvido usando o algoritmo *Random Forest*. A árvore de decisão contém muitas camadas, o que a torna complexa. Para mais rótulos de classe, a complexidade computacional da árvore de decisão pode aumentar.

Também existe outra métrica, *log_loss* que pode ser usada como critério para uma DT, mas que não foi contemplada como critério da DT no projeto.

4.3.2 K-nearest neighbors (KNN)

K-nearest neighbors é um método de aprendizagem supervisionado não paramétrico desenvolvido em 1951 por Evelyn Fix e Joseph Hodges e adaptado por Thomas Cover. É usado para classificação e regressão.

No caso de classificação, a saída é uma associação de classe. O objeto é classificado por um voto dos seus vizinhos, com o objeto de atribuído à classe mais comum entre os seus vizinhos mais próximos (k é um inteiro positivo, tipicamente pequeno). Se k = 1, então o objeto é simplesmente atribuído à classe desse único vizinho mais próximo.

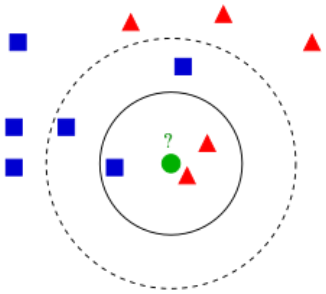


Figura 20 - Exemplo de classificação KNN (*K-Nearest Neighbors Algorithm - Wikipedia*, n.d.)

No outro caso, da regressão, a saída é o valor da propriedade do objeto. Este valor é a média dos valores de k vizinhos mais próximos. Se k = 1, então a saída é simplesmente atribuída ao valor desse único vizinho mais próximo.

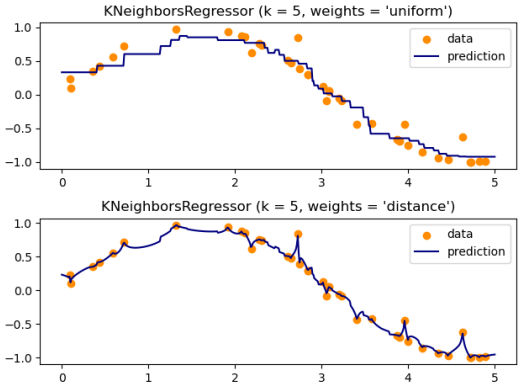


Figura 21 - Exemplo de regressão KNN (*Nearest Neighbors Regression — Scikit-Learn 1.2.2 Documentation*, n.d.)

A distância até o k-ésimo vizinho mais próximo também pode ser vista como uma estimativa de densidade local e, portanto, também é um *score* de *outlier* frequente na deteção de anomalias. Quanto maior a distância para o KNN, menor a densidade local, maior a probabilidade de o ponto de consulta ser um *outlier*.

4.3.3 Isolation Forest (IF)

Isolation Forest (IF) é uma técnica utilizada para identificar *outliers* em dados. Foi introduzida por Fei Tony Liu e Zhi-Hua Zhou em 2008. A abordagem consiste em árvores binárias para detetar anomalias, resultando numa complexidade de tempo linear e baixo uso de memória que é adequado para processar grandes conjuntos de dados. Cada árvore é criada particionando as instâncias recursivamente, selecionando aleatoriamente um atributo e um valor dividido entre os valores máximo e mínimo do atributo selecionado. Cada fase de divisão representa um nó em uma árvore e o número de partições necessárias para isolar um ponto é equivalente ao comprimento do caminho do nó raiz até um nó final. Desde a sua introdução, o *Isolation Forest* ganhou popularidade como um algoritmo rápido e confiável para deteção de anomalias em vários campos, como segurança cibernética, finanças e pesquisa médica (*Isolation Forest | Anomaly Detection with Isolation Forest*, n.d.; Meira et al., 2019).

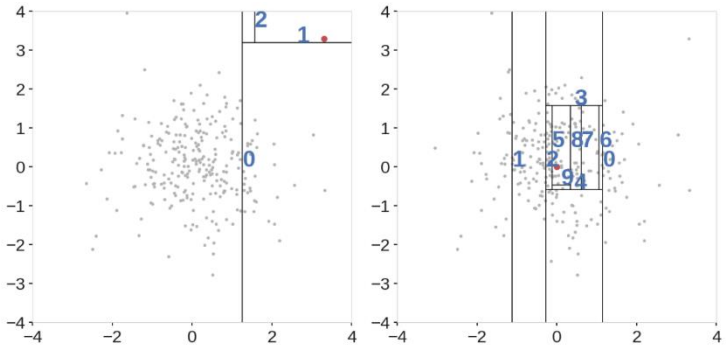


Figura 22 - Exemplo de *Isolation Forest*

4.3.4 Modelos pré-treinados

A aplicação de modelos pré-treinados é uma alternativa. No entanto, não foram encontrados modelos públicos. Os benefícios das alternativas anteriores estudadas não eram contemplados, existia a necessidade de mais recursos e os resultados não eram garantidos devido a problemas como *underfit* da aprendizagem por transferência. Deste modo, esta alternativa foi excluída e optou-se por testar e avaliar as outras duas alternativas.

4.4 Fluxo de deteção de fraude

Com os modelos estudados, foi idealizado um fluxo do funcionamento da deteção de fraudes e *outliers*.

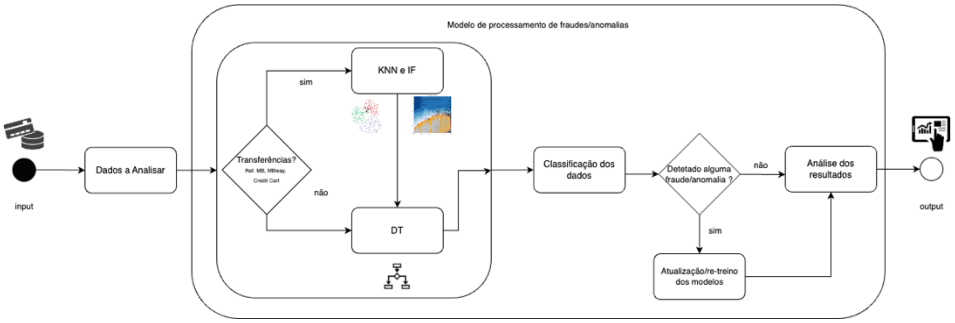


Figura 23 - Fluxo de deteção de fraudes/anomalia

Como representado na Figura 23 o *input* são os dados que se pretendem analisar. Esses dados encontram-se noutro sistema à parte e quando são recebidos são classificados previamente pelo tipo de dados (Clientes, Transferências e TransferênciasMB) para serem atribuídos aos algoritmos a aplicar.

Depois dos algoritmos processarem os dados é atribuído o nível de probabilidade de determinados dados serem fraude/*outlier*. Caso algum dos dados seja classificado como fraude/*outlier* será desencadeado outro processo para recalcular o nível de risco por *merchant*, de outra forma são processados e analisados todos os resultados para compilar o relatório final.

4.5 Alternativas de Arquitetura

Existem diferentes abordagens que se podem escolher para a arquitetura da solução. Uma possível solução encontra-se representada na Figura 24.

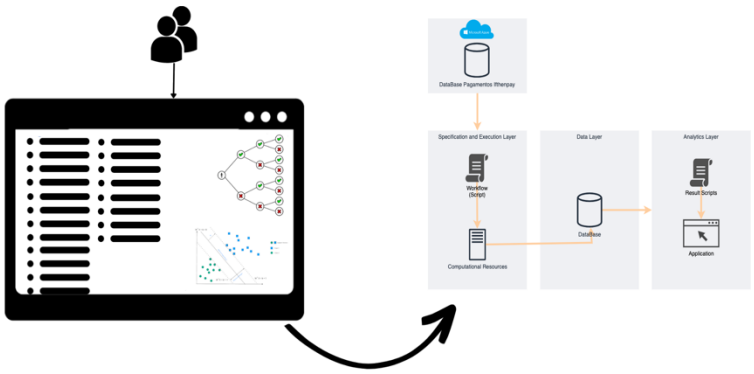


Figura 24 - Arquitetura Alternativa

A solução encontra-se num sistema à parte de todos os já implementados. Os dados são importados a partir da Base de Dados (BD) de pagamentos da Ifthenpay que se encontram na *cloud Azure* para o sistema. Essa importação passa por pré-tratamento dos dados e consequentemente é guardada na BD da aplicação. Posteriormente esses dados são analisados pelo modelo representado na Figura 18 e é apresentado um relatório detalhado de todas as conclusões. Este projeto não contempla esta arquitetura, mas já foi desenvolvido com a ideia de ser implementada no futuro. Como irá ser apresentado mais à frente, nas secções 5.1.1 e 7.3, foram feitos alguns desenvolvimentos que poderão ser aproveitados para esta arquitetura, desde o uso de espaço do HDD para processamento em vez de RAM e o armazenamento dos modelos depois do seu treino , questão importante para a abordagem do *Concept Drift.*

5 Implementação

Nesta secção é descrito o processo de implementação deste projeto passando por diversas etapas como instalação e configurações das ferramentas necessárias, experimentação e aprendizagem dessas ferramentas e modelos necessários, construção e descrição dos *dataset*, processamento dos dados do *dataset*, criação e treino dos modelos e utilização dos mesmos para deteção de fraude.

5.1 Instalação

5.1.1 Máquina

Para o desenvolvimento e implementação foi analisada qual a máquina a ser utilizada, devido à capacidade de armazenamentos dos dados (disco), processamentos (CPU e memória RAM) e criação e treino do modelo (GPU).

Foram exploradas as máquinas virtuais da *Microsoft Azure*, visto que a empresa já trabalha com as mesmas.

Instância	vCPU(s)	RAM	Temporário armazenamento	GPU	Pay as you go com AHB
NC4as T4 v3	4	28 GiB	180 GiB	1X T4	€358,6754/mês
NC8as T4 v3	8	56 GiB	360 GiB	1X T4	€512,7831/mês
NC16as T4 v3	16	110 GiB	360 GiB	1X T4	€820,9986/mês
NC64as T4 v3	64	440 GiB	2880 GiB	4X T4	€2967,5961/mês

Figura 25 - Custo das máquinas virtuais na *Microsoft Azure*

Foram analisadas as máquinas virtuais da série NCas_T4_v3, concebidas especificamente para as cargas de trabalho de IA e *Machine Learning*.(*Preços - Máquinas Virtuais Do Windows | Microsoft Azure*, n.d.)

Como pode ser observado na Figura 25, os preços das mesmas eram elevados, como tal a empresa optou por só usar as máquinas do *Azure* depois de validar a utilidade do projeto desenvolvido.

De seguida, foram comparadas as máquinas físicas disponíveis de modo a selecionar qual a utilizar.

Tabela 17 - Comparação de Máquinas físicas

	Processador	*RAM*	Memória disco	Placa Gráfica
Fixo	Intel® Core™ i7-7700K 4.20GHZ	16 GB	SSD 232 GB SSD 465 GB HDD 929 GB	ASUS Rog Strix GeForce GTX 1060 OC
Portátil	Apple M1	16 GB	SSD 245	Integrated 8-core GPU

Foram comparados os tempos de processamento de cada máquina com diferentes partes do projeto e a diferença entre estes não foi relevante no desenvolvimento inicial. No entanto à medida que se foi aumentando a quantidade de dados o portátil apresentou melhores resultados face ao fixo. Sendo sistemas operativos (SO) diferentes, o portátil com os dados todos conseguiu gerir a RAM melhor que o fixo, principalmente no cálculo da *silhouette score*, o fixo não conseguiu gerir as 70GB que o modelo requeria, sendo necessário proceder a configurações extra. Por isso, os resultados foram executados com portátil. Foram feitos alguns testes recorrendo *tempfile.mktemp* para usar espaço do HDD em vez do SSD, por sua vez a nível de processamento o portátil continuava a ser superior e como consequentemente o mesmo lidava com grandes quantidades de RAM não foi explorada essa situação. Outro teste efetuado foi o de guardar os modelos depois do seu treino recorrendo a *joblib*. Esta situação não representou grande interesse devido ao uso do *notebook,* o modelo ficava guardado em

memória e podia-se continuar com as experiências sem necessitar de o treinar. Esta situação pode-se tornar relevante na aplicação do *Concept Drift* como abordado na secção 7.3.

5.1.2 Python

Para utilizar a linguagem *Python* e as funcionalidades associadas foi utilizada a versão mais recente disponibilizada no site da mesma (*Download Python | Python.Org*, n.d.). Para o projeto foi utilizada a versão 3.11.3.

5.1.3 Notebooks

Como o desenvolvimento deste projeto consiste em análise e exploração dos dados e nos seus resultados, usar diretamente *scripts* de *Python* não seria muito percetível e intuitivo devido a ter de executar os mesmos só por uma linha do código. Para tal, foram analisadas três alternativas.

Os *notebooks* do *Google Colab* ou "*Colaboratory*", permitem escrever e executar algoritmos de IA em código *Python* no navegador, sem necessidade de configurações da máquina, permitindo ter acesso a GPU sem custos e guardar todo o desenvolvimento e partilhá-lo (*Damos-Lhe as Boas-Vindas Ao Colaboratory - Colaboratory*, n.d.). No entanto, foram detetados alguns problemas com os mesmos após algumas utilizações. Estes ficavam em *standby* após algum tempo sem atividade, o que dificultava tarefas necessárias durante a implementação, como o treino de modelos. Mas devido à natureza dos dados sensíveis usar o Google para os processar levantava algumas questões de RGPD e sigilo bancário.

Os *Jupyter Notebooks*, apesar de não fornecerem GPU remotos, requererem instalações e configurações na máquina local, mantêm-se ativos e funcionais durante longos períodos sem atividade e permitem manter os dados protegidos. Mas sendo só uma aplicação no navegador e como o objetivo é ter uma aplicação em execução, o seu desenvolvimento em *Jupyter Notebooks* tornar-se-ia complicado.

Tendo todos estes fatores em conta, foi decidido usar *Jupyter Notebooks* nativamente no *VS Code* pois a instalação é mais fácil e podemos contar com inúmeras extensões para o desenvolvimento (Working with Jupyter Notebooks in Visual Studio Code, n.d.).

5.1.4 Ambiente virtual

Ao utilizar o *VS Code* é possível configurar o *kernel* a utilizar, para tal foi criado um ambiente virtual na pasta do projeto e adicionado como *kernel* dos *notebooks*.

```
python3 -m pip install virtualenv
python3 -m virtualenv .venv

#MacOs
source .venv/bin/activate

#Windowns
.\.venv\Scripts\activate
```

Código 1 - Criação de ambiente virtual

Após todas as instalações necessárias, foi possível instalar no ambiente virtual todas as bibliotecas necessárias, utilizando o excerto de Código 2 apresentado abaixo.

```
# MacOs
echo "export LANG=pt_PT.UTF-8" >> .venv/bin/activate
source .venv/bin/activate
##change locale
export LANG="pt_PT.UTF-8"
pip install --upgrade pip

# Windowns
echo "export LANG=pt_PT.UTF-8" >> .venv/bin/activate
.\.venv\Scripts\activate
##change locale
set LANG="pt_PT.UTF-8"
python -m ensurepip # force pip install
python -m pip install --upgrade pip

#Install all Dependences
pip install scikit-learn
pip install -r requirements.txt
```

Código 2 - Instalação das bibliotecas

Dependendo do SO os comandos a executar são diferentes. Para não existir problemas com o formato das datas foi forçado a *locale* para "pt_PT.UTF-8" e atualizado o *pip*. Todas as bibliotecas necessárias para o projeto desenvolvido como também algumas para os desenvolvimentos futuros, como API e *User Interface*, encontram-se no ficheiro requirements.txt apresentado no Anexo B.

5.2 Experimentação

Numa fase inicial foram realizados alguns testes com exemplos de modo a adquirir conhecimento e experiência das ferramentas existentes para a linguagem selecionada bem

como o seu potencial. Foram recolhidos exemplos de DT e KNN e alguns exemplos de deteção de fraude de cartão de crédito usando *Python*.[2]

Esta fase foi bastante importante, já que permitiu obter as bases necessárias para o desenvolvimento deste projeto e ainda explorar e aprender mais sobre esta área.

Como o objetivo final consiste na apresentação de uma *Decision Tree* com os resultados das previsões de um determinado cliente ou transação, as mesmas são apresentadas na secção 6.4 devido a servirem também como avaliação dos respetivos modelos, principalmente no caso das transações onde não existe uma *label* objetivo depois do treino do modelo, sendo só possível avaliar o seu resultado com a construção de uma DT.

Ao longo da Experimentação, foram recolhidos alguns tempos dos processamentos, este resumo está presente no Anexo F. Os tempos apresentados podem ser diferentes noutras execuções. Os mesmos variaram dependendo do trabalho que estava a ser executado no portátil, nem sempre estava só a ser usado para execução isolada de uma tarefa e não contemplam todos os passos do início ao fim. Desta forma os tempos apresentados são só uma referência.

5.2.1 DataSet

Depois de realizar os testes na fase de experimentação e antes de implementar a solução foram aprofundados os *datasets* na fase de *Data Preparation* para construir os *datasets* finais a serem utilizados. No final foram elaborados 3 *datasets* (Clients, Transactions, TransactionsMB).

Tabela 18 – *DataSets* Descrição

	Clients	Transactions	TransactionsMB*
N.º Colunas	5	12	12
N.º Linhas	24305	244517	6773170
Tipos de Dados	1 numérico	5 numéricos	6 numéricos
	4 texto	7 texto	6 texto
Algoritmos Aplicados	Decision Tree	[KNN, IF] + Decision Tree	[KNN, IF] + Decision Tree

**DataSet* analisado, mas não implementado devido ao seu tamanho, mas com o mesmo princípio do *Transaction*.

Todos os dados encontram-se guardados numa BD no *Azure* sendo carregados para a aplicação através de uma classe que gere todas as conexões. Foram adicionados dois *datasets* (países e CAE) que podem ser consultados no Anexo A, para facilitar a análise dos dados. O *dataset* dos países serve para agrupar os clientes por continente, fazendo a ligação entre os países e os respetivos continentes. Relativamente ao *dataset* dos CAE, este destina-se a converter o CAE de um determinado cliente na respetiva secção, um grupo com menos detalhe.

[2] (Credit Card Fraud Detection Using Machine Learning & Python | by Pranjal Saxena | Towards Data Science, n.d.; Decision Trees — Scikit-Learn Documentation, n.d.; Nearest Neighbors — Scikit-Learn Documentation, n.d.)

5.2.1.1 Clients

O *dataset clients* na fase de *Data Preparation,* passou por uma fase de limpeza e tratamento para retirar colunas que não eram necessárias e preencher os valores que estavam como NA e NULL, bem como a normalização de alguns países que se encontravam escritos de maneiras diferentes (ex.: Suiça-> Suíça).

```
Dataset Length:  24305
Dataset Shape:  (24305, 10)
Dataset Head:      cod         localidade nif dataregisto risco  estado    cae setorinstituicional      pais
0    1              Cascais   1 2008-07-08  None  Activo   None                S14  Portugal     NaN
1    2                Faro    5 2008-05-28  None     N/C  47784                   Portugal     NaN
2    3  Vila Nova Famalicão   5 2008-07-10  None  Activo  62090                S11  Portugal     NaN
3    4            Angustias   5 2008-06-25  None  Activo   None                S11  Portugal     NaN
4    5            Bombarral   5 2008-06-25  None  Activo  62020                S11  Portugal     NaN
Dataset Types:  cod                          int32
localidade                  category
nif                           object
dataregisto            datetime64[ns]
risco                         object
estado                      category
cae                           object
setorinstituicional         category
pais                        category
fraude                       float64
dtype: object
```

```
Dataset Length:  24305
Dataset Shape:  (24305, 5)
Dataset Head:     estado setorinstituicional  fraude cae_seccao Continente
0  Activo                S14       0          -     Europe
1     N/C                            0          G     Europe
2  Activo                S11       0          J     Europe
3  Activo                S11       0          -     Europe
4  Activo                S11       0          J     Europe
Dataset Types:  estado                 category
setorinstituicional    category
fraude                    int32
cae_seccao             category
Continente             category
dtype: object
```

Figura 26 - *Dataset Clients* Original vs. Transformado

O *dataset* original contava com dez colunas de diferentes tipos como pode ser visto na Figura 26 do lado esquerdo. No final o *dataset* ficou com cinco colunas, quatro do tipo categoria/texto e uma do tipo inteiro.

	fraude
count	24305.000000
mean	0.001193
std	0.034522
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

	estado	setorinstituicional	fraude	cae_seccao	Continente
0	Activo	S14	0	-	Europe
1	N/C		0	G	Europe
2	Activo	S11	0	J	Europe
3	Activo	S11	0	-	Europe
4	Activo	S11	0	J	Europe
...	...	...	...	...	...
4300	Activo	S11	0	G	Europe
4301	Activo	S14	0	Q	Europe
4302	Activo	S14	0	M	Europe
4303	Activo	S14	0	G	Europe
4304	Activo	S11	0	R	Europe

Figura 27 - Estatísticas descritivas *DataSet* Cliente

Logo na fase inicial observou-se uma situação que mais tarde veio-se a verificar um problema para o modelo, a existência de poucos registos marcados como fraude em comparação com o tamanho do *dataset*, 29 para 24305, sendo esta a *label* objetivo. Na secção 5.3 é apresentado como foi analisada esta situação.

5.2.1.2 Transactions e TransactionsMB

Os *dataset Transactions* e *TransactionMB* são os principais deste trabalho, por representarem a informação de todas as transferências e serem a melhor forma para detetar fraude ou *outlier*.

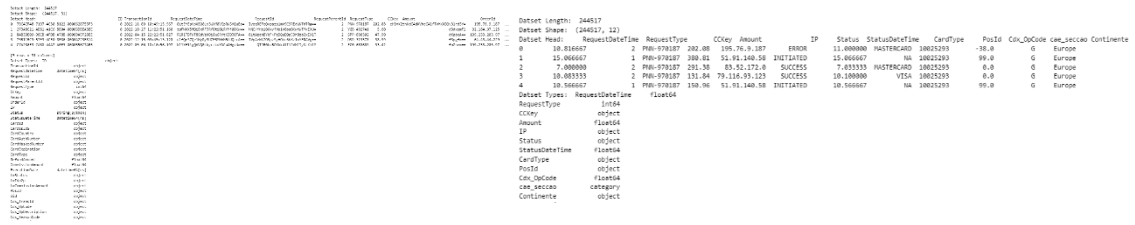


Figura 28 - *Dataset Transactions* Original vs. Transformado

No *dataset Transaction* existiam muitas colunas que não representavam relevância para o projeto, colunas que representavam *id* e correspondência a outras tabelas. No início existiam 31 colunas, ficando no final com doze colunas que representam unicamente os dados importantes para análise, tais como:

- *RequestType*: Tipo de pedido;
- *CCKey*: Identificação do cliente;
- *Amount*: Valor da transferência;
- IP: Endereço de onde foi efetuada a transferência;
- *Status*: Estado da transferência;
- *StatusDateTime*: Data do estado;
- *CardType*: Tipo de cartão/serviço que efetuou a transferência.

Quanto ao *dataset TransactionMB,* apesar de ter colunas com nomes diferentes, estas representam o mesmo que o *dataset Transaction*. Como já referido, o *dataset* não foi explorado como os outros devido à sua imensidão de dados (6773170 registos) mas o modelo a aplicar é o mesmo que o das *Transactions.*

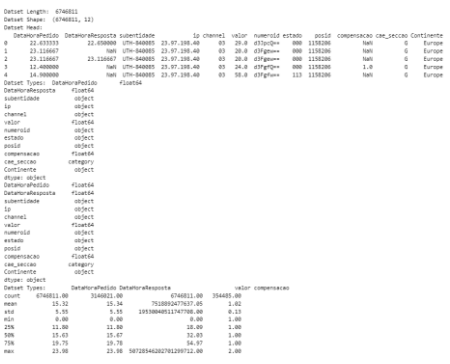


Figura 29 - *Dataset TransactionsMB*

5.2.2 Processamento dos dados

Como demonstrado na Figura 18 e Figura 23, os dados irão passar por uma fase de processamento de modo a transformá-los e atribuir os algoritmos respetivos.

Esta fase deve ser aplicada aos dados de treino e de teste, para garantir que são equivalentes. Este processamento é também necessário para obter a lista de *features* e as *labels* a estas associadas que irão ser utilizadas durante o treino dos modelos, ou seja, a criação de um *dataset* processado e personalizado à necessidade dos mesmos.

Foram definidas algumas variáveis globais de modo a facilitar a implementação ao longo dos diferentes *notebooks* e outras apenas locais para o *notebook* em execução. Algumas dessas variáveis são:

a) _random_state: Serve para controlar a aleatoriedade aplicada aos dados antes de aplicar a divisão. Também é usada como *seed* para inicializar o gerador de números aleatórios, para obter os mesmos valores em execuções de diferentes momentos (*Python Random Seed() Method*, n.d.; *Sklearn.Model_selection.Train_test_split — Scikit-Learn 1.4.Dev0 Documentation*, n.d.);
b) _test_size: É definido entre 0,0 e 1,0 para representar a proporção do conjunto de dados a ser incluída na divisão de teste. Se não for definido, o valor é definido como o complemento do tamanho da amostra e se não for definido o *train_size*, por defeito será 0,25 (Sklearn.Model_selection.Train_test_split — Scikit-Learn 1.4.Dev0 Documentation, n.d.);
c) _max_depth: Define a profundidade máxima da árvore de uma *Decision Tree*. Se não for definido, os nós serão expandidos até que todas as folhas sejam puras ou até que todas as folhas contenham menos que *min_samples_split* amostras(*Decision Trees — Scikit-Learn Documentation*, n.d.).
d) _min_samples_leaf: Representa o número mínimo de amostras necessárias para estar num nó folha da *Decision Tree*. Um ponto de divisão em qualquer profundidade só será considerado se deixar pelo menos amostras de treino *min_samples_leaf* em cada uma das ramificações esquerda e direita. Isso pode ter o efeito de suavizar o modelo, especialmente na regressão. Se for inteiro, considera-se *min_samples_leaf* como o número mínimo. Se decimal, então *min_samples_leaf* é uma fração e a fórmula [ceil(min_samples_leaf * n_samples)] é o número mínimo de amostras para cada nó (*Decision Trees — Scikit-Learn Documentation*, n.d.; *Sklearn.Model_selection.KFold — Scikit-Learn 1.2.2 Documentation*, n.d.).
e) _number_of_folds: Número de conjuntos ou número de iterações de divisão usados no *K-Folds cross-validator*. Deve ser pelo menos 2, normalmente entre 5 a 10 (*Sklearn.Model_selection.KFold — Scikit-Learn 1.2.2 Documentation*, n.d.).
f) _top: Representa o número de dados com que trabalhar. Se não definido serão contemplados todos os dados devolvidos da BD. Serve principalmente na fase de experimentação para limitar a quantidade de dados com que trabalhar.

g) Best_k: Depois de analisados os k definidos é atribuído o que teve melhores resultados nas métricas analisadas.
h) Clf_better: Melhor classificador de *Decision Tree.*

Com o objetivo de auxiliar e simplificar o código foram criadas algumas funções, num ficheiro global, podendo ser utilizadas nos diferentes *noteboks*, tais como:

```
# Function to perform training
def train_DecisionTree(X_train,X_test,y_train,crit=None,cl_w=None):
    clf = DecisionTreeClassifier(# Creating the classifier object
        criterion= crit, random_state=_random_state,
        max_depth=_max_depth, min_samples_leaf=_min_samples_leaf,
        class_weight= cl_w,)
    clf.fit(X_train, y_train)# Performing training
    return clf
```

Código 3 - Função treino DT

Esta função auxilia a criação de uma DT passando só os parâmetros necessários, tais como o critério pretendido e os dados, deixando os restantes atribuídos a partir de variáveis globais como mencionado acima.

```
# Function to calculate
def cal_metrics(y_test, y_pred,clf,X,y ,cmap=None):
...
    MatrixDisplay(cm, y_pred, cmap)
...
    report = cross_val_score(clf,  X, y, cv=kf,).round(4),
            cross_val_score(clf,  X, y, cv=kf, scoring='precision_macro',).round(4), \
            cross_val_score(clf,  X, y, cv=kf, scoring='recall_macro',).round(4), \
            cross_val_score(clf,  X, y, cv=kf, scoring='f1_macro',).round(4)
...
    return
f'|{"{:.6f}".format(Accuracy)}\t|\t{"{:.6f}".format(Precision)}\t|\t{"{:.6f}".format(Recall)}\t|\t{f1}'
```

Código 4 - Função calcular métricas

Para validar os resultados das DT foi criado uma função com diferentes métricas, tais como *Cross_val_score*, *Matrix*, *Accuracy*, *F1-score* entre outras, como pode ser verificado no Anexo B.

```
# Function to calculate
def convertDatetoFloat(Dates):
    return pd.to_datetime(Dates).dt.hour+
                (pd.to_datetime(Dates).dt.minute*100/60)/100
```

Código 5 - Função conversão de datas

Alguns dos problemas encontrados ao longo da exploração dos dados foi de como se iria tratar as datas para que os modelos as pudessem processar. Não se optou por retirar do *dataset* por serem uma fonte de informação importante. Existem diferentes abordagens que se podem tomar tais como (*Machine Learning with Datetime Feature Engineering: Predicting Healthcare Appointment No-Shows | by Andrew Long | Towards Data Science*, n.d.) :

- Extrair recursos de data e hora e dividir o campo em componentes de data e hora;
- Criando recursos de atraso através do deslocamento dos valores de uma variável de série temporal de volta no tempo por um determinado número de etapas;
- Criação de características sazonais para captura de padrões nos dados que se repetem em intervalos regulares;
- Tempo percorrido desde o último evento;

Para simplificar o processo optou-se por criar a função acima que recebe o *dataframe* da coluna respetiva e converte as horas para o respetivo número. Por exemplo para 13h30m devolverá 13.5.

```
def getTransformer(columns,data):
    transformer = make_column_transformer(
    (OneHotEncoder(), columns),remainder="passthrough",)
    transformer.fit(data)
    return transformer
```

Código 6 - Função *Transformer* dos dados

Como pode ser verificado na Tabela 18 os *datasets* contêm colunas com valores alfanuméricos, o que os algoritmos de ML não estão preparados para aceitar para além de valores numéricos. Para tal é necessário proceder a uma transformação dos dados. Existem pelo menos três formas de o fazer:

- *Ordinal Encoding* ou *Label Encoding* envolve o mapeamento de cada rótulo exclusivo para um valor inteiro. Esse tipo de codificação só é realmente apropriado se houver uma relação conhecida entre as categorias. Também pode ser útil usar uma codificação ordinal, pelo menos como um ponto de referência com outras formas de codificação. Converte as classes categóricas em números que as representam (ex: masculino/feminino -> 0/1)[3].
- *Frequency Encoding* utiliza a frequência das categorias como rótulos. Nos casos em que a frequência está relacionada um pouco com a variável de destino, ela ajuda o modelo a entender e a atribuir o peso em proporção direta e inversa, dependendo da natureza dos dados[4].
- *One-Hot Encoder* por sua vez utiliza as categorias de uma coluna e divide-as em várias colunas, cada uma representando uma categoria. Depois essas colunas são preenchidas com 0 ou 1 dependendo da categoria respetiva. Este tipo de codificação é apropriado para dados categóricos onde não existe relação entre categorias[5].

Como a maioria das colunas dos *datasets* não representam relação entre eles o *One-Hot Encoder* foi a codificação selecionada para usar no projeto.

[3] (*All about Categorical Variable Encoding | by Baijayanta Roy | Towards Data Science*, n.d.; *Codificação de Variáveis - Label vs One-Hot Encoder | Viniboscoa.Dev*, n.d.; *Ordinal and One-Hot Encodings for Categorical Data - MachineLearningMastery.Com*, n.d.)
[4] (*All about Categorical Variable Encoding | by Baijayanta Roy | Towards Data Science*, n.d.)
[5] (*All about Categorical Variable Encoding | by Baijayanta Roy | Towards Data Science*, n.d.; *Codificação de Variáveis - Label vs One-Hot Encoder | Viniboscoa.Dev*, n.d.; *Ordinal and One-Hot Encodings for Categorical Data - MachineLearningMastery.Com*, n.d.)

```
def findBestk(k_values,metric,p,data):
...
    for k in k_values:
        # Train a KNN model
        knn_model = NearestNeighbors(n_neighbors=k,metric=metric,p=p)
...
        distances, _ = knn_model.kneighbors(data)

        # Calculate precision, recall, and F1-score
...
        precision_scores.append(precision)
        recall_scores.append(recall)
        f1_scores.append(f1)
...
    return best_k
```

Código 7 - Função procurar melhor k

Para o algoritmo *k-Nearest Neighbors* (KNN) um dos parâmetros mais importantes é o valor do número de *Neighbors,* mais conhecido como k. Para tal o Código 7 foi utilizado para percorrer um *array* de valores de k e devolver o melhor valor, considerando as distâncias entre os diferentes pontos e através de diferentes métricas, como a média das distâncias, *precision*, *F1-score* entre outros. O exemplo completo encontra-se no Anexo B.

```
def findBestthreshold(best_k,metric,p,threshold_values,data):
    nbrs = NearestNeighbors(n_neighbors=best_k,metric=metric,p=p)
    nbrs.fit(data) # fit model
    distances, indexes = nbrs.kneighbors(data)
...
    beta = 1.0
    for threshold in threshold_values:
...
        outlier_ratio = outliers.sum() / len(outliers)
...
        score_silhouette = silhouette_score(data.toarray(), labels=labels)
...
        precision = precision_score(labels, np.ones(l))
        recall = recall_score(labels, np.ones(l))
        fbeta = fbeta_score(labels, np.ones(l), beta=beta)
        tradeoff_metric = (1 + beta**2) * (precision * recall) / (beta**2 * precision + recall)
...
        fpr, tpr, thresholds = roc_curve(labels, outlier_scores)
        auc_score = auc(fpr, tpr) # Compute the AUC
...
        gmm = GaussianMixture(n_components=2)
        gmm.fit(data.toarray())
```

Código 8 - Função procurar o melhor *threshold*

Depois de descobrir o melhor valor de k e como o objetivo da aplicação deste algoritmo é detetar *outliers,* também foi necessário descobrir o melhor *threshold*.

```
def findBestthreshold(best_k,metric,p,threshold_values,data):
    nbrs = NearestNeighbors(n_neighbors=best_k,metric=metric,p=p)
    nbrs.fit(data) # fit model
    distances, indexes = nbrs.kneighbors(data)
...
    beta = 1.0
    for threshold in threshold_values:
...
        outlier_ratio = outliers.sum() / len(outliers)
...
        score_silhouette = silhouette_score(data.toarray(), labels=labels)
...
        precision = precision_score(labels, np.ones(l))
        recall = recall_score(labels, np.ones(l))
        fbeta = fbeta_score(labels, np.ones(l), beta=beta)
        tradeoff_metric = (1 + beta**2) * (precision * recall) / (beta**2 * precision + recall)
...
        fpr, tpr, thresholds = roc_curve(labels, outlier_scores)
        auc_score = auc(fpr, tpr) # Compute the AUC
...
        gmm = GaussianMixture(n_components=2)
        gmm.fit(data.toarray())
```

Código 8 tem um comportamento semelhante ao Código 7 mas foram utilizadas mais algumas métricas como *roc curve*, *GuassianMixture* entre outras. O exemplo completo encontra-se no Anexo B.

```
def savemodel(model,name):
    joblib.dump(value=model, filename='Models/'+ name + '.joblib')
def loadmodel(name):
    return joblib.load(filename='Models/'+ name + '.joblib')
```

Código 9 - Função *save* e *load* Modelos

Ao longo do desenvolvimento, os modelos levavam elevado tempo de processamento, foi necessário por isso encontrar uma solução. Optou-se por usar *Joblib* que faz parte do ecossistema *SciPy* e fornece funcionalidades para trabalhos de *pipelining* em *Python*. Foi usado para armazenar e carregar objetos de *Python* que usam estruturas de dados *NumPy* de forma

eficiente. No caso do projeto foi utilizado para os algoritmos de ML que exigem muitos parâmetros ou armazenam todo o conjunto de dados[6].

5.3 Modelo Cliente

Para a construção do modelo de cliente foi utilizado o algoritmo *Decision Tree* (DT) com o objetivo de obter a previsão de fraude e classificação do *Merchant*/cliente.

Começou-se por fazer uma exploração dos dados já existentes para perceber os seus atributos e qualidade dos mesmos. Encontraram-se muitos registos com valores a NULL e NaN o que levou a um tratamento posterior.

```
df = db.getClients()
# Printing the dataswet properties
print("Dataset Length: ", len(df))
print("Dataset Shape: ", df.shape)
print("Dataset Head: ", df.head())
print("Dataset Types: ", df.dtypes)
```

Código 10 - Exploração dos Clientes

```
Dataset Length:  24305
Dataset Shape:  (24305, 5)
Dataset Head:     estado setorinstituicional  fraude cae_seccao Continente
0  Activo               S14       0         -     Europe
1     N/C                       0         G     Europe
2  Activo               S11       0         J     Europe
3  Activo               S11       0         -     Europe
4  Activo               S11       0         J     Europe
Dataset Types:  estado                  category
setorinstituicional    category
fraude                    int32
cae_seccao             category
Continente             category
dtype: object
```

Figura 30 - Resultados da Exploração dos Clientes

Como também mencionado anteriormente na secção 5.2.1 foi adicionado ao *dataframe* o respetivo continente e a secção do respetivo CAE depois do tratamento do mesmo. Posteriormente, recorrendo ao método *seaborn.pairplot()*, foram construídas duas matrizes de *plots* do *dataframe* para análise global dos CAE e dos continentes.

[6] (*Save and Load Machine Learning Models in Python with Scikit-Learn - MachineLearningMastery.Com*, n.d.)

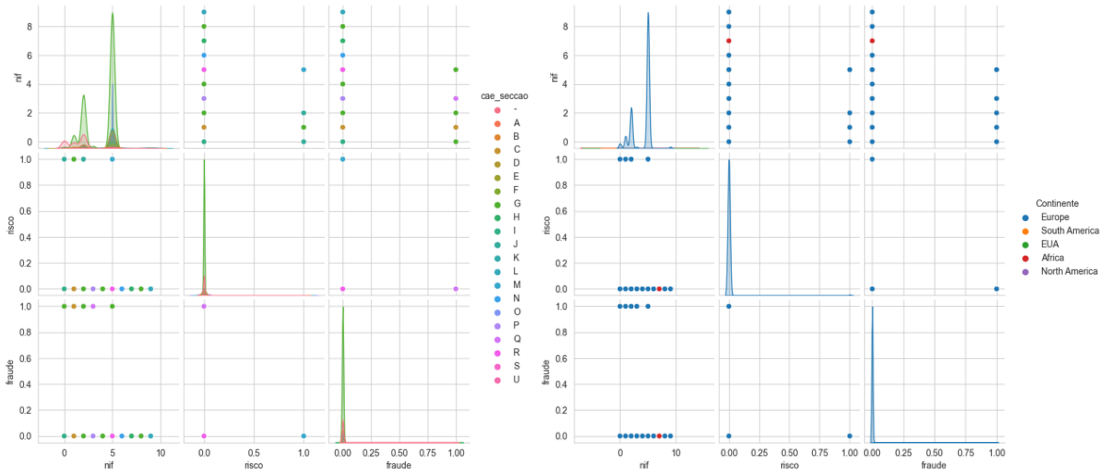


Figura 31 - Matriz Plots Clientes

Através das matrizes da Figura 31 podemos observar que:

- A maioria dos clientes são Pessoas Coletivas (NIF 5);
- A fraude é praticamente inexistente;
- A pouca fraude que existe é atribuída a Pessoa Singular (NIF 1, 2, 3), Pessoas Coletivas (NIF 5) e a estrangeiros (NIF 0);
- A esmagadora maioria dos clientes pertence à Europa e a seguir ao continente Africano. Os restantes não são percetíveis nos gráficos;

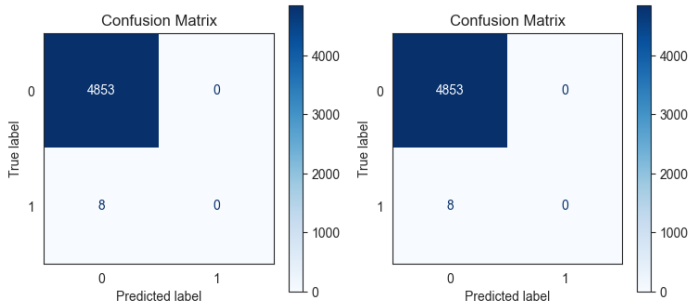


Figura 32 - *Decision Tree Confusion Matrix* Comparação

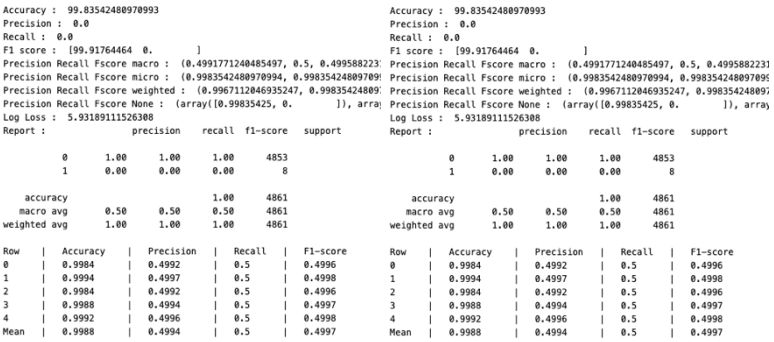


Figura 33 - *Decision Tree* Métricas Comparação

No início da fase de experimentação, usando tanto o critério *gini* e *entropy* para o *dataset* de clientes estes eram muito semelhantes se não iguais, como observado na Figura 32 e Figura 33. Também cedo se apercebeu de um problema no *dataset,* uma *accuracy* elevada, aproximadamente 99%, mas com uma *precision* e *recall* de 0%. Isto deveu-se a um desequilíbrio do *dataset* em relação ao valor objetivo (fraude). Num total de 24305 registos só existem 29 marcados como fraude o que representa 0.1193% dos dados.

Para tentar mitigar o desequilíbrio foram testadas quatro alternativas, *Under-sampling* e *Weighted classification, Bagging Classifier, SMOTE,* e comparadas com os já obtidos.

5.3.1 Under-sampling

Representa uma subamostragem que reduz aleatoriamente a amostra da classe que tem mais amostras para o mesmo tamanho da classe com menos amostras. Reduz a amostra dos clientes não fraudulentos para ter o mesmo tamanho dos clientes fraudulentos. Ao subamostrar, estamos a desperdiçar a maior parte dos dados e portanto, geralmente não será a melhor abordagem (*How to Deal with Unbalanced Data. What Is Precision and Recall | Towards Data Science*, n.d.).

```
# Create an instance of RandomUnderSampler to perform under-sampling
rus = RandomUnderSampler(random_state=_random_state)
X_train_resampled, y_train_resampled = rus.fit_resample(X_train, y_train)
clf_gini = train_DecisionTree(X_train_resampled, X_test, y_train_resampled,criterion="gini")
clf_entropy = train_DecisionTree(X_train_resampled, X_test, y_train_resampled,criterion="entropy")
```

Código 11 – Exemplo de *Under-sampling*

5.3.2 Weighted classification

A classificação ponderada envolve treinar um classificador com pesos maiores nas amostras pertencentes à classe com menos amostras. Para escolher os pesos apropriados, utiliza-se a seguinte equação para cada classe (*How to Deal with Unbalanced Data. What Is Precision and Recall | Towards Data Science*, n.d.):

$$w_i = \frac{\sum_i^C N_i}{c + N_i} \tag{19}$$

```
# calcular os pesos de classe usando a função compute_class_weight
cl_w = compute_class_weight("balanced", classes=np.unique(y), y=y)
clf_gini = train_DecisionTree( X, X_test, y, class_weight=dict(enumerate(cl_w)),criterion="gini")
clf_entropy = train_DecisionTree(X, X_test, y, class_weight=dict(enumerate(cl_w)),criterion="entropy")
```

Código 12 – Exemplo de *Weighted classification*

5.3.3 Bagging Classifier

Consiste num meta-estimador de conjuntos que ajusta várias versões do estimador base, treinando com um conjunto de dados modificados utilizando a técnica de amostragem de bagging. Esta técnica pode resultar num conjunto de dados duplicados ou único. O preditor final combina as previsões feitas por cada classificador por votação ou por média (*Bagging Classifier Python Code Example - Data Analytics*, n.d.).

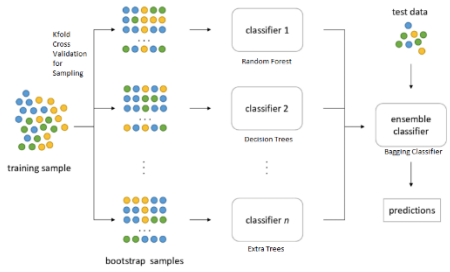


Figura 34 - Exemplo Bagging Classifier

```
# Create a base decision tree classifier with max_depth=3
dt = DecisionTreeClassifier(max_depth=_max_depth, random_state=_random_state)
# Create a bagging classifier that uses 10 decision tree estimators
bag = BaggingClassifier(base_estimator=dt, random_state=_random_state)
```

Código 13 - Exemplo Bagging Classifier

5.3.4 SMOTE

Geralmente benéfico com dados de baixa dimensão, não atenua o viés em direção à classificação na classe maioritária para a maioria dos classificadores quando os dados são de alta dimensão e é menos eficaz do que a subamostragem aleatória.

O SMOTE é benéfico para classificadores KNN para dados de alta dimensão se o número de variáveis for reduzido realizando algum tipo de seleção de variável, caso contrário, a classificação KNN é tendenciosa para a classe minoritária. Na prática, no cenário de alta dimensão, apenas classificadores KNN baseados na distância euclidiana parecem beneficiar-se substancialmente do uso do SMOTE. (Blagus & Lusa, 2013; *SMOTE. A Technique to Overcome Class Imbalance… | by Emilia Orellana | Medium*, n.d.).

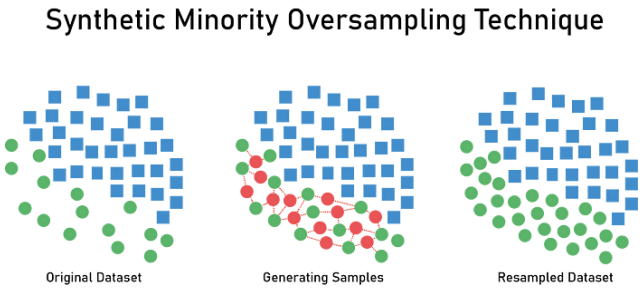


Figura 35 - SMOTE Exemplo

SMOTE é utilizado quando há poucos pontos de dados em uma classe. Como mencionado anteriormente, é importante corrigir o desequilíbrio de classes no conjunto de dados porque, aquando da criação dos modelos, deve existir uma hipótese igual para o modelo prever uma classificação.

A técnica de sobreamostragem de minoria sintética (SMOTE) pode ser entendida pela Figura 35. Pode-se observar que há alguns círculos verdes e muitos outros quadrados azuis. Depois de gerar amostras, ele basicamente usa os recursos dos conjuntos de dados fornecidos e é assim que as amostras são geradas. Por fim, vê-se na última imagem que há mais quadrados verdes e parece mais uniforme (*SMOTE. A Technique to Overcome Class Imbalance… | by Emilia Orellana | Medium*, n.d.).

```
# Apply SMOTE to generate synthetic examples of the minority class
sm = SMOTE(random_state=_random_state)
X_res, y_res = sm.fit_resample(X_train, y)
```

Código 14 - Exemplo SMOTE

5.4 Modelo Transferências

O modelo transferências é um modelo de deteção de *outliers*. Como o *dataset* não tem um rótulo objetivo, construiu-se o modelo para tentar perceber o que se encontra fora do normal, o que representa um *outlier* e depois aplicou-se um DT para analisar as regras geradas pelos algoritmos.

Como mencionado anteriormente o modelo é dividido em duas partes, a primeira para detetar os *outliers* onde foram usados os algoritmos *Isolation Forest* (IF) e o KNN em paralelo, como uma dupla confirmação e a segunda parte uma DT com os resultados de cada algoritmo.

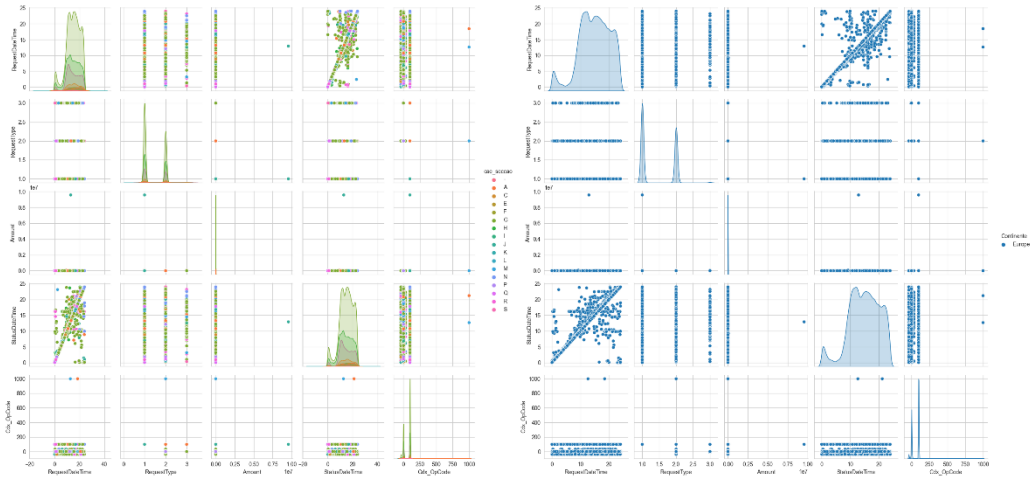


Figura 36 - Matriz *Plots* Transferências

Através das matrizes da Figura 36 podemos observar que:

- A maioria dos *Requests* são entre as 8h e 20h;
- A relação entre a data do *Request* e *Status* são as mesmas, salvo algumas exceções;
- O *Request type* dominante é 1;
- As secções do CAE dominantes são a F e G;
- Só existe o continente Europeu;

5.4.1 Isolation Forest (IF)

Isolation Forest (IF) como mencionado na secção 4.3.3 é uma técnica utilizada para identificar *outliers* em dados, de complexidade de tempo linear e baixo uso de memória, adequado para processar grandes conjuntos de dados.

Foi necessário escolher o *n_estimators* e *contamination* mais adequado para o problema e para tal recorreu-se à função *GridSearchCV*() para percorrer diferentes valores e encontrar o melhor.

```
clf = IsolationForest() # Create an Isolation Forest instance
param_grid = {# Define the parameter grid for grid search
    'n_estimators': [10,20,30,40,50,],
    'contamination': [0.05, 0.1, 0.15,0.2,0.25,0.3]
}
scoring = { 'accuracy': 'accuracy', 'precision': 'precision', 'recall': 'recall', 'f1': 'f1'}
# Perform grid search using the custom scoring function
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring=scoring, refit='f1')
grid_search.fit(X_train, np.ones(X_train.shape[0]))
```

Código 15 – *Isolation Forest* com *GridSearchCV*

```
Best n_estimators: 40
Best contamination: 0.05
```

Figura 37 – Resultado dos melhores parâmetros

Para encontrar os melhores parâmetros usou-se *accuracy, precision, recall* e *F1-score (scoring)* para avaliar o desempenho do modelo de validação cruzada no conjunto de dados. No caso de haver considerações além da pontuação máxima na escolha do melhor estimador usou-se o *F1-score* (refit) para a função que retorna o *best_index_* selecionado *cv_results_*. Nesse caso, o *best_estimator_*e *best_params_*será definido de acordo com o retornado *best_index_* enquanto o *best_score_* atribuído não estiver disponível (*GridSearchCV — Scikit-Learn 1.2.2 Documentation*, n.d.).

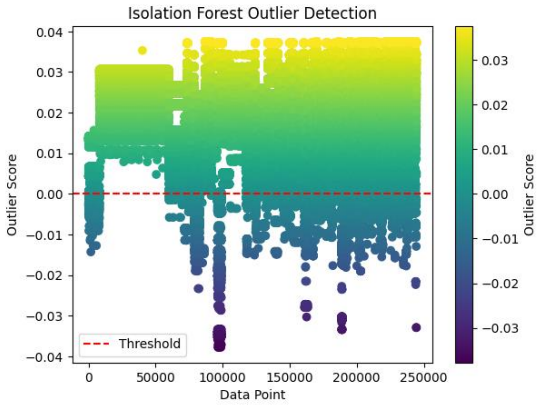


Figura 38 - *Isolation Forest* Resultado

Como pode ser analisado na Figura 38, foram detetados alguns *outliers* no *dataset*, num total de 12166 o que corresponde a 4,98% do *dataset*. Com um *threshold* de 0 obteve-se *socres* -0.0000068063, -0.0377521686 e -0.0071335975 correspondendo ao mínimo, máximo e média dos scores.

```
min_outlier =y_scores[outliers==-1].min()
print(dt[y_scores == min_outlier]["CCKey"])
display(dt.loc[dt['CCKey'].isin(dt[y_scores == max_outlier]["CCKey"])])
```

Código 16 - Procurar o maior *outlier*

Recorrendo ao Código 16, detetou-se o cliente com *CCkey* “AMX-241021" como o maior *outlier* com 4 transações num total de 2849 do cliente. Pode-se verificar que o *request* foi efetuado depois das horas normais e o IP foi o mesmo em todos os *outliers*, entre outras diferenças.

	RequestDateTime	RequestType	CCKey	Amount	IP	Status	StatusDateTime	CardType	PosId	Cdx_OpCode	cae_seccao	Continente
96504	21.883333	1	AMX-241021	160.86	52.149.108.159	INITIATED	21.883333	NA	10025293	99.0	G	Europe
96824	21.616667	1	AMX-241021	164.66	52.149.108.159	INITIATED	21.616667	NA	10025293	99.0	G	Europe
97675	21.716667	1	AMX-241021	164.66	52.149.108.159	INITIATED	21.716667	NA	10025293	99.0	G	Europe
98657	22.033333	1	AMX-241021	250.84	52.149.108.159	INITIATED	22.033333	NA	10025293	99.0	G	Europe

	RequestDateTime	RequestType	CCKey	Amount	IP	Status	StatusDateTime	CardType	PosId	Cdx_OpCode	cae_seccao	Continente
96131	13.583333	1	AMX-241021	561.88	52.149.108.255	INITIATED	13.583333	NA	10025293	99.0	G	Europe
96132	15.916667	1	AMX-241021	188.69	52.149.108.255	INITIATED	15.916667	NA	10025293	99.0	G	Europe
96133	13.383333	1	AMX-241021	133.71	52.149.108.255	INITIATED	13.383333	NA	10025293	99.0	G	Europe
96134	16.000000	2	AMX-241021	128.48	199.233.203.97	SUCCESS	16.033333	VISA	10025293	0.0	G	Europe
96135	9.433333	1	AMX-241021	173.99	52.149.108.159	INITIATED	9.433333	NA	10025293	99.0	G	Europe
...	...	...	...	...	...	...	...	...	...	...	...	...
98975	14.116667	2	AMX-241021	33.56	199.233.203.97	ERROR	14.166667	VISA	10025293	-38.0	G	Europe
98976	18.150000	1	AMX-241021	296.99	52.149.108.255	INITIATED	18.150000	NA	10025293	99.0	G	Europe
98977	13.016667	1	AMX-241021	137.73	52.149.108.255	INITIATED	13.016667	NA	10025293	99.0	G	Europe
98978	21.900000	2	AMX-241021	852.78	199.233.203.97	ERROR	22.016667	VISA	10025293	-38.0	G	Europe
98979	11.600000	1	AMX-241021	238.74	20.50.135.254	INITIATED	11.600000	NA	10025293	99.0	G	Europe

Figura 39 - Transações com o *outlier* e Transações do Cliente

5.4.2 KNN

Como já mencionado anteriormente, o KNN é um *score* de *outlier* e frequente na deteção de anomalias. Quanto maior a distância para o KNN, menor a densidade local, maior a probabilidade de um ponto de consulta ser um *outlier*.

O Código 7 e Código 8, como mencionado anteriormente, auxiliou na descoberta do melhor valor de k e *threshold* a usar.

```
K =2|Average: 39.9240|Max:9481696.0000|STD:19174.8963|AVG:39.92|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =3|Average: 40.4962|Max:9507567.0006|STD:19227.3713|AVG:40.50|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =4|Average: 40.8298|Max:9523232.0000|STD:19259.1516|AVG:40.83|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =5|Average: 41.0545|Max:9533474.2500|STD:19279.8642|AVG:41.05|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =6|Average: 41.1988|Max:9533474.2500|STD:19279.8647|AVG:41.20|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =7|Average: 41.3358|Max:9533474.2500|STD:19279.8661|AVG:41.34|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =8|Average: 41.4346|Max:9533474.2500|STD:19279.8662|AVG:41.43|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =9|Average: 41.8494|Max:9533474.2500|STD:19279.9545|AVG:41.85|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
K =10|Average: 42.0430|Max:9533474.2505|STD:19280.0082|AVG:42.04|Outliers Max(percentile 95%):12226 | Outliers Mean(percentile 95%): 12226
Best K Avg Distance: 2
Best K F1: 2
Best K silhouette: 2
```

```
Best Outlier Ratio Threshold: 30
Best Outlier Ratio Score: 0.00013
Best Silhouette Threshold: 35
Best Silhouette Score: 0.9674671672777995
Best Mean Threshold: 35
Best Mean Score: 6
Best Metric Threshold: 30
Best Metric Score: 0.00013999020068595198
Best AUC Threshold: None
Best AUC: -1
Best GMM Threshold: 35
Best GMM: -669.5478102298657
```

Figura 40 - Melhor k e *threshold*

Construíram-se dois gráficos para analisar os resultados com o k=2 para melhor compreensão dos dados, como pode ser verificado na Figura 41.

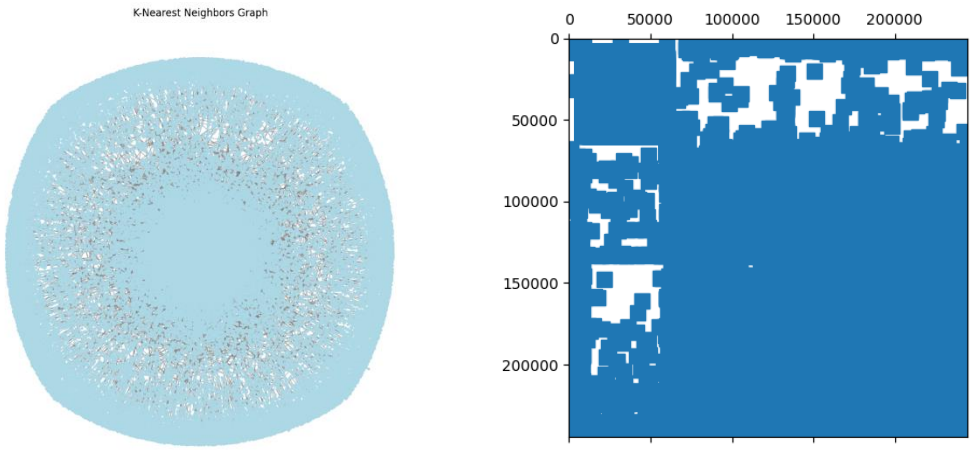


Figura 41 - KNN Gráfico, k=2

Na Figura 41, no gráfico a direta é percetível os dois grupos criados mas com tamanhos diferentes visíveis. No gráfico da esquerda não é tão percetível devido à densidade dos dados, mas consegue-se perceber a existência de dois grupos também e as ligações entres eles apesar de não ser muito visível. A Figura 42 foi recolhida a partir de uma amostra com 10000 linhas do *dataset Transaction* com o objetivo de exemplo de demonstração das ligações entre os dados, onde os que não tem ligação são *outliers*.

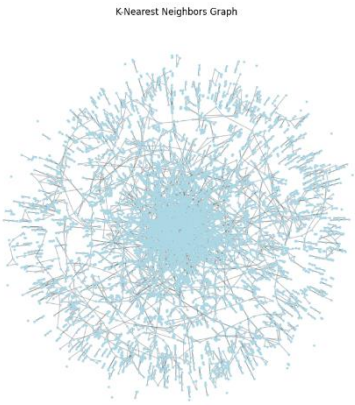


Figura 42 - KNN Gráfico simplificado

Com as distâncias entre os dados devolvidos pelo knn_model.kneighbors(X) construiu-se um gráfico *plot*. Devido à existência de um *outlier* extremamente distante de 9481696, a perceção dos *outliers* era praticamente nula. Para tal optou-se por retirar esse *outlier* para efeito de gráfico e deparou-se com uma melhor compreensão. Consequentemente retirou-se outros três maiores *outliers* e a concentração cada vez foi mais percetível. A Figura 43 é uma demonstração desse resultado.

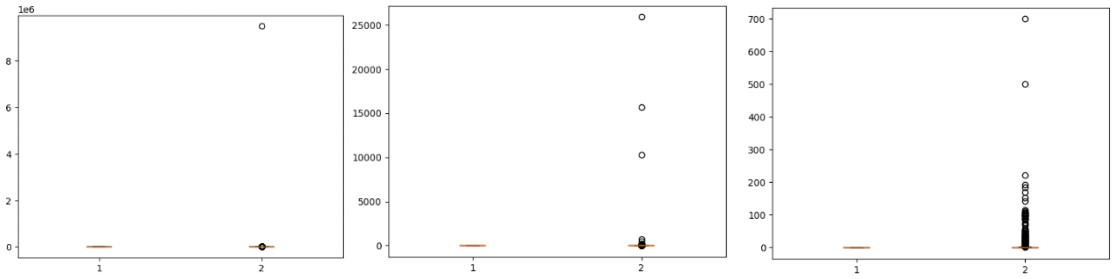


Figura 43 - KNN *Plot*

Com o k=2 e o *threshold* contido no intervalo [30,35] de acordo com as métricas usadas no Código 8 construiu-se um gráfico com todas as distâncias para melhor interpretação. Tal como na análise através dos gráficos *plot* a perceção era pouca devido a alguns *outliers* extremamente distantes. Foram removidas as linhas com as sete maiores distâncias. Tratou-se os diferentes *outliers* e a média das distâncias (19.96) como pode ser observado na Figura 44 no gráfico da direita. Se a média fosse definida para o *threshold* iriam conter muitos *outliers* devido a esta estar a contemplar distâncias muitos dispersas tornando a média uma fraca medida para o *threshold*. No final foi decidido usar o valor máximo encontrado através do Código 8, 35 para o *threshold*.

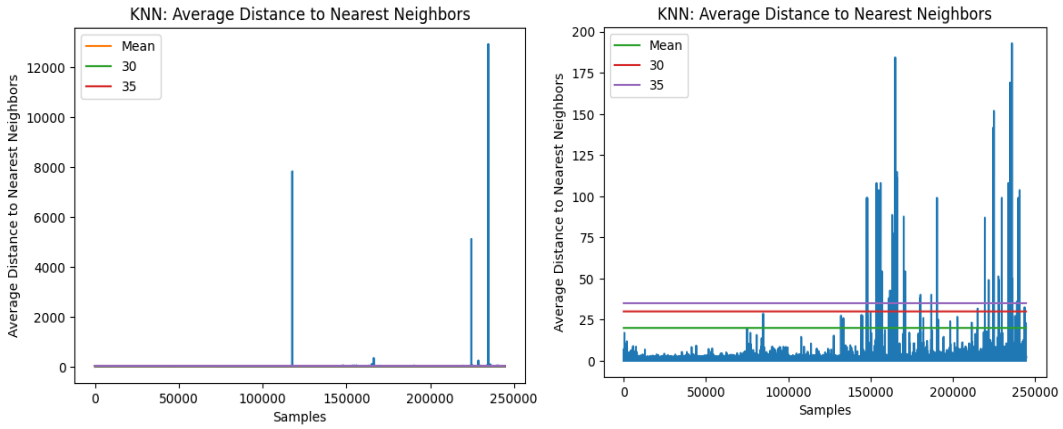


Figura 44 - KNN *threshold* gráfico

5.4.3 TransactionsMB

O *dataset TransactionsMB* também foi inicialmente explorado como já referido.

Figura 45 - Matriz *Plots* TransferênciasMB

Através das matrizes da Figura 45 podemos observar que:

- A maioria dos Pedidos são entre as 10h e 22h;
- A relação entre a data dos Pedidos e Respostas são as mesmas, salvo algumas exceções;
- O *Channel* dominante é 3;
- O estado e compensação dominante é o 0;
- A secção do CAE dominante é o G;
- O continente Europeu é dominante, só é percetível a existência do continente africano num valor mais elevado;

6 Avaliação

No desenvolvimento de uma solução é importante avaliar se os objetivos foram cumpridos e em que medida, através de diferente técnicas e metodologias. Numa fase inicial deste capítulo serão apresentadas as hipóteses, os indicadores e a metodologia de avaliação adequados. Por fim, são apresentados os resultados obtidos sobre a solução pretendida.

6.1 Hipótese

Para se conseguir avaliar a solução desenvolvida, que consiste no desenvolvimento de um software de ML, é necessário definir hipóteses de avaliação tendo em conta os objetivos previamente definidos para o projeto. O objetivo principal deste projeto, como já referido, é a deteção de fraudes nas transações efetuadas.

É necessário verificar o modelo com os mecanismos já existentes para validar os resultados apresentados. Com esta verificação é possível formular as seguintes hipóteses:

- Hipótese Nula (H0) – O modelo induz mais em erro que os mecanismos já implementados.
- Hipótese Alternativa (H1) – O modelo pode substituir os mecanismos já implementados.

6.2 Indicadores e Fontes de Informação

Para conseguir validar a hipótese formulada é preciso analisar o desempenho de cada algoritmo, através da avaliação da *accuracy* nas diferentes transações. Contudo, a *accuracy* só por si, pode não ser a abordagem mais correta, principalmente para problemas mais complexos. No caso onde haja um desequilíbrio de classe significativo nos dados fornecidos, uma classe tem mais instâncias de dados do que as outras classes, um modelo pode prever a classe maioritária para

todos os casos e ter uma alta pontuação de *accuracy*, quando não está a prever as classes minoritárias (*What Is a Confusion Matrix in Machine Learning?*, 2023).

Através dos trabalhos relacionados analisados, na aplicação de ML é normal recorrer à matriz de confusão e a quatro métricas: *Accuracy, Precision, Recall, F1-score*. Outra prática comum também referida é o treino do modelo de ML num conjunto de 80% e a sua avaliação de desempenho num conjunto de dados nos restantes 20% (Aslam et al., 2022).

6.2.1 Matriz de Confusão

A matriz de confusão é uma matriz n por n, onde n representa o número de classes usadas para avaliar o desempenho dos modelos de classificação para um determinado conjunto de dados se os valores verdadeiros dos dados de teste forem conhecidos. A própria matriz pode ser facilmente entendida, mas em alguns casos pode ser confusa, uma vez que mostra os erros no desempenho do modelo na forma de uma matriz, também conhecida como matriz de erros (Aniruddha Bhandari, 2023).

Como forma de exemplo vai ser utilizada uma matriz de confusão 2 por 2, com dois valores possíveis (positivo e negativo). Como representado na Figura 46, as linhas representam os valores previstos e as colunas os valores reais (Aniruddha Bhandari, 2023).

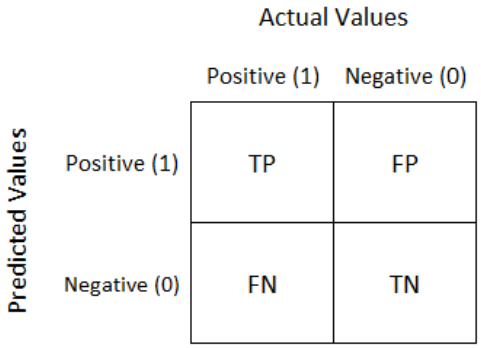


Figura 46 - Matriz de Confusão (Sarang Narkhede, 2018)

Na matriz representada na Figura 46 encontram-se quatros conceitos (Sarang Narkhede, 2018):

- Verdadeiro positivo (*TP*) – valor previsto igual ao valor real, ambos positivos, correspondendo a uma classificação positiva correta;
- Falso positivo (*FP*) – valor previsto diferente do valor real, com o real negativo e o previsto positivo, correspondendo a uma classificação incorreta de um valor negativo como positivo;
- Verdadeiro negativo (*TN - Type 1 Error*) – valor previsto igual ao valor real, ambos negativos, correspondendo a uma classificação negativa correta;

- Falso negativo (*FN - Type 2 Error*) – valor previsto diferente ao valor real, com o real positivo e o previsto negativo, correspondendo a uma classificação incorreta de um valor positivo como negativo.

6.2.2 Métricas de avaliação

Nos trabalhos relacionados analisados são usadas várias métricas de avaliação. De acordo com a análise realizada as métricas mais comuns são *Accuracy*, *Precision*, *Recall*. Outra métrica importante de usar na avaliação é o *F1 score*, que relaciona a *Precision* com o *Recall*.

Ao longo do trabalho desenvolvido foram usadas outras métricas como *Silhouette Score, AUC e ROC Curve* e outras mais simples como Média, Máximo, Minimo e STD

6.2.2.1 Accuracy

É importante para determinar a precisão da classificação. Define com que frequência o modelo prevê uma resposta correta. Também pode ser calculada como a razão do número de previsões corretas para o número total de previsões (*Confusion Matrix in Machine Learning - Javatpoint*, n.d.).

A *Accuracy* deverá ser mais alta possível.

$$Accuracy = \frac{Previsões\ Correta}{Previsões} = \frac{TP + TN}{TP + FP + TN + FN} \quad (20)$$

6.2.2.2 Precision

Representa a quantidade das precisões corretas que foram fornecidas ou todas as classes positivas que foram previstas corretamente, a quantidade que foram realmente verdadeiras, o que determina a confiança no modelo (Confusion Matrix in Machine Learning - Javatpoint, n.d.).

A *Precision* deverá ser mais alta possível.

$$Precision\ = \frac{TP}{TP + FP} \quad (21)$$

6.2.2.3 Recall

Representa quantos casos positivos foram previstos corretamente (*Confusion Matrix in Machine Learning - Javatpoint*, n.d.).

O *Recall* deverá ser mais alto possível.

$$Recall\ = \frac{TP}{TP + FN} \quad (22)$$

6.2.2.4 F1 score

Se dois modelos tiverem baixa *Precision* e alto *Recall* ou vice-versa, é difícil comparar esses modelos. O *F1 score* ajuda a medir o *Recall* e *Precision* ao mesmo tempo. É representado pela média harmónica no lugar da média aritmética entre ambos (Sarang Narkhede, 2018).

$$F1\ score\ = \frac{2}{Recall^{-1} + Precision^{-1}} = 2\frac{Precision \cdot Recall}{Precision + Recall} \quad (23)$$

6.2.2.5 Silhouette Score

O coeficiente de silhueta ou pontuação de silhueta é uma métrica usada para calcular a qualidade de uma técnica de agrupamento. O seu valor varia de -1 a 1, onde 1 significa que os *clusters* estão bem separados uns dos outros e -1 significa que os *clusters* estão atribuídos de maneira errada.

$$Silhouette\ Score\ = \frac{(ba)}{\max(b, a)} \quad (24)$$

A distância média intra-cluster (a) e a distância média do cluster mais próximo (b) para cada amostra. Só é definido se o número de rótulos for [2 <= n_labels <= n_samples – 1].(*Silhouette Coefficient. This Is My First Medium Story, So… | by Ashutosh Bhardwaj | Towards Data Science*, n.d.; *Sklearn.Metrics.Silhouette_score — Scikit-Learn 1.2.2 Documentation*, n.d.).

6.2.2.6 AUC e ROC Curve

A curva ROC é uma medida de desempenho para os problemas de classificação em várias configurações de limite onde ROC é uma curva de probabilidade e AUC representa o grau ou medida de separabilidade. Representa o quanto o modelo é capaz de distinguir as classes. Quanto maior a AUC, melhor o modelo está a prever 0 classes como 0 e 1 classes como 1. Por analogia, quanto maior a AUC, melhor o modelo está em distinguir entre pagamentos como *outlier* e *inlier* (*Understanding AUC - ROC Curve | by Sarang Narkhede | Towards Data Science*, n.d.).

A curva ROC é traçada com TPR contra o FPR onde TPR está no eixo y e FPR está no eixo x.

$$TPR/\ Recall\ /\ Sensibilidade\ = \frac{TP}{TP + FN} \quad (25)$$

$$Specificity\ = \frac{TN}{TN + FP} \quad (26)$$

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity \quad (27)$$

Um modelo excelente tem AUC próximo de 1, o que significa que tem uma boa medida de separabilidade enquanto 0 significa um modelo que tem a pior medida de separabilidade. Ele está a prever 0 como 1 e 1 como 0. Se o resultado for aproximadamente 0.5, significa que o

modelo não tem nenhuma capacidade de separação de classes (*Classificação: Curva ROC e AUC | Machine Learning | Google for Developers*, n.d.).

6.2.2.7 Gaussian Mixture

Um modelo de mistura gaussiana (GMM) tenta encontrar uma mistura de distribuições de probabilidade gaussianas multidimensionais que melhor modelam qualquer conjunto de dados de entrada. De forma simples, os GMMs podem ser usados para encontrar *clusters* da mesma maneira que o *k-means* (*Gaussian Mixture Model - GeeksforGeeks*, n.d.).

6.3 Metodologia de Avaliação

Como no desenvolvimento optou-se por seguir a metodologia CRIPS-DM, a avaliação será efetuada na etapa de *Evaluation* (Avaliação do modelo). Quando terminado o modelo será confrontado com os requisitos e analisado com a empresa o seu resultado. Se este resultado for positivo é implementado o modelo em produção, caso contrário o modelo será revisto e serão discutidas as alterações necessárias.

Como já mencionado também é importante comparar ambos os modelos aplicados independentemente se o primeiro modelo apresentar um resultado positivo e seguir para a etapa seguinte de *Deployment.*

Nesta fase, a validação da robustez e exatidão do algoritmo para diferentes conjuntos de dados é importante. Os modelos de ML nem sempre são estáveis, se um modelo treinado for exposto a novos dados pode não prever com a mesma precisão e falhar ao generalizar sobre os novos dados (*overfiting*) ou o modelo poderá não encontrar padrões nos dados de treino, o que implicaria um mau desempenho nos conjuntos de testes (*underfitting*) (Lakshana G V, 2021).

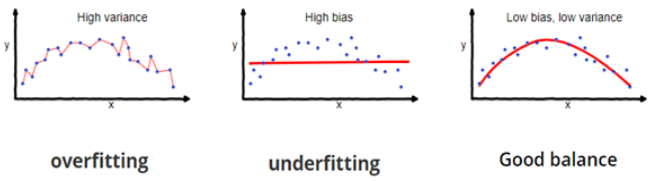


Figura 47 - Problemas de desempenho de ML

Para ajudar a evitar estes problemas uma possível solução é a técnica chamada *Cross-Validation* (validação cruzada) usada em alguns dos trabalhos analisados. Ela consiste em dividir o conjunto de dados em dados de treino e dados de teste. Os dados de treino são usados para treinar o modelo e os de teste são usados para previsões (Lakshana G V, 2021).

Um dos métodos a utilizar é o método simples de *Holt Out*, onde o conjunto de dados é dividido aleatoriamente por dois conjuntos, de treino e de teste. Podendo ser divididos em 70-30, 75-

25,80-20 ..., mas em regra geral os dados de treino tendem a ser maiores do que o de teste. O artigo de Aslam et al. utiliza uma divisão 80-20. Seria assim interessante utilizar os mesmos valores para a avaliação juntamente com as métricas já mencionadas (Aslam et al., 2022; Lakshana G V, 2021).

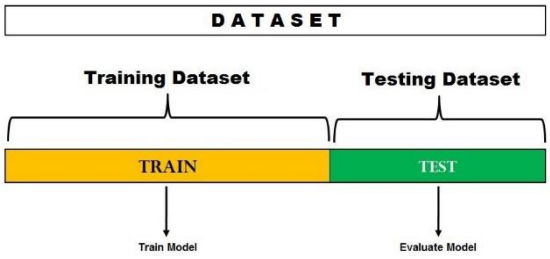


Figura 48 - Método *Holt Out* (Lakshana G V, 2021)

Uma alternativa é o método de *K-Fold Cross-Valid*ation, onde todos dados são divididos em k conjuntos de tamanhos praticamente iguais. O primeiro conjunto é selecionado como o conjunto de teste e o modelo é treinado nos conjuntos k-1 restantes. A taxa de erro de teste é então calculada após o ajuste do modelo aos dados de teste. Na segunda interação, o segundo conjunto é selecionado como um conjunto de teste e os conjuntos k-1 restantes são usados para treinar os dados e o erro é calculado. Este processo continua para todos os conjuntos k (Lakshana G V, 2021).

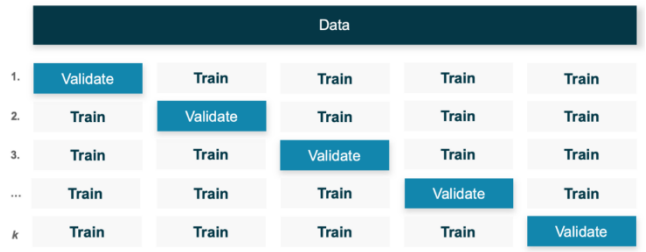


Figura 49 - Método de *K-Fold Cross-Valid* (Lukas Feick, 2019)

Existem mais métodos como *Leave One Out Cross-Validation*, Stratified K-Fold Cross-Validation, *Bias - Variance Tradeoff* entre outros, que poderão ser alternativas a usar, caso os dois métodos anteriores não sejam suficientes ou algum deles for descartado durante o processo de desenvolvimento.

6.4 Avaliação dos modelos

Como já referido anteriormente na secção 5.2, todos os modelos resultam numa *Decision Tree* com as previsões de clientes e transações. Como tal a avaliação de cada modelo resume-se a aplicar as métricas (*Accuracy, Precision, Recall, Lost, F1-score*) e cada DT através do *Holt Out* e *K-Fold Cross-Valid*ation.

O código correspondente aos *scores* obtidos de cada modelo recorrendo a *GridSearchCV* encontra-se no Anexo D, e as regras geradas da *Decision Tree* no Anexo E

6.4.1 Modelo Cliente

Como demonstrado ao longo da secção 5.3 foram usadas diferentes alternativas para construir a DT e comparadas as suas métricas. O resultado dessa comparação encontra-se no Anexo C.

À medida que se foi comparando os resultados do *Holt Out* e *K-Fold Cross-Validation* notou-se que:

- Os resultados entre o critério *Gini* e *Entropy* só eram diferentes usando o método *ClassWeigh* e mesmo assim a diferença não era significativa.
- Existia uma diferença considerável da *Precision* e *Recall* entre *Holt Out* e *K-Fold Cross-Validation*.
- O método que melhor resultados obteve ao longo dos testes, foi o SMOTE.

No entanto, quanto mais se analisava, dúvidas surgiam de qual o melhor método a usar pela quantidade de métricas a comparar e pelos resultados bastante diferentes. Para auxiliar o processo de avaliação e decisão recorreu-se a *GridSearchCV* passando os critérios para cada caso e para as métricas desejadas (*Accuracy, Precision, Recall, Lost, F1-score*) obtendo-se um único *score* como resultado mais fácil para comparar e avaliar o modelo.

O melhor score obtido foi o do SMOTE com 81,74% e com os parâmetros:

```
Best parameters: {
   'criterion': 'gini',
   'max_depth': None,
   'min_samples_leaf': 1,
   'min_samples_split': 2
}
```

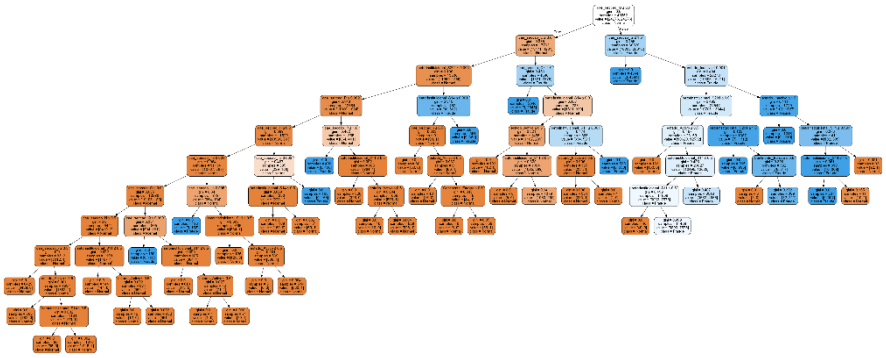


Figura 50 - *Decision Tree SMOTE*

O resultado foi a DT representado na Figura 50, onde as folhas da árvore “azuis” representam resultados marcados como fraude e as regras da construção da árvore encontra-se no Anexo C onde a classe 0 representa um resultado normal e 1 resultado como fraude.

```
samples_Normal = Getpredict(Estado.Activo,
SetorInstituicional.S14,CaeSeccao.H,Continente.Europe)
samples_Fraud = Getpredict(Estado.Activo,
SetorInstituicional.S201,CaeSeccao.G,Continente.Europe)
```

Código 17 - Exemplo de previsão Normal e Fraude

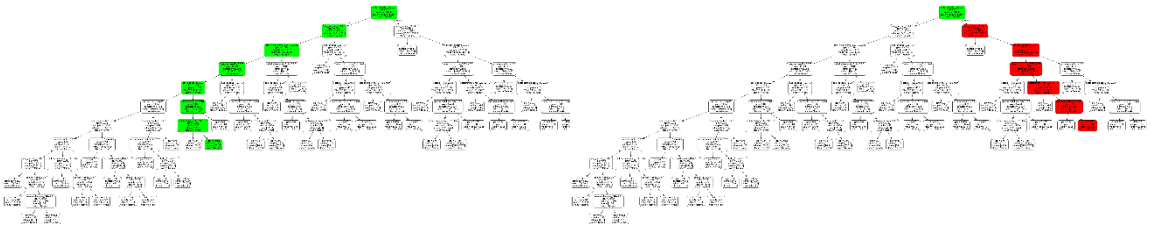


Figura 51 - Resultado *Decision Trees* Normal e Fraude

Na Figura 51 a árvore à esquerda representa a previsão “Normal” e a da direita a Fraude representadas no Código 17. Todos os nós designados por “Normal” são representados a verde na previsão e a partir de um nó onde a fraude seja dominante o mesmo começa a ser “marcado” a vermelho.

```
Accuracy :  56.63855173832545
Precision :  0.16122913505311076
Recall :  58.620689655172406
F1 score :  [72.29276757  0.32157382]
Log Loss :  1562.9050115939338
```

Row	Accuracy	Precision	Recall	F1-score
0	0.9984	0.4992	0.5	0.4996
1	0.9994	0.4997	0.5	0.4998
2	0.9984	0.4992	0.5	0.4996
3	0.9988	0.4994	0.5	0.4997
4	0.9992	0.4996	0.5	0.4998
Mean	0.9988	0.4994	0.5	0.4997

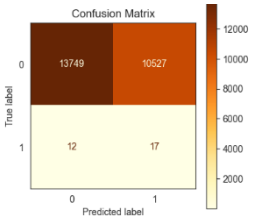


Figura 52 - Avaliação da DT gerada - Clients

Com a DT criada e as previsões efetuadas obteve-se os resultados demonstrados na Figura 52 com valores mais distantes na *accuracy* entre o *Holt Out* (à esquerda) e *K-Fold Cross-Valid*ation (à direita).

6.4.2 Modelo Transferências

Como já referido anteriormente, o modelo Transferências consiste em dois algoritmos *Isolation Forest* (IF) e KNN para deteção de *outliers*. A ideia é funcionar como dupla validação, no caso dos dois algoritmos devolverem positivamente um *outlier*, é um grande indicador que não se trata de um falso positivo.

6.4.2.1 Isolation Forest (IF)

Como demonstrado ao longo da secção 0 recorreu-se também *GridSearchCV* com as métricas (*Accuracy, Precision, Recall, Lost, F1-score*) obtendo-se o melhor score com os resultados do *Isolation Forest* (IF) para a construção de uma DT, com um score de 67.24% e parâmetros:

```
Best parameters: {
     'criterion': 'gini',
     'max_depth': 7,
     'min_samples_leaf': 1,
     'min_samples_split': 7
     }
```

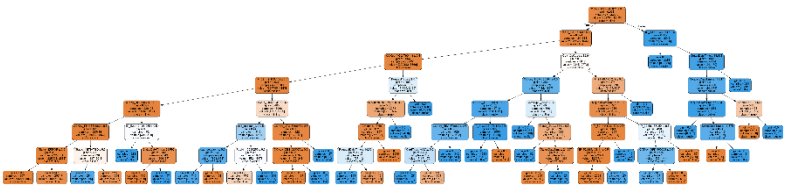


Figura 53 - *Decision Tree Isolation Forest*

Como a DT da Figura 53, os “azuis” representam resultados marcados como “maus”, neste caso representando *outliers*.

```
sample_data_inline = [[10,  2   ,'PNN-970187',  202.08  ,'195.76.9.187',    'ERROR',    11,
'MASTERCARD',   '10025293'  ,-38.0, 'G',    'Europe']]
sample_data_outlier = [[10, 2   ,'DBD-316701',  20252.08    ,'52.149.108.159',  'ERROR',    11,
'MASTERCARD',   '10025269'  ,99.0,  'A',    'Europe']]
```

Código 18 - Exemplo de previsão *Inlier* e *Outlier* - IF

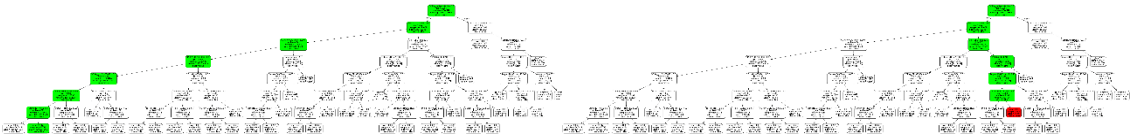


Figura 54 - Resultado *Decision Tree Inlier* e *Outlier*

Na Figura 54 a árvore à esquerda representa a previsão *Inlier* e a da direita o *Outlier*, representadas no Código 18. Todos os nós designados por *Inlier* são representados a verde na previsão e a partir de um nó onde a *Outlier* seja dominante o mesmo começa a ser "marcado" a vermelho.

```
Accuracy :  98.70193074510156
Precision :  98.6159169550173
Recall :  74.96301167187242
F1 score :  [99.32124374 85.17792099]
Log Loss :  46.78715829862867
```

Row	Accuracy	Precision	Recall	F1-score
0	0.9869	0.9876	0.871	0.9206
1	0.9876	0.9742	0.8906	0.9279
2	0.9863	0.984	0.8694	0.9182
3	0.9876	0.9884	0.8784	0.9257
4	0.9861	0.9848	0.8689	0.9181
Mean	0.9869	0.9838	0.8757	0.9221

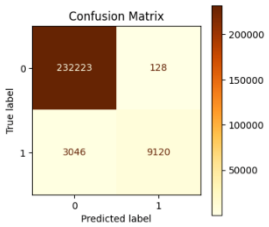


Figura 55 - Avaliação da DT gerada – *Transaction* IF

Com a DT criada e as previsões efetuadas obteve-se os resultados demonstrados na Figura 55 com valores semelhantes nas diferentes métricas de cada modelo.

6.4.2.2 KNN

À semelhança do IF, depois da construção do modelo recorreu-se também *GridSearchCV* com as métricas (*Accuracy, Precision, Recall, Lost, F1-score*) obtendo-se o melhor *score* com os resultados do KNN para a construção de uma DT, com um *score* de 30.81% e parâmetros:

```
Best parameters: {
     'criterion': 'gini',
     'max_depth': None,
     'min_samples_leaf': 9,
     'min_samples_split': 5
     }
```

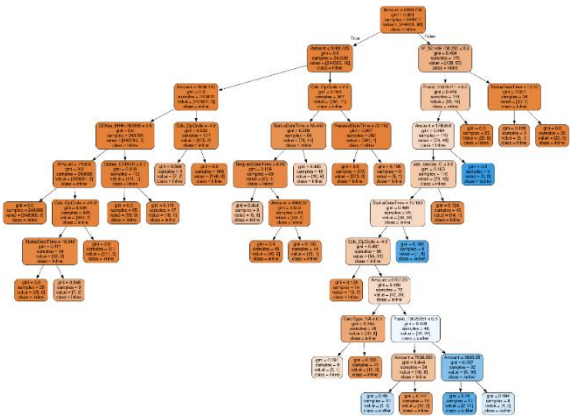


Figura 56 - *Decision Tree* KNN

```
sample_data_inline = [[10,  2   ,'PNN-970187',  202.08  ,'195.76.9.187',    'ERROR',    11,
'MASTERCARD',   '10025293'  ,-38.0, 'G',    'Europe']]
sample_data_outlier = [[18, 2   ,'PNN-970187',  17838.00    ,'82.223.151.247',  'INITIATED',    11,
'MASTERCARD',   '10025293'  ,-38.0,  'G',    'Europe']]
```

Código 19 - Exemplo de previsão Inlier e Outlier - KNN

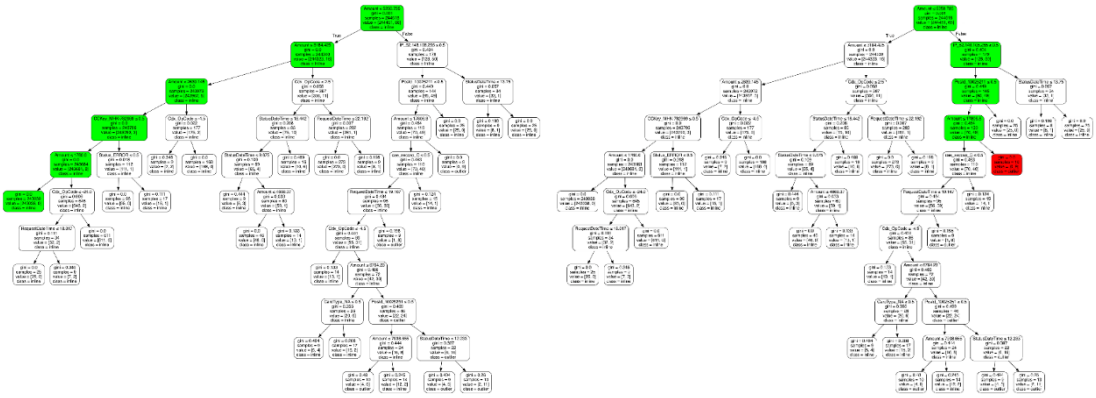


Figura 57 - Resultado *Decision Tree Inlier* e *Outlier*

A construção e análise da DT gerada segue o mesmo princípio da gerada para o *Isolation Forest*.

```
Accuracy :  95.0064821668841
Precision :  6.0
Recall :  0.024658885418379087
F1 score :  [9.74392745e+01 4.91159136e-02]
Log Loss :  179.9846259692048
```

Row	Accuracy	Precision	Recall	F1-score
0	0.9976	0.9906	0.9854	0.9879
1	0.9978	0.9907	0.9862	0.9884
2	0.9972	0.9902	0.9799	0.985
3	0.9977	0.9937	0.9814	0.9875
4	0.9972	0.991	0.9811	0.9857
Mean	0.9975	0.9912	0.9828	0.9869

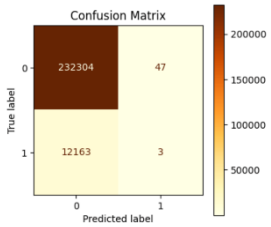


Figura 58- Avaliação da DT gerada – *Transaction* KNN

Com a DT criada e as previsões efetuadas obteve-se os resultados demonstrados na Figura 58 com valores mais distantes no *recall* entre o *Holt Out* (à esquerda) e *K-Fold Cross-Valid*ation (à direita).

6.4.2.3 Validação Cruzada

```
sample_data_outlier_IF_KNN = [[10, 2   ,'DBD-
316701',  20252.08    ,'52.149.108.159',  'ERROR',    11,'MASTERCARD',   '10025269'  ,99.0,  'A',
 'Europe']]
```

Código 20 - Exemplo de previsão *Outlier* no IF e KNN

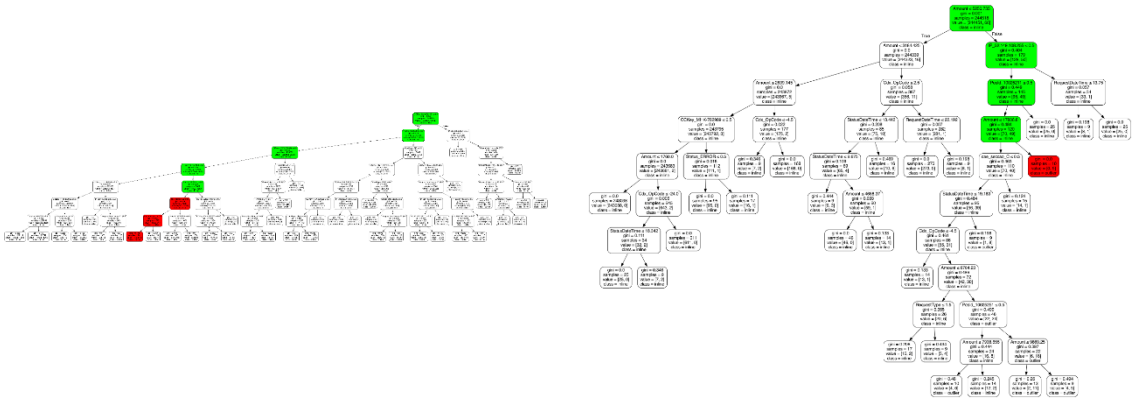


Figura 59 -Resultado *Decision Tree Outlier* no IF e KNN

Como mencionado, o objetivo de usar dois algoritmos diferentes, IF e KNN, é para fazer uma dupla validação ou validação cruzada, diminuindo a probabilidade de falsos positivos. O Código 20 é exemplo de uma transação que foi detetada como *outlier* nos dois modelos.

6.5 Avaliação solução final

Apesar da avaliação de todos os modelos ser positiva, o modelo global desenvolvido neste momento não consegue substituir os mecanismos já implementados. Estar a treinar o modelo sempre que seja preciso fazer uma análise ou quando um modelo deve ser treinado novamente não foi desenvolvido ao longo do projeto. Em consideração aos requisitos funcionais e não funcionais idealizados no início do projeto a solução também não cumpre alguns deles.

A construção de dois modelos distintos para a deteção de *outlier* nas transações revelou ser uma forma útil de diminuir os falsos positivos, levando-os a colmatar as falhas de cada um dos algoritmos.

O modelo cliente levanta algumas dúvidas devido à questão de poucos registos marcados como fraude, tendo de ser avaliado mais a pormenor do que os outros modelos que contam com uma grande diferença de dados entre si.

Algumas destas questões poderão ser mitigadas com trabalhos futuros e os requisitos alcançados também.

Em suma, pode-se concluir que a hipótese proposta no início desta secção é possível, mas que não foi totalmente alcançada neste projeto.

7 Conclusões

Esta última secção tem como objetivo refletir sobre as limitações sentidas ao longo do projeto, bem como melhorias e trabalho futuro.

É feita uma apreciação geral do projeto discutindo os objetivos alcançados.

7.1 Limitações

Uma grande limitação sentida ao longo deste projeto foi a capacidade de processamento necessária vs. a usada. Em alguns casos, a necessidade de RAM superava as 70 GB onde o portátil conseguia gerir devido ao SO contudo, o tempo de execução era elevadíssimo. Alguns algoritmos e avaliações superavam os 10 dias de execução e alguns deles por requererem elevada quantidade de RAM e processamento o *kernel* desligava-se como representado na mensagem de erro da Figura 60.

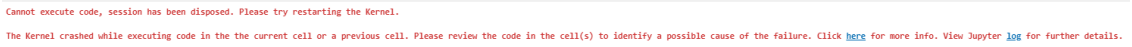


Figura 60 - Erro *Kernel*

Uma limitação encontrada, mas que não foi explorada ficando para um trabalho futuro é a questão da API e *interface* dos resultados que levantavam questões do *Concept Drift* quanto ao treino do modelo como atualização do mesmo.

7.2 Melhorias

Como mencionado nas limitações, a capacidade da máquina afeta o desempenho e, até mesmo a forma como os modelos e solução é construída e treinada. Para tal uma das melhorias seria a obtenção de uma máquina com mais capacidade, de preferência virtual, devido à sua escalabilidade.

Outra melhoria era tentar usar não só o *One-Hot Encoder* mas tentar agregar a outros métodos como por exemple o *Label Encoding*, tirando partido dos pontos fortes de cada uma na codificação dos dados, como utilizado em projetos idênticos (Meira et al., 2019).

7.3 Trabalho futuro

Após as melhorias, o primeiro passo seria a disponibilização do modelo para ser usado através de API e consequentemente a criação da interface dos resultados, bem como todos requisitos funcionais e não funcionais relacionados.

Com uma aplicação e API funcional, seria importante resolver a questão relacionada com o *Concept Drift* para que os modelos pudessem evoluir ao longo do tempo, mas de maneira que o sistema continue fiável à deteção de fraudes e anomalias.

7.4 Apreciação final

Em relação aos objetivos propostos, pode-se concluir que o principal objetivo, a criação de modelos para os dados já existentes, foi alcançada. A solução desenvolvida permite detetar anomalias e prever o risco para um cliente. As anomalias podem ser verificadas por dois modelos diferentes, o que leva a uma deteção cruzada e posterior análise detalhada numa *Decision Tree*.

Em suma, este projeto foi um desafio devido à falta de experiência e conhecimento com algoritmos de *Machine Learning*. Contudo, permitiu estudar e aprofundar conceitos de AI e ML, em amplas áreas, trazendo competências numa área em constante expansão e interesse para o trabalho profissional.

Referências

All about Categorical Variable Encoding | by Baijayanta Roy | Towards Data Science. (n.d.). Retrieved June 22, 2023, from https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

Ammarapala, Veeris Chinda, Thanwadee Pongsayaporn, Pimnapa Ratanachot, Wit Punthutaecha, Koonnamas Janmonta, & Koson. (2018). Cross-border shipment route selection utilizing analytic hierarchy process (AHP) method. *Songklanakarin Journal of Science and Technology*, *40*(1), 31–37. https://doi.org/10.14456/SJST-PSU.2018.3

Aniruddha Bhandari. (2023, February 7). *Confusion Matrix for Machine Learning*. https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

Aslam, F., Hunjra, A. I., Ftiti, Z., Louhichi, W., & Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and machine learning. *Research in International Business and Finance*, *62*. https://doi.org/10.1016/J.RIBAF.2022.101744

Ataques financeiros online dispararam 233% durante pandemia. (2022, April 5). Diário de Notícias. https://www.dn.pt/sociedade/ataques-financeiros-online-dispararam-233-durante-pandemia-14743538.html

Bagging Classifier Python Code Example - Data Analytics. (n.d.). Retrieved June 26, 2023, from https://vitalflux.com/bagging-classifier-python-code-example/

Berk, R. A. (2008). *Statistical learning from a regression perspective*. 358.

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(1), 1–16. https://doi.org/10.1186/1471-2105-14-106/FIGURES/7

Brandon Dan. (2020). The state of machine learning. *Journal of Computing Sciences in Colleges*, *35*(9), 102–103. https://doi.org/10.5555/3417682.3417696

Classificação: curva ROC e AUC | Machine Learning | Google for Developers. (n.d.). Retrieved May 26, 2023, from https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=pt-br

Codificação de Variáveis - Label vs One-Hot Encoder | viniboscoa.dev. (n.d.). Retrieved June 22, 2023, from https://www.viniboscoa.dev/blog/codificacao-de-variaveis-label-vs-one-hot-encoder

Confusion Matrix in Machine Learning - Javatpoint. (n.d.). Retrieved February 18, 2023, from https://www.javatpoint.com/confusion-matrix-in-machine-learning

Craja, P., Kim, A., & Lessmann, S. (2020). Deep learning for detecting financial statement fraud. *Decision Support Systems*, *139*, 113421. https://doi.org/10.1016/J.DSS.2020.113421

Credit Card Fraud Detection Using Machine Learning & Python | by Pranjal Saxena | Towards Data Science. (n.d.). Retrieved June 1, 2023, from https://towardsdatascience.com/credit-card-fraud-detection-using-machine-learning-python-5b098d4a8edc

CRISP-DM Help Overview - IBM Documentation. (n.d.). Retrieved February 5, 2023, from https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview

Damos-lhe as boas-vindas ao Colaboratory - Colaboratory. (n.d.). Retrieved June 5, 2023, from https://colab.research.google.com/?utm_source=scs-index#scrollTo=Nma_JWh-W-IF

Decision Tree - GeeksforGeeks. (n.d.). Retrieved May 4, 2023, from https://www.geeksforgeeks.org/decision-tree/

Decision Trees — scikit-learn documentation. (n.d.). Retrieved June 3, 2023, from https://scikit-learn.org/stable/modules/tree.html

Download Python | Python.org. (n.d.). Retrieved May 1, 2023, from https://www.python.org/downloads/

DRM Associates. (2016). *Value Analysis and Function Analysis System Technique THE CONCEPT OF VALUE*.

Gaussian Mixture Model - GeeksforGeeks. (n.d.). Retrieved May 11, 2023, from https://www.geeksforgeeks.org/gaussian-mixture-model/

GridSearchCV — scikit-learn 1.2.2 documentation. (n.d.). Retrieved June 27, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Gurney, K. (2018). *An Introduction to Neural Networks*. https://doi.org/10.1201/9781315273570

Houtao Deng. (2018, December 7). *An Introduction to Random Forest | by Houtao Deng | Towards Data Science*. https://towardsdatascience.com/random-forest-3a55c3aca46d

How to Deal with Unbalanced Data. What is Precision and Recall | Towards Data Science. (n.d.). Retrieved June 21, 2023, from https://towardsdatascience.com/how-to-deal-with-unbalanced-data-d1d5bad79e72

Huang, T. Y., & Huang, C. (2019). Fraud payment research- Payment through credit car. *ACM International Conference Proceeding Series*, 189–194. https://doi.org/10.1145/3345035.3345059

ifthenpay | Pagamentos Digitais para a sua empresa. (n.d.). Retrieved December 5, 2022, from https://ifthenpay.com/

Isolation Forest | Anomaly Detection with Isolation Forest. (n.d.). Retrieved June 22, 2023, from https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/

JEFF SALTZ. (2022, May 2). *CRISP-DM is Still the Most Popular Framework for Executing Data Science Projects - Data Science Process Alliance*. https://www.datascience-pm.com/crisp-dm-still-most-popular/

k-nearest neighbors algorithm - Wikipedia. (n.d.). Retrieved May 10, 2023, from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Koen, P. A., Ajamian, G. M., Boyce, S., Clamen, A., Fisher, E., Fountoulakis, S., Johnson, A., Puri, P., & Seibert, R. (n.d.). *FuzzyFrontEnd: Effective Methods, Tools, and Techniques*.

Koen, P. A., Bertels, H. M. J., & Kleinschmidt, E. (2014). Managing the front end of innovation-part I: Results from a three-year study. *Research Technology Management*, *57*(2), 34–43. https://doi.org/10.5437/08956308X5702145

Koen, P., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'amore, R., Elkins, C., Herald, K., Incorvia, M., Johnson, A., Karol, R., Seibert, R., Slavejkov, A., & Wagner, K. (2001). *PROVIDING CLARITY AND A COMMON LANGUAGE TO THE "FUZZY FRONT END."*

L. D. Miles. (1962). *Techniques of Value Analysis and Engineering*. 6th Annual Inland Empire Quality Control Confonference. https://minds.wisconsin.edu/bitstream/handle/1793/5616/526.pdf?sequence=1&isAllowed=y

Lakshana G V. (2021, May 21). *Cross-Validation Techniques in Machine Learning for Better Model*. https://www.analyticsvidhya.com/blog/2021/05/4-ways-to-evaluate-your-machine-learning-model-cross-validation-techniques-with-python-code/

Lee, S.-W. (Seong-whan), & Verri, Alessandro. (2002). Pattern Recognition with Support Vector Machines . *First International Workshop, SVM 2002 Niagara Falls, Canada*.

Liu, C., Gao, Y., Sun, L., Feng, J., Yang, H., & Ao, X. (2022). User Behavior Pre-training for Online Fraud Detection. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *1*, 3357–3365. https://doi.org/10.1145/3534678.3539126

Lukas Feick. (2019, October 2). *Evaluating Model Performance by Building Cross-Validation from Scratch*. https://www.statworx.com/en/content-hub/blog/evaluating-model-performance-by-building-cross-validation-from-scratch

Machine Learning with Datetime Feature Engineering: Predicting Healthcare Appointment No-Shows | by Andrew Long | Towards Data Science. (n.d.). Retrieved June 20, 2023, from https://towardsdatascience.com/machine-learning-with-datetime-feature-engineering-predicting-healthcare-appointment-no-shows-5e4ca3a85f96

Madhurya, M. J., Gururaj, H. L., Soundarya, B. C., Vidyashree, K. P., & Rajendra, A. B. (2022). Exploratory analysis of credit card fraud detection using machine learning techniques. *Global Transitions Proceedings*, *3*(1), 31–37. https://doi.org/10.1016/J.GLTP.2022.04.006

Matheus Zambinati Rosa. (2020, November 9). *CRISP-DM: A metodologia ideal para Ciência de Dados | LinkedIn*. https://www.linkedin.com/pulse/crisp-dm-metodologia-ideal-para-ci%C3%AAncia-de-dados-matheus/?originalSubdomain=pt

Meira, J., Andrade, R., Praça, I., Carneiro, J., & Marreiros, G. (2019). Comparative results with unsupervised techniques in cyber attack novelty detection. *Advances in Intelligent Systems and Computing*, *806*, 103–112. https://doi.org/10.1007/978-3-030-01746-0_12

Nearest Neighbors — scikit-learn documentation. (n.d.). Retrieved June 3, 2023, from https://scikit-learn.org/stable/modules/neighbors.html

Nearest Neighbors regression — scikit-learn 1.2.2 documentation. (n.d.). Retrieved June 15, 2023, from https://scikit-learn.org/stable/auto_examples/neighbors/plot_regression.html#sphx-glr-auto-examples-neighbors-plot-regression-py

Nicola, S. (n.d.). *ANÁLISE DE VALOR INESC-TEC*.

Ordinal and One-Hot Encodings for Categorical Data - MachineLearningMastery.com. (n.d.). Retrieved June 22, 2023, from https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

Osterwalder, A., & Pigneur, Y. (2003). Modeling value propositions in e-business. *ACM International Conference Proceeding Series*, *50*, 429–436. https://doi.org/10.1145/948005.948061

Osterwalder, A., Pigneur, Y., Bernarda, G., Smith, A., & Papadakos, T. (2014). *Value Proposition Design*.

Pandemia marca o ritmo no comércio eletrónico - E-Commerce - Jornal de Negócios. (2022, November 8). Jornal de Negócios. https://www.jornaldenegocios.pt/negocios-em-rede/e-commerce/detalhe/pandemia-marca-o-ritmo-no-comercio-eletronico

Paulo Marmé. (2022, December 6). *Fraudes online: natal pode trazer amargos de boca. Está preparado? - Forbes Portugal*. Forbes Portugal. https://www.forbespt.com/fraudes-online-natal-pode-trazer-amargos-de-boca-esta-preparado/

Preços - Máquinas Virtuais do Windows | Microsoft Azure. (n.d.). Retrieved April 15, 2023, from https://azure.microsoft.com/pt-pt/pricing/details/virtual-machines/windows/?cdn=disable

PRISMA. (n.d.). Retrieved June 27, 2023, from http://prisma-statement.org/prismastatement/flowdiagram.aspx

Python Random seed() Method. (n.d.). Retrieved June 20, 2023, from https://www.w3schools.com/python/ref_random_seed.asp

Rodrigo Ribeiro G. (2020, January 22). *CRISP-DM - O que é e como usar? | LinkedIn*. https://www.linkedin.com/pulse/crisp-dm-o-que-%C3%A9-e-como-usar-rodrigo-ribeiro/?originalSubdomain=pt

Rokach, L., & Maimon, O. (2007). Data mining with decision trees: Theory and applications. *Data Mining with Decision Trees: Theory and Applications*, 1–244. https://doi.org/10.1142/6604

Ryman-Tubb, N. F., Krause, P., & Garn, W. (2018). How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artificial Intelligence*, *76*, 130–157. https://doi.org/10.1016/J.ENGAPPAI.2018.07.008

Saaty, T. L. (1988a). What is the Analytic Hierarchy Process? *Mathematical Models for Decision Support*, 109–121. https://doi.org/10.1007/978-3-642-83555-1_5

Saaty, T. L. (1988b). What is the Analytic Hierarchy Process? *Mathematical Models for Decision Support*, 109–121. https://doi.org/10.1007/978-3-642-83555-1_5

Saaty, T. L. (1991). Some Mathematical Concepts of the Analytic Hierarchy Process. *Behaviormetrika*, *18*(29), 1–9. https://doi.org/10.2333/BHMK.18.29_1

Saaty, T. L., & Katz, J. M. (1990). How to make a decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, *48*, 9–11.

Sadgali, I., Sael, N., & Benabbou, F. (2019). Performance of machine learning techniques in the detection of financial frauds. *Procedia Computer Science*, *148*, 45–54. https://doi.org/10.1016/J.PROCS.2019.01.007

Sarang Narkhede. (2018, May 9). *Understanding Confusion Matrix | by Sarang Narkhede | Towards Data Science*. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Saravanan, R., & Sujatha, P. (2019). A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification. *Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018*, 945–949. https://doi.org/10.1109/ICCONS.2018.8663155

Save and Load Machine Learning Models in Python with scikit-learn - MachineLearningMastery.com. (n.d.). Retrieved June 27, 2023, from https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/

Seify, M., Sepehri, M., Hosseini-far, A., & Darvish, A. (2022). Fraud Detection in Supply Chain with Machine Learning. *IFAC-PapersOnLine*, *55*(10), 406–411. https://doi.org/10.1016/J.IFACOL.2022.09.427

Silhouette Coefficient. This is my first medium story, so… | by Ashutosh Bhardwaj | Towards Data Science. (n.d.). Retrieved June 1, 2023, from https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

sklearn.metrics.silhouette_score — scikit-learn 1.2.2 documentation. (n.d.). Retrieved June 1, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

sklearn.model_selection.KFold — scikit-learn 1.2.2 documentation. (n.d.). Retrieved June 21, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

sklearn.model_selection.train_test_split — scikit-learn 1.4.dev0 documentation. (n.d.). Retrieved June 20, 2023, from https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.train_test_split.html#sklearn.model_selection.train_test_split

SMOTE. A technique to overcome class imbalance… | by Emilia Orellana | Medium. (n.d.). Retrieved June 26, 2023, from https://emilia-orellana44.medium.com/smote-2acd5dd09948

Stripe: panorama da fraude online. (n.d.). Retrieved February 11, 2023, from https://stripe.com/pt-br-us/guides/state-of-online-fraud

Taher, M., Guermah, H., & Nassar, M. (2019). MCDM method for Financial Fraud Detection: A review. *Proceedings of the 4th International Conference on Big Data and Internet of Things*. https://doi.org/10.1145/3372938

Turing, A. M. (1950). COMPUTING MACHINERY AND INTELLIGENCE. *Computing Machinery and Intelligence. Mind*, *49*, 433–460.

Understanding AUC - ROC Curve | by Sarang Narkhede | Towards Data Science. (n.d.). Retrieved May 26, 2023, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

van Belle, R., Baesens, B., & de Weerdt, J. (2023). CATCHM: A novel network-based credit card fraud detection method using node representation learning. *Decision Support Systems*, *164*, 113866. https://doi.org/10.1016/J.DSS.2022.113866

van Engelen, J. E., Hoos, H. H., Fawcett Jesper E van Engelen, T. B., & Hoos hh, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440. https://doi.org/10.1007/s10994-019-05855-6

Viraj Lakshitha. (2021, January 21). *What is a Decision Tree in ML?. What is Decision Tree? | by Viraj_Lakshitha | Medium*. https://vitiya99.medium.com/what-is-a-decision-tree-in-ml-5bd76efc2232

Wang, L. (Ed.). (2005). *Support Vector Machines: Theory and Applications*. *177*. https://doi.org/10.1007/B95439

What is a Confusion Matrix in Machine Learning? (2023, February 16). https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning

What is Artificial Intelligence (AI)? | IBM. (n.d.). Retrieved December 6, 2022, from https://www.ibm.com/cloud/learn/what-is-artificial-intelligence

What is CRISP DM? - Data Science Process Alliance. (n.d.). Retrieved February 5, 2023, from https://www.datascience-pm.com/crisp-dm-2/

Will Koehrsen. (2017, December 27). *Random Forest Simple Explanation. Understanding the Random Forest with an… | by Will Koehrsen | Medium*. https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

Working with Jupyter Notebooks in Visual Studio Code. (n.d.). Retrieved April 1, 2023, from https://code.visualstudio.com/docs/datascience/jupyter-notebooks

Yuan, S., Wu, X., Li, J., & Lu, A. (2017). Spectrum-based Deep Neural Networks for Fraud Detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. https://doi.org/10.1145/3132847

Zach Bedell. (2018, December 7). *Support Vector Machines Explained | by Zach Bedell | Medium*. https://medium.com/@zachary.bedell/support-vector-machines-explained-73f4ec363f13

Anexo A

Tabela CAE

DIVISÃO	DESIGNAÇÃO	SECÇÃO
1	Agricultura, produção animal, caça e actividades dos serviços relacionados	A
2	Silvicultura e exploração florestal	A
3	Pesca e aquicultura	A
5	Extracção de hulha e lenhite	B
6	Extracção de petróleo bruto e gás natural	B
7	Extracção e preparação de minérios metálicos	B
8	Outras indústrias extractivas	B
9	Actividades dos serviços relacionados com as indústrias extractivas	B
10	Indústrias alimentares	C
11	Indústria das bebidas	C
12	Indústria do tabaco	C
13	Fabricação de têxteis	C
14	Indústria do vestuário	C
15	Indústria do couro e dos produtos do couro	C
16	Indústrias da madeira e da cortiça e suas obras, excepto mobiliário | fabricação de obras de cestaria e de espartaria	C
17	Fabricação de pasta, de papel, cartão e seus artigos	C
18	Impressão e reprodução de suportes gravados	C
19	Fabricação de coque, de produtos petrolíferos refinados e de aglomerados de combustíveis	C
20	Fabricação de produtos químicos e de fibras sintéticas ou artificiais, excepto produtos farmacêuticos	C
21	Fabricação de produtos farmacêuticos de base e de preparações farmacêuticas	C
22	Fabricação de artigos de borracha e de matérias plásticas	C
23	Fabricação de outros produtos minerais não metálicos	C
24	Indústrias metalúrgicas de base	C
25	Fabricação de produtos metálicos, excepto máquinas e equipamentos	C
26	Fabricação de equipamentos informáticos, equipamento para comunicações e produtos electrónicos e ópticos	C
27	Fabricação de equipamento eléctrico	C
28	Fabricação de máquinas e de equipamentos, n.e.	C
29	Fabricação de veículos automóveis, reboques, semi-reboques e componentes para veículos automóveis	C
30	Fabricação de outro equipamento de transporte	C
31	Fabricação de mobiliário e de colchões	C
32	Outras indústrias transformadoras	C
33	Reparação, manutenção e instalação de máquinas e equipamentos	C

35	Electricidade, gás, vapor, água quente e fria e ar frio	D
36	Captação, tratamento e distribuição de água	E
37	Recolha, drenagem e tratamento de águas residuais	E
38	Recolha, tratamento e eliminação de resíduos | valorização de materiais	E
39	Descontaminação e actividades similares	E
41	Promoção imobiliária (desenvolvimento de projectos de edifícios) | construção de edifícios	F
42	Engenharia civil	F
43	Actividades especializadas de construção	F
45	Comércio, manutenção e reparação, de veículos automóveis e motociclos	G
46	Comércio por grosso (inclui agentes), excepto de veículos automóveis e motociclos	G
47	Comércio a retalho, excepto de veículos automóveis e motociclos	G
49	Transportes terrestres e transportes por oleodutos ou gasodutos	H
50	Transportes por água	H
51	Transportes aéreos	H
52	Armazenagem e actividades auxiliares dos transportes(inclui manuseamento)	H
53	Actividades postais e de courier	H
55	Alojamento	I
56	Restauração e similares	I
58	Actividades de edição	J
59	Actividades cinematográficas, de vídeo, de produção de programas de televisão, de gravação de som e de edição de música	J
60	Actividades de rádio e de televisão	J
61	Telecomunicações	J
62	Consultoria e programação informática e actividades relacionadas	J
63	Actividades dos serviços de informação	J
64	Actividades de serviços financeiros, excepto seguros e fundos de pensões	K
65	Seguros, resseguros e fundos de pensões, excepto segurança social obrigatória	K
66	Actividades auxiliares de serviços financeiros e dos seguros	K
68	Actividades imobiliárias	L
69	Actividades jurídicas e de contabilidade	M
70	Actividades das sedes sociais e de consultoria para a gestão	M
71	Actividades de arquitectura, de engenharia e técnicas afins | actividades de ensaios e de análises técnicas	M
72	Actividades de Investigação científica e de desenvolvimento	M
73	Publicidade, estudos de mercado e sondagens de opinião	M
74	Outras actividades de consultoria, científicas, técnicas e similares	M
75	Actividades veterinárias	M
77	Actividades de aluguer	N
78	Actividades de emprego	N

79	Agências de viagem, operadores turísticos, outros serviços de reservas e actividades relacionadas	N
80	Actividades de investigação e segurança	N
81	Actividades relacionadas com edifícios, plantação e manutenção de jardins	N
82	Actividades de serviços administrativos e de apoio prestados às empresas	N
84	Administração Pública e Defesa | Segurança Social Obrigatória	O
85	Educação	P
86	Actividades de saúde humana	Q
87	Actividades de apoio social com alojamento	Q
88	Actividades de apoio social sem alojamento	Q
90	Actividades de teatro, de música, de dança e outras actividades artísticas e literárias	R
91	Actividades das bibliotecas, arquivos, museus e outras actividades culturais	R
92	Lotarias e outros jogos de aposta	R
93	Actividades desportivas, de diversão e recreativas	R
94	Actividades das organizações associativas	S
95	Reparação de computadores e de bens de uso pessoal e doméstico	S
96	Outras actividades de serviços pessoais	S
97	Actividades das famílias empregadoras de pessoal doméstico	T
98	Actividades de produção de bens e serviços pelas famílias para uso próprio	T
99	Actividades dos organismos internacionais e outras instituições extra-territoriais	U

Tabela Paises

Continente	paisEn	paisPT
Africa	Algeria	Argélia
Africa	Angola	Angola
Africa	Benin	Benin
Africa	Botswana	Botsuana
Africa	Burkina Faso	Burkina Faso
Africa	Burundi	Burundi
Africa	Cameroon	Camarões
Africa	Cape Verde	Cabo Verde
Africa	Central African Republic	República Centro-Africana
Africa	Chad	Chade
Africa	Comoros	Comores
Africa	Congo, Democratic Republic of the	República Democrática do Congo
Africa	Congo, Republic of the	Congo
Africa	Cote d'Ivoire	Costa do Marfim
Africa	Djibouti	Djibouti

Africa	Egypt	Egito
Africa	Equatorial Guinea	Guiné Equatorial
Africa	Eritrea	Eritreia
Africa	Ethiopia	Etiópia
Africa	Gabon	Gabão
Africa	Gambia	Gâmbia
Africa	Ghana	Gana
Africa	Guinea	Guiné
Africa	Guinea-Bissau	Guiné-Bissau
Africa	Kenya	Quênia
Africa	Lesotho	Lesoto
Africa	Liberia	Libéria
Africa	Libya	Líbia
Africa	Madagascar	Madagascar
Africa	Malawi	Malawi
Africa	Mali	Mali
Africa	Mauritania	Mauritânia
Africa	Mauritius	Maurício
Africa	Morocco	Marrocos
Africa	Mozambique	Moçambique
Africa	Namibia	Namíbia
Africa	Niger	Níger
Africa	Nigeria	Nigéria
Africa	Rwanda	Ruanda
Africa	Sao Tome and Principe	São Tomé e Príncipe
Africa	Senegal	Senegal
Africa	Seychelles	Seychelles
Africa	Sierra Leone	Serra Leoa
Africa	Somalia	Somália
Africa	South Africa	África do Sul
Africa	South Sudan	Sudão do Sul
Africa	Sudan	Sudão
Africa	Swaziland (Eswatini)	Suazilândia
Africa	Tanzania	Tanzânia
Africa	Togo	Togo
Africa	Tunisia	Tunísia
Africa	Uganda	Uganda
Africa	Zambia	Zâmbia
Africa	Zimbabwe	Zimbábue
Asia	Afghanistan	Afeganistão
Asia	Armenia	Armênia
Asia	Azerbaijan	Azerbaijão
Asia	Bahrain	Bahrein
Asia	Bangladesh	Bangladesh

Asia	Bhutan	Butão
Asia	Brunei	Brunei
Asia	Cambodia	Camboja
Asia	China	China
Asia	Georgia	Geórgia
Asia	India	Índia
Asia	Indonesia	Indonésia
Asia	Iran	Irã
Asia	Iraq	Iraque
Asia	Israel	Israel
Asia	Japan	Japão
Asia	Jordan	Jordânia
Asia	Kazakhstan	Cazaquistão
Asia	Kuwait	Kuwait
Asia	Kyrgyzstan	Quirguistão
Asia	Laos	Laos
Asia	Lebanon	Líbano
Asia	Malaysia	Malásia
Asia	Maldives	Maldivas
Asia	Mongolia	Mongólia
Asia	Myanmar (Burma)	Mianmar (Birmânia)
Asia	Nepal	Nepal
Asia	North Korea	Coreia do Norte
Asia	Oman	Omã
Asia	Pakistan	Paquistão
Asia	Palestine	Palestina
Asia	Philippines	Filipinas
Asia	Qatar	Catar
Asia	Russia	Rússia
Asia	Saudi Arabia	Arábia Saudita
Asia	Singapore	Singapura
Asia	South Korea	Coreia do Sul
Asia	Sri Lanka	Sri Lanka
Asia	Syria	Síria
Asia	Taiwan	Taiwan
Asia	Tajikistan	Tadjiquistão
Asia	Thailand	Tailândia
Asia	Timor-Leste (East Timor)	Timor-Leste
Asia	Turkey	Turquia
Asia	Turkmenistan	Turcomenistão
Asia	United Arab Emirates	Emirados Árabes Unidos
Asia	Uzbekistan	Uzbequistão
Asia	Vietnam	Vietnã
Asia	Yemen	Iêmen

Europe	Albania	Albânia
Europe	Andorra	Andorra
Europe	Austria	Áustria
Europe	Belarus	Bielorrússia
Europe	Belgium	Bélgica
Europe	Bosnia and Herzegovina	Bósnia e Herzegovina
Europe	Bulgaria	Bulgária
Europe	Croatia	Croácia
Europe	Cyprus	Chipre
Europe	Czech Republic	República Tcheca
Europe	Denmark	Dinamarca
Europe	Estonia	Estônia
Europe	Finland	Finlândia
Europe	France	França
Europe	Germany	Alemanha
Europe	Greece	Grécia
Europe	Hungary	Hungria
Europe	Iceland	Islândia
Europe	Ireland	Irlanda
Europe	Italy	Itália
Europe	Kosovo	Kosovo
Europe	Latvia	Letônia
Europe	Liechtenstein	Liechtenstein
Europe	Lithuania	Lituânia
Europe	Luxembourg	Luxemburgo
Europe	Malta	Malta
Europe	Moldova	Moldávia
Europe	Monaco	Mônaco
Europe	Montenegro	Montenegro
Europe	Netherlands	Países Baixos
Europe	North Macedonia (Macedonia)	Macedônia do Norte
Europe	Norway	Noruega
Europe	Poland	Polônia
Europe	Portugal	Portugal
Europe	Romania	Romênia
Europe	San Marino	San Marino
Europe	Serbia	Sérvia
Europe	Slovakia	Eslováquia
Europe	Slovenia	Eslovênia
Europe	Spain	Espanha
Europe	Sweden	Suécia
Europe	Switzerland	Suíça
Europe	Ukraine	Ucrânia
Europe	United Kingdom	Reino Unido

North America	Antigua and Barbuda	Antígua e Barbuda
North America	Bahamas	Bahamas
North America	Barbados	Barbados
North America	Belize	Belize
North America	Canada	Canadá
North America	Costa Rica	Costa Rica
North America	Cuba	Cuba
North America	Dominica	Dominica
North America	Dominican Republic	República Dominicana
North America	El Salvador	El Salvador
North America	Grenada	Granada
North America	Guatemala	Guatemala
North America	Haiti	Haiti
North America	Honduras	Honduras
North America	Jamaica	Jamaica
North America	Mexico	México
North America	Nicaragua	Nicarágua
North America	Panama	Panamá
North America	Saint Kitts and Nevis	São Cristóvão e Névis
North America	Saint Lucia	Santa Lúcia
North America	Saint Vincent and the Grenadines	São Vicente e Granadinas
North America	Trinidad and Tobago	Trinidad e Tobago
North America	United States of America (USA)	Estados Unidos da América (EUA)
Oceania	Australia	Austrália
Oceania	Fiji	Fiji
Oceania	Kiribati	Kiribati

Oceania	Marshall Islands	Ilhas Marshall
Oceania	Micronesia	Micronésia
Oceania	Nauru	Nauru
Oceania	New Zealand	Nova Zelândia
Oceania	Palau	Palau
Oceania	Papua New Guinea	Papua Nova Guiné
Oceania	Samoa	Samoa
Oceania	Solomon Islands	Ilhas Salomão
Oceania	Tonga	Tonga
Oceania	Tuvalu	Tuvalu
Oceania	Vanuatu	Vanuatu
South America	Argentina	Argentina
South America	Bolivia	Bolívia
South America	Brazil	Brasil
South America	Chile	Chile
South America	Colombia	Colômbia
South America	Ecuador	Equador
South America	Guyana	Guiana
South America	Paraguay	Paraguai
South America	Peru	Peru
South America	Suriname	Suriname
South America	Uruguay	Uruguai
South America	Venezuela	Venezuela
Europe	Gibraltar	Gibraltar
North America	Guadeloupe	Guadalupe
Europe	England	Inglaterra

Anexo B

Requirements.txt

```
django
gunicorn
python-dotenv
whitenoise
django-admin-datta==1.0.6
python-dotenv
djangorestframework
django-mssql-backend
drf_yasg
#API
mssql-django
django-rest-swagger
# Lib for ML
numpy
pandas
pyodbc --no-binary :all:
sqlalchemy
dtreeviz
graphviz
seaborn
pydotplus
autopep8
imblearn
dotenv
networkx
scipy==1.4.1
joblib
```

Função calcular Métricas da Decision Tree

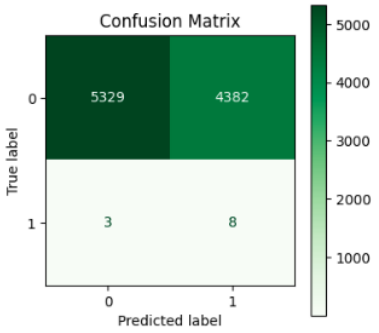
```
# Function to calculate
def cal_metrics(y_test, y_pred,clf,X,y ,cmap=None):
    cm = confusion_matrix(y_test, y_pred)
    print("Confusion Matrix: \n", cm)
    MatrixDisplay(cm, y_pred, cmap)
    # Accuracy classification score.
    Accuracy = accuracy_score(y_test, y_pred) * 100
    print("Accuracy : ", Accuracy)
    # Compute the precision.
    Precision = precision_score(y_test, y_pred) * 100
    print("Precision : ", Precision)
    # Compute the recall.
```

```
    Recall = recall_score(y_test, y_pred) * 100
    print("Recall : ", Recall)
    # Compute the F1 score, also known as balanced F-score or F-measure.
    f1 = f1_score(y_test, y_pred, average=None)
    print("F1 score : ", f1 * 100)
    # Compute precision, recall, F-measure and support for each class.
    macro = precision_recall_fscore_support(y_test, y_pred, average="macro")
    print("Precision Recall Fscore macro : ", multiplyValues(macro, 100.00))
    micro = precision_recall_fscore_support(y_test, y_pred, average="micro")
    print("Precision Recall Fscore micro : ", multiplyValues(micro, 100.00))
    weighted = precision_recall_fscore_support(y_test, y_pred, average="weighted")
    print("Precision Recall Fscore weighted : ", multiplyValues(weighted, 100.00))
    prf = precision_recall_fscore_support(y_test, y_pred, average=None)
    print("Precision Recall Fscore None : ", multiplyValues(prf, 0))
    LogLoss = log_loss(y_test, y_pred) * 100.00
    # Log loss, aka logistic loss or cross-entropy loss.
    print("Log Loss : ", LogLoss)
    # Build a text report showing the main classification metrics.
    print("Report : ", classification_report(y_test, y_pred))
    k = _number_of_folds  # number of folds
    # set the random state for reproducibility
    kf = KFold(n_splits=k, shuffle=True, random_state=_random_state)
    # Calculate precision, recall, and F1-score using classification_report
    report = cross_val_score(clf,  X, y, cv=kf,).round(4),\
            cross_val_score(clf,  X, y, cv=kf, scoring='precision_macro',).round(4), \
            cross_val_score(clf,  X, y, cv=kf, scoring='recall_macro',).round(4), \
            cross_val_score(clf,  X, y, cv=kf, scoring='f1_macro',).round(4)
    print("Row   | Accuracy   | Precision  | Recall  | F1-score")
    for i in range(_number_of_folds):
        print(f'{i}     | {report[0][i]}     | {report[1][i]}     | {report[2][i]}     | {report[3][i]}')

    print(f"Mean   | {report[0].mean().round(4)}     | {report[1].mean().round(4)}     |
{report[2].mean().round(4)}     | {report[3].mean().round(4)}")
    return f'|{"{:.6f}".format(Accuracy)}\t|\t{"{:.6f}".format(Precision)}\t|\t{"{:.6f}".format(Recall)}\t|\t{f1}'
```

Resultado da Função cal_metrics

```
Results Using Gini Index:
Predicted values:
[1 0 1 ... 1 1 1]
Confusion Matrix:
 [[5329 4382]
 [   3    8]]
```




```
Accuracy :  54.896111911129395
Precision :  0.18223234624145787
Recall :  72.7272727272273
F1 score :  [70.85022934  0.36355374]
Precision Recall Fscore macro :  (50.062984140112476, 63.80159331966561, 35.60689154016911, None)
Precision Recall Fscore micro :  (54.896111911129395, 54.896111911129395, 54.896111911129395, None)
Precision Recall Fscore weighted :  (99.83086033848204, 54.896111911129395, 70.7704768809537, None)
Precision Recall Fscore None :  (array([0.99943736, 0.00182232]), array([0.54875914, 0.72727273]), array([0.70850229, 0.00363554]), array([9711,   11]))
Log Loss :  1625.7089087767818
Report :               precision    recall  f1-score   support

           0       1.00      0.55      0.71      9711
           1       0.00      0.73      0.00        11

    accuracy                           0.55      9722
   macro avg       0.50      0.64      0.36      9722
weighted avg       1.00      0.55      0.71      9722

Row    |   Accuracy    |   Precision   |   Recall   |   F1-score
0      |   0.9984      |   0.4992      |   0.5      |   0.4996
1      |   0.9994      |   0.4997      |   0.5      |   0.4998
2      |   0.9984      |   0.4992      |   0.5      |   0.4996
3      |   0.9988      |   0.4994      |   0.5      |   0.4997
4      |   0.9992      |   0.4996      |   0.5      |   0.4998
Mean   |   0.9988      |   0.4994      |   0.5      |   0.4997
```

```
def findBestk(k_values,metric,p,data):
    best_k = None
    best_avg_distance = np.inf
    silhouette_scores = []
    precision_scores = []
    recall_scores = []
    f1_scores = []
    auroc_scores = []

    for k in k_values:
        # Train a KNN model
        knn_model = NearestNeighbors(n_neighbors=k,metric=metric,p=p)
        knn_model.fit(data)
        result = ""
        distances, _ = knn_model.kneighbors(data)
        # Calculate the average distance to the k nearest neighbors
        avg_distance = np.mean(distances[:, -1])
        # Print the average distance for the current K value
        result=f"""K ={k} | Average Distance: {"%0.4f" % avg_distance }| Max Distance:{"%0.4f" %
np.max(distances[:, -1])}| STD Distance:{"%0.4f" % np.std(distances[:, -1])}"""
```

```
        if avg_distance < best_avg_distance:
            best_avg_distance = avg_distance
            best_k = k
        # Determine the outlier labels
        outlier_labels = np.zeros(data.shape[0])
        outlier_labels[np.sum(distances, axis=1) > np.percentile(np.sum(distances, axis=1), q=95)] = 1
        # Calculate precision, recall, and F1-score
        precision = precision_score(outlier_labels, np.ones_like(outlier_labels))
        recall = recall_score(outlier_labels, np.ones_like(outlier_labels))
        f1 = f1_score(outlier_labels, np.ones_like(outlier_labels))
        precision_scores.append(precision)
        recall_scores.append(recall)
        f1_scores.append(f1)
        # Compute the outlier scores based on the maximum distance to the K nearest neighbors
        outlier_scores = np.max(distances, axis=1)
        # Set a threshold to determine outliers (you can adjust this threshold as needed)
        threshold = np.percentile(outlier_scores, 95)  # Set threshold to top 5% of scores
        # Identify outliers based on the threshold
        outliers = data[outlier_scores > threshold]
        #result = result + f" | Outliers found for k={k}: {len(outliers.toarray())}"
        result = result + f" | Outliers Max(percentile 95%): {len(outliers.toarray()):.0f}"
        # Calculate the average distance to K nearest neighbors
        avg_distances = np.mean(distances, axis=1)
        # Set a threshold to determine outliers (you can adjust this threshold as needed)
        threshold = np.percentile(avg_distances, 95)  # Set threshold to top 5% of average distances
        # Identify outliers based on the threshold
        outliers = data[avg_distances > threshold]
        result = result + f" | Outliers Mean(percentile 95%): {len(outliers.toarray()):.0f}"
        print(result)
    return best_k


def findBestthreshold(best_k,metric,p,threshold_values,data):
    # instantiate model
    nbrs = NearestNeighbors(n_neighbors=best_k,metric=metric,p=p)
    nbrs.fit(data) # fit model
    # distances and indexes of k-neaighbors from model outputs
    distances, indexes = nbrs.kneighbors(data)
    best_outlier_threshold = None
    best_outlier_ratio = -1
    best_silhouette_threshold =None
    best_silhouette_score =-1
    best_avg_dist_threshold=None
    best_avg_dist_score=float('inf')
    best_threshold_mean = None
    best_outliers_mean = float('inf')
    best_threshold_metric = None
    best_fbeta_score_metric = -1
    best_auc = -1
```

```
best_threshold_auc=None
best_threshold_gmm = None
best_score_gmm = -np.inf
# Set the beta value for the F-beta score
beta = 1.0
for threshold in threshold_values:
    # Determine the outliers based on the threshold
    outliers = distances.max(axis=1) > threshold
    # Compute the score based on the ratio of inliers to outliers
    outlier_ratio = outliers.sum() / len(outliers)
    # Update the best threshold value if necessary
    if outlier_ratio > best_outlier_ratio:
        best_outlier_ratio = outlier_ratio
        best_outlier_threshold = threshold

    outlier_scores = np.mean(distances, axis=1)
    # Detect outliers based on the threshold
    outliers = np.where(outlier_scores > threshold)[0]
    # Compute the silhouette score
    inliers = np.setdiff1d(np.arange(len(data.toarray())), outliers)
    labels = pd.DataFrame(data=data).index.isin(outliers).astype(int)
    score_silhouette = silhouette_score(data.toarray(), labels=labels)
    # Update the best threshold and score if necessary
    if score_silhouette > best_silhouette_score:
        best_silhouette_threshold = threshold
        best_silhouette_score = score_
    num_outliers_mean = len(outliers)  # Count the number of outliers
    # Update the best threshold and best number of outliers if necessary
    if num_outliers_mean < best_outliers_mean:
        best_threshold_mean = threshold
        best_outliers_mean = num_outliers_mean

    labels = pd.DataFrame(data=data).index.isin(outliers).astype(int)
    # Compute precision, recall, and F-beta score
    l = len(data.toarray())
    precision = precision_score(labels, np.ones(l))
    recall = recall_score(labels, np.ones(l))
    fbeta = fbeta_score(labels, np.ones(l), beta=beta)
    # Compute the trade-off metric (F-beta score in this case)
    tradeoff_metric = (1 + beta**2) * (precision * recall) / (beta**2 * precision + recall)
    # Update the best threshold and best F-beta score if necessary
    if tradeoff_metric > best_fbeta_score_metric:
        best_threshold_metric = threshold
        best_fbeta_score_metric = tradeoff_metric

    labels = np.zeros(l)
    labels[outliers] = 1
    # Compute the false positive rate, true positive rate, and thresholds
    fpr, tpr, thresholds = roc_curve(labels, outlier_scores)
```

```
        auc_score = auc(fpr, tpr) # Compute the AUC
        # Update the best threshold and best AUC if necessary
        if auc_score > best_auc:
            best_threshold_auc = threshold
            best_auc = auc_score
        # Create a binary label array indicating outliers (1) and inliers (0)
        labels = np.where(outlier_scores > threshold, 1, 0)
        # Fit the Gaussian Mixture Model
        gmm = GaussianMixture(n_components=2)
        gmm.fit(data.toarray())
        # Compute the log-likelihood of each sample
        log_likelihoods = gmm.score_samples(data.toarray())
        # Compute the score based on the log-likelihood and labels
        score = np.mean(log_likelihoods[labels == 0]) - np.mean(log_likelihoods[labels == 1])
        # Update the best threshold and best score if necessary
        if score > best_score_gmm:
            best_threshold_gmm = threshold
            best_score_gmm = score

    threshold =
np.array([best_outlier_threshold,best_silhouette_threshold,best_threshold_mean,best_threshold_metric,best
_threshold_auc,best_threshold_gmm])
    scores
=np.array([best_outlier_ratio,best_silhouette_score,best_outliers_mean,best_fbeta_score_metric,best_auc,be
st_score_gmm])
    best_threshold = scores.max()
    display(best_threshold)
    plt.boxplot(distances)
    print(pd.DataFrame(distances).describe())
    print(pd.DataFrame(distances).std().mean())
    print(pd.DataFrame(distances).var().mean())
    print(pd.DataFrame(distances).mean().mean())
    #Plotting the average distance to k nearest neighbors
    avg_distances = distances.mean(axis=1)
    display(avg_distances)
    display(avg_distances.mean())
    plt.show()
    plt.plot(distances.mean(axis =1))
    plt.plot(sorted(avg_distances))
    avg_distances[avg_distances==avg_distances]=avg_distances.mean()
    plt.plot(avg_distances)
    #plt.axhline(y = avg_distances.mean(), color = 'r', linestyle = '-')
    plt.xlabel("Samples")
    plt.ylabel("Average Distance to Nearest Neighbors")
    plt.title("KNN: Average Distance to Nearest Neighbors")
    plt.show()
```

Anexo C

Métricas Decision Tree Clientes

Tecnica / Método		Unbalance					UnderSampler					ClassWeigh					SMOTE					BaggingClassifier				
		Accuracy	Precision	Recall	Lost	F1-score	Accuracy	Precision	Recall	Lost	F1-score	Accuracy	Precision	Recall	Lost	F1-score	Accuracy	Precision	Recall	Lost	F1-score	Accuracy	Precision	Recall	Lost	F1-score
Holt Out - test size 0,2	Gini	99,84	-	-	5,93	99,75	62,29	0,16	37,50	1359,14	76,62	98,97	-	-	37,07	99,32	76,98	69,61	97,27	829,55	75,90	99,84	-	-	5,93	99,75
	Entropy	99,84	-	-	5,93	99,75	62,29	0,16	37,50	1359,14	76,62	98,97	-	-	37,07	99,32	77,40	69,78	98,08	814,70	76,29	99,84	-	-	5,93	99,75
K-Fold Cross-Validation CV =5	Gini	99,88	99,76	99,88	-4,30	99,82	55,00	54,80	55,00	-16,22	53,33	66,47	99,76	66,47	-12,08	77,29	99,88	99,76	99,88	-4,30	99,82	99,88	99,76	99,88	-4,30	99,82
	Entropy	99,88	99,76	99,88	-4,30	99,82	55,00	54,80	55,00	-16,22	53,33	65,77	99,77	65,77	-12,34	75,45	99,88	99,76	99,88	-4,30	99,82	99,88	99,76	99,88	-4,30	99,82
Holt Out - test size 0,3	Gini	99,86	-	-	4,94	99,79	41,91	0,21	90,00	2093,81	58,91	24,95	0,16	90,00	2705,25	39,76	77,27	69,85	97,22	819,31	76,25	99,86	-	-	4,94	99,79
	Entropy	99,86	-	-	4,94	99,79	41,91	0,21	90,00	2093,81	58,91	24,95	0,16	90,00	2705,25	39,76	77,28	69,86	97,23	819,06	76,25	99,86	-	-	4,94	99,79
K-Fold Cross-Validation CV =5	Gini	99,88	99,76	99,88	-4,30	99,82	42,12	39,82	42,14	-2085,38	38,24	66,47	99,60	66,47	-12,08	77,29	99,88	99,76	99,88	-4,30	99,82	99,88	99,76	99,88	-4,30	99,82
	Entropy	99,88	99,76	99,88	-4,30	99,82	42,12	39,82	42,14	-2085,38	38,24	625,77	99,77	65,77	-12,34	75,45	99,88	99,76	99,88	-4,30	99,82	99,88	99,76	99,88	-4,30	99,82
Holt Out - test size 0,4	Gini	99,89	-	-	4,08	99,83	54,90	0,18	72,73	1625,71	70,77	98,97	-	-	37,07	99,32	77,06	69,67	97,12	826,81	76,02	99,84	-	-	5,93	99,75
	Entropy	99,89	-	-	4,08	99,83	54,90	0,18	72,73	1625,71	70,77	98,97	-	-	37,07	99,32	77,07	69,67	97,13	826,62	76,02	99,84	-	-	5,93	99,75
K-Fold Cross-Validation CV =5	Gini	99,84	99,76	99,88	-4,30	99,82	53,57	64,48	53,57	-16,73	53,19	66,47	99,76	66,47	-12,08	77,29	98,88	99,76	99,88	-4,30	99,82	99,88	99,76	99,88	-4,30	99,82
	Entropy	99,84	99,76	99,88	-4,30	99,82	53,57	64,48	53,57	-16,73	53,19	65,77	99,77	65,77	-12,34	75,45	98,88	99,76	99,88	-4,30	99,82	99,88	99,76	99,88	-4,30	99,82
GridSearchCV 'criterion': ['gini', 'entropy'], 'max_depth': [None, 3, 5, 7,10,20], 'min_samples_split': [1, 2,3 ,4,5,6,7,8,9, 10,20], 'min_samples_leaf': [1, 2,3, 4,5,6,7,8,9]		0,0000%					68,8132%					0,4891%					81,7397%					0,0000%				
		Best parameters: { 'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2 }					Best parameters: { 'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 2 }					Best parameters: { 'class_weight': { 0: 0,5005972977426265, 1: 419,05172413793105 }, 'criterion': 'entropy', 'max_depth': 7, 'min_samples_leaf': 3, 'min_samples_split': 2 }					Best parameters: { 'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2 }					Best parameters: { 'base_estimator__criterion': 'gini', 'base_estimator__max_depth': None, 'base_estimator__min_samples_leaf': 1, 'base_estimator__min_samples_split': 2, 'max_features': 0,5, 'max_samples': 0,5, 'n_estimators': 10 }				

Anexo D

Calcular Melhor Métricas da Decision Tree

```
X = dt.drop("fraude", axis=1)
y = dt["fraude"]
transformer = make_column_transformer(
    (OneHotEncoder(), ["estado", "setorinstituicional", "cae_seccao", "Continente"]),
    remainder="passthrough",
)
transformer.fit(dt.drop("fraude", axis=1))
X_train = transformer.transform(X) # Fit and transform the training data using the transformer
dt_classifier = DecisionTreeClassifier()
param_grid = {# Define the parameter grid for grid search
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 3, 5, 7,10,20],
    'min_samples_split': [1, 2,3 ,4,5,6,7,8,9, 10,20],
    'min_samples_leaf': [1, 2,3, 4,5,6,7,8,9]
}
scoring = { 'accuracy': 'accuracy', 'precision': 'precision', 'recall': 'recall', 'f1': 'f1'}
# Perform grid search with cross-validation
grid_search = GridSearchCV(dt_classifier, param_grid, scoring=scoring,refit="f1" ,cv=5)
grid_search.fit(X_train, y)
# Print the best parameters and the corresponding mean cross-validated score
print("Best parameters:", grid_search.best_params_)
print("Best score:", grid_search.best_score_)
```

Calcular Melhor Métricas da Decision Tree dos Resultados do Isolation Forest Transactions

```
dt['outlier']=  np.where(outliers==-1, 1, 0)
X = dt.drop("outlier", axis=1)
y = dt["outlier"]
transformer = make_column_transformer(  (OneHotEncoder(), ["CCKey","IP","Status","CardType","PosId","cae_seccao","Continente"]),
    remainder="passthrough",
)
transformer.fit(dt.drop("outlier", axis=1))
dt_classifier = DecisionTreeClassifier()
param_grid = {# Define the parameter grid for grid search
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 3, 5, 7,10,20],
    'min_samples_split': [1, 2,3 ,4,5,6,7,8,9, 10,20],
    'min_samples_leaf': [1, 2,3, 4,5,6,7,8,9]
}
scoring = { 'accuracy': 'accuracy', 'precision': 'precision', 'recall': 'recall', 'f1': 'f1'}
# Perform grid search with cross-validation
grid_search = GridSearchCV(dt_classifier, param_grid, scoring=scoring,refit="f1" ,cv=5)
grid_search.fit(transformer.transform(X), y)
# Print the best parameters and the corresponding mean cross-validated score
```

```
print("Best parameters:", grid_search.best_params_)
print("Best score:", grid_search.best_score_)
```

Calcular Melhor Métricas da Decision Tree dos Resultados do KNN Transactions

```
clf = DecisionTreeClassifier()# Create an Isolation Forest instance
param_grid = {# Define the parameter grid for grid search
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 3, 5, 7,10,20],
    'min_samples_split': [1, 2,3 ,4,5,6,7,8,9, 10,20],
    'min_samples_leaf': [1, 2,3, 4,5,6,7,8,9]
}
scoring = {
    'accuracy': 'accuracy',
    'precision': 'precision',
    'recall': 'recall',
    'f1': 'f1'
}
# Perform grid search using the custom scoring function
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring=scoring, refit='f1')
grid_search.fit(X_train,dt_Knn_trains['outlier'])
```

Calcular Melhor Métricas da Decision Tree dos Resultados do Isolation Forest TransactionsMB

```
dt = dc
dt = dt.drop("ID", axis=1, errors="ignore")
transformer = make_column_transformer(
    (OneHotEncoder(), ["subentidade","ip","numeroid","cae_seccao","Continente"]),
    remainder="passthrough",
)
transformer.fit(dt)
X_train = transformer.transform(dt)
# Create an Isolation Forest instance
clf = IsolationForest()
# Define the parameter grid for grid search
param_grid = {
    'n_estimators': [50, 100, 200],
    'contamination': [0.05, 0.1, 0.15]
}
scoring = {
    'accuracy': 'accuracy',
    'precision': 'precision',
    'recall': 'recall',
    'f1': 'f1'
}
# Perform grid search using the custom scoring function
grid_search = GridSearchCV(clf, param_grid, cv=5, scoring=scoring, refit='f1')
grid_search.fit(X_train, np.ones(X_train.shape[0]))
```

Anexo E

Decision Tree Rules Clients:

```
|--- cae_seccao_G <= 0.00
|   |--- cae_seccao_C <= 0.00
|   |   |--- setorinstituicional_S201 <= 0.00
|   |   |   |--- cae_seccao_R <= 0.00
|   |   |   |   |--- cae_seccao_H <= 0.00
|   |   |   |   |   |--- cae_seccao_I <= 0.00
|   |   |   |   |   |   |--- cae_seccao_Q <= 0.00
|   |   |   |   |   |   |   |--- cae_seccao_N <= 0.50
|   |   |   |   |   |   |   |   |--- cae_seccao_- <= 0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- cae_seccao_- >  0.50
|   |   |   |   |   |   |   |   |   |--- estado_Activo <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |   |   |   |   |   |--- setorinstituicional_S14 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- setorinstituicional_S14 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- cae_seccao_N >  0.50
|   |   |   |   |   |   |   |   |--- setorinstituicional_S11 <= 0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- setorinstituicional_S11 >  0.50
|   |   |   |   |   |   |   |   |   |--- estado_Activo <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- cae_seccao_Q >  0.00
|   |   |   |   |   |   |   |--- cae_seccao_Q <= 1.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- cae_seccao_Q >  1.00
|   |   |   |   |   |   |   |   |--- setorinstituicional_S14 <= 0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- setorinstituicional_S14 >  0.50
|   |   |   |   |   |   |   |   |   |--- estado_N/C <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- estado_N/C >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- cae_seccao_I >  0.00
|   |   |   |   |   |   |--- cae_seccao_N <= 0.00
|   |   |   |   |   |   |   |--- setorinstituicional_S11 <= 0.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- setorinstituicional_S11 >  0.50
|   |   |   |   |   |   |   |   |--- estado_Activo <= 0.50
```

```
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- cae_seccao_N >  0.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- cae_seccao_H >  0.00
|   |   |   |   |   |--- cae_seccao_- <= 0.00
|   |   |   |   |   |   |--- setorinstituicional_S14 <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- setorinstituicional_S14 >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- cae_seccao_- >  0.00
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- cae_seccao_R >  0.00
|   |   |   |   |--- cae_seccao_R <= 1.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- cae_seccao_R >  1.00
|   |   |   |   |   |--- setorinstituicional_S11 <= 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- setorinstituicional_S11 >  0.50
|   |   |   |   |   |   |--- estado_Activo <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |--- setorinstituicional_S201 >  0.00
|   |   |   |--- setorinstituicional_S14 <= 0.00
|   |   |   |   |--- cae_seccao_- <= 0.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- cae_seccao_- >  0.50
|   |   |   |   |   |--- estado_Activo <= 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |   |--- Continente_Europe <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Continente_Europe >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |--- setorinstituicional_S14 >  0.00
|   |   |   |   |--- class: 1
|   |--- cae_seccao_C >  0.00
|   |   |--- cae_seccao_C <= 1.00
|   |   |   |--- class: 1
|   |   |--- cae_seccao_C >  1.00
|   |   |   |--- setorinstituicional_S14 <= 0.00
|   |   |   |   |--- estado_Activo <= 0.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |--- setorinstituicional_S11 <= 0.50
```

```
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- setorinstituicional_S11 >  0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- setorinstituicional_S14 >  0.00
|   |   |   |   |--- setorinstituicional_S14 <= 1.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- setorinstituicional_S14 >  1.00
|   |   |   |   |   |--- estado_Inactivo <= 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- estado_Inactivo >  0.50
|   |   |   |   |   |   |--- class: 0
|--- cae_seccao_G >  0.00
|   |--- cae_seccao_G <= 1.00
|   |   |--- class: 1
|   |--- cae_seccao_G >  1.00
|   |   |--- estado_Inactivo <= 0.00
|   |   |   |--- setorinstituicional_S201 <= 0.00
|   |   |   |   |--- estado_Activo <= 0.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- estado_Activo >  0.50
|   |   |   |   |   |--- setorinstituicional_S14 <= 0.50
|   |   |   |   |   |   |--- setorinstituicional_S11 <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- setorinstituicional_S11 >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- setorinstituicional_S14 >  0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- setorinstituicional_S201 >  0.00
|   |   |   |   |--- setorinstituicional_S201 <= 1.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- setorinstituicional_S201 >  1.00
|   |   |   |   |   |--- Continente_Europe <= 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Continente_Europe >  0.50
|   |   |   |   |   |   |--- class: 1
|   |   |--- estado_Inactivo >  0.00
|   |   |   |--- estado_Inactivo <= 1.00
|   |   |   |   |--- class: 1
|   |   |   |--- estado_Inactivo >  1.00
|   |   |   |   |--- setorinstituicional_S14 <= 0.00
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- setorinstituicional_S14 >  0.00
|   |   |   |   |   |--- setorinstituicional_S11 <= 0.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- setorinstituicional_S11 >  0.00
|   |   |   |   |   |   |--- class: 1
```

Decision Tree Rules Transaction Isolation Forest:

```
|--- CCKey_AMX-241021 <= 0.50
|   |--- PosId_10025269 <= 0.50
|   |   |--- CCKey_YGG-790444 <= 0.50
|   |   |   |--- IP_52.149.108.159 <= 0.50
|   |   |   |   |--- CCKey_ULG-983048 <= 0.50
|   |   |   |   |   |--- CCKey_PHH-352962 <= 0.50
|   |   |   |   |   |   |--- Status_ERROR <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Status_ERROR >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- CCKey_PHH-352962 >  0.50
|   |   |   |   |   |   |--- Status_INITIATED <= 0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Status_INITIATED >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- CCKey_ULG-983048 >  0.50
|   |   |   |   |   |--- IP_52.149.108.255 <= 0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- IP_52.149.108.255 >  0.50
|   |   |   |   |   |   |--- StatusDateTime <= 21.59
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- StatusDateTime >  21.59
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- IP_52.149.108.159 >  0.50
|   |   |   |   |--- PosId_10025241 <= 0.50
|   |   |   |   |   |--- cae_seccao_G <= 0.50
|   |   |   |   |   |   |--- cae_seccao_- <= 0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- cae_seccao_- >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- cae_seccao_G >  0.50
|   |   |   |   |   |   |--- PosId_10025255 <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- PosId_10025255 >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- PosId_10025241 >  0.50
|   |   |   |   |   |--- CCKey_XTQ-619845 <= 0.50
|   |   |   |   |   |   |--- CCKey_DBE-323370 <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- CCKey_DBE-323370 >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- CCKey_XTQ-619845 >  0.50
|   |   |   |   |   |   |--- class: 1
|   |   |--- CCKey_YGG-790444 >  0.50
```

```
|   |   |   |--- RequestType <= 1.50
|   |   |   |   |--- RequestDateTime <= 19.83
|   |   |   |   |   |--- Amount <= 17.66
|   |   |   |   |   |   |--- RequestDateTime <= 14.65
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- RequestDateTime >  14.65
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Amount >  17.66
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- RequestDateTime >  19.83
|   |   |   |   |   |--- class: 1
|   |   |   |--- RequestType >  1.50
|   |   |   |   |--- class: 1
|   |--- PosId_10025269 >  0.50
|   |   |--- Cdx_OpCode <= 52.00
|   |   |   |--- CCKey_NDP-426164 <= 0.50
|   |   |   |   |--- CCKey_PPL-260943 <= 0.50
|   |   |   |   |   |--- CCKey_REA-578920 <= 0.50
|   |   |   |   |   |   |--- CardType_VISA <= 0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- CardType_VISA >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- CCKey_REA-578920 >  0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- CCKey_PPL-260943 >  0.50
|   |   |   |   |   |--- IP_213.228.178.18 <= 0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- IP_213.228.178.18 >  0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- CCKey_NDP-426164 >  0.50
|   |   |   |   |--- CardType_VISA <= 0.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- CardType_VISA >  0.50
|   |   |   |   |   |--- Status_SUCCESS <= 0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- Status_SUCCESS >  0.50
|   |   |   |   |   |   |--- StatusDateTime <= 21.57
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- StatusDateTime >  21.57
|   |   |   |   |   |   |   |--- class: 1
|   |   |--- Cdx_OpCode >  52.00
|   |   |   |--- IP_94.46.170.249 <= 0.50
|   |   |   |   |--- RequestDateTime <= 21.86
|   |   |   |   |   |--- IP_52.149.108.159 <= 0.50
|   |   |   |   |   |   |--- IP_52.149.109.11 <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- IP_52.149.109.11 >  0.50
```

```
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- IP_52.149.108.159 >  0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- RequestDateTime >  21.86
|   |   |   |   |   |--- IP_20.50.135.254 <= 0.50
|   |   |   |   |   |   |--- CCKey_GFK-860369 <= 0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- CCKey_GFK-860369 >  0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- IP_20.50.135.254 >  0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- IP_94.46.170.249 >  0.50
|   |   |   |   |--- class: 1
|--- CCKey_AMX-241021 >  0.50
|   |--- IP_20.50.135.254 <= 0.50
|   |   |--- class: 1
|   |--- IP_20.50.135.254 >  0.50
|   |   |--- StatusDateTime <= 14.97
|   |   |   |--- RequestDateTime <= 13.82
|   |   |   |   |--- RequestDateTime <= 13.64
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- RequestDateTime >  13.64
|   |   |   |   |   |--- StatusDateTime <= 13.67
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- StatusDateTime >  13.67
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- RequestDateTime >  13.82
|   |   |   |   |--- Amount <= 180.67
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Amount >  180.67
|   |   |   |   |   |--- class: 1
|   |   |--- StatusDateTime >  14.97
|   |   |   |--- class: 1
```

Decision Tree Rules Transaction KNN:

```
|--- Amount <= 5250.73
|   |--- Amount <= 3184.43
|   |   |--- Amount <= 2639.15
|   |   |   |--- CCKey_MHK-782989 <= 0.50
|   |   |   |   |--- Amount <= 1760.00
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Amount >  1760.00
|   |   |   |   |   |--- Cdx_OpCode <= -24.00
|   |   |   |   |   |   |--- RequestDateTime <= 18.32
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- RequestDateTime >  18.32
|   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |--- Cdx_OpCode >  -24.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- CCKey_MHK-782989 >  0.50
|   |   |   |   |--- Status_ERROR <= 0.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Status_ERROR >  0.50
|   |   |   |   |   |--- class: 0
|   |   |--- Amount >  2639.15
|   |   |   |--- Cdx_OpCode <= -4.50
|   |   |   |   |--- class: 0
|   |   |   |--- Cdx_OpCode >  -4.50
|   |   |   |   |--- class: 0
|   |--- Amount >  3184.43
|   |   |--- Cdx_OpCode <= 2.50
|   |   |   |--- StatusDateTime <= 18.44
|   |   |   |   |--- StatusDateTime <= 9.67
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- StatusDateTime >  9.67
|   |   |   |   |   |--- Amount <= 4668.37
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Amount >  4668.37
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- StatusDateTime >  18.44
|   |   |   |   |--- class: 0
|   |   |--- Cdx_OpCode >  2.50
|   |   |   |--- RequestDateTime <= 22.19
|   |   |   |   |--- class: 0
|   |   |   |--- RequestDateTime >  22.19
|   |   |   |   |--- class: 0
|--- Amount >  5250.73
|   |--- IP_52.149.108.255 <= 0.50
|   |   |--- PosId_10025211 <= 0.50
|   |   |   |--- Amount <= 17806.80
|   |   |   |   |--- cae_seccao_C <= 0.50
|   |   |   |   |   |--- RequestDateTime <= 19.17
|   |   |   |   |   |   |--- Cdx_OpCode <= -4.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Cdx_OpCode >  -4.50
|   |   |   |   |   |   |   |--- Amount <= 6764.23
|   |   |   |   |   |   |   |   |--- CardType_NA <= 0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- CardType_NA >  0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Amount >  6764.23
|   |   |   |   |   |   |   |   |--- PosId_10025251 <= 0.50
|   |   |   |   |   |   |   |   |   |--- Amount <= 7938.66
|   |   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |   |   |   |   |--- Amount >  7938.66
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- PosId_10025251 >  0.50
|   |   |   |   |   |   |   |   |--- StatusDateTime <= 12.23
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- StatusDateTime >  12.23
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- RequestDateTime >  19.17
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- cae_seccao_C >  0.50
|   |   |   |   |   |--- class: 0
|   |   |   |--- Amount >  17806.80
|   |   |   |   |--- class: 1
|   |   |--- PosId_10025211 >  0.50
|   |   |   |--- class: 0
|   |--- IP_52.149.108.255 >  0.50
|   |   |--- StatusDateTime <= 13.75
|   |   |   |--- class: 0
|   |   |--- StatusDateTime >  13.75
|   |   |   |--- class: 0
```

Anexo F

Tempos de Execução

Data Set	Algoritmo	Etapa	Tempo (min)	Tempo (h)	Tempo (dias)
Clients	Desision Tree	Recolha de Dados e Tratamento	15	0,25	0,0
		Procurar Metricas	533	8,88	0,4
		Treino do Modelo	10	0,17	0,0
		Previção	1	0,02	0,0
Transations	IsolationForest	Recolha de Dados e Tratamento	50	0,83	0,0
		Procurar Metricas	50	0,83	0,0
		Procurar Metricas para DT	1050	17,50	0,7
		Treino do Modelo	15	0,25	0,0
		Previção	1	0,02	0,0
	Knn	Recolha de Dados e Tratamento	50	0,83	0,0
		Procurar Metricas	3450	57,50	2,4
		Procurar Metricas para DT	890	14,83	0,6
		Treino do Modelo	35	0,58	0,0
		Previção	5	0,08	0,0
TransationMB	IsolationForest	Recolha de Dados e Tratamento	300	5,00	0,21
		Procurar Metricas	4250**	70,83	2,95
		Treino do Modelo	-	-	-
		Previção	-	-	-
	Knn	Recolha de Dados e Tratamento	300	5,00	0,2
		Procurar Metricas	-	-	250*

*previsão, foi testado 10000 num loop e os primeiros 6 testes rondavam a volta de 1h. Como são mais de 6 000 000 de registo resulta em 6000h = 6000/24=250 dias

**previsão, o algoritmo até ao memento da entrega encontrava-se a correr. Devido a tal o tempo pode ser bem superior ao apresentado.