Bruno Ricardo Barbosa Miranda Valente

ACM – Active Campaign Manager

Relatório de Dissertação/Estágio
Mestrado em Engenharia Informática
orientada por Prof. Álvaro Rocha e Eng. Maria Castro
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Janeiro de 2018

· U C ·

Universidade de Coimbra

Resumo

Com as empresas de telecomunicação a serem forçadas a competirem pela confiança dos clientes, elas tendem a melhorar os seus produtos a cada dia que passa, com o objetivo de apresentar serviços que impressionem a grande maioria do mercado. Neste projeto foi desenvolvido um *software* que permite identificar quais os clientes que se encontram mais suscetíveis a uma determinada campanha publicitária. Pois, é possível aumentar as receitas das empresas aderentes, reduzindo o envio de campanhas desinteressantes para os clientes, traduzindo-se no aumento da satisfação de ambas as partes. Assim é utilizado um modelo de *Data Mining* para tratar a informação, modelar o problema e obter resultados. Na realização da modelação foi utilizado o algoritmo *Perceptron* e um conjunto de métricas para testar a sua performance. Os dados são provenientes da plataforma ACM desenvolvida na empresa Altice labs que contém as informações de utilização dos dispositivos móveis dos clientes na rede móvel. As informações foram tratadas e organizadas de forma a serem utilizadas no algoritmo. O *software* desenvolvido apresenta dois grandes objetivos, melhorar as adesões de uma determinada campanha e identificar qual a melhor campanha para um determinado cliente. Foram aumentadas as adesões às campanhas em aproximadamente 120%, com uma redução de mensagens enviadas na ordem dos 60%. A associação dos clientes a uma determinada campanha veio revolucionar o processo de operar o ACM e demonstrou-se mais eficaz que o processo manual em aproximadamente 30%. A solução desenvolvida apresenta robustez nos vários cenários testados.

Palavras-Chave

"Data mining", "Data Science", "Machine Learning", "ACM", "NBO", "OMS".

Abstract

With the telecommunication enterprises being forced to compete for the attention of new clients they tend improve their products each day with the final goal of presenting services that impress the majority of the market. In this project a software was developed that allows to identify which clients are the most susceptible to a certain publicity campaign, so that the profits of the affiliate companies will increase since their publicity is correctly directed to their targets. For that, a Data Mining model is used to process the information, model the problem and obtain results. A Perceptron algorithm and a set of metrics to test its performance was used in the modeling process. The data comes from the ACM platform developed by Altice labs which contains the information's of the mobile devices of their clients. The information's has been processed and organized to be used by the algorithm. Several tests were then executed with the algorithm to understand the effects of many possible values in its properties. This software has two main functionalities, one is to improve the accessions of a certain campaign and the other is to identify which was the best campaign for a certain client. Campaign accessions increased in approximately 120% with a decrease of sent messages around 60%. The association of the clients to a certain campaign came to revolutionize the process of operating the ACM and it has proved more effective than the manual process by approximately 30%. The developed solution shows robustness in many of the tested scenarios.

Keywords

"Data mining", "Data Science", "Machine Learning", "ACM", "NBO", "OMS".

Acrónimos

ACM	–	Active Campaign Manager
ARFF	–	Attribute-Relation File Format
BI	–	Business Intelligence
CAM	–	Campanha
CLI	–	Command Line Interface
CPT	–	Campaign Performance Tracking
CRM	–	Customer Relationship Management
CRISP-DM	–	Cross Industry Standard Process
CSV	–	Comma-Separated Values
EM	–	Expectation Maximization
GC	–	Grupo de controlo
IVR	–	Interactive Voice Response
JVM	–	Java Virtual Machine
NBO	–	Next Best Offer
NBO/A	–	Next Best Offer/Action
OMS	–	Operation and Maintenance System
PA	–	Público-alvo
REST	–	Representational State Transfer
RMI	–	Remote Method Invocation
SMS	–	Short Message Service
SVM	–	Support Vector Machine
SOAP	–	Simple Object Access Protocol
SQL	–	Structured Query Language
UC	–	Use case
URI	–	Uniform Resource Identifier

Índice

Lista de Figuras

Capítulo 1
Introdução

Este documento descreve o estágio realizado na área de *Data Mining* e *Data Analysis*, na empresa Altice Labs, no âmbito do Mestrado em Engenharia Informática do Departamento de Engenharia Informática na Faculdade de Ciências e Tecnologias da Universidade de Coimbra. Este estágio foi realizado com a orientação da Eng. Maria Castro e do Prof. Álvaro Rocha.

Nesta secção é descrita a organização onde se realizou o estágio, o contexto do problema, a motivação que levou à sua realização e a estrutura do documento.

1.1 Altice Labs

A Altice Labs foi criada em 2015 com a compra integral da PT Portugal SGPS, SA pelo valor de 7,4 mil milhões de euros pela Altice Group, líder no fornecimento de serviços de telecomunicações com presença em França, Israel, Bélgica e Luxemburgo, Portugal, Antilhas Francesas / Área do Oceano Índico e República Dominicana ("Territórios Ultramarinos") e Suíça [1].

A missão é apoiar os clientes desenvolvendo soluções com inovação tecnológica, que possibilite a criação de valor e conforto aos clientes.

1.2 Contexto do problema e motivação

Hoje em dia todas as operadoras tentam chegar aos seus clientes utilizando várias formas de publicidade. Essa publicidade é massificada por vários canais de comunicação, para que seja possível chegar ao maior número de pessoas. Com este processo é esperado que o número de adesões seja proporcional ao número de pessoas que são abordadas pela campanha. Este é um cenário idealista que os produtores de publicidade pretendem obter. Contudo, nem sempre é possível, principalmente quando se trata de uma empresa que utiliza campanhas publicitárias como forma de gerar lucro a curto e médio prazo.

A Altice Labs desenvolveu uma plataforma chamada ACM (*Active Campaign Manager*) que permite às empresas de telecomunicações desenhar e executar em tempo real campanhas e promoções contextualizadas, publicando-as automaticamente para os respetivos telemóveis dos clientes. As campanhas são geridas e configuradas a partir da plataforma, que agrega informação de contexto dos clientes, permitindo ao provedor de serviço antecipar ofertas no mercado que contribuam para a fidelização dos seus clientes. Esta ferramenta torna possível divulgar uma campanha promocional utilizando o SMS (*Short Message Service*) e o IVR (*Interactive Voice Response*) como canal de comunicação.

Esta solução veio substituir os processos manuais que existiam anteriormente, o que faz com que as operadoras clientes consigam ser consideravelmente mais rápidas a lançar em produção uma campanha do que eram anteriormente, com o uso de recursos consideravelmente mais baixos. O que acontece é que em média aproximadamente 6% dos utilizadores alvos da campanha aderem às campanhas que são lançadas, enquanto os restantes 94% continuam a utilizar o serviço sem usufruir das diversas campanhas lançadas.

Estes resultados demonstram que existe um grande desperdício de recursos e resistência por parte dos utilizadores em aderir a campanhas lançadas, fazendo com que aconteça um fenómeno denominado como poluição do canal.

A poluição do canal consiste no envio de campanhas aos clientes que não estão recetivos a este tipo de publicidade ou simplesmente o tipo de campanha apresentada não se adequa às suas necessidades. Como consequência o utilizador sente desconforto sempre que recebe uma promoção, o que baixa a probabilidade de adesão e aumenta a insatisfação do cliente. Caso seja mantido este processo por algum tempo o utilizador deixa de ler as promoções recebidas, descartando-as completamente.

Esta consequência tem vindo a ser atenuada pela adição de filtros na plataforma ACM, reduzindo o público-alvo de cada campanha e minimizando a poluição resultante. Todavia, já existe dificuldade em melhorar os filtros implementados sentindo-se a necessidade de recorrer a soluções com um carácter tecnológico diferente para resolver ou minimizar este problema.

1.3 Objetivos

O objetivo desta dissertação é aplicar um algoritmo que consiga identificar quais os clientes das operadoras que estão mais suscetíveis à adesão de determinada campanha promocional, possibilitando a sua implementação nos serviços da organização, até ao final do corrente estágio, tendo como propósito não incomodar os clientes que provavelmente não estão interessados, maximizando a taxa de adesões. Este algoritmo deve ser alimentado com os dados que são providenciados pelas empresas de telecomunicações, sendo os dados tratados e utilizados numa sequência lógica sem interrupções ou paragens. Os dados obtidos permitem avaliar os clientes, segmentar com base nos dados individuais e realizar análises preditivas e descritivas. É prioritário que exista uma comunicação permanente com as aplicações da organização para que haja uma troca de informação rápida, interferindo ao mínimo com os tempos de execução apresentados pelas aplicações. Este projeto pode ter várias aplicações dentro da organização o que torna crucial, o sistema construído ser modelar, apresentando uma separação clara do algoritmo dos restantes sistemas.

No tratamento dos dados deve-se garantir a formatação adequada para uma total compatibilidade com o algoritmo. Este processo é importante para que haja uma manipulação dos dados de forma transparente para o utilizador, permitindo unir o processo de tratamento de informação com a análise propriamente dita.

Pretende-se também disponibilizar à Altice Labs um conjunto de informações estatísticas referente à eficácia do algoritmo, contendo uma simulação dos lucros das empresas de telecomunicações caso o algoritmo estivesse em funcionamento nas campanhas lançadas anteriormente. O objetivo desta análise é mostrar à organização a viabilização deste projeto e justificar o processo de implementação do mesmo nas suas soluções.

1.4 Estrutura do documento

Esta dissertação encontra-se dividida em oito capítulos principais.

O primeiro capítulo descreve o âmbito do problema com uma pequena introdução, uma descrição da empresa onde está a decorrer o estágio, a contextualização do problema e por fim os objetivos assumidos.

No segundo capítulo são estudadas as metodologias e ferramentas ligadas à área de *Data Mining* e plataforma ACM, identificando os principais aspetos para a sua realização de forma organizada e otimizada.

No terceiro capítulo são utilizados diagramas e processos para conseguir tornar o problema o mais claro possível, identificando os pontos principais para o desenvolvimento deste projeto.

No quarto capítulo são apresentadas as etapas pertencentes ao desenvolvimento do projeto, a abordagem aos desafios encontrados e os resultados alcançados.

No quinto capítulo são os testes realizados no enquadramento do projeto, percorrendo os testes pelos diferentes atributos funcionais e não-funcionais identificados.

No sexto capítulo são apresentados os riscos que o processo de desenvolvimento deste projeto está sujeito. Inicialmente, foi realizado um levantamento dos riscos, em seguida uma avaliação e finalmente realizado um plano de mitigação para os resolver.

No sétimo capítulo é apresentado o plano realizado no início do primeiro e segundo semestre identificando as etapas necessárias para a realização do projeto. Estas informações servem para avaliar e estimar o tempo de realização do projeto e em comparação com a realidade identificar possíveis dévios.

No oitavo e último capítulo é apresentada uma conclusão do trabalho.

Capítulo 2
Análise das ferramentas e conceitos de Data Mining

2.1 Active Campaing Manager

O ACM (*Active Campaing Manager*) é uma plataforma desenvolvida pela Altice Labs que visa criar, lançar e executar campanhas para serviços de telecomunicações. Assim, é possível criar campanhas que decorrem em tempo real, possibilitando o aumento do número de clientes, de receitas e de participação no mercado partilhado [2].

A plataforma é utilizada de uma forma intuitiva utilizando um conjunto de ferramentas:

- Uma interface que permite desenhar campanhas flexíveis e robustas;
- Um conjunto de painéis que monitorizam os serviços e ferramentas que estão a correr na plataforma;
- Uma ferramenta de produção de relatórios, fáceis de interpretar para que os operadores de telecomunicação saibam em tempo real os valores de adesão às campanhas.

A plataforma possibilita que as campanhas sejam lançadas num período de tempo onde a probabilidade de adesão é maior, integração de canais de comunicação, plataformas e sistemas de CRM (*Customer Relationship Management*) [2].

O objetivo inerente ao ACM passa por:

- Conceito de "Configuração de negócios" para configurar, testar, publicar e gerenciar mudanças dentro das definições da oferta (ou parte das definições da oferta);
- Maximizar os resultados das campanhas para que os objetivos das empresas sejam cumpridos;
- Segmentação flexível por qualquer cliente ou característica de cliente, como região geográfica, idade, perfil de consumo, entre outros;
- Aumentar o número de clientes e promoção de *cross-selling*;
- Diminuir o tempo gasto no lançamento das campanhas, utilizando uma interface intuitiva;
- Centralizar um conjunto de ferramentas, facilitando a sua utilização.

Para conseguir satisfazer os objetivos indicados o ACM comprime as seguintes quatro aplicações:

- **Desenhador de campanhas**: Interface intuitiva que permite a criação, manutenção e lançamento de campanhas promocionais. Não necessita de ser instalado, facilitando a distribuição e adição de novos serviços;
- **Motor de execução em tempo real**: Especializado na execução de campanhas. São utilizados mecanismos de tolerância a falhas, maior robustez, interação com vários canais e interoperabilidade da plataforma;
- **Módulo de relatórios sobre campanhas**: Esta aplicação fornece indicadores em *near real-time* detalhados sobre o decorrer das campanhas que foram lançadas, permitindo assim a diminuição do tempo de resposta para ajustes necessários às campanhas, e adequações às mudanças de mercado;

- **Consola de operação e administração**: Aplicação que permite a operação e administração do ACM.

2.1.2 Tipos de campanhas existentes

No lançamento de campanhas, existe sempre um objetivo global, que pode ser diferente consoante o tipo de campanha que se pretende realizar. Existem neste caso seis tipos de campanhas que podem ser criadas:

- **Incentivo à recarga**: Consiste em atribuir benefícios aos clientes que carregarem uma determinada quantia de dinheiro no tarifário do telemóvel;
- **Incentivo ao consumo**: Consiste em atribuir um benefício caso os clientes efetuem um determinado número de ações;
- **Incentivo a subscrição de um serviço**: Consiste na atribuição de benefícios caso o cliente adira a um determinado serviço;
- **Retenção de clientes na operadora**: É utilizado quando um cliente não demostra atividade considerada normal nos parâmetros do operador de telecomunicações e visa aumentar a atividade do utilizador para evitar desistências da operadora em causa, *churn*.

2.1.3 Configuração de campanhas

O modo de configurar as campanhas promocionais com a ferramenta ACM influência diretamente a construção da campanha. Dependendo das restrições criadas e da forma como está desenhada a campanha, os clientes escolhidos para serem incentivados mudam drasticamente. Para se perceber as mudanças são descritas as fases cruciais para a criação de uma campanha. As fases identificadas são:

- Atribuição das características da campanha;
- Segmentação do público-alvo;
- Ciclo de vida da campanha.

Com esta análise pretende-se perceber a transformação dos dados e como podem ser utilizados num ambiente de produção

2.1.3.1 Atribuição das características da campanha

Nesta fase são atribuídas as informações que identificam cada campanha. Torna-se imprescindível para a plataforma conseguir nas diferentes etapas identificar qual a campanha que está a ser trabalhada. Assim, para cada campanha lançada são configurados os seguintes campos:

- **Código da campanha** – É um código que permite identificar a campanha nos diferentes sistemas utilizados pelo ACM.
- **Público-alvo envolvido** – Limita do ponto de vista do marketing quais os clientes adequados para uma determinada campanha. Para este passo pode ser escolhida uma das seguintes opções:
    - Acima da linha: Quando todos os clientes estão cientes da campanha;
    - Abaixo da linha: Quando só alguns clientes estão cientes da campanha;
- **Tipo de canal** – Carateriza o canal usado para anunciar a campanha. Podem ser escolhidas as seguintes opções:

    - Externas: Quando utiliza sistemas de publicidade massificada utilizando a *media* (ex.: TV, rádio, etc.);
    - Internas: Quando a campanha utiliza canais de comunicação existentes na ferramenta (ex.: chamadas, SMS, etc.);
- **Categorias de campanha** – É utilizado para categorizarem as campanhas;
- **Subcategorias das campanhas** – É utilizado para subdividir as categorias das campanhas;
- **Data inicial** – É a data estabelecida para o início da campanha;
- **Data final -** É a data estabelecida para o final da campanha.

2.1.3.2 Segmentação do público-alvo

Nesta fase é onde o utilizador vai decidir qual é o conjunto de clientes que irão estar aptos para entrar na campanha. É importante para estratégia de marketing definir bem o público-alvo para aumentar a probabilidade de adesão dos clientes.

São utilizados os seguintes campos para segmentar o público-alvo:

- **Percentagem do controlo grupo** – Aqui é definida se a campanha irá ter grupo de controlo ou não. O conceito de grupo de controlo é abordado mais à frente neste relatório;
- ***TAG* de cliente** – Este campo permite a selecionar que TAG é atribuída a cada cliente pertencente ao público-alvo da campanha. Se for uma TAG exclusiva, então o cliente será excluído da possibilidade de entrar na campanha. Se for uma TAG inclusiva, então o cliente terá a possibilidade de entrar na campanha;
- **Público-alvo da campanha** – São utilizadas expressões regulares que permitem restringir a entrada de clientes que não têm condições para aderir à campanha.

2.1.3.3 Ciclo de vida

O ciclo de vida da campanha é um grafo que permite desenho conceptual a configuração da campanha, utilizando estados que permitem visualmente identificar os pontos críticos do processo. Assim, é possível criar condições nas mudanças de estados e atribuir benefícios na fase do ciclo adequada para o utilizador. Contudo, para se perceber o processo é necessário entender um conjunto de aspetos pertencentes à plataforma que serão explicados nas secções seguintes.

2.1.3.3.1 Estados

As campanhas criadas podem ter os seguintes estados:

- ***Start* –** Todos os clientes que entram na campanha passam por este estado passando para o estado seguinte. Este é o estado que inicia o ciclo (Estado inicial);
- ***End* –** É o estado que permite terminar a execução da campanha. É sempre o último estado configurado (Estado final);
- ***All* –** Permite fazer a transição de qualquer estado existente no ciclo de vida para um outro estado configurado pelo utilizador. Permite configurar mais facilmente a transição de qualquer estado da campanha para um dado estado;
- ***New State* –** É utilizado entre o estado inicial e final, permitindo a criação de estados intermédios que dão origem a novos campanha.

2.1.3.3.2 Transições

As transições são a passagem de um estado para o outro despoletado por um evento relativo ao cliente. As transições para cada cliente acontecem uma de cada vez.

2.1.3.3.3 Eventos

Os eventos são ações que acontecem interna e externamente no ACM, que representam um conjunto de atividades. As atividades externas são normalmente atividades realizadas pelo cliente, sendo os exemplos mais comuns chamadas, mensagens e recargas. Os eventos internos são operações realizadas dentro da plataforma ACM, podendo ser operações tais como notificações, atribuição de benefícios e transições de estados.

2.1.3.3.4 Ações

As ações são ativadas nas transações do ciclo de vida que permitem atribuir benefícios ao utilizador, enviar notificações, agendar benefícios, agendar notificações, cancelar agendamentos, submeter eventos numa campanha, etc.

Os tipos de ações que se podem encontrar são:

- **Benefícios –** Atribuído um benefício ao utilizador;
- **Notificações –** Envia uma notificação ao utilizador;
- **Atributos das campanhas –** Faz *update* aos atributos das campanhas;
- **Sinalizador de campanhas –** Despoleta de entre dois sinalizadores de campanha, mediante a avaliação de uma condição;
- **Validação –** Utiliza condições para despoletar sinalizadores de campanhas;
- **Cancelamento de agendamentos** – Cancela agendamentos marcados.

2.2 Processo de criação de aplicações de *data mining*

O Data Science, também conhecido como *data-driven science*, combina diferentes campos da estatística e computação que possibilita a interpretação da informação realizando decisões de mercado [3]. Uma das áreas que a constitui é o *Data Mining*.

O Data Mining, também conhecido como *knowledge discovery in databases,* é o processo usado por companhias para transformar informação em bruto em algo utilizável com valor. É normalmente utilizado *software* que procura padrões em grandes quantidades de dados [4].

O termo *Data Mining* nasceu nos anos noventa juntamente com base de dados multidimensionais e novos sistemas com maiores capacidades de armazenamento e processamento [5]. Com a disponibilidade de grandes volumes de dados as empresas têm vindo a criar processos que visam a exploração dos dados, identificando padrões e estabelecendo relações, para obterem vantagens sobre os seus concorrentes. Inicialmente foram formadas equipas para analisar os dados manualmente, mas devido ao volume elevado de dados e à sua uniformidade, tornou-se extremamente complicado de o fazer ou em alguns casos humanamente impossível. Com a evolução dos sistemas computacionais, o aparecimento das redes ubíquas, a criação de algoritmos e a possibilidade de estar ligado a várias bases de dados em simultâneo, proporcionou o aparecimento e crescimento de aplicações de *Data Mining*. Hoje em dia estas aplicações são encontradas em diferentes áreas, tais como: medicina, finanças, *marketing*, campanhas *online*, recomendações para *cross-selling*, etc. [5] [6].

Um bom princípio da *Data Science* é que *Data Mining* é um processo bem desenvolvido e documentado que deve ser percebido e executado. Os processos são usados para organizar e entender os problemas propostos, medir o seu progresso e chegar aos melhores resultados. Com o intuito de estandardizar o processo a CRISP-DM (*Cross Industry Standard Process*) publicou o processo que desenvolveu, identificando claramente as fases que o constituem e as suas interações. Atualmente é utilizado como referência no desenvolvimento deste tipo de aplicações [6].

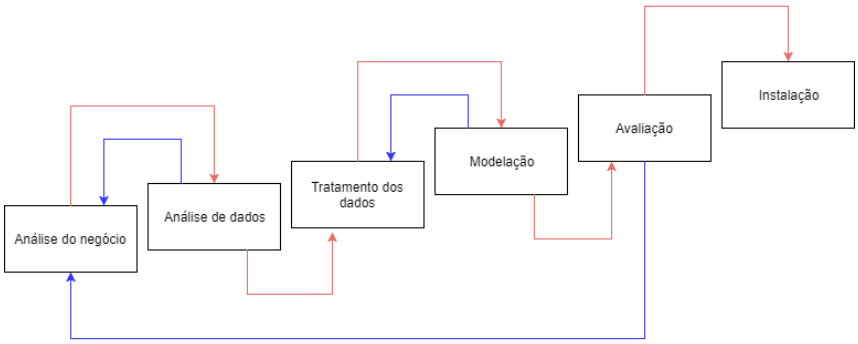

Figura 1 - Processo de *data mining* pertencente à CRISP

Como podemos observar na **Figura 1** é explícito que o processo apresentado privilegia a interação entre as fases, havendo a possibilidade de voltar a fases anteriores caso os resultados não sejam satisfatórios numa das fases intermédias. Assim sendo, com o conhecimento adquirido nas primeiras interceções é possível melhorar os resultados, aumentando o rigor e a granularidade na análise dos dados. Para cada fase existem objetivos bem definidos que podem ser consultados em seguida neste documento.

2.2.1 Análise do negócio

Nesta etapa é crucial entender qual o problema a ser resolvido. Este processo pode não ser trivial devido ao pouco conhecimento inicial dos dados e ambiguidades que possam surgir na contextualização do problema. É importante que exista um analista de sistemas experiente em várias áreas da computação, que consiga transformar um problema de negócio em um ou mais problemas de *Data Science*. Com isso é esperado que cada problema identificado se traduza em modelos que possibilitem fazer classificações, regressões, probabilidades estatísticas, etc.

A técnica que predomina nesta etapa é a criação de cenários. Esta técnica é utilizada quando não se tem a certeza quanto à solução a implementar. Para tal, são feitas assunções sobre o futuro, identificando o impacto da implementação da solução proposta [6].

2.2.2 Análise de dados

Enquanto resolver o problema de negócio é o objetivo do projeto, entender os dados e perceber quais os pontos fortes e fracos para o seu desenvolvimento. Geralmente os dados armazenados não são estruturados a pensar em futuras melhorias dos sistemas, apresentando um foco na solução previamente construída. Portanto, os dados estão estruturados de forma a realizar o objetivo do sistema implementado, dificultando a análise dos dados.

A avaliação sobre o custo de obtenção dos dados toma um papel importante. É crucial perceber se os dados são obtidos de uma forma direta ou se é necessário extrapola-los com a utilização de *software* específico. O entendimento da estrutura de armazenamento dos dados possibilita a associação da informação a problemas de *Data Mining* [6].

2.2.3 Tratamento dos dados

As técnicas de análises no tratamento dos dados têm algumas especificidades que devem ser respeitadas. Para que o resultado seja o melhor possível é preciso proceder às alterações da forma que os dados são representados. Um exemplo disso é criar uma tabela onde esteja a informação representada, removendo ou inferindo valores em falta e convertendo os dados em formatos diferentes. Este tipo de transformações é de extrema importância devido a algumas técnicas de *Data Mining* serem desenvolvidas para operarem com símbolos ou categorias, enquanto outras com valores numéricos. Quando se opera com valores numéricos é preciso ter em atenção que os valores devem ser normalizados ou escaláveis para possibilitar comparação entre eles. A transformação dos dados deve ser adequada à técnica de *Data Mining* que se pretende utilizar [6].

Uma das preocupações dos desenvolvedores destas tecnologias é o denominado "*Leaks*". Os *leaks* consistem na utilização de informação que ainda não está completa. Imaginemos que temos uma variável que nos informa quantas vezes nós entramos num *website* diariamente. A informação só estará completa quando o dia atual acabar, como tal utilizar esta informação durante esse dia seria enviesar o resultado real. A escolha das variáveis deve ser tomada cuidadosamente durante esta fase, porque os dados trabalhados são tipicamente dados de histórico, podendo apresentar indignamente os acontecimentos reais [6].

2.2.4 Modelação

A modelação é a fase onde as técnicas de *Data Mining* e algoritmos de *Machine Learning* são aplicadas aos dados. É importante ter conhecimento sobre as ideias fundamentais sobre

esses temas, que incluem um conjunto de técnicas e de algoritmos. No conjunto de fases descritas esta é a que contém mais componentes de *Data Science*.

As tecnologias e algoritmos característicos nesta fase são descritos no capítulo de Algoritmos de *Machine Learning*. Assim, este assunto não será aqui abordado ao detalhe, sendo necessário consultar o capítulo referido [6].

2.2.5 Avaliação

Nesta fase é realizada uma avaliação dos resultados obtidos do modelo de *Data Mining*, com o objetivo de verificar a confiabilidade dos dados, antes de se passar à fase seguinte. A avaliação do modelo de *Data Mining* é feita a partir de um laboratório controlado, pois torna-se mais económica, fácil, rápido e seguro de o fazer. Contudo, nem sempre um modelo que obtém sucesso no laboratório pode ser implementado no problema real, devido às especificações dos objetivos do projeto.

Os *stakeholders* são importantes para avaliar os dados na perspetiva de negócio e *markting*. Este pensamento fora da caixa possibilita antecipar futuras catástrofes. No entanto nem sempre os modelos são de fácil compreensão, o que dificulta a análise dos *stakeholders*. Torna-se imprescindível que a complexidade do modelo seja bem gerida.

Depois desta fase é esperado que o modelo de *Data Mining* apresente os requisitos necessários para ser implementado no problema real [6].

2.2.6 Instalação

Com o intuito de satisfazer o objetivo do projeto o modelo é implementado no problema real, esperando-se o retorno do investimento. Um exemplo óbvio de um contexto real é a implementação do modelo num sistema de informação ou num processo de negócio. Quando se coloca o modelo em produção tipicamente é necessário efetuar restruturações, para aumentar a velocidade e a compatibilidade com o sistema já implementado. Como é despendido bastante tempo e orçamento nesta fase geralmente é delegada esta tarefa para a fase de avaliação. Assim, é realizado um paralelismo das duas tarefas para que o tempo seja reduzido.

Esta fase deve ser a consolidação das anteriores esperando-se um produto robusto que apresente as funcionalidades propostas e com uma fiabilidade elevada. Contudo, apesar de não estar representado na **Figura 1** é possível voltar a fases anteriores do projeto. Assim caso haja um fracasso na implementação do produto no contexto real se possa voltar e refazer o produto [6].

2.3 Análise exploratória dos dados

Com o intuito de entender os dados envolvidos num projeto é necessário elaborar hipóteses que validem a relação entre variáveis. Este método é bastante utilizado na literatura de análises estatísticas, utilizando técnicas como análises de séries temporais, testes de controlos de qualidade e testes não paramétricos. Contudo, estes testes podem não ser eficazes quando são confrontados com conjuntos de dados com volumes elevados. Isso torna a tarefa de verificar as relações entre os dados bastante complicados. Para colmatar este problema é utilizado *exploratory data analysis* (EDA). Permite estudar as variáveis que estão relacionadas entre si, criar histogramas com variáveis numéricas e examinar a distribuição das variáveis. Isso faz com que o modelo se torne estável [7].

2.4. Análise e estrutura dos dados

Hoje em dia existem várias fontes onde se pode retirar informação, tornando a informação ambígua. Pode-se obter informação de transferências bancárias, questionários, estatísticas, inscrições em plataformas, pesquisas na internet, etc. Com a recente evolução na tecnologia é possível armazenar uma grande quantidade de informação a um custo baixo. Contudo, é necessário entende-la e estruturá-la de forma a proporcionar uma mais-valia às empresas interessadas. Este processo pode ser difícil, pois a maioria da informação nunca foi revista ou analisada. Assim, é necessário entender alguns conceitos básicos que foram desenvolvidos para preparar e analisar a informação [7].

2.4.1 Forma standard

A informação deve ser reorganizada no formato estandardizado, possibilitando trabalhar a informação com diferentes algoritmos de *Data Mining*.

Na forma estandardizada um conjunto de valores correspondem a uma interação com o objeto, dando-se o nome de registro ou instância. À informação total que é armazenada é chamada de conjunto de dados. Geralmente os conjuntos de dados são representados por tabelas de uma base de dados sendo as linhas denominadas instâncias. As colunas da tabela são comumente chamadas de atributos. Na **Figura 2** verifica-se um exemplo de uma tabela na forma *standard* [7].

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	Second
A	B	B	B	B	Second
B	A	A	B	A	Second
A	A	A	A	B	First
A	A	B	B	A	First
B	A	A	B	B	Second
.....	.....	.....	.....	.....	.....
A	A	B	A	B	First

Figura 2 - Tabela na forma *standard* [7].

2.4.2 Informação labelled e unlabelled

O tratamento da informação pode ser realizado de duas formas diferentes, dependendo se é *labelled* ou *unlabelled* [7].

- ***Labelled/Supervised Learning***: Informação com os atributos bem definidos com o objetivo de prever um atributo em falta, numa determinada instância utilizando os dados anteriormente analisados. Os valores utilizados podem ser por categorias (ex: Bom, Médio, Mau) ou numéricos [7].
- ***Unlabelled/ Unsupervised Learning***: Informação que não é associada a nenhum atributo, tendo como objetivo única e exclusivamente de obter alguma informação útil [7].

2.4.3 Tipos de variáveis

Para medir as propriedades dos objetos existem vários tipos de variáveis. É importante analisar e entender qual o propósito de cada tipo para facilitar a compreensão dos dados e proceder a uma transformação vantajosa para o projeto. Em seguida serão apresentados os tipos de dados existentes e uma breve descrição de cada um.

- **Variáveis nominais**: São utilizadas para categorizar os atributos. Geralmente são utilizados nomes, mas também podem ser utilizados números. Contudo, os números são representativos não tendo nenhuma interpretação matemática;
- **Variável binária**: É uma variável nominal que só utiliza dois valores válidos (ex: 0 ou 1);
- **Variável ordinal**: É parecido com as variáveis nominais, mas os valores podem ser colocados numa ordem significativa. (ex: bom, médio, mau);
- **Variável inteira**: É usada para fazer cálculos aritméticos com o intuito de obter informação significativa;
- **Variável escala em intervalo**: É uma variável numérica que tem valores numéricos que são medidos a partir de um ponto zero ou ponto de origem. O exemplo mais conhecido é a escalada fahrenheit;
- **Variável de rácio**: É uma relação entre duas grandezas que podem ser expressas sobre a forma de quociente ou percentagem.

2.5 Preparação dos dados

Mesmo depois de organizar a informação e utilizar os tipos de dados corretamente os erros podem aparecer em vários sítios de difícil deteção. Para tentar colmatar esses erros é utilizado um conjunto de processos que tendem a diminui-los, e em alguns casos prevenilos. Os erros podem ser tão complexos como perceber se os resultados dados são fidedignos ou simplesmente um erro do sistema. Sendo assim, são apresentados alguns processos que se devem ter em conta na preparação dos dados [7].

2.5.1 Valores em falta

Por vezes as instâncias que são trabalhadas não têm valores em todos os atributos que o constituem. Isso não significa que exista um problema com a recolha dos dados, pois dependendo da situação os valores podem ou não existir. Isso faz com que se crie uma instabilidade no conjunto de dados, podendo traduzir-se em erros futuros na utilização do modelo. Para resolver este problema são apresentadas duas abordagens:

- **Descartar instâncias**: Este método é mais conservador, pois tenta reduzir ao máximo a introdução de valores errados no conjunto de dados guardados, removendo as instâncias que não apresentam valores em todos os atributos. Porém, este método só pode ser utilizado caso o número de instâncias sem valores seja residual em comparação com o conjunto de dados [7].
- **Substituir pela média das restantes**: Para que o impacto seja quase nulo na utilização dos dados, é atribuído às instâncias sem valores o valor da média das instâncias restantes. Em consequência, a influência desse valor no modelo que se pretende utilizar é mínima [7].

2.5.2 Redução do número de atributos

Com a diminuição dos preços para armazenar dados, as empresas guardam cada vez mais informação. Isso resulta num elevado número de atributos associados a cada instância, adicionada ao conjunto de dados. Para este propósito geralmente é utilizado o termo de *dimension reduction* para resolver esse problema.

O *dimension reduction* pode ser visto como o processo de redução de características que são apresentadas numa determinada instância, e visa baixar a complexidade do conjunto de dados para aumentar a precisão do modelo e diminuir o custo de medição. [7].

2.5.3 Transformação da informação

As variáveis tendem a ter valores muito discrepantes. Para combater isso o desenvolvedor da aplicação deve identificar e normalizar esses casos. Isso permite que o impacto das discrepâncias seja minimizado e o erro apresentado seja significativamente reduzido. Existem várias técnicas para resolver este problema, contudo vamos analisar as duas mais utilizadas. Vamos assumir que X é o valor recolhido e X* é o valor normalizado [8].

- **Normalização de Máximos-Mínimos**: É o processo de identificar a diferença que existe entre o valor mínimo e valor máximo reduzindo a sua variação e colocando-a numa escala. Na escala estarão representados valores próximos que não irão interferir com os cálculos efetuados pelo algoritmo [8].

$$X^* = \frac{\mathrm{X} - \min(\mathrm{X})}{\mathrm{range}(\mathrm{X})} = \frac{\mathrm{X} - \min(\mathrm{X})}{\max(\mathrm{X}) - \min(X)}$$

- ***Z-Score Standardization***: É uma técnica muito utilizada no mundo das análises estatísticas, e consiste em obter a diferença entre o valor do campo da instância em causa e a média das restantes instâncias. Em seguida é escalada a diferença entre o desvio padrão dos atributos [8].

$$X^* = \frac{\mathrm{X} - \mathrm{mean}(\mathrm{X})}{\mathrm{SD}(\mathrm{X})}$$

2.5.4 Método numérico de identificação de outliers

Uma forma de encontrar *outliers* é utilizando o *z-score standardization*. Geralmente é possível encontrar os *outliers* percebendo se o desvio padrão é maior que 3 e o *z-score standardization* apresenta valores acima de 3 e abaixo de -3. Caso estes valores sejam obtidos deve ser identificado o motivo e perceber se os resultados são provenientes de um erro presente no algoritmo ou simplesmente faz parte do range de valores [8].

2.6 Algoritmos de *Machine Learning*

O termo *Machine Learning* é por definição, dentro do âmbito da inteligência artificial, disciplina preocupada com o desenvolvimento de *software* que consiga aprender autonomamente [9]. A inteligência artificial é a capacidade de um computador ou de um robô controlado por computador realizar tarefas comumente associadas a seres inteligentes [10]. Os algoritmos de *Machine Learning* estão contidos dentro do processo de *Data Mining* que corresponde à fase de modelação descrita anteriormente neste documento. Na criação destes algoritmos foram recolhidos conhecimentos de diferentes áreas, tais como física, matemática, biologia, ciência neuronal e estatística. Com o cruzamento deste aglomerado de informação surgiram inúmeros algoritmos nos últimos anos, com abordagens e técnicas diferentes. Sendo assim, é necessário, antes de cada projeto, fazer uma análise de qual a técnica mais indicada para o problema que se pretendem resolver.

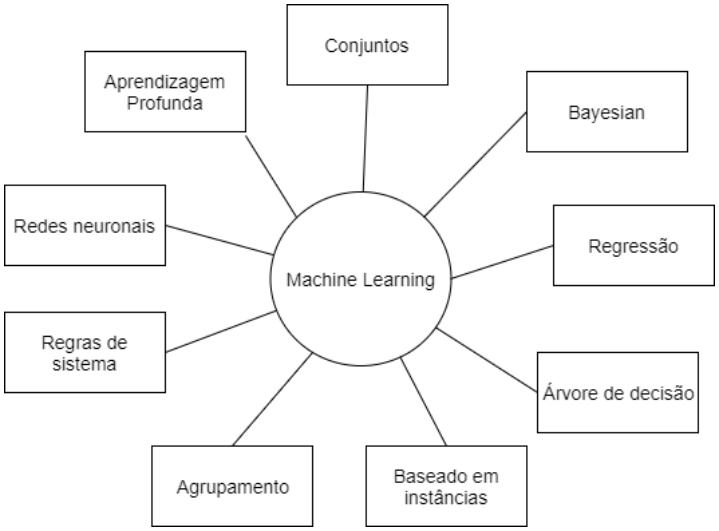

Figura 3 - Algoritmos de *Machine Learning*

Escolher o melhor algoritmo de *Machine Learning* é uma tarefa de análise. É possível utilizar diferentes algoritmos para resolver um problema, apesar dos resultados dos algoritmos serem diferentes. Portanto, é uma questão de perceber qual é que se adapta melhor ao problema apresentado e maximizar o seu rendimento. Como podemos verificar na **Figura 3**, é possível colocar os algoritmos nas seguintes categorias:

- **Algoritmos de aprendizagem profunda**: São algoritmos que foram criados usando como inspiração o funcionamento do cérebro humano. Os algoritmos tentam modelar abstrações de alto nível de dados usando um grafo profundo com várias camadas de processamento, compostas de várias transformações lineares e não lineares;
- **Algoritmos de conjunto**: São utilizados múltiplos algoritmos com o objetivo de resolver problemas complexos. Assim, são utilizados dois ou mais algoritmos diferentes, avaliando os resultados obtidos utilizando um método estatístico para realizar a decisão;
- **Algoritmos de redes neuronais**: Os algoritmos deste tipo foram, à semelhança do algoritmo de aprendizagem profunda, desenvolvidos com inspiração do cérebro humano, contudo com um grau de profundidade menor comparado aos de aprendizagem profunda;

- **Algoritmos de regras de sistema**: Os algoritmos baseados neste sistema resolvem problemas para os quais são conhecidas as ações a tomar nos diferentes cenários que se podem encontrar. As regras necessárias para o funcionamento do algoritmo são aplicadas consoante a situação;
- **Algoritmos de regressão**: São utilizadas fórmulas estatísticas para realizar análises sobre os dados existentes, obtendo um intervalo de valores que podem ser analisados e utilizados para realizar previsões;
- **Algoritmos *Bayesian***: Estatística *Bayesian* é um ramo da estatística que quantifica o interesse, tratando os dados como variáveis aleatórias, extraindo conclusões da distribuição dos dados;
- **Algoritmos de árvore de decisão**: É um método de *supervised learning* e não paramétrica, utilizado para realizar classificações e regressões. O objetivo é criar um modelo que preveja o valor de uma variável, aprendendo regras de decisão simples inferidas através dos conjuntos de dados. Os dados de decisão são decompostos em vários problemas mais pequenos, sendo esta estratégia utilizada para cada decomposição efetuada;
- **Algoritmos baseados em instâncias**: Utiliza dois conjuntos de dados, o conjunto de dados de treino e o conjunto de dados de teste. Utilizando estes dois grupos, os algoritmos utilizam os dados de treino para memorizar padrões que possam existir, utilizando-os para prever quais os resultados de um determinado campo (atributo) no conjunto de dados de teste;
- **Algoritmos de agrupamento**: Os algoritmos de agrupamento fazem uma divisão dos dados em grupos com objetos similares. Os grupos são compostos por objetos que são similares aos elementos que os constituem e diferentes comparativamente aos restantes grupos.

Cada uma das categorias contém diferentes algoritmos. Isso faz com que exista um número enorme de algoritmos que podem ser estudados. Com a restrição temporal deste projeto, foram estudados alguns algoritmos de diferentes categorias para se obter uma visão geral do funcionamento de cada um deles. Portanto, é possível escolher o algoritmo que vai ao encontro das necessidades do projeto.

2.6.1 *K-Nearest Neighbor*

O *K-Nearest Neighbor* é um algoritmo utilizado para fazer estimações estatísticas e reconhecimento de padrões. Este permite fazer classificações e regressões, sendo considerado *instance-based learning*. O *instance-based learning* ou em alguma literatura *lazy learning,* consiste em ter um conjunto de dados de treino já classificado, guardados em forma de vetor num espaço com n-dimensões que é utilizado para comparar dados que entram no algoritmo com o intuito de obter um ou mais atributo em falta. Os atributos são classificados quando é encontrado um ou mais registos que estejam próximos da instância considerada no espaço dimensional. A origem da palava *lazy* vem do facto do algoritmo demorar mais a classificar do que a carregar os dados de treino. Os dados são divididos por classes, tendo como característica a proximidade considerada pelo KNN [14].

Para procurar elementos próximos que não pertencem ao conjunto de treino, o KNN procura K elementos do conjunto de treino que estejam mais próximos do elemento desconhecido. O K indica o número de vizinhos que serão utilizados na execução do algoritmo. O parâmetro K faz com que algoritmo consiga uma classificação mais refinada, porém o valor ótimo de K varia de um problema para o outro, o que faz com que, para cada conjunto de dados, sejam testados vários valores diferentes de forma a descobrir qual o melhor valor de K para determinado problema. [8].

O K influência de formas diferentes no caso de o algoritmo ser de regressão ou classificação. Quando o K> 1 na classificação são selecionados os vizinhos mais frequentes. Quando é na regressão é calculada a média dos valores dos K vizinhos.

Como é possível verificar na **Figura 4** quanto maior for o K, mais abrangente é a sua ação.

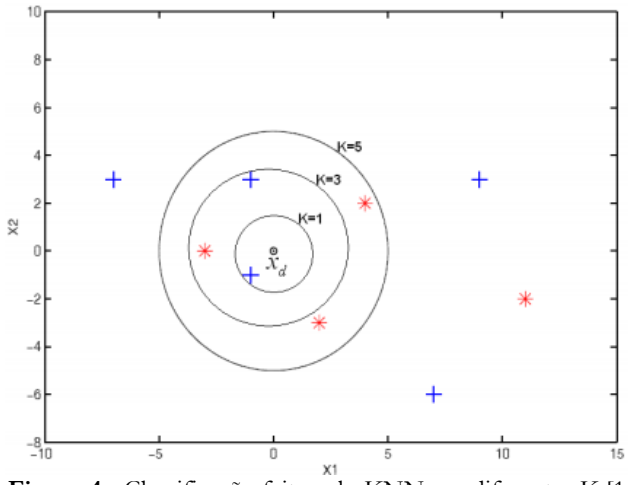

Figura 4 - Classificação feita pelo KNN par diferentes K [14]

O algoritmo funciona satisfatoriamente quando se trata de um número reduzido de variáveis de entrada, mas quando o número é grande começa a existir uma resistência. O número de dimensões que são utilizadas no algoritmo depende do número de variáveis de entrada. Caso sejam duas variáveis de entrada o funcionamento do algoritmo seria bidimensional. À medida que o número de variáveis de entrada aumenta, o número de dimensões aumenta a uma taxa exponencial. Em dimensões elevadas as instâncias que são semelhantes podem estar afastadas, o que quebra a lógica do algoritmo. Este fenómeno é chamado de *"Curse of Dimensionality"* [14].

Para calcular as distâncias entre os pontos no espaço são utilizadas métricas. Essas métricas são escolhidas consoante o objetivo do algoritmo e dos resultados que se consegue obter

com cada uma delas. Para tal, são apresentadas as métricas mais conhecidas e utilizadas na implementação deste algoritmo.

Seja x = (x_1,x_2,…x_n) e y = (y_1,y_2,…y_n) dois pontos do R

A distância *Euclidiana* entre X e Y é dada por:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

A distância *Manhattan* entre X e Y é dada por:

$$d(x,y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

A distância *Minkowski* entre X e Y é dada por:

$$d(x,y) = (|x_1 - y_1|^q + |x_2 - y_2|^q + \cdots + |x_n - y_n|^q)$$

A distância *Euclidiana* é utilizada geralmente quando as variáveis são semelhantes em género. Quando tipo de atributos é muito discrepante a tendência é de utilizar a distância *Minkowski*. A última distância é a generalização das duas distâncias anteriores. Quando q = 1, esta distância representa a distância de *Manhattan* e quando q = 2, a distância *Euclidiana*. Pode também atribuir pesos a cada uma das fórmulas. Fazendo com que se possa dar relevância a determinados pontos [14].

2.6.1.1 Conjunto de dados de treino e de teste

Como foi referido anteriormente é necessário criar um conjunto de dados de treino para que quando haja um pedido de classificação ao algoritmo ele possa utilizar esse conjunto para comparar e calcular a distância entre os novos dados e os de treino. É chamado de conjunto de teste os valores que são enviados ao algoritmo para serem classificados.

O sucesso do algoritmo depende em parte da seleção das instâncias que constituem o conjunto de treino. Portanto, deve-se fazer a normalização dos dados e uma seleção pormenorizada das instâncias que a constituem, com o intuito de minimizar os erros de classificação que possam ocorrer.

2.6.1.2 Função de peso

Quando os vizinhos mais próximos são encontrados, é possível avaliá-los utilizando diferentes métodos de ponderação. Para cada resultado identificado é adicionado um peso ao total dessa classe. No final é escolhida a classe que possua um peso maior perante as outras. Esta função é criada para evitar que as instâncias que se encontram mais afastadas da instância classificada tenham menos peso na votação da classe.

Serão apresentadas em baixo as funções de peso mais utilizadas:

- ***None***: Todos os vizinhos têm peso igual [14];
- ***Fraction***: Digamos que i indica a ordem dos vizinhos mais próximos, $i = 1 \ldots k$. A função de pesos é $1/i$. Portanto, o peso dos vizinhos é inversamente proporcional às posições apresentadas na lista. A fração do peso diminui abruptamente em ordem a i [14];
- ***Stairs***: Digamos que i indica a ordem dos vizinhos mais próximos, $i = 1 \ldots k$. A função de peso é $(k - i + 1)$. Mais uma vez o peso dos vizinhos é inversamente

proporcional às posições apresentadas na lista. A fração do peso diminui levemente em ordem a i [14];

- ***InverseDistance***: Afirmemos que d é a distância do vizinho da instância testada. A função de peso é $1/d$. O peso do vizinho é inversamente proporcional à distância da instância testada [14];
- ***InverseSquareDistance***: Afirmemos que d é a distância do vizinho da instância testada. A função de peso é $1/d^2$. O peso do vizinho é inversamente proporcional à distância da instância testada [14].

2.6.2 C4.5

O algoritmo C4.5 é um algoritmo de árvore de decisão que é utilizado para realizar classificações. É baseado numa árvore de decisão, construída utilizando o conceito de entropia. O conceito de entropia, de acordo com o princípio de máxima e mínima informação, possibilita encontrar a distribuição de probabilidade que mais se adequa aos dados, na qual é minimizado o uso inadvertido de qualquer tipo de informação que não há explicitamente disponível, podendo ser encarado como um ramo da inferência estatística [40]. Depois da construção da árvore é possível começar a pesquisa da raiz seguindo cada teste até uma folha ser alcançada. Essa estrutura recursiva pode ser definida por:

- Um nó folha que corresponde a uma classe ou atributo;
- Um nó de decisão que contém uma operação de resultado binário.

Este algoritmo é considerado "guloso" pois não se preocupa com o resultado final, somente se preocupa com a decisão local referente ao nó em questão. A sua construção é feita pelo ponto de partida chamado raiz e cria múltiplas subárvores até chegar ao nó folha. Até chegar à folha são criados nós de decisão que são frações do problema proposto inicialmente. Na **Figura 5** podemos verificar um exemplo de uma árvore criada pelo algoritmo.

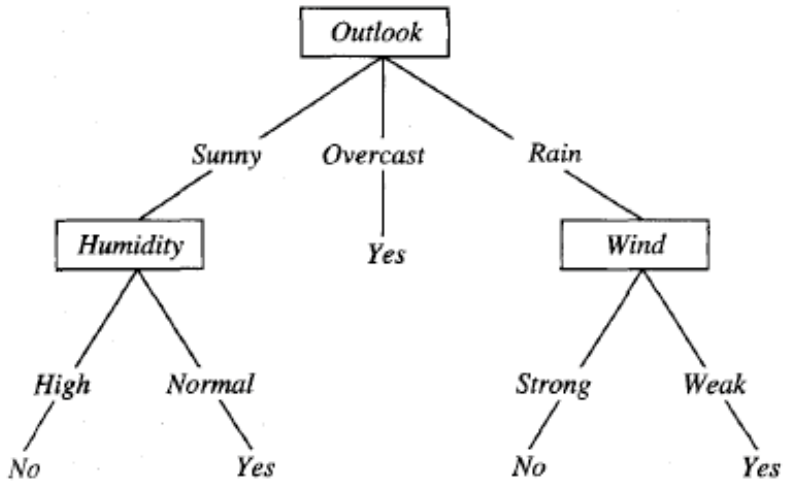

Figura 5 - Exemplo da execução do algoritmo C4.5 [30]

2.6.2.1 Selecionar melhores atributos de particionamento

Os resultados que este algoritmo produz são influenciados pela forma como são escolhidos os atributos que vão ser particionados. Para identificar quais os atributos mais adequados para a divisão dos dados são utilizadas algumas destas técnicas de entropia:

- **Ganho máximo**: É escolhido o atributo que se obtenha o menor tamanho na distribuição da árvore [31];
- **Índice Gini**: O índice Gini, desenvolvido por Conrado Gini em 1912, mede o grau de heterogeneidade dos dados. Portanto pode ser utilizado para medir a impureza dos dados [32];
- **Razões de ganho**: Para solucionar o problema do ganho de informação, foi proposto em 1993 a Razão de Ganho (do inglês *Gain Ratio*), que nada mais é do que o ganho de informação relativo (ponderado) como critério de avaliação [33].

2.6.2.2 Poda

A técnica da poda é utilizada para diminuir a árvore de decisão quando os ramos se tornam demasiado específicos para o problema tratado, originando o *overfitting*. O *overfitting* ocorre quando a árvore cria bastantes ramos com um número enorme de pontos de decisão, diminuindo a probabilidade de conseguir chegar a uma conclusão correta. Portanto, esta técnica pode ser utilizada em duas alturas no ciclo de funcionamento do algoritmo. Na fase de criação da árvore de decisão, denominando-se por poda descendente ou quando a árvore já se encontra no seu estado final, denominando-se por poda ascendente [32].

- **Poda descendente**: À medida que a árvore está a ser construída o algoritmo avalia a confiabilidade da divisão. Caso seja confiável é acrescentado à árvore. Este processo pode ser menos eficaz que o ascendente, devido ao facto de poder interromper o crescimento da árvore e o nível de detalhe do ponto atual não ser o adequado;
- **Poda ascendente**: No final da árvore estar completa é feita uma avaliação e exclusão de ramos com o uso dos métodos de evoluções consideráveis. Contudo, este método pode ser ineficaz, pois uma árvore grande pode ser reduzida a uma árvore mínima.

Para facilitar o entendimento do conceito é mostrado na **Figura 6** e **Figura 7** um exemplo do algoritmo utilizando dados fictícios, e em seguida uma descrição do processo efetuado.

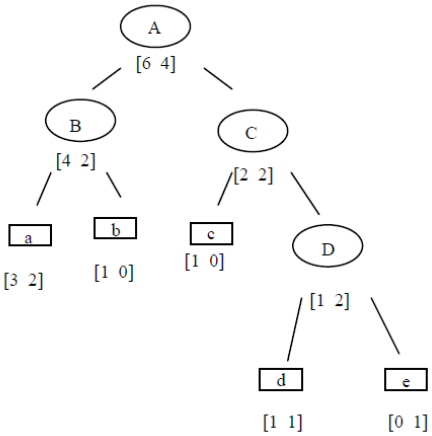

Figura 7 - Árvore sem poda [32]

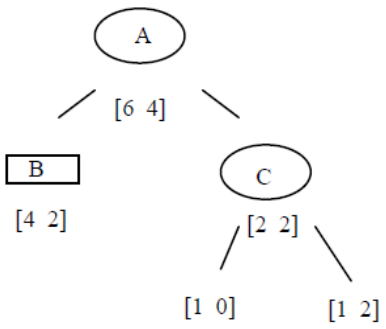

Figura 6 - Árvore com poda [32]

O processo para efetuar esta redução pode ser realizado da seguinte forma [32]:

1. Percorrer a árvore em profundidade;
2. Para cada nó de decisão calcular:
   - Erro no nó;
   - Soma dos erros nos nós descendentes;
3. Se o erro do nó for menor ou igual à soma dos erros dos nós descendentes então o nó é transformado em folha.

2.6.3 Naive Bayes

O *Naive Bayes* é um classificador baseado em *Bayes theorem* sendo este uma condição probabilística. Este classificador tem características muito simplistas, assumindo que os atributos que são usados para realizar as classificações têm a mesma importância. Assim o objetivo é calcular a probabilidade de cada amostra desconhecida e associá-la à classe mais provável. Neste processo são criados o conjunto de dados de treino e o conjunto de teste. O conjunto de treino são os dados que vão servir de base para conseguir aferir uma probabilidade aos dados que estão no conjunto de teste. A probabilidade é obtida pela utilização da seguinte fórmula:

$$P(c1) = \frac{número\ de\ exemplos\ de\ c1}{número\ de\ exemplos}$$

Como podemos verificar na fórmula, o cálculo é realizado dividindo-se o número de exemplos de determinada classe pelo número total de exemplos do conjunto de treino [44].

Depois de calcular essa probabilidade é calculada a probabilidade de cada atributo do conjunto de teste para cada classe existente. A forma de o fazer é utilizando a fórmula referente ao *Bayes theorem*. A fórmula é representada da seguinte maneira:

$$p(\frac{c_j}{d}) = \frac{\mathrm{p}\left(\frac{\mathrm{d}}{c_j}\right)\mathrm{p}(c_j)}{p(d)}$$

De modo a explicar superficialmente a fórmula apresentada, em seguida será subdividida em várias partes e explicado o propósito de cada uma.

- $p(\frac{c_j}{d})$ = Probabilidade da instância d estar dentro do conjunto c_j. É a representação matemática do que estamos a tentar classificar;
- $\mathrm{p}\left(\frac{\mathrm{d}}{c_j}\right)$ = Probabilidade de gerar a instância d dada uma classe c_j. Perceber qual a semelhança entre os dados;
- $\mathrm{p}(c_j)$ = Probabilidade de ocorrer a classe c_j. Representa a frequência em que c_j se encontra na base de dados;
- $p(d)$ = Probabilidade da instância d ocorrer. Esta probabilidade pode ser ignorada se for a mesma para todas as classes.

Este algoritmo é bastante rápido a realizar classificações comparativamente aos restantes, pois somente realiza uma interação com os dados de treino. A rápida classificação também ocorre pelo facto do algoritmo não ser sensível a atributos irrelevantes e trabalhar com vários tipos de dados de forma natural e pacífica. Contudo, demostra uma fragilidade na estrutura dos dados, pois não permite trabalhar com atributos relacionados, descartando informação relevante acerca dos dados trabalhados. [34].

2.6.4 K-means

O *k-means* é *unsupervised learning*, o que significa que não existe nenhum conjunto de treino previamente identificado e classificado para servir como termo de comparação. Para resolver esse problema o programa agrupa em conjuntos os dados utilizados de forma a que quando um novo registo entra em funcionamento seja inserido num dos grupos existentes. Isso faz com que seja possível inferir um grupo a cada dado inserido.

A primeira etapa do algoritmo é definir k centros, para servir de referência na criação dos k grupos utilizados, sendo atribuído um por cada grupo. Para melhorar o funcionamento do algoritmo, deve-se manter a máxima distância entre os centros de forma a delimitar bem as fronteiras de cada grupo. Para finalizar esta etapa inicial é necessário juntar os pontos associados ao conjunto de teste, adicionando-os aos vizinhos mais próximos. Consoante o número inserido de cada grupo aumenta, os centros movem-se no gráfico adequando-se o melhor possível aos dados. Quando todos os pontos pertencerem a um grupo o processo chega ao fim e os centros não se movem mais [35].

Para calcular o local exato onde se encontram os pontos tendo como referência os centros, é utilizada a seguinte função objetiva:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{x} \| x_i^{(j)} - c_j \|^2$$

Esta função representa a distância medida entre os pontos inseridos $x_i^{(j)}$ e o centro c_j . Sendo realizado para todos os n pontos existentes no plano.

Este algoritmo é fácil de implementar e permite que sejam automaticamente atribuídos dados aos grupos sem ser necessário realizar trabalho extra. As localizações dos centros de referência adaptam-se consoante os dados inseridos. Contudo, mesmo que os dados inseridos não tenham características para nenhum dos grupos que foram criados, o algoritmo força que este seja associado a um determinado grupo. Isso pode tornar o resultado enviesado na realização da classificação [36].

2.6.5 Random Forest

Random Forest é um algoritmo *supervised learning* que utiliza o método de *bagginng* para realizar classificações e regressões. O *bagging* consiste na distribuição da informação de forma aleatória em diferentes árvores de decisão que são executadas de forma independente. Depois do *bagging* é efetuada uma votação por maioria, sendo escolhido o resultado que aparece mais vezes entre as árvores de decisão [38].

Este algoritmo herda muitas das características das árvores de decisão, pois apresenta a mesma característica de dividir-para-conquistar o que se pode traduzir em algumas características que fazem destacar-se das técnicas restantes, sendo algumas delas [39]:

- Algoritmo mais poderoso do que comparado somente a uma árvore de decisão;
- Boa taxa de acerto em diferentes conjuntos de dados;
- Técnica exata;
- Evita o *overfitting*;
- Menos sensível a ruídos;
- Classificação aleatória sem intervenção humana.

Em seguida será mostrada na **Figura 8** com um exemplo da típica utilização do *Random Forest* para se ter a perceção visual de todos os conceitos aqui descritos.

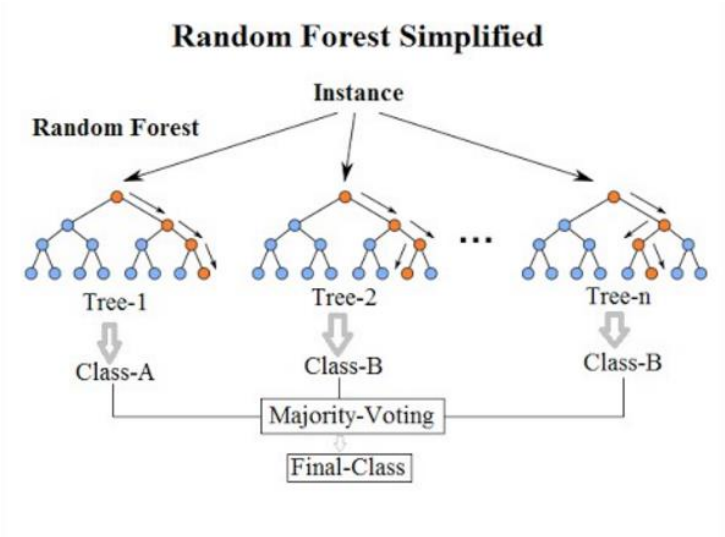

Figura 8 - Exemplo do funcionamento do algoritmo *Random Forest* [40]

2.6.6 Perceptron

O *Perceptron* é um algoritmo de *supervised learning* que suporta um reforço de informação no meio do seu funcionamento. É um algoritmo de classificação linear que faz uma separação dos pontos no espaço [45]. As principais funções apresentadas são:

- Classificação da informação com recurso a padrões;
- Deteção de anomalias e irregularidades no funcionamento do sistema;
- Processamento de sinais utilizando compressões, filtragens e separação;
- Aproximação a uma função objetiva.

2.6.6.1 Topologia das redes neuronais

As topologias de redes neuronais são compostas por três camadas. Essas camadas podem tornar o algoritmo mais ou menos complexo, dependendo do número de trocas de informações existentes [45]. As camadas existentes são:

- **Camada de entrada**: Que permite o recebimento de dados/sinais/amostras para análise, bem como atribuição dos pesos;
- **Camada intermédia ou escondida**: Permite fazer a filtragem, tratamento e extração da informação recebida;
- **Camada de saída**: Analisa a informação tratada anteriormente e fornece uma resposta ao problema tratado.

Na **Figura 9** podemos verificar as camadas que foram identificadas.

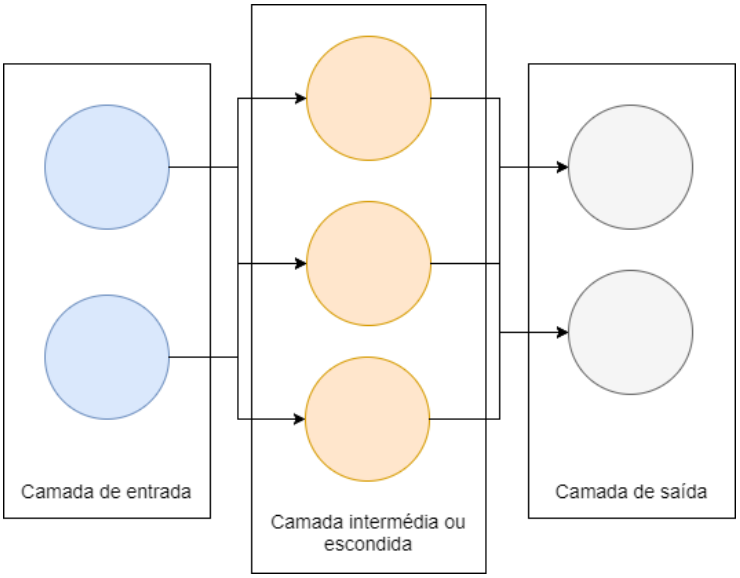

Figura 9 - Topologia de redes neuronais

As camadas apresentadas podem ser utilizadas de formas diferentes, consoante o contexto do problema [45]. Assim, existem algumas modificações conhecidas que podem ser usadas neste algoritmo, tais como:

- **Feed Forward de camada simples**: É uma camada de entrada associada a vários neurónios, que por sua vez produzem o resultado final. Podemos aferir que a camada escondida não é utilizada nesta arquitetura;
- **Feed Forward de camadas múltiplas**: Construída tipicamente pelas camadas de entrada, escondidas e de saída. Esta arquitetura pode apresentar várias camadas escondidas, aumentando a sua complexidade. Pode ser utilizada em várias áreas da informática, tais como: problemas de aproximação de funções, classificação de padrões, identificação de sistemas, otimização, robótica e controle de processos;
- **Recorrente ou alimentada**: É criado um ciclo onde os resultados produzidos na camada de saída podem ser novamente enviados para os neurónios pertencentes à camada escondida. Isso faz com que sejam refinados os resultados finais, melhorando a classificação, recorrendo à mudança de parâmetros. Esta arquitetura pode ser utilizada nas seguintes funcionalidades: previsões de séries temporais, otimização, identificação de sistemas e controle de processos;
- **Estrutura reticulada**: Os neurónios são colocados distribuídos numa dimensão com vista a ajustar os pesos e seus limiares. São utilizados para as seguintes funções: agrupamento, reconhecimento de padrões, otimização de sistemas, etc.

Com o intuito de melhorar a compreensão do algoritmo é apresentado na **Figura 10** um diagrama demostrativo do seu processo simplificado. A complexidade pode depender da arquitetura. Portanto, é demonstrado o processo mais simples dentro do neurónio.

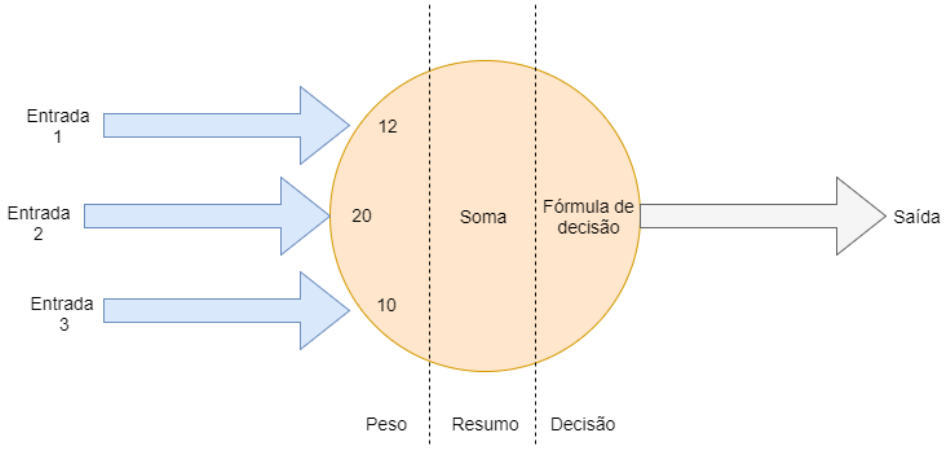

Figura 10 - Ciclo de um neurónio

Nas entradas dos dados são multiplicados os valores recebidos pelos pesos atribuídos individualmente a cada uma das entradas. Esses pesos podem ser ajustados na fase de aprendizagem do algoritmo baseando-se na experiência e histórico [45]. Os valores obtidos são somados num único valor, podendo ser melhorado com base em resultados anteriores.

Esta característica é o que faz com que o algoritmo consiga aprender e melhorar gradualmente os resultados de saída. Antes de sair o resultado é utilizada uma função ativa, que permite realizar a classificação dos registos.

2.6.6.2 Função ativa Perceptron

A forma mais básica de uma função ativa é uma simples função binária que tem somente dois possíveis resultados:

$$f(x) = \begin{cases} 1 \text{ if } \mathrm{w}.\mathrm{x} + \mathrm{b} > 0 \\ 0 \text{ Restantes} \end{cases}$$

Utilizando esse método é possível obter uma classificação com dois valores 1 e 0. Faz a avaliação com a utilização dos *arrays* x e w que correspondem respetivamente aos valores de entrada e de peso. O b é denominado por inclinador que é gerido pelo algoritmo. Este algoritmo utiliza funções lógicas para calcular e classificar os dados inseridos. Contudo, apresenta limitações no tipo de funções utilizadas. Somente consegue utilizar as funções de OR e AND, sendo impossível para o algoritmo utilizar a função XOR [45]. Como o algoritmo em cada neurónio só consegue separar o espaço bidimensional em duas metades, calcular com base no XOR tornasse impossível. Com o intuito de demonstrar uma função AND com o *Perceptron*, é realizado um pequeno exemplo na linha seguinte:

$$(0,0) = 0$$

$$(0,1) = 0$$

$$(1,0) = 0$$

$$(1,1) = 1$$

Os valores da classificação podem ser diferentes consoantes as funções ativas apresentadas. As funções mais utilizadas para este algoritmo são:

- **Step function**: Nesta função os valores podem ser 0 e 1;
- **Sign function**: Nesta função os valores podem ser 1 e -1;
- **Sigmoid function**: Nesta função os valores podem ser 0 e 1.

2.6.7 One rule

One rule é um algoritmo de *supervised learning* com o funcionamento semelhante aos algoritmos das árvores de decisão, somente com um nível. A avaliação de registos novos é realizada com base no rácio de erro que cada registo apresenta, em comparação com as regras criadas pelo algoritmo. A classificação é feita escolhendo sempre a regra calculada que apresenta o rácio de erro mais pequeno. Só é finalizada a avaliação das regras quando o algoritmo calcular o rácio de todas as regras [41]. Verifica-se na **Tabela 1** o relacionamento dos dados com classificação exata. Assim, caso o conjunto de teste tenha a maioria dos casos possíveis, a classificação vai apresentar uma percentagem de assertividade bastante elevada.

Tempo	Humidade	Dia da semana	Vento	Jogar futebol
Quente	Baixa	Segunda-feira	Sim	**Sim**
Médio	Alta	Quinta-feira	Sim	**Sim**
Frio	Alta	Sábado-feira	Não	**Não**
Frio	Alta	Segunda-feira	Não	**Não**
Médio	Alta	Quinta-feira	Sim	**Sim**
Frio	Baixa	Terça-feira	Não	**Sim**
Quente	Baixa	Quarta-feira	Não	**Não**
Frio	Alta	Quinta-feira	Sim	**Não**

Tabela 1 - Exemplo de um conjunto de dados de teste

Neste exemplo o objetivo é identificar qual os dias em que uma determinada pessoa tenciona ir ao futebol, tendo em conta a temperatura, Humidade, dia da semana e a existência do vento. Esta tabela representa os dados que se encontram no conjunto de treino, que são a base para se realizar a classificação. Quando é submetido um novo registo para classificação são comparados os atributos que ele possui com os atributos dos dados de treino. Quando for encontrado o registo de treino com mais atributos iguais ao registo de teste, então será o resultado desse registo que será atribuído.

2.6.8 Boltzmann Machines

O *Boltzmann Machines* é uma rede ligada de unidades binárias estocásticas. É amplamente utilizado na redução da dimensionalidade, no de ruído, na codificação de dados, no aprendizado de recursos, etc. O seu funcionamento tem como base a distribuição da informação utilizando um modelo gráfico probabilístico. Contudo, este tipo de análise pode ser bastante demorado e custosa para o computador. O problema pode ser simplificado impondo restrições à topologia de rede, conhecido como *Restricted Boltzmann Machines* [42].

2.6.8.1 Restricted Boltzmann Machines

O *Restricted Boltzmann Machines* é um modelo parametrizado que representa uma distribuição de probabilidades, proposto nos anos 1980s. Com o objetivo de alinhar a informação do grupo de treino com a distribuição de probabilidades, a informação é organizada de diferentes formas até apresentar uma organização favorável ao problema. Assim, o algoritmo separa a informação em dois tipos de unidades, visíveis e invisíveis. As unidades são colocadas em duas camadas diferentes [43]:

- **Unidades visíveis (v)**: São o que constituem a primeira camada e correspondem aos componentes de uma observação (ex: pode ser uma unidade para cada pixel existentes em uma imagem);

- **Unidades escondidas (h)**: São dependências entre os componentes em observação (dependências apresentadas entre os pixéis das imagens) e pertencem a segunda camada.

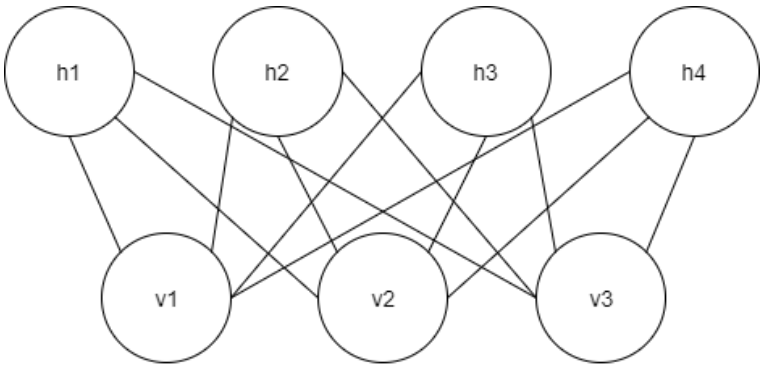

Figura 11 - Funcionamento do *Restricted Boltzmann Machines*

Como é inferida uma restrição neste modelo não existem ligações entre unidades visíveis ou escondidas. Assim, torna-se mais leve utilizar este modelo em comparação com o modelo de *Deep Bolzmann Machines*, onde existe ligações entre o mesmo tipo de unidades. O modelo representado na **Figura 11** pode ser representado pela fórmula energética [43]:

$$E = -b'v - c'h - h'Wv$$

Sendo o W representante do peso das conexões entre as unidades escondidas e as unidades visíveis. O b e c representam o balanço entre as duas camadas.
Esta fórmula pode ser traduzida diretamente para a seguinte fórmula energética livre [43]:

$$F(v) = -b'v - \sum_i \log \sum_{hi} x^{h_i(c_i + W_i v)}$$

Devido às restrições impostas pelo algoritmo, podemos dizer que as duas camadas são condicionalmente independentes. Esta afirmação pode ser traduzida matematicamente pelas seguintes fórmulas [43]:

$$p(h|v) = \prod_i p(h_i|v)$$
$$p(h|v) = \prod_j p(v_i|h)$$

2.6.8.2 Amostragem

Este algoritmo pode também ser apresentado como um modelo de grafo indireto denominado por campos aleatórios de *Markov*. Este método de operar do algoritmo pode não ser a forma mais simples, contudo este método pode ser facilmente aplicável na forma de amostras de *Gibbs* [43].

A amostra de *Gibbs* é um algoritmo que faz uma aproximação estatística, de uma sequência de observações, quando a avaliação direta da amostra é bastante difícil de obter. Isso possibilita que a aprendizagem do algoritmo e a obtenção dos resultados seja mais rápida [43]. Para realizar a amostragem é utilizado N variáveis aleatórias $S = (s_1,\ldots,s_n)$ é feita através de um subconjunto $s_i \sim p(s_i|s_{-i})$ quando s_{-i} contém o $N-1$ de outras variáveis aleatórias no S excluindo s_{-i}. O S significa um conjunto de unidades visíveis e escondidas. Devido ao facto de serem condicionalmente independentes é possível fazer amostragens das unidades visíveis simultaneamente com as invisíveis. É possível espelhar esse processo explicando as seguintes fórmulas matemáticas [43]:

$$h^{(n+1)} \sim sigm(W'v^{(n)} + c)$$
$$v^{(n+1)} \sim sigm(W'v^{(n+1)} + b)$$

O $h^{(n)}$ é o conjunto de unidades escondidos nas diferentes etapas apresentadas na cadeia de *Markov*. As expressões $h^{(n+1)}$ e $v^{(n+1)}$ são escolhidas com as funções de probabilidade $sigm(W'v^{(n)} + c)$ e $sigm(W'v^{(n+1)} + b)$ respetivamente.

2.6.8.3 Divergência Contrastiva

São utilizados processos para aumentar a velocidade da recolha e processamento das amostras [43]:

- A utilização de um conjunto de treino que esteja adequado ao problema que se pretende resolver, aumentando a velocidade de convergência;
- O algoritmo não espera que sejam executadas todas as etapas para realizar a convergência.

Este sistema tem um mecanismo de persistência dos dados. Cada vez que é executado um estado de *Marcov* esses estados são preservados para não serem afetados por modificações realizadas posteriormente.

2.6.9 Métricas de avaliação dos modelos

Para conseguir avaliar a exatidão e precisão de um algoritmo de classificação existem várias técnicas de medição que podem ser utilizadas. Essas técnicas têm como base a utilização da matriz de confusão, que permite contabilizar o número de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN) no conjunto de resultados que o algoritmo apresenta. Para conseguir distinguir os valores dentro dos parâmetros da matriz são comparados os resultados reais com os previstos pelo algoritmo, representando-os numa tabela. Para facilitar a compreensão da matriz é apresentado um exemplo na **Tabela 2** com uma breve descrição.

		Valores Verdadeiros (confirmado por análise)	
		Positivos	**Negativos**
Valores Previstos Pelo Algoritmo	**Positivos**	**VP** Verdadeiro Positivo	**FP** Falso Positivo
	Negativos	**FN** Falso Negativo	**VN** Verdadeiro Negativo

Tabela 2 - Descrição da matriz de confusão

Em seguida serão apresentadas as métricas que podem ser utilizadas para medir a eficácia do algoritmo:

- **Precisão**: É a taxa que indica os positivos são efetivamente positivos.

$$Precisão = \frac{\text{VP}}{VP + FP}$$

- ***Recall***: É a taxa que indica se os verdadeiros positivos foram identificados ou erradamente foram identificados como negativos.

$$Recall = \frac{\text{VP}}{VP + FN}$$

- **TNR:** É a taxa que indica se os verdadeiros positivos são verdadeiramente positivos ou se foram alguns identificados erradamente.

$$TNR = \frac{\text{VP}}{VP + FP}$$

- **FPR**: É a taxa que indica se os falsos positivos que foram identificados são efetivamente verdadeiros negativos.

$$TNR = \frac{\text{FP}}{TN + FP}$$

- **Taxa de acerto**: É a taxa que indica qual a eficácia apresentada pelo algoritmo. Tem em conta os dados apresentados na tabela e verifica a percentagem de acerto que o algoritmo apresenta consoante os dados obtidos.

$$Taxa\ de\ acerto = \frac{\mathrm{TP} + \mathrm{TN}}{TP + TN + FP + FN}$$

- **Taxa de erro**: É a taxa que indica a percentagem de erro que o algoritmo apresenta.

$$Taxa\ de\ erro = 1 - taxa\ de\ acerto$$

- **KAPPA**: É métrica que é utilizada para avaliar vários algoritmos e consiste em comparar os resultados da classificação com os resultados reais.

- **Curva Roc**: A curva de roc é o balanço realizado pelas métricas TRC e FPR.

Estas métricas possibilitam que exista uma avaliação pormenorizada dos resultados obtidos, medindo a eficácia do algoritmo e maximizando a precisão dos resultados [25].

2.7 Desvantagens do *Data Mining*

Até esta fase do documento existe uma tentativa de enaltecer os aspetos que se encontram no âmbito da *Data Mining* e *Data Science*. Mas como tudo no mundo existem vantagens e desvantagem na sua utilização que devem ser estudas e percebidas para maximizar o benefício e minimizar as perdas. Nesta seção serão apresentadas desvantagens do *Data Mining* e de que forma podem influenciar negativamente nos projetos.

- **Problemas de privacidade**: Com o aumento das políticas de segurança em relação aos dados pessoais esta recente tecnologia vem ameaçar o cumprimento dessas leis. Cada vez mais as pessoas demonstram preocupações com os seus dados e exigem garantias de que os dados estão a ser usados corretamente e não para atividades ilícitas [15];
- **Problemas de segurança:** As informações que são armazenadas são cada vez mais sensíveis do ponto de vista da segurança. Existe nessa mesma informação números de cartões de crédito, moradas, números de telefones, etc. Isso faz com que a população cada vez que se apercebe de ataques informáticos em grandes companhias se sinta desprotegida e ameaçada devido ao acesso indevido as suas informações [15];
- **Problemas disputados por informação imprecisa**: A maioria da informação utilizada é retirada automaticamente de sistemas informáticos que debitam *terabytes* de informação. Torna-se impossível avaliar se a informação está correta e vai ao encontro da realidade. No entanto, esta informação é utilizada para fazer decisões de mercado, o que pode levar a decisões catastróficas se a informação que esteve por base da decisão não for a correta [15];
- **Problemas na obtenção dos recursos necessários**: Para colocar um sistema de *Data Mining* a funcionar pode ser muito custoso e complicado. Quando se trata de uma aplicação que processa *terabytes* é possível que, caso não exista recursos físicos suficientes, o tempo de espera para o processamento da informação seja bastante elevado. Contudo, os custos para a melhoria dos recursos físicos podem ser muito elevados também, limitando por vezes alguns projetos em desenvolvimento [15].

2.8 Linguagens de programação

Para este projeto foram estudadas as linguagens *R, Java* e *Python*. Essas são as linguagens utilizadas pelo departamento onde este projeto se inseriu, e por esse motivo foram as únicas abordadas neste documento.

Em seguida será apresentada uma breve descrição das linguagens selecionadas e a identificação das vantagens e desvantagens. No final será feita a avaliação das opções estudadas.

2.8.2.1 R

A linguagem *R* foi inicialmente desenvolvida por matemáticos como uma alternativa *open source* para os *softwares* SAS e MATLAB, tornando-se atualmente uma das linguagens mais populares do seu segmento. Com a análise de dados a ser um foco das grandes empresas, o R tornou-se uma ferramenta bastante utilizada para fazer relatórios, análises e gráficos.

Diferente do paradigma orientado a objetos das linguagens de programação *Java* e *Python*, o R é processual. Significa que depende de uma sequência de sub-rotinas para executar um programa, fazendo com que seja possível identificar no código as várias operações realizadas de forma intuitiva. Em contrapartida é necessário escrever mais linhas de código na criação de um determinado algoritmo em comparação com o paradigma orientado a objetos [16].

As vantagens desta linguagem são:

- Bom a realizar análises de dados sensíveis (Modelos financeiros);
- Uma panóplia de recursos para produzir relatórios;
- Constantes *updates* das bibliotecas de análise, devido à sua enorme comunicação.

As desvantagens desta linguagem são:

- O *R* foi desenhado a pensar na análise de dados, o que o torna mais lento em comparação com o *Python* e *Java*;
- Não é adequado para um projeto de grande escala. Geralmente é usado para fazer protótipos, sendo a implementação feita numa linguagem mais flexível (ex: *Python* ou *Java*);
- Para pessoas com experiência nas áreas da matemática esta é a linguagem perfeita. Contudo, quando a formação é na área da informática torna-se pouco intuitiva e com algumas limitações.

2.8.2.2 Java

O *Java* é uma linguagem robusta, que contém várias *frameworks* e bibliotecas que permitem adaptar soluções em termos de escalabilidade e portabilidade. Estas ferramentas estão localizadas na denominada *java platform*. Devido à sua robustez, o java é utilizado para resolver problemas de performance em algoritmos que manipulem grandes volumes de dados. Para ajudar na otimização do código o java utiliza o paradigma orientado a objetos que oferece flexibilidade e extensibilidade devido à sua estrutura modular [16].

As vantagens desta linguagem são:

- É excelente em termos de performance e escalabilidade. Superando as outras linguagens em análise;
- Com a JVM (*Java Virtual Machine*) e o *Scala*, linguagem de programação que funciona em conjunto com JVM, é relativamente rápido criar soluções customizadas.

As desvantagens desta linguagem são:

- Quando se trata de análises mais detalhadas o *Java* é a linguagem menos indicada das três analisadas, pois não apresenta bibliotecas e *frameworks* tão desenvolvidas tecnicamente como as restantes.

2.8.2.3 Python

O *Python* é rápido, eficaz e dá importância à reutilização de código. Contrariamente ao Java e ao C++, o *Python* consegue expressar conceitos com poucas linhas de código, o que torna o *debug* simples. Similar ao *R*, contém bibliotecas que permitem a realização de relatórios, análises e gráficos. Em destaque estão as bibliotecas *NumPy* e *matplotlib* que permitem utilizar funcionalidades presentes no MATLAB. Analisando estas três linguagens o *Python* apresenta as vantagens das duas restantes com um grau de menor eficácia. Ou seja, consegue realizar análises consideravelmente complexas com a possibilidade de implementar mecanismos para gerir a performance e escalabilidade do *software* [16].

As vantagens desta linguagem são:

- Caso a solução se traduza num conjunto de diferentes sistemas, o *Python* apresenta ferramentas que podem fazer a sua integração;
- Considerada uma linguagem de programação adequada tanto para protótipos como para produção;
- Relativamente fácil de aprender a programar;
- Apresenta agilidade e flexibilidade a extrair informação de documentos de texto sem indentação, *websites* e redes sociais.

As desvantagens desta linguagem são:

- Quando a análise é detalhada ou a velocidade de processamento é um requisito com prioridade alta o *Python* não é a melhor opção;
- Problemas a paralelizar as tarefas, o que resulta num aumento da latência na comunicação em comparação ao Java (Problema: GIL*(Global Interpreter Lock*)).

Em suma para este projeto que a linguagem mais indicada é o *Python*, pois apresenta as vantagens das restantes linguagens. Isso proporciona uma flexibilidade importante para resolver problemas inesperados que possam surguir no processo de desenvolvimento da solução.

2.8.3 Ferramentas de Machine Learning

Existem dezenas de bibliotecas que possuem ferramentas para realizar projetos de *Machine Learning*. Contudo, a maioria apresenta uma especialização em alguns tipos de algoritmos de *Machine Learning*. Para manter a flexibilidade do projeto, foram pesquisadas as ferramentas

mais genéricas. Sendo assim, foram escolhidas para análise a biblioteca *scikit-learn* e a aplicação *Weka*.

2.8.3.1 Scikit-learn

A *scikit-learn* é, definitivamente, uma das bibliotecas *Machine Learning* mais popular entre as línguas. Tem uma grande quantidade de recursos para *Data Mining* e análise de dados, tornando-se uma escolha de topo para pesquisadores e desenvolvedores. É construído sobre as populares bibliotecas *NumPy, SciPy* e *matplotlib*, por isso terá uma sensação familiar para as muitas pessoas que já usam essas bibliotecas. Embora, em comparação com muitas das outras bibliotecas existentes, este é um nível um pouco mais baixo e tende a atuar como base para muitas outras implementações de *Machine Learning* [22].

2.8.3.2 Weka

O *Weka* foi desenvolvido pela Universidade de Waikato, na Nova Zelândia, e possibilita a implementação de algoritmos inteligentes a um conjunto de dados. Assim como, possibilita a transformação dos dados recorrendo a algoritmos de pré-processamento, análise de resultados e métricas de performance.

É possível encontrar vários algoritmos de *Machine Learning* contidos dentro do *Weka*, proporcionando uma flexibilidade nos testes realizados, podendo-se alterar o tipo de algoritmo consoante os resultados obtidos, comparando-os num ambiente proporcionado pelo *software*. Esta aplicação inclui todos os algoritmos *standards*: regressão, classificação, agregação, regras de associação e seleção de atributos [17].

As seguintes funcionalidade podem ser encontradas no *Weka* [18]:

- **Pré-processamento**: Carregamento e manipulação de um conjunto de dados num formulário;
- **Classificação**: Classificação ou regressão dos dados;
- **Agrupamento**: Utilização de algoritmos de agrupamento;
- **Associação**: Utilização de algoritmos de associação;
- **Seleção de atributos**: Utilizar algoritmos que visam a escolha dos melhores atributos com o objetivo de melhorar a performance dos algoritmos de *Machine Learning*;
- **Visualização**: Visualização da relação entre atributos.

Ao utilizar o *Weka* é possível escolher uma de quatro interfaces existentes. As interfaces podem ser [19]:

- ***Simple CLI*** *(Command Line Interface)*: É a interface de linha de comandos que permite executar os comandos do *Weka*. É a interface mais flexível, contudo é necessário ter um conhecimento sólido dos comandos existentes.
- ***Explorer***: O *Explorer* é uma interface gráfica que permite aceder às ferramentas de uma forma intuitiva. São utilizados menus e formulários de forma a impedir o utilizador de realizar operações não permitidas, são usados *pop ups* informativos em cada campo existente e valores por *default* para possibilitar a obtenção de resultados com o mínimo esforço.
- ***Knowledge Flow***: Permite o desenvolvimento de projetos num ambiente gráfico com a utilização de fluxos de informação. Sendo assim, permite arrastar caixas que representam algoritmos ou ferramentas, criando uma configuração que inclui os dois

tipos de recursos existentes. Isso permite que seja possível utilizar na mesma configuração ferramentas de pré-processamento, algoritmos inteligentes e métodos de avaliação de performance. Possibilita também, uma perspetiva global do programa criado. É necessário referir que esta interface tem uma desvantagem em comparação com as restantes, pois guarda a configuração na memória principal. Isso significa que só é possível criar nesta interface modelos com uma escala pequena ou média.

- ***Experimenter***. É um ambiente gráfico utilizado para realizar testes com diferentes técnicas de regressão e classificação, permitindo comparar os resultados entre si. Este processo pode também ser realizado no *Explorer*, contudo neste ambiente a maior parte das operações são feitas automaticamente, os dados podem ser divididos em vários grupos de testes, é possível coletar dados estatísticos e realizar testes de performance. Na análise comparativa é possível utilizar o RMI para paralelizar as experiências pretendidas.

O *Weka* tem um formato de dados proprietário denominado *arff*. Contudo, é possível modificar certos formatos de ficheiros dentro do próprio *Weka*, facilitando o trabalho de realizar manualmente a transformação dos dados. Mas caso a extensão do ficheiro não esteja contida na lista que o programa apresenta, a transformação pode ser realizada da seguinte forma.

@RELATION "relação existente nos dados"

@ATTRIBUTE M {*array* de atributos}

@ATTRIBUTE "identificar atributos existentes"

É obrigatório colocar este cabeçalho no topo do ficheiro que se quer utilizar, sendo os dados colocados logo em seguida da *tag* "@DATA". Para finalizar a transformação é obrigatório guardar os dados com a extensão *arff*.

2.8.3.2 Pentaho

É um *software* de inteligência no negócio *opensource* que permite realizar análise de dados e produção automática de relatórios. É uma ferramenta bastante utilizada na área de BI (*Business Intelligence*) para extrair conhecimento de grandes volumes de dados, oferecendo balanço entre custo beneficio nos modelos produzidos pela aplicação [20] [21].

O *software* utiliza *dashboards* para facilitar a análise, integração e criação dos relatórios das informações tratadas. É possível utilizar diferentes formatos na realização dos relatórios. Além de ser *opensource,* com a possibilidade de utilizar diferentes formatos de dados e com uma arquitetura própria que permite a fácil compreensão da ferramenta, o *Pentaoh* apresenta outras vantagens, sendo elas [20] [21]:

- **Entrega de dados controlada**: Ele combina dados confiáveis e oportunos para análise de dados poderosa em escala para todos os usuários em todos os ambientes;
- **Compatibilidade com diferentes formatos**: O *Pentaoh* permite trabalhar com diferentes formatos de dados, possibilitando o uso de informação proveniente da *cloud*, modelos de dados móveis e modelos híbridos**;**
- **Integração dos diferentes formatos**: É possível de um modo intuitivo fazer a integração dos diferentes modelos de dados sem grande esforço;
- **Simples e intuitiva**: A ferramenta é simples e intuitiva de utilizar, devido ao uso de *dashboard* em detrimento de código de programação.

2.9 Avaliação da eficácia da campanha

Para perceber se a campanha foi bem-sucedida é utilizada pelo ACM uma forma de medir a eficácia da campanha. Isso permite perceber qual o grau de influência que a campanha produziu nos clientes e identificar futuras melhorias para as versões seguintes da campanha. Para o fazer é utilizado o grupo de controlo e o *Lift*.

O grupo de controlo é um grupo separado do resto da experiência em que a variável independente testada não pode influenciar os resultados. Isso isola os efeitos da variável independente no experimento e pode ajudar a excluir explicações alternativas dos resultados experimentais [23].

O *Lift* é uma forma de medir como uma campanha afeta uma métrica chave. No marketing móvel, é possível medir o aumento do carregamento, despesa nas aplicações ou das chamadas telefónicas. O *Lift* é calculado como a percentagem de aumento ou diminuição em cada métrica para os clientes que receberam um incentivo em detrimento dos clientes que na mesma situação não foram incentivados, a estes últimos clientes definimos como sendo pertencentes ao grupo de controlo [24].

Na execução de uma campanha no ACM é possível escolher, dentro dos clientes aptos para as campanhas, a percentagem que vai para o grupo de controlo (CG) e a que vai para o público-alvo (PA). Isso possibilita a comparação entre os dois segmentos e identifica o grau de eficácia da campanha. A percentagem escolhida para os dois grupos é variável consoante o desejo da equipa que define as campanhas.

Para avaliar a eficácia do grupo de controlo é utilizada a fórmula:

$$Eficácia\ GC\ = \frac{CG\ aderentes\ a\ campanha}{CG} * 100$$

Para avaliar a eficácia do público-alvo é utilizada a fórmula:

$$Eficácia\ PA\ = \frac{PA\ aderentes\ a\ campanha}{PA} * 100$$

Para avaliar a eficácia referente ao *Lift* do GC da campanha é utilizada a fórmula:

$$Lift\ = \frac{(Eficácia\ PA - Eficácia\ CG)}{Eficácia\ CG} * 100$$

Capítulo 3
Objetivos e Método de Abordagem

3.1 Descrição do produto

Nesta secção é apresentado o problema atual e a situação que levou ao desenvolvimento da solução.

O problema que o ACM apresenta tem origem nos mecanismos de filtragem existentes para a escolha do público-alvo. É esperado que com o uso dos mecanismos que a percentagem de adesão às campanhas seja o mais próximo dos 100%. Contudo, os resultados atuais rondam os 6%, bastante abaixo do ideal.

Assim sendo, a empresa tem vindo a explorar novas tecnologias que consigam identificar quais os clientes que são mais suscitáveis a uma campanha lançada pelo ACM, utilizando o histórico de campanhas anteriores. É expectável que se consiga reduzir o número de mensagens enviadas e aumentar o número de adesões às campanhas ou identificar quais as campanhas mais adequadas para um determinado cliente. Este projeto foi denominado como **NBO (***Next Best Offer***)** ou NBO/A (*Next Best Offer/Action*).

Para se conseguir explicar o fluxo de informação e o problema descrito são apresentados o diagrama de fluxo de dados e o diagrama sequência.

3.1.1 Diagrama de fluxo de dados

O diagrama de fluxo de dados é útil para identificar os problemas que estão presentes no fluxo de trabalho habitual. Portanto, são identificadas as partes envolvidas e os pontos onde são realizadas trocas de informação. Nas trocas de informações são identificados os problemas que devem ser tratados na solução que se pretende desenvolver.

Neste diagrama foi identificado um único problema que devido ao seu grau de severidade tem extrema relevância para este projeto.

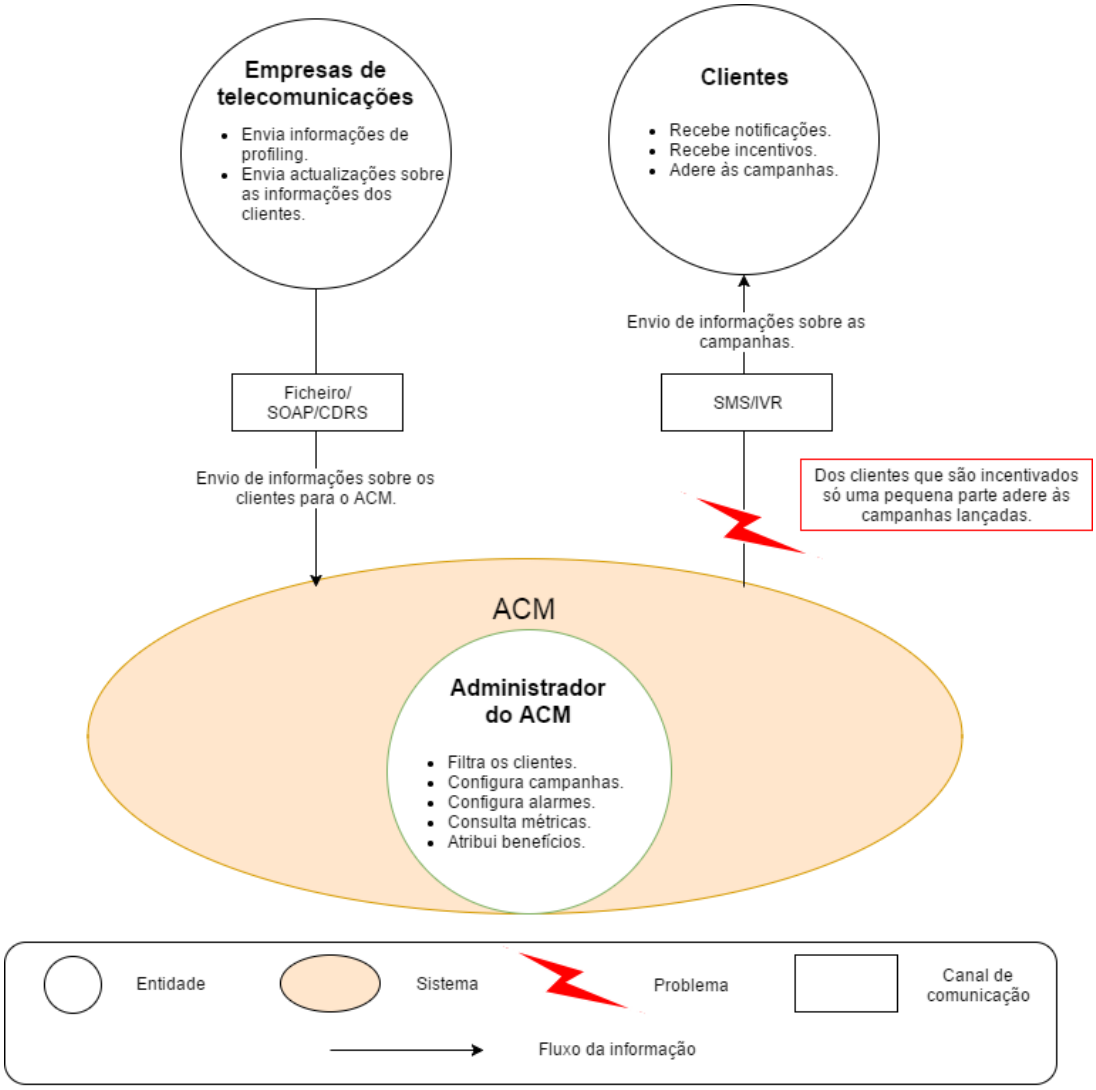

Figura 12 – Diagrama de fluxo de dados

3.1.2 Diagrama de sequência

O problema principal deve ser analisado de forma cuidadosa com a utilização do diagrama de sequência. Contudo, para chegar ao problema é necessário realizar um conjunto de etapas que não estão contempladas neste diagrama. Para manter o foco assume-se que as etapas anteriores foram realizadas com sucesso.

Como se pode ver na **Figura 13**, a sequência de atividades demonstra que o ACM está bem estruturado e que cada operação tem uma finalidade bem definida. Mas uma das operações não está de acordo com as expectativas da empresa, não se traduzindo nos resultados esperados.

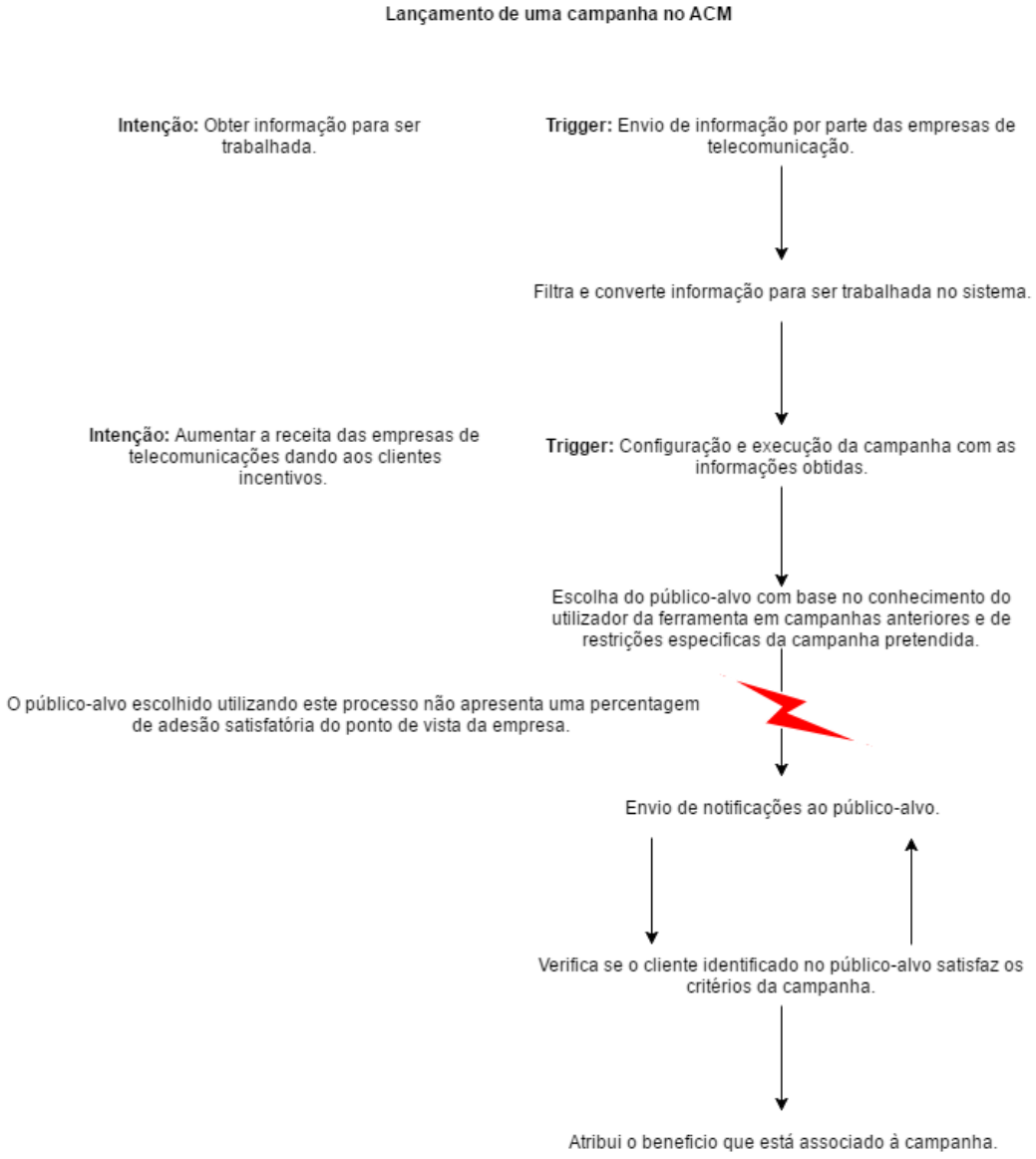

Figura 13 – Diagrama de sequência

3.2 Ambiente operacional

O sistema será posto em funcionamento em sistemas operativos *Red Hat Linux* com uma base de dados *Oracle*.

A base de dados Oracle é uma coleção de dados tratados como uma unidade. O objetivo de um banco de dados é armazenar e recuperar informações relacionadas. Foi a primeira base de dados a ser criada para o uso industrial, apresentado mecanismos que facilitam o desenvolvimento em larga escala [26].

3.3 Restrições

O sistema vai ser desenvolvido de forma a que após o estágio seja possível ser usado pelos elementos que compõem a equipa deste projeto. Portanto, a solução deve ser desenvolvida com ferramentas que os elementos estejam familiarizados, tanto na fase do planeamento como do desenvolvimento.

O sistema deve estar integrado com a plataforma ACM e deve ser pensado de forma a ser possível implementá-lo com outras tecnologias existentes dentro da empresa.

O protocolo de comunicação deve ser mantido o mesmo que está implementado no ACM (*REST*).

O tempo de desenvolvimento do projeto está sujeito ao tempo de realização do estágio.

3.4 Assunções e dependências

Até ao momento só existem duas assunções sobre o projeto:

- Será concedido total acesso aos ambientes de produção da plataforma ACM e bases de dados associadas;
- Será concedida informação de campanhas recentes para que seja possível medir o desempenho do produto desenvolvido.

3.5 Requisitos de interfaces externas

Interfaces de Hardware

- Para o desenvolvimento deste projeto é necessário um servidor que esteja ligado à internet e que contenha os recursos de memória suficientes para que o produto desempenhe a tarefa pretendida.

Interfaces de Software

- A máquina deve apresentar o sistema operativo *x86-64-redhat-linux-gnu*, com o interpretador de *Python* e de *Java* instalados. Para realizar comunicação entre componentes deve ser implementado a *API Rest* com recurso a um modulo desenvolvido em *Python* .

3.6 Arquitetura do software

Nesta secção do relatório será apresentada uma vista geral da forma como o sistema foi constituído e a devida explicação.

A função desta descrição é definir os objetivos da arquitetura, os casos possíveis de utilização do *software* e indicar as escolhas tomadas para o desenvolvimento do projeto.

3.6.1 Proposta

Com esta secção é possível perceber as diferenças da arquitetura como um todo, fornecendo diferentes pontos de vista. Assim, é possível segmentar a audiência utilizando os diagramas mais adequados.

Para conseguir retratar o *software* da melhor forma possível será utilizada a estrutura baseada no modelo arquitetural de *Phiippe Kruchten's "4+1"* [37].

Figura 14 - Phiippe Kruchten's "4+1"

O modelo representado na **Figura 14** permite dar a conhecer aos diferentes públicos-alvo o projeto desenvolvido e a sua organização.

3.6.2 Âmbito do projeto

Para esta secção é esperado uma explicação detalhada da arquitetura do NBO e da integração necessária do ACM.

O foco está nos aspetos da arquitetura do projeto com maior significado. É importante nesta fase perceber o projeto como um todo, identificando as ferramentas necessárias e os paradigmas subjacentes. Este documento serve para complementar os documentos que foram escritos até ao momento.

3.6.3 Representação arquitetural

Os diagramas que foram utilizados para descrever a arquitetura do NBO foram:

- Vista de casos de uso:
    - Audiência: Todos os *stackholders* do sistema, incluindo utilizadores finais;
    - Área: Na vista dos casos de uso são descritos em detalhes os requisitos funcionais existentes no projeto;
    - Artefactos relacionados: Diagrama de casos de uso, casos de uso.

- Vista lógica:
    - Audiência: Todos os *stackholders* do sistema, incluindo utilizadores finais;
    - Área: Descreve os casos de uso mais importantes e explica o propósito do projeto;
    - Artefactos relacionados: Diagrama lógico de contexto e arquitetura de blocos.
- Vista de processo:
    - Audiência: Integradores;
    - Área: Na vista de processo é possível descrever o comportamento dos aspetos dinâmicos, os processos do sistema, a forma como eles comunicam e o comportamento em execução;
    - Artefacto relacionado: Diagrama de componente conector.
- Vista de desenvolvimento:
    - Audiência: Programador;
    - Área: Descreve os módulos e subsistemas da aplicação;
    - Artefacto relacionado: Diagrama de camadas.
- Vista física:
    - Audiência: Responsáveis pela instalação;
    - Área: Na vista de desenvolvimento é ilustrado o sistema da perspetiva do programador e das suas preocupações;
    - Artefactos relacionados: Diagrama de implementação.

3.6.4 Objetivos da arquitetura e restrições / Requisitos não funcionais

Os requisitos não-funcionais definem as qualidades gerais e atributos de qualidade do sistema implementado. Do conjunto de requisitos existentes foram escolhidos os que são mais importantes para o projeto. É importante salientar que os requisitos escolhidos afetaram diretamente a construção da arquitetura. O conjunto de atributos avaliados foi: disponibilidade, segurança, interoperabilidade, segurança, modificabilidade e funcionalidade. Na escolha dos atributos de qualidade é boa prática escolher somente três ou quatro, pois torna-se impossível espelhar todos numa arquitetura de *software*.

Os atributos de qualidade mais relevantes para este projeto são:

- **Desempenho**: É evidenciado pela capacidade do sistema em cumprir as suas funções no tempo especificado;
- **Interoperabilidade**: Traduz o grau com que dois ou mais sistemas podem trocar informações significativas entre si de forma útil;
- **Disponibilidade**: Descreve até que ponto um elemento do sistema realiza as tarefas que lhe são atribuídas sempre que lhe são requisitadas;
- **Funcionalidade**: É a capacidade do sistema realizar as tarefas para que foi concebido.

3.6.4.1 Descrição dos atributos de qualidade

Consideramos o **desempenho** como um dos atributos de qualidade prioritários devido ao facto do sistema ter de responder num espaço de tempo muito curto aos pedidos efetuados

pelo ACM com o objetivo de classificar os registos dos utilizadores que pretendem entrar na campanha promocional.

A **interoperabilidade** é um atributo de qualidade importante neste tipo de sistemas visto que este envolve vários subsistemas que interagem entre si. Na verdade, só o facto de estar a operar em conjunto com a plataforma ACM já torna necessário a identificação deste atributo.

A **disponibilidade** foi um dos atributos de qualidade considerados prioritários uma vez que o objetivo do NBO deve estar sempre operacional nas campanhas lançadas pelo ACM. Como podem ser lançadas dezenas de campanhas por dia e os clientes devem ficar associados à medida que vão entrando registos no sistema, o NBO tem de estar sempre pronto a fazer classificações dos dados.

A **funcionalidade** é um atributo que permite verificar se os objetivos do projeto estão a ser cumpridos. Devemos pensar neste atributo como a validação dos objetivos que foram identificados no início do projeto.

A razão que leva à criação de cenários para os atributos é demonstrar como alguns problemas podem ser resolvidos, e identificar formas claras de medir os atributos.

3.6.4.2 Cenários dos atributos qualidade

Nesta secção são identificados os atributos qualidade mais indicados para este projeto. Para o fazer foram formulados e subdivididos em categorias cenários de possíveis utilizações da solução desejada. Pretende-se assim identificar as origens de cada cenário, o estímulo que levou a que o cenário fosse criado, o ambiente em que o cenário se realizou, os elementos do sistema que estão envolvidos e a medida que se vai utilizar para medir o desempenho do atributo identificado.

Em seguida são apresentadas tabelas com as informações acima identificadas. Os cenários já se encontram subdivididos.

Título: Desempenho	**ID cenário**: 1
Origem	Administrador do ACM pretende lançar uma campanha utilizando o NBO.
Estímulo	Processar registos de utilizadores novos.
Ambiente	Ambiente de funcionamento normal.
Elementos do sistema	Utilização do NBO.
Resposta do sistema	Classificar os registos recebidos no menor tempo possível.
Medidas significativas	Deve processar pelo menos 200.000 registos por minuto.

Tabela 3 - Cenário 1

Título: Interoperabilidade	**ID cenário**: 2
Origem	Administrador do ACM pretende lançar uma campanha utilizando o NBO.
Estímulo	Enviar informação para o NBO para ser processada.
Ambiente	Ambiente de funcionamento normal.
Elementos do sistema	Utilização do NBO.
Resposta do sistema	Receção dos registos tendo como ponto de envio o ACM.
Medidas significativas	A informação deve passar do ACM para o NBO com sucesso de 100%.

Tabela 4 - Cenário 2

Título: Disponibilidade	**ID cenário**: 3
Origem	Administrador do ACM pretende lançar uma campanha utilizando o NBO.
Estímulo	Processar registos de utilizadores novos.
Ambiente	Ambiente de funcionamento normal.
Elementos do sistema	Utilização do NBO.
Resposta do sistema	O sistema processa a informação recebida e enviar para o ACM novamente.
Medidas significativas	O sistema deve estar em funcionamento em 99% do tempo.

Tabela 5 - Cenário 3

Título: Funcionalidade	**ID cenário**: 4
Origem	Administrador do ACM pretende lançar uma campanha utilizando o NBO.
Estímulo	Processar registos de utilizadores novos.
Ambiente	Ambiente de funcionamento normal.
Elementos do sistema	Utilização do NBO.
Resposta do sistema	O sistema processa a informação recebida e enviar para o ACM novamente.
Medidas significativas	O sistema deve processar os registos recebidos sem nenhum erro.

Tabela 6 - Cenário 4

3.6.5 Vista de casos de uso

Nesta secção são apresentados os possíveis casos de uso para o sistema que se pretende desenvolver. Os casos de uso são uma forma de identificar quais os atores, máquinas ou pessoas, que interagem com o sistema e qual o comportamento que o sistema deve apresentar perante as suas possíveis ações. Isso ajuda a identificar as pré-condições, interfaces, respostas do sistema e exceções.

É apresentado primeiro o diagrama de contexto na **Figura 15** de forma a que o leitor tenha uma perceção do sistema.

Figura 15 - Diagrama de contexto

3.6.6 Casos de uso

Nesta secção é apresentado o diagrama de casos de uso e os respetivos casos de uso, com as devidas informações detalhadas sobre o sistema. Os casos de uso são uma representação das ações que se podem realizar no sistema que se pretende desenvolver. São compostos pelos seguintes critérios:

- **Referência**: Identifica unicamente o caso de uso.
- **Nome**: Nome de identificação do caso de uso.
- **Actor principal**: Identifica o autor que desencadeia o caso de uso.
- **Prioridade**: Qual a importância do caso de uso para o funcionamento da aplicação e para o cliente. Para identificar a prioridade vai ser utilizado a priorização *MoSCoW* que classifica da seguinte forma:
  - ***Must*** – Descreve um requisito que tem de ser obrigatoriamente cumprido;
  - ***Should*** – Descreve um requisito de elevada prioridade;
  - ***Could*** – Descreve um requisito que os clientes têm interesse em implementar, mas não é obrigatório;
  - ***Won't*** – Descreve um requisito que pode não ser implementado até ao lançamento da aplicação, mas existe intenção de se implementar mais tarde;
- **Descrição**: Explica em que é que consiste o caso de uso;
- **Pré-condições:** Condições necessárias para efetuar o caso de uso;
- **Última revisão:** Data da última modificação do caso de uso;
- **Garantias mínimas**: O mínimo que o programa apresenta ao utilizador na execução do caso de uso;
- **Garantias máximas**: O esperado pelo programa na execução do caso de uso;
- **Processo**: Descrição das etapas necessárias para a realização do caso de uso;
- **Exceções**: Descrição de situações em que os casos de uso não têm o fluxo esperado de eventos.

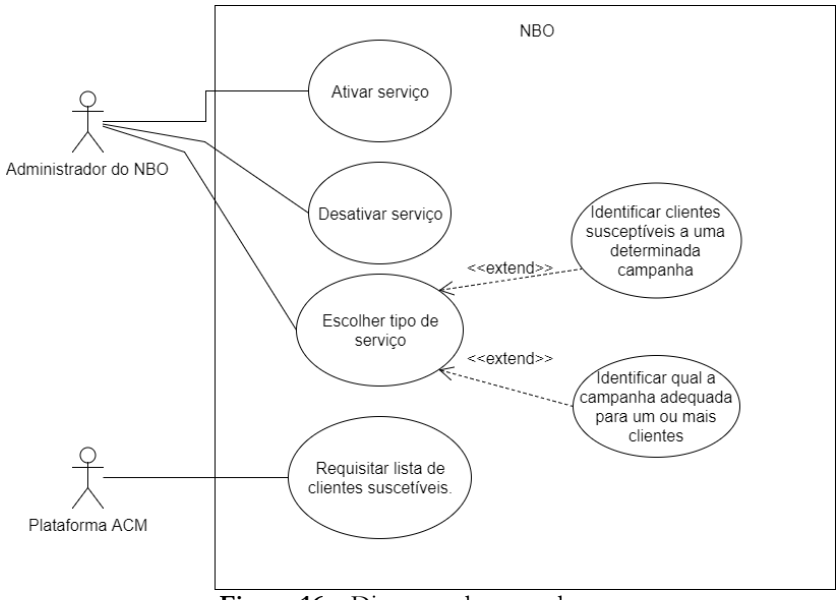

Figura 16 – Diagrama de casos de uso

Referência	UC1
Nome	Ativar serviço.
Autor principal	Administrador do ACM.
Prioridade	*Could*
Descrição	Ativa o serviço proveniente do NBO.
Pré-condições	Ter acesso na plataforma ACM.
Última revisão	19/05/2017
Garantias mínimas	Conseguir visualizar o campo que permite desligar e ligar o serviço.
Garantias máximas	O serviço fica ativo e pronto a funcionar.
Processo	1. Iniciar conta na plataforma ACM. 2. Ir às definições do público-alvo. 3. Modificar o campo para ativo.
Exceções	1.a Não ter ligação à internet. 1.a.1 É mostrada no ecrã uma mensagem de erro informando o utilizador. 1.b Não ser o login ou password corretos. 1.b.1 É mostrada no ecrã uma mensagem de erro informando o utilizador.

Tabela 7 - Caso de uso 1

Referência	UC2
Nome	Desativar serviço.
Autor principal	Administrador do ACM.
Prioridade	*Could*
Descrição	Desativa o serviço proveniente do NBO.
Pré-condições	Ter acesso na plataforma ACM.
Última revisão	19/05/2017
Garantias mínimas	Conseguir visualizar o campo que permite desligar e ligar o serviço.
Garantias máximas	O serviço fica desativado.
Processo	1. Iniciar conta na plataforma ACM. 2. Ir às definições do público-alvo. 3. Modificar o campo para desativo.
Exceções	1.a Não ter ligação à internet 1.a.1 É mostrada no ecrã uma mensagem de erro informando o utilizador. 1.b Não ser o login ou password corretos. 1.b.1 É mostrada no ecrã uma mensagem de erro informando o utilizador.

Tabela 8 - Caso de uso 2

Referência	UC3
Nome	Identificar clientes suscetíveis a uma determinada campanha.
Autor principal	Administrador do ACM.
Prioridade	*Must*
Descrição	Utilizando registos de campanhas anteriores são identificados clientes que sejam suscetíveis à campanha.
Pré-condições	Ter acesso na plataforma ACM.
Última revisão	19/05/2017
Garantias mínimas	Nada é modificado.
Garantias máximas	Sistema fica pronto para operar neste modo.
Processo	1. Iniciar conta na plataforma ACM. 2. Ir às definições do público-alvo. 3. Modificar o campo para ativo. 4. Colocar o campo ”Modo de utilização” como detetor de clientes suscetíveis.
Exceções	1.a Não ter ligação à internet. 1.a.1 É mostrada no ecrã uma mensagem de erro informando o utilizador. 1.b Não ser o login ou password corretos. 1.b.1 É mostrada no ecrã uma mensagem de erro informando o utilizador.

Tabela 9 - Caso de uso 3

Referência	UC4
Nome	Identificar qual a campanha adequada para um ou mais clientes.
Autor principal	Administrador do ACM.
Prioridade	*Must*
Descrição	Utilizando registos de vários tipos de campanhas diferentes são comparados clientes aderentes a essas campanhas com os registos submetidos no NBO. É esperado que se consiga inserir os clientes submetidos nas campanhas adequadas ao seu perfil.
Pré-condições	Ter acesso na plataforma ACM.
Última revisão	19/05/2017
Garantias mínimas	Nada é modificado.
Garantias máximas	Sistema fica pronto para operar neste modo.
Processo	1. Iniciar conta na plataforma ACM. 2. Ir às definições do público-alvo. 3. Modificar o campo para ativo. 4. Colocar o campo “Modo de utilização” como identificação de campanhas adequadas.
Exceções	1.a Não ter ligação à internet. 1.a.1 É mostrada no ecrã uma mensagem de erro informando o utilizador. 1.b Não ser o login ou password corretos. 1.b.1 É mostrada no ecrã uma mensagem de erro informando o utilizador.

Tabela 10 - Caso de uso 4

Referência	UC5
Nome	Requisitar lista de clientes suscetíveis à campanha.
Autor principal	Administrador do ACM.
Prioridade	*Must*
Descrição	A plataforma faz o pedido da classificação ao ACM de forma automática caso o serviço esteja ativo.
Pré-condições	Ter acesso na plataforma ACM.
Última revisão	19/05/2017
Garantias mínimas	Envio da campanha para os clientes.
Garantias máximas	Envio para clientes que são suscetíveis à campanha.
Processo	1. Iniciar conta na plataforma ACM. 2. Ir às definições do público-alvo. 3. Modificar o campo para ativo. 4. Enviar campanha.
Exceções	1.a Não ter ligação a internet. 1.a.1 É mostrada no ecrã uma mensagem de error informando o utilizador. 1.b Não ser o login ou password corretos. 1.b.1 É mostrada no ecrã uma mensagem de erro informando o utilizador. 4.a O algoritmo não consegue processar toda a informação 4.a.1 É enviada uma notificação para o administrador do ACM.

Tabela 11 – Caso de Uso 5

3.6.7 Requisitos funcionais

Nesta seção será apresentada uma tabela com os requisitos funcionais. A tabela é constituída pela referência do requisito, o nome, descrição e a sua prioridade.

Referência	Nome	Descrição	Prioridade
UC1	Ativar serviço.	Ativa o serviço proveniente do NBO.	*Should*
UC2	Desativar serviço.	Desativa o serviço proveniente do NBO.	*Should*
UC3	Identificar clientes suscetíveis a uma determinada campanha.	Utilizando registos de campanhas anteriores são identificados clientes que sejam suscetíveis à campanha.	*Must*
UC4	Identificar qual a campanha adequada para um ou mais clientes.	Utilizando registos de vários tipos de campanhas diferentes são comparados clientes aderentes a essas campanhas com os registos submetidos no NBO. É esperado que se consiga inserir os clientes submetidos nas campanhas adequadas ao seu perfil.	*Must*
UC5	Requisitar lista de clientes suscetíveis à campanha.	A plataforma faz o pedido da classificação ao ACM de forma automática caso o serviço esteja ativo.	*Must*

Tabela 12 - Requisitos funcionais

3.6.8 Vista lógica

O objetivo desta secção é detalhar na sua generalidade a arquitetura que foi utilizada no desenvolvimento do projeto. Os componentes do sistema estão identificados, explicados e apresentados com uma estrutura bem definida. A arquitetura torna-se imprescindível para o planeamento, implementação e organização do produto. Os diagramas utilizados estão implicitamente ligado e pertencem ao mesmo projeto, contudo os conteúdos que são analisados podem ser diferentes.

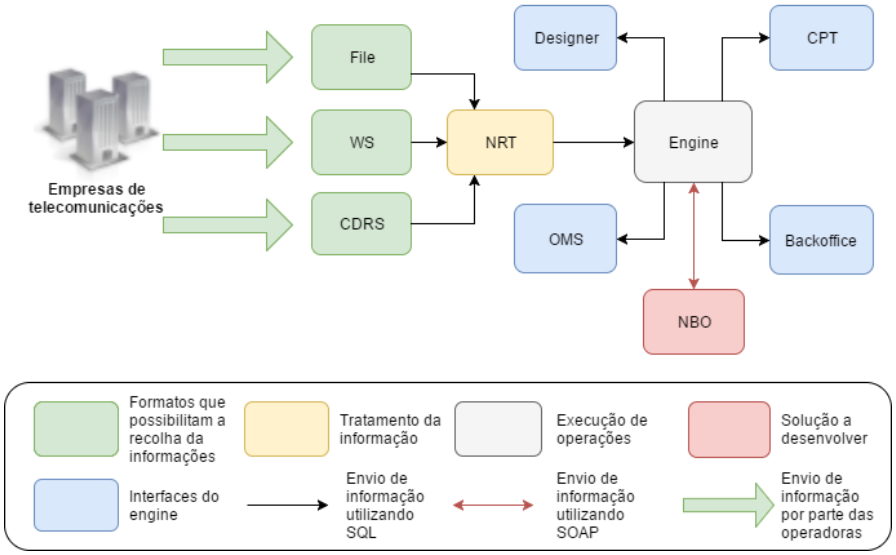

Figura 17 – Diagrama lógico de contexto

Na **Figura 17** está representada a arquitetura do ACM com a introdução do NBO. Podemos verificar que os dados referentes aos utilizadores são recebidos de três formas diferentes: Ficheiro, *Web Service* e Base de dados. Em seguida são tratados no NRT que filtra, estrutura e envia a informação para o *Engine*, que por sua vez é constituído pelo *designer*, CPT, OMS e *backoffice*.

- ***Designer***: É utilizado para criar, configurar e lançar campanhas promocionais, atribuir restrições ao público-alvo, configurar métricas e alarmes;
- ***Campaign Performance Tracking* (CPT)**: É utilizado para apresentar estatísticas sobre indicadores medidos na plataforma;
- ***Backoffice***: Apresenta os *logs* das operações que foram efetuadas no âmbito do ACM;
- **Operation and Maintenance System (OMS)**: Administra funcionalidades e interfaces existentes.

O NBO é um serviço modular que possibilita a discriminação dos clientes que estão ou não estão suscetíveis a uma determinada campanha. É alcançado com a utilização do algoritmo na entrada dos registos de clientes nas campanhas promocionais. Isso torna imprescindível que o algoritmo seja rápido a classificar para não influenciar o tempo de execução normal da plataforma ACM.

Para ajudar a perceber a composição do sistema é possível verificar na **figura 18** o diagrama de classes, que possibilita entender a forma como será desenvolvido o *software*. É importante entender os pontos principais que existem na solução, avaliando os métodos e variáveis existentes.

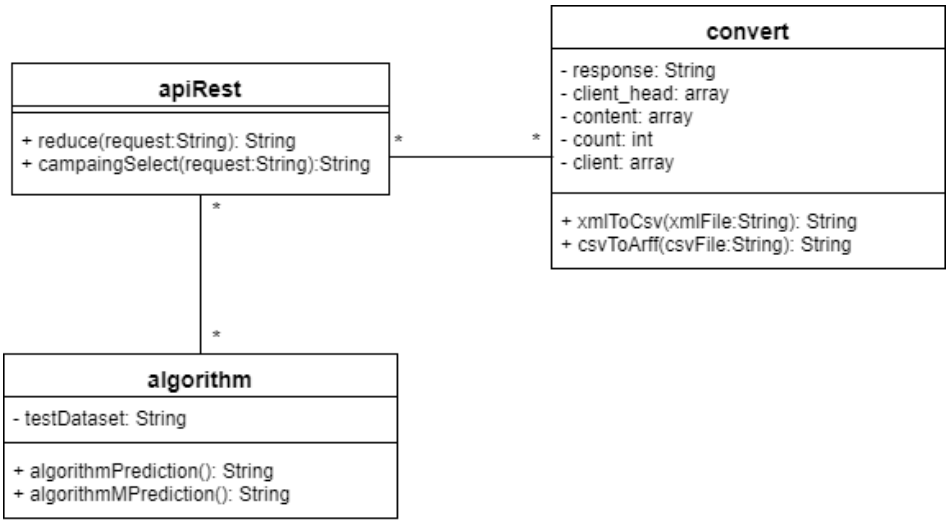

Figura 18 – Diagrama de classes

É possível verificar pelo diagrama de classes que existem três classes que constituem a solução. A classe *apiRest* que tem a responsabilidade de receber *requests* com a informação a ser classificada, convertendo-a utilizado a classe *convert,* para o formato ao qual a classe *algorithm* consiga interpretar e realizar a classificação. Utilizando esses recursos a classe *apiRest* retorna a classificação em formato texto para a plataforma ACM, garantindo a interoperabilidade entre sistemas.

Nesta fase será descrito o funcionamento de cada classe que é apresentada no diagrama.

Classe *apiRest*:

- *Reduce:* É o método responsável pela redução do público-alvo;
- *CampaingSelect:* É o método responsável por identificar a campanha mais adequada para um determinado cliente.

Classe *convert:*

- *xmlToCsv:* É responsável por realizar a conversão dos dados de XML para CSV;
- *csvToArff:* É responsável por realizar a conversão dos dados de CSV para ARFF.

Classe *algorithm:*

- *algorithmPrediction:* É o algoritmo configurado para fazer a classificação da redução do público-alvo;
- *algorithmMPrediction:* É o algoritmo configurado para fazer a classificação que permite identificar a melhor campanha para um determinado cliente.

3.6.9 Vista de processo

Nesta secção são apresentadas as ligações dos componentes com as entidades externas. A arquitetura é referenciada na **Figura 19** que indica as ligações existentes entre os componentes.

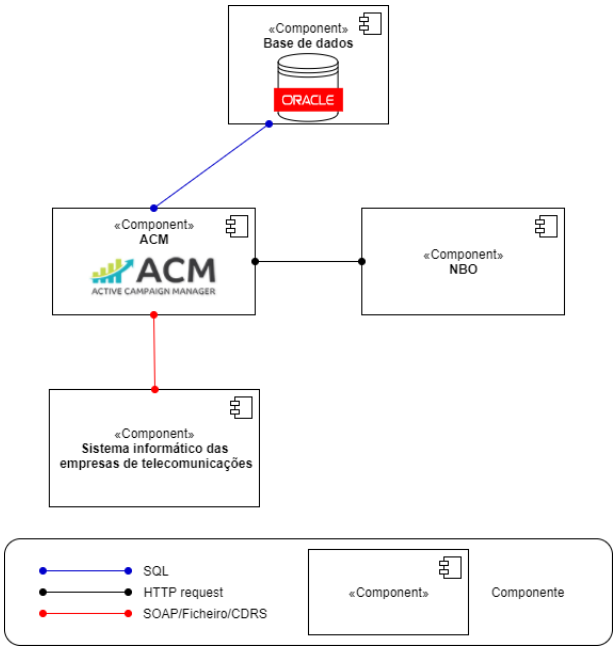

Figura 19 - Diagrama de componente conector

Como podemos verificar o NBO consegue obter informação a partir de *requests* criados pelo ACM. Assim cada vez que um cliente vai entrar em uma determinada campanha é classificado e avaliado perante a sua suscetibilidade.

Na **Figura 20** é possível analisar claramente as ligações existentes entre camadas e os respetivos componentes associados. Percebemos os níveis que existem no ACM em conjunto com o NBO.

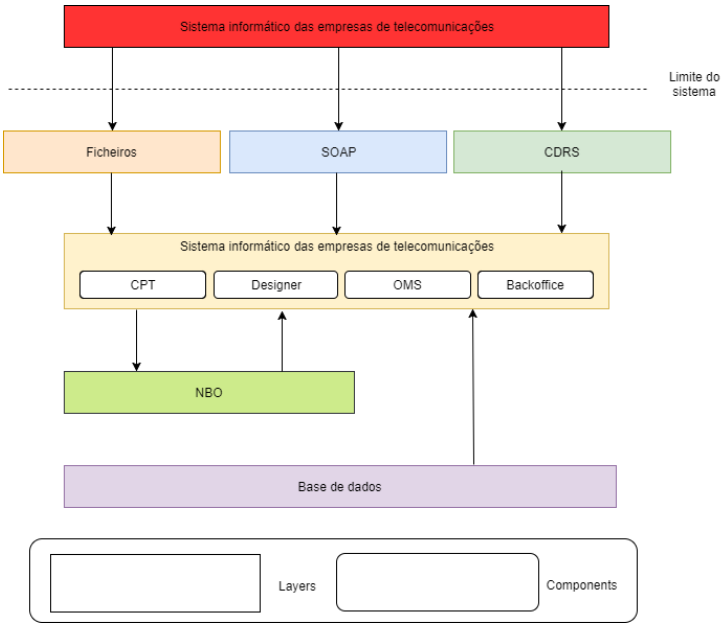

Figura 20 - Diagrama de camadas

3.6.10 Vista de desenvolvimento

A arquitetura do NBO é constituída por um conjunto de blocos responsáveis pelo seu correto funcionamento. Esses blocos estão representados na **Figura 21**.

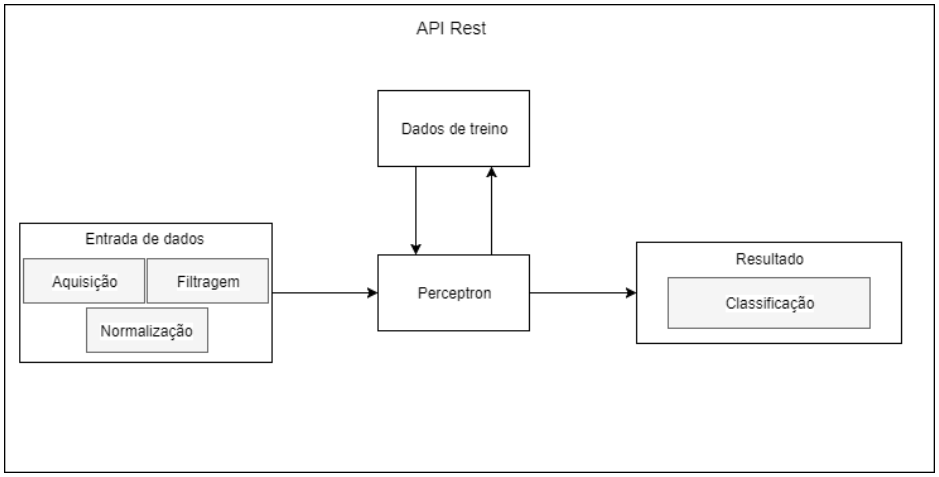

Figura 21 – Arquitetura de blocos do NBO

O NBO é composto por quatro blocos principais, que por sua vez podem estar associados a blocos mais pequenos. Os blocos são:

- **API REST:** É responsável pela comunicação entre o NBO e o ACM. É utilizada uma *framework* em *Python* denominada *Flask* para implementar a API;
- **Entrada de dados**: A entrada dos dados é composta por três passos para possibilitar a realização da classificação. O primeiro é a forma como os dados são **adquiridos** pelo sistema, que são recebidos com os *request* realizados pela plataforma ACM. Em seguida é feita a **filtragem** dos dados sendo apenas escolhidos os campos que têm significado para este processo, assim como exclusão de valores nulos que possam existir. Por fim, é feita a **normalização** dos resultados para que os dados não sejam discrepantes;
- **Dados de treino**: São dados previamente escolhidos que vão servir de referência para o algoritmo *Perceptron* realizar classificações dos dados. Estes dados devem conter cerca de 50% dos registos de clientes que tenham aceitado campanhas semelhantes. Os restantes 50% devem ser registos de clientes que tenham rejeitado essas mesmas campanhas. A renovação destes registos deve acontecer anualmente;
- **Perceptron**: O algoritmo é responsável por receber os dados de teste e utilizá-los como referência para a classificação. Este foi o algoritmo escolhido devido à eficácia que apresentou nos teste realizados com diferentes algoritmos e por não precisar de grandes configurações para apresentar bons resultados. Para a sua implementação será utilizada a *framework Weka* desenvolvida para o *Python*;
- **Resultados**: Os resultados gerados pelo algoritmo podem ser em forma de classificação. A classificação atribui um valor "1" ou "0" a cada registo, sendo o "1" suscetível e o "0" não suscetível. Contudo, quando se trata de uma classificação com várias classes (ex: número de diferentes campanhas) os resultados podem ser de "0" a N, sendo N o número total de campanhas. Esta última classificação é utilizada para identificar qual a campanha mais adequada para um determinado cliente.

3.6.11 Vista física

A **Figura 22** apresenta a alocação ou arquitetura de desenvolvimento necessária para o NBO se apresentar em funcionamento. É composto por um servidor (servidor partilhado) a funcionar com uma distribuição *Red Hat Linux*. Os componentes instalados no servidor são as bibliotecas pertencentes ao *Python* e o servidor *REST*.

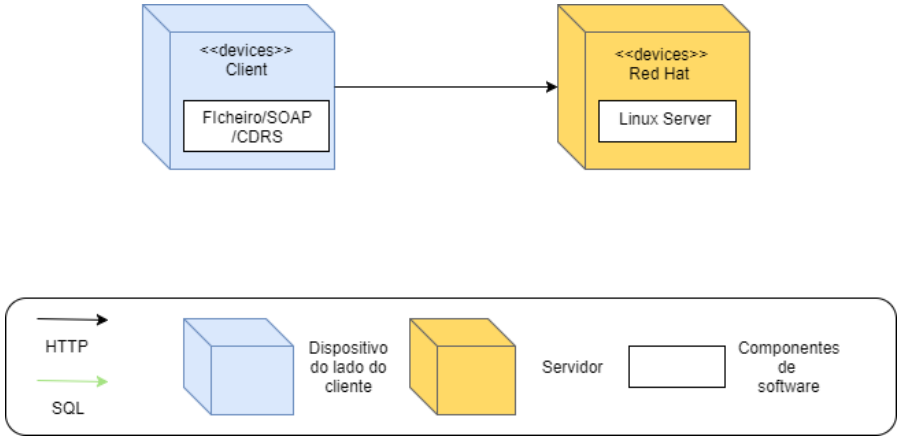

Figura 22 - Arquitetura de implementação

3.7 Metodologia utilizada

Nesta dissertação a metodologia utilizada para desenvolver o projeto foi a metodologia *waterfall*.

O *waterfall* é uma metodologia constituída por cinco etapas onde todas elas são executadas sequencialmente. É uma abordagem formal e com um nível de detalhe bastante elevado. Devido a esse fator neste modelo só se pode avançar para a próxima etapa quando a anterior estiver 100% fechada, dificultando possíveis alterações das etapas anteriores. As etapas pertencentes a esta metodologia são: recolha de requisitos, planeamento, implementação, testes e manutenção [27].

Na **Figura 23** é demonstrado o fluxo de trabalho que deve ser realizado nesta metodologia.

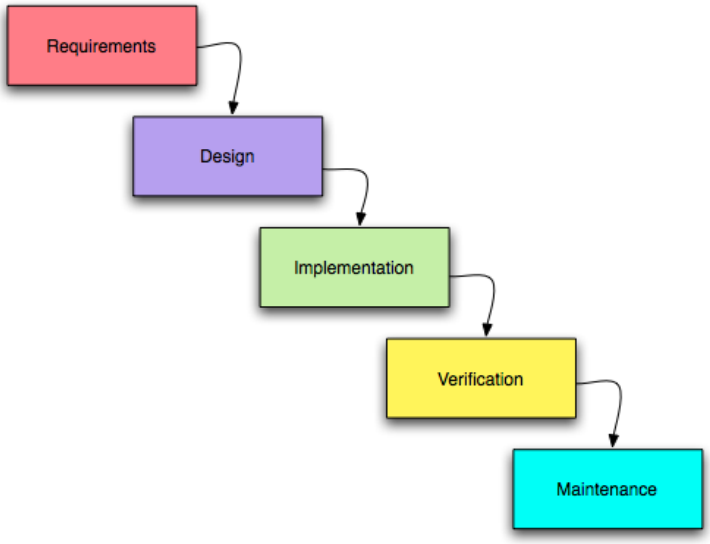

Figura 23 - Metodologia *Waterfall* [28]

A escolha desta metodologia está de acordo com a ideologia de trabalho da empresa. Para uniformizar os processos a empresa adotou a mesma metodologia em todos os departamentos, para que quando é necessário consultar trabalho realizado por outras equipas, o colaborador se sinta confortável com a forma como foi realizado o projeto e com a organização da documentação. Para seguir essa linha de raciocínio adotei a metodologia existente. Contudo, é de salientar que a metodologia assenta perfeitamente no âmbito deste estágio.

Capítulo 4
Trabalho realizado

Nesta secção é descrito o desenvolvimento do projeto, identificando e explicando as etapas que estão subjacentes ao desenvolvimento da solução. São identificados os pontos principais e as decisões tomadas para resolver dificuldades que foram surgindo.

4.1 Obtenção da informação

Os dados utilizados para realizar o projeto foram retirados da plataforma ACM, tendo como referência dados provenientes das duas empresas de telecomunicações. Os dados fornecidos encontram-se em formato CSV, porque ao realizar operações diretamente na base de dados podia colocar em causa o bom funcionamento da plataforma. A utilização dos dados neste formato implicou um tratamento extra da informação, sendo necessário juntar e tratar a informação proveniente de vários ficheiros diferentes. Assim, foram utilizadas ferramentas como o *Pentaho*, *Excel* e *scripts* em *Python* que contêm múltiplos recursos para fazer a manipulação e tratamento de informação.

As informações fornecidas são constituídas por trezentos e dois campos, proveniente das informações pessoais dos clientes e métricas das suas atividades na rede. Esta informação foi selecionada com base no conhecimento da equipa de BI e dos objetivos que foram selecionados para este projeto. Portanto, foram escolhidos quarenta e quatro atributos, sendo dois deles a identificação do cliente e as restantes métricas de medição do consumo de múltiplos serviços.

Para este projeto foram utilizadas cinco campanhas promocionais, sendo elas:

- **Campanha 1 (CAM 1)**: Campanha de incentivo à recarga, que apresenta um total de clientes de 414 623. A campanha apresenta 3 830 clientes que aderiram a campanha e 407 793 que não aderiram;
- **Campanha 29 (CAM 29)**: Campanha de incentivo à recarga, que apresenta um total de clientes de 73 515. A campanha apresenta 4 207 clientes que aderiram a campanha e 69 308 que não aderiram;
- **Campanha 32 (CAM 32)**: Campanha de incentivo ao consumo, que apresenta um total de clientes de 120 006. A campanha apresenta 1 509 clientes que aderiram a campanha e 118 497 que não aderiram;
- **Campanha 75 (CAM 75)**: Campanha de incentivo à recarga, que apresenta um total de clientes de 50 271. A campanha apresenta 1 945 clientes que aderiram a campanha e 48 326 que não aderiram;
- **Campanha 86 (CAM 87)**: Campanha de incentivo à recarga, que apresenta um total de clientes de 50 271. A campanha apresenta 2 912 clientes que aderiram a campanha e 378 667 que não aderiram.

Como podemos verificar pela lista apresentada a maioria das campanhas são do tipo incentivo à recarga, o que limitou os testes do *software* desenvolvido. Contudo, foi possível perceber com os dois tipos de campanhas existentes o comportamento do algoritmo.

Os atributos que foram escolhidos para o desenvolvimento deste *software* são apresentados e descritos na **Tabela 13**.

ID (BI)	ID (ACM)	Caraterística	Formato ACM	Casas Decimais (Números) / Formato (Datas)	Valor por omissão	Categoria
1002	618234131	Família tarifário	Número	0	-1	Dados Pessoais/Cartão
1008	140573392	Segmento estratégico	Número	0	-1	Dados Pessoais/Cartão
1009	1387915571	Segmento estratégico - 1M	Número	0	-1	Dados Pessoais/Cartão
1019	1645875847	Escalão do segmento sensibilidade ao saldo cliente	Texto		N/E	Comportamento (diário)
1021	105568781	Escalão valor - 1M	Número	0	-1	Rentabilidade/Valor
1023	312023487	Tarifário	Número	0	-1	Dados Pessoais/Cartão
1050	1559425004	Valor mínimo carregado - 6M	Número	2	0	Comportamento (mensal)
1051	824192102	Valor máximo carregado - 6M	Número	2	0	Comportamento (mensal)
1053	32052439	Frequência de carregamentos por mês	Número	2	0	Comportamento (diário)
1085	1262629232	Escalão valor	Número	0	0	Rentabilidade/Valor
1087	1836131023	Escalão consumo	Número	0	0	Rentabilidade/Valor
1088	1649938488	Escalão consumo - 1M	Número	0	0	Rentabilidade/Valor
1095	169359089	Dur seg TOR voz	Número	0	0	Tráfego Originado - voz
1096	1813885812	Val TOR voz (sem IVA)	Número	2	0	Tráfego Originado - voz
1102	1191552365	Dur seg TOR voz ORM	Número	0	0	Tráfego Originado - voz
1103	1633990815	Val TOR voz ORM (sem IVA)	Número	2	0	Tráfego Originado - voz
1109	923972723	Dur seg TOR voz RF	Número	0	0	Tráfego Originado - voz
1110	1832150901	Val TOR voz RF (sem IVA)	Número	2	0	Tráfego Originado - voz
1144	1190410174	Qtd TOR SMS	Número	0	0	Tráfego Originado - SMS
1145	1981828049	Val TOR SMS (sem IVA)	Número	2	0	Tráfego Originado - SMS
1162	1012840657	Antiguidade do cartão	Número	0	0	Dados Pessoais/Cartão
1222	1549161713	Mercado	Número	0	0	Dados Pessoais/Cartão
1223	1152616251	Definição de mercado	Número	0	0	Dados Pessoais/Cartão
1224	651780040	Segmento de mercado	Número	0	-1	Dados Pessoais/Cartão

1225	627106443	Qtd de carregamentos - 6M	Número	0	0	Comportamento (diário)
1226	881071262	Média de dias entre carregamentos - 6M	Número	2	0	Comportamento (diário)
1227	1636216783	Média do valor carregado - 6M	Número	2	0	Comportamento (diário)
1234	764286440	Categoria cartão	Texto		-1	Dados Pessoais/Cartão
1244	1054619596	Valor mínimo carregado - 3M	Número	2	0	Comportamento (mensal)
1245	1769092783	Valor máximo carregado - 3M	Número	2	0	Comportamento (mensal)
1246	2189966	Nr dias entre carregamentos - 3M	Número	2	0	Comportamento (mensal)
1247	1641362836	Média do valor carregado por mês - 3M	Número	2	0	Comportamento (mensal)
1248	1742267329	Saldo médio antes de recarga - 3M	Número	2	0	Comportamento (mensal)
1251	258317398	ARPU - 3M	Número	2	0	Rentabilidade/Valor
1252	64002313	ARPU - 1M	Número	2	0	Rentabilidade/Valor
1254	1104655276	Val acumulado de recarga	Número	2	0	Rentabilidade/Valor
1259	2140967126	Val TOR voz onnet - 3M	Número	2	0	Tráfego Originado - voz
1264	1188036398	Val TOR voz offnet ORM - 3M	Número	2	0	Tráfego Originado - voz
1265	1495254132	Val TOR voz offnet RF - 3M	Número	2	0	Tráfego Originado - voz

Tabela 13 - Atributos selecionados

Na tabela podemos verificar que existem sete campos identificados, sendo eles:

- **ID (BI)**: É denominada por referência externa, facilitando a identificação dos atributos para aplicações externas ao ACM;
- **ID (ACM)**: É denominado por referência interna, que é utilizada para realizar as operações internas na plataforma;
- **Característica**: Definição das características existentes;
- **Formato ACM**: Tipo de dados de cada atributo;
- **Casas decimais**: Número de casas decimais utilizadas para os tipos de dados numéricos;
- **Valor por omissão**: São valores previamente definidos para serem atribuídos quando não existem inserções por parte do utilizador;
- **Categorias**: Identificação dos diferentes tipos de informação que estão associadas a cada atributo existente.

Para a obtenção dos atributos num único ficheiro foi utilizado um esquema no *Pentaoh*, permitindo assim obter os casos de testes necessários para o desenvolvimento do projeto. Esta ferramenta permite criar fluxogramas com pontos de ação, que realizam transformações interativas da informação. Isso faz com que o utilizador tenha uma perceção visual em que fase se encontram os dados, como também facilita a manutenção do

programa. Para perceberem como foi realizada a transformação dos dados, o diagrama que foi utilizado no *Pentaoh* pode ser consultado na **Figura 24**.

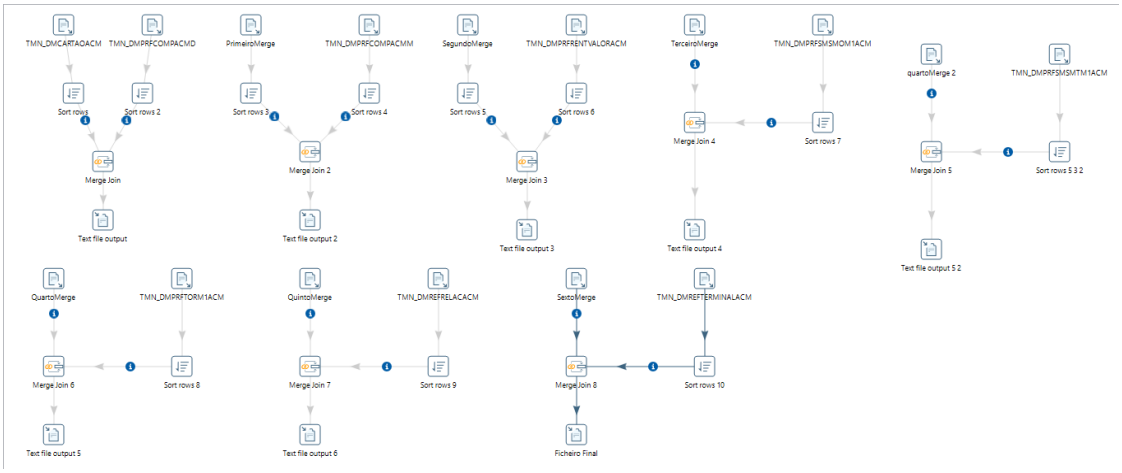

Figura 24 - Fluxos contidos no projeto do *Pentaoh*

É possível verificar o processo de agrupamento da informação dos diferentes ficheiros CSV. Em cada fluxo apresentado é tratado um ficheiro. Assim assegura-se que a informação é fidedigna e coesa em cada etapa, possibilitando a mitigação de erros que possam surgir no decorrer da transformação e junção dos ficheiros. No último fluxo é obtido o ficheiro necessário para se retirar os atributos pertencentes a cada utilizador.

4.2 Tratamento dos dados

As informações provenientes das duas fontes passaram pelo mesmo processo de tratamento de dados, apresentando as seguintes fases:

- Análise e tratamento de ausência de valores nos campos selecionados;
- Averiguação se os valores *default* estão a ser atribuídos corretamente pela plataforma;
- Modificação de campos de texto para campos numéricos, facilitando a classificação dos algoritmos de deteção de padrões;
- Verificação de *outliers* e classificação da sua severidade;
- Substituição dos *outliers* caso se verifique que não estão enquadrados com o campo pertencente.

4.2.1 Tratamento dos campos com ausência de valores

Foram percorridos os registos de cada utilizador existente e substituídos os campos vazios por o valor *default* correspondente ao campo em avaliação. Permitindo, que o *software* consiga trabalhar sem anomalias e apresente índices de precisão satisfatórios. Uma das estratégias que se podia adotar nesta fase seria retirar os registos que continham valores em falta. Contudo, o objetivo deste projeto é conseguir maximizar o número de clientes que irão aderir a uma determinada campanha, deixando assim subjacente que esta estratégia não poderia ser utilizada neste contexto.

4.2.2 Modificação de campos de texto para numéricos

Com o intuito de aumentar o desempenho dos algoritmos de deteção de padrões, são transformados os valores dos atributos de texto em numéricos. Assim permite que os campos sejam de fácil leitura para o algoritmo e assegura compatibilidade com os diferentes algoritmos testados. Para fazer a modificação desses campos foi necessário relacionar elementos numéricos com elementos textuais (ex:"baixo" = 1,"medio" = 2,"alto" = 3).

4.2.3 Identificação e classificação de outliers

Para identificar e classificar se os *outliers* eram legítimos ou não, foi utilizado o *Weka* para organizar e visualizar os dados. Permitindo que fosse possível visualizar a distribuição dos dados e os valores admissíveis para um determinado atributo. Nos casos dos valores não serem admissíveis eram substituídos pelos valores *default*.

4.3 Escolha do conjunto de treino

Para obter os melhores resultados foi necessário entender quais as características que um cliente deveria apresentar para ser integrado no conjunto de teste. Assim, foi importante analisar um conjunto de campanhas para identificar quais os clientes que deveriam ou não ser considerados adequados. Em seguida é apresentada a forma como os tipos de campanhas influenciam o funcionamento do algoritmo e como foram avaliadas as adesões, utilizando como recurso aos ciclos de vida da campanha.

4.3.1 Tipos de campanhas

As campanhas lançadas no ACM são sempre associadas a um determinado tipo de campanha. Esse tipo vai influenciar a construção da campanha, que por sua vez influencia na escolha dos clientes que integram o conjunto de teste. Isso significa que o algoritmo deve ser capaz, com o histórico de uma campanha do mesmo tipo, determinar os possíveis clientes que aceitarão a campanha. Assim sendo, foi realizada uma separação de campanhas pelo seu tipo, sendo os testes realizados em torno deste conceito. Portanto, para uma nova campanha do tipo consumo é utilizada uma campanha com histórico do mesmo tipo para constituir o conjunto de treino. É possível assim assegurar a semelhança entre campanhas.

4.3.2 Análise do ciclo de vida das campanhas

No ACM é possível encontrar diversas campanhas que têm o mesmo objetivo, mas comportam-se de forma diferente. Para identificar os melhores dados foi analisado o comportamento da campanha e identificadas as melhores formas de retirar os resultados finais das campanhas que cada cliente possui. Para se perceber o processo utilizado é feita uma descrição das fases realizadas para as campanhas CAM 87 e CAM 29.

A campanha CAM 87 é bastante simples, tendo como foco somente perceber o processo que é utilizado para identificar quais os clientes que aderiram a uma determinada campanha. Na **Figura 25** é representado o ciclo de vida da campanha.

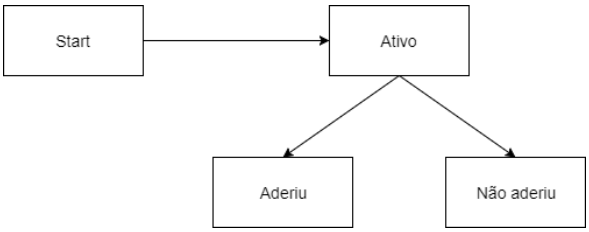

Figura 25 - Ciclo de vida da campanha CAM 87

É possível verificar que a campanha começa no estado *Start* e passa para o estado ativo depois de ser enviado o incentivo para os telemóveis dos clientes na forma de SMS ou chamada de voz. Caso a pessoa aceite a campanha passa para o estado "Aderiu", caso não aceite passa para o estado "Não aderiu". Para os clientes que estão no estado de "Aderiu" é lhes atribuído o número um. Aos restantes é atribuído o número zero.

Esta campanha é bastante simples e fácil de classificar. Contudo, quando se trata de campanhas mais complexas onde podem existir várias interações, é importante avaliar em que situações podemos considerar sucesso ao não. Para se compreender melhor o problema é apresentado um exemplo mais complexo na **Figura 26** que representa a campanha CAM 29.

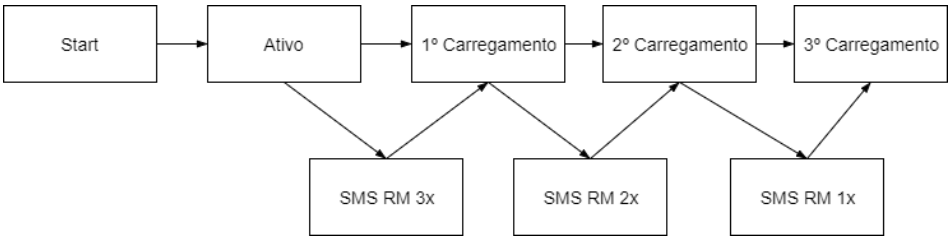

Figura 26 - Ciclo de vida da campanha CAM 29

Nesta campanha surgiu a necessidade de realizar uma análise dos estados para identificar onde fazia mais sentido considerar que o cliente tinha aderido ou não à campanha. Do *Start* para o ativo é enviado o incentivo para o utilizador. Aqui existem duas opções possíveis, carregar e receber o bónus ou simplesmente não carregar. Quando o cliente não carrega por um período de tempo predefinido pelo utilizador é enviada novamente a mensagem para relembrar os utilizadores. Quando carrega passa para o estado de "1º carregamento", podendo fazer o mesmo processo mais duas vezes. Neste caso a adesão do cliente não é clara visto que ele pode aderir duas vezes à campanha e não realizar o último carregamento. Portanto, foi considerado que o cliente adere à campanha quando entra no primeiro estado de benefício, ou seja, no estado "1º Carregamento".

Nesta fase é importante que haja uma uniformidade de critério para manter os resultados coerentes. Quanto mais uniforme forem as escolhas melhor o algoritmo irá conseguir classificar.

4.3.3 Organização dos dados de treino

Para realizar as classificações é criado para cada tipo de campanha um conjunto de treino. Sendo assim é possível atribuir o valor zero para os que não têm perfil para aderir à campanha e o valor um para os que têm. Este processo é utilizado sempre que o objetivo for conseguir perceber se um determinado cliente tem perfil para uma determinada campanha. Quando o objetivo é perceber qual a campanha adequada para o perfil do utilizador é utilizado um conjunto de treino onde se encontram as informações de todos os tipos de campanhas existentes, identificadas por um número interior. O classificador quando quer

atribuir qual o tipo de campanha adequada para o utilizador é emitido o número associado a esse mesmo tipo.

Portanto, para o algoritmo classificar os clientes necessita de um conjunto de treino para cada tipo de campanha e um conjunto de treino constituído pela informação de todos os conjuntos criados dos diferentes tipos de campanhas.

4.4 Algoritmo e resultados

Para realizar este projeto foram realizados vários testes com diferentes algoritmos de *Machine Learning* com o objetivo de identificar o algoritmo mais adequado para o problema. Isso levou à realização de uma vasta gama de testes que podem ser consultados no **Apêndice A**. Contudo, nesta secção é somente explicado o algoritmo escolhido para o desenvolvimento do projeto, analisando e descrevendo os resultados obtidos.

O *Peceptron* foi o algoritmo implementado na solução, devido aos resultados satisfatórios obtidos que vão ao encontro dos objetivos identificados na secção dos requisitos funcionais, e pelo facto de com o mesmo algoritmo se conseguir satisfazer os requisitos funcionais. Logo, para cada requisito será apresentada a solução encontrada.

4.4.1 Redução do público-alvo

Inicialmente foram efeituados testes nos dados referentes à CAM 1 para perceber a redução do público-alvo e identificar qual a percentagem de perda que se iria obter na sua execução. Assim, foram escolhidos seis mil registos de clientes para serem utilizados como dados de treino, sendo que três mil eram clientes que aderiram à campanha e três mil que não aderiram à campanha. É pretendido assim balancear o número de clientes que aderiram ou não aderiram à campanha, mantendo a coerência dos dados. Como a percentagem de adesões foi muito pequena restaram poucos dados para serem utilizados no público-alvo. Contudo, este é um cenário real que será encontrado no ambiente de produção, o que leva a que este contratempo tenha de ser levado em atenção, e consequentemente testado.

Em seguida é apresentada uma tabela com o resultado do teste realizado e uma breve explicação de cada atributo da **Tabela 14**.

#SUT	#INT	#SUPA	# INPA	#EFR	#NPA	#TRPA	#SP	#TP	#EFA
3000	3000	830	407793	0,20%	153602	62%	53	6%	0.50%

Tabela 14 - Resultados obtidos

#SUT- Sucessos no conjunto de teste;
#INT – Insucesso no conjunto de testes;
#SUPA – Número de sucessos no público-alvo;
#INPA – Número de insucesso no público-alvo;
#EFR – Efetividade real. É obtida em detrimento do cálculo dos sucessos sobre os insucessos;
#NPA – Novo público-alvo;
#TRPA – Taxa de redução do público-alvo. É obtido da percentagem da diferença do total com o #NPA;
#SP – Número de sucessos perdidos;
#TP – Taxa de perda referente a #SP;
#EFA – Taxa da eficácia do algoritmo.

Verifica-se que foram utilizados três mil registos de sucesso e o mesmo número de insucessos. Os dados de sucesso no público-alvo eram de oitocentos e trinta e os de insucesso eram os restantes. Com o algoritmo foi possível reduzir o público-alvo de 62% com uma perda de clientes de 6%. Isso faz com que seja possível atingir o objetivo de

reduzir o número de mensagens enviadas aos utilizadores que não estão interessados em mais de 50% e incentivar mais de 90% dos utilizadores que iriam aceitar a campanha.

4.4.2 Campanha mais adequada para um determinado cliente

Com o objetivo de conseguir identificar qual a campanha mais adequada para um determinado cliente, foi efetuada uma alteração no conjunto de treino para que seja possível classificar múltiplas campanhas promocionais.

Mais uma vez foram realizados vários testes com vários algoritmos que podem ser consultados no **Apêndice A**. O *Perceptron* teve duas fases de testes, sendo elas:

- **Utilização da informação em bruto**: São utilizados todos os atributos existentes no conjunto de dados.
- **Utilização de informação adequada ao algoritmo**: É utilizado um algoritmo que permite identificar os atributos mais relevantes para o *Perceptron*.

Para estes testes foram utilizados os dados de duas campanhas. As campanhas são a CAM 75 e CAM 32. Cada uma delas foi dividida em dados de treino e dados de teste, sendo a divisão feita com uma separação de 66%. Aos dados de treino da CAM 75 foi atribuído um novo campo chamado resultado com o valor um e aos dados da CAM 32 foi atribuído o mesmo campo com o valor dois associado. Aos dados aos quais os clientes não aderiram a nenhuma das campanhas ficaram associados ao valor zero.

Com esta distribuição dos dados foi realizada uma execução do algoritmo com todos os atributos existentes. Os resultados estão na **Tabela 15** contidos em uma matriz de confusão representam os valores que foram atribuídos pelo algoritmo.

	Sem campanha	**CAM 32**	**CAM 75**
Sem campanha	151	52	84
CAM 32	61	191	27
CAM 75	45	18	187

Tabela 15 - Resultados do algoritmo com os atributos todos

A tabela indica que os resultados do algoritmo apresentam perdas significativas de clientes que aderiram à campanha. Isso faz com que houvesse a necessidade de introduzir uma nova técnica para aumentar a precisão do algoritmo. Portanto, foi utilizado um algoritmo que identifica quais os atributos que têm maior influência nos resultados do *Perceptron*.

O algoritmo usado foi o *InfoGainAttributeEval* que se encontra na ferramenta do *Weka*. Esse algoritmo permite obter a probabilidade da influência que cada atributo apresenta na classificação do algoritmo de *Machine Learning*. Assim, é possível aumentar a percentagem de acerto, traduzindo-se em diferenças significativas demonstradas na **Tabela 16**.

	Sem campanha	**CAM 32**	**CAM 75**
Sem campanha	93	126	68
CAM 32	2	248	16
CAM 75	12	27	224

Tabela 16 - Resultados do algoritmo com atributos selecionados

Com estes resultados é possível verificar que a perda é pequena comparada ao número existente de clientes que foram bem classificados. Num ambiente real, o número de perdas vai andar próximo dos 15%, portanto quando os clientes não se adaptam a uma determinada campanha podem ser sempre encaminhados para uma campanha que esteja de acordo com o seu perfil.

Estes resultados foram obtidos a partir dos seguintes atributos presentes na **Tabela 17**.

ID (BI)	ID (ACM)	Caraterística	Formato ACM	Casas Decimais (Números) / Formato (Datas)	Valor por omissão	Categoria
1234	764286440	Categoria cartão	Texto		-1	Dados Pessoais/Cartão
1227	1636216783	Média do valor carregado - 6M	Número	2	0	Comportamento (diário)
1251	258317398	ARPU - 3M	Número	2	0	Rentabilidade/Valor
1254	1104655276	Val acumulado de recarga	Número	2	0	Rentabilidade/Valor
1144	1190410174	Qtd TOR SMS	Número	0	0	Tráfego Originado - SMS
1095	169359089	Dur seg TOR voz	Número	0	0	Tráfego Originado - voz

Tabela 17 - Atributos com melhor rendimento para o algoritmo

4.5 Interoperabilidade entre sistemas

Para garantir a interoperabilidade entre sistemas foi implementado uma *API REST* que permite que vários sistemas se consigam comunicar com a solução. Assim existem dois URIs que permitem invocar as funcionalidades implementadas. As informações da *API REST* estão presentes na **Tabela 18**.

Recurso	URI	HTTP método suportado
ACM	/reduceTA	Post
	/campaign/select	Post

Tabela 18 - Informações sobre a *API REST*

Aqui são representadas as duas funcionalidades que podem ser executadas pela *API*. A chamada "/reduceTA" permite ao ACM fazer a restrição do público-alvo à medida que os clientes entrem na campanha. Então o único processo que têm de realizar é formular um pedido *HTTP* com informação XML dentro do *body* que esteja organizado da seguinte forma:

```
<?xml version="1.0"?>
<data>
        <cliente>
                <tipo_camp>recarga</tipo_camp>
                <avg_amt_rec_6m> 10.714286 </avg_amt_rec_6m>
                <arpu_3m>17.341463</arpu_3m>
                <val_acumulado_rec>210</val_acumulado_rec>
                <qtd_tor_sms>574</qtd_tor_sms>
                <dur_seg_tor_voz >2189</dur_seg_tor_voz >
        </cliente>
</data>
```

Os valores são de exemplo, contudo a organização tem de ser esta. Inicialmente suportava para vários clientes ao mesmo tempo, mas pelo facto do ACM criar diferentes processos para os diferentes clientes esta solução foi a implementada. O URI "/campaign/select" possibilita que cada cliente seja avaliado pela sua suscetibilidade pelas diferentes campanhas. Contudo, o *request* deve conter as mesmas informações. O XML é o espelho dos atributos considerados relevantes para o algoritmo *Perceptron*.

Com a evolução desta aplicação é esperado que o número de funcionalidades aumente, aumentando consequentemente os URI. Assim, além de possibilitar a interoperabilidade entre sistemas é possível afirmar que a manutenabilidade do *software* está assegurada.

4.6 Método de avaliação do algoritmo

A utilização do grupo de controlo somente na entrada da campanha torna impossível medir o impacto do NBO no lançamento de uma campanha. Assim sendo, foi preciso desenvolver um novo método de avaliação que conseguisse abranger os dois sistemas. Para resolver este problema foi necessário acrescentar um novo grupo de controlo para possibilitar a medição, tanto da campanha como do NBO. O diagrama apresentado demostra o processo de medição, com os valores percentuais escolhidos para este exemplo.

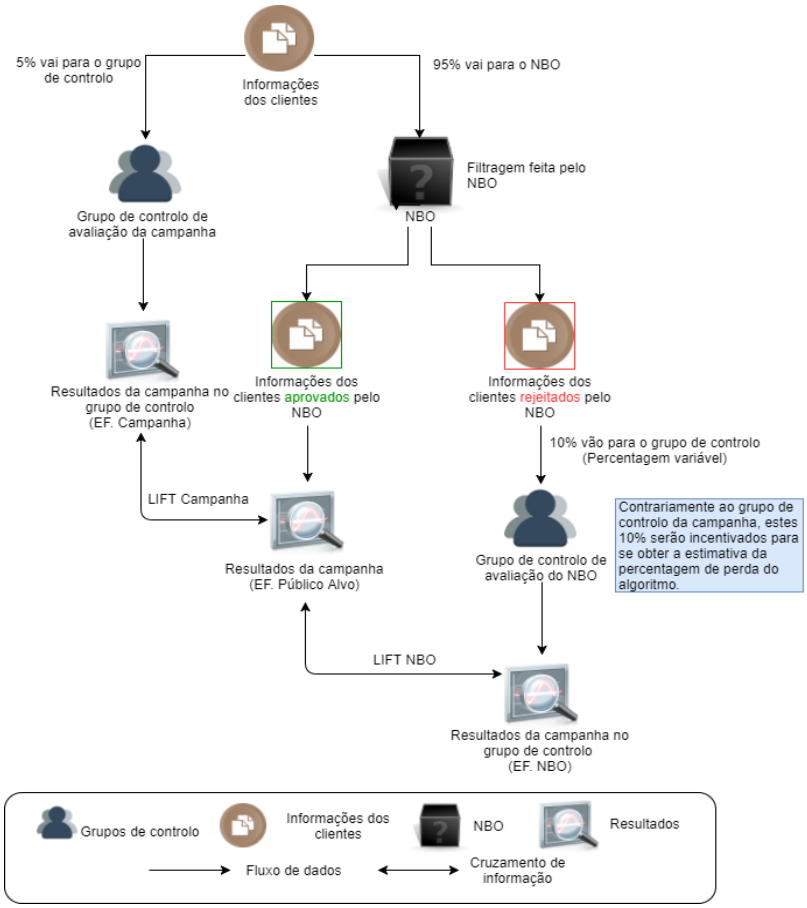

Figura 27 - Diagrama de teste de eficácia do ACM e NBO

Inicialmente a informação é dividida em dois grupos, 10% (valor que pode ser alterado pelo administrador do ACM) vai para o grupo de controlo e os restantes para o público-alvo. Em seguida o NBO faz a classificação e cria dois novos grupos. O grupo dos clientes que vão ser incentivados e os que não vão ser incentivados. No grupo dos clientes que não vão ser incentivados são retirados 10% para grupo de controlo do NBO. Contudo, este grupo de controlo vai ser um pouco diferente do criado anteriormente. Os clientes deste grupo irão ser incentivados para que se consiga obter informação estimada de qual a percentagem de erro que o algoritmo está a ter. Caso existam muitos clientes a aderir à campanha indica que o algoritmo falhou na escolha dos clientes rejeitados.

Com estes dois grupos de controlo e com o resultado do algoritmo é possível obter todos os índices estatísticos necessários para avaliar a eficácia do algoritmo e do ACM na campanha lançada.

Capítulo 5
Testes

Nesta secção são descritos um conjunto de testes que permitem avaliar a viabilidade da solução desenvolvida e a identificação de possíveis falhas no seu comportamento. Os testes servem fundamentalmente para verificar se os requisitos funcionais e não-funcionais estão implementados na solução final.

5.1 Testes de aceitação

O uso dos testes de aceitação tem como objetivo verificar a implementação dos requisitos funcionais e a conformidade dos resultados com o esperado. É expectável que no final da análise se consiga obter uma tabela com a indicação de quais os requisitos que foram implementados e os que não foram. Na realização dos testes, devido ao número reduzido de requisitos, os testes foram realizados manualmente identificando somente duas variáveis no processo. Se o requisito está implementado ou não na solução e se o seu comportamento está de acordo com o esperado. Se o requisito não estiver implementado a avaliação do comportamento vai ser "Não Aplicável".

.

Funcionalidades	**Implementado**	**Comportamento Esperado**
Ativa o serviço proveniente do NBO.	Sim	Não
Desativa o serviço proveniente do NBO.	Sim	Não
Identificar clientes suscetíveis a uma determinada campanha.	Sim	Sim
Identificar qual a campanha adequada para um ou mais clientes.	Sim	Sim
Requisitar lista de clientes suscetíveis à campanha.	Sim	Sim

Tabela 19 - Tabela de aceitação referente ao projeto

Como podemos verificar na tabela apresentada, todos os requisitos foram implementados. Contudo, nem tudo como estava planeado. Os primeiros dois requisitos deviam ter sido implementados tanto na plataforma, onde seria esperado um conjunto de opções que possibilitasse a ativação e desativação do serviço, e na aplicação onde se deveria manter o programa em *standby* caso estivesse desativo e em funcionamento caso estivesse ativo. A parte referente ao programa está implementada, contudo a parte do ACM a implementação não foi possível, devido às mudanças poderem ter um impacto negativo no *software*. Só a equipa de *frontend* deve fazer alterações no *layout* da aplicação.

Os restantes requisitos foram implementados de acordo com o esperado.

5.2 Testes de interoperabilidade

Para garantir a interoperabilidade entre aplicações é utilizado uma *API REST* que permite que os *softwares* com um cliente implementado desta tecnologia possam usufruir dos serviços prestados. Assim, são realizados *requests* que estão sujeitos ao tempo de processamento do algoritmo, respeitando a ordem dos pedidos realizados.

Para verificar isso foram executados *requests* num programa em *python* e utlizada uma ferramenta chamada *Postman* que permite realizar *requests* de forma simples e intuitiva. Os *requests* foram respondidos sem haver nenhum tipo de incompatibilidade, sendo as respostas sempre dadas em formato texto.

5.3 Testes de desempenho

Foram realizados testes de desempenho às duas funcionalidades apresentadas na aplicação desenvolvida. Estes testes utilizaram como unidade de medida os segundos, tendo como finalidade perceber qual o tempo de processamento por registo que a aplicação apresenta. Assim, podemos perceber qual a velocidade esperada e se o tempo satisfaz os requisitos. Para este teste foi utilizado o ambiente de testes do ACM que apresenta os recursos de uma máquina utilizada para a sua instalação. A máquina é composta por 4GB de memória *RAM*, processador *dual core*, 30 GB de disco rígido e tem o sistema operativo *Linux Redhat* instalado.

Vários conjuntos de informação foram tratados e classificados permitindo obter a velocidade de processamento de cada registo individualmente. Na **Tabela 20** são apresentados os valores obtidos.

Número de registos testados	Média por registos(Segundos)	Desvio
100	5.0028	0.00028
1000	5.029	0.0029
10000	5.25	0.025
100000	7.6	0.26

Tabela 20 - Tempos de execução do programa

Estes resultados vêm demonstrar um comportamento bastante positivo ao nível do desempenho. Sendo que o objetivo era ter no mínimo duzentos mil registos processados por minuto, o que na realidade acontece é que esses mesmos registos podem ser processados em aproximadamente 15.2 segundos. É possível perceber que este resultado ultrapassa bastante as expectativas, tendo um impacto bastante positivo no projeto.

Apesar do teste aparentar produzir incorretamente, é de relembrar que o mesmo tipo de classificação na linguagem *R* demora aproximadamente três minutos a processar. Portanto, esta é a consequência da eficácia apresentada pela *framework Weka* desenvolvida para *Python*, e não pelo facto dos testes realizados estarem mal medidos ou desenquadrados com a realidade.

5.4 Testes de disponibilidade

Para garantir a disponibilidade foi utilizada um *framework* chamada *flask* para configurar a *API REST*, que permite ter tolerância a falhas na aplicação. Assim, mesmo quando a aplicação desenvolvida apresenta um erro, o algoritmo continua em funcionamento. Isso faz com que o tempo que o sistema se encontra *offline* seja reduzido.

Para executar o teste foi realizado um pedido à aplicação colocando propositadamente um erro. O que aconteceu foi que o programa lançou uma mensagem de erro, mantendo a aplicação funcional.

Capítulo 6
Plano de gestão dos riscos

6.1 Objetivos a serem cumpridos para que o projeto seja um sucesso

Nesta secção são apresentados os mínimos que o projeto tem de atingir para que seja considerado um projeto de sucesso. Este processo permite medir se o projeto considerado está a cumprir com as expectativas da organização.

A próxima lista de pontos define os objetivos críticos estipulados para o projeto e para o processo envolvido:

Objetivos considerados para o sucesso do produto

- O projeto tem de satisfazer os requisitos funcionais até à data final;
- O projeto tem de satisfazer todos os atributos de qualidade/atributos não funcionais até à data final;
- A empresa deve ficar satisfeita com as entregas faseadas e a entrega final;

Objetivos considerados para o sucesso do processo

- A estimativa feita no início do projeto tem de estar semelhante ao que se verifica no final;
- Utilizar a metodologia escolhida inicialmente em todas as fases do projeto.

6.2 Identificação dos riscos

Nesta secção são identificados os riscos que o projeto apresenta. O risco é um evento que pode ocorrer durante o desenvolvimento do projeto, tendo um impacto negativo no seu processo. Os riscos definidos neste projeto podem ser possíveis fontes que podem produzir eventos indesejados ou condições que desencadeiam eventos inesperados. Qualquer um destes pode provocar um impacto negativo com diferentes graus de severidade.

Nas tabelas seguintes são identificados os riscos que afetam o projeto em geral, os ricos para cada etapa do desenvolvimento do projeto e a consequência que cada um dos riscos provoca.

ID	Fonte do risco ou condição	Consequências
1	Problemas de comunicação com a empresa.	O projeto pode não ser aceite pela empresa
2	Inexperiência do desenvolvedor do projeto.	Apresentar pouca produtividade e pouca qualidade, não correspondendo aos objetivos implícitos do estágio.
3	Problemas nas decisões tomadas pelo responsável do projeto.	Pode levar a atrasos nas entregas e incoerência entre a visão da organização e o projeto apresentado.
4	Os processos na fase inicial de desenvolvimento podem ser mais lentos do que o esperado.	O planeamento de tarefas feito pode ter de ser restruturado o que pode levar ao incumprimento das metas estabelecidas.
5	A realização do percurso académico em paralelo.	Introdução de tarefas a meio do projeto reduzindo o tempo do responsável do projeto, influenciando negativamente o plano efetuado

Tabela 21 - Riscos gerais

ID	Fonte do risco ou condição	Consequências
6	Erro na identificação dos requisitos	Significa uma falha do projeto que se pretende desenvolver. A sua reformulação no meio do projeto pode ser impossível.
7	Sistema desenvolvido não apresenta os requisitos a implementados.	Projeto não apresenta todas as funcionalidades necessárias.
8	O cliente não saber o que deseja.	Dificuldade em conseguir estabelecer todos os requisitos no início do projeto.

Tabela 22 – Riscos de requisitos

ID	Fonte do risco ou condição	Consequências
9	A arquitetura não esta adequada aos objetivos do projeto.	Pode aumentar o volume de trabalho, podendo causar atraso na tese.
10	A arquitetura apresenta falhas na sua conceção.	Pode aumentar o volume de trabalho, podendo causar atraso na tese.

Tabela 23 – Riscos da arquitetura

ID	Fonte do risco ou condição	Consequências
11	Má utilização dos recursos numa determinada tarefa.	Pode aumentar o volume de trabalho, podendo causar atraso na tese.
12	Erro na estimação do tamanho do projeto.	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, podendo causar atraso na tese.
13	Erro na estimação dos riscos do projeto.	Pode aumentar o volume de trabalho, podendo causar atraso na tese.
14	A metodologia adotada pode não ser adequada ao projeto.	Pode aumentar o volume de trabalho, podendo causar atraso na tese.

Tabela 24 – Riscos do planeamento

ID	Fonte do risco ou condição	Consequências
15	Demasiado tempo a aprender as tecnologias necessárias para o projeto	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, podendo causar atraso na tese.
16	Tecnologias escolhidas não serem adequadas para o projeto.	Fracasso na implementação dos requisitos e aumento no trabalho a realizar, podendo causar atraso na tese.
17	Dependência de tecnologias desconhecidas para o autor da tese.	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, podendo causar atraso na tese.
18	Falha na integração dos componentes.	Fracasso na implementação dos requisitos e aumento no trabalho a realizar, podendo causar atraso na tese.
19	Diferenças existentes entre o ambiente de testes e a produção.	Fracasso na implementação dos requisitos e aumento no trabalho a realizar, podendo causar atraso na tese.

Tabela 25 – Riscos da implementação

ID	Fonte do risco ou condição	Consequências
20	O ambiente de testes não estar disponível.	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, podendo causar atraso na tese.
21	Muito tempo despendido em testes.	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, podendo causar atraso na tese.
22	Poucos testes efetuados.	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, podendo causar atraso na tese.
23	Dependência de bibliotecas externas para realizar os testes.	Pode baixar a produtividade, qualidade e aumento o volume de trabalho, influenciado negativamente a nota da tese.
24	Erros na criação do plano de testes.	Pode aumentar o volume de trabalho, podendo causar atraso na tese.

Tabela 26 – Riscos da fase de testes

ID	Fonte do risco ou condição	Consequências
25	Falha na integração dos sistemas.	Influência negativa na nota da tese.
26	Falha na entrega final.	A tese deverá ser repetida para o próximo ano letivo.

Tabela 27 – Riscos da entrega do projeto entrega

6.3 Avaliação dos riscos

Nesta secção é realizada a avaliação detalhada dos riscos identificados, utilizando métricas para que seja possível criar medidas de prevenção.

São apresentadas nas tabelas seguintes o ID de cada risco identificado, o impacto que cada risco tem no projeto, a probabilidade de ocorrer e a estimativa de quanto tempo pode demorar para que o risco se manifeste.

Foi decidido pelo gestor do projeto que o impacto poderá ser identificado como catastrófico, crítico e marginal. O catastrófico significa que o projeto não pode recuperar desse risco, crítico significa que será difícil recuperar desse risco e por fim o marginal que facilmente se pode recuperar do risco caso ele ocorra. Para a probabilidade podem ser atribuídos os indicadores altos (> 70%), média (entre 70% e 40%) e baixa (40%>). Para o tempo que demorar a se manifestar podem ser atribuídos o longo (> 2 meses), médio (entre 1 e 2 meses) e curto (< 1 mês).

Na **Tabela 28** são indicados os riscos, apresentado o seu nível de impacto e a sua probabilidade. Na **Tabela 29** é composta pela matriz de confusão para que resume a informação. Utilizando estes dados é possível priorizar os riscos e mitigá-lo.

ID	Impacto	Probabilidade	Tempo
1	Crítico	Média	Longo
2	Catastrófico	Médio	Médio
3	Crítico	Médio	Longo
4	Crítico	Médio	Longo
5	Crítico	Alta	Longo
6	Catastrófico	Média	Média
7	Crítico	Médio	Médio
8	Crítico	Média	Média
9	Crítico	Alta	Média
10	Crítico	Média	Média
11	Catastrófico	Baixa	Longo
12	Crítico	Alta	Longo
13	Crítico	Média	Longo
14	Catastrófico	Baixa	Curto
15	Crítico	Média	Longo
16	Catastrófico	Baixa	Média
17	Crítico	Média	Longo
18	Catastrófico	Baixa	Curto
19	Crítico	Alta	Longo
20	Marginal	Média	Curto
21	Marginal	Baixa	Curto
22	Marginal	Média	Curto
23	Marginal	Alta	Curto
24	Marginal	Média	Curto
25	Catastrófico	Média	Curto
26	Catastrófico	Baixa	Longo

Tabela 28 – Avaliação dos riscos

Impacto \ Probabilidade	Baixa	Média	Alta
Catastrófico	11, 14, 16, 26	2, 6, 25	
Crítico		1, 3, 4, 7, 8, 10, 13, 15, 17	9, 12, 19
Marginal	21	20, 22, 24	23

Tabela 29 – Matriz de exposição ao risco

6.3 Plano de mitigação dos riscos

Nesta secção é apresentada a mitigação dos riscos, com o objetivo de reduzir tanto a probabilidade dos riscos ocorrerem, como o impacto que esses riscos podem ter no projeto. Portanto é criado um plano de contingência que abrange os riscos que estão identificados como catastróficos ou críticos, não se justificando fazer um plano para todos os riscos devido à dimensão do projeto.

Plano de mitigação dos riscos	**Riscos envolvidos**
Realizar várias reuniões com o cliente para falarmos sobre os requisitos funcionais, tecnologias a utilizar e quais as limitações implícitas do projeto.	1,2,3
Haver uma validação de todas as decisões tomadas no projeto para que seja reduzido o tempo de avaliação das decisões.	3,4

Tabela 30 – Gerais

Plano de mitigação dos riscos	**Riscos envolvidos**
Os requisitos devem ser revistos por várias entidades envolvidas para se chegar a um consenso.	5
Devem ser promovidas várias apresentações com a empresa para perceber se está a ser realizado o que é esperado	6,7,8

Tabela 31 – Requisitos

Plano de mitigação dos riscos	**Riscos envolvidos**
Deve-se promover encontros periódicos com os orientadores para definir o âmbito do projeto e analisar o trabalho realizado até ao momento.	9,10

Tabela 32 – Arquitetura

Plano de mitigação dos riscos	**Riscos envolvidos**
Se a estimação não esteja de acordo com o esperado, será feita uma nova estimação tendo em conta o desvio temporal já registado.	12
Se existir algum risco não avaliado ou mal avaliado, o plano de riscos será alterado com o devido registo da alteração.	13
Se a metodologia do projeto não estiver adequada o projeto será restruturado para uma nova. Será registado devidamente a alteração.	14

Tabela 33 – Planeamento

Plano de mitigação dos riscos	**Riscos envolvidos**
Para mitigar os problemas da nova tecnologia, são utilizados tutoriais referidos na documentação das bibliotecas.	15, 17
Se alguma tecnologia não for adequada ao projeto alterando, caso necessário, a arquitetura.	16, 18, 19, 25

Tabela 34 – Implementação

Plano de mitigação dos riscos	**Riscos envolvidos**
Para mitigar os riscos avaliados para a fase de testes do projeto deve ser desenvolvido um plano de testes que consiga abordar a maior parte do projeto.	20, 21, 22, 23, 24

Tabela 35 – Testes

Plano de mitigação dos riscos	Riscos envolvidos
Para mitigar os riscos avaliados para a fase de testes do projeto deve ser desenvolvido um plano de testes que consiga abordar a maior parte do projeto.	20, 21, 22, 23, 24

Tabela 36 – Entrega

Capítulo 7
Plano de Trabalho e Implicações

O planeamento do projeto é a primeira tarefa realizada, e visa a organização das tarefas usando a linha temporal como referência. Na sua realização foi utilizado o diagrama de *Gantt* para uma melhor visualização do calendário de atividades.

7.1 Planeamento do primeiro semestre

A **Tabela 37** e a **Figura 28** apresentam o planeamento referente ao primeiro semestre, sendo os focos principais a identificação dos requisitos que precisam ser satisfeitos e um protótipo onde estejam presentes algumas funcionalidades do produto final.

Nome da tarefa	Duração	Início	Termino
Planeamento do primeiro semestre	2 dia	Seg 20/02/2017	Ter 21/02/2017
Seleção da metodologia	2 dias	Qua 22/02/2017	Qui 23/02/2017
Contextualização do projeto	2 dias	Sex 24/02/2017	Seg 27/02/2017
Estado da arte	21 dias	Ter 28/02/2017	Ter 28/03/2017
Levantamento de requisitos	18 dias	Qua 29/03/2017	Sex 21/04/2017
Prototipagem	20 dias	Seg 24/04/2017	Sex 19/05/2017
Esboço do relatório intermédio	16 dias	Seg 22/05/2017	Seg 12/06/2017
Relatório final	9 dias	Ter 13/06/2017	Sex 23/06/2017
Apresentação	5 dias	Seg 26/06/2017	Seg 03/07/2017

Tabela 37 - Tarefas referentes ao primeiro semestre

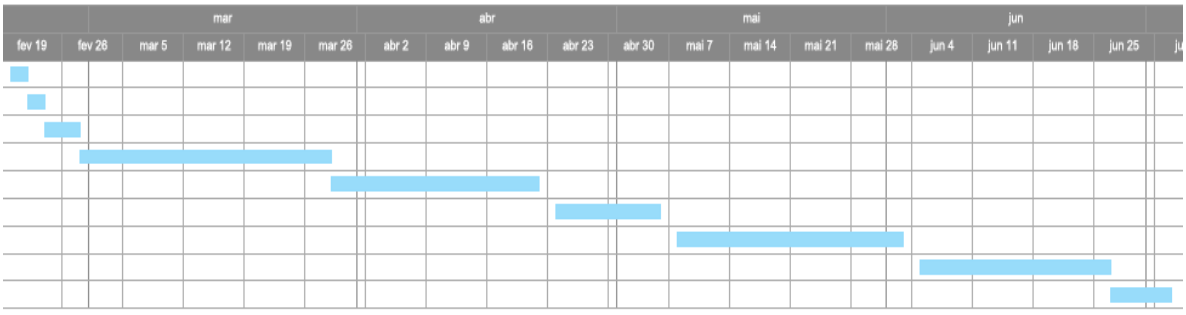

Figura 28 - Diagrama de Gantt do planeamento das tarefas do primeiro semestre

7.1.1 Tempo real das tarefas do primeiro semestre

A **Tabela 38** e a **Figura 29** apresentam o planeamento referente ao primeiro semestre com o tempo gasto na realizada em cada tarefa.

Nome da tarefa	Duração	Início	Termino
Planeamento do primeiro semestre	2 dia	Seg 20/02/2017	Ter 21/02/2017
Seleção da metodologia	2 dias	Qua 22/02/2017	Qui 23/02/2017
Contextualização do projeto	2 dias	Sex 24/02/2017	Seg 27/02/2017
Estado da arte	21 dias	Ter 28/02/2017	Ter 28/03/2017
Levantamento de requisitos	18 dias	Qua 29/03/2017	Sex 21/04/2017
Prototipagem	20 dias	Seg 24/04/2017	Sex 19/05/2017
Esboço do relatório intermédio	16 dias	Seg 22/05/2017	Seg 12/06/2017
Relatório final	9 dias	Ter 13/06/2017	Sex 23/06/2017
Apresentação	5 dias	Ter 27/06/2017	Seg 03/07/2017

Tabela 38 - Tempo real das tarefas realizadas no primeiro semestre

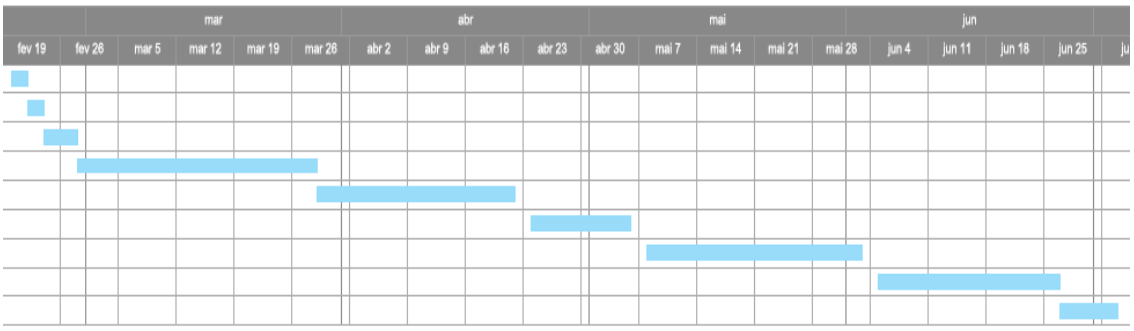

Figura 29 - Diagrama de *Gantt* com o tempo real das tarefas do primeiro semestre

Como podemos verificar pela comparação das duas tabelas e figuras, o tempo atribuído a cada tarefa no planeamento ficou semelhante ao tempo despendido na realidade. Como não existe nenhuma diferença significativa sugere que o planeamento foi bem realizado e adequado ao projeto.

7.1.2 Descrição do planeamento referente ao primeiro semestre

Na **Tabela 39** são indicadas as tarefas globais do que foi realizado no primeiro semestre, tendo cada uma delas uma breve descrição dos assuntos abordados.

Tarefas	**Descrição**
Planeamento do primeiro semestre	Consiste na criação de artefactos (documentos) de apoio ao projeto, com o objetivo de organizar as tarefas necessárias para o decorrer do primeiro semestre.
Seleção da metodologia	Consiste na escolha de um conjunto estruturado de práticas que irão influenciar o processo de desenvolvimento do projeto com base nos processos diferentes que cada uma apresenta. Essa escolha é realizada com base nas características tanto do projeto como do cliente, sendo o cliente o facto mais importante dos dois. Os tipos de metodologias mais utilizadas são o desenvolvimento em cascata, orientada a objetos e desenvolvimento ágil.
Contextualização do projeto	Aquisição de uma primeira visão geral do projeto. É nesta fase que se obtém uma visão da necessidade que o cliente apresenta, quais os objetivos globais que devem ser atingidos e quais as características genéricas que o programa deve apresentar.
Estado da arte	Análise de aplicações semelhantes ao projeto em desenvolvimento. Pretende-se perceber os pontos fortes e fracos da concorrência e com essa informação aumentar a qualidade evitando os problemas identificados.
Levantamento de requisitos	Utilizando uma série de técnicas de levantamento de requisitos (contextual design, casos de uso, etc.) é pretendido identificar os requisitos funcionais e não funcionais necessários para que a aplicação, no final esteja em conformidade com a vontade do cliente, mesmo o próprio não sabendo à partida o que deseja.
Prototipagem	A prototipagem é realizada para verificar com o cliente se toda a análise executada até esta fase está em conformidade com os seus requisitos. Nesta etapa é natural elaborar protótipos de baixa e média fidelidade para se testar os requisitos de uma maneira rápida e flexível.
Esboço do relatório intermédio	Iniciar o desenvolvimento do relatório intermédio, com várias interações com os orientadores.
Relatório final	Finalização do relatório intermédio.
Apresentação	Elaboração de um documento em formato *PowerPoint* para a realização da apresentação intermédia.

Tabela 39 – Descrição das tarefas realizadas no primeiro semestre

7.2 Planeamento do segundo semestre

A **Tabela 40** e a **Figura 30** apresentam o planeamento referente ao primeiro semestre, sendo o foco principal a identificação dos requisitos que precisam ser satisfeitos e um protótipo onde estejam presentes algumas funcionalidades do produto final.

Nome da tarefa	Duração	Início	Término
Correção do relatório	15 dias	Sex 01/09/2017	Qui 21/09/2017
Recolha de dados	5 dias	Sex 22/09/2017	Qui 28/09/2017
Limpeza dos dados recolhidos	15 dias	Sex 29/09/2017	Qui 19/10/2017
Transformação dos dados	15 dias	Sex 20/10/2017	Qui 09/11/2017
Implementação do algoritmo de *Machine Learning*	15 dias	Sex 10/11/2017	Qui 30/11/2017
Testes de performance	5 dias	Sex 01/12/2017	Qui 07/12/2017
Esboço do relatório final	15 dias	Sex 08/12/2017	Qui 28/12/2017
Relatório conclusão final	18 dias	Sex 29/12/2017	Ter 23/01/2018
Apresentação	5 dias	Qua 24/01/2018	Ter 30/01/2018

Tabela 40 - Tarefas referentes ao segundo semestre

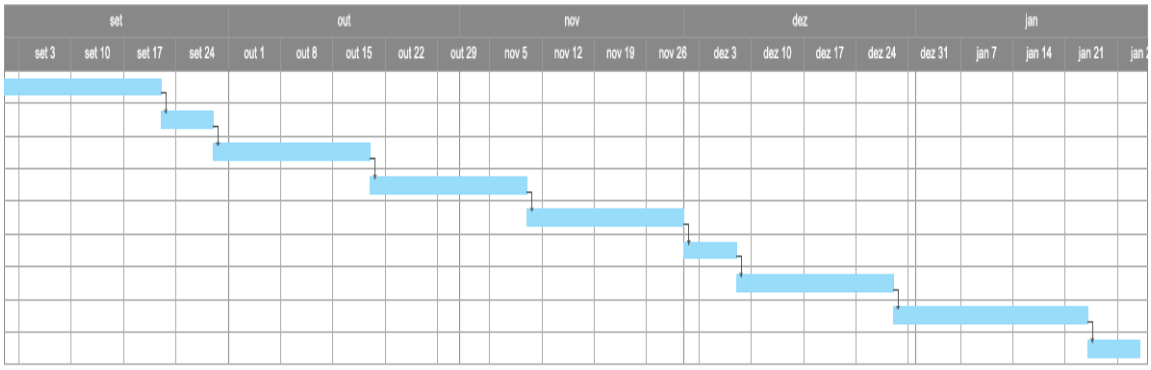

Figura 30 - Diagrama de *Gantt* do planeamento das tarefas do segundo semestre

7.2.1 Tempo real das tarefas do segundo semestre

A **Tabela 41** e a **Figura 31** apresentam o planeamento referente ao segundo semestre com o tempo gasto na realizada em cada tarefa.

Nome da tarefa	**Duração**	**Início**	Término
Correção do relatório	15 dias	Sex 01/09/2017	Qui 21/09/2017
Recolha de dados	5 dias	Sex 22/09/2017	Qui 28/09/2017
Limpeza dos dados recolhidos	15 dias	Sex 29/09/2017	Qui 19/10/2017
Transformação dos dados	15 dias	Sex 20/10/2017	Qui 09/11/2017
Implementação do algoritmo de *Machine Learning*	15 dias	Sex 10/11/2017	Qui 30/11/2017
Testes de performance	5 dias	Sex 01/12/2017	Qui 07/12/2017
Esboço do relatório final	15 dias	Sex 08/12/2017	Qui 28/12/2017
Relatório conclusão final	18 dias	Sex 29/12/2017	Ter 23/01/2018
Apresentação	5 dias	Qua 24/01/2018	Ter 30/01/2018

Tabela 41 – Tarefas com resultados reais referentes ao segundo semestre

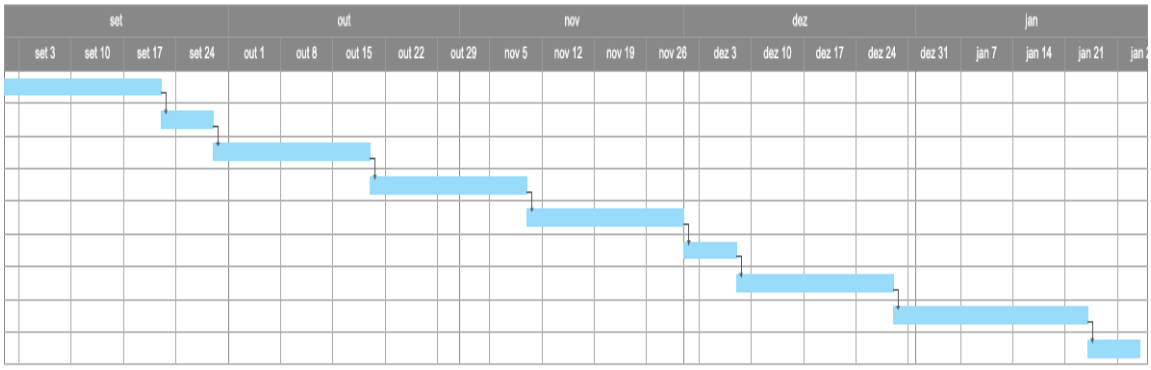

Figura 31 - Diagrama de Gantt com o tempo real das tarefas do segundo semestre

Verificamos com a análise destes dois gráficos que o plano foi executado na perfeição. Podemos concluir que o tempo dado às tarefas foi bem calculado, não havendo a necessidade de ajustar o plano.

7.2.2 Descrição do planeamento referente ao segundo semestre

Na **Tabela 19** estão indicadas as tarefas globais do que foi realizado no segundo semestre, tendo cada uma delas uma breve descrição dos assuntos abordados.

Tarefas	**Descrição**
Correção do relatório	Alterações realizadas ao relatório referentes à apresentação intermédia, realizada na Universidade de Coimbra.
Recolha de dados	Identificação dos processos de armazenamento de informação e tecnologias envolvidas. Nesta fase são identificadas as alterações necessárias para comunicar com a plataforma ACM para realizar a recolha.
Limpeza dos dados recolhidos	Análise dos dados recolhidos, com atenção aos dados nulos, formatação dos dados e valores inválidos.
Transformação dos dados	Nesta fase serão feitos vários testes com o objetivo de perceber como se poderá melhorar a organização dos dados, aumentando assim a eficácia do algoritmo que se pretende desenvolver.
Implementação do algoritmo de *Machine Learning*	Implementação do algoritmo de *Machine Learning* com alteração dos vários atributos que ele possui com o intuito de testar qual o grupo de configurações mais adequado a este projeto.
Testes de *performance*	Perceber os impactos da solução no ambiente de produção do ACM e identificar quais os pontos críticos da aplicação.
Esboço do relatório final	Iniciar o desenvolvimento do relatório final, com várias interações com os orientadores.
Relatório conclusão final	Finalização do relatório final.
Apresentação	Elaboração de um documento em formato *PowerPoint* para a realização da apresentação final.

Tabela 19 – Descrição das tarefas realizadas no segundo semestre

Capítulo 8
Conclusões

Nesta dissertação, com os recursos disponibilizados pela Altice Labs, foi realizado um projeto onde foram aplicadas diferentes ferramentas da área de inteligência artificial, dando origem a um *software* que possibilita a interação com outros sistemas, com vista a atingir resultados superiores aos que até então se haviam registado.

Para avançar com o projeto foi escolhido o processo que melhor se adequava aos objetivos e restrições identificadas, possibilitando a criação da arquitetura, a escolha das ferramentas e técnicas a implementar. Outro ponto crucial foi a escolha de qual algoritmo de *Machine Learning* deveria de ser utilizado para se obter os melhores resultados.

A análise em volta dos algoritmos foi a mais extensa e demorada etapa realizada. Para que a escolha fosse ponderada e fundamentada foi realizado um estudo dos algoritmos de *Machine Learning* existentes, bem como a lógica inerente ao seu funcionamento. Contudo, só isso não foi o suficiente, realizando-se testes utilizando uma panóplia de algoritmos, com o intuído de identificar qual se adaptava melhor aos dados.

Os testes foram realizados com base em cenários reais, onde foi possível medir e adaptar da melhor maneira o sistema desenvolvido. Nesta fase, também se avaliaram as funcionalidades de acordo com os requisitos estabelecidos e restrições impostas pela empresa.

Em termos globais o *software* desenvolvido foi um sucesso, pois resolve os problemas indicados pela empresa, adaptando-se perfeitamente aos sistemas existentes já em produção. É de salientar que o tempo do projeto foi bem medido e a entrega realizada na data estabelecida. As eventualidades que o projeto apresentou foram bem identificadas no planeamento dos riscos, facilitando a sua abordagem e resolução.

Para melhorias futuras considera-se a realização de uma extensão do algoritmo de forma a alcançar as subcategorias existentes no ACM e adaptar o algoritmo para um maior número de plataformas existentes na Altice Labs.

Referências

[1] PT, A evolução histórica da empresa PT, https://www.telecom.pt/pt-pt/a-pt/Paginas/historia.aspx, Último acesso: 02/03/2017

[2] ALTICELABS, ACM Active Campaign Manager, http://www.alticelabs.com/site/acm/wp-content/uploads/sites/2/2016/02/BR_ACM_ALB_EN.pdf , Último acesso: 02/03/2017

[3] INVESTOPEDIA, Data Science, http://www.investopedia.com/terms/d/data-science.asp, Último acesso: 04/04/2017

[4] INVESTOPEDIA, Data Mining, http://www.investopedia.com/terms/d/datamining.asp, Último acesso: 04/04/2017

[5] FERREIRA, Sistema de Recomendação baseado em Data Mining, https://recipp.ipp.pt/bitstream/10400.22/8664/1/DM_BrunoFerreira_2016_MEI.pdf, Último acesso: 04/04/2017

[6] PROVOST, F., AND FAWCETT,.T. , 2013 ,Data Science for Business, Gravenstein Highway North, Sebastopol, CA 95472.

[7] BRAMER,.M. , 2007 , Principles of Data Mining, University of Portsmouth, UK.

[8] LAROSE,.D.T. , 2005 , Discovering Knowledge in Data, Hoboken, New Jersey.

[9] HOSCH, W.L., Machine learning, https://global.britannica.com/technology/machine-learning, Último acesso: 19/05/2017

[10] COPELAND, B.J., Artificial intelligence, https://global.britannica.com/technology/artificial-intelligence, Último acesso: 19/05/2017

[11] MICROSOFT, Data Mining Algorithms, https://docs.microsoft.com/pt-br/sql/analysis-services/data-mining/data-mining-algorithms-analysis-services-data-mining, Último acesso: 22/05/2017

[12] JADHAV, S.D, AND CHANNE, H.P., Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques, https://www.ijsr.net/archive/v5i1/NOV153131.pdf, Último acesso: 30/05/2017

[13] TAMOUK, T., AND ALLAHAKBARI, F., A comparison among accuracy of KNN, PNN, KNCN, DANN and NFL, https://www.ijcsi.org/papers/IJCSI-9-3-1-319-322.pdf, Último acesso: 30/05/2017

[14] OZSAKAVASI,.F , 2008 , In partial fulfillment of the requirements for the degree of master of science in geodetic and geographic information technologies, Department of Geodetic and Geographic Information Technologies, https://etd.lib.metu.edu.tr/upload/3/12609815/index.pdf, Último acesso: 19/05/2017

[15] ZENTUP, Advantages and Disadvantages of Data Mining, http://www.zentut.com/data-mining/advantages-and-disadvantages-of-data-mining/, Último acesso: 19/05/2017

[16] KENNAN, T., R vs. Java vs. Python: Which Is Right for Your Project?, https://www.upwork.com/hiring/data/r-vs-java-vs-python-which-is-best/, Último acesso: 27/05/2017

[17] SVETLANA S. AKSENOVA, WEKA Explorer Tutorial, http://people.sabanciuniv.edu/berrin/cs512/lectures/WEKA/WEKA%20Explorer%20Tutorial-REFERENCE.pdf, Último acesso: 29/05/2017

[18] BROWNLEE JASON, What is the Weka Machine Learning Workbench, https://machinelearningmastery.com/what-is-the-weka-machine-learning-workbench/, Último acesso: 29/05/2017

[19] BOUCKAERT at al., WEKA Manual for Version 3-6-8, https://www.google.pt/search?q=weka+manual+tutorial&oq=weka+manual+tutorial&aqs=chrome..69i57.7144j0j7&sourceid=chrome&ie=UTF-8, Último acesso: 29/05/2017

[20] QUORA, https://www.quora.com/What-is-Pentaho-and-what-services-does-it-provide, Último acesso: 29/05/2017

[21] PENTAHO, Pentaho Data Integration User Guide, http://docs.huihoo.com/pentaho/pentaho-business-analytics/4.8/pdi_user_guide.pdf, Último acesso: 29/05/2017

[22] ROBINSON, S., The Best Machine Learning Libraries in Python, http://stackabuse.com/the-best-machine-learning-libraries-in-python/, Último acesso: 29/05/2017

[23] HELMENSTINE, T., Group?, https://www.thoughtco.com/control-and-experimental-group-differences-606113, Último acesso: 29/05/2017

[24] MUNIR, A., Why Lift Analysis is the Only Way to Understand the Impact of Your Messaging, http://info.localytics.com/blog/using-lift-analysis-for-measuring-push-and-in-app-messaging-impact, Último acesso: 29/05/2017

[25] SALFNER,.F.,LENK, M., AND MALEK, M., A Survey of Online Failure Prediction Methods, https://www.informatik.hu-berlin.de/de/Members/salfner/publications/salfner10survey.pdf, Último acesso: 29/05/2017

[26] ORACLE, Database Concepts, https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm, Último acesso: 29/05/2017

[27] Dicas para o desenvolvimento, https://1up4dev.files.wordpress.com/2008/06/waterfall_model.png, Último acesso: 29/05/2017

[28] Hughey, D, http://www.umsl.edu/~hugheyd/is6840/waterfall.html, Último acesso: 29/05/2017

[29] WALT, S., Process Models in Software Engineering, http://www.ics.uci.edu/~wscacchi/Papers/SE-Encyc/Process-Models-SE-Encyc.pdf, Último acesso: 29/06/2017

[30] DEBRAY, Classification in Class Imbalanced Datasets, http://www.netstorm.be/images/stories/classifimbalanced/fig2-1.png, Último acesso: 29/05/2017

[31] TRAN QUE, Improving Random Forest Algorithm through Automatic Programming, https://brage.bibsys.no/xmlui/bitstream/handle/11250/293870/15-00486-6%20thesis.pdf%20235121_1_1.pdf?sequence=1, Último acesso: 29/05/2017

[32] PUB-RIO, Árvore de Decisão, https://www.maxwell.vrac.puc-rio.br/7587/7587_4.PDF, Último acesso: 29/06/2017

[33] VON ZUBEN,AND ATTUX,Árvores de Decisão, ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia004.../topico7_IA004_1s10.pdf , Último acesso: 29/06/2017

[34] DUBA, HART, STORK, WILEY, AND SONS, Naive Bayes Classifier, http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf, Último acesso: 29/06/2017

[35] MILANE P.,A Tutorial on Clustering Algorithms, https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html, Último acesso: 30/06/2017

[36] LISBOA T., Algoritmos para biologia computacional, http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/index651a.html?id=147, Último acesso: 30/06/2017

[37] KRUCHTEN B., The 4+1 View Model of Architecture, http://www.ics.uci.edu/~andre/ics223w2006/kruchten3.pdf, Último acesso: 30/06/2017

[38] DEVREUVE, AND MORPHEME T., An introduction to random forests, University Nice Sophia Antipolis, https://perso.math.univ-toulouse.fr/motimo/files/2013/07/random-forest.pdf, Último acesso: 30/06/2017

[39] LORENZETT C., AND TELÖCKEN, Estudo Comparativo entre os algoritmos de Mineração de Dados Random Forest e J48 na tomada de Decisão, http://www.lbd.dcc.ufmg.br/colecoes/spdc/2016/004.pdf, Último acesso: 01/07/2017

[40] JAGANNATH V., Random Forest Template for TIBCO Spotfire® - Wiki page, https://community.tibco.com/wiki/random-forest-template-tibco-spotfirer-wiki-page, Último acesso: 01/07/2017

[41] MAHMOOD, AND QAZAFI, LC an effective classification based association rule mining algorithm, http://eprints.hud.ac.uk/24274/1/Qazafi_-Thesis_Modified_version_-_Final.pdf, Último acesso: 18/12/2017

[42] HINTO,AGEIFFREY ,Boltzmann machine, http://www.scholarpedia.org/article/Boltzmann_machine, Último acesso: 18/12/2017

[43] DEEPPLEARNING, Restricted Boltzmann Machines, http://deeplearning.net/tutorial/rbm.html, Último acesso: 18/12/2017

[44] STARSOFT, Naive Bayes Classifier, http://www.statsoft.com/textbook/naive-bayes-classifier, Último acesso: 18/12/2017

[45] ROJAS, Neural Networks, https://page.mi.fu-berlin.de/rojas/neural/chapter/K4.pdf, Último acesso: 18/12/2017

Apêndices

Apêndice A – Testes dos algoritmos de Machine Learning

Na busca pelo algoritmo mais adequado e eficaz para o projeto desenvolvido foram realizados vários testes em torno dos dados existentes. Na secção seguinte serão apresentados e descritos os resultados obtidos e apresentadas as razões pela qual os algoritmos não foram escolhidos.

A.1 Testes na redução do público-alvo

Os algoritmos apresentados têm características diferentes, o que influencia positivamente e negativamente os resultados apresentados. É importante referir que os algoritmos foram testados várias vezes, contudo só estão indicados os melhores resultados de cada, possibilitando a comparação do potencial individual com recurso aos melhores resultados. O objetivo destes testes era conseguir atingir uma redução mais de 50% do público-alvo e não excluir mais de 10% dos clientes aderentes à campanha. Para este teste será utilizada a CAM 1.

Nas tabelas serão usadas siglas para facilitar a perceção e identificação dos valores críticos que estão indicados. Assim, são explicadas as siglas na tabela seguinte.

Siglas	**Descrição**
#SUT	Sucessos no conjunto de testes.
#INT	Insucesso no conjunto de testes.
#SUPA	Número de sucessos no público-alvo.
#INPA	Número de insucesso no público-alvo.
#EFR	Efetividade real. É obtida em detrimento do cálculo dos sucessos sobre os insucessos.
#NPA	Novo público-alvo.
#TRPA	Taxa de redução do público-alvo. É obtido da percentagem da diferença do total com o #NPA.
#SP	Número de sucessos perdidos.
#TP	Taxa de perda referente a #SP.
#EFA	Taxa da eficácia do algoritmo.

Tabela 42 – Siglas referentes às tabelas

A.1.1 Algoritmo KNN

Os resultados obtidos para o KNN são indicados na **Tabela 44**, utilizado o *K* igual a três e os restantes parâmetros a *default*.

#SUT	**#INT**	**#SUPA**	**# INPA**	**#EFR**	**#NPA**	**#TRPA**	**#SP**	**#TP**	**#EFA**
3000	3000	830	407793	0,20%	163842	60%	110	13%	0.44%

Tabela 43 – Resultados do KNN

A.1.2 Algoritmo J48

Os resultados obtidos no algoritmo *J48* são indicados na **Tabela 45.**

#SUT	#INT	#SUPA	# INPA	#EFR	#NPA	#TRPA	#SP	#TP	#EFA
3000	3000	830	407793	0,20%	72570	82%	159	24%	0.92%

Tabela 44 – Resultados do *J48*

A.1.3 Random Tree

Os resultados obtidos no algoritmo *Random Tree* são indicados na **Tabela 46.**

#SUT	#INT	#SUPA	# INPA	#EFR	#NPA	#TRPA	#SP	#TP	#EFA
3000	3000	830	407793	0,20%	72955	82%	219	36%	0.83%

Tabela 45 – Resultados do Random Tree

A.1.4 Random Florest

Os resultados obtidos no algoritmo *Random Florest* são indicados na **Tabela 47.**

#SUT	#INT	#SUPA	# INPA	#EFR	#NPA	#TRPA	#SP	#TP	#EFA
3000	3000	830	407793	0,20%	72955	82%	219	36%	0.83%

Tabela 46 – Resultados do Random Florest

Como podemos verificar, os valores apresentados nos diversos algoritmos mostram que os algoritmos conseguiram reduzir bastante o público-alvo. Contudo, não obedeceu ao segundo parâmetro avaliado, apresentando uma perda de clientes superior a 10%. Isso faz com que os algoritmos sejam excluídos da possibilidade de integrarem a solução e o *Perceptron* que está indicado no corpo do relatório na secção de desenvolvimento tenha sido o escolhido.

A.2 Campanha mais adequada para um determinado cliente

Nesta secção são apresentados os testes realizados para a escolha do algoritmo mais adequado no caso de uso responsável por declarar a melhor campanha para um determinado cliente. Utilizando matrizes de confusão para apresentar os dados, conseguimos entender o comportamento do algoritmo e avaliar os resultados imparcialmente. Estes indicadores foram analisados considerando a percentagem de cada campanha, tendo que obrigatoriamente apresentar uma percentagem de erro inferior a 15%.

Para obter os melhores resultados foram necessários dois momentos de avaliação. Um primeiro momento onde se avalia os resultados do algoritmo com a utilização de todos os atributos existentes nas campanhas, e um segundo momento onde são identificados os atributos que influenciam positivamente o algoritmo, tentando assim aumentar a assertividade, removendo os restantes atributos.

Os testes foram desenvolvidos com base em duas campanhas, a CAM 32 e a CAM 75. Dessas campanhas foram retirados três mil clientes de sucesso e três mil clientes de insucesso. Para cada campanha os clientes escolhidos não podiam estar contidos no grupo

da restante campanha. Assim, os clientes de sucesso e os de insucesso são completamente diferentes.

Na escolha dos atributos mais indicados para realizar a classificação, é utilizado um algoritmo do *WEKA* chamado *InfoGainAttributeEval* que fornece os atributos com a percentagem mais alta de influência no algoritmo. Os atributos escolhidos para cada atributo estarão presentes na secção **A.3**.

Apesar dos resultados de alguns dos algoritmos serem bons, o *Perceptron* também apresenta valores igualmente bons, o que faz com que exista uma preferência em manter o mesmo algoritmo em todas as etapas do projeto.

A.2.1 Algoritmo de KNN

Resultados apresentados na matriz de confusão da **Tabela 48** são obtidos com recurso a todos os atributos existentes.

	Sem campanha	**Campanha A**	**Campanha B**
Sem campanha	159	60	68
Campanha A	75	143	48
Campanha B	65	27	171

Tabela 47 – Resultados do KNN com duas campanhas

Resultados apresentados na matriz de confusão da **Tabela 49** são obtidos com recurso somente aos atributos com maior influência no algoritmo.

	Sem campanha	**Campanha A**	**Campanha B**
Sem campanha	136	82	69
Campanha A	37	189	40
Campanha B	17	6	240

Tabela 48 - Resultados do *KNN* com duas campanhas e atributos selecionados

A.2.2 Algoritmo J48

Resultados apresentados na matriz de confusão da **Tabela 50** são obtidos com recurso a todos os atributos existentes.

	Sem campanha	**Campanha A**	**Campanha B**
Sem campanha	150	84	53
Campanha A	49	195	22
Campanha B	49	9	205

Tabela 49 – Resultados do *J48* com duas campanhas

Resultados apresentados na matriz de confusão da **Tabela 51** são obtidos com recurso somente aos atributos com maior influência no algoritmo.

	Sem campanha	Campanha A	Campanha B
Sem campanha	101	115	71
Campanha A	5	242	19
Campanha B	10	22	231

Tabela 50 – Resultados do *J48* com duas campanhas e atributos selecionados

A.2.3 Decision Tree

Resultados apresentados na matriz de confusão da **Tabela 52** são obtidos com recurso a todos os atributos existentes.

	Sem campanha	Campanha A	Campanha B
Sem campanha	110	105	72
Campanha A	23	234	9
Campanha B	15	20	228

Tabela 51 – Resultados do Decision Tree com duas campanhas

Resultados apresentados na matriz de confusão da **Tabela 53** são obtidos com recurso somente aos atributos com maior influência no algoritmo.

	Sem campanha	Campanha A	Campanha B
Sem campanha	110	105	72
Campanha A	23	234	9
Campanha B	15	20	228

Tabela 52 - Resultados do *Decision Tree* com duas campanhas e atributos selecionados

A.3 Atributos selecionados para utilizar nos algoritmo

Nesta secção estão representados os atributos que foram utilizados em cada algoritmo para potencializar os resultados já obtidos com a informação total.

ID (BI)	ID (ACM)	Caraterística	Formato BI	Formato ACM	Casas Décimais (Números) / Formato (Datas)	Valor por omissão	Categoria
1002	618234131	Família tarifário		Número	0	-1	Dados Pessoais/Cartão
1223	1152616251	Definição de mercado		Número	0	0	Dados Pessoais/Cartão
1225	627106443	Qtd de carregamentos - 6M		Número	0	0	Comportamento (diário)

Tabela 53 – Atributos selecionados para o KNN

ID (BI)	ID (ACM)	Caraterística	Formato BI	Formato ACM	Casas Décimais (Números) / Formato (Datas)	Valor por omissão	Categoria
1227	1636216783	Média do valor carregado - 6M		Número	2	0	Comportamento (diário)
1244	1054619596	Valor mínimo carregado - 3M		Número	2	0	Comportamento (mensal)
1245	1769092783	Valor máximo carregado - 3M		Número	2	0	Comportamento (mensal)
1247	1641362836	Média do valor carregado por mês - 3M		Número	2	0	Comportamento (mensal)
1251	258317398	ARPU - 3M		Número	2	0	Rentabilidade/Valor

Tabela 54 – Atributos selecionados para o *J48*

ID (BI)	ID (ACM)	Caraterística	Formato BI	Formato ACM	Casas Décimais (Números) / Formato (Datas)	Valor por omissão	Categoria
1002	618234131	Família tarifário		Número	0	-1	Dados Pessoais/Cartão
1019	1645875847	Escalão do segmento sensibilidade ao saldo cliente		Texto		N/E	Comportamento (diário)
1095	169359089	Dur seg TOR voz		Número	0	0	Tráfego Originado - voz
1223	1152616251	Definição de mercado		Número	0	0	Dados Pessoais/Cartão
1251	258317398	ARPU - 3M		Número	2	0	Rentabilidade/Valor
1265	1495254132	Val TOR voz offnet RF - 3M		Número	2	0	Tráfego Originado - voz

Tabela 55 – Atributos selecionados para o algoritmo *Decision Tree*