Faculdade de Ciências e Tecnologia

Departamento de Engenharia Informática

Implementação de um sistema de recomendação aplicado a um sistema de gestão de campanhas

Diogo André Cadima Alves

Dissertação no âmbito do Mestrado em Engenharia Informática, Especialização em Sistemas de Informação, orientado pelo Prof. Luís Macedo, Maria Manuel Castro e Ricardo Ângelo Filipe e apresentado à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Julho 2019

1 2 9 0

UNIVERSIDADE D
COIMBRA

Esta página foi intencionalmente deixada em branco.

Agradecimentos

À Altice Labs pela oportunidade de estágio num ambiente enriquecedor, no qual aprendi bastante.

A toda a equipa do ACM por me terem recebido da melhor forma. Em especial, aos meus orientadores Maria Manuel Castro e Ricardo Ângelo Filipe, pela disponibilidade e boa-vontade demonstradas em ajudarem-me no que precisei e por todo o conhecimento transmitido.

Ao meu orientador do DEI, Prof. Luís Macedo, por toda a sua ajuda no estudo realizado e nas revisões deste documento.

A todos os meus amigos, que sempre me acompanharam e torceram pelo meu sucesso.

À Joana Gomes, por toda a força, ajuda e compreensão que me deu e demonstrou nesta etapa importante.

À minha família, por todo o apoio que sempre me deram e por fazerem com que chegasse onde cheguei hoje, tanto a nível pessoal como a nível académico.

Esta página foi intencionalmente deixada em branco.

Abstract

The plethora of information available on the Internet nowadays, makes difficult for clients to find useful products for their necessities. This problem should be tackled by the industry with two main goals: first, to increase companies profits and secondly to improve quality-of-service. Hence, recommender systems are useful tools to solve this problem by providing customized and personalized recommendations for products and services.

The goal of this internship is to study and develop a recommender system to incorporate in the Active Campaign Manager (ACM) platform. This platform enables telecommunications operators to manage their advertising campaigns. This study looks at several techniques that allow ACM to obtain a list of campaign recommendations that can be sent to a particular client. Recommendations should be made taking into account the characteristics of each client and their history of subscriptions in other campaigns.

Taking into account the study performed on recommendation algorithms, it was considered three diferent methods for obtaining the recommendations. First, using open-source frameworks for recommender systems, the second one through the use of entreprise software that allows the implementation of recommender engines on the cloud and the final method uses the subscription history of the clients alongside their characteristics and applies clustering methods to generate the campaign recommendations. It is considered that a study that covers these three approaches will be a very complete study that will address the problem of this internship and will allow to determine the best approach for ACM.

Keywords

Recommender systems, telecommunication operators, advertising campaigns, data analysis, collaborative filtering.

Esta página foi intencionalmente deixada em branco.

Resumo

A enorme quantidade de informação disponível na Internet atualmente dificulta a tarefa de um utilizador encontrar o que realmente lhe interessa. Este problema deve ser endereçado pelas empresas de forma a oferecerem aos seus clientes os produtos ou serviços que lhes sejam de maior agrado, tendo em vista o aumento de vendas. Sistemas de recomendação são utilizados para resolver este problema, fornecendo recomendações de produtos e serviços de acordo com os gostos e interesses dos utilizadores.

O objetivo deste estágio é o estudo e desenvolvimento de um sistema de recomendações para aplicar no Active Campaign Manager (ACM), uma plataforma que permite a operadoras de telecomunicação gerirem as suas campanhas publicitárias. Este estudo procura analisar várias técnicas que permitam ao ACM obter uma lista de recomendações de campanhas que possam ser enviadas a um determinado cliente. As recomendações são baseadas nas características de cada cliente e no seu histórico de adesões a campanhas.

Tendo em conta o estudo realizado sobre algoritmos de recomendação, consideraram-se três meios diferentes para obter as recomendações. O primeiro é recorrendo a uma *framework open-source* para implementação de sistemas de recomendação de filtragem colaborativa, o segundo utilizando software empresarial que permite a construção de um motor de recomendação na *cloud* e a terceira abordagem que utiliza o histórico de adesões juntamente com as características dos cliente e aplica métodos de *clustering* para gerar as recomendações de campanhas.

Considera-se que um estudo que abranja estas três abordagens, será um estudo consideravelmente completo que dará resposta ao problema deste estágio e permitirá determinar qual a melhor abordagem para o ACM.

Palavras-Chave

Sistema de recomendações, operadoras de telecomunicações, campanhas publicitárias, análise de dados, filtragem colaborativa.

Esta página foi intencionalmente deixada em branco.

Conteúdo

Esta página foi intencionalmente deixada em branco.

Glossário

ACM Active Campaign Manager. v, vii, 2–5, 7, 8, 39, 49, 79, 80

ARPU Average Revenue Per User. 51

GSM Global System for Mobile Communications. 32

KS Katz Similarity. 25

MAE Mean Absolute Error. 18, 41, 42

ML Machine Learning. 29, 45, 49, 50, 72, 79

MRR Mean Reciprocal Rank. 41, 42

PCA Principal Component Analysis. xiv, 45, 54–56, 58, 62, 77

RMSE Root Mean Squared Error. 18, 41, 42

SR Sistema de Recomendação. 72, 73

UMTS Universal Mobile Telecommunications System. 32

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

Esta página foi intencionalmente deixada em branco.

Lista de Tabelas

Esta página foi intencionalmente deixada em branco.

Capítulo 1

Introdução

O presente documento consiste no relatório de estágio realizado pelo aluno Diogo André Cadima Alves, no âmbito do Mestrado em Engenharia Informática, ramo de Sistemas de Informação, na Universidade de Coimbra. O estágio foi realizado na Altice Labs, empresa com sede em Aveiro, especializada no desenvolvimento de soluções tecnológicas para Telecomunicações e Sistemas de Informação. Este estágio foi realizado sob orientação do Prof. Luís Macedo da Universidade de Coimbra e os orientadores Maria Manuel Castro e Ricardo Ângelo Filipe da Altice Labs.

1.1 Enquadramento e Problema

O crescimento exponencial de informação disponível na Internet torna cada vez mais complexa e demorada a tarefa dos seus utilizadores de encontrarem informação que realmente lhes é relevante. Torna-se uma obrigação para as empresas terem a capacidade de perceberem o que os clientes realmente pretendem e apresentar-lhes informação que lhes é a mais útil possível. O objetivo é motivá-los para que estes continuem a utilizar os seus serviços, a melhoria da sua experiência e a angariação de novos clientes. Naturalmente, estas ações têm em vista o aumento das receitas da empresa.

A recomendação de produtos/serviços pelas empresas aos seus clientes pode substancialmente melhorar a relação entre ambos. Isto resulta na geração de maiores lucros para as empresas devido ao aumento do número de vendas/adesões a serviços e à maior fidelidade dos clientes pela empresa [3]. Existem atualmente grandes empresas, como a *Netflix*, *Amazon* e *Spotify*, que utilizam a recomendação de produtos/serviços tanto para aumentar as suas vendas como o nível de utilização dos seus serviços por parte dos clientes. Por exemplo, para o caso da *Netflix*, as recomendações desempenham um papel tão importante que cerca de 75% a 80% das visualizações feitas através da plataforma de *streaming* devem-se às recomendações feitas pelo serviço.

Apresentar produtos que o cliente provavelmente desconhece pode despertar o seu interesse e fazer com que ele venha, efetivamente, a adquirir os produtos que lhe foram recomendados. A empresa estará assim a promover produtos/serviços, dando-lhes um maior alcance e a garantir que não passam despercebidos aos clientes, aumentando as hipóteses destes virem a adquiri-los.

Tal como outras empresas de diferentes áreas, também as operadoras de telecomunicações podem, com a recomendação das suas campanhas publicitárias, aumentar os seus lucros e

o número de clientes que aderem aos seus serviços e/ou promoções.

Tendo isso em conta, a Altice Labs desenvolveu uma plataforma para desenho e lançamento de campanhas, o Active Campaign Manager (ACM). Direcionada às operadoras de telecomunicações, esta plataforma permite-lhes, de forma independente, configurar e lançar ações promocionais com os seguintes objetivos: o aumento da satisfação dos clientes, o aumento das receitas, promoção da aquisição de produtos/serviços, redução de custos de operação, entre outros. Mais detalhes acerca deste produto serão abordados no capítulo 2.

Um dos desafios de uma empresa que utiliza campanhas publicitárias afim de promover os seus produtos ou serviços é garantir uma forte adesão dos clientes às campanhas a que são incentivados. Para tal, as campanhas enviadas deverão ser o mais adequadas possível ao cliente, para que o seu interesse em aderir à campanha seja maior.

Se a plataforma ACM possuir conhecimento das campanhas que deve recomendar e quando as deve recomendar, menos campanhas irrelevantes serão enviadas aos clientes, tendo isto dois efeitos imediatos:

- **psicologicamente** torna-se mais apelativo para o cliente, pois não é incomodado constantemente com campanhas que não lhe interessam. Quando tal acontece, os clientes podem mais tarde estar menos propensos a aderir a uma campanha que lhes foi enviada, mesmo que esta se adeque a ele.
- **computacionalmente** torna-se menos dispendioso, pois menos recursos computacionais e de rede serão utilizados, resultando numa maior eficiência do sistema;

Para tal, pretende-se a adição de um sistema de recomendação ao ACM capaz de sugerir campanhas publicitárias mais relevantes para um determinado cliente, de forma a que este se sinta mais motivado e enquadrado no perfil da campanha.

1.2 Objetivos

O estágio tem como finalidade a implementação de um sistema de recomendação para o ACM, que seja capaz de recomendar uma ou mais campanhas publicitárias para determinado cliente. As recomendações de campanhas serão realizadas tendo em conta as características do cliente. Desta forma, sendo a campanha direcionada a um perfil específico de cliente, existe um aumento da taxa de adesão pois os clientes terão uma maior afinidade com a campanha.

As campanhas recomendadas também permitirão ao cliente ter acesso a serviços, produtos ou promoções que lhe sejam mais adequadas e do seu agrado, permitindo-o usufruir de opções que este considera mais benéficas e/ou úteis. Desta forma, a satisfação do cliente para com os serviços prestados pela operadora aumenta, algo que é vantajoso quer para o cliente, quer para a empresa.

Para realizar as recomendações, o sistema necessitará de dados provenientes do operador, com informação relativa aos seus clientes. Consequentemente, o sistema de recomendações irá analisar esta informação e retornar uma ou mais campanhas que considere relevantes. Esta relevância terá em conta as características dos clientes analisadas.

Tendo em conta o que é pretendido, os dados a analisar deverão conter informação do histórico de campanhas que o cliente já tenha participado. Mais especificamente, os dados conterão a data que foi incentivado a aderir, a data da adesão e se o cliente pertence ao

grupo de controlo ou não. Na secção 2.3 é explicado com mais detalhe do que se trata este último ponto. Caso o histórico de campanhas não exista, existem técnicas que podem ser aplicadas de forma a resolver este problema, como mais à frente neste relatório é descrito como o *Cold Start Problem*.

Com base no ciclo de vida de uma campanha no ACM, que pode rondar alguns meses, e no facto de ser desejável que cada cliente tenha entrado num número mínimo de campanhas, considerou-se que um valor razoável para este estudo seria 2 anos de histórico. Será útil para a eficácia das recomendações analisar também dados que contenham as características do cliente a partir do momento da adesão. De acordo com as normas europeias definidas no Regulamento Geral sobre a Proteção de Dados (RGPD)[5], os dados provenientes das operadoras e utilizados neste estudo são anonimizados. Esta anonimização acontece a dois níveis: ao nível da campanha, onde são criados identificadores únicos que não revelam qual o nome da campanha ou operadora a que pertence e ao nível do utilizador, uma vez que os dados que são fornecidos pela operadora já tratam cada cliente com um identificador interno e não com o seu número pessoal, ou qualquer outra informação que permita que seja identificado.

Concluindo, existem dois objetivos bem definidos para a introdução de um sistema de recomendação na plataforma ACM. Primeiro, o aumento das receitas geradas pelas operadoras que utilizem o ACM, através da maior adesão de clientes às campanhas publicitárias lançadas por estas. Segundo, a melhoria da qualidade de serviço prestado aos clientes, através da apresentação de campanhas que este considere mais vantajosas e de seu interesse. O sistema de recomendação irá desempenhar o papel de uma prova de conceito, servindo para apurar se a sua introdução no ACM será útil para a resolução dos problemas descritos.

1.3 Metodologia e Estrutura do Documento

Contendo este estágio uma grande parte ligada a *data mining* e tendo em conta os objetivos definidos, é necessário adotar uma metodologia que estandardiza o planeamento do projeto. O objetivo é ter um processo no qual seja possível entender o problema a ser resolvido, identificar as várias fases pela qual o projeto necessita de passar, medir os progressos feitos e avaliar os resultados obtidos.

A metodologia escolhida para tal é a *Cross-Industry Process for Data Mining* ou CRISP-DM [6][7]. Esta metodologia fornece uma abordagem estruturada para o planeamento de um projeto de *data mining*, apresentando várias fases do projeto e idealizando a interação entre estas. As fases do CRISP-DM e os respetivos capítulos e secções deste documento onde são apresentadas são os seguintes:

- Análise do negócio, nas secções 1.1 e 1.2;
- Compreensão dos dados, no capítulo 6;
- Tratamento dos dados, nas secções 7.1, 7.4.1, 7.4.2.1, 7.4.2 e 7.4.2.2;
- Modelação, nas secções 7.2, 7.3 e 7.4.3;
- Avaliação, nas secções 7.2.1, 7.3.1 e 7.4.3.2;
- *Deployment*, na secção 10.1.

Mais informações sobre cada uma das fases deste processo estão em anexo a este documento, no apêndice A.

No capítulo 2, o produto ACM e alguns dos seus componentes são apresentados assim como a criação e a monitorização das campanhas. No capítulo 3 deste documento são apresentados alguns conceitos teóricos sobre o sistemas de recomendação. No capítulo 4 são descritos alguns sistemas de recomendação utilizados na indústria e outros utilizados na academia, consistindo este capítulo no estado da arte. No capítulo 5 são descritas as ferramentas e tecnologias que são pensadas utilizar para a realização deste projeto. O capítulo 6 explica a análise dos dados realizada até à data. No capítulo 7 são descritas as abordagens seguidas para gerar as recomendações de campanhas, por diferentes métodos. Quanto ao capítulo 8, este apresenta o plano de riscos que este projeto possui e, finalmente, no capítulo 9 é apresentado o plano de trabalho pensado para o 1º e 2º semestres deste estágio.

Capítulo 2

Conceitos Gerais do Produto Active Campaign Manager

Neste capítulo é descrito o produto ACM e alguns conceitos com ele relacionados. O ACM [8] é uma plataforma desenvolvida pela Altice Labs que visa criar, lançar e executar campanhas para operadoras de telecomunicações. O lançamento de campanhas tem como objetivo proporcionar às operadoras que utilizam o ACM o aumento do número de clientes, de receitas e de participação no mercado.

2.1 Componentes do ACM

O ACM dispõe de uma *interface* simples e intuitiva na qual é possível:

- Desenhar campanhas flexíveis e robustas;
- Monitorizar os serviços e ferramentas que estão a correr na plataforma;
- Gerar relatórios do desempenho das campanhas, fáceis de interpretar que permitam, em tempo real, saber os valores de adesão às campanhas.

Em termos globais, o ACM possuí os seguintes componentes:

- **Desenhador de campanhas**: Interface na qual é possível a criação, manutenção e lançamento de campanhas promocionais (Figura 2.1);
- **Motor de execução em tempo real**: Componente que executa as campanhas, contendo mecanismos de tolerância a falhas, o que garante maior robustez, interação com vários canais e interoperabilidade da plataforma;
- **Módulo de relatórios sobre campanhas**: Fornece indicadores detalhados sobre o comportamento das campanhas que foram lançadas, procurando facilitar a interpretabilidade a quem analisa as campanhas para que possam realizar ajustes nelas, caso necessário (Figura 2.2).
- **Consola de operação e administração**: Permite a operação e administração do ACM.

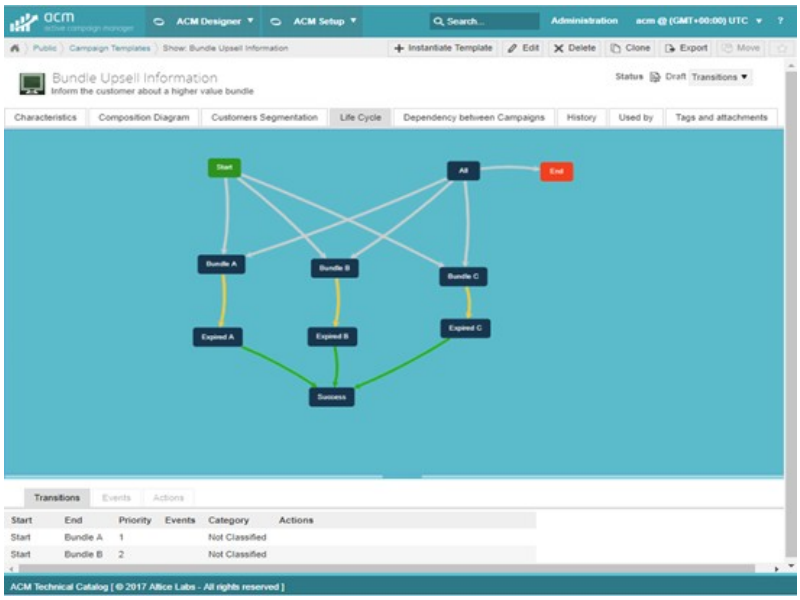


Figura 2.1: Interface do desenhador de campanhas do ACM.

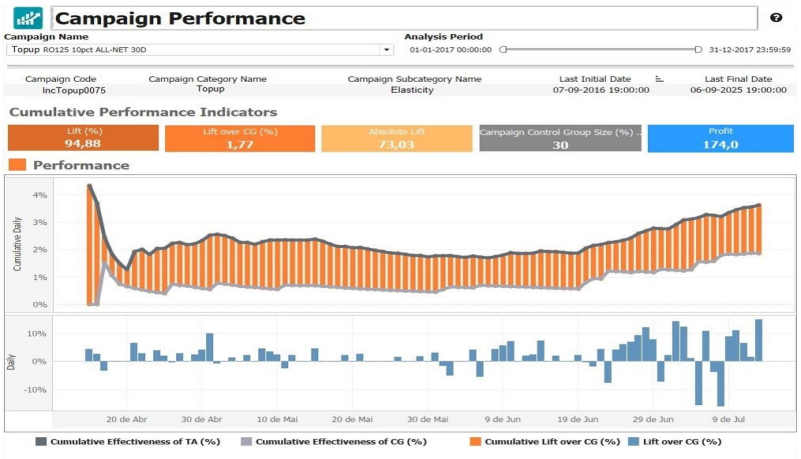


Figura 2.2: Módulo de relatórios sobre campanhas do ACM.

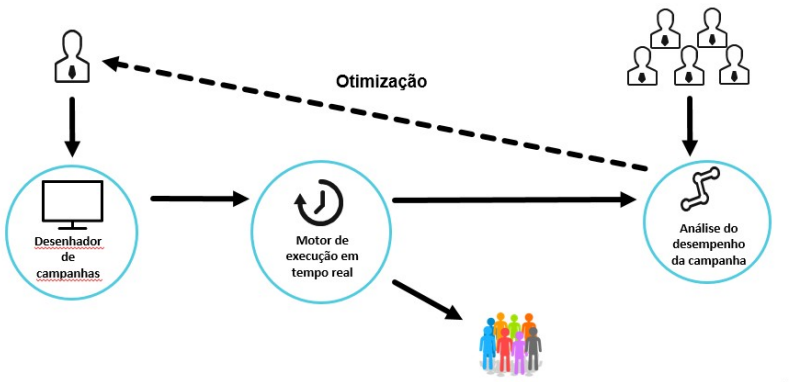


Figura 2.3: Ciclo de desenho e monitoramento de uma campanha no ACM.

Na Figura 2.3 é possível observar o ciclo de uma campanha no ACM, deste a sua criação, passando pelo lançamento aos clientes, pela sua monitorização e, caso seja necessário, a sua otimização.

2.2 Tipos de campanhas

As campanhas enviadas pelo ACM podem chegar aos clientes via SMS, via *Interactive Voice Response* (IVR) ou via e-mail. Consoante o objetivo que se pretende ter perante o cliente, as campanhas podem consistir em vários tipos e oferecer vários benefícios. Os tipos de campanha com que o ACM lida podem ser de:

- **Incentivo à recarga**: o cliente é incentivado com benefícios caso carregue o telemóvel com determinada quantia;
- **Incentivo ao consumo**: o cliente é incentivado com benefícios caso efetue um determinado número de ações;
- **Incentivo à subscrição de um serviço**: o cliente é incentivado com benefícios caso adira a um determinado serviço;
- **Retenção de clientes**: quando um cliente não demonstra uma atividade considerada normal dentro dos parâmetros da operadora de telecomunicações, é incentivado com benefícios que visam aumentar a sua atividade e, portanto, evitar o *churn*[1].

2.3 Público-alvo e Grupo de Controlo

O ACM utiliza os conceitos de público-alvo e de grupo de controlo para avaliar o desempenho das campanhas perante os clientes.

O público-alvo de uma campanha consiste nos clientes aos quais esta é direcionada. Os clientes pertencem ao público-alvo de uma campanha caso respeitem determinadas condições, criadas através de expressões [2]. Através destas, são escolhidos os clientes que se consideram adequados para aderir à campanha. O público-alvo varia, naturalmente, de campanha para campanha, dependendo da sua natureza.

O grupo de controlo trata-se do grupo de clientes que não são incentivados com nenhuma campanha, mas cujo comportamento é monitorizado. A razão dessa monitorização é para que se possa comparar esse comportamento com o dos clientes do público-alvo e, a partir disso, poder estimar o sucesso de uma campanha. Caso o grupo de controlo tenha um comportamento semelhante ao público-alvo, então tal indica que os incentivos às campanhas enviados aos clientes do público-alvo não estão a ter muito impacto.

A segmentação destes dois grupos é feita aquando da execução da campanha, onde, dos clientes aptos para a campanha, é definida a percentagem que irá pertencer ao grupo de controlo e a percentagem que irá pertencer ao público-alvo.

[1] *Churn* é o nome dado ao abandono, por parte de um cliente, dos serviços de uma operadora de telecomunicações.

[2] Por exemplo: perfil de cliente == A AND arpu > 3 euros

2.4 Métricas do ACM

O ACM possui determinadas métricas que permitem medir a eficácia das campanhas e qual o impacto produzido nos clientes, possibilitando a identificação de melhorias que possam ser realizadas. Estas métricas são denominadas de eficácia e *uplift*. A métrica de eficácia é calculada tanto para o público-alvo, como para o grupo de controlo, através da seguinte fórmula:

$$Eficácia = \frac{PA\_Positivos}{PA} \times 100 \tag{2.1}$$

onde PA diz respeito ao tamanho do público-alvo e $PA\_Positivos$ e à quantidade de clientes do público-alvo que aderiu à campanha. O *uplift* compara as eficácias do público-alvo e do grupo de controlo:

$$Uplift = (\frac{PA\_Positivos}{PA} \times 100) - (\frac{GC\_Positivos}{GC} \times 100) \tag{2.2}$$

Analisando a equação do *uplift*, verifica-se que um valor negativo é possível (e não desejável), significando que o impacto da campanha é negativo tendo em consideração o PA *versus* o GC.

Capítulo 3

Noções de Sistemas de Recomendação

Neste capítulo são apresentados alguns conceitos teóricos transversais aos sistemas de recomendação. São apresentados os tipos de sistemas de recomendação que existem, o tipo de dados que estes podem utilizar e métricas de avaliação de sistemas de recomendação.

Sistemas de recomendação são sistemas que filtram a informação dos utilizadores de determinado serviço, tal como o comportamento sob determinado produto, e têm a capacidade de sugerir outros produtos que possam ser de interesse para o utilizador [9]. Estas sugestões visam apoiar os utilizadores no processo de tomada de decisões, tais como que produtos comprar, que música ouvir ou que notícias ler [3]. A informação utilizada por um sistema de recomendação pode consistir em *likes*, *dislikes* e *ratings* atribuídos a um item por um utilizador, ou por indicadores não tão explícitos tais como o tempo que um utilizador passa numa página ou o número de cliques feitos em determinado produto [10].

Existem hoje em dia diversas empresas e serviços que utilizam sistemas de recomendação para facilitar a compreensão do que é que os seus utilizadores possam ter mais interesse para lhes poderem fazer sugestões dos seus produtos, melhorando a relação entre utilizador e empresa. Alguns dos casos mais populares falados neste relatório são: *Netflix*, para a recomendação dos filmes ou séries, a *Amazon*, para a recomendação de produtos, o *Spotify*, para a recomendação de músicas e o *Youtube* que sugere vídeos aos utilizadores. A importância dos sistemas de recomendação para o negócio é tal que, de acordo com a *Netflix*, cerca de 75% a 80% das suas visualizações provêm de recomendações de produtos, 35% da receita da *Amazon* provém do seu motor de recomendações e mais de 70% do que é visualizado no *Youtube* é recomendado através dos seus algoritmos de recomendação [11] [12] [13].

3.1 Tipos de Sistemas de Recomendação

Baseado na maneira como filtram a informação disponível e realizam as sugestões, os sistemas de recomendação classificam-se em diversas categorias. Nas secções seguintes estão descritas as principais, sendo elas as seguintes: sistemas baseados em conteúdo (*content-based*), sistemas de filtragem colaborativa (*collaborative filtering*), sistemas híbridos e sistemas *context-aware*[14].

3.1.1 Sistemas Baseados em Conteúdo

Sistemas baseados em conteúdo recomendam itens que são similares aos que o utilizador achou interessantes no passado, por meio de um *rating*, de uma pesquisa, de uma compra, etc.. A similaridade dos itens é calculada através das características que os itens que estão a ser comparados possuem. Geralmente, sistemas de recomendação baseados em conteúdo utilizam 3 abordagens:

- Desenho do perfil, que pode ser obtido explicitamente através de questionários acerca das preferências do utilizador dos itens;
- Perfis de utilizador, que podem ser construídos de forma implícita, analisando as semelhanças entre itens que o utilizador gostou ou não gostou;
- Modelos de utilizador, que podem ser aprendidos por um método automático de aprendizagem, que usa descrições dos itens como *input* e devolve apreciações dos itens feitas pelos utilizadores como *output*;

De maneira a melhorar a sua eficácia, os sistemas baseados em conteúdo necessitam de descrições bastante completas dos itens e perfis de utilizador construídos adequadamente [15].

3.1.2 Sistemas de filtragem colaborativa

Em sistemas de filtragem colaborativa, as recomendações são realizadas através da identificação de outros utilizadores com um gosto semelhante. São recomendados itens que um utilizador com gostos similares também tenha demonstrado interesse, ou seja, assenta na ideia de que, se os interesses do utilizador x e y são similares, os itens que o utilizador y demonstra interesse podem ser recomendados ao utilizador x. Esta técnica é considerada a mais popular e a mais implementada.

É denominado de "vizinhança" o conjunto de utilizadores que partilhem os mesmos interesses. O cálculo da similaridade entre utilizadores "vizinhos" pode ser realizado através das seguintes métricas:

- Correlação de *Pearson*, em que a similaridade entre dois utilizadores a e u é dada por [9]:

$$s(a,u) = \frac{\sum_i (r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_i (r_{a,i} - \overline{r_a})^2}\sqrt{\sum_i (r_{u,i} - \overline{r_u})^2}} \tag{3.1}$$

  onde s(a,u) representa a similaridade entre dois utilizadores a e u, $r_{a,i}$ é o *rating* atribuído ao item i pelo utilizador a, $\overline{r_a}$ é a média de *ratings* dados pelo utilizador a.

- Medida do Cosseno, em que a similaridade entre dois utilizadores a e u obtém-se através do cálculo do ângulo entre 2 vetores correspondentes a cada um dos utilizadores a ser comparado. É dada pela fórmula :

$$s(a,u) = \sum_i \frac{r_{a,i}}{\sqrt{\sum_{k \in I_a} {r^2}_{a,k}} \times \sqrt{\sum_{k \in I_u} {r^2}_{u,k}}} \tag{3.2}$$

A medida do cosseno é habitualmente usada para o cálculo da similaridade entre itens, por exemplo documentos, onde cada documento é tratado como um vetor de frequências de palavras. Contudo, com algumas alterações na fórmula e considerando que os utilizadores representam os documentos e os *ratings* representam as frequências de cada palavra, consegue-se adaptar esta métrica para uma abordagem de filtragem colaborativa [16].

As recomendações são realizadas com base na previsão de um *rating* que o utilizador possa atribuir a determinado item. Um *rating* acima de um nível previamente definido para um item significa a sua recomendação ao utilizador pois espera-se que seja um *rating* próximo daquele o utilizador daria ao item. O cálculo da previsão é feito a partir da soma ponderada dos *ratings* de utilizadores vizinhos. A fórmula de previsão generalizada é a seguinte:

$$p(a,i) = \overline{r_a} + \frac{\sum_i (r_{u,i} - \overline{r_u}) \times s(a,u))}{\sum_i s(a,u)} \tag{3.3}$$

em que p(a,i) denota a previsão de um *rating* dado pelo utilizador a a um item i e $s_{a,u}$ é o semelhança calculada entre os utilizadores a e u.

A Figura 3.1 ilustra a diferença entre sistemas baseados em conteúdo e sistemas de filtragem colaborativa. Tal como explicitado na figura, em sistemas baseados em conteúdo são recomendados ao utilizador itens que são similares a outros itens que ele já tenha visto, comprado, gostado, etc. Em sistemas de filtragem colaborativa são recomendados itens que utilizadores semelhantes tenham visto. Na figura, como os dois utilizadores são semelhantes por já terem lido o mesmo livro, é recomendado a ele um livro que ele ainda não leu mas que ela já.

Nos próximas sub-secções estão descritos com mais detalhe dois tipos de filtragem colaborativa que existem.

3.1.2.1 Filtragem Colaborativa baseada nos utilizadores

Algoritmos que utilizem este tipo de Filtragem Colaborativa encontram utilizadores cujo histórico de *ratings* seja similar ao do utilizador ativo (U_a), ou seja, utilizadores da vizinhança de U_a, e utilizam esses *ratings* de outros itens para prever o que U_a possa gostar [17]. Um exemplo de um algoritmo deste tipo é o algoritmo *k-Nearest Neighbour* ou k-NN.

Pode ser utilizado um limite de *vizinhança* que indica quais os utilizadores que pertencem à vizinhança de U_a e quais estão fora. Desta forma alguns utilizadores poderão receber melhores recomendações que outros, por terem mais utilizadores na sua vizinhança, mas também garante que não existem utilizadores esquecidos por existirem sempre outros com gostos muito mais semelhantes aos do U_a, aproveitando assim a informação de todos eles [18].

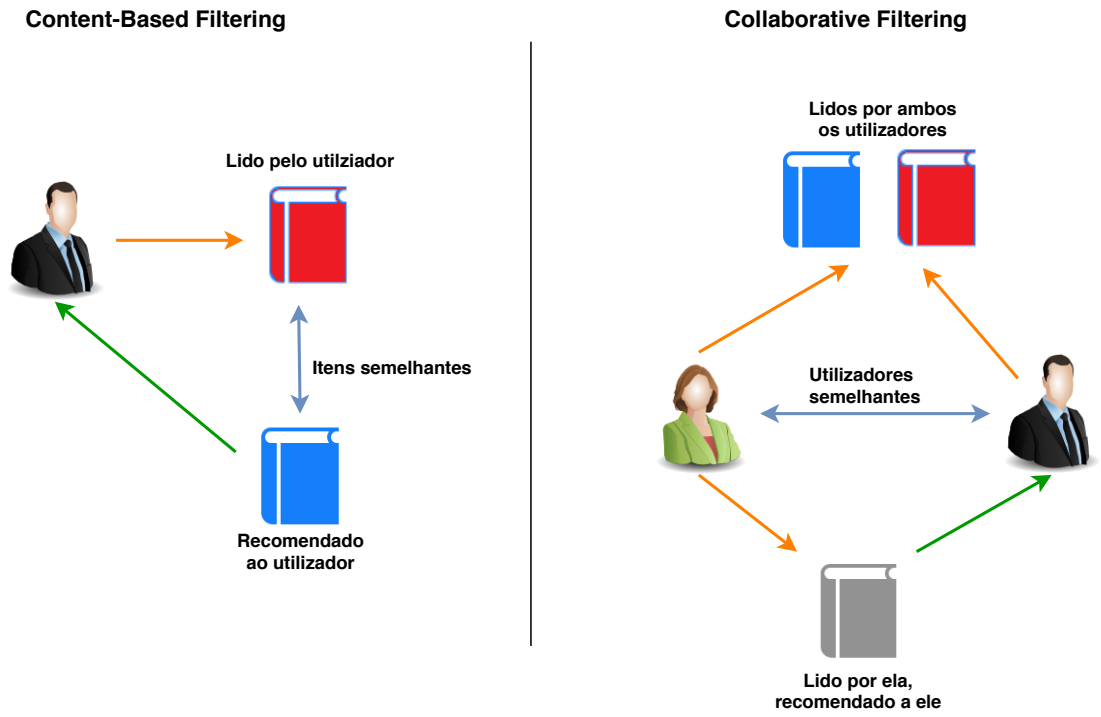


Figura 3.1: Diferença entre sistemas baseados em conteúdo e sistemas de filtragem colaborativa, adaptado de [1].

3.1.2.2 Filtragem Colaborativa baseada nos itens

Neste tipo de Filtragem Colaborativa, ao contrário do que acontece com algoritmos baseados nos utilizadores, considera-se a similaridade entre diferentes itens olhando para quantos utilizadores demonstraram interesse pelo item X e também pelo item Y. Se a correlação for suficientemente alta, considera-se que existe uma similaridade entre os itens e a partir daí, o item Y é recomendado a utilizadores que compraram ou deram um bom *rating* ao item X e vice-versa [17].

Este técnica pode parecer idêntica a métodos utilizados em sistemas baseados em conteúdo, porém, neste caso, a similaridade entre itens é deduzida das preferências de utilizadores e não calculada a partir das características dos itens.

A imagem 3.2 ajuda a perceber a diferença entre os dois diferentes tipos de filtragem colaborativa.

No que toca a filtragem colaborativa baseada nos utilizadores, vemos na imagem que os itens que são recomendados ao utilizador 3 são aqueles que o utilizador 1, semelhante a ele, comprou e o utilizador 3 ainda não comprou. Os utilizadores 1 e 3 são semelhantes pois adquiriram produtos em comum. Relativamente a filtragem colaborativa baseada nos itens, a recomendação realizada é de um item semelhante ao que o utilizador 3 adquiriu. No exemplo, o ananás e as uvas são considerados itens semelhantes pois foram comprados pelos utilizadores 1 e 2 em simultâneo. Uma vez que o utilizador 3 também comprou o ananás, é de esperar que também tenha interesse em comprar as uvas.

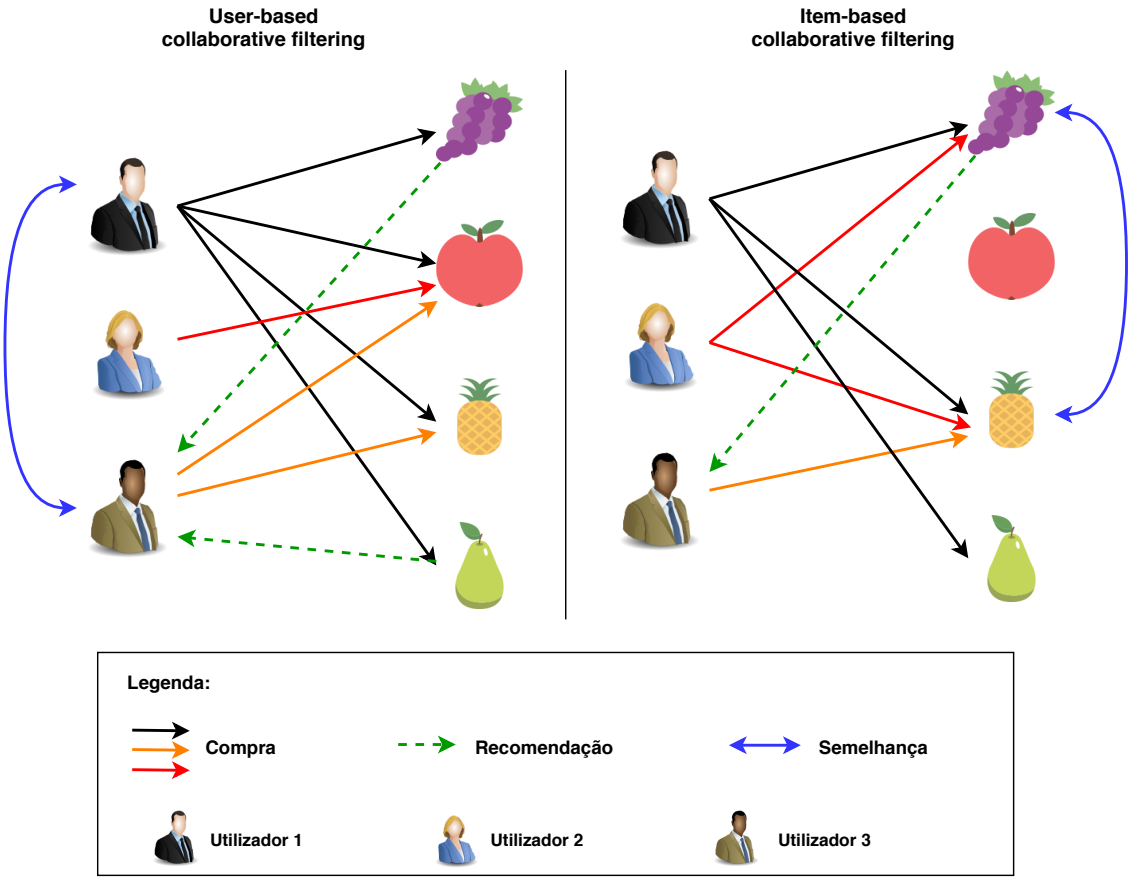


Figura 3.2: Diferença entre sistemas de filtragem colaborativa baseada nos utilizadores e sistemas de filtragem colaborativa baseada nos itens, adaptado de [2].

3.1.2.3 Limitações da filtragem colaborativa

Os sistemas de recomendação de filtragem colaborativa revelam algumas limitações tais como:

- **Problema da Escalabilidade (*the Scalability Problem*)**: um sistema de recomendação pode ser usado por milhões de utilizadores e possuir milhões de itens que podem ser recomendados. É importante que a complexidade computacional dos algoritmos utilizados seja escalável com o número de utilizadores e itens que o sistema comporta [19]. É possível lidar com este problema através da utilização da técnica de Decomposição em Valores Singulares (*Singular Value Decomposition (SVD)*) [9].
- **Problema da Esparsidade (*the Sparsity Problem*)**: este problema surge como resultado da falta de informação no sistema, mais especificamente, quando poucos dos itens presentes na base de dados do sistema têm um *rating* a si atribuídos. Este problema é comum numa fase inicial quando ainda o número de utilizadores é pequeno. Uma forma de o resolver é usar informação do perfil do utilizador quando se determina a semelhança entre utilizadores. Por exemplo, dois utilizadores podem ser considerados semelhantes, não apenas por terem avaliado os mesmos filmes, mas também por pertencerem ao mesmo segmento demográfico [20]. Estas extensões das técnicas tradicionais de Filtragem Colaborativa são por vezes designadas por "Filtragem Demográfica".
- **Problema da Inicialização (*the Cold-Start Problem*)**: ocorre quando um uti-

lizador ou um item são novos no sistema e portanto existe pouca informação para que se possam fazer recomendações eficazes. Quando se trata de um utilizador novo, o sistema não possui informação acerca das suas preferências pois ainda não avaliou um número de itens suficientes, o chamado problema do novo utilizador. O problema do novo item ocorre quando existem itens recentes no sistema que ainda não possuem um número de *ratings* considerável para que possam ser recomendados. Possíveis formas de resolver o problema do novo utilizador são, aquando da sua entrada no sistema, ser-lhe pedido que avalie um conjunto de itens ou responda a questões demográficas, sendo depois utilizados estereótipos para inferir os seus gostos. Este método implica que haja esforço adicional por parte do utilizador e pode resultar em recomendações pouco eficazes uma vez que os estereótipos podem ser bastante errados e até ofensivos [3]. Outras alternativas baseadas na popularidade de itens, entropia dos itens, personalização do utilizador e combinação de todas estas foram também exploradas em [21]. Contudo, ambos os problemas de novo utilizador e novo item podem ser endereçados através de abordagens de recomendação híbrida, descritas na secção seguinte.

3.1.3 Sistemas híbridos

Os sistemas de recomendação híbridos são sistemas que resultam da junção de duas ou mais técnicas de recomendação. A ideia geral é que a combinação dos algoritmos irá gerar recomendações mais precisas e eficazes do que aquelas geradas por apenas um algoritmo, uma vez que as desvantagens de um deles podem ser ultrapassadas pelo outro [9]. O que acontece mais regularmente é a combinação de técnicas de sistemas baseados em conteúdo e de sistemas de filtragem colaborativa, que beneficia da inexistência do problema de *cold-start* em sistemas baseados em conteúdo, nomeadamente do problema do novo item. Quando tal ocorre, um sistema de recomendação híbrido pode usar texto de descrição de um item para o comparar com outro, através do cálculo da similaridade, permitindo que possam ser recomendados apesar de serem novos no sistema.

Existem diferentes métodos de combinação das técnicas de recomendação num sistema híbrido [22]. Alguns desses métodos são os seguintes:

3.1.3.1 Método *Weighted*

O valor dado às previsões de um *rating* a um item é obtido a partir dos resultados de todas as técnicas de recomendação presentes no sistema. Estes sistemas atribuem um determinado peso às recomendações provenientes das técnicas de filtragem colaborativa e outro peso às técnicas baseadas em conteúdo, como é o caso do sistema P-Tango [23]. Para além disso, este sistema ainda vai gradualmente ajustando os pesos dados às técnicas conforme os *ratings* reais dados pelos utilizadores.

3.1.3.2 Método *Switching*

Neste método, o sistema utiliza certos critérios para trocar entre técnicas de recomendação. A técnica utilizada no final será a que obteve melhores resultados, num determinado contexto. O sistema de recomendação do *DailyLearner* por exemplo, usa inicialmente uma técnica baseada em conteúdo, caso o sistema não consiga fazer recomendações com uma determinada confiança, então uma técnica de filtragem colaborativa é utilizada.

3.1.3.3 Método *Mixed*

As recomendações provenientes das diferentes técnicas são misturadas e apresentadas ao mesmo tempo na mesma lista. Um sistema que faz uso deste método é o sistema PTV [24], que utiliza técnicas baseadas em conteúdo nas descrições textuais dos programas de televisão e informação proveniente de filtragem colaborativa para determinar as preferências dos utilizadores. As recomendações obtidas das duas técnicas são combinadas no final.

3.1.3.4 Método *Feature Combination*

Este método utiliza as previsões calculadas por uma das técnicas como um atributo ou característica da outra técnica, por exemplo, utiliza informação proveniente da técnica de filtragem colaborativa como um atributo adicional de cada item e depois utiliza técnicas baseadas em conteúdo nessa informação aumentada, uma vez que a última técnica analisa as características representativas do item.

3.1.3.5 Método *Cascade*

Uma das técnicas de recomendação é utilizada para aperfeiçoar as recomendações obtidas pela outra. A aplicação da segunda técnica é realizada para efeitos de "empate" entre itens recomendados ou para aqueles que ainda precisem uma distinção adicional. Desta forma o sistema evita aplicar a segunda técnica em itens que já estão bem diferenciados dos outros ou naqueles que foram suficientemente mal avaliados com um *rating*. Por essa razão, este método é mais eficaz do que por exemplo o método *weighted*.

3.1.4 *Context-aware systems*

Os sistemas de recomendação clássicos baseiam as suas recomendações nas preferências e gostos dos utilizadores, tal como já referido anteriormente, mas ignoram a importância de perceber em que situação os utilizadores consomem os itens recomendados [25]. Sistemas baseados em contexto ou *context-aware systems* utilizam informação contextual no processo de recomendação para que este tenha em conta as circunstâncias em que o utilizador se encontra. Por exemplo, utilizando o contexto temporal, um sistema de recomendações de viagens irá recomendar ao utilizador determinados destinos no inverno e diferentes destinos no verão. O objetivo destes tipos de sistemas não é apenas utilizar informação contextual, mas sim incorporá-la com a restante informação que o sistema possa utilizar nas suas recomendações para melhorar a sua precisão e eficácia.

Desta forma, as preferências do utilizador são modeladas em função dos utilizadores, itens e contexto e a função de previsão dos *ratings* é definida como R: $Utilizador \times Item \times Contexto \rightarrow Rating$, onde *Utilizador* e *Item* são os domínios dos utilizadores e itens, respetivamente, e *Contexto* é a informação contextual que define os domínios referidos.

Os passos de um típico *context-aware system* são os seguintes:

1. **Recolha de dados** - nesta fase é recolhida a informação relativa às preferências dos utilizadores pelos itens que o sistema irá recomendar, e a informação contextual do utilizador enquanto este expressa as suas preferências. Para esta última, é gerado um perfil de utilizador baseado no contexto para as tarefas de previsão dos *ratings*, proposto em [3].

Os tipos de informação contextual recolhida podem dizer respeito aos seguintes tópicos:

- **contexto do utilizador**, que representa o seu perfil, local, demografia, atividade atual, emoções e pessoas por perto;
- **contexto físico**, que representa o tempo, a posição, meteorologia, luz e temperatura nos arredores;
- **contexto social**, que fornece informação como o papel que outras pessoas à volta do utilizador desempenham e se o utilizador está ou não sozinho aquando da utilização do sistema;
- **contexto de interação com meios de comunicação**, que descreve o dispositivo no qual o utilizador está a aceder ao sistema;
- **contexto modal** que representa o objetivo, a experiência, o estado de espírito e capacidades cognitivas do utilizador.

A informação de contexto recolhida deve ser a mais relevante possível para que não seja adicionado ao sistema informação "ruído" que prejudique a sua performance e precisão [26]. Para tal, é feita uma filtração dos atributos contextuais considerados com a finalidade de remover informação irrelevante e redundante.

2. **Geração do modelo de *rating*** - a geração do modelo de *rating* pode ser representada em duas formas, numa estrutura hierárquica como uma árvore, em que cada uma dessas estruturas tem um tipo particular de contexto, ou num modelo multidimensional, onde as suas dimensões são o domínio dos utilizadores, o domínio dos itens e as restantes dimensões a informação contextual em que cada uma delas é um subconjunto do produto cartesiano de alguns atributos que têm um ou mais valores. Para clarificar, vemos o exemplo da Figura 3.3, em que os *ratings* são armazenados para o espaço de recomendação *Utilizador* $\times$ *Item* $\times$ *Tipo de dia*, onde as três tabelas definem um conjunto de utilizadores, itens e tipos de dia associados às dimensões de *Utilizador*, *Item* e *Tipo de dia*, respetivamente. O *rating* $R(101; 7; 1) = 6$ por exemplo significa que o utilizador com ID = 101 deu um *rating* de 6 ao item com ID = 7 durante um dia da semana (definido como *weekday* na figura).
3. **Construção da recomendação** - tal como noutros tipos de sistemas de recomendação, as recomendações podem ser uma previsão, um *rating* previsto para um item dado por um utilizador baseado em informação contextual fornecida, ou uma recomendação, uma lista ordenada dos itens que o utilizador mais gostará. Por outras palavras, a construção de uma recomendação consiste na geração de uma lista de recomendações através da previsão de valores em falta. Essa previsão é baseada na informação de contexto do utilizador. Os itens recomendados são aqueles que obtiveram uma melhor classificação como *rating* previsto, depois dos itens terem sido ordenados conforme esse valor.
4. **Métricas de avaliação de *context-aware recommender systems*** - Ao contrário de outros sistemas, a avaliação de algoritmos *context-aware* é diferente pois o contexto é adicionado ao processo de avaliação como *input* adicional. Mais à frente neste relatório serão apresentadas as métricas de avaliação frequentemente utilizadas em sistemas de recomendação.

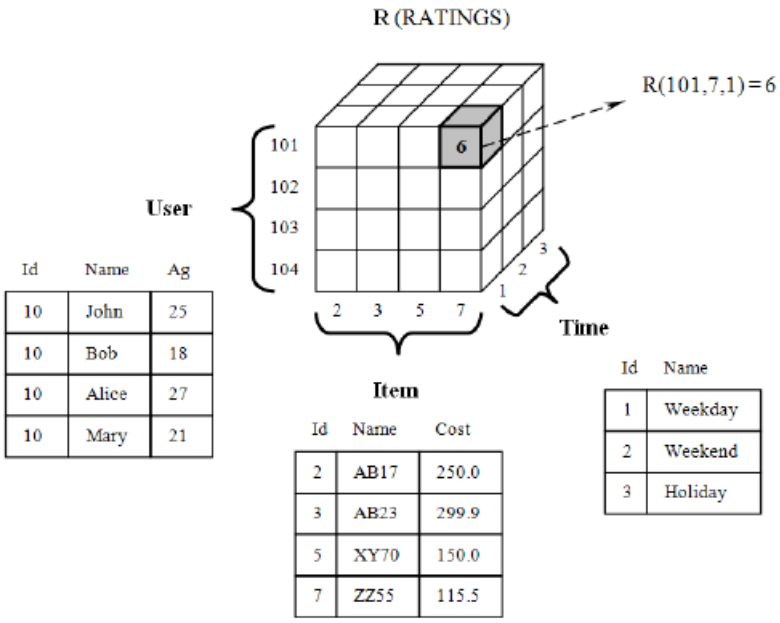


Figura 3.3: Modelo multi-dimensional para o espaço de recomendação *Utilizador* × *Item* × *Tipo de dia*, retirado de [3].

3.2 Tipos de *feedback*

Os sistemas de recomendação necessitam de conhecer informação acerca dos utilizadores para que seja possível a construção do modelo ou perfil de utilizador, fundamentais para as recomendações realizadas pelo sistema. A informação recolhida, que consiste nos *inputs* que o sistema recebe, pode-se classificar em dois tipos [9][27]:

- ***Feedback* implícito**, que consiste em inferir as preferências do utilizador através da análise da sua atividade no sistema. Esta informação é registada automaticamente pelo sistema e pode consistir em histórico de compras, históricos de pesquisas, padrões de pesquisa, tempo gasto em determinadas páginas *web* e até cliques do rato.
- ***Feedback* explícito**, em que o sistema pede ao utilizador para que atribua *ratings* a itens.
- ***Feedback* híbrido**, onde é feita uma combinação das duas técnicas anteriores, a fim de melhorar a *performance* do sistema. Tal pode ser obtido, por exemplo, através do uso de *feedback* implícito como verificação do *feedback* explícito, fornecido pelo utilizador.

A informação que é explicitamente recolhida consiste em informação que será mais confiável e mais transparente, uma vez que se trata de informação dada diretamente pelo utilizador e que reflete as suas preferências. A não ser que o utilizador tenha atribuído *ratings* falsos que não estão de acordo com os seus interesses, a eficácia das recomendações dadas usando este tipo de *feedback* será alta. No entanto, a recolha de *feedback* explícito pode nem sempre ser possível, uma vez que é necessário esforço por parte do utilizador na atribuição de um *rating*, algo que nem todos estão dispostos a ter, ou porque o próprio sistema não permite realizar tal recolha. Daí a escolha por *feedback* implícito, em que a partir do

momento que um utilizador use o sistema e aceite que seja feita a recolha de informação da sua utilização, mais nenhum esforço adicional será necessário da sua parte.

3.3 Métricas de avaliação de sistemas de recomendação

A qualidade das recomendações de um sistema de recomendação pode ser avaliada no que toca à sua precisão, ou seja, a fração de recomendações corretas do conjunto total de recomendações feitas. Esta avaliação pode ser feita através da comparação das recomendações que o sistema realizou com um *test set* de *ratings* conhecidos. Métricas que utilizam métodos estatísticos para tal são a Mean Absolute Error (MAE) e a Root Mean Squared Error (RMSE) [28].

A MAE é a mais popular e mais usada e consiste na média das diferenças entre o valor da previsão e o valor de teste e é computada da seguinte fórmula:

$$MAE = \frac{\sum_{u,i} |p_{u,i} - r_{u,i}|}{N} \tag{3.4}$$

onde $p_{\mathrm{u,i}}$ é a previsão do *rating* atribuído ao item i pelo utilizador u e $r_{\mathrm{u,i}}$ o *rating* real atribuído.

A RMSE também mede a média da magnitude dos erros, através da seguinte fórmula:

$$RMSE = \sqrt{\frac{\sum_{u,i}(p_{u,i} - r_{u,i})^2}{N}} \tag{3.5}$$

Ambas as métricas expressam o erro entre valores de teste e valores atuais previstos, portanto quanto menor o seu valor, melhor os resultados. No entanto, como a RMSE aplica o quadrado aos erros, é dado um peso mais alto a erros de maior valor, portanto, a RMSE deverá ser mais útil em casos onde valores de erro muito grandes não são desejáveis. Isso poderá ser o pretendido na avaliação de um sistema de recomendação. Apesar disso, ambas as métricas devem ser utilizadas como termo de comparação e tiradas as devidas conclusões [29].

O foco principal na avaliação de sistemas de recomendação é determinar o valor da lista de recomendações obtida, ou seja, avaliar se o utilizador estará interessado em algumas das recomendações dessa lista. Para tal, podem ser utilizadas métricas como *precision* e *recall* [30].

Em sistemas que devolvem uma lista de recomendações, existe a necessidade de adaptar a definição tradicional de *precision* e *recall* e utilizar a terminologia utilizada em *Information Retrieval* (IR). De modo a calcular estas duas métricas, é necessário dividir o *dataset* em dados de treino (*training set*) e dados de teste (*test set*). Os dados de teste correspondem aos dados que servem para avaliar as recomendações, em oposição ao *Training Set*, ou conjunto de dados de treino, que são os dados que o algoritmo utiliza para gerar as recomendações.

A lista de recomendações obtida será aqui denominada de *Top-N set*. A ideia é corresponder os itens que estão presentes no *test set*, que não é utilizado pelo sistema de recomendações, com os itens que estão presentes no *Top-N set* obtido. Itens que aparecerem nos dois conjuntos serão itens pertencentes a um conjunto chamado *hit set*, utilizando a terminologia de IR.

Sendo assim, é possível definir a métrica *precision* como a fração de itens recomendados que são realmente relevantes para o utilizador [31]. A sua fórmula é:

$$Precision = \frac{SizeHitSet}{SizeTopNSet} \tag{3.6}$$

onde $SizeHitSet$ é o tamanho do conjunto *Hit Set*, ou seja, o número de itens recomendados corretamente e $SizeTopNSet$ o tamanho do conjunto *Top-N set* , ou seja, o número total de itens recomendados.

A métrica *recall* diz respeito à fração de itens relevantes que fazem parte do conjunto de itens recomendados [9], e obtém–se pela fórmula

$$Recall = \frac{SizeHitSet}{SizeTestSet} \tag{3.7}$$

onde $SizeHitSet$ corresponde novamente ao número de itens recomendados corretamente e $SizeTestSet$ corresponde ao número total de itens relevantes recomendados.

O problema com estas duas métricas é que a *recall* tende a aumentar quando o número total de itens recomendados aumenta, enquanto que a *precision* decresce na mesma situação. Como é desejável obter um valor alto de *precision* e de *recall* em simultâneo, pode–se usar a métrica *F1-measure* que junta as outras duas métricas.

O valor da métrica *F1-measure* é dado pela fórmula:

$$F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.8}$$

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Estado de arte

Neste capítulo são mostrados exemplos de sistemas de recomendação na indústria e na academia. É apresentada uma descrição das metodologias e técnicas que estes sistemas utilizam para realizar as suas recomendações.

4.1 Exemplos de sistemas de recomendação

4.1.1 Netflix

A *Netflix*, empresa que faz *stream* de séries televisivas e filmes, foi um dos grandes impulsionadores do aperfeiçoamento dos sistemas de recomendação, nomeadamente dos de filtragem colaborativa. Isto deveu-se à organização de uma competição em 2006 onde a empresa ofereceu 1 milhão de dólares à equipa que conseguisse desenvolver um algoritmo que melhorasse o que a *Netflix* usava na altura, o *Cinematch*, algo que efetivamente aconteceu resultando num algoritmo que melhorou em 10% as recomendações realizadas [32]. Mais especificamente, a solução obtida apresentou um valor de RMSE de 0.8914 comparativamente ao valor de 0.9525 já obtido pela *Netflix*.

Apesar da melhoria que a solução da equipa *BellKor's Pragmatic Chaos* revelou, esta nunca foi implementada pois a empresa percebeu que os ganhos em eficácia das previsões que ia obter não justificariam o esforço computacional e de engenharia necessários para colocar a solução no ambiente de produção [33]. Saber avaliar os *trade-offs* de uma determinada solução é fulcral para perceber se esta deve ser implementada, uma vez que serão gastos recursos no seu desenvolvimento. Naturalmente, no contexto de uma empresa e da sua perspetiva de negócio, o desperdício de recursos deve ser o mínimo possível e é necessário, por vezes, optar entre performance das soluções e custos associados a estas, como foi este o caso.

No que toca às recomendações de filmes ou séries feitas pelo serviço de *streaming* da *Netflix*, estas estão organizadas em linhas. Para cada uma dessas linhas, um algoritmo diferente (tipicamente um algoritmo por linha) atua para ordenar as sugestões dadas ao utilizador. Alguns exemplos são o *Personalized Video Ranker*, que mostra vídeos de um determinado género, mas personalizados para determinado perfil de utilizador, significando que utilizadores diferentes poderão ser recomendados com filmes ou séries diferentes dentro do mesmo género, o *Trending Now* que recomenda vídeos que poderão ser mais vistos ou procurados durante um determinado espaço de tempo que pode ir de alguns minutos até alguns dias, algo que combinado com alguma personalização pode oferecer uma boa

previsão do que é que os utilizadores estarão interessados em assistir e o *Continue Watching* que recomenda os vídeos que o utilizador poderá querer voltar a ver ou rever, tendo em conta *feedback* implícito como o tempo que passou desde que o utilizador viu a série pela última vez, se parou de ver no início, no fim ou a meio, que outros vídeos viu entretanto [11].

Como todas as recomendações necessitam de uma forma de *ranking*, é preciso definir um modelo de *ranking* que ordene sugestões. Para tal, a *Netflix* utiliza a popularidade dos itens e previsões de *ratings*, o que junta a ideia de ser mais provável um utilizador ver o que outros estejam a ver, com previsões de *ratings* que ele possa dar a determinado programa. Este modelo de *rankings* associado à disponibilidade de grandes quantidades de informação proveniente dos utilizadores e até de fontes externas, que fornecem por exemplo informação acerca das críticas a um filme ou dos seus lucros de bilheteira, são extremamente importantes para a personalização das recomendações. No que toca à seleção, treino e teste dos modelos, várias técnicas de *machine learning* são utilizadas desde métodos de *unsupervised learning* como *clustering* e métodos de *supervised learning* como classificadores e modelos de regressão, que mostraram um número favorável de resultados [34].

4.1.2 Amazon

O sistema de recomendações da *Amazon*, gigante no mercado de venda *online* de produtos a retalho, acenta em quatro aspetos: itens comprados no passado por um utilizador, itens que ele possua no seu carrinho de compras digital, itens que ele gostou ou atribuiu um *rating* e o que outros utilizadores tenham visto ou comprado [35].

De acordo com a *Amazon*, o algoritmo de recomendação utilizado personaliza altamente o *site* de acordo com os interesses dos utilizadores. Este foi desenvolvido pela própria empresa e é denominado de Filtração Colaborativa Item-a-Item (*Item-to-Item Collaborative Filtering*) [36]. O algoritmo foca-se em encontrar itens similares, ao invés de utilizadores similares. Para cada um dos itens comprados e classificados (com um *rating*), tenta encontrar itens semelhantes, agrega-os e recomenda-os.

Este processo é realizado da seguinte maneira: é feita a correspondência entre cada item que o utilizador mostrou interesse (comprou ou atribuiu um *rating*) a outros itens semelhantes, colocando-os depois numa lista de recomendação. A similaridade dos itens é calculada pela pesquisa de produtos que são comprados em pares, o que resulta numa tabela de itens similares. A partir desta tabela, o algoritmo encontra itens semelhantes aos que o utilizador mostrou interesse, agrega-os todos e recomenda aqueles que forem mais populares ou tiverem mais correlações com outros [36].

O pseudo-código seguinte mostra, de forma simplificada como este processo é realizado:

```
for each product A in product catalog do
    for each user C who bought A do
        for each item B bought by C do
            record that a customer purchased A and B in matrix
        end
    end
    for each item B do
        compute the Cosine Similarity between A and B
    end
end
```

O algoritmo regista na tabela de itens semelhantes os que foram comprados juntamente com outros itens. É depois calculada a similaridade entre esses pares de itens, através da métrica *Medida do Cosseno*, onde cada item é representado como um vetor e cujas dimensões correspondem ao número de clientes que os comprou. As recomendações são feitas com base nos valores desta métrica.

Porém, a *Amazon* deu a conhecer este algoritmo já há alguns anos. É normal que uma empresa com a magnitude da *Amazon* tenha melhorado as suas recomendações de produtos usando técnicas mais sofisticadas. De facto, em [37] é referido que a empresa utiliza redes neuronais para a personalização das suas recomendações. Mais especificamente, utilizam a Deep Scalable Sparse Tensor Network Engine (DSSTNE) [38], uma biblioteca desenvolvida pela *Amazon* para construir modelos de *Deep Learning* para recomendações.

4.1.3 Spotify

O serviço de *stream* de música do *Spotify* sugere uma lista de músicas recomendadas aos utilizadores denominada "Descobertas da Semana" (*Discover Weekly*), onde semanalmente são mostradas músicas ao utilizador baseadas no seu histórico de reproduções e nos de outros utilizadores com gostos semelhantes [39]

O *Spotify* consegue criar esta lista de 30 músicas semanalmente através da junção de 3 modelos de recomendação, nomeadamente modelos de Filtragem Colaborativa, modelos de aúdio e modelos de Processamento de Linguagem Natural (PLN) [40].

Os modelos de Filtragem Colaborativa seguem o conceito que já foi referido anteriormente, fazendo as recomendações com base no que outros utilizadores com os mesmos gostos ouçam. Neste caso cada utilizador e cada música são representados como vetores. O algoritmo depois compara-os e recomenda músicas que sejam mais similares entre si, e músicas que outros utilizadores similares também ouçam. Para os modelos de PLN, são usados textos de artigos, *blogs* e outros encontrados na Internet. Descobrindo o que é que as pessoas dizem sobre determinado artista ou música, que tipo de linguagem utilizam e que outros artistas e músicas são mencionados junto com eles, é ser possível determinar a similaridade das músicas com a finalidade de as recomendar a utilizadores. Modelos de áudio analisam as características do som como timbre, ritmo, acústica, vivacidade, etc., para determinar quais as músicas mais parecidas. A adição deste modelo não só melhora a eficácia e precisão das recomendações como também torna possível a recomendação de novas músicas e não apenas as músicas mais populares que outras pessoas tenham ouvido, o que soluciona uma das desvantagens da Filtragem Colaborativa [41].

4.1.4 Youtube

O *Youtube*, uma das maiores plataformas de partilha de vídeos do mundo com cerca 1.9 mil milhões de utilizadores ativos por mês e 5 mil milhões de vídeos vistos por dia [42], utiliza um sistema de recomendações para disponibilizar aos seus utilizadores um conjunto personalizado de vídeos que sejam relevantes aos seus interesses [43]. Dada a magnitude da empresa, torna-se um desafio ter um sistema de recomendação que seja escalável para o número de utilizadores e vídeos que o sistema comporta. A taxa com que os vídeos são carregados para a plataforma torna também imperativo que as recomendações sejam o mais atualizadas possível, tendo em conta quer vídeos recentemente carregados, quer a atividade recente do utilizador [44].

Tal como noutros já referidos, o sistema de recomendação do *Youtube* utiliza duas grandes

classes de dados: (1) metadados dos vídeos, como título e descrição e informação *raw* como duração, largura e comprimento, *frame rate*, etc. e (2) atividade dos utilizadores, que novamente pode ser dividida em implícita e explícita.

É utilizada a técnica de *Association Rule Mining* para determinar qual o vídeo mais provável um utilizador querer ver depois de ter visto outro. Uma explicação simples do processo consiste em: é registado o número de vezes que cada par de vídeos é visto em conjunto e definida uma pontuação de parentesco (*relatedness score*) entre os vídeos. Os vídeos são ordenados de acordo com a pontuação obtida e são recomendados aqueles que fiquem no topo desta lista (o número de vídeos escolhidos no topo pode variar). Como é definido valor mínimo para esta pontuação, poderão existir vídeos que não tenham uma lista de vídeos relacionados [43].

Para o processo de geração de um *ranking* de vídeos a serem recomendados, o sistema possui duas redes neuronais, uma para geração de candidatos, outra para os *rankings*. A primeira utiliza informação do histórico da atividade do utilizador e retorna um conjunto de vídeos relevantes para ele, sendo a similaridade entre os vídeos expressa em termos dos seus ID's, *querys* de pesquisa e dados demográficos. Para obter o *ranking* de vídeos a apresentar, a segunda rede neuronal atribui uma classificação a cada vídeo de acordo com uma função objetivo e utilizando um conjunto de características descritivas de cada vídeo e de cada utilizador. Os vídeos com maior classificação serão os recomendados. A junção das duas redes neuronais permite a recomendação de vídeos pertencentes a uma coleção bastante grande e garante também a personalização das recomendações, ou seja, que estas sejam as mais adequadas para o utilizador.

Resumindo, as quatro secções anteriores apresentaram exemplos de sistemas de recomendação utilizados por grandes empresas. A Tabela 4.1 mostra a categoria desses sistemas de recomendação, o tipo de *feedback* que utilizam e alguns dos algoritmos que fazem parte do seu processo de geração de recomendações.

Tabela 4.1: Visão global de sistemas de recomendação famosos.

	Categoria	**Tipo de feedback**	**Algoritmos**
Netflix	Híbrido	Implícito e Explícito	Clustering, Regressão Linear, Regressão Logística, Fatorização de Matrizes, Cadeias de Markov, Regras de Associação
Amazon	Filtragem Colaborativa	Implícito e Explícito	Redes Neuronais
Spotify	Híbrido	Implícito	Processamento de Linguagem Natural, Fatorização de Matrizes, Latent Space Representation
Youtube	Híbrido	Implícito e Explícito	Redes neuronais, Regras de associação

4.2 Exemplos de estudos e problemáticas discutidas na academia sobre sistemas de recomendação

Nesta secção são apresentados sistemas de recomendação que abordam diferentes conceitos e temáticas que são relevantes para o estudo realizado neste estágio. As sub-secções seguintes mostram exemplos desses sistemas e descrições das metodologias e técnicas que utilizam.

4.2.1 Solução para o Problema da Iniciação e Problema da Esparsidade

Soluções para o problema da Iniciação e para o problema da Esparsidade, problemas comuns aos sistemas de filtragem colaborativa, são abordadas em [45]. O artigo propõe uma solução que incorpora informação adicional no processo de recomendação, tal como *scores* de confiança explícitos dados pelos utilizadores a outros ou relações de confiança retiradas implicitamente das conexões sociais entre utilizadores. O objetivo é criar uma rede de confiança que gera recomendações a um utilizador, baseadas nas pessoas em que este confia.

Nesta abordagem, foi utilizada a métrica de similaridade Katz Similarity (KS) para selecionar os *k-nearest neighbours* de um utilizador. Estes serão os utilizadores pertencentes à vizinhança do utilizador ativo. Foi utilizada a métrica referida porque, para além de calcular as similaridades entre utilizadores, permite a escolha do tamanho máximo de um caminho na rede de confiança. Desta forma, é possível decidir até quão longe na rede é desejável propagar a confiança.

Combinada com técnicas de normalização para que a eficácia das recomendações seja maior, a métrica KS mostrou ser útil para a seleção de vizinhos e mostrou melhores resultados do que outras abordagens relacionadas.

4.2.2 Medição da confiança das recomendações realizadas

O objetivo de um sistema de recomendações é sugerir itens relevantes a utilizadores. De modo a aumentar o número de itens relevantes recomendados, um sistema de recomendação sistema pode medir a confiança nas suas próprias recomendações, para que seja possível decidir se um item deve ser recomendado ou não. Caso esta decisão tiver em conta a informação disponível sobre o utilizador ou sobre o item, é expectável que a *performance* melhore no que toca a cobertura e diversidade das recomendações.

Em [46], é proposto um sistema de recomendação que utiliza a técnica *K-Nearest-Neighbour* e que decide que um item apenas é recomendado se pelo menos n dos k vizinhos tenham atribuído um *rating* àquele item. Para além disso, é também considerado um limite mínimo de incerteza da previsão. Mais especificamente, é utilizado um algoritmo que permite o cálculo do desvio padrão de uma previsão de um *rating* e caso este valor for menor que um determinado limite definido, aquele item não é recomendado. Um exemplo de algoritmo que permite este cálculo é o algoritmo de fatorização de matrizes.

O sistema foi avaliado em termos de precisão, cobertura, novidade e diversidade das recomendações, uma vez que é crucial obter um balanço entre estas dimensões. A razão é que um sistema de recomendação pode demonstrar uma alta precisão de recomendações se recomendar apenas um item a um utilizador, revelando assim, no entanto, pouca cobertura.

4.2.3 Lidar com recomendações intrusivas

Para um sistema de recomendação, não é apenas suficiente determinar quais recomendações este deve sugerir. O sistema também deve lidar com o problema de incomodar os utilizadores, ou seja, deve estar preocupado em realizar as recomendações de forma não intrusiva.

Em [47], é proposta uma abordagem que utiliza tecnologias *mobile* juntamente com informação contextual do utilizador para endereçar o problema. As tecnologias referidas consistem em aplicações e sensores embebidos no dispositivo móvel do utilizador. O objetivo é avaliar a situação que o utilizador se encontra no momento em que lhe é feita a recomendação, para determinar se este deve ser enviada.

O processo de determinação do nível de intrusão de uma situação é dividido em duas etapas:

- **Modelação da situação**: as rotinas diárias do utilizador são modeladas como um conjunto de situações. Estas são representadas por fatores como parte do dia, dia da semana e atividade do utilizador. Numa determinada situação, o sistema identifica a atividade atual do utilizador (conduzir, enviar mensagens, *browsing*,etc) utilizando os sensores e aplicações embebidas no dispositivo móvel. Assim, o sistema consegue perceber se o utilizador está numa reunião, por exemplo, de acordo com a sua agenda ou a conduzir verificando o sensor de velocidade.
- **Avaliação da intrusão**: a avaliação é feita através da uma função probabilística que considera a probabilidade de aceitação de uma recomendação, sabendo a situação atual do utilizador, e a probabilidade de rejeição de uma recomendação. De acordo com o valor desta função, o sistema decide se a situação do utilizador é apropriada para lhe fazer a recomendação.

Os valores de nível de intrusão obtidos foram depois comparados com valores reais obtidos de um *user study* realizado a 1400 participantes. A estes, foi-lhes perguntado se aceitariam a recomendação, dada uma determinada situação. Foi utilizada a métrica *Mean Average Precision* (MAP) para avaliar a precisão dos níveis de intrusão obtidos do estudo com todos os participantes.

4.2.4 Utilização de Regras de Associação para recomendar itens

Existem várias abordagens que podem ser utilizadas em sistemas de recomendação, dependendo do contexto em que o sistema é desenvolvido e das suas necessidades.

A abordagem proposta em [48] consiste na combinação de algoritmos de associação e métodos baseados em conteúdo. A combinação das duas técnicas é realizada porque o algoritmo Apriori apresenta problemas de *performance* quando aplicado a uma matriz ($Utilizador \times Item$) esparsa. Essa combinação pretende resolver o problema, que é comum a vários sistemas de recomendação.

O algoritmo Apriori é um algoritmo de associação bastante conhecido para gerar regras de associação entre itens. Estas regras são utilizadas para a descoberta de conjuntos de itens frequentes ou *frequent itemsets*, dizendo estes respeito ao conjunto de itens que ocorrem frequentemente juntos numa transação.

Uma vez que são considerados os itens Favoritos e Não Favoritos, o sistema irá recomendar todos os itens a que foi atribuído um *rating*, quer este tenha sido um *rating* alto ou baixo.

A abordagem proposta é dividida em 4 passos:

1. Aplicar o algoritmo Apriori ao espaço dimensional *Utilizador* × *Item* para gerar as regras de associação;
2. Dividir os itens a que foram atribuídos um *rating* em duas categorias: Favoritos e Não Favoritos;
3. Utilizar as regras de associação geradas para descobrir *frequent itemsets* dentro do conjunto de itens Favoritos e encontrar correlações entre eles para recomendar novos itens ao utilizador;
4. Aplicar a abordagem baseada em conteúdo aos itens Não Favoritos para recomendar novos itens ao utilizador.

No que toca ao primeiro passo, o algoritmo Apriori recebe como *input* uma matriz Utilizador/Item, o suporte mínimo e a confiança mínima. A partir da matriz original de *ratings* que contém informação acerca de que *rating* foi atribuído a que item e por qual utilizador, é construída a matriz Utilizador/Item. Esta consiste numa matriz de zeros e uns com utilizadores como linhas e items como colunas, sendo que o valor 0 significa que o utilizador não deu um *rating* ao item, e o valor 1 significa o contrário. O suporte é o valor que exprime quão popular um *itemset* é, medido pelo número de transações em que este aparece. A confiança refere-se à probabilidade de um item X estar associado a um item Y.

As regras de associação são representadas através da forma: X $\rightarrow$ Y, significando que o item X está associado ao item Y.

No segundo passo, é criado um *array* de itens que possuem *rating* dado por um utilizador. Esses itens são depois divididos em duas classes, Favoritos ou Não Favoritos, consoante o *rating* que receberam. No terceiro passo, para os itens marcados como Favoritos, é verificado a que outros itens eles estão associados e feita a recomendação com base nessa verificação. Se o item estiver associado ao item favorito e não tiver um *rating* atribuído, esse item é recomendado ao utilizador.

No último passo e relativamente aos itens marcados como Não Favoritos, é utilizada a abordagem baseada em conteúdo para recomendar itens similares e que não tenham um *rating* atribuído pelo utilizador. Para tal, os valores de similaridade entre itens são calculados através da métrica de similaridade de Jaccard, baseando-se nas palavras descritivas que cada item contém para calcular esses valores.

4.2.5 Métodos de *Clustering* para geração de recomendações

A aplicação de métodos de *clustering* combinados com filtragem colaborativa podem ser utilizados num sistema de recomendações. Esta combinação é apresentada em [49], onde o desafio de obter escalabilidade para um grande conjunto de utilizadores é resolvido com a aplicação de *clustering*. *Data sets* do domínio do *e-commerce* e também no domínio dos operadores de telecomunicação, lidam com um grande número de utilizadores. Algoritmos de filtragem colaborativa que calculam a vizinhança de cada um dos utilizadores podem revelar problemas quando o número de utilizadores é elevado. Na abordagem proposta, é aplicado *clustering* aos utilizadores baseado no seu histórico de *ratings*. A vizinhança de um utilizador serão os restantes utilizadores que estiverem no mesmo *cluster* que ele.

Após os utilizadores estarem devidamente segmentados em *clusters*, são aplicados algoritmos de filtragem colaborativa apenas dentro do *cluster* no qual o utilizador ativo que se quer fornecer recomendações se encontra. Isto reduz significativamente o tempo e esforço computacional dos algoritmos de vizinhança.

K-Means clustering também é utilizado num sistema de recomendação híbrido, apresentado em [50]. O *clustering* é baseado nos *ratings* que os utilizadores atribuíram aos itens. São depois calculados os representativos de cada *cluster*, que são os *ratings* previstos que cada item irá ter. Os *ratings* representativos do itens num determinado *cluster* são calculados por duas formas:

- *Rating* mais frequente de cada item - caso a frequência de todos os *ratings* seja igual, o representativo é igual a 0.
- Média dos *ratings* de cada item;

São estes *ratings* representativos dos itens que consistirão nas previsões feitas pelo sistema e que serão recomendados aos utilizadores, conforme o seu valor.

4.2.6 Sistemas de recomendação para operadoras de telecomunicações

Existem estudos já realizados sobre sistemas de recomendação no âmbito das operadoras de telecomunicações. Em [51] é proposto um sistema de recomendação para reduzir a *churn rate* das operadoras. Neste estudo, o sistema de recomendações consistem na lista de produtos mais bem cotados, ou que receberam melhores *ratings* no *website* da operadora, ou seja nos produtos mais populares e mais utilizadores. As recomendações são também baseadas na localização e género do utilizador.

É também proposto em [52] um sistema de recomendação que utiliza vários algoritmos de filtragem colaborativa para gerar recomendações para um *data set* de clientes de telecomunicações. Estas usam o conceito de vizinhança entre utilizadores e itens, já falado na sub-secção 3.1.2, *clusters* de utilizadores e o algoritmo Slope One.

Em [53], é utilizada uma junção de filtragem colaborativa baseada nos utilizadores e baseada nos itens. A técnica baseada nos itens é utilizada para criar as previsões e gerar a matriz Utilizador x Item que contém os *ratings* atribuídos por cada utilizador aos itens. A técnica baseada nos utilizadores para calcular semelhança entre utilizadores e selecionar os mais parecidos, obtendo-se depois as recomendações. A junção destas duas técnicas serve como uma solução ao problema da esparsidade, uma vez que faz uso da informação horizontal (utilizadores) e vertical (itens) de uma matriz Utilizador x Item.

Capítulo 5

Ferramentas e Tecnologias

Neste capítulo são descritas as ferramentas e tecnologias que permitiram analisar e tratar os dados bem como implementar os algoritmos de recomendação.

A linguagem utilizada é o Python 2.7 pois foi a linguagem escolhida pela empresa para projetos que envolvam a vertente de *data mining* devido à sua relativa facilidade em lidar com grandes quantidades de dados.

O Python contém diversas bibliotecas que permitem, quer o tratamento de grandes quantidades de dados, quer a implementação de algoritmos de recomendação. Uma das mais conhecidas e poderosas para o último ponto é o Surprise [54]. Esta biblioteca foi utilizada pois contém vários algoritmos de previsão *ratings* e métricas para cálculo de similaridades e métricas de avaliação de recomendações já referidas neste documento.

Os dados foram tratados maioritariamente com recurso à biblioteca Pandas [55], do Python. Esta permite lidar com grandes estruturas de dados e ferramentas de análise de dados com relativa facilidade, fornecendo também uma boa *performance*.

Diversas implementações dos algoritmos utilizados no estudo e nas diferentes abordagens presentes na biblioteca *scikit-learn* [56]. Esta biblioteca é uma das mais populares do Python para treino modelos de Machine Learning (ML), pré-processamento de dados, seleção e avaliação de modelos, etc.

A *framework* Dask [57], em conjunto com a biblioteca de ML integrada *Dask-ML* [58] foram também utilizadas neste estágio, uma vez que permitem que haja computação paralela, e portanto, apresentam uma melhor *performance* ao tratamento de dados de grandes dimensões e ao treino de modelos.

Outra ferramenta utilizada foi o serviço de ML da IBM, denominado *Watson Machine Learning* [59]. Este serviço na *cloud* permite a construção de um motor de recomendações de produtos e foi explorado com o objetivo avaliar a sua viabilidade para uma utilização integrada no ACM.

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Análise global dos dados

Projetos que lidem com uma quantidade considerável de dados necessitam de uma metodologia própria que permita conhecer o conteúdo dos dados e permita o seu tratamento. Neste capítulo são descritas as ações de análise dos dados disponíveis e a informação que foi possível extrair destes.

Numa primeira fase foi estudado qual o objetivo deste projeto de modo a perceber que tipo de informação os dados deverão conter para que seja possível atingi–lo. O principal objetivo do projeto é obter uma lista de recomendações de campanhas para um determinado utilizador. Considerou-se, portanto, que o histórico de adesões a campanhas seria o tipo de informação que os dados deveriam conter.

O *data set* consiste numa lista de registos de incentivos a campanhas realizado a um grande número de clientes. Estes registos são correspondentes ao período de tempo de 2 anos. Na Tabela 6.1 são apresentados os campos pertencentes aos dados e uma breve descrição do que cada campo significa.

Apenas os campos "COD_UNICO_CARTAO", "COD_CAMPANHA_ACM" e "FLAG_ADESAO" foram considerados relevantes para o estudo que se pretende fazer. São estes campos que indicam o histórico de adesões de um cliente às campanhas. Foram ainda adicionados os campos "ID_CAMPANHA" e "ID_USER" que correspondem, respetivamente, a um identificador do tipo *int* para cada campanha e para cada cliente. Os valores destes campos estão definidos numa *range* de 0 ao número de campanhas e número de clientes existentes no *dataset*.

Algumas estatísticas sobre os dados podem ser observadas na Tabela 6.2. O número de incentivos diz respeito ao número de registos que o *dataset* contém. Cada registo significa um incentivo de uma campanha feito a um cliente. Existem no *dataset* um total de 6610833 utilizadores incentivados e um total de 286 campanhas. Foi calculado também o número total de adesões presentes no *dataset*, sendo este número 1460281, havendo uma média de 8250 adesões por campanha.

No gráfico da Figura 6.1 é possível observar o número de incentivos realizados por campanha. O número médio de incentivos realizados por campanha é de 160630, como mostra a linha horizontal vermelha no gráfico. Neste gráfico o eixo horizontal dos gráficos diz respeito aos identificadores das campanhas, pelo que o número de incentivos observados diz respeito a cada campanha que está contida no *dataset*.

Existem campanhas que apresentam um elevado número de incentivos comparativamente às restantes. Esta diferença notória do número de incentivos deve-se aos diferentes públicos-

Tabela 6.1: Descrição dos campos presentes no *dataset*.

Campo	**Descrição**	**Tipo de dado**
COD_MES	Identificador do mês em que ocorreu o incentivo da campanha.	*int*
COD_DIA_CAMPANHA	Identificador do dia e mês em que ocorreu o incentivo da campanha.	*int*
COD_HORA	Identificador da hora em que ocorreu o incentivo da campanha.	*int*
COD_MIN_SEG	Identificador do minuto e segundo em que ocorreu o incentivo da campanha.	*int*
COD_UNICO_CARTAO	Identificador do cliente que foi incentivado.	*int*
MSISDN	*Mobile Station International Subscriber Directory Number* - número que identifica de forma unívoca o subscritor de um serviço numa rede móvel Global System for Mobile Communications (GSM) ou Universal Mobile Telecommunications System (UMTS).	*int*
COD_CARTAO	Identificador do cliente que foi incentivado.	*int*
COD_CAMPANHA_ACM	Identificador da campanha.	Texto
FRASE	Frase descritiva do que consiste a campanha.	Texto
PRAZO_PARA_ADESAO	Identificador do dia e mês correspondente ao prazo que o cliente tem para aderir à campanha.	*int*
FLAG_ADESAO	Valor 0 ou 1 que indica se o cliente aderiu (valor 1) ou não (valor 0) à campanha.	*int*
COD_DIA_ADESAO	Identificador do dia e mês em que o cliente aderiu à campanha.	*float*
COD_HORA_ADESAO	Hora em que o cliente aderiu à campanha.	*float*
COD_MIN_SEG_ADESAO	Minuto e segundo em que o cliente aderiu à campanha.	*float*
DESC_ESTADO_ACM	Descrição do estado do ciclo de vida no ACM.	Texto

Tabela 6.2: Estatísticas dos dados analisados.

Estatística	**Valor**
Número de incentivos	46422213
Número de clientes incentivados	6610833
Número de campanhas	286
Número total de adesões	1460281

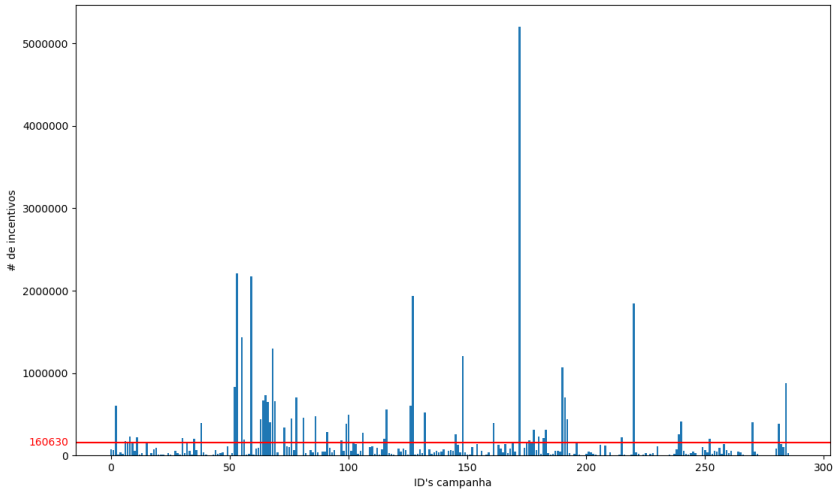


Figura 6.1: Número de incentivos por campanha.

alvo a que se destinam as campanhas. Podem existir campanhas que são direcionadas a um número muito grande de clientes e outras direcionadas a um conjunto mais restrito.

Outra explicação para isto acontecer é o facto de existirem campanhas que são incentivadas aos clientes em ciclos, ou seja, a mesma campanha é incentivada em determinados intervalos de tempo. Isto faz com o que o número de incentivos para essas campanhas seja muito superior. Existem campanhas em que não faz sentido possuírem esta característica caso se tratem, por exemplo, de campanhas específicas para uma altura do ano como o Natal.

Semelhante ao gráfico da Figura 6.1, foi elaborado o gráfico da Figura 6.2 com o número de adesões por campanha. Com uma média de 160630 mostrada na linha vermelha, é possível ver bem a diferença na magnitude de valores comparativamente ao número de incentivos. Tal como na Figura 6.1, o eixo horizontal do gráfico 6.2 corresponde aos identificadores das campanhas.

Foi calculada a eficácia de cada campanha tendo em conta o seu número de incentivos e número de adesões. A eficácia é dada pela fórmula

$$Eficacia = \frac{NumeroAdesoes}{NumeroTotalIncentivos}$$

onde $NumeroAdesoes$ corresponde ao número de adesões que aquela campanha obteve e $NumeroTotalIncentivos$ o número total de incentivos feitos daquela campanha.

O gráfico da Figura 6.3 mostra a eficácia calculada por campanha. Podem-se observar algumas campanhas com valor de eficácia igual a 1 o que significa que todos os clientes

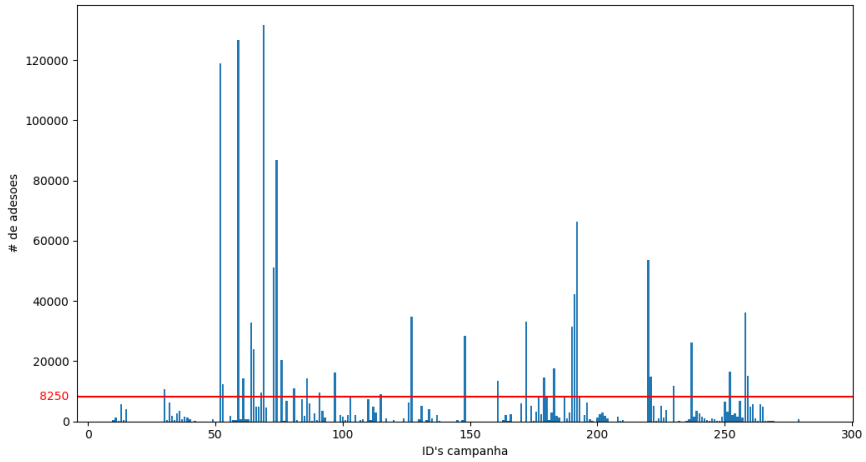


Figura 6.2: Número de adesões por campanha.

incentivados com aquela campanha aderiram. No entanto isso pode ser apenas devido à campanha ser incentivada poucas vezes.

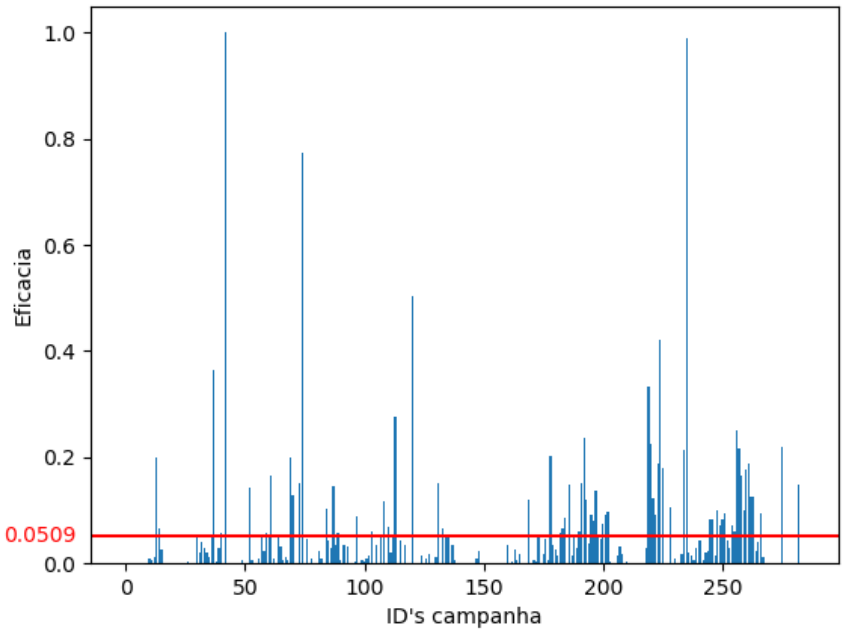


Figura 6.3: Eficácia por campanha.

O gráfico da Figura 6.4 apresenta o número total de clientes que aderiram um determinado número de vezes a campanhas. Isto mostra que existem cerca de 700 mil clientes que aderiram 1 vez a campanhas e um número cada vez menor à medida que o número de vezes que aderiu aumenta. Na barra correspondente ao número 3 por exemplo, é possível observar que o número de clientes que aderiu 3 vezes a campanhas já é cerca de 50 mil.

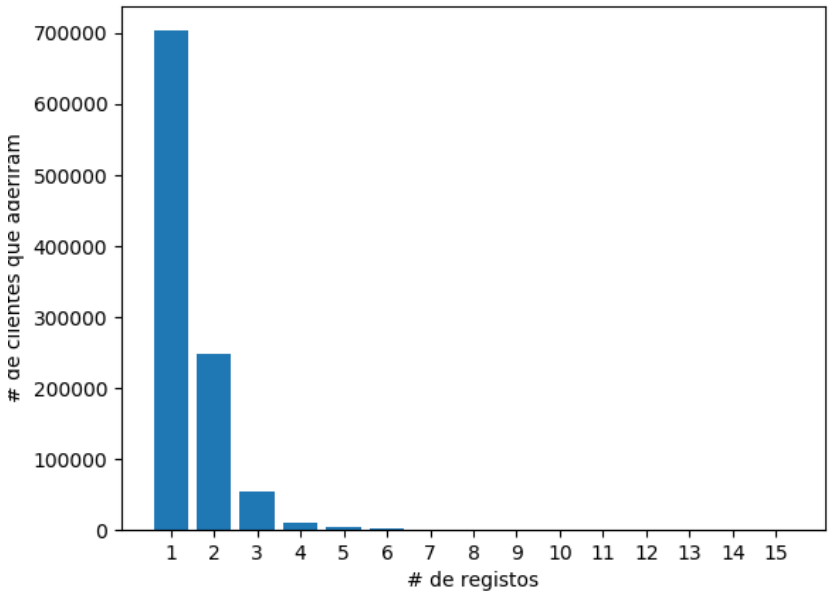


Figura 6.4: Número de clientes que aderiam X vezes às campanhas.

Na Figura 6.5 são apresentados o número de clientes por campanha que aderiram sempre que incentivados 1, 2, 3 e 4 vezes. O que isto significa é que existem clientes que, independentemente do número de vezes que são incentivados, aderem sempre à campanha. A este tipo de clientes é dada a denominação de "clientes perfeitos" para efeitos de simplificação neste relatório. Com isto é possível obter uma ideia das campanhas que possuem uma grande taxa de adesão. Tal informação poderá ser útil para o sistema de recomendações, pois poderão se tratar de campanhas que os utilizadores estejam mais suscetíveis a aderir, então poderão ser-lhes recomendadas.

Vimos que, à medida que o número de incentivos aumenta de gráfico para gráfico, o número de adesões diminui, no entanto, consegue-se observar que em algumas das campanhas mantêm-se um elevado número de adesões, mesmo quando o número de incentivos também aumenta. Nestes gráficos, no eixo horizontal estão representados os números identificadores de cada campanha.

Como complemento aos gráficos da Figura 6.5, a Figura 6.6 mostra o número de campanhas em que houveram clientes perfeitos com X incentivos, sendo X o número expresso no eixo horizontal do gráfico. Por exemplo, o valor da barra correspondente ao valor 1 significa que existem perto de 175 campanhas nas quais existem casos de clientes que são incentivados 1 vez e aderem 1 vez. Já para o valor 5 existem perto de 40 campanhas com clientes perfeitos com 5 incentivos e consequentes 5 adesões.

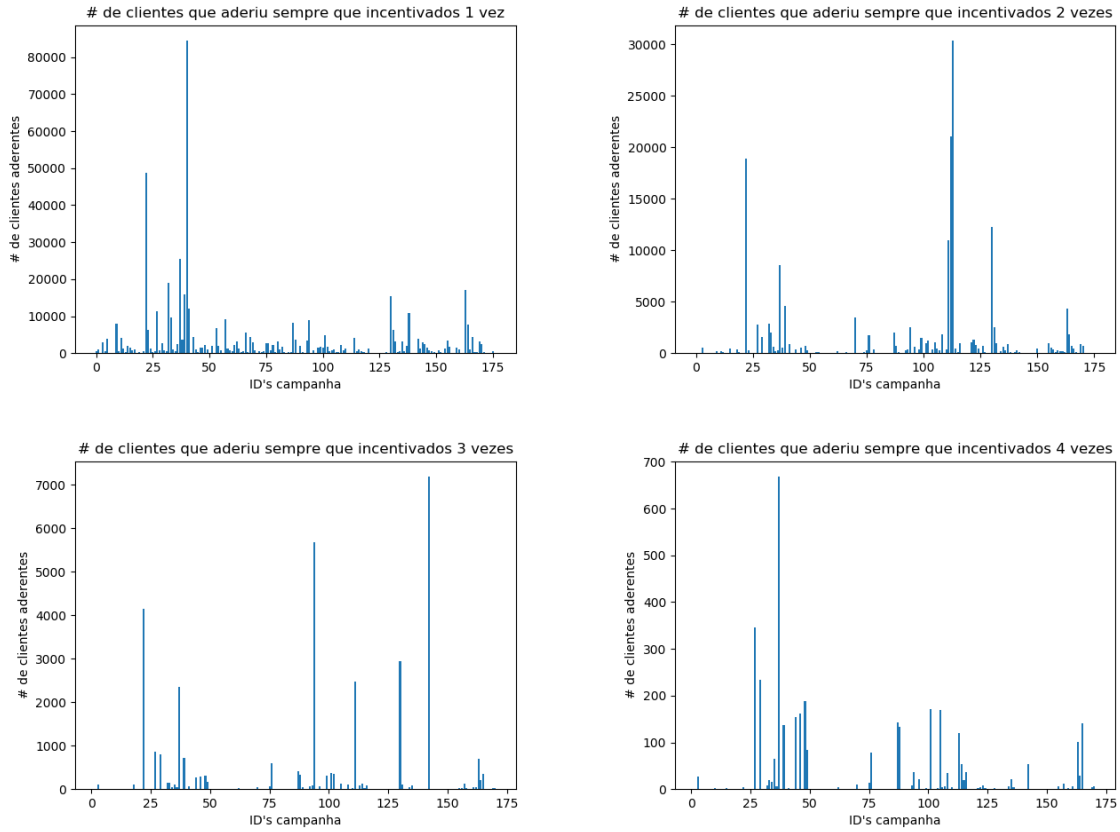


Figura 6.5: Número de clientes por campanha que aderiram sempre que incentivados 1, 2, 3 e 4 vezes

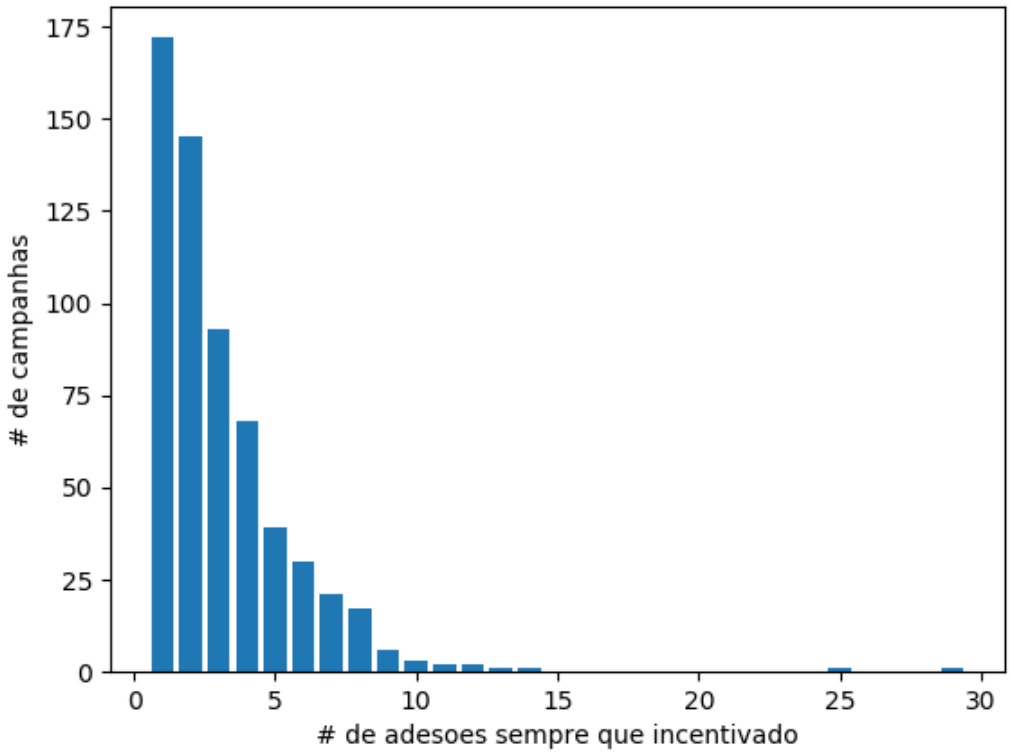


Figura 6.6: Número de campanhas em que existem clientes perfeitos com X incentivos.

Esta página foi intencionalmente deixada em branco.

Capítulo 7

Abordagens

Neste capítulo são descritas as abordagens e métodos que foram investigados e que fizeram parte deste estudo. A primeira abordagem foi a utilização de uma *framework open-source*, o Surprise, para implementar um sistema de recomendações com base no histórico de adesões dos clientes, que é descrita na secção 7.2. A segunda foi a exploração das ferramentas *Watson Machine Learning* e *IBM Watson* com a finalidade de perceber se o Active Campaign Manager (ACM) beneficiaria da utilização deste tipo de *software* empresarial licenciado. Estes serviços, que permitem para a construção de um motor de recomendações de produtos, garantem a segurança e escalabilidade do sistema e estão descritos na secção 7.3. Por último, a terceira abordagem é descrita na secção 7.4 e consistiu na utilização de características de cliente e modelos de *clustering* para obtenção de recomendações baseadas nessas características e também no histórico de adesões a campanhas dos clientes.

Antes de qualquer uma das abordagens ser seguida, foi necessário transformar o *feedback* implícito provenientes dos dados do histórico de adesões a campanhas em *feedback* explícito, como descrito na secção 7.1. Esta transformação serviu como base a algumas das abordagens que foram seguidas posteriormente.

7.1 Tranformação de *feedback* implícito em *feedback* explícito

Como referido no capítulo 6, os dados disponíveis contêm informação de incentivos a campanhas e adesões que os clientes receberam durante um período de 2 anos, tratando-se, portanto, de informação na forma de *feedback* implícito. Neste estudo, os clientes não fornecem *ratings* ou *likes* explicitamente a uma campanha, o que dificulta a compreensão do verdadeiro interesse que estes possam demonstrar por uma. Tendo isso em conta, foram considerados o número de incentivos que cada cliente recebeu de cada campanha e o número de adesões que estes efetuaram de forma a criar um rácio entre ambos. Desta forma, foi possível transformar o *feedback* implícito disponível em *feedback* explícito, na forma de um rating. Por exemplo, se um cliente X recebeu 6 notificações para uma campanha A e subscreveu a campanha 2 vezes, o rácio terá o valor de $2/6 = 0.33(3)$. Este valor foi depois mapeado para um intervalo de 0 a 10, ou seja, no exemplo anterior, o *rating* do cliente X à campanha A será de 3. São estes *ratings* que constituem o *data set* final fornecido aos algoritmos de recomendação testados.

Foi considerada esta abordagem em oposição a utilizar apenas *ratings* de zeros e uns caso os clientes tenham ou não aderido à campanha, respetivamente. A razão de tal não ter sido

realizado é que, desta forma, não seria possível distinguir a verdadeira aceitação dos clientes a uma campanha específica. Clientes que foram incentivados 30 vezes e aderiram 1 vez são diferentes de clientes que foram incentivados 3 vezes e aderiram 1 vez também. Considerando o valor de rácio mencionado, obtém-se uma ideia mais clara acerca do interesse do cliente na campanha.

Observou-se que cerca de 79% dos clientes presentes no *data set* apenas forneceram *ratings* a uma campanha – em média, todos os clientes deram *ratings* a $1,27$ campanhas. Isto iria dificultar o processo de geração de recomendações, influenciando erradamente os algoritmos. De modo a resolver esta situação, os algoritmos foram treinados com o *data set* filtrado, considerando-se apenas os registos utilizador-campanha onde o número de adesões é maior ou igual que um. Isto não significa que o treino dos algoritmos não considera *ratings* iguais a zero – poderão ainda existir utilizadores incentivados várias vezes e terem aderido apenas uma, resultando num rácio bastante baixo mapeado depois para o valor 0 – mas, desta forma, a proporção de clientes e campanhas que não fornecem informação relevante para as recomendações diminui bastante. Devido à aplicação deste filtro, o número de campanhas desceu de 286 para 177. O *data set* obtido após a aplicação deste filtro foi o utilizado para o treino dos algoritmos. Na Tabela 7.1 são apresentadas mais algumas estatísticas do *data set* após aplicação do filtro referido, comparado com o *data set* original.

Tabela 7.1: Estatísticas dos dados analisados.

Estatística	*Data set* original	*Data set* filtrado
Número de incentivos	46422213	8255599
Número de clientes incentivados	6610833	804750
Número de campanhas	286	177
Número total de adesões	1460281	1460281

A aplicação do filtro referido, implica a diminuição do número de incentivos, clientes incentivados e campanhas consideradas, uma vez que são apenas considerados os incentivos aos clientes que resultaram em adesões da parte deles. A diminuição do número de campanhas também evidencia que algumas das campanhas não apresentavam qualquer adesão.

7.2 Surprise

A biblioteca Surprise dispõe de vários algoritmos de recomendação disponíveis no *package* "predictions_algorithms". Para este estudo foram analisados os algoritmos SVD, SVDpp, NormalPredictor, BaselineOnly, SlopeOne e CoClustering. Outros algoritmos presentes na biblioteca, como os algoritmos de *nearest neighbours*, foram considerados, mas devido ao excessivo esforço computacional necessário para executar os modelos com a quantidade de dados desejada, estes algoritmos foram descartados da análise. Com o Surprise, é possível obter a previsão de um *rating* que cada utilizador daria a cada uma das campanhas presente no *data set*. Ao conjunto de campanhas ordenado por *rating* previsto para cada utilizador, é denominado, neste relatório, como lista de recomendações. Naturalmente, esta lista poderá ser diferente de utilizador para utilizador.

Tabela 7.2: Classificação dos *ratings* previstos para um *threshold*(TH) específico.

	Rating previsto >= TH	Rating previsto <TH
Rating real >= TH	True Positive	False Negative
Rating real <TH	False Positive	True Negative

7.2.1 Métodos de Avaliação

Foram escolhidas métricas de avaliação que tivessem significado e fizessem sentido de acordo com o tipo de resultados obtidos. Uma vez que o Surprise devolve previsões de *ratings*, foram utilizadas métricas que calculam o erro entre valor real e valor previsto, tais como MAE e RMSE. A partir destas métricas, é possível inferir quão perto estão os valores previstos dos *ratings* dos valores reais dados pelos utilizadores.

Para além disso, métricas como *precision*, *recall*, *F1-measure* e *Specificity* foram utilizadas. Estas métricas têm em conta o número de *true positives*, *false negatives*, *false positives* e *true negatives* presentes na lista de recomendações. De modo a classificar os resultados nestas quatro denominações, foi definido um valor de *threshold*, correspondente ao valor acima do qual um *rating* é considerado como sendo positivo ou negativo, ou seja, se o *rating* reflete uma adesão à campanha. A classificação em *true positives*, *false negatives*, *false positives* e *true negatives* foi feita de acordo com a Tabela 7.2.

A métrica *specificity* representa a proporção de *true negatives*, fornecendo informação acerca dos clientes que, corretamente, não estão a ser notificados com campanhas. Esta métrica foi considerada para avaliar a capacidade dos algoritmos identificarem clientes que não devem ser notificados com campanhas, uma vez que o sistema deverá ter isso em conta, para que os clientes não sejam incomodados desnecessariamente. A métrica *F1-measure* foi calculada para facilitar a interpretação das métricas *precision* e recall, pois consiste na sua combinação.

Foi também calculada a métrica Mean Average Precision at K (MAP@K), de acordo com a abordagem descrita em [60]. Obtendo as listas de recomendações de vários utilizadores, ordenadas decrescentemente pelo *rating* previsto, é considerado apenas o primeiro elemento para o cálculo da *precision*, depois os dois primeiros, depois os três primeiros, até k elementos. Para apenas um utilizador, este é o cálculo da *Average Precision at K*. Para vários utilizadores, a média destas valores é calculada, daí o termo *Mean Average Precision* (média de médias). A fórmula matemática da métrica é descrita na equação 7.1:

$$MAP@K = \frac{1}{|U|} \sum_{u=1}^{|} U|(\text{AP@K})_u \tag{7.1}$$

Na equação 7.1, $|U|$ é o número de utilizadores considerados e $AP@K$ é a *Average Precision at K*. O valor de K pode variar de 1 até ao número de campanhas que o utilizador atribuiu um *rating*.

Outra métrica calculada foi o Mean Reciprocal Rank (MRR). Através desta métrica, é verificada qual a posição, ou *rank*, que a campanha com *rating* real mais alto ocupa na lista de recomendações. Este *rank* é, portanto, diferente para cada utilizador, dependendo de qual campanha ele atribuiu um *rating* mais elevado. O valor da métrica é obtido pela média de *ranks* calculados para vários utilizadores. A sua fórmula matemática está descrita na equação 7.2.

$$MRR = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{rank_u} \tag{7.2}$$

Foi calculada ainda uma variação do MRR, aplicando-a a cada campanha, para verificar em que posição cada campanha aparecia nas listas de recomendação, em média, de vários utilizadores. Por exemplo, uma campanha com valor 30 desta métrica, ocupa, em média, a posição 30 nas listas de recomendações dos utilizadores. De modo a ser mais interpretável, o valor desta métrica corresponde à posição da campanha na lista, e não ao inverso desse valor, como é o caso do MRR, como é possível observar na equação 7.2.

7.2.2 Resultados

Nesta subsecção são apresentados os resultados obtidos das métricas descritas na subsecção 7.2.1, algumas especificações do treino e da avaliação dos algoritmos.

O *data set* com os *ratings* atribuídos a campanhas pelos utilizadores foi dividido em dois conjuntos: *training set*, que representa 75% do *data set* total e *test set*, que representa os restantes 25%. O valor da *threshold* deve ser uma decisão estratégica feita pela operadora de telecomunicações. Isto porque se trata do valor acima do qual, um *rating* é considerado como refletor de um interesse genuíno do utilizador na campanha. Operadoras diferentes poderão considerar valores diferentes para esse interesse. Para esta estudo, foi definido o valor de $7,5$. Foi considerado portanto que, um cliente que adira 75% das vezes que é notificado está realmente interessado naquela campanha. Campanhas com previsões de *ratings* acima da *threshold* devem ser então recomendadas ao utilizador.

Os valores das métricas referidas na secção 7.2.1 estão presentes na Tabela 7.3. É possível observar que em termos das métricas RMSE e MAE, os algoritmos BaselineOnly e SVD mostram resultados superficialmente melhores, com *ratings* previstos a desviaram-se de *ratings* atuais em média ≈ 2.5 e ≈ 1.85, respetivamente. Olhando para a métrica *F1-measure*, que combina *precision* e *recall*, os mesmos algoritmos também mostram bons resultados. O que significa que os *ratings* previstos refletem corretamente o comportamento apresentado nos *ratings* reais correspondentes, quer esse comportamento seja de interesse numa campanha ou não.

Tabela 7.3: Valores das méttricas dos vários algoritmos.

Algoritmo	RMSE	MAE	Precision	Recall	F1-Score	Specificity	MRR
BaselineOnly	2.518	1.888	83.9%	81.7%	82.7%	63.2%	0.0286
SVD	2.558	1.908	82.8%	82.5%	82.7%	59.9%	0.0287
SVDpp	2.762	2.312	73.4%	94.1%	82.5%	20.6%	0.2176
CoClustering	3.143	2.537	72.6%	89.9%	80.3%	20.7%	0.0132
SlopeOne	2.959	2.433	73.6%	91.1%	81.4%	23.4%	0.0261
NormalPredictor	3.769	2.864	70.1%	58.8%	63.9%	41.3%	0.0299

Os resultados para a *specificity* dos algoritmos BaselineOnly e SVD são também os melhores quando comparados com os restantes, o que permite perceber que possuem uma boa capacidade de identificar clientes que não devem ser incentivados com determinadas campanhas. Os valores do MRR, são bastante semelhantes em praticamente todos os algoritmos, o que mostra que as campanhas com *ratings* reais mais altos ocupam apro-

Tabela 7.4: Valores, em percentagem, da métrica MAP@K para diferentes K's.

	MAP									
Algoritmo	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
BaselineOnly	81.63	74.54	73.81	73.70	73.68	73.68	73.68	73.68	73.68	73.68
SVD	89.75	82.62	81.99	81.90	81.88	81.88	81.88	81.88	81.88	81.88
SVDpp	93.59	86.51	85.83	85.73	85.72	85.72	85.72	85.72	85.72	85.72
CoClustering	93.30	84.11	83.30	83.20	83.18	83.18	83.18	83.18	83.18	83.18
SlopeOne	94.40	86.23	85.50	85.40	85.39	85.38	85.38	85.38	85.38	85.38
NormalPredictor	60.42	53.89	53.26	53.17	53.15	53.15	53.15	53.15	53.15	53.15

ximadamente a mesma posição nas listas de recomendação. Existe, porém, a exceção do algoritmo SVDpp que coloca estas campanhas num lugar mais cimeiro nas listas.

Na Tabela 7.4 estão os valores para a métrica MAP@K metric, em percentagem. Esta métrica foi calculada para vários valores de K, com recurso à biblioteca em Python "ml_metrics", disponível em [61].

É possível ver que para um valor maior que 6, os valores de MAP@K mantém-se iguais. Isto deve-se ao facto de, o valor máximo de K corresponder ao número máximo de campanhas que os utilizadores atribuíram um *rating*. A maioria dos utilizadores, aproximadamente 80%, apenas deram *rating* a uma campanha. Em média, todos os utilizadores deram um *rating* a apenas ≈ 1.27 campanhas. Sabendo isso, pode-se concluir que os valores presentes na Tabela 7.4 para K's mais pequenos são aqueles que melhor refletem o verdadeiro comportamento dos algoritmos.

Foi realizada uma análise mais detalhada para perceber a dispersão dos *rankings* para todos os algoritmos e para todas as cmapanhas. O gráfico *box plot* na Figura 7.1 representa os valores da métrica de *ranking* adaptada referida na subsecção 7.2.1. O valor do eixo YY do gráfico representa o *ranking* nas listas de campanhas, portanto valores mais baixos correspondem aos *rakings* melhores. Todos os algoritmos mostram um comportamento similar entre sim, com exceção do NormalPredictor. Este algoritmo demonstra *rankings* médios de todas as campanhas muito similares, o que é uma desvantagem desta abordagem.

A Figura 7.1 mostra que o *ranking* mínimo é bastante baixo, independentemente do algoritmo (com exceção do NormalPredictor), o que demonstra que algumas campanhas estão sempre a ser recomendadas nos primeiros lugares das listas de recomendação. De modo a perceber que campanhas aparecem no topo da lista de recomendações para os diferentes algoritmos, foi elaborado o gráfico da Figura 7.2. O objetivo foi tentar perceber se são as mesmas campanhas que são recomendadas no topo, ou se este topo contém campanhas muito variadas, de algoritmo para algoritmo.

O eixo dos XX da Figura 7.2 representa as campanhas – todas as 177. A barra de *score* representa o *ranking* da campanha, sendo uma cor mais saturada sinal de um valor de *ranking* menor, ou seja, as campanhas melhor posicionadas. O eixo dos YY distingue os 6 algoritmos considerados.

É possível observar que algumas campanhas demonstram um comportamento similar em diferentes algoritmos, algo que está de acordo com o que também é observado na Figura 7.1. Observam-se linhas verticais muito saturadas, indicadores de que, para diferentes algoritmos, as mesmas campanhas estão a ser recomendadas bastantes vezes nas posições mais de topo das listas de recomendação. Isto é observável, por exemplo, perto da campanha 150. Na Figura 7.2, o algoritmo NormalPredictor demonstra o mesmo comportamento eviden-

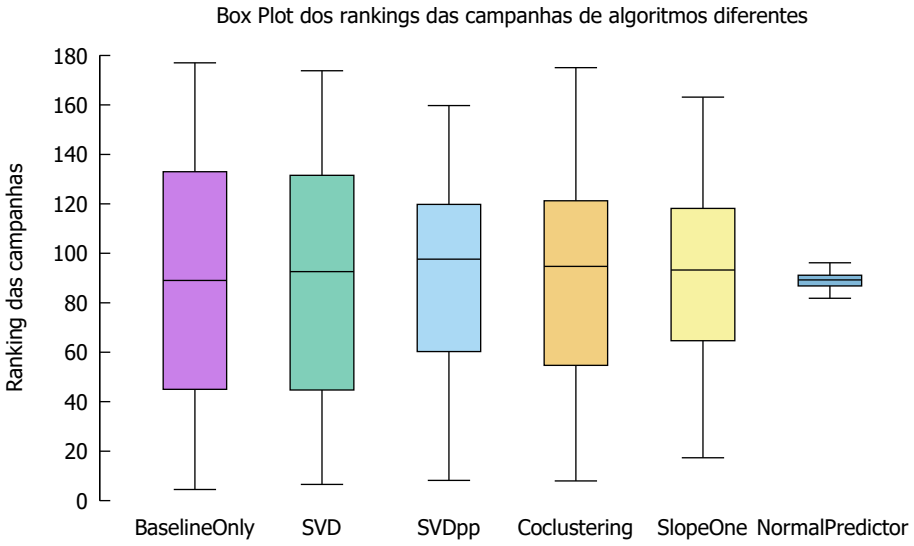


Figura 7.1: Gráfico *box plot* dos *rankings* das campanhas para os diferentes algoritmos.

ciado na Figura 7.1, que é todas as campanhas possuírem *rankings* médios nas listas de recomendação dos utilizadores bastante similares, nomeadamente *rankings* próximos do valor 90.

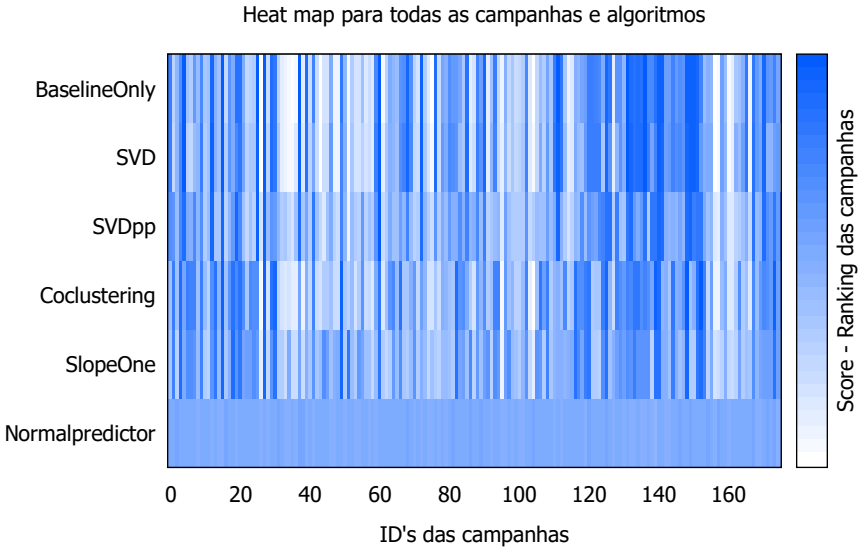


Figura 7.2: Heatmap com o *ranking* de todas as campanhas, para todos os algoritmos.

Na Tabela 7.5 estão, em segundos, os tempos de duração dos processos de treino e de teste dos algoritmos considerados. O algoritmo NormalPredictor possui os tempos de treino mais rápido, mas tendo em conta as Figuras 7.1 e 7.2, este algoritmo não gera as melhores recomendações. Portanto, esta poderá ser uma razão para descartar este algoritmo para sistema de recomendação, num cenário de produção. Os melhores algoritmos, tendo em conta os gráficos e as Tabelas 7.3, 7.4 e 7.5, são os algoritmos BaselineOnly e SVD. Estes possuem uma boa *performance* em termos de tempo de treino do modelo, e também bons resultados no que toca às métricas de avaliação consideradas.

Tabela 7.5: Tempos de treino e teste dos diferentes algoritmos.

Algoritmo	Tempo de treino (segundos)	Tempo de teste (segundos
BaselineOnly	4.613	3.474
SVD	42.844	4.994
SVDpp	103.917	4.381
CoClustering	76.721	3.440
SlopeOne	7.944	3.096
NormalPredictor	1.144	2.638

7.3 IBM Watson Machine Learning

O IBM Watson Machine Learning é um serviço de *cloud* da IBM que permite o treino e o teste de modelos de ML, com garantias de escalabilidade e segurança. Através do serviço, é possível construir modelos analíticos e redes neuronais e treiná-los com os dados desejados. Os modelos podem variar desde modelos de classificação, processamento de linguagem natural, reconhecimento de imagens, entre outros. O serviço também suporta *notebooks* Jupyter [62], que permitem executar código para processamento dos dados e visualizar os resultados.

Com recurso a uma biblioteca de cliente de Python para o serviço Watson Machine Learning [63], é possível com o *notebook* testar e realizar os *deployments* dos modelos criados. Os resultados obtidos desta abordagem foram obtidos a partir do *notebook* disponível em [64], com algumas alterações necessárias tendo em conta os dados e objetivo deste estudo.

A partir do *data set* de *ratings* construído e descrito na secção 7.1, foi construída uma matriz Utilizador x Campanha, em que cada posição da matriz corresponde ao *rating* atribuído pelo utilizador daquela linha à campanha daquela coluna. Esta matriz consistiu nos dados que foram inseridos no IBM Watson e utilizados para o treino do modelo.

A matriz Utilizador x Campanha criada contém bastantes valores a nulo devido ao número reduzido de campanhas que cada utilizador atribui um *rating*. O gráfico da Figura 7.3 apresenta esta estatística, onde se pode observar que existem 638812 utilizadores que apenas forneceram *ratings* a 1 campanha. Os valores na matriz para todas as outras campanhas que os utilizadores não tenham atribuído um *rating* estarão a nulo.

De forma a visualizar a distribuição dos clientes, conforme o seu histórico de adesões, num espaço bi-dimensional, foi aplicado Principal Component Analysis (PCA) à matriz. Este algoritmo é explicado mais detalhadamente na sub-secção 7.4.2.1. Resumidamente, este transforma a informação presente em várias colunas num número menor de colunas. Para o caso da matriz de *ratings*, o número PC's escolhidos foi de apenas 2, correspondendo às coordenadas X e Y do cliente no espaço bi-dimensional. O objetivo aqui foi agregar toda a informação de cada linha da matriz, ou seja, de cada cliente, e transformá-la em dois valores numéricos, que diferenciassem os clientes conforme as campanhas que atribuíram *ratings* e conforme esses valores. Clientes que tenham atribuído *ratings* iguais exatamente às mesmas campanhas são representados por pares de valores iguais. Clientes com *ratings* diferentes ou atribuídos a campanhas diferentes têm pares de valores diferentes.

A Figura 7.4 apresenta a distribuição dos clientes no espaço bi-dimensional, conforme o seu histórico de *ratings*. É possível ver no gráfico da figura que os clientes estão bastante segmentados no espaço bi-dimensional, o que revela a pouca desigualdade que existe grupos de clientes muito parecidos em termos do seu histórico de adesões. Esta análise poderá ser relevante para a escolha do número de *clusters*.

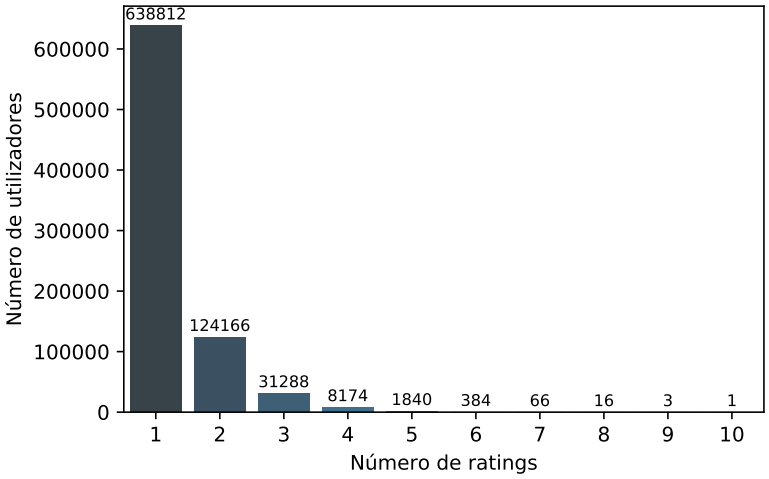


Figura 7.3: Distribuição do número de utilizadores e do número de campanhas às quais atribuíram um *rating*.

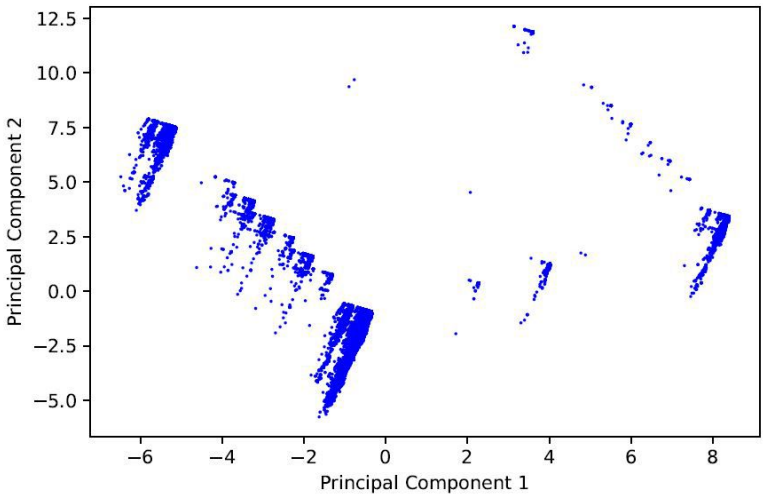


Figura 7.4: *Scatter plot* dos utilizadores conforme o seu histórico de adesões.

Assim, o modelo de *clustering* será treinado com informação que lhe permite definir melhor os *clusters* de clientes.

O modelo construído para a geração de recomendações tratou-se de um modelo de *clustering*, mais especificamente construído com o algoritmo K-Means do *PySpark*, com recurso à implementação disponível em [65]. Explicando de forma resumida o algoritmo, este atribui um *cluster* a cada utilizador presente no *data set*, de acordo com os *ratings* que este atribuiu às campanhas.

O número de *clusters* é previamente definido. Para tentar chegar a um valor adequado de número de *clusters* tendo em conta os dados em questão, foi utilizado o método do "cotovelo" ou *Elbow Method*, como descrito em [66]. Este método, através da análise da soma das distâncias ao quadrado de cada amostra (utilizador ao centro do *cluster* mais próximo. A análise é realizada para um número significativo de K's, sendo K o número de *clusters*. À medida que o valor de K aumenta, a soma das distâncias ao quadrado deverá tender para zero. Este será o comportamento natural, uma vez que se definirmos o número de K's como o seu valor máximo, ou seja, o número de utilizadores presente no

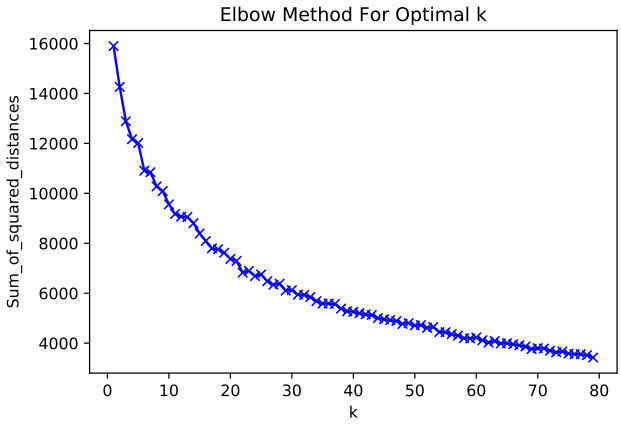


Figura 7.5: Gráfico com a soma das distâncias das amostras aos *clusters*, para diferentes números de *clusters*.

data set, a distância de cada um ao *cluster* mais próximo é zero. O valor de K adequado será o valor a partir do qual a curva das distâncias no gráfico começa a estabilizar.

O intervalo de número de *clusters* no qual este teste é aplicado foi escolhido tendo como base a análise à Figura 7.4. Observam-se alguns grupos de clientes, não excedendo um total de, aproximadamente, 50 grupos. O teste então poderá ser realizado entre os 0 e 80 *clusters*.

O *Elbow Method* não fornece uma verdade absoluta ou irrefutável acerca do número de *clusters* com que um determinado *data set* deverá ser treinado. Serve apenas de auxílio à escolha desse valor, podendo essa escolha depender de muitos outros fatores.

Analisando o gráfico da Figura 7.5, pode-se observar que o valor começa a estabilizar sensivelmente a partir de K igual a 40. Esse foi, portanto, o valor escolhido para o número de *clusters* do modelo. O gráfico da Figura 7.6 mostra a distribuição dos clientes, agrupados nos 40 *clusters*, distinguidos por cores. Conclui-se também da análise desta figura que o *clustering* foi bem realizado, uma vez que os clientes com a mesma cor – do mesmo *cluster* – estão juntos no espaço bi-dimensional.

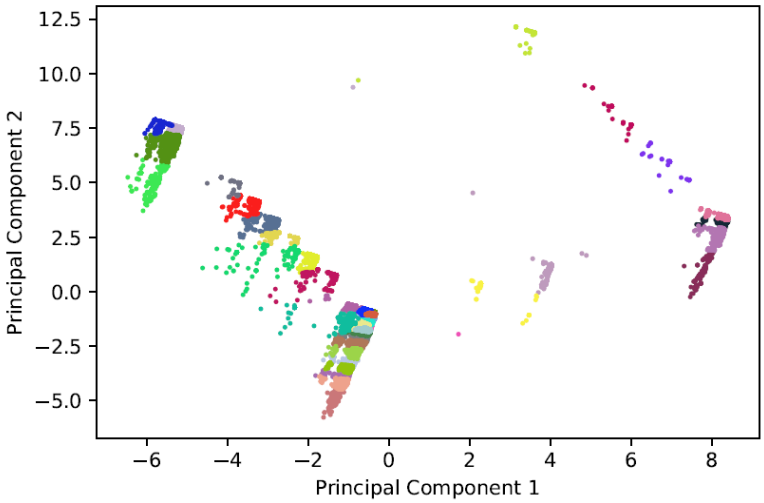


Figura 7.6: *Scatter plot* dos utilizadores conforme o seu histórico de adesões, agrupados em *clusters*.

As recomendações são dadas aos utilizadores consoante o *cluster* que ele esteja inserido. Todos os *clusters* possuem um conjunto de campanhas populares. Estas correspondem às campanhas que os utilizadores de cada *cluster* atribuíram um *rating* e a sua popularidade é calculada pela soma dos *ratings*. As recomendações que um utilizador recebe, com esta abordagem, são as campanhas mais populares no *cluster* no qual ele se encontra, exceto aquelas a que ele já atribuiu um *rating*.

Consegue-se perceber então, que desta forma, são recomendadas ao utilizador campanhas novas às quais ele nunca atribuiu um *rating*, um aspeto que poderá ser de interesse no âmbito das operadores de telecomunicações, dependendo da abordagem que estas queiram seguir para obter recomendações. As campanhas recomendadas ao utilizador consistem no *top* N da lista de campanhas, ordenada por popularidade. O valor de N pode variar.

7.3.1 Comparação com o Surprise

Ao contrário da abordagem com o Surprise, as recomendações obtidas não podem ser avaliadas em termos de métricas de precisão ou de erro uma vez que o algoritmo não funciona prevendo um *rating* que um utilizador daria a uma campanha. Sendo os algoritmos de *clustering* métodos não supervisionados, torna-se difícil perceber se as recomendações obtidas foram, de facto, úteis ou não para o utilizador.

Foi realizada uma comparação com as listas de recomendações obtidas com o Surprise. Foi escolhida uma amostra de 1000 clientes aleatórios dos quais foram retirados resultados do Surprise e do IBM Watson. As listas de recomendação correspondem ao top 20 de campanhas recomendadas aos utilizadores. A comparação foi realizada apenas aos algoritmos SVD e BaselineOnly, que foram aqueles que mostraram melhores resultados relativamente à abordagem com o Surprise.

As métricas escolhidas para avaliar a similaridade recomendações das duas abordagens foram:

- Contagem das campanhas comuns nas duas listas;
- Média das diferenças entre os *ranks* das campanhas comuns;

O gráfico da esquerda da Figura 7.7 mostra o número de utilizadores com campanhas em comum, comparados com o algoritmo SVD. Observa-se a maioria dos utilizadores obtiveram listas de recomendação com 5 campanhas em comum, das 20 possíveis. Relativamente ao algoritmo Baseline a maioria dos clientes foi recomendando com 3 campanhas em comum. Um resultado positivo seria obter listas de recomendação com um maior número de campanhas em comum dos dois sistemas. Significaria que os dois sistemas estavam em concordância relativamente às campanhas que deviam recomendar, e, portanto, haveria uma maior probabilidade dessas recomendações estarem corretas.

Tendo em conta as campanhas em comum para cada cliente, foram analisadas as posições que cada uma dessas campanhas ocupava na lista de recomendações. Para tal, foi calculada a média da diferença das posições para cada um dos clientes. Por exemplo: o utilizador X é recomendado, em ambos os sistemas, com a campanha A e com a campanha B. A campanha A é lhe recomendada na 10^{a} posição na lista obtida do Surprise e na 16^{a} posição na lista do IBM Watson. A campanha B é lhe recomendada na 3^{a} posição na lista obtida do Surprise e na 1^{a} posição na lista do IBM Watson. O valor desta métrica, para o utilizador X será: $\frac{|10-16|+|3-1|}{2} = 4$. Desta forma é possível perceber se os dois

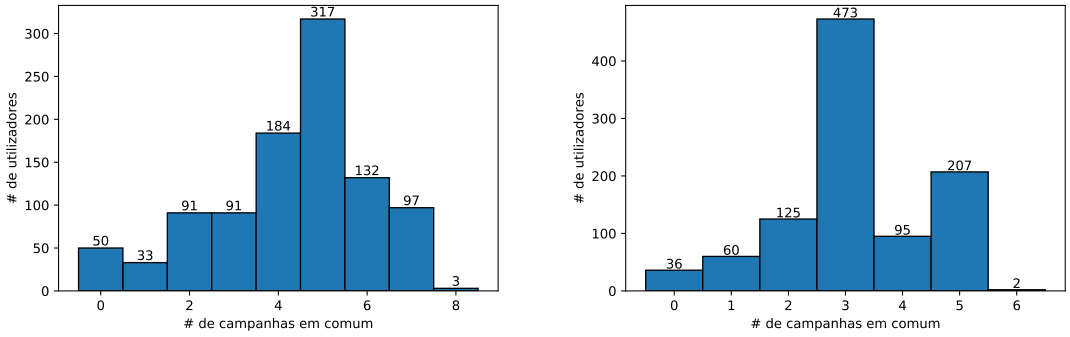


Figura 7.7: Histograma das campanhas comuns comparado com o algoritmo SVD (esquerda) e com o algoritmo BaselineOnly (direita).

sistemas recomendam a mesma campanha em posições muito diferentes na lista. Para vários utilizadores é depois calculada a média das médias.

Algumas estatísticas desta métrica podem ser vistas na Tabela 7.6, comparadas com os algoritmos SVD e BaselineOnly. A contagem de valores é diferente de 1000, que corresponderia aos 1000 utilizadores escolhidos, pois existem utilizadores que não possuem campanhas em comum nas listas de recomendação dos dois sistemas. A média de todos os clientes das médias das diferenças entre posições nas listas é relativamente baixa, o que significa que os dois sistemas estão a recomendar as mesmas campanhas aos utilizadores aproximadamente na mesma posição.

Tabela 7.6: Estatísticas da métrica de diferença média entre ranks, comparando o IBM Watson com o algoritmo SVD e com o algoritmo BaselineOnly.

Algoritmo	SVD	BaselineOnly
Contagem	948	962
Média	4.936	5.820
Mínimo	0	0
Máximo	18	16

7.3.2 Análise da ferramenta

A ferramenta IBM Watson foi explorada com o intuito de perceber se esta seria benéfica para o ACM e para servir como *host* do motor de recomendações de campanhas. Verificou-se que o IBM Watson possui uma grande componente de análise de dados, que permite a sua exploração e a possibilidade de encontrar relações que lhes são inerentes. Possui também várias ferramentas que permitem a visualização dos dados das mais variadas formas – gráficos interativos, grafos, tabelas,etc – e relações entre si. Para além da execução de *notebooks* Jupyter, como já referido no início da secção 7.3, permite treinar e guardar modelos de ML e efetuar o seu *deploy*.

A ferramenta revelou alguns problemas de *performance* com o plano *lite*, no que toca ao tempo de treino dos modelos para muitos clientes, daí a razão de ter utilizado um *sub-set* de clientes mais pequeno. Foi utilizado este plano uma vez que era o único que era grátis, e para esta fase de exploração não valia a pena utilizar um plano que envolvesse custos. O sistema também apresentou tempos de demora a devolver recomendações para um grupo

de 100 utilizadores bastante elevados.

De uma análise superficial realizada a esta ferramenta, que deu origem à decisão de a explorar, pensou-se que com o *notebook* utilizado, disponível em [64], iria fornecer um *front-end* que permitisse de forma intuitiva introduzir o cliente e obter as recomendações para ele. Acontece que, para um grande número de produtos recomendados, neste caso campanhas, este *front-end* não funciona.

A integração com sistemas externos ao IBM Watson é realizada através de *endpoints* gerados na ferramenta. O *deploy* dos modelos de ML, por exemplo, gera um *endpoint* que pode depois ser acedido por aplicações externas. Enviando um pedido contendo o histórico de adesões de um cliente em formato JSON, o motor devolve depois a resposta com as campanhas que lhe seriam recomendadas. O histórico de adesões de um cliente também pode ser passado através da interface do IBM Watson ao modelo *deployed*.

7.4 Utilização de características de cliente

Como referido na secção 7.1, os dados que os modelos receberam como treino consistem apenas nos clientes que possuem pelo menos uma adesão a qualquer campanha. Para clientes que não possuam histórico de adesões, é impossível recomendar-lhes campanhas uma vez que não existem dados que o permitam. Este é conhecido como o já referido problema do *Cold Start*. De modo a obter recomendações para os clientes que nunca aderiram a nenhuma campanha, foram utilizadas características dos clientes de uma operadora de telecomunicações.

Com recurso à informação das características de cada cliente em termos de saldos, recargas, tipo de tarifário, etc., foram criados *clusters* de clientes, baseados nessas características. Estes *clusters* tanto possuem clientes que já aderiram a campanhas como clientes que nunca aderiram a nenhuma. Caso um cliente que nunca tenha aderido a nenhuma campanha pertencer ao mesmo *cluster* que um cliente que já tenha adesões no seu histórico, pode ser recomendado com as campanhas que este último aderiu. A intuição desta abordagem é que clientes parecidos estarão, à partida, propensos a aderir às mesmas campanhas.

As próximas sub-secções explicam os métodos de tratamento de dados e redução da dimensionalidade que foram necessários efetuar antes de aplicar *clustering*, assim como os resultados obtidos desta abordagem.

7.4.1 Tratamento dos dados

As características de cliente presentes no *data set* podem-se dividir em categorias de:

- *Profilling* comportamental diário e mensal, que consiste em informação de carregamentos efetuados, quantidade de dias sem tráfego, etc.;
- *Profilling* de faturação e consumo mensal, que consiste em informação de saldos, consumos, quantidade de dados móveis, etc.;
- *Profilling* de tráfego GPRS mensal, que consiste em informação sobre sessões GPRS efetuadas pelos clientes;
- Propensão mensal, que consiste em informação sobre se o cliente possui banda larga ou *internet* móvel;

- Rentabilidade, mensal, que consistem em informação de métricas como Average Revenue Per User (ARPU) e sobre escalões de rentabilidade definidos pela operadora;
- Tráfego de voz originado e recebido, mensal, que consiste em informação acerca da quantidade de chamadas efetuadas e recebidas pelo cliente;
- Tráfego SMS recebido de voz e recebido, mensal, que consiste em informação acerca da quantidade de SMS's enviadas e recebidas pelo cliente;
- Informação de geografia e de dados pessoais do cliente, como distrito, freguesia onde vive, idade, sexo, tipo de tarifário, tipo de equipamento que este possui, entre outras.

O *data set* total de características contém informação sobre 6374127 clientes e engloba um total de 207 características, pertencentes às categorias acima mencionadas. Nem todos os clientes possuem informação de todas as características, ou seja, existem valores nulos no que toca a determinadas características para alguns clientes.

As Figuras 7.8 e 7.9 representam o número de clientes para o qual cada característica possui valores, de um total de 6374127 clientes.

É possível observar que a distribuição do número de valores para cada característica varia bastante. Algumas características apresentam valores para praticamente todos os clientes do *data set*, enquanto que outras possuem valores a nulo para grande maioria dos clientes.

Tendo em conta esta informação, verificou-se que algumas delas podiam ser, à partida, descartadas, pelas seguintes razões:

- são colunas com muitos valores a nulo;
- são colunas cujos valores consistem apenas numa descrição escrita de outras colunas;
- são colunas cujos valores são todos iguais;
- são colunas que são iguais a outras;

Relativamente às colunas com muitos valores a nulo, foram descartadas aquelas que possuem mais de 90% dos valores nessa condição, uma vez que contribuirão com pouca informação para os modelos. Após descartadas todas as colunas que se encontram nas condições referidas, o *data set* conta com um total de 160 características.

As características dos clientes consistem em colunas de valores contínuos e numéricos: características acerca do saldo médio, ARPU ou média de valores carregados do cliente, por exemplo, e colunas categóricas: freguesia e distrito, tipo de tarifário, escalão de consumo, entre outras. Das 160 características agora presentes no *data set*, 47 dessas foram consideradas como colunas categóricas.

Analisados os valores de cada características, verificou-se que algumas delas possuíam valores que deviam ser tratados como nulo. A razão disto é evitar passar para os modelos valores que tenham o mesmo significado, neste caso a ausência de informação, com representações diferentes. Esses valores são os seguintes:

- '-';
- 'I';
- 0 (numérico);
- -1 (*string*);
- 'N/E';
- 'X';
- -1 (numérico);
- -2 (*string*);

As substituições destes valores por valores a nulo, mais especificamente *NaN*, não foi realizada em todas as colunas. Apenas em algumas colunas os valores -1 devem ser entendidos

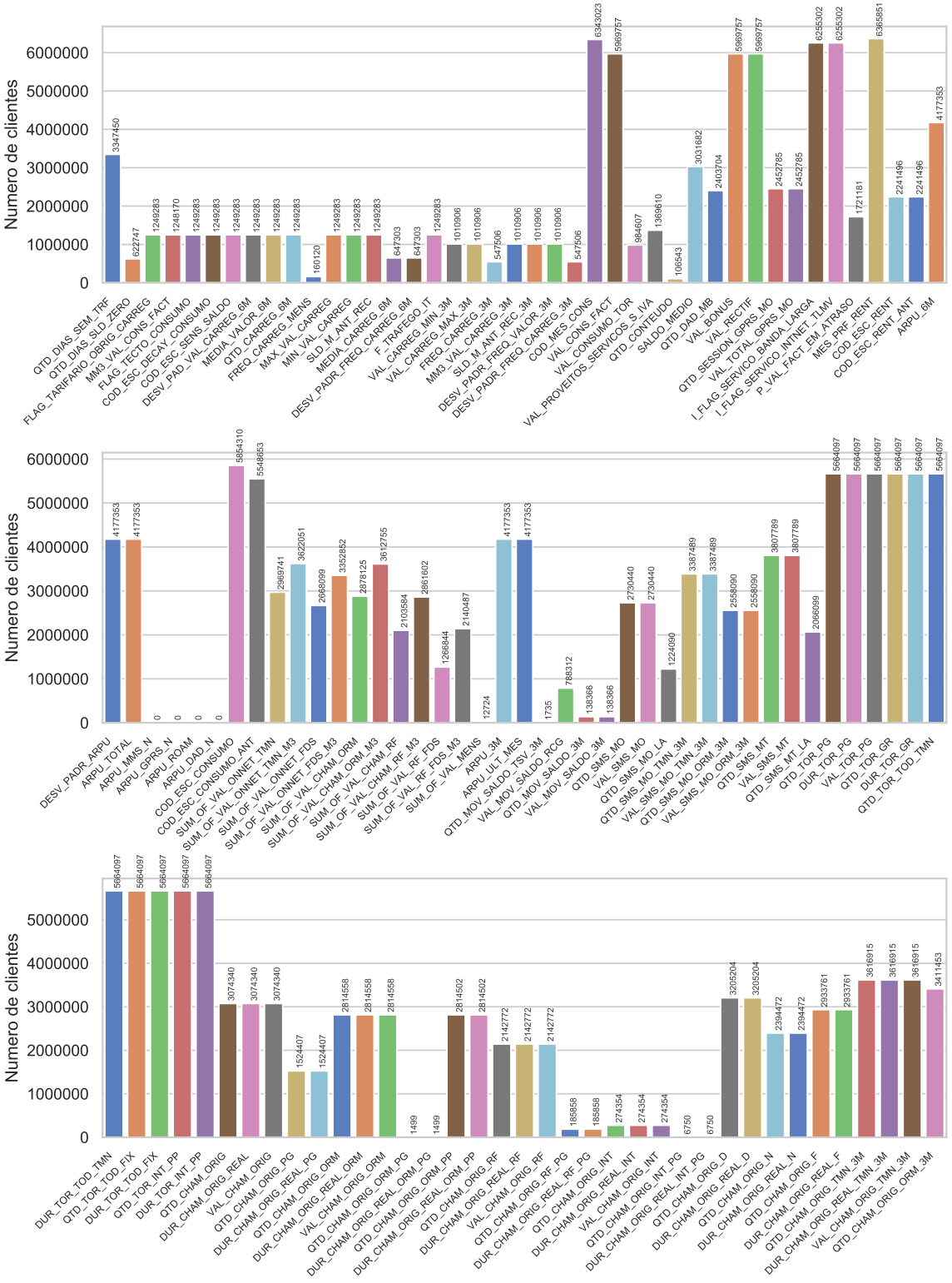


Figura 7.8: Número de clientes para o qual cada característica apresenta valores - parte 1.

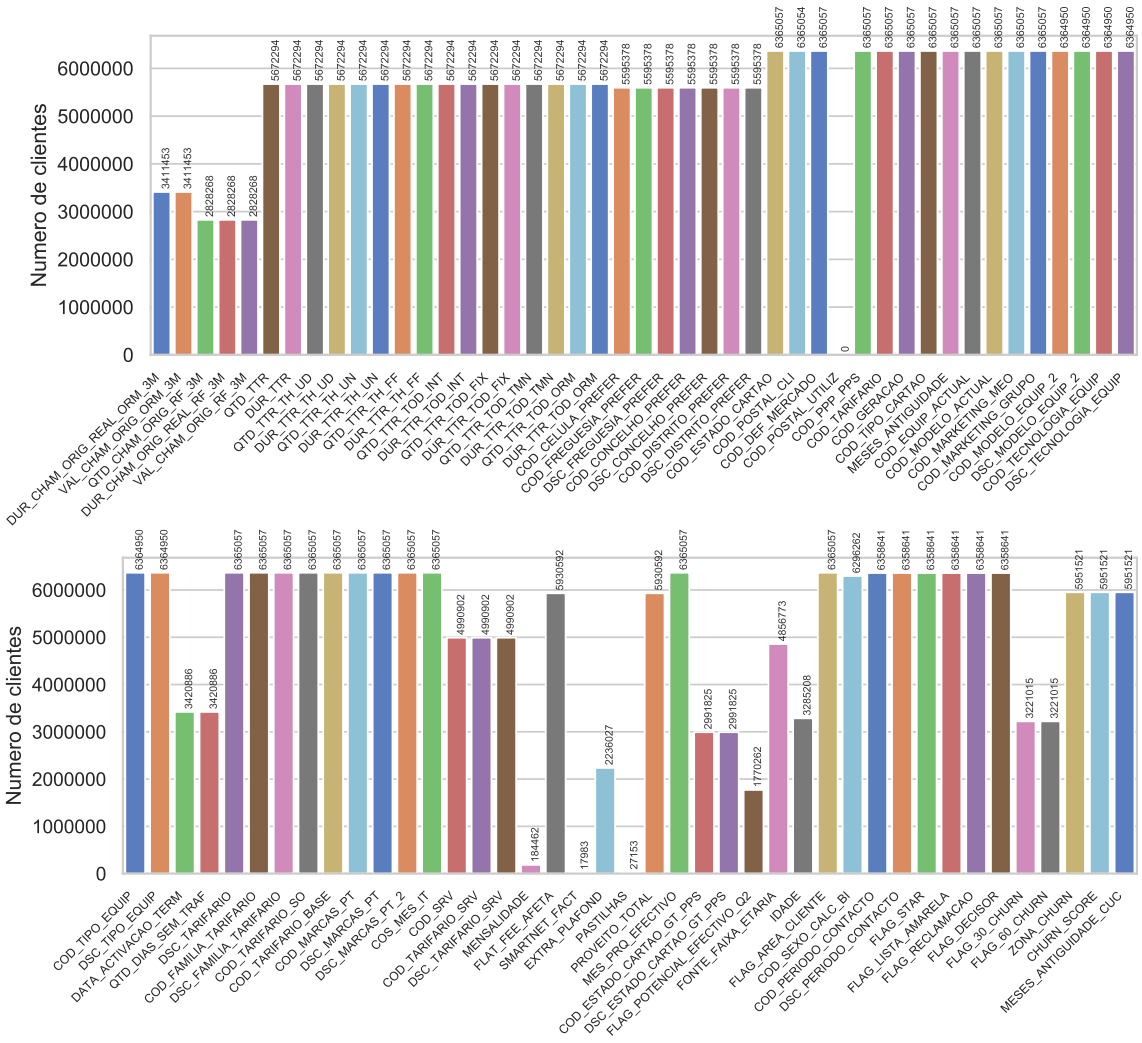


Figura 7.9: Número de clientes para o qual cada característica apresenta valores - parte 2.

como ausência de valor, como por exemplo nas colunas categóricas que representam um escalão/código. Em colunas de valores contínuos este valor já deverá ser interpretado como um valor correto. O mesmo acontece para o valor 0, que foi apenas substituído em colunas onde tal faça sentido, como são os casos das colunas *IDADE*, *SALDO_MEDIO* e *MESES_ANTIGUIDADE*, por exemplo. A substituição de alguns dos valores em determinadas colunas deveu-se a indicações fornecidas pela operadora de telecomunicações que disponibilizou as características, acerca dos valores que deveriam ser considerados como nulo.

Antes da aplicação de qualquer uma das abordagens descritas na secção 7.4.2, é preciso realizar um tratamento especial às colunas categóricas. Uma vez que estas possuem valores não numéricos, passá-las a um modelo de PCA ou calcular a correlação entre essas colunas e colunas numéricas seria errado. Sendo o objetivo final aplicar métodos de *clustering* ao *data set* com as características de clientes, uma possível abordagem seria aplicar o algoritmo *k-modes*[67], indicado para *clustering* de dados categóricos. Este algoritmo, ao invés da distância euclidiana entre as amostras e os centróides dos *clusters*, utiliza a dissimilaridade entre estes, que é mais adequada para valores categóricos. Também ao invés da média, utiliza os chamados *modes* que são os valores que ocorrem mais vezes, para determinar qual o elemento mais representativo do *cluster*. Uma vez que os dados neste estudo não são apenas categóricos, mas mistos (numéricos e categóricos), optou-se pela abordagem descrita em [68].

Resumidamente, o tratamento de colunas categóricas proposto consiste em substituir os valores de cada categoria de cada característica categórica pela frequência relativa dessa categoria. As Tabelas 7.7 e 7.8, adaptadas de [68], mostram um exemplo do tratamento realizado a três colunas categóricas. Na Tabela 7.7 é apresentado um valor a nulo na coluna Tarifário, para exemplificar o que acontece nesses casos.

Tabela 7.7: Valores fictícios para colunas categóricas.

Cliente	Sexo	Tarifário	Equipamento
001	M	B	E1
002	F	C	E1
003	F	A	E3
004	F	A	E2
005	M	NaN	E3
006	F	A	E4

A frequência relativa das categorias de cada uma das colunas categóricas da Tabela 7.7 é:

- **Sexo**:
  - M: $\frac{2}{6} = 0.33(3)$;
  - F: $\frac{4}{6} = 0.66(6)$;
- **Tarifário**:
  - A: $\frac{3}{5} = 0.6$;
  - B: $\frac{1}{5} = 0.2$;
  - C: $\frac{1}{5} = 0.2$;
- **Equipamento**:
  - E1 - $\frac{2}{6} = 0.33(3)$;

Tabela 7.8: Valores substituídos pela frequência relativa das categorias das colunas categóricas.

Cliente	Sexo	Tarifário	Equipamento
001	0.33(3)	0.2	0.33(3)
002	0.66(6)	0.2	0.33(3)
003	0.66(6)	0.6	0.33(3)
004	0.66(6)	0.6	0.16(6)
005	0.33(3)	NaN	0.33(3)
006	0.66(6)	0.6	0.16(6)

- E2 - $\frac{1}{6} = 0.16(6)$;
- E3 - $\frac{2}{6} = 0.33(3)$;
- E4 - $\frac{1}{6} = 0.16(6)$;

A substituição dos valores pela respetiva frequência relativa é realizada tal como é apresentado na Tabela 7.8.

Tanto a aplicação de PCA como o cálculo da correlação entre características foram feitos em *data sets* cujas características categóricas foram tratadas segundo esta abordagem. Devido à impossibilidade de juntar todas as características e clientes num único *data set* em tempo útil, devido ao grande número de clientes e características, para este estudo foi utilizado um sub-conjunto com menos clientes, mas que por sua vez também formam um *data set* mais rico em informação e com menos valores a nulo. A razão é que foram selecionados clientes que possuem valores para grande maioria das características. O *data set* de características utilizado neste estudo contém 1227941 clientes e as 160 características já mencionadas na secção 7.4.1. O *boxplot* da Figura 7.10 permite perceber melhor a proporção de clientes que têm muitas ou poucas características com valores nulos. Os pontos visíveis na parte superior revelam alguns *outliers*, sendo poucos clientes que tem um número de valores nulos superiores a 95, aproximadamente.

7.4.2 Redução de dimensionalidade

Mesmo após algumas das características terem sido descartadas, o *data set* ainda contém um número bastante elevado de colunas. Por essa razão, algumas técnicas para redução da dimensionalidade do *data set* foram aplicadas, de forma a obter-se um menor número de características, sem perda de informação relevante para os modelos de *clustering*.

Foram realizadas duas abordagens para redução de dimensionalidade:

1. através da aplicação de PCA;
2. através da eliminação de colunas correlacionadas.

Foram escolhidas estas duas abordagens distintas uma vez que, aplicando PCA fica-se em informação das características iniciais e através da eliminação de características correlacionadas, estas são mantidas. Tendo as duas abordagens, foi possível obter um termo de comparação entre as duas, e concluir acerca de qual delas fornece melhores resultados.

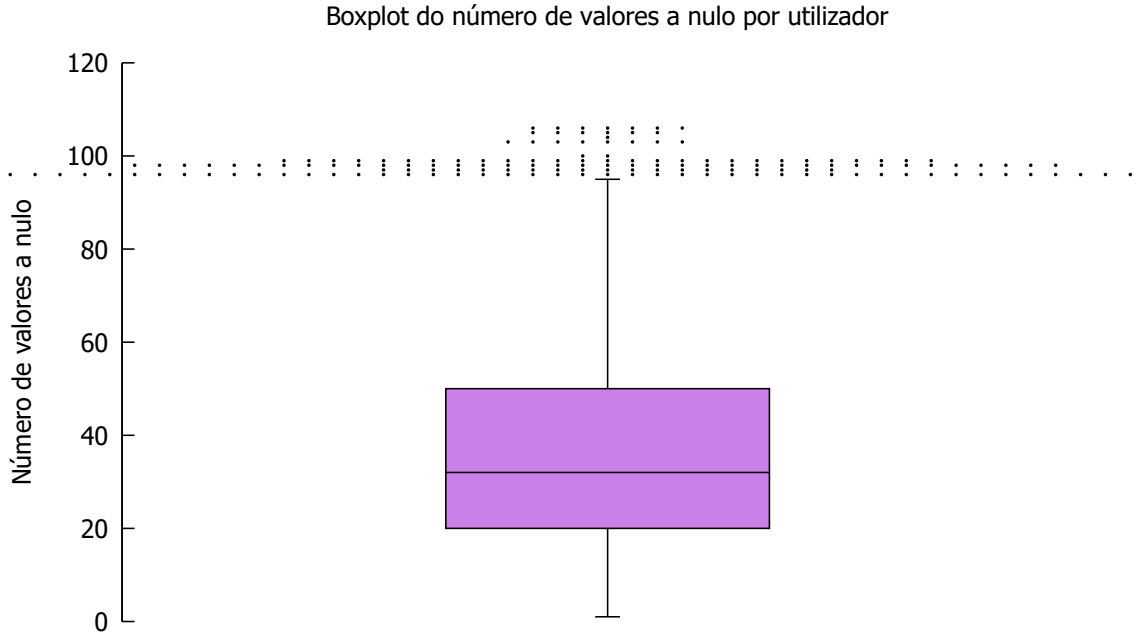


Figura 7.10: Gráfico boxplot do número de valores a nulo por utilizador.

7.4.2.1 *Principal Component Analysis*

Relativamente à abordagem nº1, o algoritmo PCA transforma um *data set* com muitas colunas, de grande dimensionalidade, num *data set* mais pequeno, com menos colunas. O que acontece é que as colunas do *data set* são transformadas em Principal Components (PC's) que contêm maior parte da informação presente nas colunas antigas. Estes PC's são independentes entre si e não correlacionados. A ideia com a utilização deste método é efetuar um *trade-off* de precisão por simplicidade e redução de dimensionalidade.

Antes de aplicar PCA, foi necessário estandardizar os dados, uma vez que as características nele contidas possuem valores de ordens bastante diferentes. Não realizar este passo poderá implicar que colunas com ordens de valores muito grandes tenham mais importância na formação dos PC's do que outras colunas, algo que não é desejável. Em [69], a importância de estandardizar os dados é evidenciada. É aplicado PCA a uma versão estandardizada de um *data set* e a uma versão não estandardizada do mesmo *data set*. Os *data sets* resultantes são utilizados depois para treinar modelos de classificação *Naive Bayes*, nos quais se verificou que a precisão dos modelos é menor com os dados não estandardizados.

Antes também de aplicar PCA, os valores a nulo foram substituídos pela média dos valores na coluna, uma vez que PCA não permite que lhe sejam passados valores a nulo no treino do algoritmo.

O algoritmo PCA foi aplicado com recurso à biblioteca *scikit-learn*, que contém uma implementação do algoritmo, disponível em [70]. Para este algoritmo, pode-se especificar o número de PC's desejados ou deixar o algoritmo escolhê-lo, de acordo com a percentagem de variância desejada. A variância é o parâmetro que indica qual a percentagem de informação que se pretende manter nos PC's gerados. O algoritmo irá escolher o número adequado para que essa percentagem mínima seja garantida. A percentagem de variância passada para o modelo PCA foi de 95%, tendo resultado na criação de 33 PC's. Houve portanto uma redução de 160 colunas para 33 novas colunas, independentes e não correlacionadas entre si.

Tabela 7.9: Variância, em percentegam, dos primeiros 10 PC's

Principal Component	Variância (%)
1	13.57
2	11.27
3	9.22
4	5.79
5	4.45
6	4.38
7	3.97
8	3.53
9	3.47
10	2.99

É possível observar na Tabela 7.9 a variância dos primeiros 10 dos PC's, ou seja, a percentagem de informação que cada um deles contém das 160 colunas anteriores. Observa-se que o primeiro PC contém cerca de 13% da informação das colunas, o segundo PC contém 11%, e assim sucessivamente. A soma da variância de todos os 33 PC's dá, aproximadamente, os 95% passados ao modelo.

7.4.2.2 Correlação entre características

No que toca à abordagem nº2 para redução da dimensionalidade do *data set* de características, foi calculada a correlação entre todas as características do *data set* mencionado na secção 7.4.1. A métrica de correlação utilizada foi o coeficiente de correlação de Pearson. O valor desta métrica varia entre -1 e 1 sendo que, o primeiro caso indica uma correlação negativa muito forte, ou seja, se os valores de uma das colunas aumenta, os valores da outra diminuem, e o segundo caso uma correlação positiva muito forte, ou seja, quando os valores de uma das colunas aumenta, os valores da outra também aumentam. Um valor igual a 0 do coeficiente de correlação de Pearson indicam que as duas colunas não possuem qualquer correlação entre si.

Foi calculada a matriz de correlação entre as características do *data set* de forma a perceber quais aquelas que tinham entre si uma forte correlação, quer fosse ela positiva ou negativa. O objetivo foi eliminar característica que estivessem correlacionadas com outras, uma vez que não trariam informação nova útil para os modelos de *clustering*. Foram então descartadas colunas que tivessem um coeficiente de correlação de Pearson maior que 0.80 e menor que -0.80, para descartar colunas correlacionados tanto positiva como negativamente.

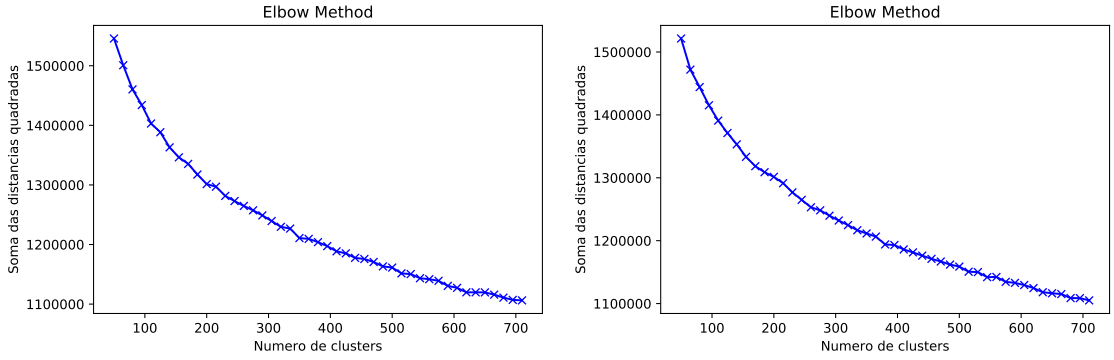
Sob estas condições, foram encontradas 52 colunas que puderam ser descartadas, passando portanto de um *data set* com 160 características para um com 107 características. Após este passo, as colunas com mais de 90% dos valores a nulo foram eliminadas e os dados foram estandardizados, tal como já foi explicado na secção 7.4.1 e na sub-secção 7.4.2.1, respetivamente.

7.4.3 *Clustering*

Após as tarefas de tratamento de dados e redução da dimensionalidade, ambos os *data sets* resultantes (provenientes do PCA e da eliminação das características correlacionadas) foram dados como treino a modelos de *clustering*.

Devido ao grande número de amostras presentes nos dados – 1227941 clientes – foi utilizada a implementação do algoritmo *K-Means* da biblioteca *dask-ml*, do Python, disponível em [71]. Foi utilizada esta implementação ao invés da mais habitualmente usada da biblioteca *scikit-learn*[56], disponível em [72]. A razão para tal é que a implementação utilizada demonstra tempos de treino mais rápidos e melhor *performance* em termos de utilização de memória, comparativamente à do *scikit-learn*. Para justificar esta decisão, foi treinado um sub-conjunto de 100000 clientes e 33 colunas, provenientes da aplicação de PCA. A média de tempos de treino dos modelos, com 5 medições, foi de 35 segundos com a implementação do *dask-ml* e 89 segundos com a implementação do *scikit-learn*.

De modo a determinar um número de *clusters* adequado aos dados, foi realizado o *Elbow Method* para os 2 *data sets* de características. Para estes testes foi definido um intervalo entre 50 e 700 *clusters*. Os gráficos dos testes realizados nos *data sets* podem ser observados nos gráficos da Figura 7.11.



(a) (b)

Figura 7.11: *Elbow Method* para o *data set* de características proveniente de PCA (Gráfico (a)) e para o *data set* com colunas correlacionadas eliminadas (Gráfico (b)).

Através da análise dos gráficos da Figura 7.11, pode-se assumir que um número adequado de *clusters* para o *data set* de características ao qual foi aplicado PCA é de 325, que é o valor aproximando a partir do qual a curva do gráfico começa a suavizar. Já para o *data set* de características correlacionadas descartadas, o número adequado de *clusters* é de 375.

7.4.3.1 Tempos de treino dos modelos

Foram medidos os tempos que os modelos *K-Means* demoraram a ser treinadas com cada um dos *data sets* de características e com números de *clusters* diferentes. A Tabela 7.10 apresenta os tempos medidos, em segundos.

Tabela 7.10: Tempos de treino, em segundos, dos modelos de *clustering*.

Data set	Número de clusters	Tempo de treino (segundos)
Características com PCA (dimensões: 1227941 x 33)	20	42.588
	50	97.328
	100	142.251
	150	244.791
	200	313.802
	250	443.459
	325	521.150
	500	951.164
	1000	2496.699
	2000	5122.836
	3000	10118.262
Características Correlacionadas Eliminadas (dimensões: 1227941 x 107)	20	127.106
	50	206.252
	100	292.179
	150	541.622
	200	508.584
	250	623.139
	375	1295.134
	500	1476.892
	1000	2937.332
	2000	6169.297
	3000	8995.424

A escolha destes números de *clusters* foi feita com o objetivo de se testar os modelos com números de *clusters* de diferentes grandezas e avaliá-los com as métricas mencionadas na sub-secção 7.4.3.2. Na Tabela constam o tempo de treino com o número de *clusters* que os *Elbow Methods* da Figura 7.11 mostraram como sendo um valor adequado.

7.4.3.2 Avaliação do *clustering*

O *K-Means*, tal como todos os métodos de *clustering*, é um método não-supervisionado, ou seja, não é fornecida para o treino dos modelos, qualquer informação acerca do resultado que é esperado obter, como acontece com modelos de classificação que recebem informação de quais as classes que devem atribuir a cada uma das amostras. O *K-Means* utiliza todas *features* dos dados, neste caso todas as características de clientes e campanhas aderidas por estes, para agrupar aqueles que são mais similares. Não é sabido, antes do treino dos modelos, em que *cluster* um cliente vai estar inserido nem quantos clientes vão estar dentro de um *cluster*. Esta informação prévia ao treino consistiria nos *ground truth labels* dos clientes e seriam os *clusters* corretos que cada cliente devia ser colocado. Quando existem *a priori ground truth labels* dos clientes, é possível avaliar a *performance* do modelo, no que toca à atribuição do *cluster* correto aos clientes.

Não havendo esta informação prévia à aplicação do *clustering* aos dados, apenas se podem aplicar métricas de avaliação que não necessitem dela. O *scikit-learn* possui uma métrica que permite a avaliação do modelo de *clustering* sem ser necessária a existência de *ground truth labels* das amostras dos dados de treino [73]. Esta métrica é chamada Índice de Davies-Bouldin.

Índice de Davies-Bouldin

A métrica Índice de Davies-Bouldin [74] revela a similaridade de cada *cluster* com o *cluster* mais parecido a si. A similaridade é calculada através do rácio entre as distâncias dentro do *cluster* e distância entre *clusters*. A equação 7.3 demonstra como a similaridade entre os *clusters* é calculada.

$$R_{i,j} = \frac{S_i + S_j}{d_{i,j}} \tag{7.3}$$

Na equação, S_i representa a distância média entre os pontos do *cluster* i e o seu centróide, e $d_{i,j}$ a distância entre os centróides dos *clusters* i e j. Para obter o Índice de Davies-Bouldin calcula-se a média das similaridades, para todos os k *clusters*, entre cada um deles e o seu *cluster* mais próximo, daí se calcular o máximo da similaridade. A métrica apresenta a seguinte fórmula:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{i,j} \tag{7.4}$$

Uma vez que o Índice de Davies-Bouldin é uma medida de similaridade, os melhores valores para esta métrica são os mais próximos de 0, o valor mínimo possível. É desejável que os *clusters* sejam o mais diferentes uns dos outros possível uma vez que isso significa uma melhor separação dos clientes com base nas suas características e nas suas adesões.

Resultados da métrica

Os gráficos da Figura 7.12 apresentam os valores do Índice de Davies-Bouldin conforme o número de *clusters* para os *data sets* das características. Na comparação da métrica para os dados de características, o gráfico foi partido em 2 por questões de apresentação e escala do eixo dos XX.

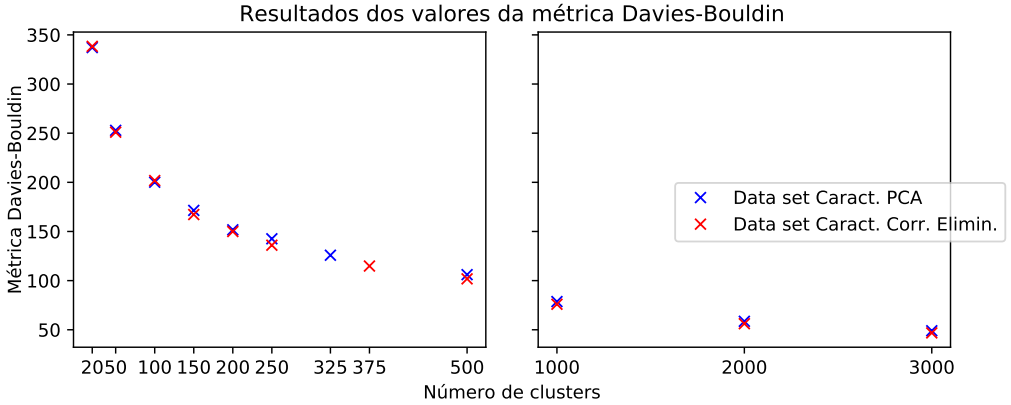


Figura 7.12: Comparação do Índice de Davies-Bouldin dos *data sets* de características.

É possível perceber que para os *data sets* de características, os Índices de Davies-Bouldin, que indicam a similaridade entre *clusters*, apresentam valores mais altos quanto menor o

número de *clusters*. Para poucos *clusters*, o Índice de Davies-Bouldin está aproximadamente no intervalo entre 200 e 350, o que para esta métrica é um valor muito elevado comparado com o desejável. Observa-se que, para 1000, 2000 e 3000 *clusters*, o valor do Índice já é relativamente baixo. Portanto, de acordo com este índice, deverá ser escolhido um número mais alto de *clusters* para garantir que haja alguma diferença entre eles.

Tendo em conta os resultados da métrica e outros métodos realizados – *Elbow Method* – considerou-se que um número adequado de *clusters* para os *data sets* de características seria de 2000. Resultados do *Elbow Method* para este número não são apresentados, mas extrapolando os resultados obtidos para um intervalo maior, considera-se que o comportamento seja semelhante, e que a soma das distâncias das amostras aos centróides dos seus *clusters* vá diminuindo à medida que estes aumentem, o que é um comportamento desejável.

7.4.4 Recomendações

As recomendações obtidas a partir da aplicação de *clustering* consistem nas recomendações mais populares dentro do *cluster* de características. A popularidade de uma campanha é dada pela soma dos *ratings* que esta recebeu dos clientes daquele *cluster*.

7.4.4.1 Processo de obtenção das recomendações

As campanhas recomendadas a clientes são baseadas nas suas características e nas campanhas que clientes semelhantes tenham atribuído um *rating*. O processo de obtenção de recomendações para um cliente, por esta abordagem, é realizado nos seguintes passos:

1. verificação de qual o *cluster* de características que o cliente se encontra;
2. identificação dos clientes que estão nesse *cluster*;
3. identificação das campanhas que esses clientes forneceram um *rating*
4. agrupamento das campanhas do ponto anterior por *rating* (soma dos *ratings* e ordenação por ordem decrescente);
5. identificação das campanhas:
   - que os *clientes* nunca aderiram;
   - com que os *clientes* nunca foram incentivados;
6. recomendação do top N de campanhas.

No ponto 4 é calculada a popularidade das campanhas através da soma dos *ratings* das campanhas dentro do *cluster*. Estas são depois ordenadas por ordem decrescente de acordo com esse valor.

São indicadas no ponto 5 duas maneiras de identificar as campanhas a serem recomendadas. Pode-se recomendar ao cliente campanhas populares no *cluster* que ele nunca tenha aderido ou que nunca lhe tenham sido incentivadas. A primeira é um tipo de recomendação que pretende que o cliente adira a campanhas que nunca aderiu, podendo já ter sido incentivado com elas ou não. A segunda pretende oferecer ao cliente ofertas que este até à data não tenha conhecimento e que são-lhe completamente novas.

O valor de N escolhido para o ponto 6 indica o top de campanhas que vão ser recomendadas para o cliente. As campanhas que são recomendadas ao cliente são as mais populares no *cluster*, calculadas no ponto 4, exceto aquelas que o cliente já aderiu, ou exceto aquelas com que ele já tenha sido incentivado, como referido no ponto 5.

Pela forma como foi implementada, esta abordagem, para além de devolver recomendações para os clientes que não tenham histórico de adesões, também devolve recomendações de campanhas a qualquer outro cliente, desde que possua características de cliente.

Esta abordagem revela uma desvantagem: caso exista um *cluster* de características que não tenha clientes com histórico de adesões, é impossível obter recomendações para clientes desse *cluster*. Este problema pode ser resolvido através da atribuição das recomendações do *cluster* de características mais próximo – calculada a distância euclidiana ou de Manhattan entre centróides dos *clusters* – e que tenha clientes com histórico de adesões aos clientes que não consigam obter recomendações. Esta solução, para já, não está implementada no sistema.

7.4.4.2 Comparação das recomendações obtidas com os diferentes *data sets*

Tal como referido na secção 7.4.2, foram consideradas duas formas de realizar o *clustering* com características de cliente, utilizando dois *data sets* diferentes. Primeiro, com um *data set* ao qual foi aplicado PCA e segundo com um *data set* com as características originais mantidas, mas as que eram correlacionadas com outras foram descartadas.

Nesta sub-secção é feita uma comparação entre as recomendações obtidas utilizando cada um desses *data sets* referidos. As campanhas recomendadas a cada cliente são campanhas em que este nunca foi incentivado, como referido no ponto 6 da sub-secção 7.4.4. É apenas este tipo de recomendações que esta sub-secção analisa uma vez que foi considerado o cenário mais interessante para a empresa poder aumentar as suas receitas através do incentivo a clientes nunca aderentes.

Foram analisadas as recomendações de um total de 688570 clientes que não possuem histórico de adesões e constam no *data set* de características. Os gráficos da Figura 7.13 dizem respeito às recomendações obtidas com o *data set* com PCA aplicado. Com este *data set*, as recomendações englobaram um total de 174 campanhas, sendo cada uma delas recomendada em diferentes quantidades, como se pode observar na figura. O gráfico *boxplot* mostra a distribuição dessas contagens.

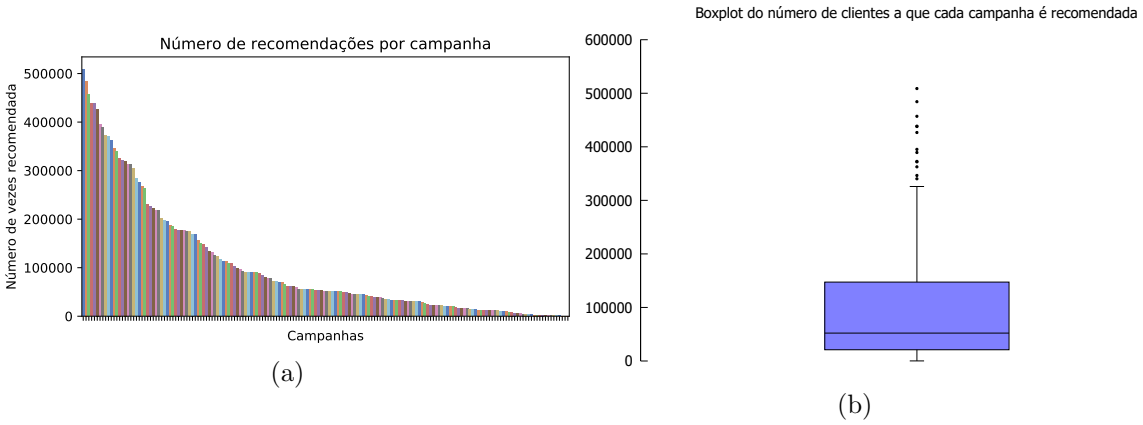


Figura 7.13: Número de clientes a que cada campanha foi recomendada ((Gráfico (a)) e *box-plot* dessa estatística (Gráfico (b)), para o *data set* com PCA aplicado.

Igualmente para o *data set* com características correlacionadas eliminadas, foi realizada a mesma análise e o resultado pode-se observar na Figura 7.14. Com este *data set*, as recomendações englobaram 173 campanhas.

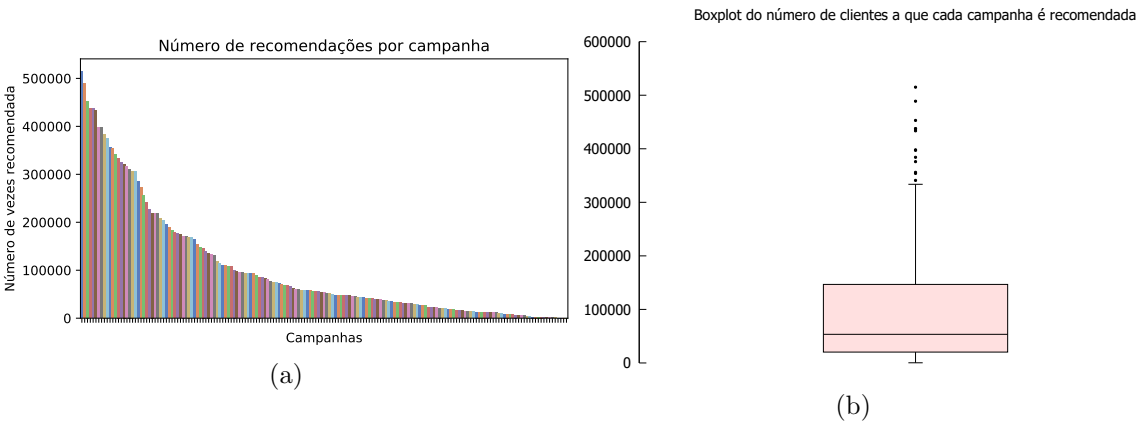


Figura 7.14: Número de clientes a que cada campanha foi recomendada ((Gráfico (a)) e *box-plot* dessa estatística (Gráfico (b)), para o *data set* com características correlacionadas eliminadas.

Observa-se que as recomendações obtidas com os dois *data sets* demonstram um comportamento bastante semelhante. Existem campanhas que são recomendadas a mais de 80% dos clientes e outras a menos de 500 clientes. Essa distribuição é demonstrada nos gráficos *box-plot* de ambas as figuras e revela também tal similaridade. A mediana de ambas as distribuições, indicada nos gráficos *box-plot* pela linha horizontal no meio da caixa, apresenta um valor baixo de 52032 com o *data set* da Figura 7.13, enquanto que com o *data set* da Figura 7.14 é de 53502, um valor bastante próximo.

Comparando as listas de campanhas recomendadas que cada cliente recebeu com os dois *data sets*, o número de campanhas em comum entre as duas listas, para cada cliente é, em média, 25.536, sendo o máximo o valor 30. Na Figura 7.15 observa-se que cerca de 100000 clientes possuem 28 campanhas em comum entre as duas listas.

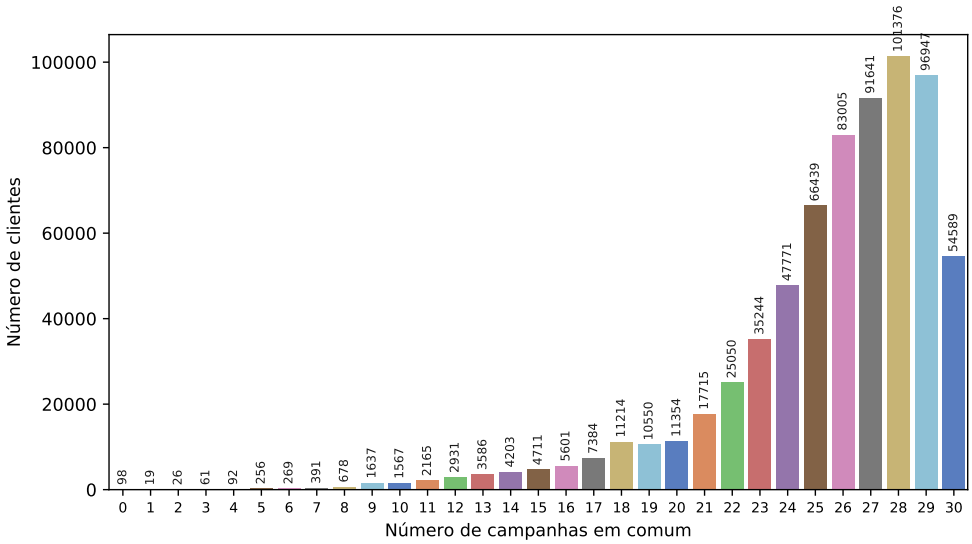


Figura 7.15: Relação entre o número de clientes e o número de campanhas em comum, comparando as listas de campanhas recomendadas com o *data set* com PCA e o *data set* com características correlacionadas eliminadas.

A diferença entre os *ranks* em que as campanhas comuns são recomendadas nas duas listas é muito baixo. Esta métrica é calculada, tal como descrito na sub-secção 7.3.1 e algumas estatísticas estão apresentadas na Tabela 7.11.

Tabela 7.11: Estatísticas da métrica de diferença média entre ranks, comparando o *data set* com PCA e o *data set* com características correlacionadas eliminadas.

Métrica	Diferença de Ranks
Contagem	687558
Média	4.106
Desvio padrão	2.066
Mínimo	0.0
25%	2.667
50%	3.955
75%	5.320
Máximo	29

As métricas 25%, 50% e 75% correspondem ao 25º, 50º e 75º percentil, respetivamente. O valor do 25º percentil, por exemplo, consiste no valor abaixo do qual 25% das amostras analisadas se encontram. Ou seja, 25% dos clientes apresentam uma diferença média de ranks nas campanhas comuns inferior a 1.185. Os valores dos outros percentis também são relativamente baixos, o que revela a similaridade dos ranks das campanhas comuns às duas listas de recomendação.

Tanto os valores desta métrica como a similaridade entre o número de clientes a que cada campanha foi recomendada são indicadores da parecença das recomendações obtidas. A escolha de um ou outro *data set* de características para obtenção das recomendações não deverá ter um grande impacto nas listas finais, e poderá ser baseada apenas na necessidade de manter as características iniciais intactas ou não.

Capítulo 8

Plano de gestão de riscos

O risco é um evento que pode ocorrer num projeto e que causa um impacto negativo no seu progresso. Os projetos possuem riscos que podem colocar em causa a sua realização ou a obtenção de um produto final de qualidade. Nas fases iniciais do projeto, é necessário analisar e registar os riscos que este possa sofrer e que possam colocar em causa a sua realização. Deve-se também desenvolver um plano de mitigação para que os riscos tenham o menor impacto possível no projeto.

8.1 Objetivos a cumprir

Em primeiro lugar, há que definir quais os objetivos que necessitam ser atingidos para que o projeto seja considerado como bem sucedido. Estes objetivos podem dividir-se em duas categorias:

8.1.1 Objetivos considerados para o sucesso do produto

- O projeto deve satisfazer os requisitos funcionais até à data final do estágio;
- O projeto deve satisfazer os requisitos não funcionais até à data final do estágio;
- A empresa deve ficar satisfeita com o produto final.

8.1.2 Objetivos considerados para o sucesso do processo

- As tarefas realizadas durante o estágio devem corresponder às tarefas que foram descritas no plano de trabalho, quer no tipo de tarefa, quer na sua duração;

8.2 Identificação dos riscos

Na Tabela 8.1, estão definidos os riscos deste projeto, sendo apresentada as suas descrições e consequências, caso ocorram.

Tabela 8.1: Riscos do projeto

ID	**Descrição do risco**	**Consequências**
1	Problemas de comunicação com a empresa.	O produto final não ser aceite pela empresa.
2	Inexperiência e/ou falta de conhecimento do desenvolvedor.	Demonstrar pouca produtividade e pouca qualidade no trabalho realizado.
3	Subvalorização dos tempos considerados para as tarefas descritas no plano de trabalho.	As tarefas demoram mais que o tempo considerado, o que gera atrasos nas entregas.
4	Realização do percurso académico em paralelo com a realização do projeto.	Diminuição do tempo que pode ser direcionado para as tarefas do projeto.
5	Requisitos do projeto incorretamente definidos.	Reformulação dos requisitos definidos a meio do projeto pode ser impossível.
6	Produto final não cumpre os requisitos definidos.	Produto final não possui as funcionalidades necessárias para atingir os objetivos requeridos.
7	Má utilização das ferramentas utilizadas em determinada tarefa.	Atrasos nas entregas e/ou produtos finais com pouca qualidade.
8	Riscos não identificados ou mal estimados.	Atrasos nas entregas e/ou produtos finais com pouca qualidade.
9	Dados não têm informação ou qualidade suficientes para a construção de um sistema de recomendação.	Requisitos do projeto não são satisfeitos.
10	Tecnologias escolhidas revelam-se pouco adequadas para o projeto.	Falha na implementação dos requisitos do projeto e atrasos nas entregas.
11	Tecnologias necessitam de dependências desconhecidas pelo desenvolvedor.	Atrasos nas entregas e/ou produtos finais com pouca qualidade.
12	Cobertura insuficiente dos testes efetuados.	Atrasos nas entregas e/ou produtos finais com pouca qualidade.
13	Execução dos testes necessita de dependências de bibliotecas externas desconhecidas pelo desenvolvedor.	Atrasos nas entregas e/ou produtos finais com pouca qualidade.
14	Plano de testes incorretamente definido.	Testes realizados não mostram possíveis *bugs* que o projeto contenha.
15	Relatório intermédio não apresenta qualidade.	Estágio interrompido.
16	Relatório final não apresenta qualidade.	Produto final com pouca qualidade e estágio repetido no próximo ano letivo.

8.3 Avaliação dos riscos

Após identificados, os riscos devem ser avaliados consoante as seguintes métricas: o impacto que poderão causar na realização do projeto, a probabilidade de ocorrerem e o tempo que demorarão a manifestar-se.

Em termos de impacto, os riscos podem ser catastróficos, críticos ou marginais. As consequências que cada categoria de impacto poderá ter no projeto são as seguintes:

- **Catastróficos**: risco que tem o potencial de afetar gravemente o processo, a performance e os custos associados à realização do projeto;
- **Críticos**: risco que tem o potencial de afetar levemente o processo, a performance e os custos associados à realização do projeto;
- **Marginais**: risco que tem o potencial de afetar relativamente pouco o processo, a performance e os custos associados à realização do projeto.

No que toca à probabilidade de ocorrência de um risco, esta divide-se em três níveis:

- **Alta**: previsível de ocorrer, com uma probabilidade maior que 70%;
- **Média**: pode vir a ocorrer, com uma probabilidade entre 40% a 70%;
- **Baixa**: não é previsível que venha a ocorrer, com uma probabilidade menor que 40%.

Relativamente ao tempo que demora desde a identificação de um risco até ser necessário lidar com ele, podem ser consideradas três categorias:

- **Longo**: Mais que três meses;
- **Médio**: Entre um e três meses;
- **Curto**: Menos que um mês.

Depois de definidas as métricas descritas acima, os riscos são classificados com base nestas, com o objetivo de determinar quais são os riscos que devem receber mais atenção e quais é que podem ser ignorados. As classificações de cada risco são apresentadas na Tabela 8.2.

Tabela 8.2: Avaliação dos riscos.

ID	Risco	Impacto	Probabilidade	Tempo
1	Problemas de comunicação com a empresa.	Crítico	Baixa	Longo
2	Inexperiência e/ou falta de conhecimento do desenvolvedor.	Catastrófico	Alta	Médio
3	Subvalorização dos tempos considerados para as tarefas descritas no plano de trabalho.	Crítico	Média	Longo
4	Realização do percurso académico em paralelo com a realização do projeto.	Crítico	Alta	Longo
5	Requisitos do projeto incorretamente definidos.	Catastrófico	Média	Médio
6	Produto final não cumpre os requisitos definidos.	Crítico	Média	Médio
7	Má utilização das ferramentas utilizadas em determinada tarefa.	Catastrófico	Média	Médio
8	Riscos não identificados ou mal estimados.	Crítico	Média	Longo
9	Dados não têm informação ou qualidade suficientes para a construção de um sistema de recomendação.	Catastrófico	Baixa	Médio
10	Tecnologias escolhidas revelam-se pouco adequadas para o projeto.	Catastrófico	Baixa	Longo
11	Tecnologias necessitam de dependências desconhecidas pelo desenvolvedor.	Crítico	Média	Longo
12	Cobertura insuficiente dos testes efetuados.	Crítico	Média	Longo
13	Execução dos testes necessita de dependências de bibliotecas externas desconhecidas pelo desenvolvedor.	Marginal	Baixa	Longo
14	Plano de testes incorretamente definido.	Marginal	Média	Longo
15	Relatório intermédio não apresenta qualidade.	Catastrófico	Média	Médio
16	Relatório final não apresenta qualidade.	Catastrófico	Média	Longo

A Tabela 8.3 apresenta a matriz de exposição aos riscos com base na classificação atribuída a cada um em termos de probabilidade e impacto.

Tabela 8.3: Matriz de exposição aos riscos.

	Impacto		
Probabilidade	Marginal	Crítico	Catastrófico
Alta		**4;**	**2;**
Média	**14;**	**3; 6; 8; 11; 12;**	**5; 7; 15; 16;**
Baixa	**13;**	**1;**	**9; 10;**

8.4 Plano de mitigação

Consoante a avaliação que receberam, os riscos deverão possuir uma medida de mitigação que identifica maneiras de prevenir a sua ocorrência e minimizar o seu impacto no projeto. A Tabela 8.4 apresenta o plano de mitigação para os riscos deste projeto.

Tabela 8.4: Plano de mitigação dos riscos do projeto.

Plano de mitigação	**Riscos envolvidos**
Estudo prévio das tecnologias, realização de tutoriais e treino com as tecnologias a serem utilizadas.	2; 7; 11;
Requisitos devem ser revistos pelas várias entidades envolvidas no projeto, nomeadamente desenvolvedor e orientadores.	5;
Relatório deve passar por várias iterações pelos orientadores de forma a ser dado feedback sobre o mesmo.	15; 16
Estimação do tempo das tarefas do plano de trabalho devem ser reformuladas considerando o tempo que já levam de atraso.	3
Realização de várias reuniões com os orientadores para perceber se o que está a ser realizado é o que é esperado.	6
Plano de riscos deve ser revisto pelas várias entidades envolvidas no projeto e realizadas as reformulações necessárias.	8
Alteração ao plano de testes, com adição de mais testes ou reformulação dos existentes.	12
Aquisição de novos dados com informação útil para recomendações	9
Alteração das tecnologias a utilizar.	10

Esta página foi intencionalmente deixada em branco.

Capítulo 9

Plano de Trabalho

O planeamento do trabalho a ser realizado durante o estágio foi feito com recurso a alguns métodos tais como o *Work Breakdown Structure* (WBS) e Estimativa de Três Pontos. O WBS é uma metodologia que consiste na divisão de tarefas mais extensas em tarefas mais pequenas, permitindo assim a gestão mais facilitada de um projeto.

Existem regras no que toca à determinação das tarefas mais pequenas, nomeadamente a regra 8/80 que implica que nenhuma tarefa dure menos que 8 horas ou mais que 80 a ser concluída. Caso uma tarefa dure mais que 80 horas, precisa de ser decomposta em sub-tarefas [75]. A estimação do esforço necessário para a realização de cada tarefa foi feita utilizando o método Estimativa por Três Pontos [76]. Para o cálculo do esforço, o método considera três valores de esforço para cada tarefa e aplica depois uma fórmula sobre eles.

Os valores de esforço considerados são os seguintes:

- Melhor caso, onde tudo corre bem;
- Caso mais provável, onde tudo corre como esperado e dentro da normalidade;
- Pior caso, onde tudo corre mal.

A fórmula aplicada aos valores referidos acima é a seguinte:

$$E = \frac{MC + (4 * CMP) + PC}{6} \tag{9.1}$$

onde E é o esforço estimado, em dias, para a tarefa. As variáveis MC, CMP e PC são, respetivamente, o melhor caso, o caso mais provável e o pior caso, todos também medidos em dias.

9.1 1º Semestre

O planeamento do 1º semestre foi realizado considerando 3 dias de trabalho por semana, uma vez que este é o tempo disponível semanal para o 1º semestre. As tarefas consideradas para este semestre são:

1. Conceitos básicos de ML e análise de Sistema de Recomendação (SR)
2. Planeamento do estágio
3. Contextualização do projeto
4. Escrita do relatório
   - 4.1. Introdução
   - 4.2. Estado da arte
   - 4.3. Plano de riscos
   - 4.4. Levantamento de requisitos
   - 4.5. Revisão do relatório
5. Preparação da defesa

Na Tabela 9.1 é apresentada uma breve descrição das tarefas que foram consideradas para o plano.

Tabela 9.1: Descrição das tarefas consideradas para o 1º semestre.

Tarefa	**Descrição**
1	Realização de um curso online de *Machine Learning* (www.coursera.org) com o objetivo de fornecer algumas bases de ML e análise de sistemas de recomendação e seu funcionamento.
2	Planeamento do tempo e das tarefas a realizar no decurso do estágio.
3	Aquisição de uma primeira visão geral do projeto e do produto ACM, obtida através da leitura de documentos como o *user's manual* do produto e através de explicações dadas pelos orientadores e membros da equipa do ACM.
4	Escrita do relatório intermédio de estágio.
4.1	Escrita da introdução do relatório. Este capítulo contém a descrição do contexto do projeto, a definição do problema a ser resolvido e os objetivos que são esperados alcançar com a realização do estágio.
4.2	Escrita do estado da arte sobre sistemas de recomendação.
4.3	Análise dos riscos e das consequências que estes possam ter na realização do projeto.
4.4	Definição dos requisitos funcionais e não funcionais que o sistema deverá ter para os objectivos propostos
4.5	Revisão da escrita do relatório e afinamento de detalhes, com base em *feedback* recebido acerca destes aspetos.
5	Preparação da defesa intermédia, nomeadamente a elaboração de um documento em formato PowerPoint, para a apresentação do relatório.

Na Tabela 9.2 encontram-se as estimativas do tempo necessário para realizar cada uma das tarefas, utilizando o método de Estimação de Três Pontos.

Tabela 9.2: Estimação do tempo de cada tarefa.

Tarefa	**Melhor caso (dias)**	**Caso mais provável (dias)**	**Pior caso (dias)**	**Caso expectado (dias)**
1	48	63	81	64
2	3	6	9	6
3	6	9	12	9
4	-	-	-	-
4.1	9	12	15	12
4.2	27	36	48	37
4.3	2	4	6	4
4.4	3	6	9	6
4.5	6	9	12	9
5	3	4	6	4

Na Tabela 9.3 é apresentado o planeamento do 1º semestre. Há que ter em atenção que algumas das tarefas são sobrepostas com outras.

Tabela 9.3: Plano do 1º semestre.

Tarefa	**Data de início**	**Data de fim**	**Duração**
Conceitos básicos de ML e análise de SR	5/Sep	30/Jan	64
Planeamento do estágio	12/Sep	21/Sep	6
Contextualização do projeto	26/Sep	12/Oct	9
Escrita do relatório intermédio	26/Sep	18/Jan	51
Introdução	26/Sep	19/Oct	12
Estado da arte	5/Oct	28/Dec	37
Plano de Riscos	30/Nov	7/Dec	4
Levantamento de Requisitos	12/Dec	21/Dec	6
Revisão do relatório	2/Jan	18/Jan	9
Preparação da defesa	24/Jan	31/Jan	4

9.2 2º Semestre

O planeamento do 2º semestre foi realizado considerando 5 dias de trabalho semanais. As tarefas consideradas para este semestre são:

1. Revisão do projeto
2. Implementação
   2.1. Análise dos dados
   2.2. Tratamento dos dados
   2.3. Construção da matriz Utilizador×Item
   2.4. Implementação de regras de associação
   2.5. Geração de recomendações com recurso a *frameworks open-source*
   2.6. Geração de recomendações através de SR licenciados

3. Testes
    3.1. Avaliação da precisão das recomendações
    3.2. Avaliação da *performance* das recomendações
4. Escrita do relatório final
5. Revisão
6. Preparação da defesa final

A Tabela 9.4 apresenta a descrição das tarefas consideradas para o plano.

Tabela 9.4: Descrição das tarefas consideradas para o 2º semestre.

Tarefa	**Descrição**
1	Revisão do relatório, do planeamento e do projeto de acordo com os comentários recebidos na apresentação do relatório intermédio.
2	Todas as tarefas que envolvem a implementação de código com o objetivo de obter o produto final.
2.1	Análise dos dados disponibilizados de modo a ser feita a avaliação de que informação será útil ou não.
2.2	Alteração na forma como os dados são representados de modo a que o resultado seja o melhor possível. Ter em atenção se existem valores indefinidos ou valores em falta, por exemplo, e efetuar alterações aos dados em conformidade com o que se observou, tais como inferência ou remoção de valores.
2.3	Construção da matriz Utilizador×Item que mostra que clientes aderiram a que campanhas.
2.4	Implementação do algoritmo Apriori que gera as regras de associação entre campanhas.
2.5	Implementação de um motor de recomendações com recurso a *frameworks* Python como o Surprise, LightFM ou CaseRecommender.
2.6	Geração de recomendações com recurso a sistemas de recomendação licenciados como o serviço *Watson Machine Learning* da IBM.
3	Planeamento e realização dos testes anteriormente definidos e alterações ao sistema em conformidade com os resultados dos testes.
3.1	Avaliação da eficácia das recomendações obtidas através das métricas anteriormente faladas.
3.2	Avaliação da *performance* do sistema em termos de tempo de execução.
4	Escrita do relatório final de estágio. Este deverá conter uma descrição do trabalho realizado e dos resultados obtidos, assim como as conclusões tiradas.
5	Revisão da escrita do relatório e afinamento de detalhes, com base em *feedback* recebido acerca destes aspetos.
6	Preparação da defesa final, nomeadamente a elaboração de um documento em formato PowerPoint, para a apresentação do relatório.

Na Tabela 9.5 encontram-se as estimativas do tempo necessário para realizar cada uma das tarefas, utilizando o método de Estimação de Três Pontos.

Tabela 9.5: Estimação do tempo de cada tarefa.

Tarefa	**Melhor caso (dias)**	**Caso mais provável (dias)**	**Pior caso (dias)**	**Caso expectado (dias)**
1	4	4	5	4
2	-	-	-	-
2.1	3	5	10	6
2.2	5	9	14	9
2.3	3	6	14	7
2.4	5	9	17	10
2.5	8	14	20	14
2.6	9	15	22	15
3	-	-	-	-
3.1	2	5	8	5
3.2	2	4	6	4
4	10	17	20	16
5	4	8	19	9
6	3	6	7	6

A Tabela 9.6 apresenta o planeamento do 2º semestre.

Tabela 9.6: Plano do 2º semestre.

Tarefa	**Data de início**	**Data de fim**	**Duração**
Revisão do projeto	4/Feb	7/Feb	4
Implementação	8/Feb	13/May	63
Análise dos dados	8/Feb	15/Feb	6
Tratamento dos dados	18/Feb	28/Feb	9
Construção da matriz Utilizador×Item	1/Mar	12/Mar	7
Implementação de regras de associação	13/Mar	26/Mar	10
Geração de recomendações com recurso a *frameworks open-source*	27/Mar	12/Apr	13
Geração de recomendações através de *software* empresarial	15/Apr	8/May	15
Testes	9/May	17/May	7
Avaliação da eficácia das recomendações	9/May	14/May	4
Avaliação da *performance* das recomendações	15/May	17/May	3
Escrita Relatório Final	20/May	11/Jun	16
Revisão do relatório	12/Jun	25/Jun	9
Preparação da defesa final	1/Jul	8/Jul	6

No capítulo de anexos deste documento seguem os diagramas de Gantt relativos ao planeamento realizado para o 1º semestre e para o 2º semestre.

9.3 Desvios ao plano estimado

O plano do 1º semestre foi seguido praticamente como foi desenhado, com as tarefas descritas nesse semestre a serem realizadas e sem grandes desvios em termos dos tempos que foram estimados. Relativamente ao 2º semestre, devido a algumas das tarefas terem sido substituídas por outras, existiu um grande desvio no plano. Haviam inicialmente 3 abordagens definidas, sendo elas a implementação de regras de associação, a utilização de *frameworks open-source* como o Surprise e a exploração do *IBM Watson*.

A abordagem que iria implementar as regras de associação baseadas no histórico de adesões dos clientes foi substituída pela utilização das características de cliente, uma vez que esta consistiria numa abordagem diferente de obter recomendações. Por essa razão, considerou-se que a abordagem seguida iria contribuir mais para o estudo realizado no âmbito deste estágio. A ordem pela qual as tarefas foram realizadas também foi alterada, comparativamente ao plano delineado no 1º semestre. O trabalho descrito na secção 7.2 resultou na escrita de um artigo científico, realizado em simultâneo com outras tarefas deste plano. O tempo despendido nesta tarefa foi também um desvio ao plano inicial.

A Tabela 9.7 apresenta as tarefas que foram executadas durante o 2º semestre do estágio.

Tabela 9.7: Plano real do 2º semestre.

Tarefa	**Data de início**	**Data de fim**	**Duração**
Revisão do documento e incorporação dos comentários da defesa intermédia	4/Fev	7/Fev	4
Análise dos dados do histórico de adesões	8/Fev	13/Fev	6
Transformação de *feedback* implícito em *feedback* explícito	14/Fev	19/Fev	4
Surprise			
- Implementação de algoritmos de recomendação com o Surprise	20/Fev	8/Mar	13
- Avaliação dos resultados	11/Mar	29/Mar	15
Escrita do artigo	1/Apr	19/Apr	15
IBM Watson			
- Exploração do *software* empresarial IBM Watson	1/Apr	17/Abr	13
- Construção da matriz Utilizador x Campanha	3/Abr	5/Abr	3
- Comparação das recomendações do IBM Watson com as obtidas do Surprise	5/Abr	14/Abr	6
Características de cliente			
- Análise dos dados das características de cliente	15/Abr	24/Abr	8
- Junção de todas as características num único *data set*	24/Abr	30/Abr	5
- Tratamento dos dados das características de cliente	1/Maio	8/Maio	6
- Aplicação de PCA	9/Maio	13/Maio	3
- Construção da matriz de correlação entre características	14/Maio	20/Maio	5
- Aplicação e validação dos modelos de *clustering*	20/Maio	29/Maio	10
- Implementação do sistema de recomendações a partir dos *clusters*	30/Maio	3/Jun	3
- Avaliação dos resultados	3/Jun	7/Jun	5
Escrita do Relatório Final	27/Maio	7/Jun	10
Revisão do Relatório Final	7/Jun	29/Jun	15
Preparação da defesa final	1/Jul	8/Jul	6

Esta página foi intencionalmente deixada em branco.

Capítulo 10

Conclusões

As abordagens que foram seguidas no âmbito deste estágio permitiram a geração de recomendações de campanhas baseadas em 2 fatores: primeiro, apenas baseadas no seu histórico de adesões e, segundo, baseadas no cruzamento de características dos clientes com o histórico de adesões. Considerou-se que, abordando o problema por estas duas vias, o estudo de sistemas de recomendação aplicado ao ACM tornava-se mais completo e enriquecedor.

Foram efetuadas algumas tarefas de tratamento dos dados, prévias à aplicação dos modelos e algoritmos de recomendação, sendo nomeadamente a transformação de *feedback* implícito em *feedback* explícito. Esta transformação permitiu que os incentivos e adesões dos clientes a campanhas fossem transformadas num *rating* numérico. Para a obtenção das recomendações, foram analisados algoritmos de filtragem colaborativa e métodos de *clustering*, aliado, este último, com técnicas de redução de dimensionalidade.

Relativamente à abordagem com o Surprise, que utiliza apenas o histórico de adesões dos clientes, foram realizadas experiências com vários algoritmos presentes na biblioteca. Com esta abordagem, obtém-se a previsão de um *rating* que um cliente daria a uma campanha, e decide-se conforme esse valor, se ele deverá ser incentivado com a campanha. Esta biblioteca possui vários algoritmos de filtragem colaborativa para gerar recomendações, que foram posteriormente avaliadas em termos de *precision*, *recall*, *mean average precision*, *mean reciprocal rank*, entre outras.

Foi explorada também a ferramenta IBM Watson, um serviço na *cloud* que permite a execução de modelos de ML. Nesta ferramenta foi testado um sistema de recomendação que utilizou *clustering*, baseado no histórico de adesões dos clientes. Os resultados desta abordagem são uma lista de campanhas recomendadas para cada cliente e foram comparados com os resultados da abordagem com o Surprise. A exploração desta ferramenta complementou ainda a análise ao histórico de adesões dos clientes, permitindo que tivessem sido tiradas algumas conclusões sobre esses dados.

Com a utilização de características de cliente, pretendeu-se resolver o problema de *cold-start*, ou seja, o que recomendar a clientes que não tenham qualquer histórico de adesões. A ideia desta abordagem foi recomendar campanhas que clientes semelhantes em termos de características tenham aderido. Estas recomendações foram obtidas através da aplicação do algoritmo de *clustering K-Means*. Por consistir numa abordagem que oferece uma solução a um dos erros mais comuns em sistemas de recomendação de filtragem colaborativa, conclui-se que a terceira abordagem que utiliza as características de cliente consiste na abordagem mais completa. Através dela, consegue-se a obtenção de recomendações

quer para clientes que tenham histórico de adesões, quer para clientes que não tenham.

Foram pensadas formas de avaliar as recomendações, de qualquer uma das abordagens, num cenário de produção, com clientes reais a serem incentivados. As adesões seriam registadas e seria possível inferir acerca da eficácia do sistema de recomendações implementado. Porém, esta avaliação consiste no trabalho futuro a realizar, uma vez que já não foi possível ser feita no âmbito deste estágio.

10.1 Trabalho Futuro

Foram consideradas algumas formas de avaliar as recomendações obtidas das vários abordagens, utilizando clientes reais que iriam ser incentivados com as campanhas que lhe foram recomendadas. Esta avaliação consiste em trabalho futuro uma vez que já não foi possível ser realizada no âmbito deste estágio. Isso deve-se ao facto de implicar a introdução do trabalho realizado neste estudo num contexto de produção e implica também envolvimento direto da operadora de telecomunicações que forneceu os dados, algo que pode consistir num processo ainda demorado.

O método de avaliação acenta numa funcionalidade do ACM que permite a escolha de quais clientes vão ser incentivados e com que campanhas, denominada *Target Import*. Utilizando esta funcionalidade, considera-se selecionar os clientes que nunca efetuaram nenhuma adesão a campanhas e incentivá-los com as que lhes foram recomendadas usando a abordagem com utilização das características de cliente. O objetivo é perceber se as recomendações estão realmente a ter um efeito positivo nos clientes e se estes expressam vontade em aderir a campanhas que clientes semelhantes a si também aderiram. Esta fase do trabalho iria consistir na fase de *deployment* do processo CRISP-DM, referido no final da secção 1.3.

10.2 Contribuições

O trabalho realizado no âmbito deste estágio resultou em diversos contributos.

Primeiro, do ponto de vista académico, através da análise do estado de arte e pelo conhecimento que foi possível obter com a investigação feita. Ambos permitem concluir acerca da inovação deste trabalho na área das operadoras de telecomunicações e na vertente de sistemas de recomendações usando *feedback* implícito.

Segundo, do ponto de vista industrial, em que foi analisada e prototipada a possibilidade de ter um sistema de recomendações para o produto ACM.

Adicionalmente, do estudo realizado no âmbito deste estágio, resultou o artigo *A Recommender System for Telecommunication Operators' Campaigns*, aceite na *19th EPIA Conference on Artificial Intelligence*, uma conferência europeia sobre Inteligência Artificial. O artigo focou-se na abordagem que utilizou o Surprise, descrita na secção 7.2.

Bibliografia

[1] An overview of recommendation systems. http://datameetsmedia.com/an-overview-of-recommendation-systems/. Consultado em Janeiro, 2019.

[2] Recommender systems – user-based and item-based collaborative filtering. https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering\-5d5f375a127f. Consultado em Janeiro, 2019.

[3] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011.

[4] Four problems in using crisp-dm and how to fix them. https://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html. Consultado em Dezembro, 2018.

[5] 2018 reform of eu data protection rules. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en. Consultado em Dezembro, 2018.

[6] What is the crisp-dm methodology? https://www.sv-europe.com/crisp-dm-methodology. Consultado em Dezembro, 2018.

[7] Phase 4 of the crisp-dm process model: Modeling. https://www.dummies.com/programming/big-data/phase-4-of-the-crisp-dm-process-model-modeling/. Consultado em Dezembro, 2018.

[8] Active campaign manager. http://www.alticelabs.com/site/acm/. Consultado em Fevereiro, 2019.

[9] F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261 – 273, 2015.

[10] S. Sivapalan, A. Sadeghian, H. Rahnama, and A. M. Madni. Recommender systems in e-commerce. In *2014 World Automation Congress (WAC)*, pages 179–184, Aug 2014.

[11] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19, December 2015.

[12] How retailers can keep up with consumers. https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers. Consultado em Outubro, 2018.

[13] Youtube's ai is the puppet master over most of what you watch. https://www.cnet.com/news/youtube-ces-2018-neal-mohan/. Consultado em Novembro, 2018.

[14] Aplicando sistemas de recomendação em situações práticas. `https://www.ibm.com/developerworks/br/local/data/sistemas_recomendacao/index.html`. Consultado em Setembro, 2018.

[15] Laurent Candillier, Kris Jack, Françoise Fessant, and Frank Meyer. State of the art recommender system. pages 1–22, 04 2009.

[16] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[17] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2):81–173, February 2011.

[18] Peter Boström and Melker Filipsson. Comparison of user based and item based collaborative filtering recommendation services, 2017.

[19] Wee Sun Lee. Collaborative learning for recommender systems. 06 2001.

[20] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.

[21] Al Rashid, Istvan Albert, Dan Cosley, Shyong Lam, Sean McNee, Joseph Konstan, and John Riedl. Getting to know you: Learning new user preferences in recommender systems. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 02 2002.

[22] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12, 11 2002.

[23] Mark Claypool, Anuja Gokhale, Tim Mir, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. *Proceedings of Recommender Systems Workshop at*, 08 1999.

[24] B Smyth and P Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2):53 – 59, 2000.

[25] Solomon Demissie Seifu and Shashi Mogalla. A comprehensive literature survey of context-aware recommender systems. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6:40–46, December 2016.

[26] A Kumaravel and P Dutta. Application of pca for context selection for collaborative filtering. *Middle - East Journal of Scientific Research*, 20:88–93, 01 2014.

[27] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Dec 2008.

[28] Tranos Zuva, Sunday O. Ojo, Seleman M. Ngwira, and Keneilwe Zuva. A survey of recommender systems techniques , challenges and evaluation metrics. 2012.

[29] Mae and rmse which metric is better. `https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bd\e13d`. Consultado em Novembro, 2018.

[30] Manolis Vozalis and Konstantinos G. Margaritis. Analysis of recommender systems' algorithms. 09 2003.

[31] What are popular ways of evaluating recommender systems. `https://www.dataminingapps.com/2016/04/what-are-popular-ways-of-evaluating-recommender-systems/`. Consultado em Janeiro, 2019.

[32] Netflix prize. `https://www.netflixprize.com/community/topic_1537.html`. Consultado em Outubro, 2018.

[33] Netflix recommendations: Beyond the 5 stars (part 1). `https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429`. Consultado em Outubro, 2018.

[34] Xavier Amatriain. Mining large streams of user data for personalized recommendations. *SIGKDD Explor. Newsl.*, 14(2):37–48, April 2013.

[35] Amazon's recommendation secret. `http://fortune.com/2012/07/30/amazons-recommendation-secret/`. Consultado em Outubro, 2018.

[36] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan 2003.

[37] Generating recommendations at amazon scale with apache spark and amazon dsstne. `https://aws.amazon.com/pt/blogs/big-data/generating-recommendations-at-amazon-scale-with-apache-spark-and-amaz\on-dsstne/`. Consultado em Janeiro, 2019.

[38] Amazon dsstne. `https://github.com/amzn/amazon-dsstne/blob/master/FAQ.md`. Consultado em Janeiro, 2019.

[39] Discover weekly. `https://support.spotify.com/us/using_spotify/playlists/discover-weekly/`. Consultado em Outubro, 2018.

[40] How does spotify know you so well? `https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19\a41ab76efe`. Consultado em Outubro, 2018.

[41] Yading Song, Simon Dixon, and Marcus Pearce. A survey of music recommendation systems and future perspectives. 2012.

[42] Youtube by the numbers: Stats, demographics and fun facts. `https://www.omnicoreagency.com/youtube-statistics/`. Consultado em Outubro, 2018.

[43] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM.

[44] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016.

[45] Tomislav Duricic, Emanuel Lacic, Dominik Kowald, and Elisabeth Lex. Trust-based collaborative filtering: Tackling the cold start problem using regular equivalence. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 446–450, New York, NY, USA, 2018. ACM.

[46] Rus M. Mesas and Alejandro Bellogín. Evaluating decision-aware recommender systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 74–78, New York, NY, USA, 2017. ACM.

[47] Imen Akermi, Mohand Boughanem, and Rim Faiz. A probabilistic model for intrusive recommendation assessment. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 441–445, New York, NY, USA, 2018. ACM.

[48] Ahmed Mohammed K. Alsalama. A hybrid recommendation system based on association rules. 2015.

[49] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Recommender systems for large-scale e-commerce scalable neighborhood formation using clustering. 50, 01 2002.

[50] Urszula Kuzelewska. Clustering algorithms in hybrid recommender system on movielens data. *Studies in Logic, Grammar and Rhetoric*, 37, 01 2014.

[51] Mulizwa Soft, David Makadani, and Ruzive Mazhandu. Recommender system for telecommunication industries: A case of zambia telecoms. *American Journal of Economics*, 7:271–273, 2017.

[52] Jian Yu, Paolo Falcarin, and Antonio Vetro. A recommender system for telecom users: Experimental evaluation of recommendation algorithms. pages 81–85, 09 2011.

[53] Zui Zhang, Kun Liu, William Wang, Tai Zhang, and Jie Lu. A personalized recommender system for telecom products and services. In *ICAART*, 2011.

[54] Nicolas Hug. Surprise, a Python library for recommender systems. `http://surpriselib.com`, 2017.

[55] Python data analysis library. `https://pandas.pydata.org/`. Consultado em Junho, 2019.

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

[57] Dask Development Team. *Dask: Library for dynamic task scheduling*, 2016.

[58] Dask-ml. `https://ml.dask.org/`. Consultado em Abril, 2019.

[59] Ibm watson machine learning service. `https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-overview.html`. Consultado em Janeiro, 2019.

[60] Mean Average Precision. `http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html`. Consultado em Fevereiro, 2019.

[61] Ben Hamner. Machine learning evaluation metrics. `https://github.com/benhamner/Metrics`, 2015. Consultado em Fevereiro, 2019.

[62] Jupyter. https://jupyter.org/. Consultado em Fevereiro, 2019.

[63] Watson machine- learning client's documentation. https://wml-api-pyclient.mybluemix.net/. Consultado em Fevereiro, 2019.

[64] Build a product recommendation engine with watson machine learning and pixieapps. https://github.com/IBM/product-recommendation-with-watson-ml. Consultado em Janeiro, 2019.

[65] pyspark.ml.clustering module. https://spark.apache.org/docs/latest/api/python/pyspark.ml.html#pyspark.ml.clustering.KMeans. Consultado em Março, 2019.

[66] Tutorial: How to determine the optimal number of clusters for k-means clustering. https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-means-clustering\-14f27070048f. Consultado em Março, 2019.

[67] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, Sep 1998.

[68] Semeh Ben Salem. Clustering categorical data using the k-means algorithm and the attribute's relative frequency. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 11, 06 2017.

[69] Importance of feature scaling. https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importance-py. Consultado em Abril, 2019.

[70] sklearn.decomposition.pca. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. Consultado em Abril, 2019.

[71] dask_ml.cluster.kmeans. https://dask-ml.readthedocs.io/en/latest/modules/generated/dask_ml.cluster.KMeans.html. Consultado em Abril, 2019.

[72] sklearn.cluster.kmeans. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. Consultado em Abril, 2019.

[73] Clustering metrics. https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics. Consultado em Maio, 2019.

[74] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979.

[75] Work breakdown structure (wbs), the basic building block for a project plan. https://project-management.com/wbs-is-the-basic-building-block-for-a-project-plan/. Consultado em Novembro, 2018.

[76] Calcular a duracao de uma atividade pert: Estimativa de tres pontos. https://medium.com/@andreluis.arruda/calcular-a-duracao-de-uma-atividade-pert-estimativa-de-tres-pontos-6cb\35a09a1b3. Consultado em Novembro 2018.

Appendices

Appendix A

CRISP-DM

A metodologia CRISP-DM engloba as fases descritas nas secções seguintes. A Figura 1 ilustra esta metodologia utilizada em projetos com uma forte vertente de *data mining* e *machine learning*.

Análise do negócio

Nesta fase devem ser compreendidos os problemas, definidos os critérios de sucesso e determinados os recursos, requisitos, riscos e benefícios do projeto. Após identificados estes tópicos, será possível formular um plano para chegar à solução final.

Compreensão dos dados

O objetivo desta fase é inspecionar, organizar e descrever todos os dados disponíveis. Para tal, é necessário primeiro fazer a sua recolha, seguida de uma exploração para conhecer o formato dos dados, o seu tamanho e os seus atributos. A finalidade destes passos é avaliar se os dados satisfazem os requisitos. Por vezes é fundamental a presença de um especialista nos dados para que possa fornecer informação de que dados serão mais relevantes.

Deverão ser considerados aspetos como a completude dos dados, a frequência de erros e a existência de dados em falta. Para tal análise, poderá ser necessária a utilização de *software* adicional. O esforço e riscos de fazer esta exploração dos dados também deve ser avaliada.

Tratamento dos dados

Esta fase do processo considera várias etapas no que toca ao tratamento dos dados, nomeadamente:

- Seleção dos dados a utilizar na seguinte fase;
- Limpeza dos dados tal como a remoção de informação incorreta ou a estimação de valores em falta;
- Integração de dados provenientes de múltiplas bases de dados, tabelas ou registos;
- Construção de dados necessários, através da geração de atributos a partir de outros já existentes, ou criação de novos atributos com novos valores.

Modelação

Nesta fase são selecionadas as técnicas de modelação que vão ser utilizadas tendo em conta o tipo de problema que se quer resolver. As técnicas de modelação podem ser de variados tipos, dividindo-se em:

- **Algoritmos de classificação**: previsão de um ou mais valores discretos de uma *range* de valores conhecida;
- **Algoritmos de regressão**: previsão de um ou mais valores contínuos de uma *range* de valores conhecida;
- **Algoritmos de *clustering***: divisão de informação em grupos em que os seus elementos apresentam propriedades similares;

- **Algoritmos de associação**: associação entre vários atributos numa base de dados através da identificação de correlações entre eles.
- **Algoritmos de análise de sequência**: transformação de sequências ou eventos frequentes em dados;

Deverão ser pensados procedimentos ou mecanismos que testem a qualidade e validade do modelo. Por exemplo num modelo de classificação, tipicamente divide-se os dados em dados para treino e dados para teste, utilizando os dados para treino na construção do modelo e os dados de teste para testá-lo.

Avaliação

A fase de avaliação é onde são realizadas as avaliações à *performance* do modelo e dos resultados obtidos, com base nos critérios definidos anteriormente. De modo a observar o seu comportamento aplicado num cenário real, o modelo deve ser testado numa aplicação real. Nesta fase também se devem identificar quais as atividades que deverão ser repetidas e quais as que merecem mais destaque.

As avaliações realizadas irão resultar em decisões positivas ou negativas no que toca a avançar com o modelo para a próxima fase ou a determinar que passos serão necessários para melhorar o modelo.

Deployment

É nesta fase que o modelo é implementado no problema real. Resulta da consolidação das fases anteriores, desejando-se obter um produto robusto, que satisfaça os objetivos de negócio e requisitos definidos. Deve ser desenhado um plano que contenha uma estratégia para executar o *deployment*, bem como os passos e instruções necessárias.

Nesta fase também deve ser realizada uma revisão dos aspetos positivos e negativos que ocorreram durante a realização do projeto, de forma a melhorar o processo para projetos futuros. Visto que este estágio tem o intuito de criar uma prova de conceito, o *deployment* poderá ser uma fase opcional, sendo o objetivo validar o protótipo sobre um *dataset* real fornecido por uma operadora de telecomunicações.

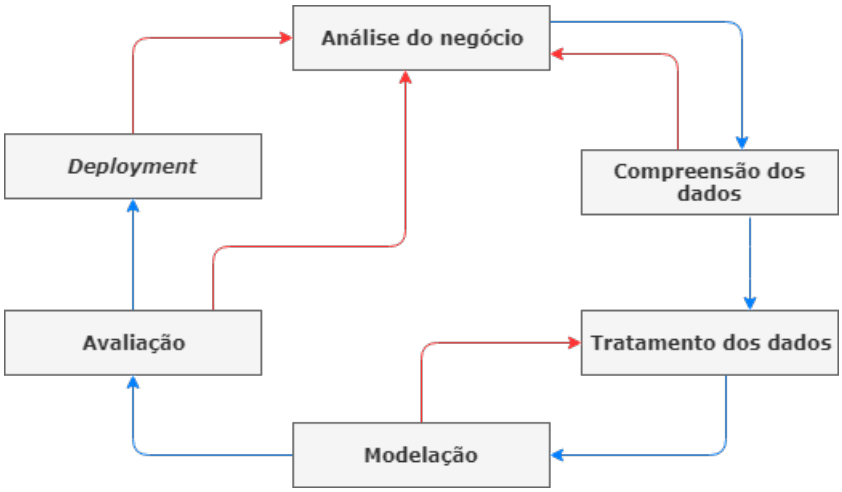


Figura 1: Ilustração das fases do *CRISP-DM*, adaptado de [4].

Como se pode observar na Figura 1, o processo considera as várias iterações entre as fases e o retorno a fases anteriores caso os resultados não sejam satisfatórios, permitindo portanto que sejam feitos melhoramentos onde for necessário.

Appendix B

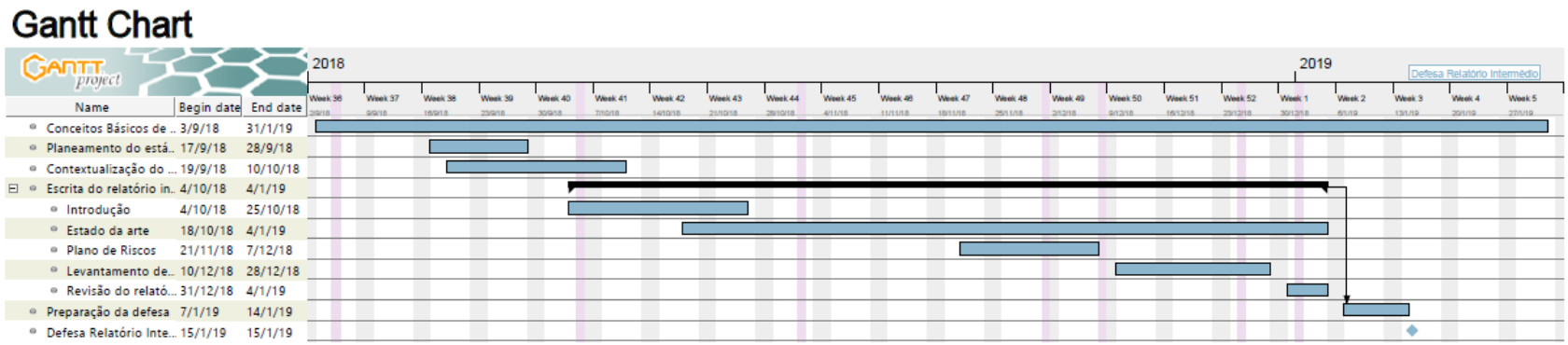


Figura 2: Plano de trabalho do 1º semestre.

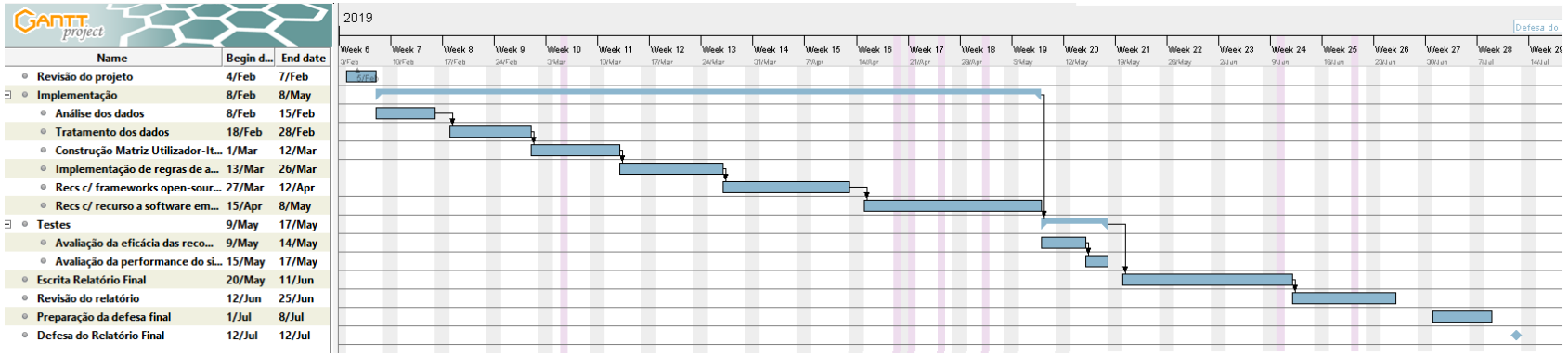


Figura 3: Plano de trabalho do 2º semestre.