Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Luís Sousa Costa

Data Mining*: Modelação de Algoritmos para Automação de *Marketing

Dezembro 2021

Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Luís Sousa Costa

Data Mining: Modelação de Algoritmos para Automação de *Marketing*

Dissertação de Mestrado
Mestrado Integrado em Engenharia Informática

Dissertação orientada por
César Analide Rodrigues

Dezembro 2021

AGRADECIMENTOS

Esta dissertação de mestrado é, no cômputo geral, o culminar do meu percurso académico. Tratou-se de uma componente altamente preponderante na minha vida e com influência direta na pessoa que hoje sou. Porém, existe um conjunto de pessoas que, ininterruptamente, me acompanharam e apoiaram em todas as fases desta longa caminhada. A elas, dedico este capítulo com um enorme orgulho e gratificação.

Aos professores, com os quais tive a fortuna de trabalhar durante estes cinco anos de curso. Em especial, ao professor Analide, por ter aceitado esta proposta e transmitindo confiança em todas as minhas capacidades.

A todos os colaboradores da *BSolus*, empresa que me acolheu para o desenvolvimento desta dissertação e que sempre me disponibilizou apoio e condições de trabalho exemplares de forma a garantir o sucesso deste projeto. Em especial, ao Diogo, Gil e Vítor, elementos da equipa designada para este projeto e com os quais foi um prazer trabalhar.

Todos os meus amigos. Os de infância e os que conheci durante esta caminhada. Ficam na memória momentos inolvidáveis e que com certeza se repetirão no futuro. Dias ou situações, por vezes complicadas, eram amenizadas com a companhia deles. As aventuras, as piadas, o apoio... é algo que me irá marcar para sempre.

À Bárbara. A minha confidente, com quem partilho todas as minhas quedas e sucessos. Que me alegra, incessantemente, pela sua animação contagiante. Que me incentiva a mais e melhor, com carinho e preocupação ímpar, em todos os momentos.

A todos os meus familiares, que sempre me motivaram a atingir os meus objetivos e estarem ao meu lado durante todos os momentos importantes da minha vida.

Por fim, emocionado, agradeço aos meus pais. Vêm, agora, a recompensa palpável de todos os sacrifícios que por mim fizeram. Toda a exigência, apoio e amor tiveram um peso preponderante no meu sucesso académico e é algo que nunca poderei retribuir. A vocês, estarei eternamente grato.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração. Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

ABSTRACT

E-Commerce is a well-known growing area, propelled by the amount of technology surrounding our society that gets better day by day. This growth in E-Commerce solutions is also explainable with the recent worldwide pandemic situation we live in.

This thesis proposal will rely on the development of a recommender engine capable of fetching data from any kind of e-commerce platform, prepare and modify all the data available in order to infer patterns and connections between them and, at last, build a set of Machine Learning algorithms with the purpose of generating accurate and customized recommendations to every customer through marketing actions, such as mailing and messaging.

The cases mentioned above are relevant in a customer re-engagement sequence point of view, that could be accomplished with orders meant to replace consumable items or through a increase of a customer interest to the products.

Also, this thesis is going to be fueled with real sets of data by the fact that it will be applied in a market case study. It is also mentionable the partitioning of the thesis between a theoretical and a practical environemnt.

KEYWORDS E-Commerce, Marketing Automation, Machine Learning, Data Science, Data Mining.

RESUMO

O comércio eletrónico, vulgo *E-Commerce*, encontra-se numa crescente expansão, propulsionada por uma sociedade cada vez mais tecnológica e, mais recentemente, pela pandemia estabelecida a nível mundial. Neste paradigma está implícita uma grande competitividade, pelo que uma estratégia de *marketing* pode ser a chave do sucesso de uma empresa.

Para esta tese, propõe-se o desenvolvimento do estudo comportamental de utilizadores de *websites* de *E-Commerce*, concretamente da Delta Portugal, de maneira a ser possível criar uma segmentação de utilizadores relativamente às suas semelhanças e, a partir delas, gerar uma série de ações de *marketing* automatizadas, tais como o envio de correio eletrónico contendo produtos recomendados para o cliente.

Neste contexto, será desenvolvido um sistema de recomendações assente num processo de *Data Mining*, capaz de abstrair dados de plataformas de comércio eletrónico, manipulá-los e desenvolver algoritmos de *Machine Learning* capazes de gerar recomendações baseadas nos gostos e preferências de cada cliente.

Estes casos mencionados destacam-se numa visão de recompromisso do consumidor para com os produtos, seja pelo facto das unidades de consumo previamente adquiridas terminarem ou pela atividade do mesmo na plataforma ser reduzida, podendo então ser maximizada.

Esta dissertação será, também, desenvolvida a partir de dados reais utilizados no mercado e onde será repartido o ambiente teórico com o prático, associado a empresas líderes de mercado.

PALAVRAS-CHAVE Comércio eletrónico, *Marketing*, Aprendizagem automática, Ciência de dados, *Data Mining*

CONTEÚDO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE LISTAGENS

LISTA DE ACRÓNIMOS

AP *Average Precision*. 51

BD Base de Dados. 30
BPMN *Business Process Model and Notation*. 4, 30

CRISP-DM *Cross-industry Standard Process for Data Mining*. 6, 22, 32, 35, 40, 50, 53, 54, 55, 56, 62

DM *Data Mining*. vi, 4, 5, 6, 7, 11, 21, 22, 23, 24, 32, 35, 62, 63

EDA *Exploratory Data Analysis*. 32, 35, 40

FBC Filtragem Baseada em Conteúdo. 17, 26, 29, 43, 46, 48, 49, 58, 60, 64
FC Filtragem Colaborativa. 17, 18, 26, 29, 43, 44, 45, 46, 47, 48, 51, 57, 58, 59, 60, 64

JSON *JavaScript Object Notation*. 24

MA *Marketing Automation*. iv, vi, 4, 5, 6, 29, 30, 31, 35, 42, 43, 53, 54, 58, 62, 63
MAP *Mean Average Precision*. 29, 51, 56, 57, 58, 59, 60, 63, 65
ML *Machine Learning*. 6, 7, 23, 29, 33, 34, 44, 52, 62, 65

SRH Sistema de recomendações híbrido. 43, 48, 49, 58, 59, 60
SVD *Single Value Decomposition*. 45, 62, 64

TF-IDF *Term Frequency - Inverse Document Frequency*. 46, 49, 64

1

INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

É possível resumir a evolução da espécie humana através de um pequeno conjunto de eventos preponderantes ao desenrolar desse fluxo: desde a descoberta e controlo do fogo até à industrialização global, passando pela invenção da roda e a magnífica descoberta da eletricidade. Todos estes avanços geraram um impacto direto na nossa espécie e em tudo o que nos rodeia.

Nos dias que correm, porém, deparámo-nos com um outro evento potencialmente influenciador do nosso futuro: a revolução dos dados. Passando facilmente despercebida, este novo panorama está diretamente relacionado com a interação feita entre nós e a informação digital à qual temos acesso, bem como o oposto.

Numa sociedade crescentemente influenciada por redes sociais e interações virtuais, os padrões são cada vez mais determinantes no sucesso de um modelo de negócio Heitmann and Hayes (2010). A influência dos referidos padrões é comprovada com a teoria de Surowiecki, apresentada no seu livro: *"The aggregation of information in groups results in decisions that are often better than could have been made by any single member of the group"* Surowiecki (2005).

Partindo deste ponto de vista comprova-se que, de facto, um conjunto de dados por si só é pouco informativo e errático. No entanto, quando trabalhado e transformado segundo várias perceções, poderá tornar-se essencial à análise, descoberta de conhecimento e, devido a esta última, ser possível a tomada de uma potencial decisão. Esta descoberta de conhecimento a partir de dados e respetivos padrões é um dos principais fundamentos da mineração de dados – ***Data Mining*** Fayyad et al. (1996).

Tendo ainda em conta o crescimento exponencial do comércio *online* - alavancado ainda mais pela decorrente pandemia que assola todo o planeta -, inúmeras empresas suportadas por plataformas de comércio virtual começam a questionar-se sobre quais as estratégias capazes de aprimorar a experiência de utilização dos seus consumidores.

Consumir produtos *online* tornou-se vulgar e generalizado devido à sua simplicidade e conveniência. Contudo, a enorme panóplia de ofertas neste universo afeta negativamente a experiência ao utilizador, que é denotado como um dos fatores preponderantes na decisão do consumidor - milhões e milhões de produtos são expostos, sendo frequentemente exaustiva e cansativa a procura pelo produto desejado -. Este processo leva, de forma progressiva, a uma perda de interesse generalizada por parte dos consumidores e, consequentemente, ao

défice de vendas e de lucros às empresas. Assim sendo, com uma crescente competição, a personalização é considerada como o fator chave de sucesso para as plataformas Kocakoç and Erdem (2010).

Com o propósito de munir as plataformas de comércio eletrónico de ferramentas capazes de disponibilizar uma experiência única e pessoal a cada cliente, é fulcral atingir previamente um equilíbrio entre oferta e procura. O segredo do sucesso está, pois, em alcançar esse mesmo equilíbrio Anderson (2006), sendo que o DM poderá oferecer as respostas necessárias para o atingir, utilizando uma estratégia de tomada de decisão baseada em dados Provost and Fawcett (2013), partindo de encomendas anteriores e semelhanças entre clientes.

Aplicando DM, torna-se viável a previsão, com um relativo grau de confiança, de um cliente encomendar um produto em particular. Desta forma, clientes pouparão tempo a procurar produtos numa plataforma, tendo imediatamente recomendações baseadas nos seus gostos pessoais e que, potencialmente, darão azo à exploração de produtos semelhantes aos que pretendia, que de outra forma poderiam passar despercebidos. Tal como uma vulgarmente conhecida citação de Steve Jobs refere, "as pessoas não sabem o que querem até que tu lhes mostres" Reinhardt (1998). Por outro lado, a implementação desta estratégia intensifica a aproximação das empresas a novos conjuntos de clientes e permite a obtenção de informações outrora desconhecidas sobre seus produtos, que poderão ser mais tarde reutilizadas para prever tendências ou mesmo ruturas de stock.

Com toda esta contextualização de mercado, é inegável que a implementação de uma solução deste tipo numa plataforma de comércio eletrónico seja a forma desta se manter em competição no mercado e em conformidade com as contemporâneas exigências dos seus clientes.

1.2 MOTIVAÇÃO

Ao longo do último ano, a empresa *BSolus* tem vindo a desenvolver uma plataforma de automação de *marketing*, suportando-a nas mais recentes linguagens de programação, estratégias de desenvolvimento e *pipelines* de gestão. Esta plataforma, ainda não lançada num ambiente de produção, designou-se por ferramenta de MA e será mencionada múltiplas vezes ao longo da redação desta dissertação.

Esta ferramenta, através de fluxos montados no formato BPMN, é capaz de criar, gerir e executar ações de *marketing* para conjuntos de utilizadores armazenados das plataformas aderentes, que previamente são inseridos nesta ferramenta por um administrador de plataforma.

De forma similar a muitas outras peças de *software* pensadas na automação de *marketing* em plataformas de comércio eletrónico, esta ferramenta tenta auxiliar grandes plataformas que, sem esta, teriam enormes dificuldades em gerir o aumento de produtos nas suas lojas *online*, dado não terem mecanismos de pesquisa ou sistemas de recomendações capazes de associar a procura dos consumidores à crescente oferta.

Aliás, observando a correlação entre número de encomendas e popularidade de produtos em lojas *online* geridas por esta empresa, verifica-se que aproximadamente 80 por cento de todas as compras são provenientes de 35 por cento de todos os produtos disponíveis, ou seja, muito próximo da regra 80/20 do Princípio de Pareto, mais à frente mencionado nesta dissertação e que se pode verificar na figura 2.

Sem surpresa, a compra de produtos *online* deflagrou em tempos recentes, devido à sua conveniência e, acima de tudo, à pandemia que surgiu em 2020, que influencia inúmeras pessoas a trocar a aquisição de

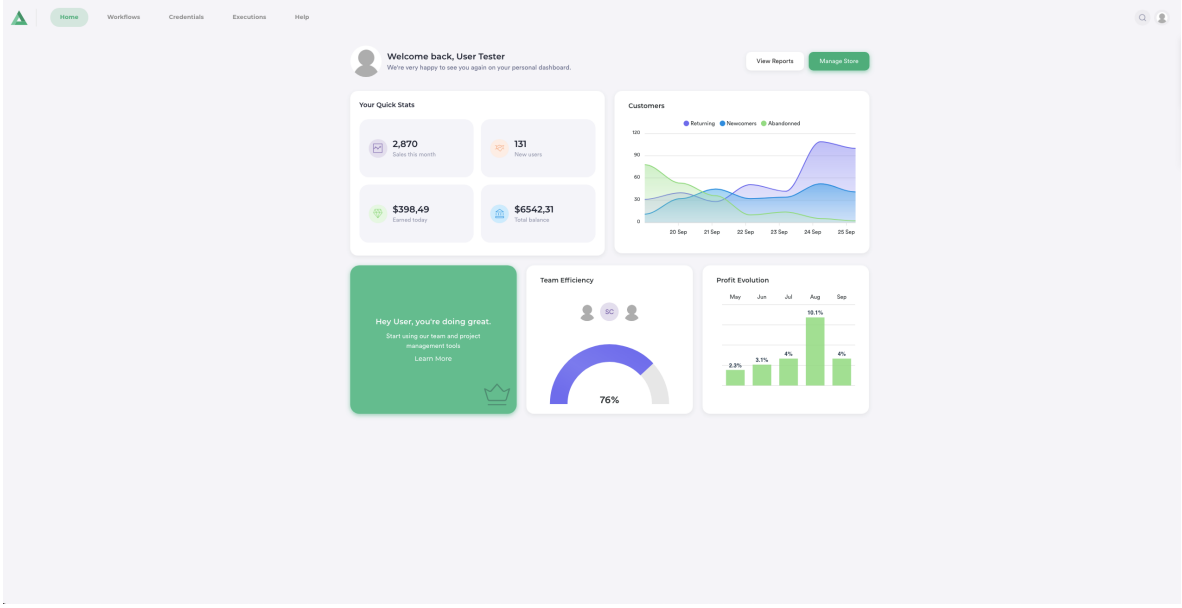

Figura 1: Página inicial da ferramenta de MA

produtos em lojas físicas pela compra dos mesmos produtos pela *web*, garantindo-lhes assim segurança e bem-estar.

Sendo uma das empresas que tenta atingir o topo na gestão de comércio eletrónico em Portugal, a *BSolus* encontra-se perante a necessidade de desenvolver novas estratégias e produtos capazes de aumentar os lucros nas lojas dos seus clientes. Como tal, esta ferramenta de MA foi idealizada para garantir ações de *marketing* personalizadas aos consumidores das lojas *online* aderentes. Este desenvolvimento poderá ser analisado pormenorizadamente na dissertação do Diogo Gonçalves, um dos elementos da equipa responsável pela ferramenta e cujo trabalho incidiu sobre a arquitetura desta ferramenta Gonçalves (2021).

Paralelamente, foi definido o desenvolvimento de um sistema de recomendações inteiramente modular para esta ferramenta, capaz de gerar recomendações de produtos personalizadas a grupos de consumidores e que, através das ações de *marketing* construídas, cheguem a estes via correio eletrónico ou SMS.

1.3 ESTRUTURA DO DOCUMENTO

Esta dissertação de mestrado inicia-se com uma introdução ao DM, bem como uma pequena contextualização do comércio nos dias de hoje e as motivações que levaram ao desenvolvimento de um sistema de recomendações para uma plataforma de comércio eletrónico. Consequentemente, no capítulo seguinte, são enumeradas as razões que levaram ao desenvolvimento deste sistema de recomendações, bem como todos os objetivos que com ele se pretendem atingir.

Ao longo do terceiro capítulo será investigada e analisada a evolução dos sistemas de recomendação até aos dias de hoje, analisando casos particulares de sucesso ao longo dos anos. Nestes últimos, serão também

mencionadas as plataformas que mais contribuíram para a evolução das técnicas de recomendação e ainda os obstáculos mais significativos que habitualmente se enfrentam ao desenvolver sistemas deste tipo.

De seguida, o quarto capítulo albergará uma extensa investigação de processos e metodologias de desenvolvimento de *software* a adotar bem como, neste caso específico, de DM. Neste, poderão ser consultadas todas abordagens e soluções adotadas no armazenamento, parametrização, documentação e registo de *logs* deste sistema, bem como uma análise detalhada à arquitetura do sistema montado e respetivos componentes. Este capítulo encerra, ainda, com a exposição da ferramenta de MA na qual o sistema desenvolvido é aplicado.

No quinto capítulo, efetua-se uma análise detalhada de toda a implementação feita neste projeto, correspondendo às fases do processo CRISP-DM. Aqui, inicialmente, é feita uma listagem pormenorizada dos dados existentes que entram no sistema, garantindo assim a sua explicabilidade e compreensão. Este capítulo irá expor, também, o tratamento feito aos dados existentes, de modo que a partir destes seja possível abstrair padrões e associações, bem como a modelação de dados criada a partir de inúmeros algoritmos de ML e que garantem a geração de recomendações. Além disto, são investigadas e aplicadas métricas de avaliação e técnicas de validação para as recomendações obtidas.

No capítulo seguinte, serão registados, analisados e validados todos os resultados obtidos a partir de execuções do sistema implementado. Aqui, inclui-se uma revisão à calendarização proposta e uma análise ao desempenho do sistema. Esta última é feita a partir de diversos fatores: primeiro, é feito um estudo minucioso sobre os resultados obtidos na avaliação do sistema e o impacto que diversos parâmetros possam ter sobre eles; depois, investiga-se o esforço computacional, escalabilidade e capacidade de armazenamento do sistema; por fim, os resultados obtidos são comparados com previsões teóricas, de forma que seja possível responder aos objetivos estabelecidos no início da dissertação.

Para finalizar, esta dissertação termina com a apresentação de todas as conclusões ao longo do sétimo e último capítulo.

2

REQUISITOS

Como foi possível abstrair no capítulo anterior, sistemas de recomendações e outras estratégias de decisão derivadas de dados aplicadas no comércio eletrónico têm o potencial de aumentar o compromisso dos consumidores nas plataformas onde estas são aplicadas, devido a uma experiência orientada às preferências de cada cliente, a quem são recomendados produtos que se assemelham aos seus gostos, sendo assim possível um aumento de lucros.

Assim, e tendo em consideração o enquadramento já exposto, esta dissertação pretende desenvolver um sistema de recomendações fidedigno, capaz de manipular e treinar dados provenientes de plataformas de comércio eletrónico, regidos por um contrato de dados pré-estabelecido e que estas devem respeitar. Este sistema será, também, capaz de modelar os dados através de algoritmos de forma a gerar recomendações sustentadas em padrões e associações encontradas entre produtos e clientes. Este sistema deverá, por outro lado, ser inteiramente modular, isto é, ser adaptável a qualquer tipo de modelo de negócio ou plataforma utilizada na sua execução.

Pretende-se, ainda, que este sistema albergue tecnologias, processos e funcionalidades de topo e que se destaquem, com o propósito de se destacar dos demais sistemas de recomendações encontrados em variadíssimas plataformas *web* nos dias que correm.

Seguem-se, então, os objetivos que se pretendem atingir com o desenvolvimento deste projeto. Aqui, percebeu-se que era possível definir, a partir dos mesmos, um conjunto finito de requisitos para o sistema. Esta listagem define todas as funcionalidades que deverão munir o projeto, tendo em conta as suas necessidades ao nível da arquitetura e do desenvolvimento:

1. **Desenvolver um sistema de recomendações utilizando metodologias de DM contextualizadas ao problema**
2. **O sistema deve disponibilizar múltiplos tipos de recomendações para a plataforma que o utiliza**
3. **O sistema de recomendações deverá ser inteiramente modular, de maneira a albergar diferentes plataformas**
4. **O sistema de recomendações deverá albergar as mais aceites e reconhecidas técnicas de ML do mercado orientadas ao comércio eletrónico**
5. **Os resultados obtidos pelo sistema devem ser testados através de métricas e processos de validação orientados a este tipo de problemas**

6. **Desenvolver um sistema de recomendações configurável, permitindo aos seus administradores ajustar parâmetros relevantes do problema**

7. **Adotar uma abordagem *open-source* ao nível das tecnologias utilizadas e, simultaneamente, desenvolver um sistema o mais baixo esforço computacional possível**

Depois de definidos todos os objetivos e requisitos funcionais para o sistema a implementar, o foco orienta-se, agora, para um estudo extensivo sobre o contexto atual dos sistemas de recomendações, bem como os mais relevantes casos de sucesso. Tudo isto será exposto em pormenor ao longo do capítulo seguinte, correspondente ao estado da arte.

3

ESTADO DA ARTE

3.1 O COMÉRCIO

Na moda. É desta forma que a nossa cultura se define no último século. Acompanhar tendências tornou-se um hábito corriqueiro, desde a música que ouvimos a caminho do trabalho, as séries que vemos numa sexta-feira à noite ou no filme que vai estrear e de que todos falam. É este o quotidiano que alimenta milhões e milhões de euros em indústrias governadas por sucessos contemporâneos.

Esta mentalidade alastrou-se, de forma expectável, para o comércio. Toda a sociedade é influenciada pelas mais recentes tendências e marcas, seja porque se tornaram um sucesso generalizado ou simplesmente porque é algo utilizado pelo seu ídolo no mundo da música. De maneira a aumentar lucros, plataformas e comerciantes focam-se cada vez mais nos seus produtos mais populares, isto é, cujas vendas são maiores e aos quais se torna mais vantajoso publicitar com o propósito de obter retorno.

3.1.1 *Princípio de Pareto*

Em 1897, o economista e sociologista Vilfredo Pareto detetou um padrão particularmente interessante aquando uma pesquisa e análise à distribuição de riqueza em Inglaterra. O padrão consistia numa consistente relação matemática entre a quantidade de pessoas e riqueza associada a esse mesmo grupo. Pareto afirmou, então, que aproximadamente vinte por cento da população usufruía de oitenta por cento da riqueza nacional Koch (1997).

Mais tarde, em 1951, o engenheiro romeno Joseph Juran, no seu reconhecido livro "Quality Control Handbook", citou que *"the economist Pareto, found that wealth was non-uniformly distributed in the same way (...) as the distribution of crime amongst criminals, the distribution of accidents among hazardous processes, etc"* Juran (1988), generalizando, então, o princípio de Pareto para a criminalidade e ocorrência de acidentes em processos industriais e laboratoriais de grande risco.

Consequentemente, este princípio é aplicável no comércio. Aliás, é seguro afirmar que aproximadamente oitenta por cento de todos os lucros advêm de vinte por cento dos produtos de uma empresa ou plataforma, possivelmente até menos Anderson (2006).

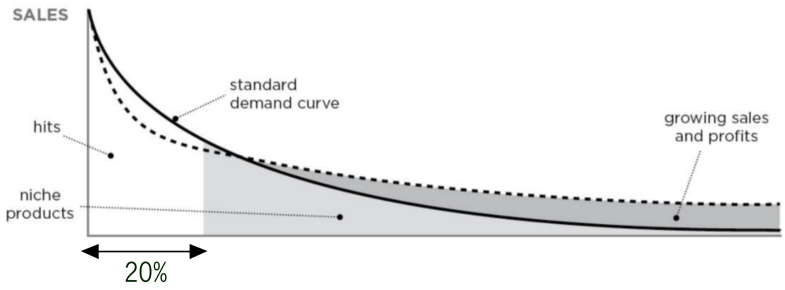

Figura 2: O princípio de Pareto no comércio e a *long tail*

Apesar deste dado ser chocante para alguns, as gerências de negócios relacionados com o comércio terão de ter em conta que, para triunfar, deverão planear e executar uma estratégia de negócio focalizada nos seus *bestsellers* para, assim, maximizar os seus rendimentos. Esta premissa é parte de uma estratégia denominada de *Customer Relationship Management*, ou, em português, Gestão de Relacionamento com o Cliente. É uma estratégia que inclui metodologias capazes de identificar, gerir e expandir relações com clientes baseando-se nas suas necessidades. Satisfazendo as mesmas e construindo uma relação de confiança entre cliente e empresa resultará sempre num aumento de rentabilidade Chen and Popovich (2003).

Devido aos custos associados aos espaços nas prateleiras das lojas físicas, as empresas convencionalmente preenchiam as mesmas com os seus melhores produtos ou, melhor dizendo, os produtos que mais vendem. Vendo bem, um conjunto de camisas que vende uma unidade por mês ocupa o mesmo espaço em loja que um outro conjunto de camisas que venda cem unidades por mês, levando assim as empresas a reduzir a sua oferta para apenas os seus *bestsellers*, minimizando assim o risco associado ao aluguer do espaço de prateleira. No entanto, com mudança do milénio, veio a explosão digital e, com ela, o início da era do comércio *online*.

3.1.2 *Mudança de paradigma*

No seu reconhecido livro intitulado "The Long Tail" Anderson (2006), Chris Anderson observou as relações entre vendas e popularidade em várias plataformas e áreas de negócio, sistematizando as suas conclusões numa descoberta inovadora. No comércio digital, o princípio de Pareto, detalhado acima, não é aplicável. Partindo do pressuposto que num *website* não existe o conceito de espaço e aluguer de prateleira, a adição de mais produtos da empresa não tem um impacto financeiro notório, pelo que as plataformas de comércio eletrónico vieram expandir de forma drástica o leque de produtos que a empresa pode comercializar.

Não obstante o espetro de produtos disponibilizados ao cliente ser agora muito mais amplo, continuaria a ser aceitável afirmar que o princípio de Pareto ainda deveria ser aplicável ou, pelo menos, ir relativamente de encontro à realidade. Mas a resposta é não. É inquestionável que os produtos mais vendidos gerem os maiores ganhos às respetivas empresas, porém Chris Anderson identificou um novo mercado. O mercado de segmentos e pequenos grupos é responsável por uma boa parcela dos lucros fora do horizonte dos *hits*. Num

caso particular, Anderson observou que, num fabricante de *jukeboxes* digitais, cerca de 98 por cento de todas as músicas foram tocadas pelo menos uma vez por trimestre. Isto representa o princípio de *long tail* (cauda longa).

Consequentemente, isto criou um mercado de produtos de nicho, cujos produtos eram vendidos por quantias mais módicas. Mas, partindo do pressuposto que o número de produtos que se podem vender é virtualmente ilimitado, no cômputo geral formou-se um grande volume de negócio. Não obstante, o dilema observado nesta estratégia reflete-se na sua própria denominação: *underground* (i.e. secreto). Grande parte dos consumidores conhece e valoriza os *bestsellers* e, apesar de apreciar alguns produtos de um segmento mais específico, estão muitas vezes desconhecedores dos mesmos.

3.1.3 *Nivelar oferta e procura*

A disponibilização de uma oferta quase ilimitada de produtos gera uma problemática. A procura deve acompanhar a subida da oferta de maneira a tornar viável toda esta estratégia de negócio, caso contrário, a cauda vai paralisar. Anderson, novamente, aborda esta situação no seu livro, dizendo que *"simply offering more variety, however, does not shift demand by itself. Consumers must be given ways to find niches that suit their particular needs and interests"* Anderson (2006), isto é, devem ser identificadas e estruturadas formas de dar a conhecer estes produtos de segmentos específicos a consumidores cujas necessidades e interesses vão de encontro ao que o produto em si pode solucionar.

Assim sendo, como será possível apresentar aos clientes estes produtos com segmentos bem particulares? A resposta assenta no desenvolvimento de sistemas de recomendação assentes em DM. Através de análises automatizadas de encomendas anteriores de clientes, bem como avaliações dos mesmos a produtos e, por fim, semelhanças entre produtos e inclusive entre clientes, várias métricas podem ser desenvolvidas para avaliar o grau de confiança de um cliente comprar um determinado produto e, com isso, explorar formas de lho apresentar.

Sem esta metodologia de recomendação, produtos com segmentos específicos continuariam anónimos para muitos, levando a uma insatisfação generalizada de utilizadores de comércio eletrónico, argumentando a (aparente) falta de diversidade de produtos providenciados. Promovendo esta visibilidade a produtos outrora ignorados, a *long tail* é reforçada quando o negócio progride de sucessos para segmentos. Esta alteração pode ser observada com maior pormenor na figura 2.

3.2 TÉCNICAS

Como mencionado previamente, as implementações de sistemas de recomendação não só promovem uma experiência personalizada ao utilizador como também exploram um mercado *underground* onde estão presentes produtos com menos visibilidade, potenciando receitas em todos os segmentos das empresas.

Cada sistema de recomendação deve ser desenvolvido em conformidade com a plataforma e área de negócio aplicável e, como tal, uma investigação pormenorizada sobre as abordagens mais utilizadas na contemporanei-

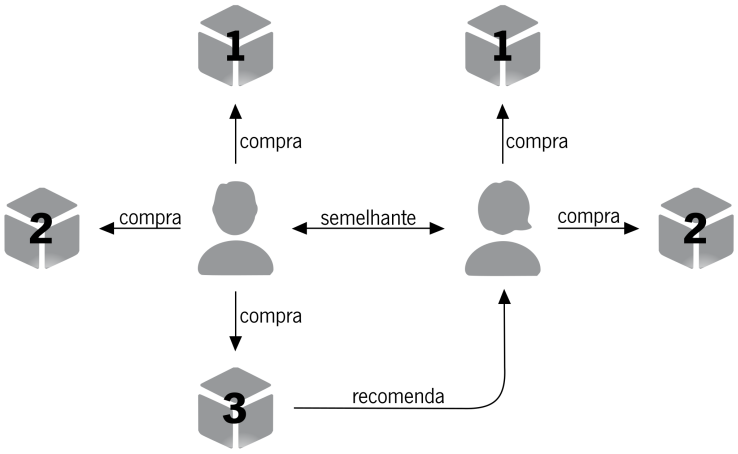

Figura 3: Esquema da filtragem colaborativa

dade é imperativa para uma tomada de decisão. Esta secção descreve, assim, as abordagens de recomendação mais utilizadas na atualidade e que serão viáveis a uma aplicação neste projeto.

3.2.1 *Filtragem colaborativa*

Filtragem colaborativa tem por base informações, ações e comportamentos do utilizador, fazendo as suas previsões e recomendações a partir destes dados. Assume-se, nesta técnica, que se determinados utilizadores têm a mesma opinião sobre um produto, muito provavelmente o mesmo comportamento se verificará para um outro produto Ekstrand et al. (2010).

A primeira abordagem à filtragem colaborativa foi, previsivelmente, orientada ao utilizador. Consistia na procura de utilizadores cujo comportamento em avaliações de produtos se assemelhasse ao do utilizador a quem se destinavam as recomendações e, de seguida, recolhidas e treinadas as avaliações, eram disponibilizadas as recomendações ao utilizador. Esta proposta, apesar de eficaz, sofre de problemas de escalabilidade.

Uma outra abordagem, ao invés de orientada ao utilizador, é orientada ao item. Contrariamente a uma orientação ao utilizador, esta abordagem procura similaridades entre avaliações e informações dos produtos Ekstrand et al. (2010). Tendo em conta que, num sistema com um grande fluxo de compras, alterar uma avaliação ou até acrescentar um novo utilizador à lista de clientes que adquiriram um determinado produto não vai alterar significativamente a similaridade entre dois itens, então comprova-se que se trata de uma abordagem altamente escalável, comparativamente a uma filtragem colaborativa orientada exclusivamente ao utilizador.

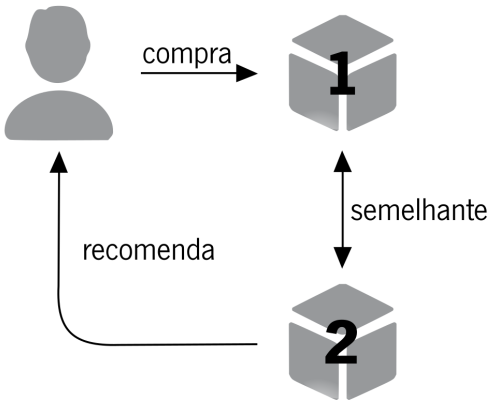

Figura 4: Esquema da filtragem baseada no conteúdo

3.2.2 *Filtragem baseada no conteúdo*

Ao passo que a filtragem colaborativa se rege por interações e decisões do utilizador, a filtragem baseada no conteúdo é influenciada pela similaridade entre produtos. Esta abordagem usa como premissa o conceito de "Mostra-me mais daquilo que eu gosto" Sharma and Singh (2016). Aqui, as associações e padrões entre produtos são identificados através das semelhanças entre as suas características relacionadas.

3.2.3 *Sistema híbrido*

Tal como o nome sugere, esta abordagem implica a junção de duas ou mais abordagens, como ilustra a figura 5. A ideia por detrás desta técnica e que a suporta é relativamente trivial e consiste em que uma filtragem colaborativa, baseada no conteúdo ou qualquer outra aqui não referenciada tem défices e, agregando duas ou mais num único sistema, os mesmos podem ser colmatados por abordagens opostas, gerando dessa forma uma previsão mais exata e confiável. Existe, na globalmente conhecida plataforma de *streaming* ***Netflix***, um excelente exemplo de um sistema de recomendação híbrido, onde se copulam filtragens colaborativas e filtragens baseadas em conteúdo com o propósito de recomendar os próximos filmes e séries a serem visualizados pelo utilizador Sharma and Singh (2016).

3.3 ANÁLISE DE MERCADO

No já referido livro "The Long Tail" Anderson (2006), o autor menciona também um evento um tanto sui generis. Em 1988, Joe Simpson, alpinista britânico, escreveu um livro intitulado "Touching the Void", no qual narra as suas experiências, algumas delas colocando até em risco a própria vida. Apesar de algumas boas avaliações, a

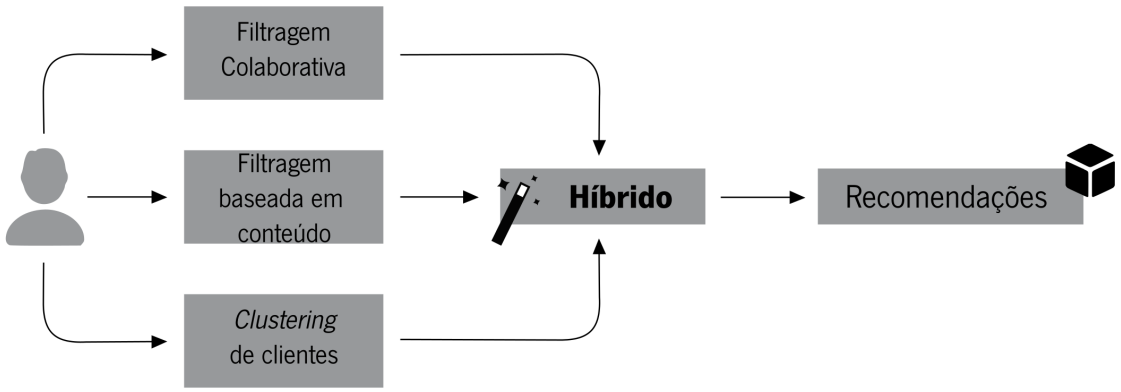

Figura 5: Esquema de um sistema híbrido de recomendações

obra foi rapidamente esquecida. Mais tarde, cerca de uma década, Jon Krakauer escreveu um livro sobre uma tragédia relacionada com o alpinismo, "Into thin Air", tornando-se instantaneamente num sucesso e, de forma similar, "Touching the Void" também disparou nas vendas. O livro outrora olvidado passou rapidamente para as prateleiras da frente nas livrarias, ficando catorze semanas na lista de *bestsellers* do New York Times. Mais à frente ainda, em 2003, foi lançado um documentário baseado no livro, sendo apelidado pelo *The Guardian* como "o documentário com mais sucesso na história do cinema britânico" Thompson (2009).

Após todas estas ocorrências, as vendas de "Touching the Void" superaram as de "Into thin Air" em mais do dobro das unidades. Como explicar este fenómeno? Recomendações. Quando "Into thin Air" chegou às lojas *online*, foram detetadas similaridades entre o mesmo e "Touching the Void" e este, estando já quase descontinuado, explodiu no número de vendas. É esta a grande potencialidade dos sistemas de recomendação: encontrar padrões e associações entre produtos e clientes, e levar produtos obsoletos – na *long tail* – para a ribalta.

3.3.1 *Casos de sucesso*

As recomendações que enviaram "Touching the Void" para o topo quase dez anos após o seu lançamento são o reflexo de anos e anos de trabalho desenvolvido por uma empresa que, após revolucionar a presença no mercado deste livro, o repetiram mais tarde para toda uma indústria de comércio eletrónico: a *Amazon*.

Esta empresa, com a colaboração de outras bem cotadas no mercado, foi pioneira na investigação e implementação de sistemas de recomendação. Isso pode ser comprovado com o facto da primeira patente lançada pela *Amazon* em relação a esta tecnologia está datada em 1998 Linden et al. (1998), isto é, há mais de duas décadas. Este investimento da empresa na busca de um modelo de filtragem colaborativa baseada em itens foi o primeiro passo na adoção da estratégia *long tail*, aproximando milhões de produtos aos seus utilizadores.

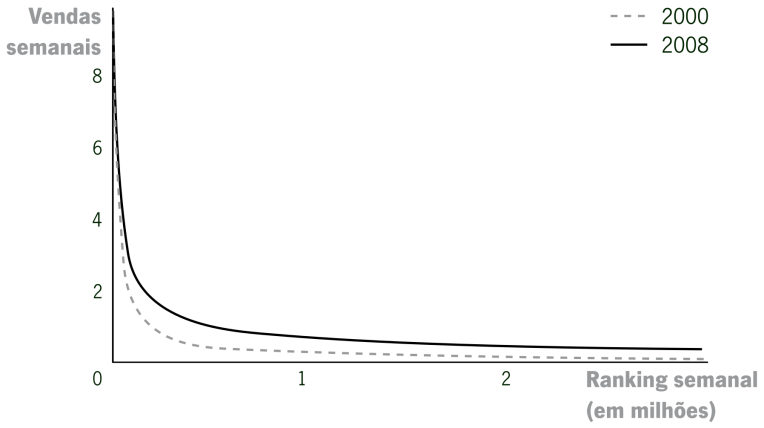

Figura 6: A *long tail* da *Amazon* em 2000 e 2008

Em 2003, o algoritmo foi lançado a uma escala global através da plataforma da *Amazon*. As recomendações foram tão amplamente divulgadas que por vezes existia a sensação de que seguiam os olhos dos utilizadores. Sugestões de produtos baseados em compras anteriores eram encontradas na página inicial e nos resultados de procuras, as páginas de carrinhos de compras expunham produtos similares aos que o utilizador lá tinha colocado, com o objetivo de fazer o utilizador agregar dois ou mais produtos perto do momento da aquisição. Na prática, todas as páginas do *website* incluíam recomendações de algum tipo. E resultou. Em 2015, um relatório de investigação estimou que cerca de trinta por cento das visualizações de produtos na plataforma da *Amazon* foram originados por recomendações Sharma et al. (2015). Grande parte destas visualizações nunca ocorreriam sem o suporte correto das recomendações feitas pela *Amazon*, e que comprovam o quão influentes podem ser estes sistemas para o crescimento e sucesso de uma plataforma.

Por outro lado, num artigo científico Brynjolfsson et al. (2010), investigadores calcularam e compararam as *long tails* da *Amazon* nos anos de 2000 e 2008. Neste período, a cauda cresceu significativamente e os ganhos gerais com livros especializados cresceram em cinco vezes. A equipa concluiu ainda que, à data, os respetivos livros especializados foram responsáveis por quase trinta e sete por cento da venda total de livros por parte da empresa.

Com isto, os resultados espelharam toda a teoria de Anderson. Refira-se que a magnitude dos produtos de sucesso nunca deve ser subvalorizada. Os produtos populares são, numa generalidade, as maiores fontes de receita das empresas e é nesses itens onde as mesmas mais investem. Todavia, os produtos mais segmentados devem ser alvo de análises e esforços, utilizando sistemas de recomendação para colocar a procura ao mesmo nível da oferta.

Atualmente, mais empresas seguiram os passos da *Amazon*, fazendo com que recomendações já estejam integradas com o nosso quotidiano de interações digitais, estando presentes em reconhecidas páginas como *Facebook*, *Netflix*, *Spotify* ou *YouTube*. De muitos casos de uso, os que potencialmente mais se assemelham à proposta desta dissertação estão presentes em lojas de vestuário *online*, que não dispensam a sua *newsletter*

digital, oferecendo aos seus utilizadores informações sobre novos produtos, cupões de desconto ou até ofertas em datas marcantes.

Tal como exposto e sugerido neste capítulo, será desenvolvido um sistema de recomendação híbrido com o propósito de disponibilizar e apresentar produtos de várias empresas via correio eletrónico aos seus clientes subscritores da *newsletter*, tendo em conta os seus interesses, pesquisas e semelhanças com outros utilizadores.

Desde 1998 e devido à patente da *Amazon* para sistemas de recomendação, o avanço em plataformas e empresas dispostas a vender produtos *online* foi colossal a nível global. Em Portugal, acompanhando a tendência, empresas começaram a adotar as mesmas tecnologias. Apesar de tudo, a sua implementação e lançamento ocorreu muito mais tarde comparativamente a outras plataformas de comércio eletrónico internacionais.

De acordo com a política de privacidade da *Wook*, um retalhista de livros *online* português, os dados de cada utilizador são agregados com o intuito de serem trabalhados de modo a apresentar conteúdo adaptado a cada tipo de utilizador Porto Editora (2018). Avançando para outro mercado *online* português, na *Dott*, uma sucursal dos CTT e da Sonae criada com o propósito de se inserir no mercado como "A *Amazon* Portuguesa", as recomendações são trivialmente detetadas.

Com um foco apenas orientado ao mercado onde a *Beevo* se insere, identifica-se a *Redicom* como um competidor direto. Esta plataforma oferece uma solução similar à da *Beevo*, porém com a vantagem da sua plataforma já disponibilizar um sistema de recomendação. Segundo a sua página *web*, os sistemas da *Redicom* processam recomendações para produtos de interesse. O mecanismo considera cada cliente, seleciona as suas preferências a partir de comportamentos anteriores e devolve produtos relacionados Redicom (2019). Trivialmente se percebe que este processo é o principal foco deste projeto, pelo que se torna essencial que a solução aqui implementada tenha uma performance e escalabilidade superior à deste competidor.

3.3.2 *Atualização de recomendações*

Em vários sistemas de recomendação é essencial existir uma constante atualização de recomendações, devido ao grande fluxo de utilizadores habitualmente a aceder a serviços, como, por exemplo, se identifica na *Netflix*. Como o consumo de produtos é constante, os gostos de cada utilizador bem como as avaliações de cada produto estão numa modificação constante. Observando o caso do *YouTube*, a procura de novas recomendações é ainda mais imperativa, tendo em conta o número de horas de vídeo carregados para a plataforma a cada segundo Convington et al. (2016).

Este foi, previsivelmente, um dos tópicos mais abordados aquando da esquematização da solução para este sistema de recomendação. Tendo em conta que cada solução deve ir ao encontro das necessidades de cada plataforma e clientes, temos na *Beevo* uma situação relativamente distinta às expostas acima. Sendo esta uma plataforma de comércio eletrónico, as suas páginas são visitadas ocasionalmente quando os consumidores tencionam adquirir um produto específico ou apenas procuram algo de novo. Nestas circunstâncias, e sabendo que o sistema criará recomendações a serem enviadas diretamente ao cliente, seja por *e-mail* ou SMS, o período de inatividade supera em larga escala o tempo de execução do sistema.

Desta forma, todas as recomendações serão armazenadas numa data e hora específica, sendo que a frequência destas atualizações dependerá única e exclusivamente na influência da recomendação, ou seja, recomendações para o utilizador sofrerão mais alterações que recomendações de produtos similares, dado que os gostos individuais de cada consumidor estão em constante mutação, ao contrário das características dos produtos, que se preservam. Tudo isto será descrito, pormenorizadamente, no capítulo seguinte.

3.4 PROBLEMAS COMUNS E DESAFIOS

Mesmo após décadas de desenvolvimento e aprimoramento, os sistemas de recomendações ainda possuem problemas intrínsecos à sua estrutura. Não sendo todos eles possíveis de erradicar, existe a possibilidade de mitigar muitas destas contrariedades através de técnicas já definidas.

3.4.1 *Cold start*

O fenómeno de *cold start* é, muito provavelmente, o problema mais frequente quando se lida com sistemas de recomendação. Este ocorre quando o sistema não tem qualquer possibilidade de abstrair padrões sobre clientes ou produtos, devido ao facto destes ainda não terem facultado à plataforma em que se inserem informação suficiente para tal. Este fenómeno pode ser ramificado em três problemas específicos: problema do novo cliente, problema do novo produto e problema do novo sistema Sharma and Gera (2013).

Para o problema do novo cliente, a partir do momento que os sistemas de recomendações utilizam perfis de utilizadores e atributos de produtos para gerar previsões, novos utilizadores ou utilizadores já existentes na plataforma com pouca informação introduzida não serão alvo de nenhuma recomendação, ou serão muito imprecisas caso sejam feitas. Uma forma de superar esta complicação consiste no incitamento ao utilizador a introduzir informações relevantes na plataforma, ou então importar dados de outras plataformas *web*, maioritariamente de redes sociais Boehmer et al. (2015).

Por outro lado, o problema de novos itens ocorre quando um novo produto é adicionado na plataforma. Nesta fase, existe pouca informação sobre este e, acima de tudo, o produto nunca foi adquirido por qualquer cliente. Assim, a falta de encomendas afeta diretamente técnicas de FC, visto que o produto não tem qualquer ligação estabelecida com algum cliente. Então, para ultrapassar esta contrariedade, a adição de produtos à plataforma deve ser feita munindo-o da máxima informação disponível, de maneira a dar visibilidade ao mesmo em técnicas de FBC. Desta forma, se existir uma combinação de FC e FBC num hipotético modelo híbrido, os produtos mais recentes manterão a sua presença em listas de produtos recomendados.

Por fim, poderá ainda ocorrer o problema de novo sistema, quando os dois problemas descritos nos parágrafos acima acontecem em simultâneo. Habitualmente, esta complicação está relacionada com a criação de novas plataformas de comércio eletrónico, dado que ainda não existem utilizadores registados e, de forma óbvia, nenhum produto foi ainda adquirido. Esta situação poderá ser combatida através dos procedimentos mencionados acima e, como trivialmente se percebe, este problema não ocorre no sistema que se pretende implementar, pelo que uma investigação mais extensiva torna-se dispensável.

3.4.2 *Esparsidade*

Devido ao crescente aumento da oferta em plataformas de comércio eletrónico, torna-se inviável encontrar clientes que tenham adquirido uma grande percentagem dos produtos disponibilizados. Assim, quando são geradas recomendações via modelos de FC, representar informações de clientes e produtos numa matriz pode levar a um elevado número de células vazias na mesma, correspondentes aos produtos não adquiridos pelo cliente, dando origem a matrizes esparsas Sharma and Gera (2013).

Este fenómeno pode influenciar negativamente a qualidade das recomendações, pelo facto do sistema gerar as mesmas a partir de um número extremamente limitado de interações entre clientes e produtos. Duas técnicas capazes de superar esta complicação consistem na redução da dimensão da matriz, feita através da projeção de clientes e produtos num espaço latente reduzido que alberga os atributos mais influentes de cada um, ou então modelos baseados em grafos, criados através associações transitivas Desrosiers and Karypis (2011).

3.4.3 *Escalabilidade*

A escalabilidade é a propriedade do sistema que garante a capacidade do mesmo em lidar com enormes e crescentes quantidades de dados de forma satisfatória. Com a já referida explosão e crescimento de produtos nas plataformas de comércio eletrónico, os sistemas de recomendação encontram-se perante a difícil tarefa de gerir estes grandes volumes de dados. Note-se, ainda, que alguns algoritmos nas técnicas já referidas efetuam cálculos que crescem exponencialmente com o número de clientes e produtos. Os algoritmos de FC são alguns desses casos, e o grande esforço computacional nestas situações torna-se inviável para as empresas responsáveis pela sua gestão e, simultaneamente, levam à obtenção de resultados cada vez mais imprecisos Sharma and Gera (2013).

Uma estratégia *divide and conquer*, isto é, orientada à distribuição de esforços, poderá mitigar estes problemas. No caso em que nos encontramos, consiste na segmentação de clientes segundo as suas características, ao invés de tratar todos os clientes existentes de uma só vez. Mais ainda, a redução da dimensão de uma matriz esparsa como foi mencionado no subcapítulo anterior poderá também levar a um melhor desempenho computacional de algoritmos de FC, ainda que também leve à perda de acerto na geração de recomendações Sarwar et al. (2001).

3.4.4 *Robustez*

Este problema é notório e relevante em plataformas que albergam sistemas de *rating* de produtos e, através destes, os clientes avaliam os produtos de forma desonesta, habitualmente devido a influências momentâneas. Isto leva a informação incorreta que, mais tarde, é providenciada aos sistemas de recomendação que, depois, irão gerar resultados enviesados Konstan and Riedl (2012).

De maneira a ser possível detetar avaliações fraudulentas, foram já desenvolvidos alguns algoritmos capazes de limitar a influência destas na modelação de recomendações. No entanto, apenas uma pequena percentagem

dos sistemas atualmente no mercado os aplicam Resnick and Sami (2007). Aliás, numa grande amostra de recomendações, avaliações fraudulentas a produtos da plataforma não irão produzir um ataque direto a este tipo de sistemas, dado que existe uma esmagadora maioria de dados fidedignos que as contrariam Lam and Riedl (2004). Mais ainda, em sistemas onde as únicas interações entre produtos e clientes são registadas nas encomendas, como é o caso em que nos encontramos, é extremamente rara a inferência de recomendações enviesadas, dado que a compra de um determinado produto é a prova final da preferência do cliente sobre este.

4

METODOLOGIAS

4.1 PROCESSOS

O desenvolvimento de uma solução deste tipo está longe de ser restringido apenas à sua codificação. É um processo contínuo e complexo com várias etapas que, seguidas e executadas corretamente, levarão a uma solução mais coesa e estruturada do que muitas outras onde o foco se centrou exclusivamente na implementação.

Inserindo a experiência académica ganha ao longo de quatro anos, torna-se claro que este processo deve ser regido por técnicas de engenharia adaptadas ao desenvolvimento de *software*. Citando Sommerville, previamente investigado por mim para conhecimento avançado de Bases de Dados, *"Software engineering is not just concerned with the technical processes of software development. It also includes activities such as software project management and the development of tools, methods and theories to support software development"* Sommerville (2016), que transparece a ideia que, para além da componente técnica, existem variadíssimas outras áreas que reforçam a qualidade de uma solução de *software* e que devem ser tidas em conta.

4.1.1 *Desenvolvimento de software*

Com a crescente necessidade de soluções tecnológicas, manter-se a par com a recente procura do mesmo e a constante alteração de requisitos torna-se cada vez mais árduo. A partir deste ponto de vista, várias metodologias devem ser adotadas com a finalidade de corresponder às necessidades dos clientes.

Devido à referida constante mudança de requisitos, os processos de desenvolvimento de *software* previamente planeados não estão preparados para lidar com desenvolvimentos céleres, tentando minimizar o tempo de comercialização. Como consequência, uma estratégia convencional em cascata é, frequentemente, longa.

O tempo de comercialização é extremamente fulcral a ter em conta num desenvolvimento de *software*. Novas tecnologias devem ser desenvolvidas o quanto antes de forma a impedir outras empresas de o desenvolver e lançar primeiro uma solução equiparada, que leva a um desperdício de gastos e perda de vendas. Regularmente, a decisão mais correta consiste no lançamento do produto na data previamente avançada, mesmo que alguns problemas venham a ocorrer, partindo do pressuposto que esses problemas sejam solucionados em lançamentos futuros. O gráfico que ilustra a lógica por detrás desta teoria é apresentado na figura abaixo:

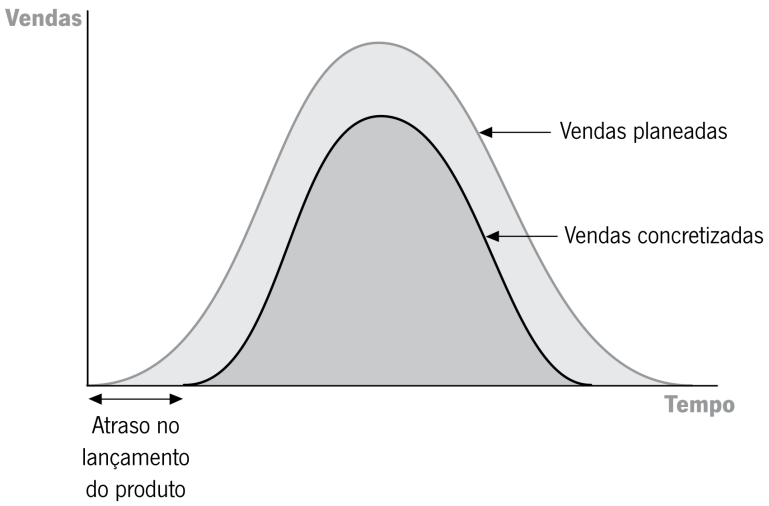

Figura 7: A consequência de adiamentos no tempo de comercialização

Considerando a necessidade de uma estratégia mais flexível e que dê azo a desenvolvimentos mais rápidos, foram estudados novos métodos. Desta forma, metodologias ***Agile*** apareceram na década de 90. Com esta abordagem, requisitos e desenvolvimento de *software* ocorrem em simultâneo, como se de um pipeline se tratasse. Toda a teoria que sustenta as metodologias *Agile* está exposta no "Agile Manifesto" Beck et al. (2001). Este indica que:

- **Envolvimento do cliente** – O papel destes consiste em providenciar e priorizar novos requisitos de sistema;
- **Aceitar a mudança** – Redefinir o sistema de forma a acomodar as mudanças nos requisitos;
- **Entrega incremental** – O cliente especifica requisitos a serem incluídos em cada incremento;
- **Manter a simplicidade** – Quando possível, trabalhar ativamente para remover complexidade ao sistema;
- **Pessoas, não processos** – As capacidades da equipa de desenvolvimento devem ser identificadas e exploradas;

4.1.2 *Data Mining*

O processo de descoberta de conhecimento é extenso, iterativo e requer várias técnicas de observação. A complexidade deste processo traduz-se na inevitabilidade do uso de uma abordagem que permita transformar problemas de negócio em tarefas, aplicar corretamente técnicas de DM e ainda disponibilizar métricas capazes de avaliar a precisão dos resultados.

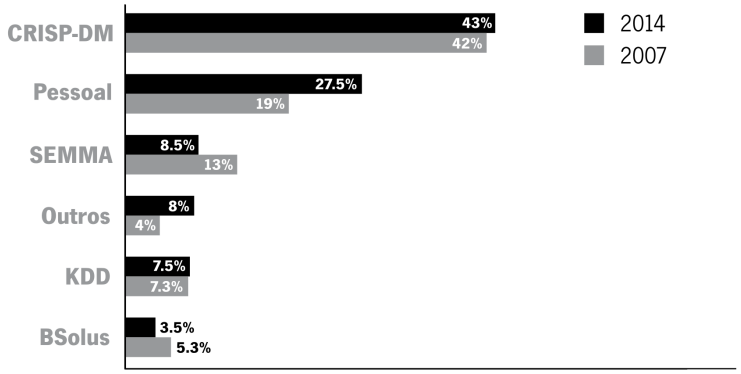

Figura 8: Principais processos de DM

Apesar de não existir nenhuma metodologia prolífica para abordar o desenvolvimento de soluções através de DM, há vários que se destacam com mais reconhecimento por parte da comunidade científica. O CRISP-DM (*"Cross-industry Standard Process for Data Mining"*) tenta solucionar parte dos problemas mencionados estabelecendo um processo independente do setor em que é aplicado, bem como da tecnologia que é utilizada e tem como propósito suportar projetos de larga escala com custo reduzido, mais fidedignos, mais modulares e, acima de tudo, mais rápidos.

O ciclo de vida de um projeto de DM que segue a estratégia CRISP-DM é composto por seis etapas distintas, apresentadas na figura abaixo. O fluxo do processo não é necessariamente sequencial, existindo dependências entre algumas etapas que estão diretamente associadas a cada projeto em particular, isto é, o fluxo deste processo vai ser determinado exclusivamente a partir dos resultados obtidos em cada etapa Wirth and Hipp (2000).

Analisando minuciosamente cada uma das etapas da estratégia CRISP-DM, é possível definir cada uma delas da seguinte forma Wirth and Hipp (2000):

- ***Business Understanding*** (Compreensão de negócio)

  Compreender os objetivos do projeto e converter os dados provenientes numa definição de um problema de DM;

- ***Data Understanding*** (Compreensão de dados)

  Obtenção de dados iniciais e técnicas de familiarização com os mesmos. Identificação de problemas de qualidade dos dados, detetar subconjuntos potencialmente relevantes para a obtenção de informações ocultas;

- ***Data Preparation*** (Preparação de dados)

Consiste em todos os métodos utilizados para a construção do conjunto de dados final. Estas tarefas serão possivelmente executadas múltiplas vezes e incluem seleção de atributos, limpeza de dados, geração de atributos, entre outros;

- ***Modeling*** (Modelação)

  Várias técnicas de modelação selecionadas e implementadas, com os respetivos parâmetros otimizados de forma a maximizar a previsão;

- ***Evaluation*** (Avaliação)

  Determinar se os resultados vão de encontro aos objetivos e respondem aos problemas de negócio formalizados previamente;

- ***Deployment*** (Lançamento)

  Colocar os modelos resultantes em prática, organizando-os e armazenando-os.

4.1.3 *Linguagem de programação*

Um dos passos fundamentais para a estruturação da metodologia a adotar neste processo é a escolha da linguagem de programação sobre a qual será feito o desenvolvimento. Como tal, foi feita uma extensa pesquisa focada nas linguagens mais populares e adotadas no desenvolvimento de processos de DM contemporâneos.

Fruto desta pesquisa, percebeu-se o grande destaque do ***Python*** no topo das escolhas, tal como indica a figura abaixo. Esta linguagem de programação de alto nível oferece a maior e mais otimizada panóplia de bibliotecas ao nível de tratamento de dados e de ML, sendo fortemente sustentada numa comunidade de suporte jovem e abrangente, conferindo-lhe então garantias de qualidade e crescimento a médio/longo prazo. Tudo isto resulta numa plataforma e linguagem de programação preparada para os desafios deste processo, nomeadamente ao nível da quantidade e fluxo de dados a tratar.

Em 2018, Weinberger mencionou que "O *Python* é, não só o melhor caminho para aprender a construir um sistema de recomendação, mas também um dos melhores caminhos para construir um sistema de recomendação no seu todo". Sendo ainda potenciado por bibliotecas como *Pandas*, *Keras*, *Scikit-learn* ou *Pytorch* aliados a integrações de alta fiabilidade com bases de dados e motores de pesquisa tais como *MySQL*, *MongoDB* ou *Elasticsearch*, pode afirmar-se que existem justificações suficientes que permitam inferir que o *Python* seja uma escolha natural para o desenvolvimento deste sistema de recomendação.

Adicionalmente, merece destaque a escolha do ***Jupyter Notebook*** como ambiente de desenvolvimento deste processo. A sua apresentação de dados através de uma estrutura amigável para o utilizador, estrutura de input/output, apresentação de gráficos e métricas de avaliação são apenas um pequeno conjunto de fatores que levam inúmeros cientistas de dados a adotar o *Jupyter Notebook* como o seu ambiente de desenvolvimento para processos de DM.

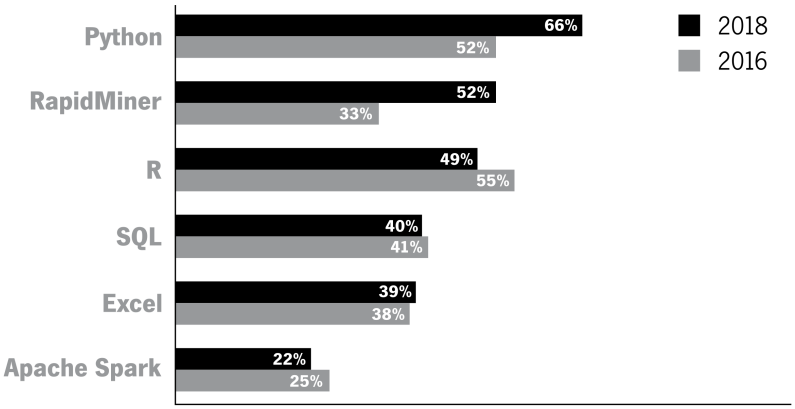

Figura 9: Principais linguagens de programação utilizadas em DM Piatetsky (2018)

4.2 ARMAZENAMENTO DE RECOMENDAÇÕES

Tendo já sido explicado, definiu-se que as recomendações deveriam ser executadas numa data e hora estabelecidas com uma frequência diária, armazenando todos os resultados obtidos numa coleção do MongoDB. Assim sendo, os modelos desenvolvidos são executados simultaneamente, produzindo recomendações para cada cliente, através dos modelos híbridos e de popularidade, bem como para cada produto, tirando partido da similaridade de produtos. De seguida, estas recomendações são armazenadas atualizando as recomendações anteriores, descartando-as da base de dados. Assim, é possível gerir trivialmente a escalabilidade do espaço ocupado por estes resultados do sistema de recomendações.

Quanto aos modelos, cada um deles é guardado num ficheiro com um prefixo de projeto atribuído – por exemplo, o modelo híbrido é denominado por *ma_hybrid.py*. Além do mais, a lista de recomendações gerada por cada modelo é armazenada no formato JSON, no qual as suas chaves correspondem ao identificador do cliente ou do produto, respetivamente. Como se verifica na listagem abaixo, o sufixo **_id** representa o identificador do cliente a quem se direcionam as recomendações do modelo híbrido. Por outro lado, o sufixo **items** representa a lista dos produtos recomendados, já ordenados pelo seu *score*. Para o caso dos produtos, a cada um dos mesmos é atribuído o identificador de produto bem como as suas categorias.

```
{
    _id: '00e2278a -8351 -459c-aa4b -b01fb966b74b ',
    items: [
        {
            product_id: '8abb3a142-c6cd-437c-b387-eaacc8b69 ',
            categories: [
                'CAPSULAS',
                'INTENSIDADES'
            ]
        },
```

```
        {
            product_id: 'ca132ff4-5457-4065-ac6b-a4257d379 ',
            categories: [
                'MAQUINAS'
            ]
        },
        (...)
    ]
}
```

Listing 4.1: Exemplo de recomendação armazenada no MongoDB

Cada documento com esta estrutura é armazenado separadamente, com o intuito de garantir a atomicidade dos dados e erradicar completamente *downtimes* nas recomendações, isto é, momentos durante a execução dos modelos durante os quais o sistema fica sem qualquer recomendação disponível. Deste modo, enquanto o sistema de recomendações executa os seus modelos a partir dos novos dados disponíveis, a plataforma mantém-se íntegra e usável.

4.3 PARAMETRIZAÇÃO DO SISTEMA

Durante a fase de investigação e definição de requisitos para o projeto, o gestor de projeto da empresa foi extremamente claro em duas vertentes. Em primeira instância, o sistema de recomendações deveria ser completamente modular e agnóstico, de maneira a ser capaz de albergar dados de plataformas de comércio eletrónico distintas (correspondente ao requisito número três). Depois, o sistema teria de ser configurável, tornando a gestão das suas execuções num processo elementar (relativo ao requisito número seis). Em conjunto, estes dois requisitos permitiram à empresa uma diminuição de custos, reduzindo a quantidade de código necessária para incutir alterações no sistema, e tornando-o plenamente adaptável a novos dados para treino que cheguem à ferramenta vindos de diferentes plataformas.

Como se percebe, estes dois requisitos cruzam-se, na medida em que a possibilidade de configuração garante um sistema mais generalizado ao nível das diferentes plataformas que trata. Para os cumprir, é então definido um conjunto de configurações, posteriormente armazenadas na base de dados do sistema. Imediatamente após o início da execução do sistema, é feita a procura e leitura dos valores atribuídos a cada um dos campos da configuração, pelo facto de cada um deles influenciar em larga escala os resultados gerados. Como tal, o sistema de recomendações tem a possibilidade de ser configurado segundo os seguintes parâmetros:

- **Evaluate**

  Trata-se de um valor binário que indica se as métricas de avaliação devem ou não ser calculadas. Depois de um processo de *tuning* no qual cada modelo foi otimizado para obter os melhores resultados, não existe a necessidade de fazer a avaliação recorrente das métricas definidas para cada caso, poupando assim tempo de execução e esforço computacional.

- **Min_score**

  Todos os produtos recomendados, como já referido, têm um *score*, que varia entre zero e um, representando a força da recomendação à qual está associado. Este parâmetro estabelece um limite inferior, eliminando da lista de recomendações todos os produtos cujo valor do *score* seja inferior ao indicado.

- **N_clusters**

  Previamente já foi descrita a forma ideal de inferir o melhor número de partições a serem feitas aos clientes – o método do cotovelo. De qualquer forma, existe a possibilidade de customizar o número de *clusters* a serem gerados pelo sistema na sua execução.

- **Weights**

  O modelo híbrido desenvolvido sustenta-se numa hibridização quantificada, ou seja, a cada um dos modelos autónomos que pertençam ao modelo híbrido estará associado um peso, regendo-se pelas fórmulas indicadas no capítulo da modelação. Com o propósito de tornar o sistema altamente configurável, estes pesos podem ser alterados a cada execução, estando definidos pelas seguintes variáveis:

  – **cbf_value**;

  – **cf_value**;

  – **cc_value**;

  Suponha-se que o administrador deste sistema pretende que as recomendações se suportem, principalmente, nos atributos de cada produto. Nessa situação, o próprio tem a possibilidade de aumentar o peso do modelo de FBC (*cbf_value*). Em contrapartida, se as interações entre clientes forem mais valorizadas, é o valor do peso relativo ao modelo de FC que deve ser incrementado (*cf_value*).

- **Last_month_orders**

  Este valor define o número de meses sobre os quais irão ser consideradas encomendas feitas na plataforma. Por outras palavras, apenas encomendas feitas nos últimos X meses serão utilizadas no sistema, sendo X o valor indicado na configuração.

- **Show_bought_products**

  Consoante os serviços e tipos de produto de cada plataforma, os clientes podem ser mais ou menos induzidos a adquirir produtos já adquiridos anteriormente. Respeitando os requerimentos, onde é essencial manter íntegra a modularidade do sistema, este parâmetro define se os produtos anteriormente adquiridos são ou não incluídos no sistema de recomendações.

4.4 SISTEMA DE *logs*

A fim de facilitar a deteção e resolução de problemas, bem como monitorizar informações relevantes durante a execução do sistema, um sistema de *logs* foi implementado e, após cada execução, estes *logs* são armazenados

na base de dados do sistema. Nestes, está registada uma data e hora específica, quais as configurações utilizadas para a execução do sistema, o número de clientes, produtos e encomendas recolhidas, todas as métricas de avaliação, o número de recomendações geradas em cada lista e, como seria de esperar, o sucesso ou erro de toda a execução do sistema.

De maneira a melhor ilustrar esta monitorização, está abaixo listado um conjunto de *logs* relativo a uma execução do sistema de recomendações, já guardado na base de dados:

```
{
    _id: ObjectId('5fe35e2ac847c004a680a3b1'),
    timestamp: ISODate('2021-10-23 T03:00:04.651Z'),
    min_score: 0.4,
    min_support: 0.1,
    n_clusters: 5,
    last_months_orders: 24,
    show_bought_products: true,
    cbf_weight: 1,
    cf_weight: 1,
    cc_weight: 1,
    ev_precision_pop: 1. 42,
    ev_recall_pop: 6. 31,
    ev_map_pop: 0. 99,
    ev_coverage_pop: 3 0. 57,
    n_complementary: 181,
    n_similar: 788,
    n_hybrid: 8226,
    outcome: 'success',
    runtime: '0:19:33.433673'
}
```

Listing 4.2: Exemplo de registos de logs armazenados no MongoDB

4.5 DOCUMENTAÇÃO

A documentação faz com que a informação seja facilmente acessível, ajuda e acelera a aprendizagem de novos colaboradores, simplifica o produto e corta gastos em suporte Trica (2020). Partindo desta sentença, todo o código desenvolvido em *Python* no âmbito desta dissertação foi documentado, seguindo o formato ***Google DocStrings***, um padrão altamente reconhecido na comunidade.

Ademais, todo o projeto desenvolvido encontra-se documentado no *Confluence*, parte integrante da plataforma de gestão de desenvolvimento *Atlassian*, utilizada pela *BSolus*. Aqui, encontra-se pormenorizadamente descrita a proposta de desenvolvimento, o contrato de dados estabelecido, todas as regras a adotar durante a implementação e, claro, todas as especificações dos desenvolvimentos feitos. Com isto, é simplificada em larga escala a introdução de novos desenvolvedores, bem como a administradores do produto, que podem consultar

todo o encadeamento de estratégias e decisões que resultaram no sistema de recomendações que se encontra disponível.

4.6 ARQUITETURA

Por último, esta secção remete para a organização e estrutura feita para o sistema desenvolvido, de modo que este responda aos requisitos estabelecidos e se sustente em todas as abordagens estipuladas. Aqui, identificam-se todos os componentes estruturais do sistema, bem como as associações entre eles Sommerville (2016).

Contrariamente a cada um dos componentes que neste capítulo irão ser identificados, a arquitetura geral do sistema não se ajusta à mudança destes, isto é, a arquitetura foi analisada, debatida e, por fim, definida com o intuito que se caracterize pela robustez e coerência. Assim sendo, ao passo que os componentes sofrem alterações constantes, ainda mais num processo de desenvolvimento Agile. A arquitetura, porém, não perturba nenhum aspeto do desenvolvimento de *software*, inclusive quando é alvo de modificações basilares.

4.6.1 *Visão geral*

Aqui, será ilustrada a abordagem que vai de encontro a todos os requisitos estabelecidos. Esta abordagem foi capaz de estabelecer métodos, algoritmos, procedimentos, técnicas e, essencialmente, *software* capaz de desenvolver e disponibilizar para lançamento o sistema de recomendações com sucesso. Assim sendo, na figura abaixo, é possível observar toda a arquitetura estabelecida, bem como todos os componentes e as relações entre si estabelecidas:

4.6.2 *Análise de componentes*

1. **Orchestrador**

   Ficheiro desenvolvido em *Python*, responsável pela orquestração e execução de todo o sistema. Este ficheiro interage com todos os restantes componentes. O processo inicia-se pelo consumo dos parâmetros a utilizar na execução, seguido da busca de dados a trabalhar, que são enviados para o ficheiro responsável pela sua preparação. Depois, os dados tratados seguem para os ficheiros de modelação, nos quais são aplicados os algoritmos correspondentes e, consequentemente, geradas recomendações, que são alvo de uma avaliação com base em múltiplas métricas. Sendo validadas, as recomendações são armazenadas na base de dados do sistema em conjunto com o ficheiro de *logs*, contendo os registos de todos os passos aqui enumerados.

2. **Preparação de dados**

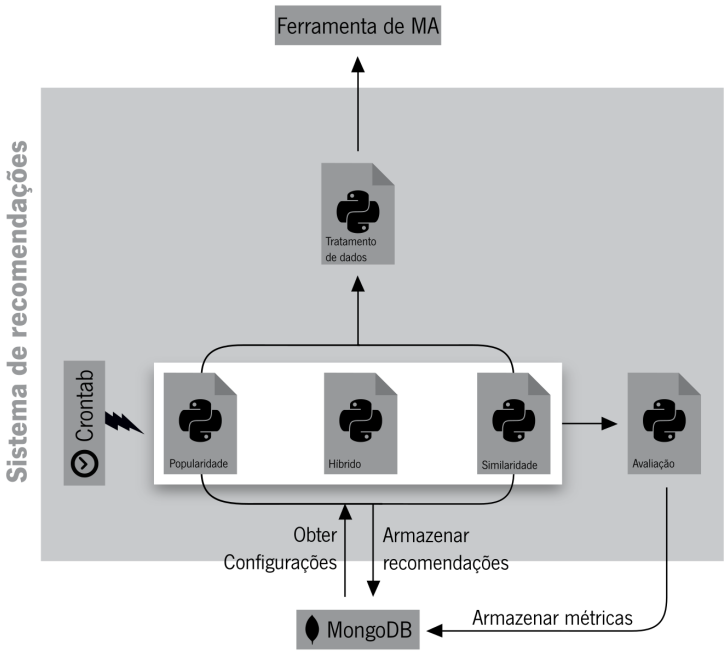

Figura 10: Arquitetura geral do sistema de recomendações implementado

Ficheiro incumbido de consumir dados armazenados na ferramenta de MA e efetuar transformações e modificações nos mesmos, de modo a mitigar possível fraca qualidade de dados e, acima de tudo, aumentar o conhecimento dos mesmos, encontrando padrões e associações.

3. **Modelação**

   Múltiplos ficheiros responsáveis pela construção de modelos de ML capazes de gerar recomendações personalizadas e orientadas a produtos e clientes da plataforma em estudo. Para tal, utiliza técnicas de FC, FBC e *clustering*.

4. **Avaliação**

   Ficheiro que recebe os resultados obtidos na modelação, validando-os segundo várias métricas, tais como a *precision*, *recall*, MAP ou a *coverage*. Todos os seus resultados são registados nos *logs* de execução, permitindo ou não que as recomendações sejam armazenadas na base de dados.

5. **Crontab**

   Agendador de execuções, capaz de executar comandos e *scripts* num momento temporal previamente especificado.

6. **MongoDB**

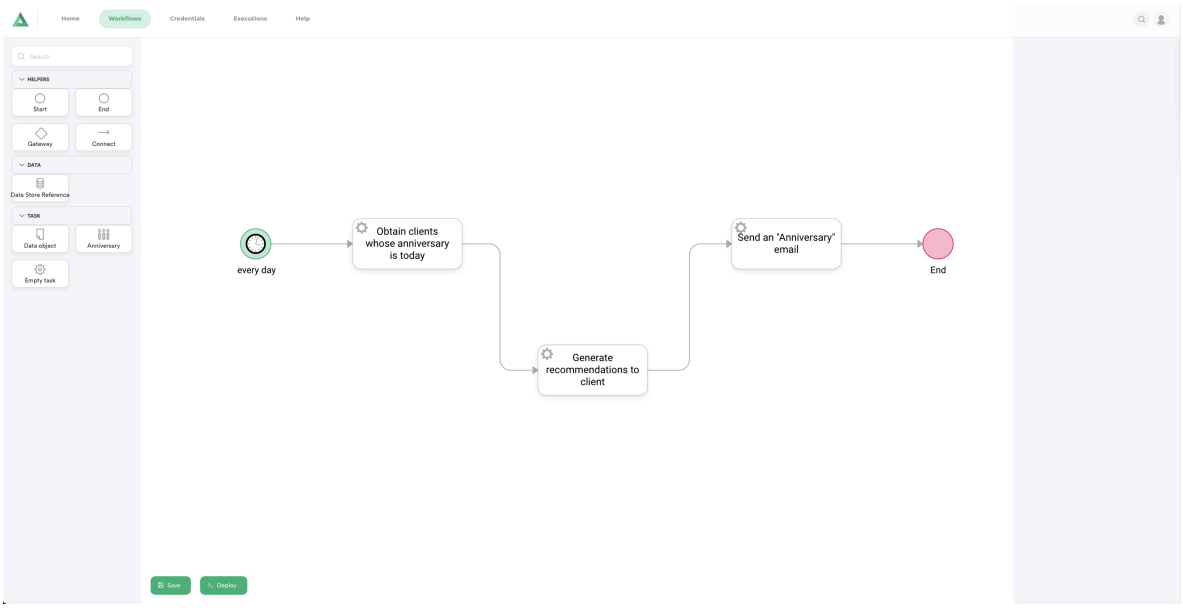

Figura 11: Exemplo de fluxo para envio de correio eletrónico personalizado no aniversário

A base de dados, não relacional, onde são armazenadas todas as recomendações feitas pelo sistema, bem como todos os *logs* de execução.

4.7 FERRAMENTA DE MA

Finalizada a implementação do sistema de recomendações proposto, segue-se a sua integração com a ferramenta que a este acede e manipula. Como já foi previamente mencionado, todas as listagens de recomendações serão armazenadas e, consequentemente, requisitadas pela ferramenta de MA desenvolvida paralelamente a este sistema.

Esta plataforma é capaz de, recorrendo a diagramas BPMN, formular fluxos de automação de *marketing* que permitam recomendar diretamente produtos a segmentos de utilizadores previamente estabelecidos com a modelação do sistema apresentado. O BPMN, sendo uma notação estruturada e albergando inúmeras peças de controlo, permite ao gestor de cada plataforma personalizar os fluxos ao mais ínfimo pormenor, de maneira a orientar as ações de *marketing* a grupos de clientes bem estabelecidos.

Como tal, é fulcral demonstrar de que forma é feita a integração entre o sistema de recomendações e esta plataforma desenvolvida. Em concreto, na figura exposta, representa-se um fluxo que envia correio eletrónico personalizado a cada cliente no seu aniversário:

Através do mecanismo de microserviços já mencionado, o orquestrador da ferramenta de MA tem a possibilidade de aceder às listas de recomendações previamente armazenadas na BD correspondente. Tendo em conta que a sua execução é diária, estas já se encontram devidamente atualizadas e em conformidade para serem utilizadas.

Assim sendo, o primeiro passo deste fluxo consiste em identificar o seu ciclo de execução. Sabendo que este caso particular remete para o envio de correio eletrónico aquando do aniversário do cliente, este ciclo estabelece-se como diário, executando numa hora específica do dia.

De seguida, e sem surpresa, é necessário identificar e agregar todos os clientes disponibilizados nos dados cujo aniversário corresponde ao dia de execução do fluxo. Este tratamento é feito integralmente pela ferramenta de MA com o propósito de filtrar uma lista de clientes com o mesmo dia de aniversário. Estes irão seguir para a "caixa" seguinte, tal como a figura sugere.

Nesta, a ferramenta de MA acede ao armazenamento de recomendações, onde irá procurar por listagens das mesmas feitas para os clientes que considerou no passo anterior. Caso não sejam encontradas recomendações para algum cliente filtrado, este será presenteado com uma listagem de produtos populares.

Por fim, toda esta informação é encaminhada para a última "caixa", que tem como propósito estruturar um *e-mail* segundo *templates* definidos e que albergue toda a informação dos clientes, bem como das recomendações geradas. Este fluxo termina, sem surpresa, com o envio de todos os *e-mails* elaborados e o respetivo término do fluxo, representado com um sinalizador vermelho na figura.

Finda a exposição de todos os elementos constituintes da arquitetura do sistema de recomendações e respetivas associações, conclui-se a descrição das metodologias de desenvolvimento adotadas neste projeto, seguindo-se a exposição detalhada de todas as implementações realizadas.

5

IMPLEMENTAÇÃO

Um processo de DM procura analisar, formular e implementar métodos que simplificam a extração de informação e padrões relevantes de um conjunto de dados não estruturado e, à partida, sem qualquer informação relevante que dele possa ser extraída.

Os processos de DM poderão ir de apenas teóricos até complexos sistemas Grossman et al. (1998) e nestes é inevitável o surgimento de impedimentos e desafios. Muitos destes foram resolvidos ou mitigados durante as últimas duas décadas e, neste capítulo, todos os desafios que apareceram durante todo o desenvolvimento serão aqui identificados e igualmente solucionados de uma forma a maximizar a qualidade do sistema no seu todo, tendo em conta os objetivos a que este se propõe.

5.1 COMPREENSÃO DE DADOS

A EDA foi introduzida por Tukey durante a década de 70, no seu livro *"Exploratory Data Analysis"*. Daí em diante, inúmeras referências a este processo são feitas por investigadores numa escala global. Em resumo, EDA é a filosofia de como a análise de dados deve ser aplicada, ao invés de um conjunto de métricas fixas e redutoras Yu (2010).

Recorrentemente, Tukey refere-se a EDA no seu livro como "trabalho de detetive" Tukey (1977). A exploração de dados permite aos desenvolvedores compreender os mesmos (corresponde ao segundo passo da metodologia CRISP-DM na qual nos encontramos) de maneira a maximizar o desempenho de algoritmos alterando pequenas imperfeições, erros ou falta de informação nos dados originais.

Com o propósito de obter os melhores aperfeiçoamentos, os cientistas de dados estão encarregues de efetuar pesquisas e investigações sobre os dados disponíveis para neles encontrar padrões, detetar anomalias, testar vários cenários e por fim validar os mesmos com o auxílio de métricas e representações gráficas Patil (2018).

5.1.1 *Análise exploratória de dados*

Neste processo, iniciou-se a análise a partir de três *dataframes*, cada um dos quais com os respetivos atributos e seus tipos, acompanhados ainda da análise das suas ocorrências. Através de uma pequena análise à figura anterior, uma incoerência nos nomes das localidades é trivialmente detetada.

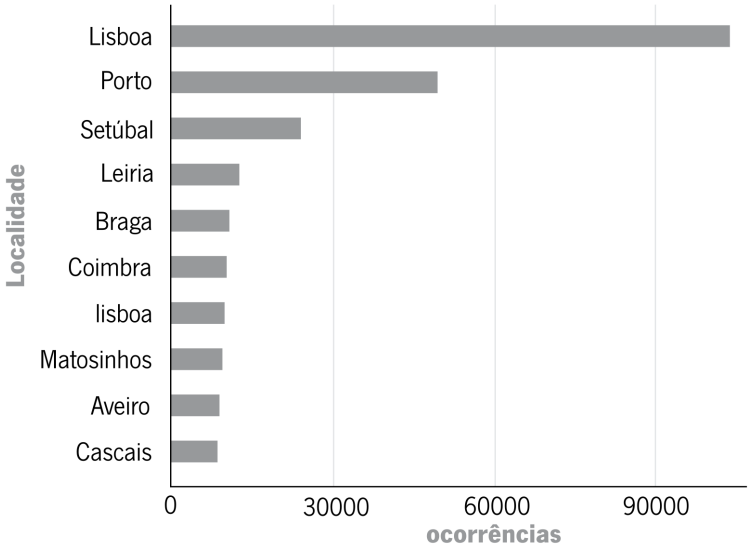

Figura 12: Distribuição das localidades dos clientes

No caso, a capital portuguesa "Lisboa" aparece com e sem maiúsculas, sendo que uma análise mais minuciosa levou ainda à deteção de entradas como "LX" e "Lx", uma abreviação comum do nome desta cidade. Todas estas ocorrências distintas remetem para a mesma cidade, porém, na fase de modelação, os algoritmos irão tratar cada uma destas ocorrências como valores distintos para o atributo ***Locality***, o que não coaduna com a realidade.

Um problema igualmente comum é a falta de dados. Valores em falta num conjunto de dados remetem para campos que se encontram vazios ou não têm a si um valor atribuído. Aquando da análise de valores em falta neste conjunto de dados, percebeu-se que existem vários elementos sem um valor associado (nomeadamente no *dataframe* de clientes, como se pode verificar na tabela abaixo). Esta anomalia afeta em larga escala a qualidade de recomendações de um sistema pelo facto de afetar diretamente as potencialidades dos algoritmos de treino, que são reféns de dados Gill et al. (2007).

Um simples exemplo de algoritmos que são fortemente prejudicados com a escassez de valores num conjunto de dados são os que se baseiam em ***clustering***. Estes algoritmos têm como propósito agrupar conjuntos de dados a partir das similaridades entre eles. Assim, a falta de valores palpáveis para cada um dos atributos em questão limitaria ou até mesmo invalidaria os resultados de certos algoritmos de ML implementados no sistema.

De seguida, destaca-se ainda a última grande complicação encontrada ao nível do conhecimento dos dados: os ***outliers***. Um *outlier* representa um valor de um atributo que não vai de encontro ao padrão de valores apresentados para o mesmo. No entanto, existem duas faces da moeda: uma delas remete para que um *outlier* poderá indicar um erro nos dados, podendo corresponder a uma gralha ou erro de medição (no caso de atributos quantitativos). Nesta situação, os valores devem ser alterados ou removidos durante a fase de tratamento, antes de qualquer modelação ou execução Dhinakaran (2018); Por outro lado, um *outlier* poderá conter dados

Entidades	Produtos		Clientes		Encomendas	
Atributos	product_id	0.00%	user_id	0.00%	order_item_id	0.00%
	parent_id	0.00%	entity_id	0.00%	order_id	0.00%
	name	0.00%	first_login	0.00%	product_id	0.00%
	stock	0.00%	last_login	0.00%	user_id	0.00%
	ordering	0.00%	email	0.00%	category	6.32%
	status	0.00%	name	0.00%	device	12.38%
	buyable	0.00%	gender	0.00%	quantity	0.00%
	language	0.00%	birth_date	0.00%	order_total	0.00%
	publish_date	0.00%	phone_number	5.23%	item_total	0.00%
	hits	32.87%	locality	52.67%	status	0.00%
	manufacturer	0.00%	city	0.00%	ship_locality	61.41%
	categories	0.00%	zip	0.00%	ship_city	0.00%
	attributes	0.00%	district	0.00%	ship_district	0.00%
	created_at	0.00%	country	0.00%	ship_country	0.00%
	updated_at	0.00%	created_at	0.00%	created_at	0.00%
			updated_at	0.00%	updated_at	0.00%

Tabela 1: Percentagem de dados em falta no conjunto de dados, por atributo

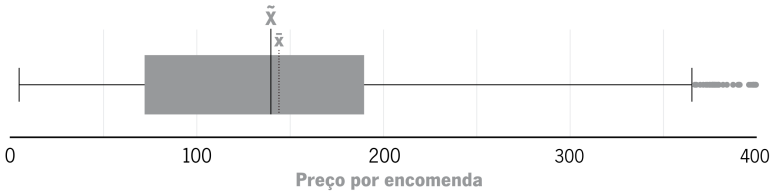

Figura 13: Distribuição de preços por encomenda

significativos para as recomendações que o sistema pretende gerar, e nessa conjuntura o mesmo deve ser mantido com o propósito de aumentar conhecimento passado aos algoritmos de ML na fase de modelação.

Para executar a deteção de *outliers* recorreu-se a diagramas de extremos e quartis, que se consideram extremamente eficazes para esse fim. Consideraram-se como *outliers* valores que se encontram fora dos extremos do diagrama, sendo eles o máximo e o mínimo. O valor máximo foi encontrado a partir da adição do terceiro quartil à amplitude interquartis multiplicada por 1.5:

$$max = Q3 + 1.5IQR$$

Ao passo que o valor mínimo foi encontrado a partir da subtração do primeiro quartil à amplitude interquartis multiplicada por 1.5:

$$min = Q1 - 1.5IQR$$

Como seria expectável, neste conjunto de dados foram detetados *outliers* em vários atributos. A título de exemplo, estes ocorrem em atributos como preços de encomendas, onde se percebe de forma célere que representam informação válida e significativa, dado que estamos perante uma loja *online* de produtos de café onde existem máquinas com um preço consideravelmente superior a cápsulas ou sacos de café. Como já havia sido mencionado, remover estas entradas do conjunto de dados seria um erro grave, que levaria à perda de informação para treino na fase de modelação.

Além da deteção de todas as condicionantes aqui mencionadas, o processo de EDA também permite um conhecimento extensivo dos dados usados no processo de DM. Como tal, foi executado nesta fase do processo CRISP-DM de maneira a não colocar em causa a integridade da metodologia que foi seguida. Assim sendo, foram criados ainda nesta fase múltiplos gráficos, expostos no apêndice deste documento, que expõem padrões de compra por parte dos clientes da Delta Portugal em determinados intervalos de tempo.

5.1.2 *Entidades e atributos*

Para o caso de estudo fornecido pela *BSolus*, a Delta Portugal, tem-se disponíveis dados reais de comércio eletrónico compreendidos há mais de três anos até aos dias de hoje. A Delta Portugal trata-se de uma reconhecida marca portuguesa de cafés, líder do mercado na venda de cápsulas e máquinas *online*.

De forma a facilitar o processo da descoberta de conhecimento, os dados habitualmente chegam estruturados em duas entidades, bem como uma ou mais interações entre os mesmos Peixoto (2021).

Em concreto, nas lojas *online* sustentadas pela plataforma da *Beevo*, as entidades definem-se por **produtos** e **clientes**, interagindo entre si através das **encomendas**. Estas três entidades contêm inúmeros atributos, como se pode verificar pela tabela abaixo, com informações específicas e valiosas sobre cada uma das entidades correspondentes.

Muitos dos atributos acima expostos são trivialmente analisados e compreendidos, porém outros levantam algumas dúvidas sobre a sua estrutura. De forma a modularizar integralmente o sistema de recomendações desenvolvido, foi especificado um **contrato de dados** que indica, obrigatoriamente, quais os atributos que as plataformas devem enviar para a ferramenta de MA, nas três entidades já apresentadas.

Abaixo é feita uma listagem descritiva de cada um dos atributos requisitados no contrato de dados, bem como o tipo de dados inerente a cada um destes:

Produtos

- ***product_id*** (*string*)

  Um código único, hexadeximal, que identifica o produto;

- ***parent_id*** (*string*)

  Um código único, hexadeximal, que identifica o produto pai. Um produto pai agrega múltiplos produtos. Por exemplo, uma máquina de café tem um produto pai e quatro produtos filho, cada um deles referente a uma máquina com cor diferente. Se um produto for produto pai, este campo deverá ser zero;

Entidades	Produtos	Clientes	Encomendas
Registos	2160	220531	525855
Atributos	product_id	user_id	order_item_id
	parent_id	entity_id	order_id
	name	first_login	product_id
	stock	last_login	user_id
	ordering	email	category
	status	name	device
	buyable	gender	quantity
	language	birth_date	order_total
	publish_date	phone_number	item_total
	hits	locality	status
	manufacturer	city	ship_locality
	categories	zip	ship_city
	attributes	district	ship_district
	created_at	country	ship_country
	updated_at	created_at	created_at
		updated_at	updated_at

Tabela 2: Entidades e atributos do conjunto de dados

- ***name*** (*string*)

  Nome do produto;

- ***stock*** (*integer*)

  Quantidade de produtos disponíveis para venda;

- ***ordering*** (*integer*)

  Ranking do produto na plataforma;

- ***status*** (*integer*)

  Estado do produto na plataforma;

- ***buyable*** (*bool*)

  Binário que define se o produto pode ou não ser comprado;

- ***language*** (*string*)

  Idioma do produto na plataforma;

- ***publish_date*** (*date*)

  Data de publicação do produto na plataforma;

- ***hits*** (*integer*)

  Número de cliques no produto por parte dos clientes;

- ***manufacturer*** (*string*)

  Fabricante do produto;

- ***categories*** (*list*)

  Lista de categorias na qual o produto se insere;

- ***attributes*** (*set*)

  Conjunto de atributos que caracterizam o produto;

- ***created_at*** (*date*)

  Data de criação do produto;

- ***updated_at*** (*date*)

  Data de modificação do produto.

Clientes

- ***user_id*** (*string*)

  Um código único, hexadeximal, que identifica o cliente;

- ***entity_id*** (*string*)

  Um código único, hexadeximal, que identifica a entidade do cliente. Este poderá ser um consumidor individual ou uma empresa;

- ***first_login*** (*date*)

  Data do primeiro *login* do cliente na plataforma;

- ***last_login*** (*date*)

  Data do último *login* do cliente na plataforma;

- ***email*** (*string*)

  E-mail do cliente;

- ***name*** (*string*)

  Nome do cliente;

- ***gender*** (*string*)

  Género do cliente;

- ***birth_date*** (*date*)

  Data de nascimento do cliente;

- ***phone_number*** (*integer*)

  Contacto telefónico do cliente;

- ***locality*** (*string*)

  Localidade onde habita o cliente;

- ***city*** (*string*)

  Cidade onde habita o cliente;

- ***zip*** (*string*)

  Código postal da morada do cliente;

- ***district*** (*string*)

  Distrito onde habita o cliente;

- ***country*** (*string*)

  País onde habita o cliente;

- ***created_at*** (*date*)

  Data de criação do cliente;

- ***updated_at*** (*date*)

  Data de modificação do cliente.

Encomendas

- ***order_item_id*** (*string*)

  Um código único, hexadeximal, que identifica o item encomendado;

- ***order_id*** (*string*)

  Um código único, hexadeximal, que identifica a encomenda realizada;

- ***product_id*** (*string*)

  Um código único, hexadeximal, que identifica o produto;

- ***user_id*** (*string*)

  Um código único, hexadeximal, que identifica o cliente;

- ***category*** (*string*)

  Categoria da encomenda;

- ***device*** (*string*)

  Dispositivo usado para efetuar a encomenda;

- ***quantity*** (*integer*)

  Quantidade do mesmo item encomendada;

- ***order_total*** (*float*)

  Custo total da encomendada;

- ***item_total*** (*float*)

  Custo total do item encomendado;

- ***status*** (*integer*)

  Estado da encomenda na plataforma;

- ***ship_locality*** (*string*)

  Localidade para onde a encomenda foi enviada;

- ***ship_city*** (*string*)

  Cidade para onde a encomenda foi enviada;

- ***ship_district*** (*string*)

  Distrito para onde a encomenda foi enviada;

- ***ship_country*** (*string*)

  País para onde a encomenda foi enviada;

- ***created_at*** (*date*)

  Data de criação da encomenda;

- ***updated_at*** (*date*)

  Data de modificação da encomenda.

5.2 PREPARAÇÃO DE DADOS

Previamente já identificada, a preparação de dados é a terceira fase do processo CRISP-DM e baseia-se num conjunto de estratégias, modificações e aperfeiçoamentos à informação ainda não tratada que chega ao sistema, após ser analisada convenientemente com o auxílio do processo de EDA, acima exposto. Este passo é crucial numa fase pré-modelação em que o sistema se encontra, visto que esta envolve a alteração, junção e remoção de dados. Este processo garante uma melhoria notável nos resultados obtidos em sistemas de recomendação pelo facto de mitigar ou até eliminar totalmente as discrepâncias nos dados provenientes da pouca qualidade dos mesmos Talend (2019).

A primeira etapa tomada para o aperfeiçoamento dos dados foi a redução. Remover informação prescindível e irrelevante para o contexto do problema não só traz benefícios na compreensão dos dados como ainda aumenta o desempenho do sistema sem abdicar de conhecimento. Ao longo da análise de valores em falta, alguns atributos foram identificados como tendo informação profundamente escassa: no caso, o atributo ***district*** relativo aos clientes e o atributo ***hits_count*** relativo às encomendas. Assim sendo, estes foram arredados do conjunto de dados tratado.

De seguida, enfrentou-se a etapa mais exaustiva e proeminente desta fase de preparação de dados: a transformação. Uma das principais complicações detetadas durante a fase de compreensão de dados consistia no facto dos clientes registarem a sua localidade num campo de texto aberto. Isto leva à incoerência de nomes de localidades, como já foi exemplificado no capítulo anterior. Neste paradigma, a implementação de um método capaz de normalizar os valores deste atributo era imperativa. Nele, o primeiro passo cifrou-se em remover os acentos de todas as palavras, seguido de uma remoção de possíveis espaços no início e no fim das mesmas e, finalmente, todas elas foram maiusculizadas.

Num cenário ideal, a aplicação de um algoritmo para normalização ou a implementação de um *geoparser* seriam as estratégias mais abrangentes a adotar. No entanto, o prazo de entrega impediu um aumento de complexidade para este caso específico e, por outro lado, o método implementado mostrou resultados interessantes, alterando mais de 20 por cento de todas as localidades presentes no conjunto de dados da Delta. Tal pode ser comprovado observando a distribuição de valores do atributo **localidade** após a sua normalização:

Com o intuito de adicionar valor e conhecimento aos dados já existentes, alguns atributos podem ser gerados e agregados ao conjunto de dados existente. Para tal, duas técnicas poderão ser utilizadas: a primeira, batizada como construção de atributos, sustenta-se na formulação e adição de novos atributos a partir de outros já presentes nos dados; outra, denominada de discretização, representa um processo de modificação de atributos contínuos com a finalidade destes serem estruturados por intervalos de valores com uma amplitude definida.

Ambas as técnicas expostas no parágrafo anterior foram, sem qualquer surpresa, aplicadas ao conjunto de dados em questão. Em concreto, a partir da data de aniversário dos clientes é possível abstrair a idade destes que, consequentemente, foi particionada em grupos de idade que seguramente irão acrescentar conexões e similaridades entre clientes. Relativamente à geração deste atributo em particular, existem ainda imbróglios a ser solucionados. O primeiro remete para os clientes que, aquando do seu registo, preencheram a sua data de nascimento com o dia em que os mesmos se estavam a registar na plataforma. Por conseguinte, foi retirado o

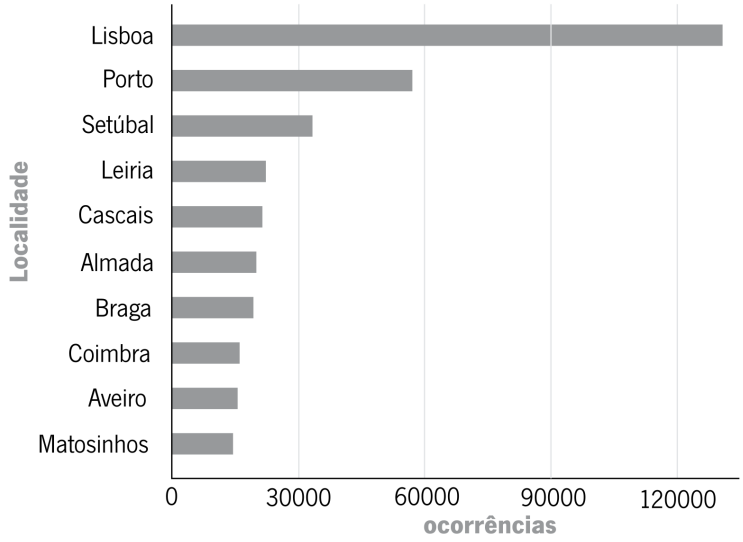

Figura 14: Distribuição das localidades dos clientes após a normalização

valor do atributo a todos os clientes cuja idade calculada é inferior a 10 anos. O segundo obstáculo é referente à partição dos grupos de idade, que acabaram por ser definidos segundo a estrutura apresentada abaixo, sendo uma de várias derivações do standard utilizado no Canadá StatCan (2017):

- **Criança:** 10 aos 17 anos;
- **Jovem:** 18 aos 25 anos;
- **Adulto:** 26 aos 64 anos;
- **Sénior:** 65 ou mais anos;

Na figura abaixo pode, inclusivamente, ser observada a distribuição da idade e dos grupos de idade dos clientes da Delta após este tratamento:

Apesar da elevada percentagem de dados em falta, o atributo ***attributes*** para os produtos é extremamente significativo, dado que se encontra estruturado num objeto chave-valor que indica os atributos de um determinado produto e o seu respetivo valor. Deste modo, foi executado um desencadeamento de cada um destes atributos, de maneira que estes fiquem no mesmo plano dos restantes. Similarmente, o atributo ***categories*** foi expandido em N atributos binários, indicando cada um deles se um produto pertence ou não à categoria especificada. Foi, ainda, aplicado um processo de normalização ao género do cliente, transformando-o num atributo binário de valores M, referentes a clientes do género masculino, e F, referentes a clientes do género feminino.

Os restantes dados em falta foram estrategicamente preservados, tendo em conta que se está perante informação útil e relevante ao sistema de recomendação, porém não existem procedimentos capazes de preencher a informação em falta de forma assertiva e com uma precisão a um nível que se possa considerar válido.

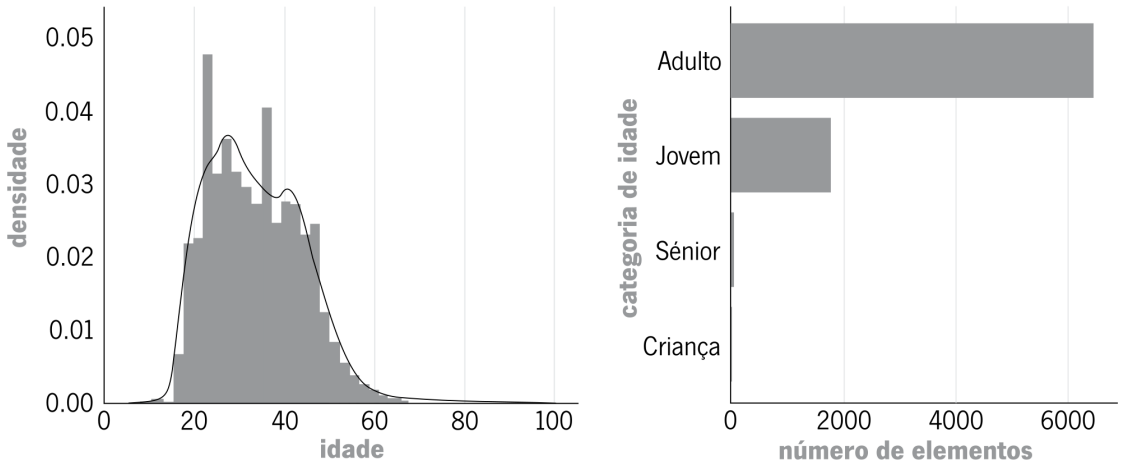

Figura 15: Distribuição das idades e respetiva categoria dos clientes

5.2.1 *Dilema da atualização*

Múltiplos serviços de *streaming*, entre os quais a já investigada *Netflix*, albergam sistemas de recomendação em constante evolução, visto que gerem milhões de contas de clientes na sua plataforma. Assim, o consumo de conteúdo por parte de um cliente é um fluxo contínuo e está em mudança constante. No caso do *YouTube*, a urgência para novas recomendações é ainda mais avultada, tornando-as ainda mais essenciais, tendo em conta as inúmeras horas de conteúdo carregadas para a plataforma a cada segundo Convington et al. (2016).

Contudo, cada abordagem é intrínseca à necessidade de cada plataforma. Isto é, ao contrário dos exemplos apresentados acima, a ferramenta de MA que alberga os dados para as recomendações requer que as plataformas aderentes lhe enviem os mesmos para que sejam tratados e manipulados. Sendo a Delta a loja *online* que se dispôs a testar esta ferramenta, está-se perante uma plataforma de comércio eletrónico visitada ocasionalmente por clientes que pretendem adquirir produtos de categorias específicas. Nestas circunstâncias, o período de inatividade na atualização de dados é incomparavelmente superior ao de atividade para a esmagadora maioria dos clientes. Além disso, a limitação computacional conhecida para o desenvolvimento desta ferramenta torna uma estratégia de atualização de dados em tempo real impraticável.

Dado este paradigma, a solução consensualizada cifrou-se na atualização dos dados numa data e hora específicas. Esta metodologia irá permitir obter novas recomendações em frequência diária, pelo que irá, de forma expectável, eliminar totalmente complicações relacionadas com a repetição de recomendações para um determinado grupo de clientes.

```
00 15 * * * /usr/local/bin/python3 /usr/src/app/ma.py >> /var/log/ma.log 2>&1
```

Listing 5.1: Agendador de execuções

Como se pode verificar com o comando acima exibido, agendar a execução de um *Jupyter Notebook* é trivial com o auxílio do **Crontab**, um agendador de execuções para sistemas Unix. Através de uma data e hora específica ou um intervalo de tempo, este consegue agendar o disparo de um comando passado como argumento nos horários atribuídos.

5.2.2 *Cold start*

Tendo já sido referenciado anteriormente, o *cold start* é um fenómeno frequente em qualquer sistema de recomendação, sendo a sua mitigação fulcral para o bom funcionamento do mesmo. No nosso paradigma, este ocorre quando não existe informação suficiente sobre um cliente para ser possível inferir uma recomendação, e essa informação centra-se nas encomendas realizadas pelo mesmo. Por outras palavras, se o cliente não tiver efetuado encomendas, não existe possibilidade de saber quais os produtos em que este se encontra interessado.

A solução que se propõe sustenta-se no desenvolvimento de outros tipos de recomendações que não dependem de informações do cliente. O problema de *cold start* em sistemas de recomendação com FC é habitualmente combatido através de um pequeno questionário inicial ao cliente de maneira a estruturar um perfil inicial, porém isso está completamente fora da área de abrangência deste projeto, visto que são plataformas externas que estão encarregues de entregar os seus dados à nossa ferramenta.

Existem, ainda, outras complicações associadas a sistemas deste tipo tais como a escalabilidade ou a possibilidade de fraude. Estas tanto não são aplicadas ao nosso paradigma como, nas identificadas, serão escrutinadas e detalhadas ao longo da fase de modelação.

5.3 MODELAÇÃO

Ao longo de todo este capítulo, irá ser apresentada, justificada e demonstrada a implementação de técnicas FC, FBC, e SRH capazes de satisfazer por completo os objetivos propostos para este projeto e que vão de encontro à explicabilidade dos dados.

Ademais, e relembrando um dos principais pilares traçados pela empresa aquando do planeamento do projeto, o sistema é capaz de gerar vários tipos de recomendações distintas que sejam válidos para utilização em diversas áreas de comércio virtual, tendo em conta os potenciais clientes da ferramenta.

Com isto em mente, foram ainda implementadas mais quatro estratégias de recomendações para as plataformas inseridas na ferramenta de MA:

- Baseadas em popularidade;
- Similaridades de produtos;
- *Clustering* de clientes;

5.3.1 *Popularidade*

Recomendações variadas e personalizadas com base no princípio de *long tail* já estudado são usualmente consideradas válidas e relevantes para os clientes a quem são atribuídas. Porém, sabe-se que a precisão das recomendações tende a diminuir à medida que avançamos na cauda Steck (2011), isto é, no eixo da popularidade.

Por esse motivo, recomendações baseadas na popularidade podem ser expressas numa estratégia comum e de trivial implementação no nosso sistema. Não sendo particularmente personalizadas e orientadas aos padrões do cliente a quem são feitas, estas recomendações apresentam precisões extremamente elevadas.

Modelos deste tipo são capazes de gerar listas dos produtos mais populares de uma plataforma e, baseando-se nesta, aplica recomendações tendo em conta a ordem da mesma. De maneira a parametrizar esta estratégia, decidiu-se calcular um valor de pontuação, vulgo *score*. Este é calculado através do seguinte conjunto de atributos, em que a cada um dos mesmos foi atribuído um peso:

- ***Orders*:** quantidade de encomendas feitas por um determinado cliente;
- ***Quantity*:** quantidade de produtos adquiridos por um determinado cliente;
- ***Status strength*:** atributo gerado a partir do estado das encomendas realizadas de maneira a traduzir-se na satisfação do cliente. Dando um simples exemplo, as encomendas canceladas representam uma baixa satisfação do cliente, ao passo que encomendas confirmadas indicam o inverso;

Para este cálculo, foi atribuído um peso de vinte por cento aos atributos *orders* e *quantity*, tendo os remanescentes sessenta por cento do peso o atributo *status strength* como destino. Tendo em conta que este último varia entre 0 e 1, a maior percentagem que lhe foi destinada no peso do *score* não lhe irá garantir uma grande preponderância, visto que os atributos *orders* e *quantity* podem atingir valores extremamente elevados. Por fim, o *score* de um produto é calculado através de uma simples média aritmética de todos os *scores* obtidos nas encomendas.

5.3.2 *Filtragem colaborativa*

Tendo já sido investigadas as estratégias de implementação de FC, sabe-se que estas se cifram em dois tipos distintos: baseadas no utilizador ou baseadas no item. Ambas são assentes em memória, utilizando assim toda a informação de clientes e produtos para serem capazes de obter previsões.

Destas, a FC baseada no item comprovou ser a solução mais estável e escalável no paradigma deste projeto. De qualquer forma, ambas as estratégias têm, comprovadamente, escalabilidade limitada para grandes conjuntos e fluxos de dados, sofrendo ainda de quedas de performance perante dados esparsos Su (2009), um problema já identificado em todos os conjuntos de dados das lojas *online* da plataforma *Beevo*.

Pelo acima mencionado, foi imperativo delinear uma nova abordagem. A FC baseada em modelos gera recomendações através de um modelo de ML sustentado em interações cliente-produto Sarwar et al. (2001).

Esta metodologia supervisiona e gere a escalabilidade de uma forma incomparavelmente mais dinâmica que abordagens de FC baseadas em memória Su (2009). Nestes referidos modelos, é possível adotar variadíssimos algoritmos, tais como:

- Redes neuronais;
- Redes de Bayes;
- *Clustering*;
- *Singular Value Decomposition* (SVD);
- Etc.

Para o nosso sistema de recomendações, decidiu-se sustentar as recomendações colaborativas num modelo de SVD com recurso a variáveis latentes. Este modelo é reconhecido e aplicado globalmente, sendo capaz de gerar recomendações com uma acurácia satisfatória e ainda oferece uma implementação mais simplificada comparativamente a algoritmos de FC semelhantes. Aliás, as potencialidades do SVD comprovam-se pelo facto deste ser, justamente, o algoritmo implementado pelo vencedor do *Netflix Prize* Koren (2009) acima investigado. Este algoritmo, na sua essência, decompõe uma matriz em três distintas, como se verifica na equação abaixo:

$$A = USV^T$$

Onde A representa uma matriz $m * n$, U uma matriz ortogonal esquerda singular $m * r$ que representa as similaridades entre clientes e variáveis latentes, S uma matriz diagonal $r * r$ que armazena os pesos de cada variável latente e, por fim, V transposta como uma matriz diagonal singular direita r$\times$n, que representa as similaridades entre produtos e variáveis latentes.

O conceito de variável latente é extremamente flexível e modular, visto que tenta descrever atributos de clientes ou produtos. Exemplificativamente, para cápsulas de café, a sua variável latente poderá cifrar-se na sua intensidade. Nesta situação, a tarefa do algoritmo de SVD consiste em reduzir consecutivamente a dimensão da matriz A, extraindo a cada iteração uma variável latente. Dessa forma, consegue mapear cada cliente e produto em respetivas matrizes de dimensão r, permitindo assim uma representação tangível de relações entre clientes e produtos.

Num algoritmo deste tipo, o número de fatores extraídos é mutável e definido como argumento. Quanto maior for o número de variáveis latentes extraídas, mais detalhada é a fatorização, levando a uma maior precisão do modelo. Por outro lado, uma extração desmedida de variáveis latentes feita com o intuito de maximizar a precisão do modelo origina um fenómeno apelidado de *overfitting*. Esta situação ocorre quando é feita uma modelação demasiado exaustiva aos dados, fazendo com que o modelo não seja capaz de generalizar os seus resultados para conjuntos de dados distintos Ying (2019) que, no sistema, lhe irão chegar como input. Devido a tudo mencionado acima, optou-se por fazer uma fatorização de dez variáveis, que serão calculados e, de seguida, utilizados para representar a matriz principal.

Atributo	Relevância
intensity	0.3536
color	0.2287
size	0.2034
weight	0.1934
capacity	0.1830

Tabela 3: Atributos mais relevantes para os clientes

Posteriormente à fatorização, a matriz é reconstruída multiplicando-a pelos fatores anteriormente gerados. A matriz resultante será, então, constituída por *scores* (no contexto do problema, representam previsões) de produtos cujo cliente ainda não adquiriu. Estes resultados obtidos são, por fim, normalizados e ordenados, dando origem a uma lista de recomendações baseadas em FC.

5.3.3 *Filtragem baseada em conteúdo*

Para o desenvolvimento de um modelo de FBC, foi utilizada uma técnica universalmente adotada em motores de pesquisa, denominada de *Term Frequency–Inverse Document Frequency* (TF-IDF). Esta técnica oferece a possibilidade de converter texto não estruturado num vetor, no qual é feito um mapeamento de todas as palavras contidas no texto para cada posição no referido vetor. Simultaneamente, a cada posição no vetor – correspondente a uma palavra – está associado um valor que quantifica a relevância de uma determinada palavra no texto Moreira (2019). Estes valores obtidos, no âmbito do caso de estudo exposto, traduzem-se na similaridade entre produtos, pelo que serão utilizados com a finalidade de criar um perfil para cada produto e, com isso, ser-se capaz de inferir produtos semelhantes.

Depois, chega-se ao último passo: modelar um perfil de um cliente. Para o conseguir, calcula-se a média de todos os perfis de produtos por um determinado cliente encomendados. Note-se que esta média estará sempre incluída no *score* de encomendas exposto acima, portanto o peso de cada produto irá, em todos os casos, estar dependente da quantidade de vezes que o cliente o encomendou (atributo *Quantity*), bem como da satisfação do cliente na aquisição (atributo *Status Strength*).

De maneira a potenciar as competências deste algoritmo TF-IDF, todos os produtos e respetivos atributos devem ser representados por um valor de texto, vulgo *string*. Consequentemente, para este modelo específico, o facto de estar disponibilizado o atributo *Attributes* na forma chave-valor contendo todos os atributos do respetivo produto demonstrou ser uma escolha válida e correta durante o desenvolvimento do contrato de dados. Como se verifica na tabela acima, os perfis modelados para cada cliente apresentam quão relevante é um conjunto de valores dentro do atributo *attributes* para um determinado cliente:

Devido ao facto de todos os produtos estarem representados no mesmo espaço vetorial, o cosseno do ângulo entre todos os vetores de produtos permite abstrair a sua similaridade. Usando, então, a similaridade por cosseno, é possível obter percentagens de similaridade entre produtos. Por fim, expectavelmente, as similaridades

por cosseno são, analogamente ao modelo baseado em FC, normalizadas e ordenadas, ficando estruturadas numa lista de recomendações.

5.3.4 *Clustering de clientes*

As recomendações feitas através de clustering são sustentadas nos atributos relativos aos clientes. O modelo tem então, como propósito, recomendar os produtos mais populares entre grupos de clientes com semelhanças, tais como:

- Idade;
- Categoria de idade;
- Género;
- Localidade;

De forma a satisfazer este enquadramento, o algoritmo escolhido foi o ***K-Means***. Este trata-se do algoritmo de *clustering* mais valorizado e escolhido entre cientistas de dados devido à sua metodologia de implementação e acerto de resultados. Apesar de exisistirem variadíssimos algoritmos de *clustering*, sendo alguns deles mais precisos nas suas recomendações, o seu custo de implementação tornou a escolha nestes inviável. Tendo também em conta o caso de estudo inerente, não existe a necessidade de um aumento de complexidade em algoritmos deste tipo.

Assim sendo, o algoritmo *K-Means* requer a especificação prévia do número de *clusters*, isto é, o número de partições a serem feitas aos dados. Tem-se, de qualquer forma, uma solução relativamente generalizada para solucionar esta questão: consiste em executar o algoritmo com números de *clusters* distintos e registar todos os resultados num gráfico no qual o eixo das abcissas representa o número de *clusters* usados e o eixo das ordenadas representa a soma dos quadrados das distâncias. Este gráfico segue a forma de uma exponencial inversa, criando uma espécie de "cotovelo". Este último indica o ponto do gráfico no qual um novo incremento do número de *clusters* não traz uma melhoria significativa de performance ao modelo.

Para o conjunto de dados estudado, os clientes serão particionados segundo os atributos mencionados acima. Partindo do facto que estes têm uma percentagem de dados em falta considerável, foi feita uma filtragem nos dados de treino removendo os clientes para os quais não existe valor correspondente em algum dos atributos indicados.

Como explicado, foi então desenvolvido o ***Elbow Method***, ou método do cotovelo, executando o modelo variando o número de *clusters* entre um e catorze e calculando a soma dos quadrados das distâncias de cada elemento até ao centro do *cluster* atribuído. Como se pode verificar no gráfico abaixo que ilustra todos os valores obtidos, é possível inferir que o número ideal de *clusters* a utilizar é três, pelo facto deste número de *clusters* corresponder à zona onde se localizou o "cotovelo":

O modelo é, então, construído, atribuindo o número de *clusters* obtido a cada cliente. Depois, o modelo de popularidade é executado em cada um dos *clusters*, gerando três listas de produtos populares. Depois de

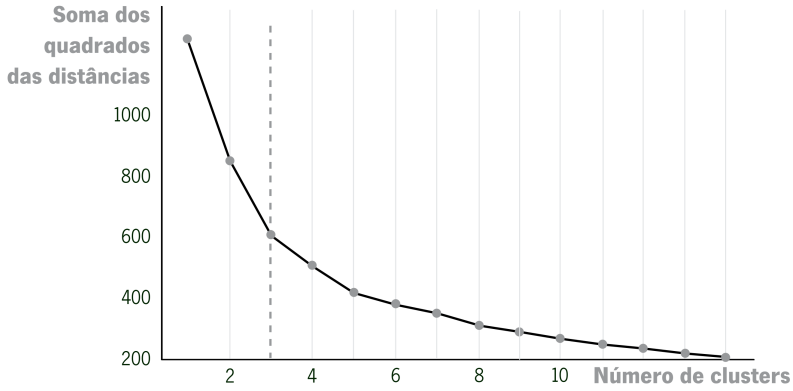

Figura 16: O número ideal de *clusters*, através do método do cotovelo

	FC	FBC	CC	SRH
Produtos sem encomendas associadas podem ser recomendados		✓		✓
Clientes sem encomendas associadas são alvo de recomendações			✓	✓
Tem em consideração o histórico de compras dos clientes	✓	✓		✓
Tem em consideração as características dos clientes			✓	✓
Tem em consideração as características dos produtos		✓		✓

Tabela 4: Prós e contras dos diferentes modelos implementados

serem, novamente, normalizados e ordenados pelo *score*, a lista de produtos populares para cada grupo de clientes encontra-se preparada para o desenvolvimento de recomendações.

5.3.5 *Híbrido*

Esta abordagem unifica duas ou mais técnicas de recomendação de maneira a mitigar as desvantagens de cada uma das abordagens a partir das vantagens de outras Suriati et al. (2017). Na tabela abaixo, é possível observar-se os prós e os contras de cada uma das abordagens estudadas e previamente mencionadas nesta dissertação: note-se que nenhuma destas incorpora todas as vantagens listadas. No entanto, uma perspetiva híbrida é capaz de gerar recomendações que satisfaçam todas as vantagens listadas. Assim sendo, um SRH que albergue as três técnicas estudadas trará dividendos ao nível de resultados.

Tendo, ainda, em consideração o facto da adição de recomendações ao sistema tendo por base o *clustering* de clientes, este SRH liberta-se da necessidade de ter apenas clientes tenham feito compras na plataforma para conseguir gerar recomendações, eliminando assim o fenómeno de "*cold start*", visto que o sistema é agora capaz de obter recomendações a partir, única e exclusivamente, dos atributos do cliente.

Vários métodos de associação de técnicas de recomendações têm vindo a ser trabalhados ao longo das duas últimas décadas, tendo sido analisados e solucionados ao longo dos últimos capítulos deste documento. Para

este sistema de recomendações, optou-se por uma hibridização quantificada. Este método oferece implementação e desenvolvimento triviais. Cada componente do sistema híbrido atribuiu um *score* a determinado produto, sendo estes posteriormente associados através de uma fórmula linear Burke (2007).

Todavia, uma hibridização deste tipo não oferece resultados de topo ao nível do desempenho, visto que apenas foi capaz de superar o melhor dos sistemas de recomendação autónomos disponíveis em dez condições, num total de trinta analisadas, de acordo com um estudo comparativo feito por Burke, em 2007 Burke (2007). Não obstante, este desempenho está intrinsecamente associado ao conjunto de dados aplicado e, apesar de uma diminuição na precisão, os ganhos ao nível da cobertura dos dados utilizados para o treino são extremamente valiosos para o caso de estudo aplicado.

De ressalvar que os pesos associados a cada um dos modelos autónomos pode ser modificado e otimizado de maneira a maximizar a acurácia do SRH, ou até mesmo para instruir a preponderância de um dos modelos autónomos sobre o sistema híbrido. Imagine-se, caso se pretenda obter recomendações orientadas à similaridade entre produtos, é imperativo aumentar o peso da FBC, de maneira a que este seja o modelo preeminente no sistema híbrido. De seguida, e de forma análoga aos restantes, os resultados obtidos neste SRH são normalizados e ordenados, a fim de obter uma lista de recomendações possível de ser armazenada e utilizada.

5.3.6 *Similaridade entre produtos*

Com o propósito de aumentar a procura e visibilidade de produtos de nicho, isto é, produtos que se encontram no fenómeno da *long tail*, os clientes têm se ser presenteados, nas ações de *marketing* da plataforma, com produtos deste tipo, similares aos que lhe são já recomendados. Este tipo de recomendações potencializa a apresentação de produtos, previamente obsoletos, ao consumidor.

Suponha-se, a título de exemplo, um *e-mail* com o propósito de recompromisso a um cliente, dado que este não consome na plataforma há mais de seis meses. Neste, pretende-se fazer uma oferta, a partir de um *voucher*, e expor produtos recomendados. Tendo em conta que se pretende cativar este cliente a uma nova compra na plataforma, é fulcral apresentar artigos que este não tenha analisado de antemão, ou até desconheça. Consequentemente, a apresentação dos referidos produtos de nicho é fundamental neste paradigma. Aliás, esta recomendação de produtos de nicho não só pode ser calculada através de produtos recomendados ao cliente, mas também através de produtos que este tenha consultado, maximizando as probabilidades de sucesso no recompromisso do cliente e, melhor ainda, expandindo a visibilidade deste tipo de produtos.

Esta abordagem segue a mesma técnica TF-IDF já descrita acima no âmbito de recomendações a partir de FBC. Sem embargo, nesta conjuntura, o perfil de um cliente ainda não está treinado e calculado, sendo apenas possível o uso do perfil de cada produto na construção de recomendações. Assim, desta vez, as recomendações são baseadas, meramente, na similaridade entre frequências de termos utilizados em cada um dos atributos do produto. O processo de cálculo do *score* a partir da similaridade por cosseno é inteiramente análogo ao descrito no subcapítulo da FBC, sendo estes normalizados e ordenados para gerar a pretendida lista de recomendações. Note-se, então, que estas recomendações não são personalizadas para um ou um grupo de clientes, sendo apenas orientadas ao produto.

5.4 AVALIAÇÃO

A avaliação é o quinto, e penúltimo, passo do processo de CRISP-DM, onde através de métricas são feitos testes, analisados resultados e, claro, comparados todos os algoritmos utilizados na fase de modelação, potenciando dessa forma a manutenção contínua de todos os modelos.

A precisão das recomendações é um conceito extremamente subjetivo e remete, invariavelmente, para os gostos de cada cliente. Só estes poderão assegurar o sucesso ou não deste sistema de recomendações, porém tal não é viável. Para colmatar esta problemática, existam várias métricas capazes de quantificar a qualidade das recomendações feitas a partir de diversos fatores. É, então, nesta secção que serão descritas, pormenorizadamente, todas as estratégias de avaliação aplicadas ao sistema de recomendações desenvolvido.

5.4.1 *Modelo benchmark*

Com o propósito de estabelecer um *benchmark*, um modelo base foi desenvolvido. Este garante o estabelecimento de um patamar inicial para comparação de algoritmos mais avançados. Por vezes, estes modelos superam o desempenho de outros mais complexos, poupando tempo de desenvolvimento e garantido um sistema mais simples, rápido e menos complexo ao nível da sua execução Ameisen (2018).

Para o sistema de recomendações em questão, este modelo base irá conter, somente, uma recomendação dos produtos mais populares. Depois de testado, verificou-se que a performance deste sistema rudimentar é relativamente satisfatória. Isto indica que, apesar do investimento na exibição de produtos de nicho de forma a diminuir o efeito de *long tail*, a grande maioria das pessoas mantém a tendência de aquisição de produtos de sucesso.

Tendo, desta forma, estabelecido o *benchmark* de sucesso para os modelos desenvolvidos, deduze-se que um modelo capaz de superar o desempenho deste, demonstra sucesso na implementação e parametrização do algoritmo em questão.

5.4.2 *Métricas*

Em cenários como este, de um sistema de recomendações, os algoritmos devolvem, como resultado, uma classificação de produtos ordenada pela probabilidade destes serem adquiridos pelos clientes a quem estes resultados são apresentados. Neste caso de estudo, pretende-se encontrar métricas capazes de recompensar o algoritmo por recomendações relevantes, isto é, tê-las o mais acima possível na sua classificação ordenada.

Depois de um estudo em panoramas de previsão, duas métricas destacaram-se das demais: ***Precision*** e ***Recall***. A *precision*, vulgo precisão, traduz a capacidade do modelo em gerar previsões relevantes, isto é, a taxa de deteção de verdadeiros positivos (TP), enquanto que a *recall* representa a probabilidade do modelo gerar a recomendação de um produto relevante. Os valores correspondentes a ambas as métricas podem ser calculados a partir das seguintes fórmulas:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Tratando-se este projeto de um sistema de recomendações, estas fórmulas apresentadas traduzem-se nas seguintes:

$$precision = \frac{\text{nº de recomendações relevantes}}{\text{nº total de recomendações}}$$

$$recall = \frac{\text{nº de recomendações relevantes}}{\text{nº de produtos relevantes}}$$

Note-se que, sendo consideradas todas as recomendações geradas pelo sistema, o valor da *recall* será sempre 1 visto que, nesse cenário, todos os produtos teriam sido recomendados. De maneira a ter a ordem da lista de recomendações em conta, a *precision* e a *recall* são calculadas a partir de um subconjunto das listas de recomendações geradas pelos modelos. Por outras palavras, será estabelecido um *ranking* de valor N, sendo para ele calculadas duas métricas: a ***precision@N*** e a ***recall@N***.

Independentemente do facto destas duas métricas recompensarem de forma correta recomendações bem sucedidas, nenhuma das duas tem em consideração o quão alto na lista de recomendações aparece um produto. Devido a este imbróglio, foi estudada uma outra métrica a incluir nesta fase de avaliação do processo: a ***Mean Average Precision*** (MAP). Esta métrica ajusta-se perfeitamente para a avaliação de classificações ordenadas de recomendações, dado que se sustenta na média aritmética das previsões. No nosso panorama, teremos a MAP@N, que é a média da ***Average Precision*** (AP) num conjunto de **N** recomendações. Assumindo e como o número de encomendas do utilizador e N como o número de recomendações geradas, é possível calcular a AP@N da seguinte forma:

$$AP@N = \frac{1}{m} \sum_{k=1}^{N} precision@k$$

A AP é diretamente proporcional à taxa de acerto nas recomendações, pelo facto da precisão do subconjunto **k** apenas se inclui na AP caso a recomendação seja correta. De forma semelhante, a precisão do subconjunto **k** é tão alta quanto mais recomendações certas forem feitas até esse ponto. Dito de outra forma, a AP premeia recomendações, à partida, corretas Sawtelle (2016).

Com a habitual crescente precisão das previsões num sistema de recomendações ao lidarem com uma maior quantidade de dados, especialmente em técnicas de FC, muitos algoritmos garantem recomendações de alta qualidade, porém descartando uma grande parcela de produtos nas mesmas, não indo de encontro a um dos propósitos deste caso de estudo, que pretende mitigar o efeito da *long tail*.

Muitas outras métricas desenvolvidas para sistemas de recomendações deste tipo são capazes de determinar a qualidade das mesmas ao nível da sua diversidade, imprevisibilidade ou até robustez. Dentro destas

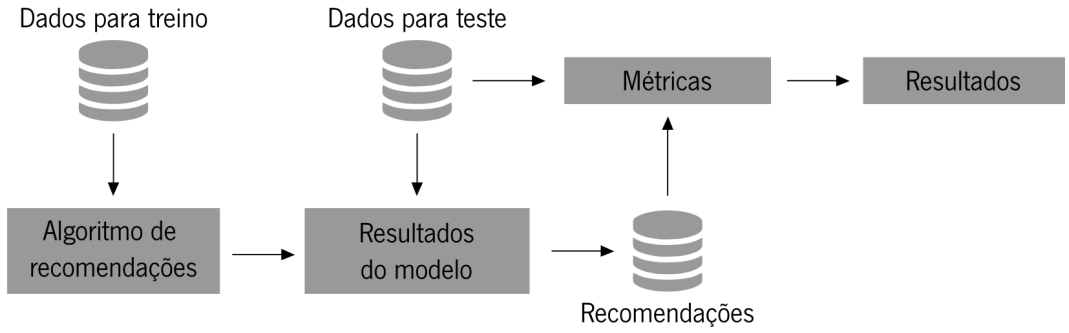

Figura 17: Processo de validação de recomendações

alternativas, encontrou-se a ***Coverage***. Esta ramifica-se em *item coverage* e *user coverage*. A primeira representa a percentagem de todos os itens que foram recomendados pelo sistema aos utilizadores. A segunda remete para a proporção de utilizadores a quem o sistema consegue recomendar itens Ricci et al. (2010).

Para o caso de estudo em questão, recorreu-se apenas à *item coverage*, que pode ser calculada da seguinte forma:

$$coverage_{item} = \frac{\text{nº de produtos recomendados}}{\text{nº de produtos disponíveis}}$$

Em cada modelo implementado, com exceção da similaridade entre produtos, todas as métricas acima mencionadas e descritas comparam as futuras encomendas dos clientes com as recomendações previamente obtidas, calculando a partir delas resultados para avaliação.

Quanto à similaridade entre produtos, tratando-se de associações apenas entre produtos, a metodologia de avaliação tem obrigatoriamente de ser distinta. No caso, a avaliação deste será feita comparando os produtos similares com produtos que foram adquiridos em conjunto com o produto estudado.

5.4.3 *Técnicas de validação*

Todo este procedimento de avaliação é essencial em projetos que envolvam modelos de ML, dado que permite comparações objetivas e minuciosas entre diferentes algoritmos e parametrização dos mesmos. Para estas comparações serem corretamente estruturadas, os modelos devem ser testados utilizando técnicas de validação.

A figura acima ilustra todo o processo de validação de recomendações em ambientes de aprendizagem supervisionada. Os dados, inicialmente, são ramificados em dois subconjuntos: o conjunto de treino e o conjunto de teste. Ao passo que o primeiro será induzido nos algoritmos implementados de forma que os modelos sejam criados, o segundo será utilizado como base para a geração de recomendações, podendo estas ser comparadas com o resultado original, presente no conjunto de teste.

Uma outra circunstância preponderante na avaliação consiste em assegurar a geração de recomendações a partir de dados com os quais o modelo não foi treinado, utilizando técnicas de validação cruzada (*cross-validation*) para esse fim. Neste projeto, será manipulada uma simples abordagem de validação cruzada designada por *hold-out*, na qual uma partição dos dados disponíveis – vinte por cento, no caso de estudo trabalhado – é posta de parte do processo de treino, ficando o seu uso exclusivo ao processo de avaliação.

Feita a segmentação das encomendas, mantiveram-se 420684 entradas para o treino dos modelos e consequente geração de recomendações, ficando as restantes 105171 responsáveis por calcular as métricas de avaliação para cada modelo.

Note-se que o momento temporal tem um peso altíssimo nas recomendações em plataformas de comércio eletrónico, como é o caso da Delta Portugal. Assim, todas as encomendas disponíveis foram previamente ordenadas pela data de criação antes de ser efetuado o particionamento descrito no parágrafo anterior. O mecanismo de separar dados de treino e de teste tendo por base uma data de referência garante uma abordagem de avaliação mais robusta e fidedigna, pelo facto de todos os dados utilizados para treino serem encomendas feitas antes da data estabelecida, ao passo que os dados utilizados para teste contêm todas as encomendas feitas posteriormente a essa data. Adicionalmente, com o recurso ao *hold-out*, é possível simular e avaliar o comportamento do sistema de recomendações num hipotético ambiente de produção, no qual não tem o conhecimento das encomendas que irão ser feitas no futuro.

De ressalvar, também, que devido ao enorme volume de recomendações e à especificação do requisito número sete enumerado acima nesta dissertação, o baixo esforço computacional deste processo aqui descrito permite incrementar o desempenho do sistema no seu todo.

5.5 LANÇAMENTO

O lançamento, vulgo *deployment*, correspondente à última etapa relativa ao processo de CRISP-DM respeitado no desenvolvimento desta dissertação, e corresponde ao processo de disponibilização do *software* desenvolvido aos utilizadores. Apesar de se tratar de um entre vários componentes, tanto o sistema de recomendações como a ferramenta de MA são parte integrante de um único projeto. Estas, no seu todo, promovem uma ferramenta capaz de disponibilizar recomendações para clientes de plataformas que com ela se encontrem conectadas, fazendo-as chegar aos respetivos clientes através de fluxos e de ações de *marketing*.

Com o intuito de agilizar este desenvolvimento, recorreu-se a *containers* para albergar todas as componentes do projeto. Estes garantem aos desenvolvedores a capacidade de criar ambientes de execução isolados das demais aplicações de uma máquina. Para este caso de estudo, ou seja, o sistema de recomendações, o *container* onde este foi alocado alberga também as dependências de *software* de todo o sistema, como é o caso do *Python* e respetivas bibliotecas desta linguagem utilizadas durante o processo de implementação.

Do ponto de vista do programador, esta estruturação de *software* é sinónimo de consistência, independentemente da plataforma que venha a ser utilizada para disponibilizar a ferramenta num ambiente de produção. Assim, é possível incrementar o tempo útil de desenvolvimento de cada programador, isto é, a sua produtividade. E, fundamentalmente, significa menos erros, tendo em consideração que os desenvolvedores podem adotar no-

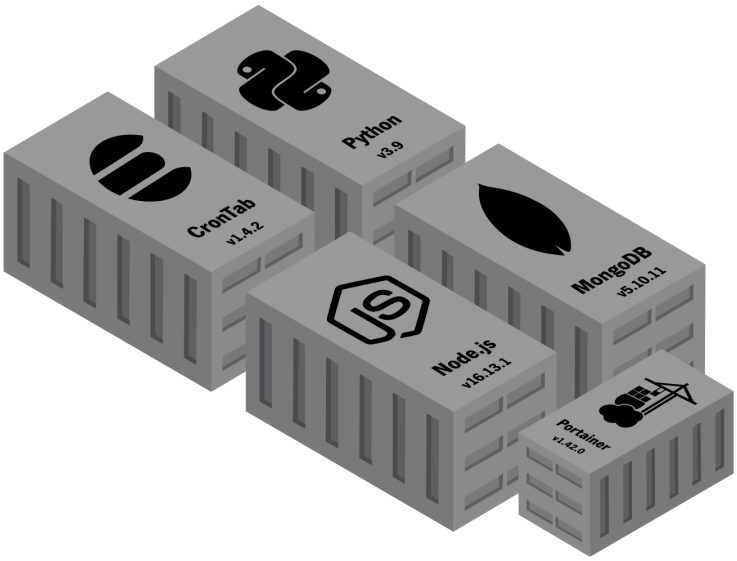

Figura 18: Ilustração dos *containers* do *Docker*

vas estratégias e assumir decisões, porque sabem que as mesmas se manterão num ambiente de produção Cloud (2021).

Dentro de um conjunto de plataformas capazes de estruturar diferentes aplicações em *containers*, optou-se pelo ***Docker***. Abaixo, é esquematizado numa figura todo o comportamento desta plataforma, que consiste num aglomerado de *containers*. Dentro de cada um destes, encontram-se dependências que são instaladas a imagem do respetivo *container* é construída. Dependências, essas, que estão definidas num ficheiro apelidado de "Dockerfile":

Extrapolando a área de estudo desta dissertação, faz-se aqui uma descrição da estrutura e desenvolvimento da ferramenta de MA, na qual este sistema de recomendações é tratado como uma componente paralela. Toda esta ferramenta foi desenvolvida pelo Diogo Gonçalves, Gil Cunha e Vítor Peixoto, a partir de ***Nest.js***: uma *framework* desenvolvida a partir da linguagem *Typescript*, que se encontra na vanguarda de novas aplicações num ambiente *Web*.

Toda esta ferramenta está, como já mencionado acima, assente numa arquitetura de microserviços, e tira partido do ***Redis*** para fazer a comunicação entre eles. Através dessa comunicação, os dados previamente carregados para a ferramenta são enviados para o microserviço que alberga o sistema de recomendação, que irá executar todo o processo de CRISP-DM redigido nesta dissertação e, por fim, armazenar todas as recomendações geradas numa base de dados ***MongoDB***. Desta forma, as recomendações encontram-se sempre acessíveis pelos microserviços responsáveis pela criação e manutenção dos fluxos de *marketing*, que serão capazes de enviar recomendações a grupos de clientes segmentados.

Recorre-se, ainda, ao ***Portainer***, uma ferramenta de gestão e manutenção de *containers* do *Docker*. Esta disponibiliza aos desenvolvedores estados, *logs* e resultados de todos os microserviços que se encontrem em execução, e encontra-se lançada num *container* independente, isto é, ao mesmo nível dos restantes.

Termina, assim, a exposição da implementação de todas as fases do processo de CRISP-DM escolhido para o desenvolvimento deste projeto. Por outro lado, ao longo do capítulo seguinte, serão apresentados e justificados todos os resultados obtidos nas execuções deste sistema.

6

RESULTADOS

Os resultados do projeto desenvolvido podem ser alvo de inúmeras análises, com perspetivas distintas. De qualquer forma, existe uma análise já documentada que terá um papel preponderante na inferência dos efeitos e consequências deste sistema: trata-se da fase de avaliação, uma de seis fases do processo CRISP-DM, na qual se encontram métricas capazes de suportar a obtenção de conclusões.

Durante esta secção, vários pontos de vista são expostos, justificados e analisados. Todos os resultados obtidos são referentes ao conjunto de dados disponibilizado pela Delta Portugal, que foram especificados acima nesta dissertação.

6.1 CALENDARIZAÇÃO

Como já referido, seguiu-se uma metodologia Agile, que fortalece a produtividade e organização em desenvolvimento de *software*. Este projeto foi desenvolvido ao longo de trinta e uma *sprints* quinzenais, compreendidas entre 2 de outubro de 2020 e 29 de novembro de 2021.

Para delinear todo o desenvolvimento, foram criadas tarefas com prazo de execução definido, estando estas agrupadas em épicos. Cada um destes épicos representava uma etapa da metodologia CRISP-DM e, mais tarde, foi criado um épico adicional para a redação da documentação. Desta forma, foi possível integrar duas metodologias, Agile e CRISP-DM, que se complementaram mutuamente e garantiram a entrega deste projeto no prazo estabelecido.

6.2 AVALIAÇÃO

6.2.1 *Resultados obtidos*

Neste espaço, encontram-se enumerados e analisados todos os resultados obtidos em relação à precisão e qualidade do sistema desenvolvido, traduzidos pelas seguintes métricas: *precision*, *recall*, MAP e *coverage*.

Tirando partido da técnica de validação *hold-out*, é exequível a simulação de um cenário real, ou seja, prever comportamentos futuros dos consumidores perante as recomendações feitas. Assim, as 420684 encomendas utilizadas para treino compreendem-se entre 9 de janeiro de 2019 e 17 de junho de 2021, enquanto as restantes

	Precision@10	*Recall@10*	*MAP@10*	*Coverage*
Popularidade	4.97%	9.78%	1.33%	37.08%
Filtragem Colaborativa	21.81%	48.34%	29.63%	51.14%
Filtragem Baseada em Conteúdo	17.44%	36.95%	28.91%	100.00%
Clustering de Clientes	20.93%	44.56%	18.92%	74.92%
Híbrido	21.62%	48.10%	35.23%	82.55%
Similaridade entre Produtos	2.59%	5.04%	0.64%	23.85%

Tabela 5: Resultados do processo de avaliação

105171 encomendas do conjunto de teste foram feitas entre 17 de junho de 2021 e 11 de novembro de 2021. Desta forma, os algoritmos irão usufruir de dados anteriores ao dia 17 de junho de 2021 para a sua execução, e as recomendações que este gerar irão ser comparadas às encomendas realizadas desse dia em diante.

Todos os resultados obtidos a partir das execuções realizadas encontram-se registados na tabela abaixo. Tenha-se em consideração que o modelo suportado na popularidade consiste no *benchmark* previamente definido, sendo útil numa análise comparativa com os restantes. Para estes testes, os pesos atribuídos a cada algoritmo no modelo híbrido foram idênticos e uniformes, ao passo que o *score* mínimo e o suporte mínimo (no caso de regras de associação) foram parametrizados com os valores de 0.4 e 0.2, respetivamente. Todas estas parametrizações são aplicadas na coleção da base de dados responsável pela gestão das configurações, que segue a seguinte estrutura:

```
{
    _id: 'delta',
    bought_products: true,
    evaluate: true,
    last_months_orders: 18,
    min_score: 0.4,
    min_support: 0.2,
    cbf_weight: 1,
    cf_weight: 1,
    cc_weight: 1,
    n_clusters: 3
}
```

Listing 6.1: Parâmetros do Sistema de Recomendações

Depois de efetivar todos os testes e analisados os respetivos resultados, várias suposições e teorias formuladas ao longo dos últimos capítulos desta dissertação. Em primeira instância, destaca-se o desempenho das recomendações geradas pelo modelo híbrido comparativamente aos respetivos modelos individuais que o constituem. Observando a métrica que melhor valida o sistema de recomendações (MAP), infere-se uma melhoria de mais de cinco por cento no modelo híbrido em relação ao melhor modelo independente que o compõe: no caso, a FC. Paralelamente, existe um ganho no desempenho a rondar os trinta por cento do modelo híbrido relativamente à média dos três modelos que o constituem.

No cômputo geral, os resultados verificados para o modelo híbrido desenvolvido são extremamente positivos, tendo em conta a esparsidade e o défice de estruturação dos dados que chegam à ferramenta de MA e, consequentemente, a este sistema de recomendações.

Além de um desempenho elevado, a abordagem híbrida expande, também, a cobertura dos dados utilizados, quando comparado com cada um dos modelos que o suportam a nível individual, sem descurar a diversidade e imprevisibilidade de recomendações potenciada por cada um destes últimos. Em concreto, o valor de 84.36% na *coverage* indica que mais de quatro em cada cinco produtos disponíveis na plataforma são recomendados aos clientes. Isto traduz-se na diminuição do efeito da *long tail*, um dos requisitos propostos neste projeto e que, nesta fase, se verifica como concretizado.

Em contrapartida, destacam-se as baixas percentagens na análise à similaridade entre produtos. Este reduzido desempenho poderá ser alusivo ao facto das recomendações apenas serem geradas a partir de padrões de compra e entre os produtos em si, descurando a influência de cada cliente. Desta forma, uma plausível falta de padrões nos produtos da plataforma levou à obtenção destes resultados. Em todo o caso, todos os modelos desenvolvidos trazem valor ao sistema devido ao conhecimento e complexidade que nele injetam.

Um outro fator relevante prende-se com o facto do modelo híbrido, o principal componente deste sistema, apresentar uma performance incomparavelmente superior ao *benchmark* estabelecido. Isto toma ainda mais preponderância quando é tido em conta que algoritmos de popularidade tendem a apresentar resultados altamente satisfatórios (Steck, 2011). Assim, comprovam-se os excelentes resultados a todos os níveis no SRH.

Por último, o algoritmo de *clustering* também apresenta valores de MAP inferiores comparativamente com modelos singulares, como é o caso da FC e da FBC. Estes números sugerem que, para o caso de estudo da Delta Portugal, recomendações sustentadas em comportamentos prévios dos clientes são mais preponderantes que outras baseadas nos atributos do cliente ou até da popularidade do produto em si.

6.2.2 *Análise comparativa*

De maneira a ser possível avaliar de forma mais minuciosa os diversos processos implementados, o desempenho do sistema no seu todo será alvo de análise após a otimização dos parâmetros de configuração inerentes ao mesmo. Finda esta otimização, o sistema encontra-se munido dos melhores resultados possíveis na sua execução.

Um dos parâmetros de configuração com mais influência na fase de avaliação e respetiva análise de resultados é o **last_month_orders**, correspondente ao intervalo de tempo compreendido entre as encomendas que se utilizam para o treino. Efetuar uma execução com encomendas mais antigas pode dar azo a recomendações que não refletem os gostos atuais dos clientes. Quando estas são filtradas, é necessário encontrar um meio termo entre recomendações com elevada fiabilidade e conhecimento necessário para a produção de recomendações. O gráfico exposto abaixo evidencia a variação da métrica MAP para cada um dos quatro modelos desenvolvidos, fazendo variar o número de meses de encomendas utilizadas para o treino.

Como se pode averiguar, a utilização apenas de encomendas efetuadas no último ano, isto é, doze meses, gera melhores resultados comparativamente aos últimos dois ou três anos, para o modelo híbrido deste sistema.

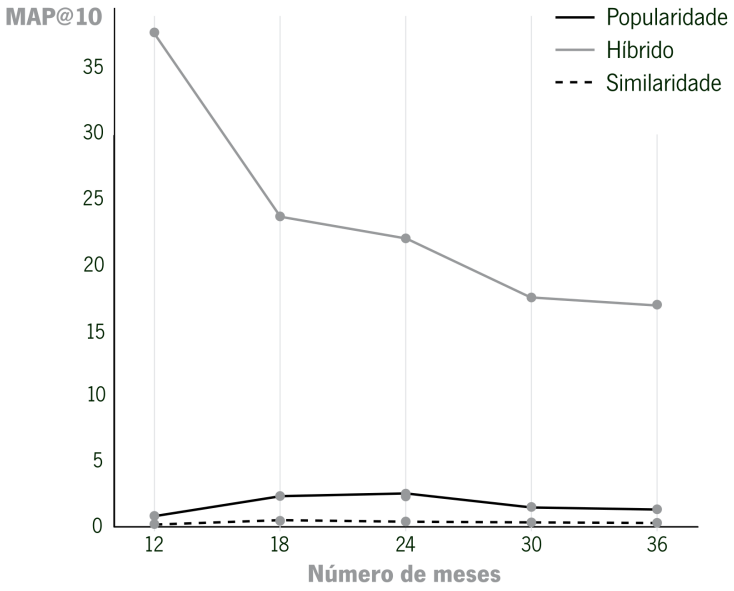

Figura 19: Resultados da métrica **MAP@10** para diferentes intervalos de últimas encomendas utilizadas

Apesar de, nessa execução, o algoritmo ser treinado com menos dados, esses mesmos refletem de forma mais aprimorada os gostos e preferências dos consumidores, levando a recomendações mais fidedignas.

No sentido contrário, tem-se o modelo de popularidade definido como *benchmark*. Analisando a figura acima, observa-se que, para este modelo, o MAP mais elevado ocorre quando são consideradas as encomendas dos últimos dezoito meses (um ano e meio). Ou seja, um maior volume de encomendas para treino auxilia este algoritmo a detetar com maior acurácia quais os produtos mais populares da plataforma em questão, a Delta Portugal. Este modelo, perante apenas encomendas mais recentes, é incapaz de formular, com um rigor satisfatório, os resultados pretendidos, dado ser facilmente influenciado por picos de vendas, nomeadamente relativas a campanhas promocionais atrativas e que, num curto período de tempo, tornam os produtos em desconto nos mais populares da plataforma. No outro extremo sucede o oposto, isto é, perante encomendas muito antigas o modelo poderá gerar recomendações de produtos cujo pico de popularidade já foi ultrapassado e, dessa forma, desfalcar o acerto do sistema.

Uma nova dedução foi inferida a partir da otimização dos pesos atribuídos a cada modelo integrante do modelo híbrido. Um modelo cujo peso seja zero faz com que este deixe de fazer parte do SRH. Então, caso todos os componentes tenham peso zero com a exceção de um modelo simples, os resultados obtidos pelo SRH serão estritamente iguais aos que foram obtidos executando o modelo simples individualmente. Considerando as encomendas dos últimos doze meses, a métrica MAP atinge o seu valor máximo quando se atribuem pesos semelhantes a cada um dos modelos, com o modelo de FC tendo um peso ligeiramente mais elevado que os restantes.

	Precision@10	*Recall@10*	*MAP@10*	*Coverage*
Híbrido s/ *clustering*	20.01%	45.70%	34.86%	77.92%

Tabela 6: Resultados do modelo híbrido sem o *clustering* de clientes

Porém, passando para um cenário onde são tidas em conta as encomendas dos últimos vinte e quatro meses, o modelo de FBC revela ter uma maior influência nos resultados do SRH. Isto ocorre devido, claro está, ao maior número de dados treinados, que se traduz em mais mudanças de preferências dos consumidores, levando a que o modelo de FC seja incapaz de detetar padrões entre produtos e clientes com tanta facilidade. Em concreto, quando se considera o último ano de encomendas, o MAP para a FC e FBC é de 29.63% e 28.91%, respetivamente, ao passo que, para dois anos de encomendas, os mesmos diminuem para os valores de 16.36% e 18.87%.

Outro tópico efetivamente de destaque prende-se com a importância do modelo de *clustering* de clientes neste sistema de recomendações. Habitualmente, um SRH combina apenas técnicas de FC e FBC mas, no paradigma apresentado, é adicionado este modelo de *clustering* com o intuito de incrementar a imprevisibilidade nas recomendações e a cobertura dos dados treinados.

Os resultados da avaliação feita ao SRH podem ser verificados na tabela acima. Porém, de forma a validar a influência positiva do modelo de *clustering* de clientes, executou-se o sistema com as mesmas configurações utilizadas na obtenção das métricas acima listadas, apenas retirando o modelo de *clustering* do SRH. Desta forma, obteve-se os seguintes resultados:

É trivial, por fim, concluir que não só o modelo de *clustering* de clientes garante mais complexidade e qualidade no SRH, como também influencia positivamente a sua precisão.

6.3 ARMAZENAMENTO

Como já exposto e justificado acima, as recomendações geradas são, integralmente, armazenadas na base de dados do sistema. No caso do modelo base sustentado na popularidade, o armazenamento de apenas um documento mostrou ser suficiente, ocupando 135.43 KB de memória. Quanto ao modelo híbrido, o principal componente de todo o sistema, foi capaz de gerar 57231 documentos, ocupando 149.01 MB e garantindo recomendações personalizadas a mais de dez por cento dos clientes da Delta Portugal. Por fim, a similaridade entre produtos deu origem a 194 recomendações, cobrindo cerca de 10% de todos os produtos disponibilizados e ocupando 1.59 MB em memória.

As baixas percentagens de cobertura de clientes e produtos acima obtidas são resultado de execuções com menos conhecimento (apenas utilizados doze meses de encomendas), bem como da esparsidade e falta de dados. Como se verificou atrás nesta secção, considerar apenas encomendas mais recentes garante uma maior precisão nas recomendações, mas, por outro lado, reduz o espetro de clientes com encomendas realizadas.

Se, por exemplo, um determinado cliente não efetuar encomendas na loja *online* da Delta Portugal no último ano, ele continuará a receber recomendações geradas para ele aquando da sua atividade na plataforma. Isto

significa que a plataforma, para este cliente, irá cessar a atualização de recomendações, ficando armazenadas na base de dados do sistema as anteriores calculadas. Deste modo, com a evolução do sistema e a passagem do tempo, o número de documentos contendo recomendações irá crescer consistentemente para a maioria dos clientes disponibilizados.

6.4 DESEMPENHO

Para um total de 173404 encomendas realizadas nos últimos doze meses na Delta Portugal, a execução de todo o processo desenvolvido neste sistema de recomendações, desde o seu início ao seu término, completou-se após 11 minutos e 33 segundos, diminuindo para 10 minutos e 23 segundos retirando do processo a etapa de avaliação, que não traz qualquer relevância para a obtenção de recomendações.

Para validar a escalabilidade do sistema, o mesmo foi colocado em execução utilizando todas as 525855 encomendas disponibilizadas pela Delta Portugal, um número que supera o triplo das encomendas utilizadas para a obtenção dos resultados otimizados. Para esta, os *logs* de execução registaram um tempo total de 19 minutos e 45 segundos, isto é, verificou-se, sensivelmente, uma duplicação do tempo de execução. Dado que o número de encomendas treinadas mais do que triplicou, este aumento de tempo de execução, além de expectável, não acompanhou o ritmo do aumento de dados, mostrando assim a escalabilidade a ele inerente.

6.5 ABSTRAÇÕES

As potencialidades deste sistema de recomendações ultrapassam, em larga escala, a avaliação dos modelos através das métricas definidas. Como abordado anteriormente, o principal objetivo de um sistema deste tipo cifra-se na criação de padrões e associações entre oferta e procura, aumentando o alcance dos clientes e dando nova vida a produtos de nicho. O crescente número destas associações leva a uma maior satisfação do cliente, dado que estes terão a perceção de uma maior gama e variedade de produtos que, potencialmente, se refletirá num avolumar de vendas e lucros.

Algumas das métricas, entre as quais a *coverage*, levam-nos a pensar que o sistema de recomendações desenvolvido será capaz de abstrair mais e mais associações, tendo em conta os resultados extremamente satisfatórios obtidos em modelos como, a título de exemplo, o híbrido. No fim de contas, será apenas durante a passagem do mesmo para um ambiente de produção que será plausível confirmar as competências deste sistema.

Este estudo poderia, no entanto, ser elaborado a partir da monitorização de novas encomendas de produtos de nicho, de forma a verificar nestas um aumento, ou através do gráfico da *long tail*, já apresentado neste documento. A partir do mesmo, tornar-se-ia possível inferir o possível avolumar da cauda, que significaria um aumento de vendas de produtos menos populares da plataforma.

6.6 OBJETIVOS ATINGIDOS

O objetivo capital com este projeto consistia no desenvolvimento de um sistema de recomendações modular e fidedigno, capaz de processar dados de uma ferramenta de MA e munindo todas as plataformas aderentes com soluções de DM ao nível do que é, hoje em dia, oferecido no mercado. De forma específica, pode-se afirmar que este sistema cumpre, metodicamente, com todos os requisitos levantados:

1. **Desenvolver um sistema de recomendações utilizando metodologias de DM contextualizadas ao problema**

   Todo o desenvolvimento respeitou as regras e indicações das metodologias Agile e CRISP-DM, garantindo dessa forma uma implementação minuciosamente planeada em todas as vertentes.

2. **O sistema deve disponibilizar múltiplos tipos de recomendações para a plataforma que o utiliza**

   A solução desenvolvida está capacitada para, no total, gerar quatro diferentes tipos de recomendações, cada qual com áreas de aplicabilidade distintas. O modelo baseado na popularidade poderá apresentar aos clientes os *bestsellers* da plataforma num e-mail de boas-vindas após o registo do cliente na plataforma. Por outro lado, o modelo sustentado na similaridade entre produtos será crucial no envio de correio eletrónico de resumo de encomenda, apresentando ao cliente produtos similares ao adquirido. Por fim, existe o modelo híbrido, de todos o mais versátil e aprimorado, que irá assegurar a maioria das recomendações feitas em ações de *marketing* pela plataforma.

3. **O sistema de recomendações deverá ser inteiramente modular, de maneira a albergar diferentes plataformas**

   Todos os algoritmos implementados nesta solução são versáteis ao nível do tipo de dados que conseguem manipular. Como tal, o único requerimento feito às plataformas aderentes consiste em ajustar os dados enviados para a ferramenta de MA, de forma a que estes respeitem um contrato de dados previamente estabelecido.

4. **O sistema de recomendações deverá albergar as mais aceites e reconhecidas técnicas de ML do mercado orientadas ao comércio eletrónico**

   Os algoritmos aplicados no sistema seguem padrões e normas que vêm sendo utilizadas de há décadas para cá por parte de inúmeros gigantes da tecnologia, como é o caso da *Amazon* e da *Netflix*, investigados nesta dissertação. A estes, juntaram-se as mais recentes implementações e algoritmos capazes de montar uma solução com alto desempenho e escalabilidade. Tal comprova-se, facilmente, pelo recurso a modelos como o SVD e o *clustering*.

5. **Os resultados obtidos pelo sistema devem ser testados através de métricas e processos de validação orientados a este tipo de problemas**

   Todas as recomendações obtidas ao longo das execuções deste sistema foram validadas por um conjunto de dados de teste, estrategicamente obtido a partir dos dados disponíveis no sistema. Inúmeras

comparações foram formuladas a partir de métricas habitualmente utilizadas em soluções de DM, tais como a *precision*, a *recall*, a MAP e, por último, a *coverage*.

6. **Desenvolver um sistema de recomendações configurável, permitindo aos seus administradores ajustar parâmetros relevantes do problema**

   Múltiplas variáveis de configuração foram adicionadas ao sistema, tendo cada uma delas um impacto direto na performance de execução dos modelos. Estas visam adaptar o sistema desenvolvido a necessidades particulares que o responsável da plataforma queira incutir às recomendações.

7. **Adotar uma abordagem *open-source* ao nível das tecnologias utilizadas e, simultaneamente, desenvolver um sistema o mais baixo esforço computacional possível**

   A implementação de uma solução altamente escalável e com atualização de recomendações diárias, ao invés das mesmas em tempo real, deu azo a que este sistema se adaptasse sem contratempos à ferramenta de MA na qual se insere.

A resolução de todos os requisitos levantados, no seu todo, levou à produção de um sistema de recomendações robusto e fidedigno. Por consequência, todas as plataformas aderentes à ferramenta de MA desenvolvida pela *BSolus*, entre as quais a Delta Portugal, terão ao dispor todas as funcionalidades que esta solução é capaz de oferecer. Este tenta cativar novamente os clientes para ambientes de comércio *online*, encontrando produtos do seu agrado e, outrora, desconhecidos. Porém, e mais importante ainda, munirá as plataformas aderentes com ações de *marketing* altamente eficazes e orientadas ao consumidor.

7

CONCLUSÕES

Os sistemas de recomendação têm tido, ao longo do tempo, um papel cada vez mais preponderante na evolução e sustentação de plataformas de comércio eletrónico no quotidiano da sociedade. As conexões e relações encontradas entre clientes e produtos permitem abstrair informações valiosas para as plataformas, que cada vez mais procuram disponibilizar uma oferta personalizada e orientada à individualidade do consumidor.

Sendo a *Beevo* uma plataforma desta área, era fulcral passar agora a ter uma ferramenta paralela, escalável e agnóstica, capaz de inferir todas estas informações e disponibilizá-las aos consumidores das suas lojas aderentes através de ações de *marketing*.

O complexo sistema de recomendações desenvolvido representa apenas uma de várias peças desenvolvidas na ferramenta disponibilizada ao gestor de cada plataforma de comércio eletrónico aderente, e na qual poderá gerir e monitorizar as já referidas ações de *marketing*, manipular os dados importados na plataforma e, claro, recomendar novos produtos aos seus utilizadores.

Em concreto, o fluxo deste sistema de recomendação inicia-se na leitura de dados brutos inseridos na ferramenta, no tratamento destes em conformidade com o caso de estudo aplicado, passando depois para a modelação através de múltiplos algoritmos. Os resultados destes, depois de avaliados minuciosamente e interpretados, dão origem a recomendações, que são armazenadas e enviadas como resultado para os fluxos montados, podendo então ser aplicadas em inúmeras ações de *marketing*. Na sua natureza híbrida, este sistema de recomendações alberga no total quatro diferentes tipos de recomendações.

A nível de modelos, o híbrido sem surpresa destaca-se. O sistema, porém, combina os três restantes modelos: FBC, que recorre ao algoritmo de TF-IDF para detetar similaridades entre atributos de produtos; FC, com uma abordagem baseada em modelos e através do algoritmo de SVD, capaz de gerar recomendações de forma célere e escalável; e *clustering*, que se sustenta no algoritmo de *K-means* para particionar os clientes em diferentes grupos, aos quais serão feitas recomendações segmentadas. Apesar de corrente, o desenvolvimento de um sistema híbrido com FC e FBC aplica todos os padrões que se exigem num motor de recomendações contemporâneo e, aliado à segmentação de clientes, trouxe resultados extremamente positivos no caso de estudo aplicado. Além do mais, os restantes modelos oferecem versatilidade e valor ao sistema, indo de encontro às necessidades de associação entre os produtos de uma plataforma e o crescente número de clientes registados na mesma.

A alteração de variáveis tem um impacto direto e significativo nas recomendações geradas e também na sua respetiva avaliação. A liberdade dada aos gestores das lojas de comércio eletrónico para adaptar as recomen-

dações aos seus modelos de negócio é garantida pela ferramenta implementada pelo Diogo Gonçalves. De mencionar, ainda, a implementação de *logs* em todo o sistema, garantindo aos desenvolvedores um controlo e monitorização simples das execuções não só no sistema de recomendações, como em toda a ferramenta.

Todo este estudo, desenvolvimento e análise trouxe resultados que atingiram e superaram expetativas para este projeto, tendo em conta a pouca estruturação, esparsidade e falta de dados, requisitados no contrato estabelecido. A significativa percentagem de alcance de itens revela que o sistema foi capaz de potenciar a recomendação de uma quantidade assinalável de produtos, bem como a percentagem de MAP, que demonstra a veracidade das recomendações geradas, tendo em conta as preferências dos consumidores. Os resultados favoráveis destas métricas indicam que foi possível adensar o efeito de *long tail* inerente aos dados da Delta Portugal, podendo isto apenas ser validado aquando da passagem de todo o sistema para um ambiente de produção.

Os resultados finais deste projeto vão, indubitavelmente, de encontro aos objetivos definidos, levando inclusive certos requisitos estabelecidos mais longe em níveis de complexidade e implementação. A aplicação de algoritmos de ML num sistema de recomendação aplicado numa ferramenta agnóstica irá permitir aos parceiros da plataforma *Beevo* desenvolverem novas estratégias de *marketing*, capazes de potenciar o compromisso dos clientes com a loja, dando-lhes a conhecer novos produtos e, implicitamente, maximizar as suas vendas.

BIBLIOGRAFIA

E. Ameisen. *Always start with a stupid model, no exceptions.* Insight, 2018.

C. Anderson. *The Long Tail*. Hyperion, New York, USA, 1st edition, 2006.

K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. *Manifesto for Agile Software Development*. 2001.

J. Boehmer, Y. Jung, and R. Wash. *E-Commerce Recommender Systems. In The International Encyclopedia of Digital Communication and Society.* American Cancer Society, 2015.

E. Brynjolfsson, Y. J. Hu, and M. D. Smith. *The Longer Tail: The Changing Shape of Amazon's Sales Distribution Curve*. SSRN Electronic Journal, 2010.

R. Burke. *Hybrid Web Recommender Systems. In The Adaptive Web: Methods and Strategies of Web Personalization.* Springer, 2007.

I. J. Chen and K. Popovich. *Understanding Customer Relationship Management (CRM): People, Process and Technology*. Business Process Management Journal pp. 672-688, 2003.

Google Cloud. *A better way to develop and deploy applications.* Google, 2021.

P. Convington, J. Adams, and E. Sargin. *Deep Neural Networks for YouTube Recommendations*. In Proceedings of the 10th ACM Conference on Recommender Systems pp. 191-198, 2016.

C. Desrosiers and G. Karypis. *A Comprehensive Survey of Neighborhood-based Recommendation Methods.* Springer US, 2011.

V. Dhinakaran. *Exploratory Data Analysis (EDA) and Data Visualization with Python*. Kite, 2018.

M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. *Collaborative Filtering Recommender Systems*. Foundations and Trends in Human-Computer Interaction pp. 81–173, 2010.

U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. *The KDD Process for Extracting Useful Knowledge from Volumes of Data*. Communications of the ACM pp. 27–34, 1996.

M. Kashif Gill, T. Asefa, Y. Kaheil, and M. McKee. *Effect of missing data on performance of learning algorithms for hydrologic predictions: Implications to an imputation technique.* Water Resources Research, 2007.

D. Gonçalves. *Development of a Marketing Automation Platform to Integrate Online E-Commerce Services.* Universidade do Minho, 2021.

R. Grossman, S. Kasif, R. Moore, D. M. Rocke, and J. Ullman. *Data Mining Research: Opportunities and Challenges. A Report of three NSF Workshops on Mining Large, Massive, and Distributed Data.* 1998.

B. Heitmann and C. Hayes. *Using Linked Data to Build Open, Collaborative Recommender Systems*. AAAI Spring Symposium pp. 76–81, 2010.

J. Juran. *Quality Control Handbook*. McGraw-Hill, New York, USA, 4th edition, 1988.

I. D. Kocakoç and S. Erdem. *Business Intelligence Applications in Retail Business: OLAP, Data Mining and Reporting Services*. Journal of Information and Knowledge Management pp. 171–181, 2010.

R. Koch. *The 80/20 Principle*. Nicholas Brealey Publishing, London, UK, 1st edition, 1997.

J. A. Konstan and J. Riedl. *Recommender systems: from algorithms to user experience.* User Modeling and User-Adapted Interaction, 2012.

Y. Koren. *The BellKor Solution to the Netflix Grand Prize.* Netflix Prize, 2009.

S. K. Lam and J. Riedl. *Shilling Recommender Systems for Fun and Profit. In Proceedings of the 13th International Conference on World Wide Web.* Association for Computing Machinery, 2004.

G. Linden, J. Jacobi, and E. Benson. *Collaborative Recommendations Using Item-to-Item Similarity Mappings*. US Patent 6,266,649 to Amazon Technologies Inc., 1998.

G. Moreira. *Recommender Systems in Python 101.* Kaggle, 2019.

P. Patil. *What is Exploratory Data Analysis? Towards Data Science.* Medium, 2018.

V. Peixoto. *Building a Hybrid Recommender Engine for E-Commerce.* Universidade do Minho, 2021.

G. Piatetsky. *Python eats away at R: Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis*. KDnuggets, 2018.

S. A. Porto Editora. *Política de Privacidade. Wook*. Porto Editora, S. A. Accessed: 2020-12-13, 2018.

F. Provost and T. Fawcett. *Data Science and its Relationship to Big Data and Data-Driven Decision Making*. Big Data pp. 51–59, 2013.

Redicom. *Marketing - Comércio B2C. Technical report*. Redicom. Accessed: 2020-12-13, 2019.

A. Reinhardt. *Steve Jobs: 'There's Sanity Returning'*. BusinessWeek pp. 62–70, 1998.

P. Resnick and R. Sami. *The influence limiter: Provably manipulation-resistant recommender systems.* RecSys'07, 2007.

F. Ricci, L. Rokach, B. Shapira, and P. Kantor. *Recommender Systems Handbook.* Springer-Verlag, 1st edition, 2010.

B. Sarwar, M. Karypis, J. Konstan, and J. Riedl. *Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International Conference on World Wide Web*. Association for Computing Machinery, 2001.

S. Sawtelle. *Mean Average Precision (MAP) For Recommender Systems.* Evening Session: Exploring Data Science and Python., 2016.

A. Sharma, J. M. Hofman, and D. J. Watts. *Estimating the Causal Impact of Recommendation Systems from Observational Data*. In Proceedings of the 6th ACM Conference on Economics and Computation, pp. 453–470, 2015.

L. Sharma and A. Gera. *A Survey of Recommendation System: Research Challenges.* Journal of Engineering Trends and Technology, 2013.

R. Sharma and R. Singh. *Evolution of Recommender Systems from Ancient Times to Modern Era: A Survey*. Indian Journal of Science and Technology, 2016.

I. Sommerville. *Software Engineering*. Pearson, Harlow, UK, 10th edition, 2016.

StatCan. *Age Categories, Life Cycle Groupings.* StatCan: National Statistical Office of Canada, 2017.

H. Steck. *Item Popularity and Recommendation Accuracy.* RecSys '11, 2011.

X. Su. *A Survey of Collaborative Filtering Techniques.* Advances in Artificial Intelligence, 2009.

S. Suriati, M. Dwiastuti, and T. Tulus. *Weighted hybrid technique for recommender system. Journal of Physics: Conference Series.* IOP Publishing, 2017.

J. Surowiecki. *The Wisdom of Crowds*. Anchor Books, New York, USA, 1st edition, 2005.

Talend. *The Definitive Guide to Data Quality.* 2019.

A. Thompson. *Best British Films 1984-2009*. The Guardian, 2009.

A. Trica. *The Importance of Documentation in Software Development.* Filtered, 2020.

J. W. Tukey. *Exploratory Data Analysis: Past, Present, and Future.* Addison-Wesley, 1977.

R. Wirth and J. Hipp. *CRISP-DM: Towards a Standard Process Model for Data Mining*. In Proceedings of the 4th International Conference on the Practical Application of Knowledge Discovery and Data Mining, pp. 29–39, 2000.

X. Ying. *An Overview of Overfitting and its Solutions*. Journal of Physics Conference Series, 2019.

C. H. Yu. *Exploratory data analysis in the context of data mining and resampling.* International Journal of Psychological Research, 2010.

Parte I

APÊNDICE

A

FERRAMENTA DE MA

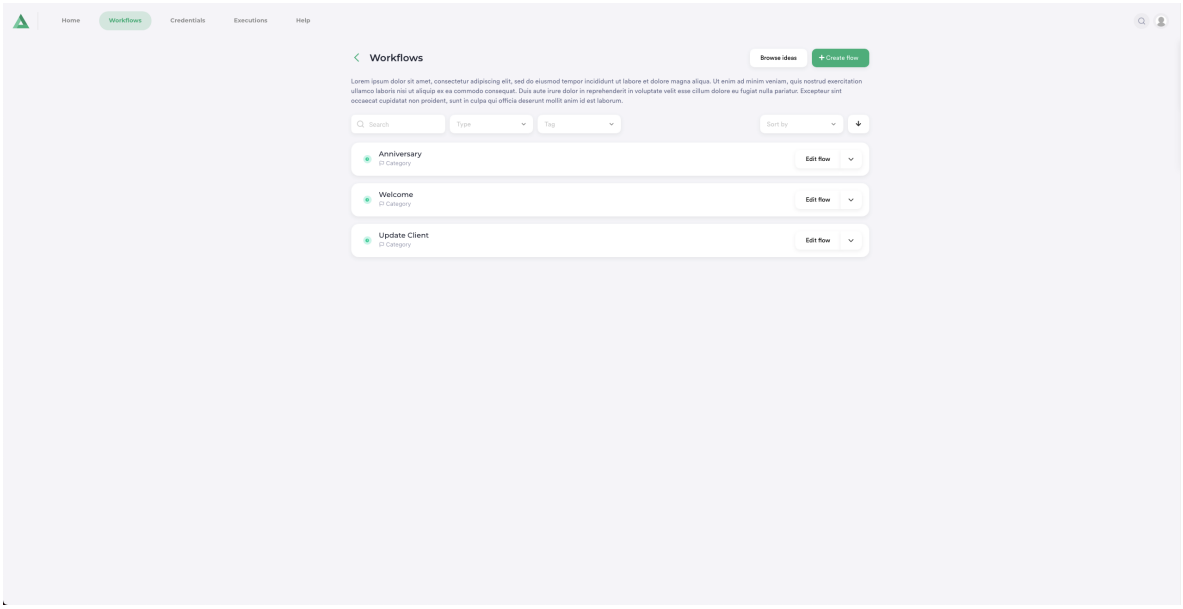

Figura 20: Listagem de fluxos de automação de *marketing* existentes

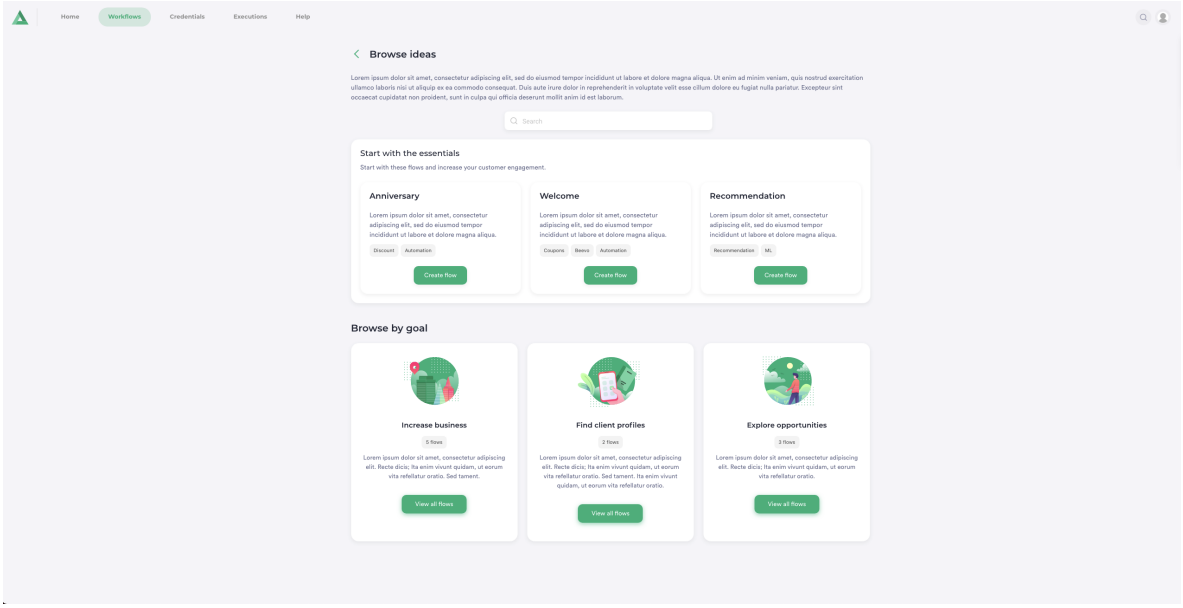

Figura 21: Ideias para criação de fluxos de *marketing*

B

COMPREENSÃO DE DADOS

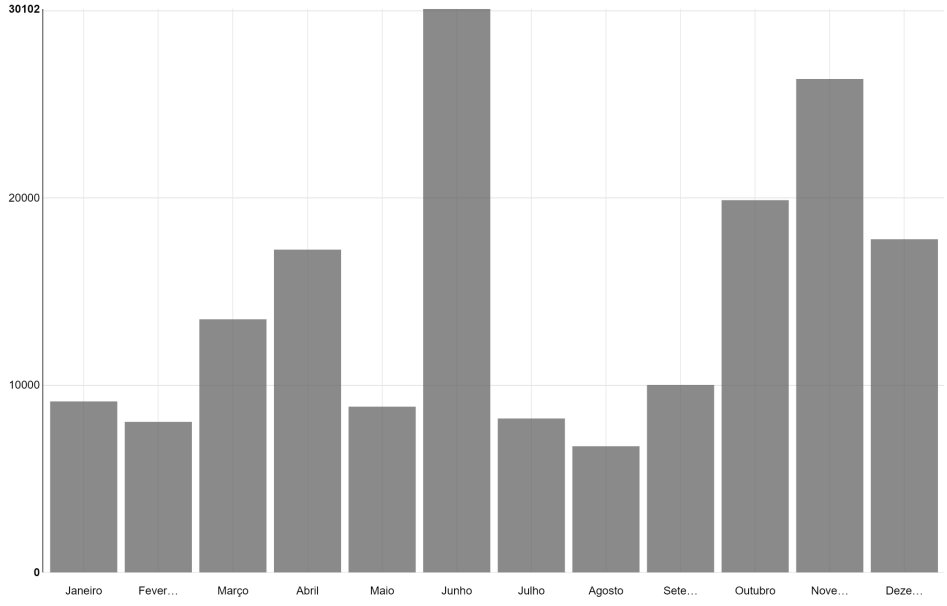

Figura 22: Fluxo mensal de encomendas

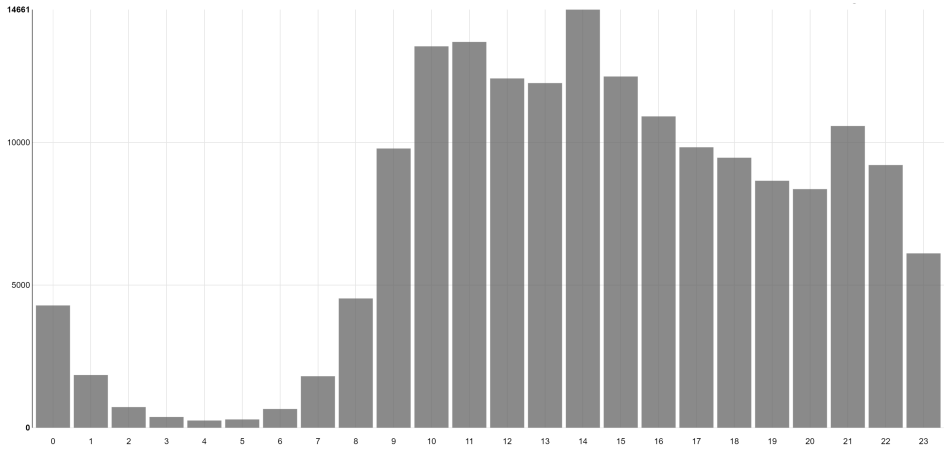

Figura 23: Fluxo horário de encomendas

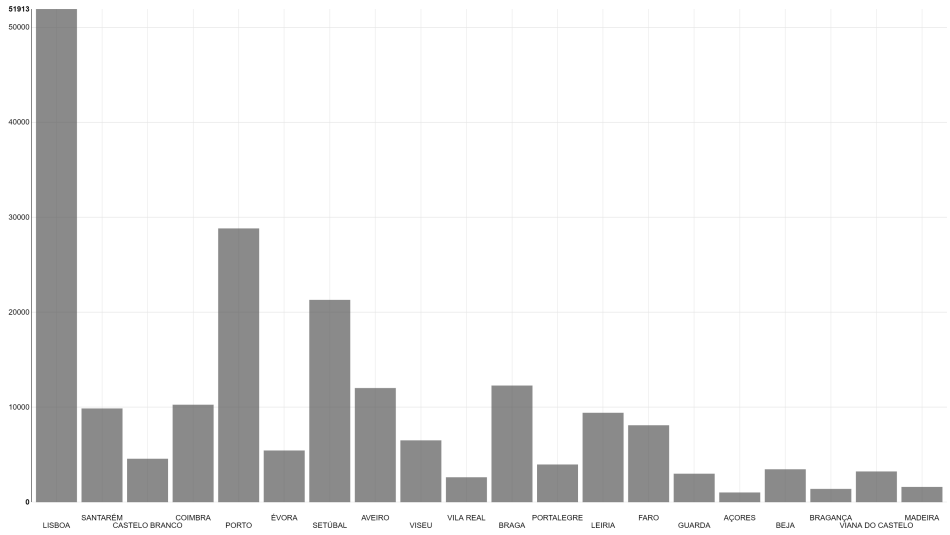

Figura 24: Fluxo de encomendas por distrito a nível nacional

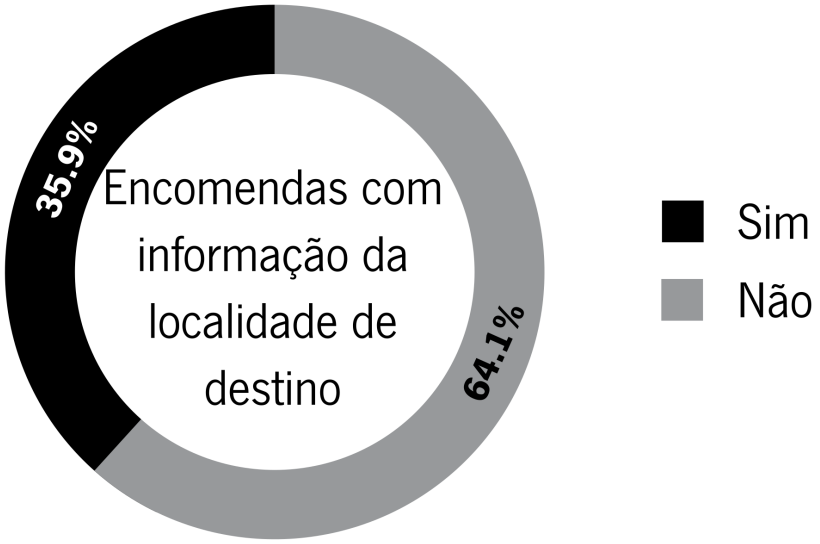

Figura 25: Percentagem de dados em falta na localidade das encomendas