Miguel Henrique Rodrigues de Almeida

Licenciado em Engenharia Informática

Análise de Dados de Atividade de Clientes para Previsão do Nível de Envolvimento

Relatório intermédio para obtenção do Grau de Mestre em

Engenharia Informática

Orientadora: Sandra Ingrez, Consultant,
Create IT, Integração e Desenvolvimento de Sistemas Informáticos

Co-orientador: João Leitão, Prof. Auxiliar,
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Fevereiro, 2019

Resumo

O tema da aprendizagem automática tem vindo a tornar-se mais popular nos últimos tempos, o que, aliado ao facto de existirem mais e melhores aplicações baseadas na *cloud* leva a que os dois temas se intercetem, utilizando assim a aprendizagem automática para evoluir os sistemas que presentemente operam na *cloud*, particularmente nesta tese, uma plataforma que gere ouso de (outros) serviços de *cloud* contratados por um conjunto de clientes.

Com o objetivo de melhorar uma plataforma existente, que permite monitorizar dados de clientes e respetivos dados de utilização de serviços online na *cloud* ao longo do tempo, surge a oportunidade de introduzir modelos de aprendizagem automática de forma a que seja possível passar a catalogar conjuntos de utilizadores com base nas suas preferências. Aliado a isto, existe também a necessidade particularmente desafiante de conseguir prever as necessidades dos diferentes utilizadores da plataforma, prevendo assim eventuais desistências e mudanças de pacote de serviços atuais. Este tipo de previsão permite uma análise muito mais completa dos dados disponíveis sobre os utilizadores e os seus serviços, bem como permite gerar mais valor proveniente da análise dos dados. Neste documento é abordado todo o percurso que culmina na implementação das funcionalidades descritas, desde a criação de um repositório que concilie todas as necessidades da plataforma, passando pelo treino dos modelos com os diferentes conjuntos de dados e finalmente a sua aplicação aos dados reais dos clientes, concluindo assim as previsões e as classificações pretendidas.

Com a aplicação das funcionalidades de classificação de clientes e de previsão de alterações de utilização por parte dos mesmos, passa a ser possível prestar um serviço com maior qualidade e com um grau de personalização completamente diferente do atual, onde o atendimento e o apoio prestado a cada cliente tem um grau de certeza e de individualização bastante superior, aumentando assim a satisfação dos clientes.

Palavras-chave: Aprendizagem Automática, Previsão de Clientes, Classificação de Clientes, Provedor de Soluções na Nuvem, Repositório de dados

Abstract

Machine learning techniques have become increasingly popular in recent times, which, together with the fact that today we have more and more complex applications based on cloud infrastructures makes this two topics become entwined, enabling the use of machine learning to evolve systems that are currently operating on cloud. In particular, in the thesis, we focus on two platforms that manages the use of (other) cloud services by two sets of clients.

In order to improve an existing platform, that monitors client data and respective usage of online services by that client over time, comes up the opportunity of introducing machine learning models to make possible the classification of users based on their preferences and usage of services. On top of this, there is also the challenging need to be able to predict the necessities of different users of the platform, looking forward to detect eventual withdrawals and changes to the current service packages contracted. This type of forecast allows a much more complete analysis of the available data regarding the customers and their services, as well as allows to generate value. We aim at the complete implementation of the described functionalities, from the creation of a repository that reconciles all the needs of the platform, through the training of the models with the different datasets, and finally, their application to the actual data of the clients, thus concluding the forecasts and the classifications of clients described above.

By applying the customer classification functionalities and predicting changes in their use, it is possible to provide a service with higher quality and with a degree of customization completely different from the current one, where the service and support provided to each client has a higher degree of certainty and individualisation, improving overall client satisfaction.

Keywords: Machine Learning, Customer Prediction, Customer Classification, Cloud Solution Provider, Repository of Data

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contexto do projeto

A palavra *cloud* é uma palavra chave nos tempos de hoje. Está a tornar-se o principal modelo de negócio para a maioria das industrias, principalmente no sector tecnológico. Neste sentido surgem os CSP (*Cloud Solution Providers*), que permitem que se monitorize o ciclo de vida de um cliente através de soluções de suporte e faturação [21].

Existem dois tipos de CSPs, o indireto e o direto. No que diz respeito ao tipo indireto este aplica-se quando é necessário fornecer serviços aos clientes mas é necessário alguma ajuda ao nível de serviços de infraestrutura. Neste sentido um provedor indireto oferece serviços de faturação, atendimento ao cliente e suporte técnico durante o ciclo de vendas e fecho pós-negócio. Estes serviços controlam inúmeros processos de *backend*, permitindo assim ao revendedor indireto que se foque apenas nas tarefas de encontrar e fechar negócios [21, 22]. Benefícios associados a este tipo de modelos passam pelo apoio fornecido, o portal comercial no qual é possível acompanhar os clientes 24/7, os guias relacionados com o negocio, que oferecem serviços complementares de forma a aumentar as receitas, bem como os termos de crédito mais flexíveis quando comparados com os canais de financiamento tradicionais.

No que toca ao modelo direto, este aplica-se quando o negócio da empresa já opera com infraestruturas de faturação, vendas e suporte funcionais, não necessitando assim que estas sejam adicionalmente fornecidas [22].

A equipa de CSP da empresa CreateIT, na qual o trabalho desta dissertação é realizado, desenvolveu a plataforma CloudCockpit, sendo que é esta a plataforma que serve de caso de estudo e foco para o desenvolvimento da solução proposta neste documento. CloudCockpit é um produto de software como serviço, do inglês *software as a service*,

onde o fornecedor do software se responsabiliza por toda a estrutura necessária à disponibilização do sistema e onde o cliente utiliza a plataforma via Internet. Neste caso o CloudCockpit é especializado em soluções Office 365, Azure, Symantec e neste momento estão em desenvolvimento novas adições.

CloudCockpit é uma plataforma que permite algumas operações, das quais é de valor destacar as seguintes:

- Permite configurar ficheiros relacionados com a utilização de serviços Microsoft por revendedor e/ou por cliente;
- Mostrar o custo de utilização por subscrição, coma opção de realizar *drill-downs* por categoria e por recurso;
- Apresentar a faturação Microsoft anual;
- Alertar o cliente relativamente a renovações de subscrições;
- Atualização automática de listagens de preços baseados na utilização dos serviços e nas licenças adquiridas;
- Mostrar as opções mais vendidas quando são adicionadas novas subscrições, como primeiras ofertas;
- Cálculos de ajustes de preços de subscrições.

A existência de informação sobre os clientes bem como os seus dados de utilização dos diferentes serviços permitem ter uma visão pormenorizada de cada cliente face aos diferentes serviços geridos pela plataforma. Neste sentido, é possível controlar os níveis de utilização dos diferentes clientes em diferentes serviços e perceber se os pacotes que estes dispõem estão de facto alinhados com as suas necessidades. Neste sentido existe também informação sobre todos os pacotes disponíveis, bem como todos os serviços pertencentes aos diferentes pacotes, esta informação é vital para perceber como diferentes pacotes de serviços permitem potencializar as diferentes necessidades dos clientes. Deste modo, a utilização de plataformas como o Cloudcockpit permitem que tanto os revendedores como os clientes tenham acesso aos seus dados de utilização de serviços ao longo do tempo.

1.2 Motivação

Enquanto que neste momento já existem plataformas como o Cloudcockpit que permitem aos seus utilizadores, sejam estes *cloud solution providers*, revendedores ou clientes finais, aceder aos seus dados de utilização de diferentes serviços existentes em diferentes pacotes, a evolução dos mercados leva à necessidade de que se dê o passo seguinte. Não limitar o acesso à informação daquilo que já existe e aconteceu no tempo, mas sim começar a

prever certos acontecimentos e eventos. Neste sentido, o desafio de melhorar a plataforma do Cloudcockpit vem na sequencia da nova tecnologia emergente no mercado eletrónico, *machine learning*.

O desafio inerente à aplicação de aprendizagem automática na plataforma do Cloudcockpit prende-se com o objetivo de conseguir prever as necessidades dos clientes, de forma a conseguir proporcionar um acompanhamento mais personalizado a cada um. É objetivo deste projeto conseguir utilizar os dados armazenados de forma a treinar modelos de aprendizagem automática que consigam detetar padrões de utilizadores e até mesmo padrões de utilização, de forma a conseguir identificar utilizadores que poderão estar em risco de deixar de ser clientes, anulando assim as suas subscrições aos pacotes disponíveis.

Á semelhança da previsão de cancelamentos de subscrições é também objetivo deste projeto a previsão detalhada de mudanças de subscrições de pacotes. Esta previsão vem no seguimento de conseguir garantir um apoio mais próximo e personalizado a cada cliente, de forma a que seja possível sugerir os pacotes que melhor se adequam a cada cliente em cada situação. Deste modo, é expectável que com a implementação desta funcionalidade seja possível prever quando é que um utilizador necessita de fazer um *upgrade* ou um *downgrade* nas suas subscrições, permitindo assim aos CSP's e aos revendedores sugerir e conciliar essas mudanças de forma atempada.

Outro desafio associado à solução a ser implementada é a possibilidade de agrupar utilizadores de acordo com os seus gastos, necessidades e consumos. Este tipo de agrupamento é bastante comum em sistemas que possuem funcionalidades de *machine learning*, posto isto a utilização dos algoritmos com este fim trará mais possibilidades de implementação de novas funcionalidades na plataforma. Associado à possibilidade de agrupar utilizadores em diferentes grupos está também a possibilidade de deteções de padrões de utilização, podendo assim identificar diferentes clientes que possuam padrões semelhantes e conseguir, de algum modo, utilizar essa informação para melhorar o serviço prestado a esses clientes.

De forma a construir uma solução que contemple todos os desafios mencionados anteriormente é necessário dividir o trabalho em diferentes etapas, etapas estas enumeradas na secção 1.3, apresentada abaixo.

1.3 Contribuições previstas

O trabalho a desenvolver, que se encontra descrito detalhadamente na secção 4, consiste na organização de dados e treino de modelos de aprendizagem automática de forma a conseguir extrair valor dos mesmos. De uma forma mais especifica, existem vários pontos a alcançar, nomeadamente:

- Construção de um repositório que concilie os dados dos clientes bem como os diferentes dados de utilização de serviços de clientes ao longo do tempo;

- Estudo e classificação dos dados disponíveis segundo um grau de relevância para o problema;
- Estudo e análise de diferentes algoritmos de aprendizagem automática a utilizar, no sentido de perceber quais os que garantem um maior grau de sucesso;
- Treino efetivo de modelos de previsão automática;
- Análise dos diferentes modelos e algoritmos obtidos;
- Análise dos resultados de previsão face aos objetivos do projeto.

1.4 Organização do documento

O restante documento apresenta uma divisão em três capítulos.

No **capítulo** 2 é fornecido algum contexto sobre a empresa na qual esta tese se enquadra, de forma a perceber que serviços dispõe, bem como quais as necessidades existentes e que visam agora ser combatidas com esta proposta de solução.

O **capítulo** 3 apresenta o estudo relacionado com o tema a ser tratado. É descrito os tipos de aprendizagem automática, sendo esta explicação mais focada na aprendizagem supervisionada, pois esta será o tipo de aprendizagem a utilizar no contexto deste projeto. É também descrito a aplicação da aprendizagem automática quando aplicada à modelação dos utilizadores, bem como os tipos de filtragem de informação que proporcionam um desenvolvimento mais objetivo de soluções.

O **capítulo** 4 apresenta em detalhe as diferentes etapas necessária à elaboração da solução, bem como o planeamento geral das atividades.

Capítulo 2

Contexto

2.1 História da CreateIT

A Create IT foi fundada em 2001 com um princípio orientador: "desenvolver projetos que antecipam o futuro e que representam um efetivo valor acrescentado para o negócio dos nossos clientes."[14]

Sediada na praça de Alvalade em Lisboa, é uma empresa de cariz tecnológico especialista na criação de sistemas multi-plataforma críticos e de suporte ao negócio. Com uma vasta experiência em projetos para clientes de setores tão diversos como as Telecomunicações, Financeiro, Indústria, Distribuição, Turismo e Administração Pública.[14]

2.2 Serviços

As soluções de serviços são focadas no aumento da produtividade de quem as usa e são pensadas desde o início para responderem aos desafios do negócio, transformando-o no caminho da otimização e potenciando o seu crescimento.

As soluções colaborativas e adoção do Office 365 são um pilar muito forte na empresa, estas ferramentas permitem que se tire partido de tudo o que o SharePoint e o Office 365 oferecem para partilhar e colaborar com os seus colegas, parceiros e clientes. [16]

Um dos pontos fortes é o desenvolvimento aplicacional rápido, o histórico em Azure, Office 365 e Integração de Sistemas são a alavanca perfeita para construir aplicações críticas de suporte ao negócio mais rápido que nunca com ajuda da plataforma de baixo código da OutSystems.[16]

Com uma equipa com mais de 15 anos de experiência e como Parceiros Gold da Microsoft na área de Integração de Aplicações, é possível criar soluções empresariais completas para responder às suas necessidades dos clientes.

2.3 Produtos e Soluções

A Create IT criou internamente um conjunto de produtos que respondem a necessidades do mercado, prontos a utilizar e de acordo com as melhores práticas do mercado. Além disso, tem as competências e a experiência para implementar soluções com base em tecnologia líder de parceiros com os quais mantém uma relação próxima e de grande confiança.[14]

O SmartPortals é a plataforma de eleição para o rápido desenvolvimento de portais, já que a sua extensibilidade suporta a integração com os sistemas dos clientes. Tendo por base o CMS Umbraco, gerência de conteúdos assente em tecnologia *opensource*, permite gerir conteúdos de forma ágil e responde a todas as dimensões de informação, destacando-se pelas suas características de escalabilidade.[29]

O DiggSpace é um portal de comunicação interna pronto a usar que assenta em SmartPortals, tirando partido das funcionalidades do Office 365. Com o DiggSpace a empresa promove a produtividade, o conhecimento e a eficiência, permitindo reforçar o sentimento de pertença, colaboração, envolvimento, compromisso e partilha entre todos os colaboradores.[29]

O CloudCockpit é uma consola web para gerir de forma centralizada a rede CSP de produtos *cloud*, permitindo gerir a informação do cliente e controlar as subscrições. Como está disponível num modelo de software como serviço (do inglês Software-as-a-Service (SaaS)), não há qualquer investimento inicial nem custos de instalação e configuração.[29]

2.4 Relevância do projeto

O tópico desta tese insere-se no produto CloudCockpit da empresa, este permite controlar e gerir informações dos clientes e controlar as diferentes subscrições que estes possuem. Atualmente a plataforma permite recolher dados de utilização de diferentes serviços *cloud*. Neste sentido é possível acompanhar o crescimento de um cliente ao longo do tempo no que diz respeito ao seu nível de utilização de serviços Microsoft.

Existem diferentes tipos de serviços, começando pelos serviços como o Azure, os quais funcionam com base na *cloud* e são faturados de acordo com a utilização que o utilizador faz dos serviços. O tipo de dados que o CloudCockpit permite acompanhar relativamente a estes serviços de utilização passam desde os dados básicos de faturação, como as datas em que um serviço é faturado, a região de faturação, bem como os preços para os diferentes revendedores ou clientes do serviço e os impostos inerentes às taxas cobradas pelos serviços. Existe também informação especifica sobre o serviço que está a ser faturado, desde o seu tipo, como por exemplo, se é um serviço de armazenamento, de máquinas-virtuais e se estes possuem replicação, cópias de segurança ou outras definições associadas. Todos estes dados estão relacionados com o cliente final, o provedor do serviço e o revendedor do mesmo, sendo que todas estas informações são também disponibilizadas na plataforma.

Para além dos serviços que são faturados com base na utilização existem ferramentas que são faturadas com base em pacotes de subscrições. É o caso de serviços do Office 365 como o skype para empresas, o SharePoint, o Teams e muitos outros.

No serviço por subscrição o tipo de dados são um pouco diferentes quando comparados com os dados dos serviços faturados por utilização. Existe a informação relativa ao nome do serviço e a que tipo de pacote este pertence, e existe a noção de licenças adquiridas e licenças utilizadas, isto porque um cliente pode adquirir várias subscrições e não as ativar para utilização. No entanto, analogamente aos serviços faturados por utilização, existe também a informação financeira associada às diferentes subscrições, como os custos associados à licença, ao revendedor e ao cliente final.

O trabalho proveniente da tese em questão tem como objetivo melhorar a plataforma atual, utilizando os diferentes dados de forma a conseguir construir modelos de aprendizagem automática que consigam prever situações que necessitem de uma potencial intervenção. Estas intervenções podem passar por sugerir a um cliente um *upgrade* ou *downgrade* do pacote de serviço, de forma a ir ao encontro do que melhor serve o cliente. Outro aspeto importante é conseguir prever se um cliente está em risco de deixar de o ser, alertando assim para uma possível intervenção atempada de forma a evitar que se perca um cliente. Estas novas possibilidades de prever o que melhor se adequa a um cliente com base nos seus dados, vem aumentar o valor do CloudCockpit, melhorando assim o serviço que é prestado.

Capítulo 3

Trabalho Relacionado

Neste capítulo é debatido trabalho relevante para a solução a desenvolver, contendo assim uma síntese inicial de trabalho relacionado. Em particular são debatidos os seguinte tópicos:

Na secção 3.1 são abordadas diferentes formas que possibilitam a aprendizagem automática.

Na secção 3.2 as especificações deste tipo de aprendizagem automática bem como os seus problemas e técnicas associadas são discutidas.

Na secção 3.3 é discutida a aplicação direta de técnicas de aprendizagem automática associada às ações e padrões dos utilizadores.

Por fim, na secção 3.4 o tema abordado passa pelos sistemas de filtragem de informação e a sua mais valia no tratamento de dados.

3.1 Tipos de aprendizagem automática

As pessoas podem cometer erros durante a análise, ou possivelmente, quando tentam estabelecer relações entre diferentes métricas, isto torna difícil encontrar soluções para determinados problemas. A utilização de aprendizagem automática pode muitas vezes ajudar na solução destes problemas, aumentando a eficiência dos sistemas.

Existem diferentes tipos de aprendizagem automática [17], pelo que a forma como a informação é apresentada aos algoritmos de aprendizagem têm influência na forma como estes processam a mesma e tiram conclusões úteis à resolução dos problemas.

3.1.1 Aprendizagem por implantação

Quando estamos na presença de uma aprendizagem por implantação direta de novo conhecimento não há inferência ou outras transformações de conhecimento no lado do

algoritmo de aprendizagem. Variantes deste tipo de aquisição de conhecimento incluem:

- Aprender por ser programado, ou seja, o mecanismo de aprendizagem é construído ou modificado por uma entidade externa, o que não apresenta nenhum tipo de esforço do lado do algoritmo no que toca à inferência de conclusões, visto que este é definido explicitamente pelo programador.[17]
- Aprender por memorização de factos dados sem inferências sobre a informação. O termo *rote learning* é utilizado primariamente neste contexto, sendo que a aprendizagem se dá por repetição. Este é o método que mais se aproxima da aprendizagem realizada pelo ser humano, uma vez que a informação é aprendidas através da sua captação por repetição.[17]

3.1.2 Aprendizagem por instruções

Numa transposição para o mundo real, é possível mapear este tipo de aprendizagem como aprender através de um professor ou outra fonte organizada. Requer que o aprendiz transforme o conhecimento oferecido de uma certa linguagem de input para uma representação interna utilizável, e utilize essa mesma informação de forma a integrar esta com a informação retida previamente de forma útil. O aprendiz tem de realizar algum trabalho, através de inferência, mas a maior fração do trabalho realizado continua do lado do professor, que apresenta e organiza o conhecimento de uma forma que aumenta incrementalmente o conhecimento do aluno.

3.1.3 Aprendizagem por analogia

Adquirir novos artefactos de conhecimento transformando e aumentando conhecimento existente que é similar àquele que é desejado no novo conceito da nova situação. Um sistema que aprende por analogia pode ser aplicado para converter um programa de computador existente e funcional, num que realiza funções similares para o qual não foi originalmente desenhado. Aprender por analogia requer mais inferência da parte do algoritmo do que a aprendizagem por implantação.

3.1.4 Aprendizagem por exemplos

Dado um conjunto de exemplos e contra-exemplos de um conceito, o algoritmo induz um conceito geral que descreve os exemplos positivos e nenhum dos contra-exemplos. É um conceito fortemente investigado na inteligência artificial. A quantidade de inferência feita pelo algoritmo é muito maior do que na técnica de aprendizagem por instruções, visto que não existem exemplos gerais a serem dados por uma entidade exterior, e é também uma inferência de alguma forma maior do que na técnica de aprendizagem por analogia, pois não existem conceitos gerais a serem dados como "sementes" dos quais novos conceitos podem surgir. Aprendizagem por exemplos pode ser sub-categorizada dependendo do tipo de fonte dos dados, nomeadamente:

- A fonte ser uma fonte externa segura, que conhece o conceito e gera sequências de exemplos que têm como objetivo ser o mais úteis possíveis ao processo de aprendizagem.

- A fonte ser o próprio algoritmo, normalmente este conhece o seu estado atual de conhecimento, mas não tem informação sobre o conceito objetivo a ser adquirido. Assim sendo o algoritmo pode gerar instâncias na informação básica que acredita serem necessárias para descriminar as diferentes descrições de contexto.

- A fonte ser um ambiente externo, neste caso o exemplo é gerado aleatoriamente, em que o algoritmo deve confiar em observações relativamente não controladas.

- Apenas exemplos positivos, em que exemplos positivos são providenciados como instância do conceito a ser adquiridos, estes têm um efeito negativo que é o facto de não providenciarem informação para prevenir generalizações excessivas. Neste tipo de aprendizagem, o excesso de generalização deve ser evitado de forma a considerar apenas a menor quantidade possível de generalizações, ou através de conhecimento fornecido a priori para inferir conceitos futuros.

- Serem fornecidos exemplos positivos e negativos. Neste tipo de situação os exemplos positivos forçam o aparecimento de generalizações, contudo a existência de exemplos negativos previnem o excesso de generalizações. Esta é a forma típica de aprendizagem por exemplos, sendo que a maioria dos sistemas de aprendizagem automática utiliza a técnica em questão.

Aprendizagem por exemplo pode ser realizada de uma só vez ou de forma incremental. A forma incremental aproxima-se da aprendizagem humana, que permite que o algoritmo utilize conceitos parcialmente aprendidos e permite à fonte de informação focar-se nos aspetos básicos de um novo conceito antes de se focar em aspetos menos centrais. Por outro lado, a aprendizagem realizada de uma só vez é menos apta a guiar para caminhos negativos, pois os exemplos iniciais que são fornecidos apresentam a totalidade dos casos positivos e negativos, pelo que o algoritmo consegue logo à partida fixar a sua base de conhecimento.[17]

3.1.5 Aprendizagem por observação e descoberta

Também chamado de *unsupervised learning*, este é uma forma bastante geral de induzir aprendizagem que inclui sistemas de descoberta, tarefas de formação de teoria e criação de critérios de classificação. Este tipo de aprendizagem obriga o algoritmo a realizar mais inferência do que qualquer tipo de abordagem discutida até agora. O algoritmo não tem qualquer tipo de conjunto de instância com conceitos particulares, nem é dado acesso a um elemento exterior que consegue classificar instância internamente geradas como instância positivas ou negativas para um dado conceito.

No entanto, no caso da técnica de aprendizagem por observação e descoberta existem dois tipos de observações possíveis de serem realizadas:

- **Observação passiva:** Onde o observador classifica múltiplos aspetos sobre o ambiente, sem alterar os mesmos.
- **Observação ativa:** Onde o observador perturba o ambiente para observar os resultados das suas perturbações. Á medida que um sistema vai adquirido conhecimento isto leva à confirmação ou negação das suas teorias, o que torna a exploração diferente à medida das necessidades e da evolução de execução do algoritmo.

3.2 Aprendizagem Automática Supervisionada

Cada instância de um conjunto de dados utilizado por algoritmos de aprendizagem automática é representado por um conjunto de atributos. Aos diferentes atributos podemos também atribuir o nome de métricas. As métricas podem ser continuas, categóricas ou binárias. Se as instâncias apresentarem etiquetas (*labels*), que correspondem aos corretos outputs ou classes às quais pertencem as instâncias, então estamos diante da utilização da técnica de aprendizagem automática supervisionada.

O objetivo deste tipo de aprendizagem automática passa por construir uma base de conhecimento através da análise dos dados em função das suas etiquetas. Desta forma, treinando os modelos de aprendizagem automática com um conjunto de dados de treino, é possível que este perceba de que modo os diferentes valores de cada métrica influencia na obtenção da respetiva etiqueta que classifica aquela instância. O que possibilita que o modelo consiga classificar um novo conjunto de dados prevendo as etiquetas de saída desse novo conjunto. [5]

A construção de um modelo de aprendizagem automática supervisionada necessita de algum cuidado no que toca à forma como o modelo retira conclusões sobre a informação do conjunto de dados. Existem por isso três categorias para classificar o tipo de modelo de aprendizagem automática obtido.[18]

- Pouco adaptado *(Underfitted)*, onde o modelo de aprendizagem não conseguiu retirar informação útil proveniente do conjunto de treino, desta forma, a atribuição da etiqueta às novas instâncias de dados não apresenta os melhores resultados. Isto acontece pois existiu uma falha no que toca à aprendizagem sobre as relações no conjunto de treino.
- Demasiado adaptado *Overfitted*, na qual o modelo de aprendizagem automática fez uma análise demasiado especifica ao conjunto de treino, resultando assim num modelo que depende em demasia dos dados em questão, o que faz com que a previsão de novas instâncias não seja a melhor, pelo simples facto destas não possuírem os exatos valores dos dados do conjunto de treino.

- Adaptação necessária *(Good Fit)*, quando foi possível evitar situações tanto de *overfitting* como de *underfitting*. Isto torna o modelo de aprendizagem automática mais robusto, adequando-se o necessário a novos dados mas conseguindo prever corretamente as respetivas etiquetas.

Figura 3.1: Representação visual dos diferentes tipos de adaptação aos dados (Adaptado de [3])

3.2.1 Problemas gerais da aprendizagem supervisionada

O primeiro passo é reunir um conjunto de dados e identificar quais as métricas que são mais relevantes. Como é possível que o conjunto de dados possua algum ruído no que toca à representação dos dados, bem como casos onde alguns valores são omissos, é necessário realizar um pré-processamento.

Tipos de dados que podem estar errados:

- Valores impossíveis ou duvidosos que tenham sido introduzidos manualmente ou incorretamente obtidos.
- Não ter sido introduzido qualquer valor (*missing value*).
- Valores irrelevantes estarem a ser apresentadas.

Se o ruído nos dados, por exemplo (*impossible values*), não poder ser resolvido de outra forma, como poder editar os dados de forma a que estes sejam reintroduzidos, então estes devem ser categorizados como *missing values* e deve-se simplesmente eliminar a informação do conjunto de dados de entrada.

Informação incompleta é um problema inevitável quando lidamos com informação real. Geralmente existem fatores importantes no que toca ao processamento de valores desconhecidos. Um dos mais importantes é a fonte dos valores desconhecidos:

- O valor é desconhecido porque foi esquecido ou perdido;
- Uma certa métrica não é aplicável numa dada instância;

- Para uma dada observação, o valor daquela métrica não importa para o treino do modelo.

Dependendo da circunstância, existem diferentes tipos de métodos nos quais podemos escolher para tratar os *missing values*, melhorando assim a qualidade geral do de conjunto de dados de entrada.

A seleção de um subconjunto de métricas é o processo de identificar e remover a maior quantidade de métricas redundantes e irrelevantes possível. Isto reduz a dimensionalidade do nosso conjunto de dados e permite que os algoritmos de *data mining* operem com maior rapidez e eficiência [15]. O facto de algumas métricas dependerem de outras induz problemas no que toca à exatidão de modelos de classificação supervisionada. Este problema pode ser abordado através da construção de novas métricas a partir do conjunto de métricas base.

A avaliação do classificador é usualmente obtida através da exatidão das previsões. Existem pelo menos três formas de classificar a exatidão do classificador. A primeira é dividir o conjunto de treino, utilizado dois terços para treino do modelo e um terço para estimar a performance. Outra técnica, conhecida como validação cruzada, é dividir o conjunto de treino em subconjuntos com a mesma dimensão e mutuamente exclusivos, sendo que para cada subconjunto o classificador é treinado na união de todos os outros subconjuntos. *Leave-one-out validation* é um caso especial da validação cruzada. Todos os subconjuntos são testados, este tipo de validação é mais dispendioso em temos de computação, mas útil quando é requerido que se tenha uma estimação da taxa de erro mais precisa [36].

3.2.2 Árvores de Decisão

As árvores decisão são árvores que classificam instâncias através da sua ordenação consoante os valores das suas métricas. Cada nó numa árvore de decisão representa uma métrica numa instância a ser classificada, sendo que cada vértice representa o valor que o nó (métrica alvo) pode assumir.

A métrica que melhor divide o conjunto de treino deve ser o nó raiz da árvore. Existem muitas maneiras de descobrir a métrica que melhor divide a árvore, e diversos estudos [17] acabaram por concluir que não existe um único "melhor método". O procedimento é depois repetido em cada partição do conjunto de dados, criando sub árvores até que o conjunto de treino esteja dividido em subconjuntos da mesma classe.

Existe dois tipos de abordagem que um algoritmo baseado numa árvore de decisão pode utilizar de forma a evitar *overfitting* dos dados durante o treino:

- Parar o algoritmo de treino antes que este faça *fit* ao conjunto de treino.
- Escolher entre diferentes árvores. Se duas árvores empregam o mesmo tipo de testes e têm a mesma exatidão de previsão, então normalmente a que tiver menos folhas é considerada preferencial para o treino.

3.2.3 Conjuntos de regras de aprendizagem

Árvores de decisão podem ser transformadas em conjuntos de regras através da criação de regras separadas para cada caminho da raiz da árvore até cada folha.[26] Contudo, regras também podem ser induzidas diretamente a partir do conjunto de treino, utilizando uma variedade de técnicas, como foi estudado por Quinlan. [26]

O objetivo é criar o menor conjunto de regras que é consistente com o modelo de dados de treino. Uma grande quantidade de regras aprendidas normalmente é indicador de que o algoritmo de aprendizagem com algum grau de *overfitting*.

A característica mais útil associada aos conjuntos de regras de aprendizagem prende-se com a sua compreensibilidade. Para a tarefa de aprendizagem de problemas binários, regras são mais compreensíveis do que árvores de decisão, pois normalmente as abordagens relacionadas com regras aprendem um conjunto de regras apenas para as classes positivas. Por outro lado, a definição para objetos de múltiplas classes continua por aprender, o algoritmo de regras deve ser executado separadamente para cada uma das classes separadamente no caso da classificação de múltiplas classes.

3.2.4 Redes neuronais

Outro tipo de algoritmo bem conhecido no que toca à aprendizagem automática são os algoritmos que têm por base a noção de perceptrão.

Um perceptrão é uma camada simples de uma rede neuronal, sendo que que ao conjunto de multiplas camadas de perceptrões se dá o nome de rede neuronal. Um perceptrão é um classificado linear (binário), que funciona de forma bastante simples: [31]

- Os valores de chegada são multiplicados por um peso W, definido nas propriedades do perceptrão.
- Os valores multiplicados são depois agregados num valor denominado de soma ponderada.
- Esta soma ponderada é depois aplicada a uma função de ativação. O exemplo mais simples de uma função de ativação é o caso da função *unit step*, na qual é retornado o valor 1 no caso do valor da soma ponderada ser positivo e 0 no caso da soma ponderada ter um valor negativo.

A forma mais comum de como estes algoritmos são utilizados é para aprenderem através de um conjunto de instâncias de treino correndo o algoritmo repetidamente através do conjunto de treino até que este encontre um vetor de predição que é correto para todo o conjunto de treino.

Peceptrões só conseguem classificar instâncias que sejam linearmente separáveis. Se uma linha reta ou plano poder ser desenhada para separar instâncias provenientes do input nas suas categorias corretas, então estas instâncias são linearmente separáveis e o

peceptrão poderá encontrar uma solução para o problema. Caso isto não aconteça então nunca haverá o ponto em que todas as instâncias são classificadas corretamente. As redes neuronais artificiais foram criadas com o intuito de resolver este problema.

Uma rede neuronal consiste num grande número de peceptrões juntos através de um padrão de conexões. Existem três tipos de classes, as unidades de input, que recebem a informação a ser processada, as unidades de output, onde os resultados do processamento são encontrados e as unidades escondidas que permanecem no meio destas. Geralmente a determinação do tamanho apropriado da camada escondida é um problema, porque uma estimativa baixa do número necessário pode levar a uma má aproximação e à generalização das capacidades, enquanto que um excesso de nós pode resultar em *overfitting* e eventualmente fazer a procura pela solução ótima mais difícil [17].

Uma rede neuronal depende de três coisas, do input, das funções de ativação e do peso atribuído a cada conexão de input. Sendo que os primeiros dois aspetos são fixos resta apenas controlar o comportamento da rede neuronal através da alteração dos pesos atribuídos.

Apesar de as redes neuronais e as árvores de decisão serem fundamentalmente diferentes, foi possível chegar a algumas conclusões gerais:

- As redes neuronais normalmente conseguem providenciar mais conhecimento incremental do que as árvores de decisão.
- O tempo de treino das redes neuronais é normalmente superior ao tempo de treino das árvores.
- As redes neuronais normalmente têm um desempenhado semelhante relativamente às árvores de decisão, mas raramente melhor.

3.2.5 Máquinas de Suporte Vetorial

Máquinas de Suporte Vetorial (SVMs) são uma técnica de aprendizagem automática supervisionada recentemente introduzida. As SVMs funcionam em redor da noção de margem, o lado de um hiperplano que separa duas classes de informação, maximizando a margem e as instâncias de cada lado deste.

Para os casos em que temos informação que seja linearmente separável, assim que o hiperplano seja encontrado os pontos que ficam na margem deste são conhecimentos como vetores de suporte e a solução é representada como uma combinação linear destes pontos. Sendo que outros pontos são ignorados. Por isto a complexidade de um modelo SVM não é afetado pelo número de métricas que o conjunto de treino possui [24]. Sendo este um aspeto bastante importante a considerar no momento da escolha do algoritmo a utilizar na construção de modelos de aprendizagem automática.

Do ponto de vista de eficiência computacional não é necessário passar a informação por um processo de seleção de métricas, enquanto que com técnicas de aprendizagem como as árvores de decisão ou redes neuronais começa a ser algo complicado obter bons

resultados se passamos a treinar conjuntos de dados com algumas centenas de métricas [20].

Se a SVM não conseguir achar a solução devido ao facto de o conjunto de entrada ter instâncias mal classificadas, o problema pode ser abordado através da suavização das margens, aceitando assim alguns erros de classificação para as instâncias de treino. No mundo real existe muita informação que não é linearmente separável, para isto não existe um hiperplano que consiga separar as instâncias positivas das negativas no conjunto de treino. Uma solução para esta limitação é o aumento da dimensionalidade e definição de um hiperplano separado.

No entanto os métodos SVM são binários, pelo que no caso de problemas com múltiplas classes deve-se reduzir o problema para um conjunto de múltiplos problemas de classificação binária, permitindo assim a classificação de várias classes distintas.

3.2.6 Comparação de técnicas de aprendizagem

Olhando para os diferentes atributos que compõem o conjunto de dados, estes podem ser denominados como métricas. As diferentes métricas são então analisadas e selecionadas de acordo com a possivel contribuição que estas podem oferecer na solução do problema.

Geralmente máquinas de suporte vetorial e redes neuronais tendem a ter um melhor desempenho no que toca a problemas com múltiplas dimensões e métricas continuas. Em contraste, sistemas baseados em lógica tendem a ter uma melhor performance quando se lida com métricas discretas ou categóricas.

No que toca à interpretação, algoritmos baseados em lógica são fáceis de interpretar, sendo que as redes neuronais e SVMs são notórios pela sua baixa interpretabilidade, este tipo de fatores pode ser preponderante na escolha do algoritmo a utilizar.

Não existe um algoritmo de aprendizagem que tenha uma melhor performance que todos os outros para todos os conjuntos de dados, pelo que para resolver a questão "Que algoritmos usar para ter um resultado com maior exatidão?" a melhor abordagem é estimar a exatidão dos algoritmos candidatos para o problema e selecionar aquele que apresente ter o melhor resultado. Esta avaliação é algo que terá de ser efetuado para cada conjunto de dados e dependendo do objetivo do problema. Pelo que diferentes problemas terão diferentes conclusões no que toca ao algoritmo que permite melhores resultados.

O conceito de combinar classificadores é proposto como uma nova maneira de melhorar o desempenho dos classificadores individualmente. O objetivo dos algoritmos de classificação é gerar resultados com maior precisão e exatidão [19]. Sendo que é útil utilizar os pontos fortes de um algoritmo para complementar os pontos fracos de outro, melhorando assim a solução final.

3.3 Aprendizagem Automática aplicada à modelação de utilizadores

Quando falamos de modelação dos utilizadores e das suas ações estas podem ser tão variadas tanto os diferentes propósitos para os quais os modelos de utilizadores são formados. É possível estar na presença de diferentes tipos de análise, nomeadamente:

- Os processos cognitivos que levaram os utilizadores a realizar certas ações;
- As diferenças entre as capacidades do utilizador e as capacidades de um *expert*;
- Padrões comportamentais de um utilizador ou as suas preferências;
- Características do utilizador.

Estes modelos podem conter informação pessoal relativa ao utilizador e informação que é depois adicionada ao utilizador que pode ou não estar diretamente ligada à adaptação do sistema a esse utilizador, mas que pode ser utilizada para categorizar o utilizador num de vários estereótipos, o que torna possível ao sistema antecipar alguns dos comportamentos desse utilizador [30].

Existem até sistema educacionais que utilizam técnicas de modelação para personalizar o processo de aprendizagem, de forma a tornar o processo mais adaptativo às capacidades dos alunos, tendo em conta as suas habilidades e o seu contexto histórico, bem como prever ações dos estudantes. Isto permite proporcionar um ensino mais personalizado [13, 34].

Métodos de aprendizagem automática têm sido aplicados a problemas de modelos de utilizadores principalmente para construir modelos de utilizadores individuais aque interagem com sistemas de informação [2, 4, 27, 28].

Uma dimensão bastante importante é a necessidade de distinguir as abordagens que dizem respeito a se o modelo é direcionado para os utilizadores individuais ou para comunidades (grupos) de utilizadores [33].

Embora pesquisa académica relacionada com aprendizagem automática se concentre sobre a modelação individual dos utilizadores [2, 4, 27, 28, 33], estão a emergir aplicações baseadas em aprendizagem automática no que toca ao comercio eletrónico que relacionam a formação de modelos genéricos de comunidades de utilizadores.

3.3.1 Personalização de informação

O objetivo da personalização de informação é poder proporcional ao utilizador aquilo que este quer ou necessita sem que este tenha de pedir explicitamente. [23]

Neste tipo de sistema existe a distinção entre as informações dos utilizadores e dos objetos.

- **Informação relativa aos objetos:** Inclui conteúdo descritivo sobre os objetos ou produtos que estão em estudo para serem recomendados.

- **Informação relativa aos utilizadores:** Inclui referências passadas dos dados de utilização de serviços ou de *ratings* passados atribuídos pelo utilizador, bem como dados pessoais ou demográficos.

Existem também diferences tipos de personalização de informação relativamente ao tipo de *feedback* que utilizam por parte dos utilizadores, nomeadamente abordagens reativas e pro-ativas. Abordagens reativas utilizam processos convencionais que requerem que o utilizador realize interações explicitas, quer na forma de interrogações (do inglês *queries*) quer na forma de *feedback* incorporado nos processo de recomendação [1]. Em sistemas de *feedback* baseados em *ratings*, os utilizadores têm de classificar todas as recomendações que são feitas, de acordo com o quanto estas se adequam àquilo que estes procuram. No que toca aos sistemas que utilizam *feedback* de preferência, ao utilizador é apresentada uma lista com recomendações e este é motivado a escolher apenas um produto, sendo este aquele que mais se adequa aos seus requisitos [6, 11].

Outro tipo de abordagem é a abordagem pro-ativa, que aprende sobre as preferências do utilizador e recomenda objetos com base na informação aprendida, não necessitando que o utilizador introduza o seu *feedback* de forma explicita [1].

3.3.2 Desafios

O treino de modelos de aprendizagem automática tem alguns desafios, identificados e discutidos em baixo:

Sistemas como o de Syskill e Webert [25], que era aplicado ao problema de recomendação de sites web, em que os utilizadores à medida que navegam na Internet dão o seu *feedback* sobre as paginas web visitadas, carregando num botão de aprovação ou desaprovação, apresentam uma limitação inerente a este tipo de aplicações de aprendizagem automática, é impossível construir um modelo com uma exatidão aceitável a não ser que se tenha um número relativamente grande de exemplos para ter como base. É por isso normal que um algoritmo de aprendizagem necessite de diferentes exemplos de treino [32].

Outro desafio direto da aplicação de aprendizagem automática a muitas tarefas de modelos de utilizadores é que as abordagens através de aprendizagem supervisionada necessitam que exista uma etiqueta (*label*), ou um rótulo explicito para a informação recebida, contudo este rótulo pode não ser atribuído corretamente através da observação do comportamento do utilizador.

Por exemplo, no sistema de Syskill e Webert era necessário que o utilizador realizasse tarefas extra de forma a que fosse possível ter estas etiquetas preenchidas, contudo isto pode ser um problema, pois a maioria dos utilizadores não realiza este tipo de tarefas

extra se não existir uma necessidade de realizar as mesmas ou se não existir um incentivo associado à sua realização.

De forma a dar a volta a este problema, uma das possíveis soluções é inferir as *labels* através do comportamento dos utilizadores, verificando se estes realizam um dado conjunto de ações, e conforme as ações realizadas assim seria atribuída uma das *labels*. Outra forma de resolver o problema passa por utilizar um pequeno conjunto de dados já catalogados com as respetivas *labels* para assim conseguir inferir a *label* correta num conjunto maior de dados, que serão depois utilizados para treinar os algoritmos de aprendizagem.

A modelação de utilizadores é algo que é muito suscetível a mudanças ao longo do tempo, uma vez que os atributos que caracterizam os utilizadores têm tendência a variarem ao longo do tempo. Assim sendo é necessário que os algoritmos de aprendizagem sejam capazes de se ajustar a estas mudanças com alguma rapidez. Este fenómeno, do ponto de vista do aprendizagem automática, tem o nome de *concept drift*. [35]

A ideia central de ajuste ao *concept drift* passa pela utilização de uma janela ajustável, onde o tamanho da janela depende dos indicadores observados, como certas mudanças em temos de distribuição. Algumas soluções, como a de Chiu e Webb [7], estudaram a indução de um duplo modelo de utilizadores, sendo que é fácil concluir que informação mais recente sobre os utilizadores espelha melhor o atual conhecimento, preferências ou estado atual de um utilizador do que informação de tempos passados. Contudo esta informação mais recente pode levar a modelos demasiado específicos. De forma a contornar este problema, Chiu e Webb utilizaram este modelo com dupla informação, em que primeiramente os algoritmos de aprendizagem utilizam as informações mais recentes para o seu treino, contudo, no caso da previsão não ter sido obtida com suficiente confiança são então consultados os dados mais antigos [7] de forma a evitar que modelo formado seja demasiado específico.

O crescimento da Internet tem tido um tremendo impacto no que toca ao aprendizagem automática aplicado à modelação de utilizadores. Se por um lado a Internet tem levado a um novo caminho com novas oportunidades para assistir os utilizadores com diferentes serviços, através da utilização dos detalhes de utilização destes, por outro lado o aumento da informação disponível, bem como o aumento do número de utilizadores online criou novos desafios relacionados com a complexidade computacional. Investigação realizada num ambiente académico mostram que existe uma preocupação relativa à complexidade computacional[33]. Quando um novo algoritmo é proposto, é comum que sejam feitas diferentes avaliações da forma como este de comporta em diferentes situações de processamento, considerando sempre a sua taxa de exatidão. Por esta razão não é errado que um algoritmo que apresente uma exatidão de 78% seja escolhido na vez de um algoritmo que garanta 80% de exatidão, pois provavelmente o tempo de execução deste segundo é consideravelmente mais elevado e como tal, impede um melhor funcionamento e processamento dos dados de um cenário com dados do mundo real e em que a latênciado processamento é relevante.

3.4 Sistemas de filtragem de informação

De forma a que os modelos de aprendizagem automática possam sugerir aos utilizadores certos itens e fazer a filtragem da informação é necessário utilizar diferentes métodos de filtragem consoante o objetivo do problema. Podemos por isso dividir o tipo de método a utilizar em dois tipos. O primeiro tipo é uma filtragem baseada no conteúdo (*content-based*), na qual o sistema aceita a informação que descreve os vários objetos, e aprende a prever quais os objetos que se enquadram no modelo de utilizador. Este tipo de filtragem só pode ser aplicada quando existem itens cujas propriedades e valores dos atributos sejam descritos. O outro tipo é a filtragem colaborativa ou social (*collaborative-based)*, na qual o sistema atualiza o modelo de utilizador e prevê os itens que se adequam a este, baseado no *feedback* obtido por parte dos diferentes utilizadores. Numa análise colaborativa um objeto é considerado interessante para um utilizador se, para outros utilizadores com gostos semelhantes, aquele objeto também era de interesse. A preferência com base na análise colaborativa tem sido utilizada em sistemas especialistas em filtragem e retorno de informação [8].

Contudo a abordagem *content-based* e *collaborative-based* não se excluem uma à outra, de facto estas podem ser combinadas de forma a criar aquilo que se chama um modelo de utilizador de preferências integradas. Em particular este tipo de modelo é construído nos termos de um conjunto de atributos predefinidos, tal como no caso dos modelos de preferência dos utilizadores *content-based*, contudo existem dois atributos nesta abordagem integrada que se caracterizam por uma abordagem *collaborative-based*, na qual um dos atributos caracteriza o utilizador e o outro caracteriza um objeto[8].

De forma a que seja possível construir automaticamente um modelo de utilizador de preferências integradas, um método de aprendizagem indutiva é aplicado a um conjunto de entradas do conjunto de dados [2].

Existe uma grande quantidade de informação que é criada e enviada através de meios de comunicação eletrónica. Estas grandes quantidades de informação revelam-se difíceis de gerar valor se não forem selecionadas, tratadas e filtradas de acordo com um objetivo específico. Desta forma, existem processos de filtragem de informação que permitem utilizar a informação proveniente dos diferentes serviços de forma a que seja gerado valor acrescentado a esta. Um sistema de filtragem de informação normalmente inclui quatro componentes básicos: (a) um componente de análise de informação, (b) um componente de filtragem, (c) um componente de modelo de utilizador e (d) um componente de aprendizagem.

- O componente de análise da informação (a) obtém e guarda objetos de informação junto dos provedores dos dados. A informação é analisada e apresentada no formato apropriado para o problema. Esta representação será o input do componente de filtragem (b).

- O componente do modelo de utilizador (c) quer implicitamente e/ou explicitamente

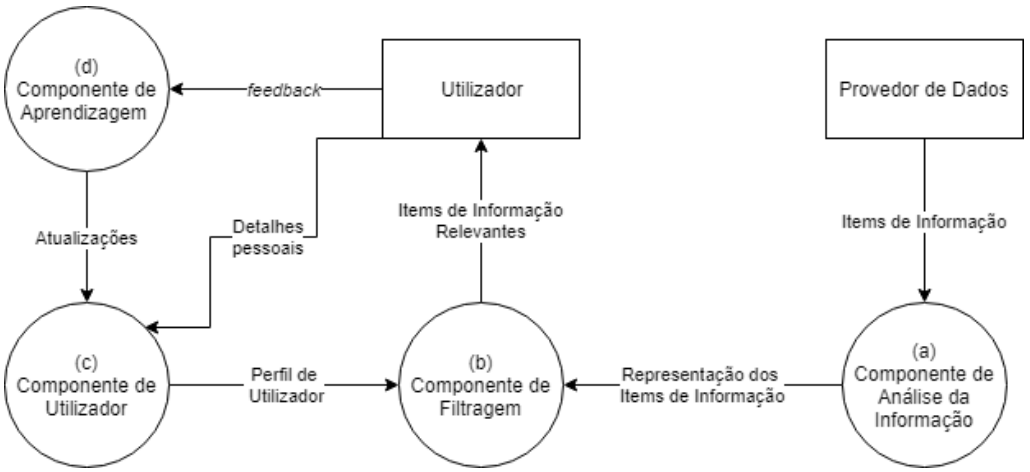

Figura 3.2: Modelo genérico de filtragem de informação

recolhe informação sobre os utilizadores e as suas necessidades e constrói um modelo de utilizadores. O modelo de utilizador será também input para o componente de filtragem.

- O componente de filtragem (b), que é o coração do sistema de filtragem de informação, faz a correspondência entre os perfis dos utilizadores e os objetos de informação representados e decide se um dado objeto na coleção é relevante para o utilizador.
- O componente de aprendizagem (d) é necessário para que se melhore a filtragem. Devido à dificuldade em modelar os utilizadores e devido também às alterações nas suas informações, os sistemas de filtragem necessitam que os processos de aprendizagem detetem estas mudanças nos interesses dos utilizadores. Caso contrário existiriam incoerências que iriam afetar os resultados da filtragem, deturpando os resultados finais.

3.5 Sumário

Neste capitulo foram clarificados alguns conceitos base que serão utilizados no decorrer da concretização dos trabalhos. Passando pelos diferentes tipos de aprendizagem automática, bem como a elucidação do tratamento que a informação do conjunto de dados deve receber antes que esta possa ser utilizada para treino de modelos de aprendizagem automática. De forma mais profunda foi debatido o tema da aprendizagem automática de tipo supervisionado, uma vez que este será o tipo de aprendizagem utilizado para dar resposta às necessidades da empresa CreateIT.

Por ultimo, ainda relativo ao trabalho relacionado, é também demonstrado como a aprendizagem automática pode ser utilizada no contexto da modelação de utilizadores, de forma a filtrar, compreender e assimilar características dos diferentes utilizadores, com o objetivo a melhorar um serviço ou produto, aumentando o nível de personalização do mesmo.

No próximo capitulo é demonstrado o ponto inicial dos trabalhos propostos bem como o tipo de tecnologias que irão estar na base do desenvolvimento de uma solução. São também enumeradas as diferentes tarefas que compõem a proposta de trabalho, nomeadamente o desenho e desenvolvimento do sistema e serviço que permite a criação de modelos de aprendizagem automática, passando pelos respetivos testes e aperfeiçoamentos possíveis de serem feitos a estes modelos e a avaliação dos resultados obtidos.

De forma a dar uma visão mais concreta de como o trabalho irá estar dividido no tempo disponível para o projeto, é também apresentado um diagrama de Gantt na qual consta a divisão das várias tarefas necessárias à elaboração dos trabalhos bem como o tempo estimado para a realização das mesmas.

Capítulo 4

Trabalho Proposto

4.1 Levantamento da situação inicial

De forma a perceber o ponto de situação inicial foi realizada uma inspeção sobre as atuais capacidades da plataforma em questão, nomeadamente o CloudCockpit, bem como o tipo de dados que se dispunha antes de iniciar qualquer trabalho na solução pretendida. Aferindo assim quais os pontos fortes do sistema atual e quais as suas limitações.

De acordo com aquilo que já foi enumerado na secção 1.1 e no capítulo 2 relativamente à plataforma CloudCockpit, esta permite aceder a diferentes dados de utilização dos clientes relativos a diferentes ferramentas de diferentes pacotes. No entanto, neste momento a plataforma não possui qualquer mecanismo de classificação de utilizadores com base nos seus dados, nem de previsão de resultados ou comportamentos. Trata-se apenas de uma plataforma que permite mostrar aos diferentes utilizadores dos serviços os diferentes preços, níveis de utilização e dados de faturação de acordo com os seus objetivos. Neste sentido é necessário desenvolver uma solução que possibilite a aplicação de algoritmos de aprendizagem automática aos dados tratados da CloudCockpit por forma a dar respostas às necessidades existentes no momento.

No que toca à existência de um repositório funcional, neste momento não existe nenhum que esteja dedicado ao problema em questão, assim sendo, é necessário criar de raiz um repositório que esteja arquitetado de forma a alimentar da melhor maneira possível os métodos de aprendizagem automática. Garantindo que toda a informação não necessária não consta neste repositório e que todas as métricas utilizadas são de facto as que permitem a obtenção de modelos de aprendizagem automática com os melhores resultados, satisfazendo os desafios da implementação.

4.2 Estudo e aprendizagem das tecnologias envolvidas

Numa fase inicial do projeto é objetivo do mesmo que exista uma aprendizagem relativa às diferentes tecnologias e plataformas a utilizar.

Os diferentes conjuntos de dados relativos aos clientes e aos seus dados de utilização dos diferentes serviços encontram-se acessíveis através de repositórios, neste sentido, a utilização da plataforma Microsoft SQL Server é fundamental para aceder os registos quer dos utilizadores propriamente ditos como dos diferentes pacotes e serviços que os compõem.

Numa primeira fase é prevista a exploração e extração de conceitos úteis à utilização da plataforma, sendo que numa segunda fase será necessário utilizar a mesma de forma a construir um repositório que será a base de aplicação para o trabalho futuro. Este repositório a ser desenvolvido irá conter o conjunto de dados do qual os algoritmos de aprendizagem automática irão extrair informação, de forma a compreender e conseguir deduzir informação necessária para realizar previsões o mais acertadas possível. Neste sentido, é previsto que um dos contributos para o projeto seja a construção de um repositório funcional e documentado que permita o armazenamento dos dados ao longo do tempo de forma a poder ser utilizado de forma eficaz para a previsão de dados.

Após construção do repositório é necessário que exista uma exploração de avaliação da melhor plataforma a utilizar para o treino de modelos de aprendizagem automática. Uma vez que a empresa em questão é parceira da Microsoft e da Amazon existem duas plataformas *cloud* sugeridas ao desenvolvimento desta etapa. Neste sentido é objetivo do trabalho realizar um estudo pormenorizado tanto sobre a plataforma Azure Machine Learning Studio como para a plataforma Amazon Web Services, Machine Learning.

No sentido de perceber qual das duas plataformas poderá originar melhores resultados é necessário um estudo e documentação dos diferentes aspetos que cada plataforma proporciona, bem como o seu grau de fiabilidade e estado atual de desenvolvimento das ferramentas disponíveis em ambas as plataformas. Após perceber quais os pontos fortes e limitações de cada uma das plataformas é necessário escolher, justificadamente, uma das duas plataformas de forma a que seja possível começar o treino de modelos de aprendizagem automática que garantam os melhores resultados possíveis.

No que toca à visualização de resultados, embora seja possível concluir sobre os mesmos diretamente nas plataformas *cloud* poderá ser mais útil e mais intuito utilizar plataformas especialistas em visualização de informação como o Tableau Software, de forma a que a visualização desta grande quantidade de dados possa ser feita de forma fácil e intuitiva a qualquer utilizador. Neste sentido, é suposto existir mais uma vez um período de estudo e documentação do desempenho da plataforma e daquilo que é possível alcançar com a mesma no contexto do problema.

4.3 Desenho e desenvolvimento do sistema e serviço

No que toca à construção do repositório de dados que servirá de base ao desenvolvimento do resto do projeto, o desenho deste partirá da análise detalhada de cada uma das métricas presentes no mesmo. O repositório contará apenas com as métricas que acrescem valor ao conjunto de dados no contexto do problema. Para isto será necessário uma análise algo aprofundada dos dados e a execução de técnicas de seleção de métricas (*feature selection*), por forma a selecionar quais as métricas que devem ser mantidas e quais as métricas que devem de ser descartadas do conjunto de dados. Será assim necessário estudar e testar diferentes tipos de algoritmos de seleção de métricas, analisando que tipos de combinações de métricas proporcionam os melhores resultados.

Assim que o repositório estiver construído é possível avançar no projeto, passando assim para uma plataforma dedicada à aprendizagem automática, tal como mencionado na secção 4.2, é necessário fazer a escolha entre a plataforma da Microsoft e da Amazon. Após a escolha da plataforma será possível fazer uma análise de quais os algoritmos que mais se adequam ao problema, é esperado que deste passo surja a seleção apropriada bem como os vários cenários de teste que permitiram a respetiva dedução, permitindo assim a criação de um modelo de aprendizagem automática que classifique as entidades corretamente, dado o que este foi criado com um conjunto de treino constituído por métricas chave.

Dada a criação dos modelos é então possível passar à sua aplicação, juntamente com a análise de resultados correspondente, permitindo assim começar a prever os comportamento corretos dos diferentes clientes face às suas necessidades e mudanças. Sendo este o objetivo principal do projeto, conseguir prever ações e necessidades dos clientes de forma a que seja possível ter um acompanhamento mais próximo e personalizado para cada cliente, é previsto que a maior parte do tempo de desenvolvimento seja dedicado a esta fase, de forma a garantir que se obtêm bons resultados no contexto do problema, conseguindo prever com a maior exatidão possível situações tanto de desistências de assinatura como de possível melhoramento de serviço.

4.4 Testes e aperfeiçoamentos

No que toca ao trabalho a realizar neste âmbito, é necessário garantir que a solução conseguida garante resultados promissores. Conseguindo alertar os revendedores de serviços *cloud* para potenciais situações de intervenção junto dos clientes, quer estas sejam devidas a uma diminuição de utilização dos serviços, quer estas sejam devido a limitações existentes devido à assinatura possuída naquele momento. Neste sentido é necessário certificar a qualidade do trabalho desenvolvido ao longo do processo.

No que toca à certificação da primeira fase, o repositório, é necessário garantir que este não apresenta erros de qualquer tipo, nem erros que dizem respeito aos dados específicos, como *missing values* que deturpem as leituras de dados, ou até mesmo erros de construção

no que diz respeito às métricas utilizadas ou à forma como os dados são inseridos no repositório.

Apesar dos algoritmos utilizados para a criação de modelos de aprendizagem automática terem sido escolhidos de acordo com as suas propriedades em função do objetivo final do projeto, as plataformas *cloud* de aprendizagem automática permitem realizar pequenas modificações nos algoritmos. Através da alteração de pequenas propriedades/-definições dos algoritmos é possível alterar o *output* dos mesmos, permitindo assim um *tuning* dos algoritmos especificamente para o problema em questão, melhorando tanto quanto possível os resultados finais.

4.5 Avaliação dos resultados obtidos

De forma a garantir a qualidade dos resultados obtidos é necessário avaliar os modelos que dão origem aos resultados. De forma a classificar os modelos de aprendizagem automática é necessário reter os conceitos de falso negativo, falso positivo, verdadeiro positivo e verdadeiro negativo.

No contexto do problema, um caso de verdadeiro positivo é quando o modelo de previsão prevê uma dada instância como sendo da classe positiva e essa é de facto a classe correta. Um verdadeiro negativo acontece quando é previsto que uma instância pertença à classe negativa e essa é de facto a classe dessa instância. Estamos por isso diante de de previsões acertadas.[10]

Relativamente ao caso de um falso positivo, isto acontece quando o modelo classifica uma dada instância como pertencendo à classe positiva contudo esta previsão está errada, uma vez que a classe correta seria a negativa. Analogamente o mesmo ocorre no caso do falso negativo.[10]

Desta forma, é possível fazer-se das seguintes métricas para medir a qualidade dos resultados obtidos:

- *Recall*, que representa o total de verdadeiros positivos sobre a soma dos verdadeiros positivos com os falsos negativos, representando assim uma proporção de todos os casos realmente positivos;

- *Precision*, número de exemplos classificados como pertencentes a uma classe, que realmente são daquela classe (positivos verdadeiros), dividido pela soma entre este número e o número de exemplos classificados nesta classe, mas que pertencem a outras (falsos positivos) [12].

- *Accuracy*, obtida através da divisão dos número correto de previsões sobre o número total de previsões, o que permite ter uma perceção geral da quantidade previsões acertadas pelo modelo em questão [9];

- Média do erro quadrático, esta função calcula a média dos erros do modelo ao quadrado. Ou seja, diferenças menores têm menos importância, enquanto diferenças maiores recebem um maior peso [12];
- Média do erro absoluto, contrariamente à média do erro quadrático, em vez de elevar a diferença entre a previsão do modelo e o valor real ao quadrado, esta métrica toma o valor absoluto. Neste caso, em vez de atribuir um peso de acordo com a magnitude da diferença, atribui-se o mesmo peso a todas as diferenças, de maneira linear [12];
- Classificação F1, esta classificação é a média harmónica entre a precisão de um modelo e o *Recall*, bastante útil quando um conjunto de dados possui classes desproporcionais e o modelo não emite probabilidades;
- Área por baixo da curva de aprendizagem (Característica de Operação do Recetor), num gráfico que mede a taxa de exemplos positivos em função da taxa de falsos positivos, é vista a área que fica por baixo da curva, quanto maior esta for melhor o modelo de classificação;

A avaliação destas métricas serve de modo a conseguir comparar os diferentes modelos e conseguir avaliar o desempenhado particular de cada um deles, possibilitando assim chegar a conclusões sobre que tipo de algoritmo melhor se aplica no contexto do problema, bem como até que ponto o *tuning* dos algoritmos permite alterar os resultados finais. Este tipo de avaliação terá um papel preponderante na fase de desenvolvimento de modelos de aprendizagem automática.

4.6 Planeamento

Nesta secção é apresentado o plano geral de trabalhos. A ordem de trabalhos foi dividida em três fases, correspondentes às fases necessárias na construção da solução ao problema. A Tabela 4.1 demonstra o tempo planeado para a realização de cada etapa bem como as diferentes atividades que compõem cada fase.

4.7 Calendarização

Relativamente à calendarização, recorrendo a um diagrama de Gantt foi possível realizar a divisão das diferentes tarefas como é possível verificar na Figura 4.1.

Tarefa	Data de Inicio	Data de Fim	Semanas
Preparação	4 Fevereiro	15 Março	6
Análise e preparação de dados			
Construção do repositório			
Seleção dos Algoritmos			
Desenvolvimento	18 Março	10 Maio	8
Aplicação dos Algoritmos			
Treino de Modelos			
Análise dos Modelos			
Documentação	13 Maio	20 Setembro	15
Análise de Resultados			
Escrita de Documentação			
Escrita da Dissertação			
Preparação para a Defesa			

Tabela 4.1: Tabela demonstrativa das tarefas a realizar

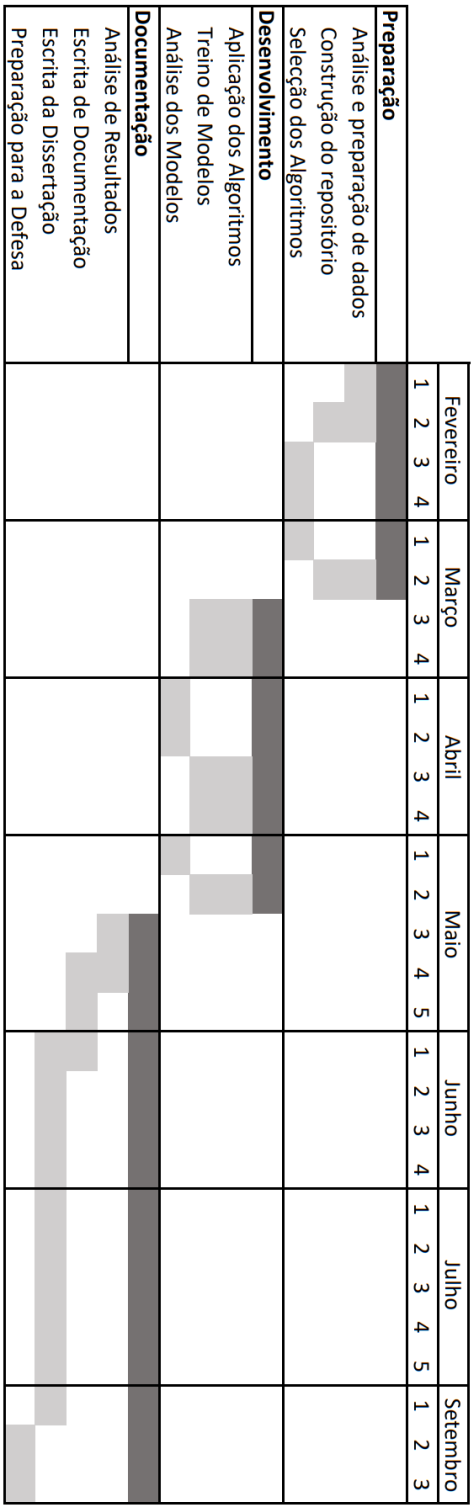

Figura 4.1: Calendarização do trabalho

Bibliografia

[1] S. S. Anand e B. Mobasher. "Intelligent techniques for web personalization". Em: *Proceedings of the 2003 international conference on Intelligent Techniques for Web Personalization*. Springer-Verlag. 2003, pp. 1–36.

[2] C. Basu, H. Hirsh, W. Cohen et al. "Recommendation as classification: Using social and content-based information in recommendation". Em: *Aaai/iaai*. 1998, pp. 714–720.

[3] A. Bhande. *What is underfitting and overfitting in machine learning and how to deal with it*. Last accessed 20 February 2019. 2016. URL: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76.

[4] E. Bloedorn, I. Mani e T. R. MacMillan. "Machine learning of user profiles: Representational issues". Em: *arXiv preprint cmp-lg/9712002* (1997).

[5] J. Brownlee. *Supervised and Unsupervised Machine Learning Algorithms*. Last accessed 20 February 2019. 2016. URL: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/.

[6] R. D. Burke, K. J. Hammond, V. Kulyukin, S. L. Lytinen, N. Tomuro e S. Schoenberg. "Question answering from frequently asked question files: Experiences with the faq finder system". Em: *AI magazine* 18.2 (1997), p. 57.

[7] B. C. Chiu e G. I. Webb. "Using decision trees for agent modeling: improving prediction performance". Em: *User Modeling and User-Adapted Interaction* 8.1-2 (1998), pp. 131–152.

[8] M. Dastani, N. Jacobs, C. M. Jonker e J. Treur. "Modeling user preferences and mediating agents in electronic commerce". Em: *Agent Mediated Electronic Commerce*. Springer, 2001, pp. 163–193.

[9] G. Developers. *Classification: Accuracy*. Last accessed 20 February 2019. 2018. URL: https://developers.google.com/machine-learning/crash-course/classification/accuracy.

[10] G. Developers. *Classification: True vs. False and Positive vs. Negative*. Last accessed 20 February 2019. 2018. URL: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative.

[11] D. R. Fesenmaier, F. Ricci, E. Schaumlechner, K. Wöber, C. Zanella et al. *DIETORECS: Travel advisory for multiple decision styles*. na, 2003.

[12] M. Filho. *As Métricas Mais Populares para Avaliar Modelos de Machine Learning*. Last accessed 20 February 2019. 2018. URL: http://mariofilho.com/as-metricas-mais-populares-para-avaliar-modelos-de-machine-learning/.

[13] P. Giangrandi e C. Tasso. "Managing temporal knowledge in student modeling". Em: *User Modeling*. Springer. 1997, pp. 415–426.

[14] N. Guerra. Private Communication. CreateIT, Lisbon, Portugal, 2019.

[15] M. A. Hall. "Correlation-based feature selection for machine learning". Em: (1999).

[16] S. Ingrez. Private Communication. CreateIT, Lisbon, Portugal, 2019.

[17] R. D.-K. J.Breuker. *Frontiers in Artificial Intelligence and Applications*. Vol. 160. IOS Press.

[18] W. Koehrsen. *Overfitting vs. Underfitting: A Conceptual Explanation*. Last accessed 20 February 2019. 2018. URL: https://towardsdatascience.com/overfitting-vs-underfitting-a-conceptual-explanation-d94ee20ca7f9.

[19] S. B. Kotsiantis, I Zaharakis e P Pintelas. "Supervised machine learning: A review of classification techniques". Em: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[20] E. Leopold e J. Kindermann. "Text categorization with support vector machines. How to represent texts in input space?" Em: *Machine Learning* 46.1-3 (2002), pp. 423–444.

[21] Microsoft. *What is the Cloud Solution Provider (CSP) program?* Last accessed 21 January 2019. 2017. URL: https://www.microsoftpartnercommunity.com/t5/Partnership-101/What-is-the-Cloud-Solution-Provider-CSP-program/td-p/2453.

[22] J. B. Morin. *What's the Difference Between a Direct and an Indirect CSP?* Last accessed 20 February 2019. 2017. URL: https://www.sherweb.com/blog/direct-indirect-csp-difference/.

[23] M. D. Mulvenna, S. S. Anand e A. G. Büchner. "Personalization on the Net using Web mining: introduction". Em: *Communications of the ACM* 43.8 (2000), pp. 122–125.

[24] B. Pang, L. Lee e S. Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques". Em: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics. 2002, pp. 79–86.

[25] C. Papatheodorou. "Machine learning in user modeling". Em: *Advanced Course on Artificial Intelligence*. Springer. 1999, pp. 286–294.

[26] J. R. Quinlan. "Simplifying decision trees". Em: *International journal of man-machine studies* 27.3 (1987), pp. 221–234.

[27] B. Raskutti e A. Beitz. "Acquiring user preferences for information filtering in interactive multi-media services". Em: *Pacific Rim International Conference on Artificial Intelligence*. Springer. 1996, pp. 47–58.

[28] P. Resnick e H. R. Varian. "Recommender systems". Em: *Communications of the ACM* 40.3 (1997), pp. 56–58.

[29] R. Ribeiro. Private Communication. CreateIT, Lisbon, Portugal, 2019.

[30] E. Rich. "Users are individuals: individualizing user models". Em: *International journal of human-computer studies* 51.2 (1999), pp. 323–338.

[31] S. SHARMA. *What the Hell is Perceptron?* Last accessed 20 February 2019. 2017. URL: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53.

[32] L. G. Valiant. "A theory of the learnable". Em: *Communications of the ACM* 27.11 (1984), pp. 1134–1142.

[33] G. I. Webb, M. J. Pazzani e D. Billsus. "Machine learning for user modeling". Em: *User modeling and user-adapted interaction* 11.1-2 (2001), pp. 19–29.

[34] G. Weber. "Adaptive learning systems in the World Wide Web". Em: *UM99 User Modeling*. Springer, 1999, pp. 371–377.

[35] G. Widmer e M. Kubat. "Learning in the presence of concept drift and hidden contexts". Em: *Machine learning* 23.1 (1996), pp. 69–101.

[36] X. Zhu, Z. Ghahramani e J. D. Lafferty. "Semi-supervised learning using gaussian fields and harmonic functions". Em: *Proceedings of the 20th International conference on Machine learning (ICML-03)*. 2003, pp. 912–919.