INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Previsão de assistencias em estádios de futebol

ARTUR MANUEL ROSA BENTO
Outubro de 2017

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Previsão de assistências em estádios de futebol

Artur Manuel Rosa Bento

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas computacionais

Orientador: Ana Maria Dias Madureira Pereira

Co-orientador: Bruno Cunha

Porto, outubro de 2017

Resumo

O presente documento tem como principal objetivo documentar todo o trabalho desenvolvido durante o projeto, realizado na empresa Flipkick.

A assistência em eventos desportivos é um aspeto bastante importante para os clubes. Se estes conseguirem prever se os seus adeptos vão assistir ao próximo jogo seria uma informação preciosa, da forma que poderão trabalhar sobre estes adeptos, através de estratégias de *markting* de modo a cativa-los para os seus estádios.

Neste contexto, surge a oportunidade de desenvolver uma aplicação capaz de efetuar tal previsão. Passando assim ao estudo das temáticas necessárias para o desenvolvimento e posteriormente implementar.

Para trabalhar com grandes quantidades de informação, é necessário recorrer a metodologias que nos permitam extrair informação, como a Aprendizagem Automática que permite realizar cálculos matemáticos de forma a realizar previsões de informação futura.

Os testes computacionais realizados demonstram que os fatores considerados para a execução de cálculos do algoritmo tem níveis de acerto mais elevados, de acordo com a quantidade utilizada para o efeito, revelando assim que mais fatores apresentam previsões mais acertadas, aproximando-se mais da realidade. O algoritmo que mais consistência demonstrou foi o `Two-Class Support Vector Machine`, apresentando percentagens de acerto mais elevadas, não dependendo este dos fatores utilizados.

A capacidade de realizar previsões do trabalho desenvolvido irá contribuir para que os clubes aumentem a assistência nos seus estádios, através da informação extraída pelo modelo de *Machine Learning* desenvolvido.

Palavras-chave: Aprendizagem Automática, Previsão, Percentagem de acerto.

Abstract

The main goal of this document is documenting all developed work during the project, realized in the company Flipkick.

Attendace at sporting events is a very importante aspect for clubs. If they can forecast when their fans will attend the next game it would be a precious information so they can work on these suportes captivate them for their stadiums through markting strategies.

In this contexto the oportunity arises to develop na application capable of making such forecast, starting study of necessary themes for development and later implemente.

In order to work with large amounts of information, it is necessary to resort some methodologies that allow us to extract information such as Machine Learning that allows to perform mathematical calculations in order to make forecasts of future information.

The computational tests carried out show that the factors considered for the execution of algorithm calculations have higher levels of success rate according to the quantity used for the effect revealing that more factors present better predictions approaching to the reality. The algorithm that presented the highest consistency was the Two-Class Support Vector Machine, presenting higher percentages of accuracy, not depending on the factors used.

The ability to carry out forecasts of this work will help clubs increase attendance at their stadiums through information derived from the developed Machine Learning model.

Keywords: Machine Learning, Forecast, Accuracy.

Agradecimentos

O desenvolvimento deste projeto não seria o mesmo sem a ajuda e o suporte dados por todos aqueles que de certa forma influenciaram o desenrolar deste trabalho de mestrado.

Gostaria então de agradecer à Dr. Ana Madureira por todo o apoio, motivação, orientação e disponibilidade. Gostaria também de agradecer ao coorientador Eng. Bruno Cunha, pela disponibilidade e ajuda prestada durante o desenvolvimento de todo o projeto.

Agradeço ao Eng. Filipe Prezado, pela disponibilidade e suporte prestados durante a aprendizagem do meio de desenvolvimento *Azure Machine Learning Studio*.

Em último lugar agradeço à minha família pelo incentivo de continuar a estudar, estando hoje a terminar mais uma etapa da minha vida.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Gráficos

Acrónimos e Símbolos

Lista de Acrónimos

TI	Tecnologia da Informação
AA	Aprendizagem automática
IA	Inteligência Artificial
PaaS	Plataforma como serviço (Platform as a service)
IaaS	Infraestrutura como serviço (Infrastructor as a service)
SaaS	Software como service (Software as a service)
SQL	Structured Query Language
SSMS	SQL Server Managment Studio
ML	Machine Learning
SNH	Sistema Nervoso Humano
RNA	Rede Neuronal Artificial
GB	GigaByte

Lista de Símbolos

∑	Somatório

1 Introdução

Na sequência do estágio, realizado no ano letivo 2016-2017 na Flipkick, este relatório pretende apresentar o trabalho desenvolvido a partir da criação de uma aplicação que permitirá visualizar relatórios sobre a assistência de jogos de futebol por parte dos respetivos adeptos. Para tal, pretende-se mostrar todo o percurso realizado durante a realização do estágio, descrevendo as ferramentas e metodologias usadas para, posteriormente, se desenvolver a aplicação.

Aprendizagem Automática (*Machine Learning* em Inglês) é uma área da Inteligência Artificial (IA) que tem como objetivo o desenvolvimento de técnicas que permite aos computadores aprender através de dados empíricos. Para tal, são utilizadas diversas técnicas de aprendizagem, cada uma com o seu objetivo, tais como Árvores de Decisão, Redes Neuronais, *Support Vector Machines*, *K-Means Clustering*, entre outros.

Este termo insere-se no contexto desta tese, na medida em que se pretende a previsão de assistência em estádios de futebol. Desta forma, este projeto tem como objetivo estimar se um adepto de um determinado clube irá assistir ao próximo jogo no estádio da sua equipa, de modo a colmatar o número de lugares vazios por jogo.

Esta previsão será efetuada através do histórico existente de um adepto, sendo estes os dados de treino - fonte de dados que são utilizados para efetuar a aprendizagem -, utilizando vários critérios e técnicas e identificando padrões de comportamento. Posteriormente, os resultados serão visíveis através de uma ferramenta analítica chamada PowerBi.

A *cloud* da Microsoft, Azure, irá alojar todos os componentes do projeto, como base de dados, metodologias de aprendizagem e todo o projeto envolvente da aplicação.

Com isto, serão obtidos resultados que serão apresentados neste relatório, e que poderão comprovar que o desenvolvimento desta aplicação trará vantagens para os clubes em questão e para o bem-estar dos adeptos, na medida em que facilitará a gestão dos recursos disponíveis, tirando, assim, o melhor proveito dos jogos.

1.1 Objetivos

Este estágio tem como objetivo identificar a probabilidade de um adepto, de um determinado clube, relativamente à assistência do próximo jogo no estádio da sua equipa.

Pretende-se apoiar os clubes de futebol com a previsão do comportamento dos seus adeptos, ou seja, se estarão presentes no próximo jogo do clube, que se realizará no seu estádio. A solução deverá passar pela utilização de algoritmos de aprendizagem automática, para calcular a probabilidade de um dado adepto estar presente no próximo jogo e, com isso, permitir que o clube atue sobre os que não estão presentes, aumentando a taxa de ocupação do estádio. A estimativa da probabilidade dos adeptos na assistência do próximo jogo será obtida através do histórico de um adepto, verificando a assiduidade deste, durante um dado período de tempo.

1.2 Motivação

Atualmente, os adeptos de futebol são cada vez mais exigentes para com as suas equipas e, por consequência, há uma necessidade de melhoramento durante a assistência dos próprios jogos. Tecnologias de linha de golo, comunicação entre árbitros através de um dispositivo e vídeo-árbitro são exemplos de algumas novidades no mundo do futebol, que podem influenciar a ida de um adepto ao estádio. O estado do tempo, os resultados da equipa, a classificação, entre outros, poderão igualmente influenciar.

Assim, o principal motivo que nos leva a realizar esta investigação passa por suportar os clubes a obter informação, de modo a que aumente o número de adeptos nos seus estádios, contribuindo, assim, para a melhoria e o desenvolvimento deste desporto que envolve uma grande massa populacional.

1.3 Planeamento

O planeamento do desenvolvimento da aplicação foi definido e dividido por quatro fases: a fase de análise e desenho da aplicação; a gestão de base de dados; o desenvolvimento do módulo de *Machine Learning* e por fim montagem/ligação dos componentes necessários para a visualização de resultados. Este planeamento foi concebido com base no tempo estimado da duração de cada tarefa e suas subtarefas.

Através de reuniões com o supervisor e com a orientadora, durante todo o período de desenvolvimento, foi usada uma metodologia ágil para gestão e planeamento de projetos de software – o *Scrum* (Figura 1).

Esta metodologia é dividida em ciclos, definidos por um período de tempo, denominados *sprints*, um conjunto de atividades/tarefas que se traduzem numa lista, mais conhecida como

sprint backlog e que serão selecionadas de uma lista maior, a *product backlog*, que contém todas as tarefas do projeto, onde devem ser desenvolvidas durante o período definido. Para que não haja atrasos significativos, é realizada uma reunião diária, normalmente de manhã e por um curto período, onde é discutido o trabalho realizado no dia anterior, identificando problemas e definindo prioridades para essa *sprint*. Quando o período definido termina é efetuada uma reunião para demonstrar o trabalho desenvolvido (*sprint review meeting*) e, assim, definir as próximas *sprints* (*sprint retrospective*).

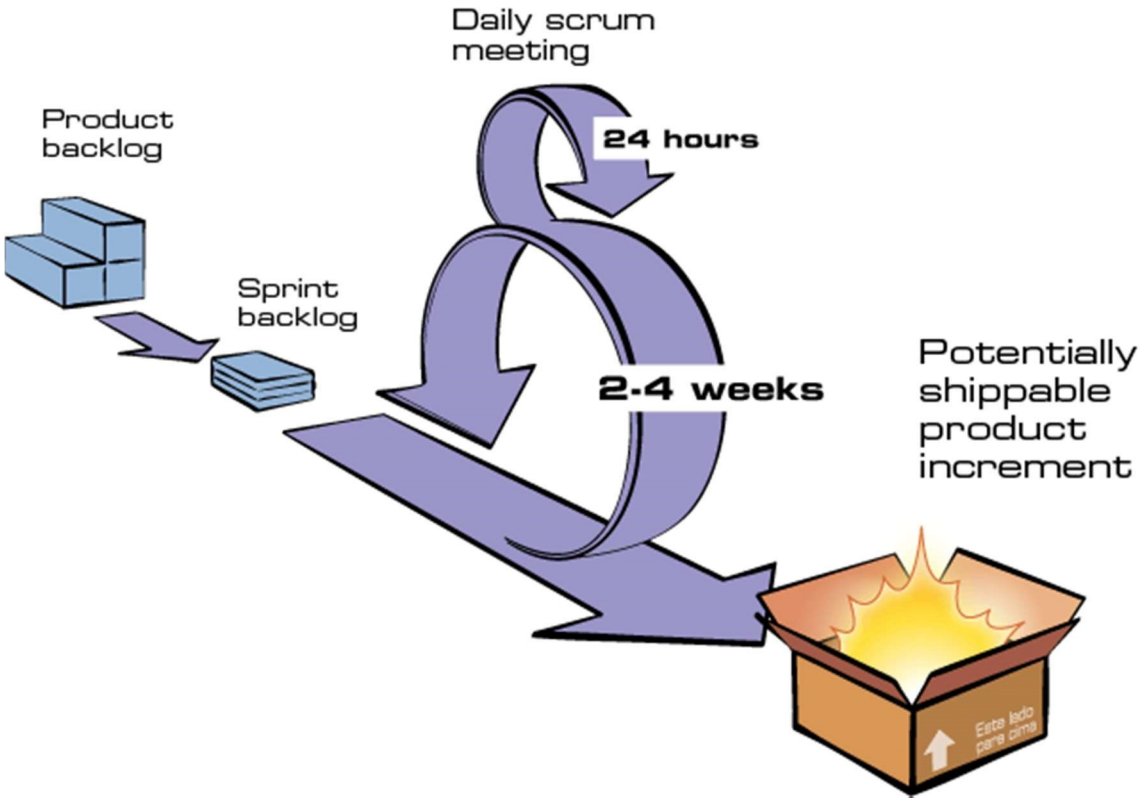


Figura 1 – *Scrum*, uma metodologia ágil (DesenvolvimentoÁgil, 2013/2014)

1.4 Estrutura do documento

A estrutura deste documento está organizada em quatro capítulos, dos quais a Introdução (Capítulo 1), Contexto (Capítulo 2), Descrição técnica (Capítulo 3), Testes computacionais (Capítulo 4) e Conclusões (Capítulo 5).

No primeiro capítulo é introduzido o tema de tese, os objetivos a atingir, a motivação para a sua realização e a estrutura deste documento.

O segundo capítulo apresenta uma descrição das ferramentas/metodologias analisadas para desenvolver o projeto e a sua análise de valor.

No terceiro capítulo é documentado todo o trabalho realizado desde o início do desenvolvimento até à solução final.

No quarto capítulo são apresentados os testes realizados aos algoritmos e demonstração de resultados alcançados.

Por último, o quinto capítulo apresenta todas as ilações retiradas no decorrer do desenvolvimento desta tese.

2 Contexto

Este capítulo irá introduzir as técnicas e ferramentas utilizadas para o desenvolvimento deste trabalho de mestrado. Serão então evidenciadas as técnicas de Aprendizagem Automática analisadas para uma possível solução de previsão do comportamento de um adepto. Posteriormente serão abordadas as restantes ferramentas utilizadas para o desenvolvimento deste projeto.

2.1 Projetos relacionados

2.1.1 Seaters

A tentativa de ocupar todos os lugares disponíveis num evento tem vindo a despertar várias oportunidades por todo o mundo e, consequentemente, a empresa Seaters criou uma aplicação que oferece às pessoas a oportunidade de adquirir o lugar de um evento desportivo mais rápido e a um preço acessível. Esta poderá ser classificada como uma "bilheteira online". (Seaters, 2017)

A aplicação também permite vender lugares, ou seja, caso uma pessoa não possa assistir ao evento inesperadamente, poderá disponibilizar o seu bilhete a um preço apelativo para outro adepto que pretenda assistir ao jogo. Apesar de ter o mesmo objetivo - preencher o maior número de lugares no estádio - este projeto não se enquadra com a previsão de assistências, devido ao facto de apenas disponibilizar lugares e não prever se estes serão ocupados ou não.

2.1.2 Fatores de influência em adeptos de desporto

Várias ligas de desporto profissional têm vindo a preocupar-se ao longo do tempo com a diminuição do interesse do seu produto, que se tem traduzido na falta de assistência em jogos da liga. Para não perderem mais assistência, apenas se tem focado nas pessoas que assistem a jogos, ignorando alguns segmentos da população que não assistem a jogos ou vão com pouca frequência. É muito importante que as ligas identifiquem esses segmentos e investiguem os vários motivos dessa falta de comparência. É necessário, por isso, identificar os fatores que influenciam as pessoas no momento de tomada de decisão para assistir aos jogos.

Para isso, foram realizados alguns estudos sobre os fatores que influenciam os adeptos a assistir a um jogo, nomeadamente investigadores como Greenstein & Marcum, Hart, Schofield que agruparam os vários fatores que influenciam a decisão de um adepto em assistir a um evento desportivo em quatro categorias:

- Atratividade do jogo: habilidade individual, classificações, desempenho, proximidade da concorrência, eventos especiais e entretenimento;
- Económica: preço dos bilhetes, promoções, salário, efeitos televisivos e concorrência de outros eventos desportivos;
- Sócio demográficas: população, idade, género, etnia, ocupação, educação e localização;
- Preferências do público: horário, metereóloga, comodidades do estádio e historial do clube.

Em alguns estudos foram analisados os fatores de atratividade de um jogo, como a presença de celebridades nas equipas, a boa classificação e o impacto de promoções no dia de jogo. Os estudos analisados indicam que os fatores são significativos para a previsão de um adepto assistir a um evento e, cada uma destas variáveis, está relacionada com uma boa assistência. Foi também identificada uma relação positiva entre um sucesso passado e a assistência, mas que para resultados menos bons a assistência tende a diminuir (Fillingham, 1977; Hill et al., 1982; Jones, 1984; Medoff, 1976; Noll, 1974; Scully, 1974; Bird, 1982; Demmert, 1984; Drever & MacDonald, 1981; Hart et al., 1975; Siegfried and Eisenberg, 1980).

Na categoria de fatores económicos, vários investigadores analisaram a disponibilidade de outras formas de entretenimento, impacto na cobertura televisiva de jogos, o impacto alternativo de atrações desportivas no mesmo mercado. A existência de diferentes formas de entretenimento, a disponibilidade de múltiplas atrações desportivas e o aumento do preço dos bilhetes foram classificadas como sendo negativas para as assistências, mas o preço dos bilhetes e a cobertura televisiva têm um impacto ainda mais negativo nas assistências (Fillingham, 1977;

Hill et al., 1982; Medoff, 1976; Noll, 1974; Bird, 1982; Demmert, 1984; Drever & MacDonald, 1981; Hart et al., 1975; Siegfried and Eisenberg, 1980).

Relativamente às variáveis sociodemográficas, o tamanho do mercado foi associado positivamente às assistências, enquanto a etnia dos mercados revelam que quanto maior for um grupo de uma etnia numa população poderá ter um impacto negativo nas assistências. Outros estudos analisaram os parâmetros de localização relacionados com a mesma, incluindo a proximidade com *franchisings*, acessibilidade ao recinto desportivo e tipo de clima, sendo pouco relevantes. Em contrapartida, havia indícios de que a proximidade de outros *franchisings* poderiam ter efeito negativo sobre as assistências (Fillingham, 1977; Medoff, 1976; Noll, 1974; Scully, 1974; Hart et al., 1975; Siegfried and Eisenberg, 1980).

Outros investigadores efetuaram pesquisas sobre fatores que envolvem horários, condições meteorológicas e se os adeptos vão sozinhos ou acompanhados. Os horários em estudo indicaram que as assistências são mais baixas quando os jogos são agendados durante a tarde, mas, à noite ou ao fim de semana e no final da temporada, estas são mais elevadas. As condições climatéricas podem ter algum impacto nas assistências, mas dependem do desporto e se existem outras alternativas ao desporto. O facto de ir a um evento desportivo acompanhado de amigos ou familiares verificou-se que as pessoas tendem a assistir a mais jogos quando vão acompanhadas, mesmo que não assistam frequentemente a jogos (Fillingham, 1977; Hill et al., 1982; Noll, 1974; Bird, 1982; Demmert, 1984; Drever & MacDonald, 1981; Siegfried and Eisenberg, 1980).

Nos estudos acima mencionados, salientam-se a atratividade do jogo e as preferências do público como positivas para as assistências. Entre as variáveis económicas, as promoções são consideradas positivas para aumentar as assistências, enquanto o preço dos bilhetes, outras atividades de entretenimento, cobertura televisiva e a concorrência de outros eventos desportivos foram considerados negativos, que consequentemente diminuem a quantidade de adeptos nos estádios (Douvis, 2014) (Hansen et al., 1989).

2.2 Microsoft Azure

O Microsoft Azure é uma coleção em crescimento de serviços *cloud* integrados que os programadores e profissionais de TI utilizam para criar, implementar e gerir aplicações através da rede global de *datacenters* (Microsoft, 2017a).

Hoje em dia, muitas empresas utilizam serviços na *cloud* (*Nuvem* – através da internet) (Figura 2) como serviços *online* para gerir emails, ouvir música, jogar, armazenar ficheiros, entre outros. A *cloud* permite aos informáticos, através de um *browser*, desenvolver aplicações/serviços, efetuar armazenamento, cópias de segurança, recuperação de dados, alojamento de páginas *web*, análise de dados e a maior parte das operações necessárias para satisfazer as suas necessidades (Microsoft, 2017b).

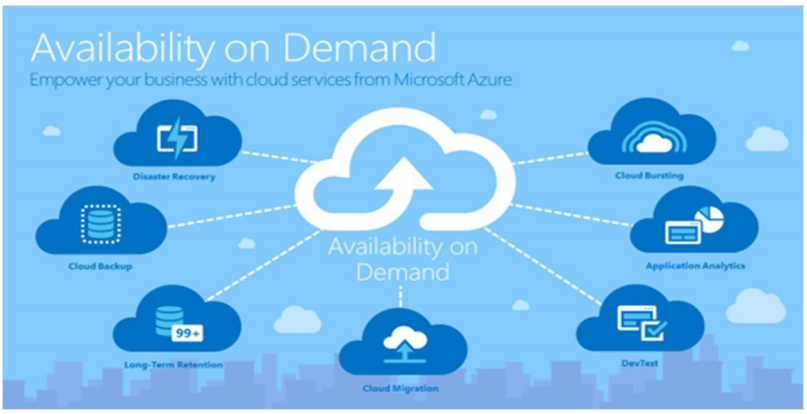


Figura 2 – Serviços da *cloud* Microsoft Azure (Ferman, 2015)

Deste modo, os serviços têm vindo a reconsiderar a ideia de como as entidades pensam nos recursos informáticos, devido aos benefícios inerentes. Os custos relativos à aquisição de *hardware*, *software* e gestão de servidores locais são eliminados, ficando tudo à responsabilidade da *cloud*, traduzindo-se, assim, no aumento de produtividade e num desempenho melhorado. Todos os recursos fornecidos são atualizados regularmente para *hardware* de última geração, rápido e eficiente (incluindo latência reduzida), retirando o tempo despendido na gestão de servidores locais. Estes serviços possibilitam o redimensionamento elástico, ou seja, permitem o aumento ou diminuição dos recursos informáticos necessários. Um dos benefícios mais importantes nestes serviços é a sua fiabilidade, tornando, assim, mais fácil a realização de cópias de segurança de dados, recuperações, após falhas e a continuidade do negócio, uma vez que os dados podem ser espalhados em vários locais redundantes (Figura 3).

Figura 3 – Principais benefícios da informática na *cloud* (Microsoft, 2017b)

Conforme mencionado anteriormente, a *cloud* oferece muitas vantagens para as empresas, permite que se faça mais com menos, tendo acesso a aplicações empresariais críticas sem a necessidade de se preocupar com a manutenção ou atualizações destas. No entanto os serviços de uma *cloud* não são caracterizados apenas por vantagens, tendo estes também as suas desvantagens. Entre as desvantagens mais importantes estão a dependência na qual a *cloud* necessita obrigatoriamente de uma conexão à *Internet*, e caso esta não exista não há forma de contornar o problema; a Fiabilidade também está do lado das desvantagens, que

apesar da *cloud* ser bastante fiável, falhas e erros são sempre possíveis de ocorrer, pelo que se a *cloud* deixar de funcionar os dados ficam comprometidos e eventualmente podem ser perdidos; a Vulnerabilidade, um tema sempre muito presente em qualquer área da informática, não é exceção pois todos os dados na *cloud* estão vulneráveis a ataques, pois quanto mais importante e confidencial for a informação mais interessante se torna no ponto de vista dos piratas (*hackers*) (Izumi e Lopes, sem data).

Existem três categorias de serviço na *cloud* (Figura 4), o IaaS (Infraestrutura como serviço) - a categoria básica dos serviços na *cloud* onde são fornecidos servidores, máquinas virtuais, armazenamento, redes e sistemas operáticos, o PaaS (Plataforma como serviço) que se refere aos serviços de informática, fornecendo ambientes para desenvolver, testar e gerir aplicações de *software*, e por último o SaaS (*Software* como serviço) - um método para fornecer aplicações de *software* através da *Internet*. Estas categorias, por vezes, são denominadas como "pilha de informática na *cloud*", pois são utilizadas umas em cima das outras (Microsoft, 2017b).

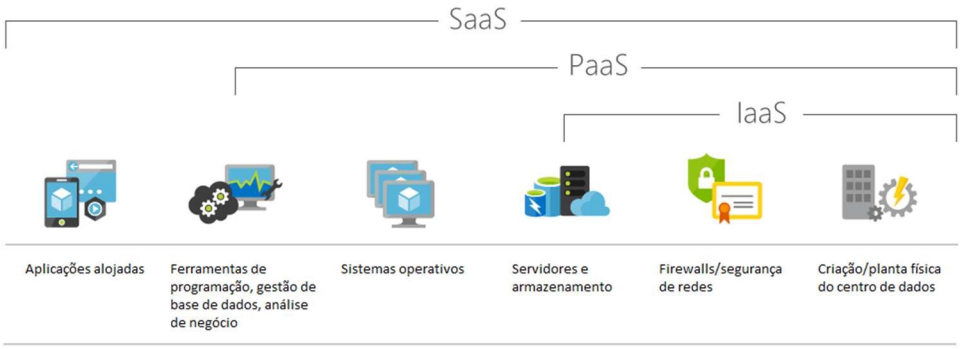


Figura 4 – Categorias de serviço na *cloud* (Microsoft, 2017i)

2.2.1 Serviço de base de dados SQL da *cloud* Azure

A base de dados SQL do Azure é uma base de dados relacional como serviço que utiliza o Motor do Microsoft SQL Server. A Base de Dados SQL é uma base de dados de elevado desempenho, fiável e segura que pode ser utilizada para criar aplicações e sites com base em dados numa qualquer linguagem de programação, sem ter de gerir infraestrutura (Microsoft, 2017c).

Para uma boa gestão da base de dados é necessário utilizar um programa externo ao portal, como o SQL Server Managment Studio (SSMS) - um ambiente de integrado para gerir qualquer base de dados SQL. Este possui um ambiente de desenvolvimento gráfico com editores de *scripts*, permitindo configurar, gerir, administrar e desenvolver componentes do SQL Server, base de dados Azure SQL e *SQL Data Warehouses*. Além disso, é possível efetuar a gestão da base de dados, através da linha de comandos do Windows ou da linha de comandos interna do portal do Azure a CLI do Azure (Stein et al., 2017).

A gestão através da linha de comandos pode ser morosa, devido à necessidade de introdução de comandos para realizar ações sobre a base de dados, tornando-se mais "*user friendly*" a utilização do SSMS e ágil durante o processo.

Como referido anteriormente, um serviço na *cloud* permite a elasticidade do mesmo, o serviço de base de dados SQL tem a possibilidade de aumentar tanto de tamanho como de desempenho, consoante as necessidades de cada projeto ou das preferências do programador.

2.2.2 Serviço de Machine Learning do Azure

O Azure Machine Learning é um serviço de análise preditiva baseado na *cloud* e que torna possível a criação e implementação rápidas de modelos preditivos como soluções de análise. O Azure Machine Learning fornece ferramentas para a análise preditiva dos modelos, como também presta um serviço totalmente gerido que pode utilizar para implementar os seus modelos preditivos como serviços Web prontos a consumir (Figura 5) (Gronlund et al., 2017).

Os modelos de *Machine Learning* são criados na *Web app* Microsoft Azure Machine Learning Studio (ML Studio), uma ferramenta *drag-and-drop*, que permite construir, testar e publicar as soluções desenvolvidas através de *web services*.

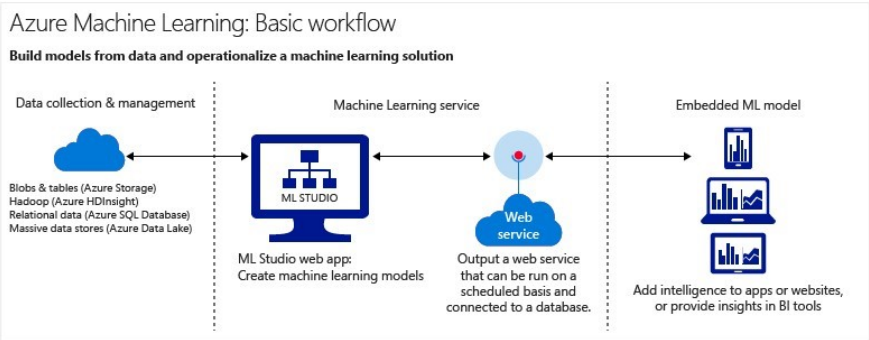


Figura 5 – *Workflow* básico do serviço de Machine Learning do Azure (Gronlund et al., 2017)

O ML Studio é um ambiente de desenvolvimento, que oferece ao utilizador uma forma rápida de implementação dos modelos de *Machine Learning*, devido ao facto de ser *drag-and-drop* e acessível em qualquer lugar através de um *browser*. Está dividido em três separadores, o separador dos módulos disponíveis para arrastar; a área para desenvolvimento dos modelos e uma terceira para a configuração de cada módulo (caso exista). Os módulos disponíveis estão divididos por categorias, destacando-se o módulo de *script*, que permite a execução de algoritmos personalizados em linguagem R ou Python e ainda a possibilidade de execução de *querys* SQL. A Figura 6 apresenta o ambiente de desenvolvimento ML Studio.

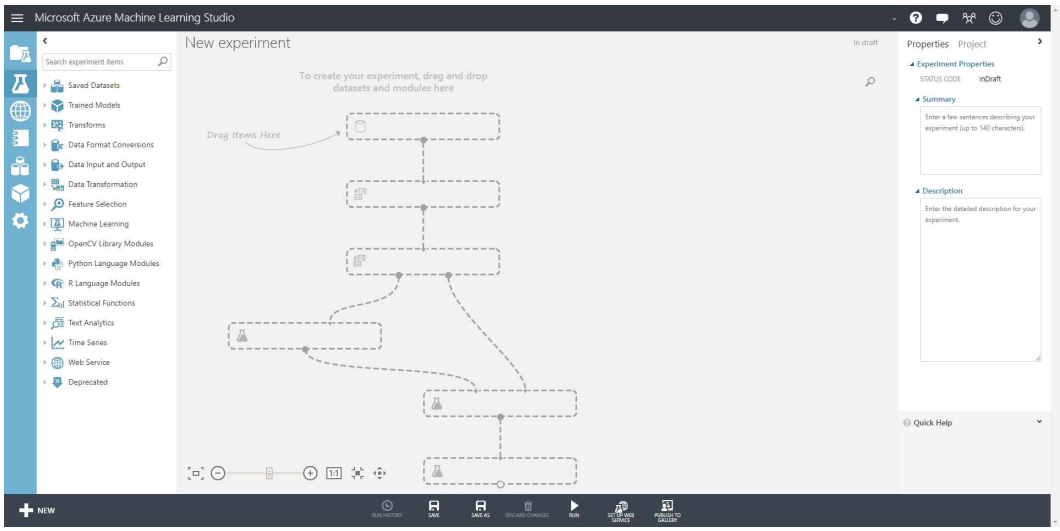


Figura 6 – Ambiente de desenvolvimento Machine Lerning Studio da *cloud* Azure

De modo a auxiliar a utilização deste serviço, existe uma galeria de soluções e discussões para que programadores possam analisar e consultar soluções partilhadas pela comunidade, facilitando assim a aprendizagem e partilha de conteúdos - "Cortana Intelligence Gallery".

2.2.3 Serviço Power BI

O Power BI é uma ferramenta de análise de negócios que, juntamente com os serviços do Azure (Figura 7), facilita o esforço de processamento de dados em análises e relatórios que oferecem informação em tempo real. O Power BI permite uma grande quantidade de ligações a fontes de dados do Azure, tornando as soluções de *business intelligence* exclusivas, possibilitando, assim, a formatação e ajustamento dos dados para gerar relatórios personalizados (Microsoft, 2017h).

Figura 7 – Serviços da *cloud* Azure e relatório no Power BI (Microsoft, 2017h)

2.3 Aprendizagem Automática

A Inteligência Artificial (IA) é o conceito mais abrangente para considerar a realização de uma tarefa por parte de uma máquina como inteligente. A sua origem remete para tempos dos mitos gregos, que contêm histórias sobre humanos mecânicos, programados para imitar o comportamento do homem.

Nos primórdios da era da tecnologia, os computadores foram concebidos como máquinas lógicas, com o intuito de criar cérebros mecânicos, porém, com o avanço tecnológico, a compreensão do funcionamento da mente fez com que o conceito de IA mudasse. Desta forma, em vez de efetuar cálculos mais complexos, a IA passou a concentrar-se em imitar o processo de tomada de decisão humana, bem como a realização de tarefas mais genuínas.

A IA foi o ponto de partida para o desenvolvimento da Aprendizagem Automática (AA), que é muitas vezes referenciada como uma subárea da IA. Um dos avanços mais importantes para a AA foi, em 1959, realizado por Arthur Samuel com o conceito de AA - "dotar as máquinas de modo a que estas aprendam por si próprias sem que sejam explicitamente programadas" ao invés de ensinar tudo sobre o mundo e como realizar tarefas. Outro aspeto importante para o avanço da AA foi, mais recentemente, o aparecimento da *Internet*, que envolve uma grande quantidade de informação. Com estas inovações, os engenheiros aperceberam-se que seria mais eficiente ensinar as máquinas a pensar como humanos e ligá-las à internet para terem acesso à informação, do que ensinar-lhes a fazer tudo (Marr, 2016).

Como a informação tem uma posição muito importante na indústria tecnológica, a sua análise é muito importante e, por conseguinte, a AA tem vindo a ganhar cada vez mais popularidade nos últimos anos devido à sua enorme capacidade para fazer previsões ou sugestões, através de um grande volume de dados. Desta forma a AA é utilizada para resolver certos problemas, como o Reconhecimento Ótico de Caracteres (OCR), reconhecimento facial, diferenciação entre emails spam e não spam, diagnósticos médicos, deteção de fraudes, entre outros. A AA enfrenta os problemas através da utilização de algoritmos, na qual são desenvolvidos para serem o mais abrangentes possível, com valor prático e eficientes de modo a encontrar soluções para os problemas (Schapire, 2008).

O principal objetivo da AA envolve a criação de regras, de modo a que estas se possam aplicar sobre acontecimentos semelhantes, pretendendo que o resultado das suas análises seja o mais preciso possível, como por exemplo o diagnóstico médico na qual se deseja que identifique facilmente se um paciente sofre ou não de alguma doença, através dos sintomas apresentados (Figura 8).

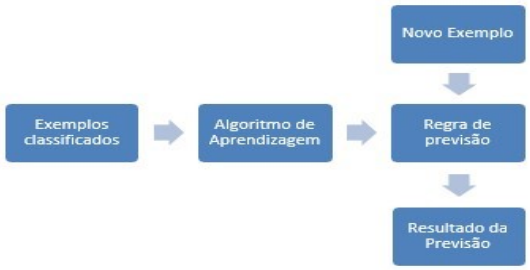


Figura 8 – Diagrama de um problema típico de aprendizagem (Schapire, 2008)

As metodologias da AA são tipicamente classificadas em três grandes categorias: a aprendizagem supervisionada, onde uma propriedade (resultado) está disponível para um conjunto de dados (dados de treino); a aprendizagem não supervisionada, que possibilita a descoberta de relacionamentos implícitos num conjunto de dados não rotulados (sem relações); e por último, a aprendizagem por reforço, que permite aos agentes de *software* determinar qual a melhor opção a tomar, num determinado contexto, de forma a maximizar o seu desempenho.

2.3.1 Aprendizagem Supervisionada

A aprendizagem supervisionada tem como objetivo analisar/estudar um conjunto de dados rotulados, de modo a conseguir fazer previsões de dados futuros. Através de um conjunto de dados classificados, uma percentagem é utilizada em treino e outra para previsão. Cada conjunto possui as suas propriedades e um valor que caracteriza essas propriedades (resultado ou sinal de supervisão), como por exemplo, um carro X que com certas características possui o preço Y. O processo de aprendizagem do algoritmo passa por mapear uma função com os dados de entrada (X), de modo a obter o valor de saída (Y). O processo do algoritmo é muito semelhante à forma da supervisão do processo de aprendizagem de um aluno, feito por um professor.

Os problemas de aprendizagem supervisionada podem ser agrupados em classificação e regressão. Um problema de classificação é caracterizado quando o valor de saída é uma categoria, (por exemplo sim/não, azul/verde, etc.), um problema de regressão é caracterizado quando o valor de saída é um valor (por exemplo percentagens, euros, peso, entre outros) (Ericson et al., 2017a) (Brownlee, 2016).

2.3.2 Aprendizagem não supervisionada

A aprendizagem não supervisionada tem como objetivo organizar os dados de entrada ou modelar a sua estrutura. Os dados de entrada não são rotulados, ou seja, não existe qualquer variável que os permita relacionar. O processo de aprendizagem do algoritmo passa por realizar operações para descobrir e apresentar uma estrutura dos dados, ou se existem relações entre os diferentes dados de entrada.

Na aprendizagem não supervisionada os problemas de aprendizagem subdividem-se em duas categorias: *clustering* e associação. O *clustering* adequa-se principalmente para a identificação de grupos, como, por exemplo, descobrir similaridade de clientes, de acordo com as suas compras. Um problema de associação envolve a descoberta de regras em grande volume de dados, como pessoas que compram o produto X também tendem a comprar o produto Y (Ericson et al., 2017a) (Brownlee, 2016).

2.3.3 Aprendizagem por reforço

A aprendizagem por reforço visa descobrir, através da tentativa-erro, qual a melhor opção a tomar, em resposta a cada conjunto de dados. É frequentemente utilizado na área da robótica, na qual são efetuadas leituras aos sensores num determinado tempo (um conjunto de dados). O algoritmo deve escolher a próxima ação do robô, que recebe posteriormente um sinal de recompensa, indicando se a decisão foi boa ou má. Deste modo, o algoritmo ajusta a sua estratégia para alcançar a recompensa mais alta e vai registando os vários estados pela qual passa, de modo a que no futuro possa efetuar comparações para melhorar o seu comportamento. Uma limitação deste algoritmo é o facto de registar de valores que levam à necessidade de uma grande quantidade de memória (Champandard, 2001-2002).

2.3.4 Algoritmos mais conhecidos

2.3.4.1 Redes Neuronais

A Rede Neuronal Artificial (RNA) é um modelo matemático baseado no Sistema Nervoso Humano (SNH) que adquire conhecimento através da experiência. O SNH é formado por grandes quantidades de células, os neurónios, tendo estes uma grande influência no funcionamento, comportamento e raciocínio do corpo humano. Como a SNH, a RNA é composta por várias unidades de processamento, também denominados de neurónios, na qual simulam o mesmo comportamento dos neurónios biológicos, apresentando assim semelhanças a nível estrutural. Ambos possuem terminais para a entrada de informação, processamento da informação e saída de resultados, correspondendo os dendritos do neurónio biológico, que são um conjunto de terminais para a entrada de informação à entrada de dados do neurónio artificial, o núcleo unidade de processamento do neurónio biológico assemelha-se ao processamento dos dados que ocorre no neurónio artificial, e os terminais do axônio ao sinal de saída onde serão emitidos os resultados do processamento (Carvalho, sem data). A Figura 9 e Figura 10 apresentam as estruturas de ambos os neurónios, biológico e artificial respetivamente, onde se podem verificar as semelhanças entre estes.

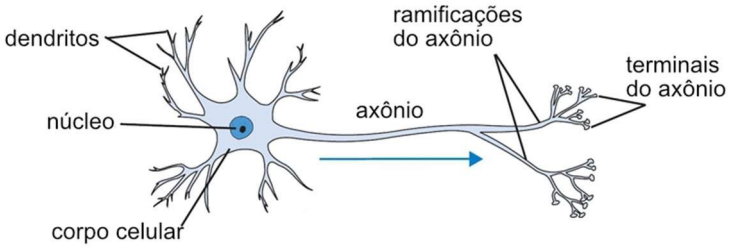


Figura 9 - Neurónio de uma RNB (Bezerra, 2016)

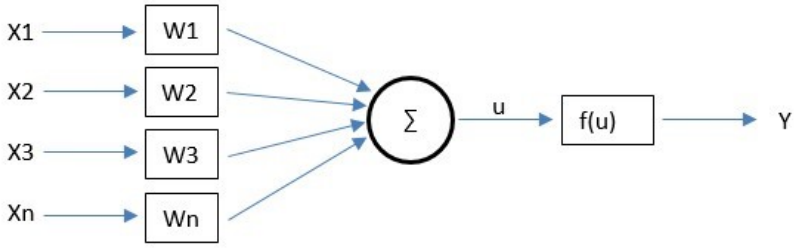


Figura 10 – Neurónio de uma RNA

A estrutura de um neurónio artificial da rede está ilustrada na Figura 10, em que o X1, X2, X3, Xn são os sinais de entrada; W1, W2, W3, Wn são os pesos sinápticos correspondentes ao sinal. Depois da passagem do sinal pelo peso, o produto dos dois valores será enviado para a Função agregadora (∑), onde serão somados todos os valores para posteriormente ser calculado o Potencial de ativação (u). O u será analisado de seguida na Função de Ativação (f), que tem como objetivo emitir um Sinal de Saída (y), normalmente valores 0 ou 1 (não ou sim).

O processo de aprendizagem da RNA consiste em ajustar os pesos, de modo a que cada iteração consiga encaixar a sua amostra numa classe (Carvalho, sem data).

2.3.4.2 Árvores de decisão

Uma Árvore de Decisão é a forma de representar uma tabela de decisão, que possui um conjunto de dados previamente classificados (valores padrão). Os dados que posteriormente serão analisados utilizam os “valores padrão”, como referência para poderem ser classificados.

As Árvores de Decisão classificam instâncias, ordenando-as desde a raiz até à folha classificadora, na qual é identificada a classificação da instância. Cada nó da árvore descreve um atributo, enquanto que cada ramo é um valor desse atributo (Freitas, sem data).

Tabela 1: Exemplos de dados de treino para Jogar Ténis (Freitas, sem data)

Dia	Aspeto	Temperatura	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

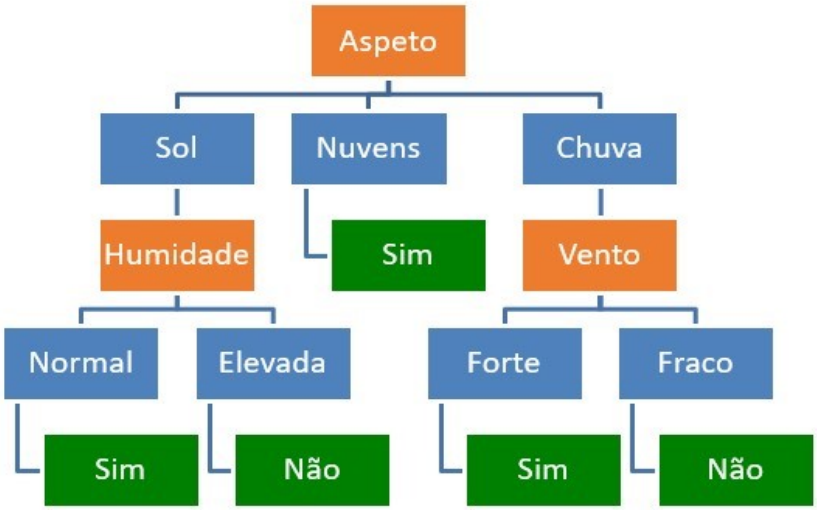


Figura 11 – Árvore de decisão para jogar Ténis (Freitas, sem data)

Através dos exemplos de treino da Tabela 1 é possível contruir a árvore de decisão da Figura 11, e caso seja pretendido efetuar uma classificação através desta árvore, para o atributo Aspeto ter o valor Sol e o atributo Humidade ter o valor Elevada, este exemplo é classificado como não, ou seja, de acordo com a tabela previamente definida, quando o Aspeto é Sol e a Humidade é Elevada não se joga ténis.

Para a construção de Árvores de Decisão pode-se recorrer à utilização do algoritmo ID3 (Inductive Decision Tree), onde se efetua uma pesquisa "Top-down" sem *backtracking*. Este algoritmo começa a construir a Árvore pela raiz (um atributo), adicionando todos os valores possíveis abaixo desta (ramos), de seguida volta a criar um nó (atributo) e os seus valores

possíveis (ramos) até existirem filhos não puros. Um filho é puro quando cada atributo tem o mesmo valor em todos os exemplos (Freitas, sem data).

2.3.4.3 Support Vector Machine

O Support Vector Machines (SVM) é um algoritmo de aprendizagem supervisionada, que pode ser utilizado para problemas de classificação, mas também para problemas de regressão. Este tem como objetivo separar um conjunto de dados linearmente separáveis, através de um ponto inserido no espaço n-dimensional, sendo o valor de cada característica o valor de uma determinada coordenada. Desta forma, a função de classificação define um hiperplano de forma a identificar grupos de classes. A SVM tenta maximizar a distância entre as duas classes, de modo a que esta ofereça uma melhor capacidade de generalização, proporcionando um melhor desempenho nas próximas avaliações e facilitando a identificação de classes (Ray, 2017).

A SVM não trabalha apenas com padrões linearmente separáveis, pelo que se considera um conjunto de dados não-linearmente separáveis quando não seja possível separar os dados através de um hiperplano. A Figura 12 apresenta um conjunto de dados linearmente e não-linearmente separáveis, respetivamente.

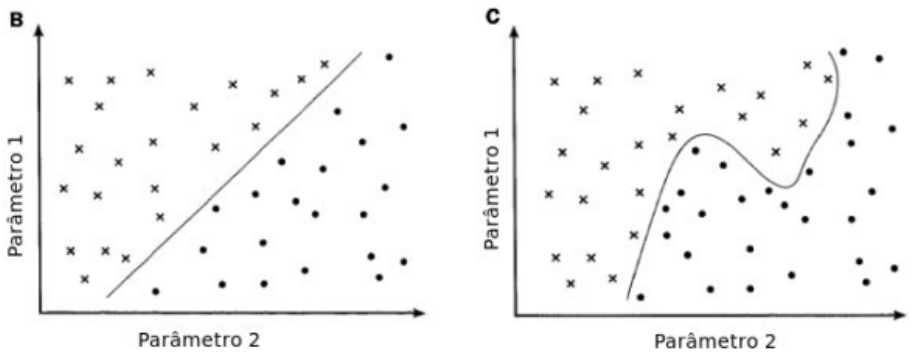


Figura 12 – Conjunto de dados linearmente e não-linearmente separáveis respetivamente (Gonçalves, sem data)

Segundo o teorema de Cover, um problema não-linear tem mais probabilidade de ser linearmente separável se colocado num espaço de dimensão mais alta. Deste modo, a SVM aplica uma mudança de dimensão através das funções `Kernel` - função que retorna o produto escalar das imagens de seus argumentos -, convertendo, assim, um problema não-linear em linearmente separável (Gonçalves sem data).

2.3.4.4 K-Means Clustering

O K-Means é um dos algoritmos de aprendizagem não supervisionada que resolve problemas de agrupamento e que pode ser considerado o problema mais importante deste tipo de aprendizagem. Este realiza um processo para descobrir grupos de objetos cujos membros têm algumas semelhanças, mas sem qualquer relação entre eles, ou seja, não estão classificados. Um *Cluster* é, portanto, um grupo de objetos similares e diferentes para outros grupos de clusters.

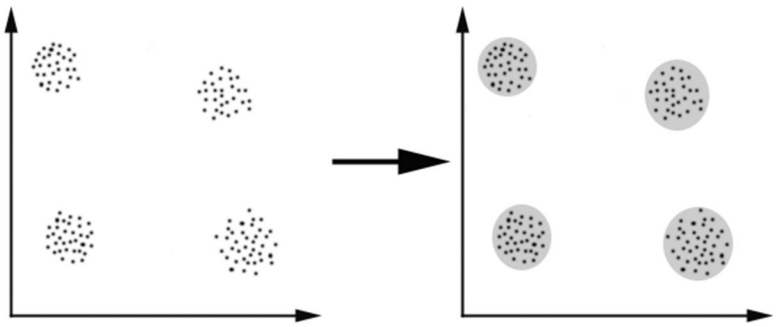


Figura 13 – Associação de conjuntos de dados a clusters (Matteucci, sem data)

Na Figura 13 são facilmente identificados quatro *clusters*, onde o conjunto de dados pode ser dividido. O processo de agrupamento de um conjunto de dados não é realizado através de medidas de similaridade simples, mas através de características específicas, de modo a satisfazer as necessidades da utilização do *clustering*. O agrupamento de dados pode, então, ser dividido em critérios, nomeadamente o *Clustering* baseado em distância e *Clustering* conceptual. Um *cluster* baseado em distância utiliza o critério de similaridade, em que dois ou mais objetos estão mais próximos de acordo com uma determinada distância, pertencendo, assim, ao mesmo *cluster*. Um *cluster* conceptual é formado se dois ou mais objetos possuírem características em comum (Matteucci, sem data).

2.4 Análise de valor

Segundo o conceito de análise de valor, "uma metodologia utilizada para identificar funções relacionando-as com os custos, de modo a reduzir custos e aumentar o desempenho", serão discriminados os cinco elementos-chave do modelo "the new concept development model" (NCD), benefícios e sacrifícios do cliente, proposta de valor do produto, modelo canvas, enquadramento do modelo de Verna Allen e uma forma de utilização do método de processo de análise hierárquica (AHP).

2.4.1 Modelo “The New Concept Develompent” (NCD)

O modelo NCD está “localizado” no topo dos fatores influenciáveis. Este modelo é composto por cinco elementos, como criação e melhoria da ideia, seleção da ideia, definição do conceito, identificação de oportunidade e análise de oportunidades. Este modelo é circular, onde todos os seus elementos estão em constante interação entre eles. As setas de entrada do modelo significam o início de uma oportunidade ou surgimento da ideia, em contrapartida com a seta de saída, que representa o modo como o conceito é levado para outro nível. A Figura 14 ilustra os cinco elementos-chave deste projeto.

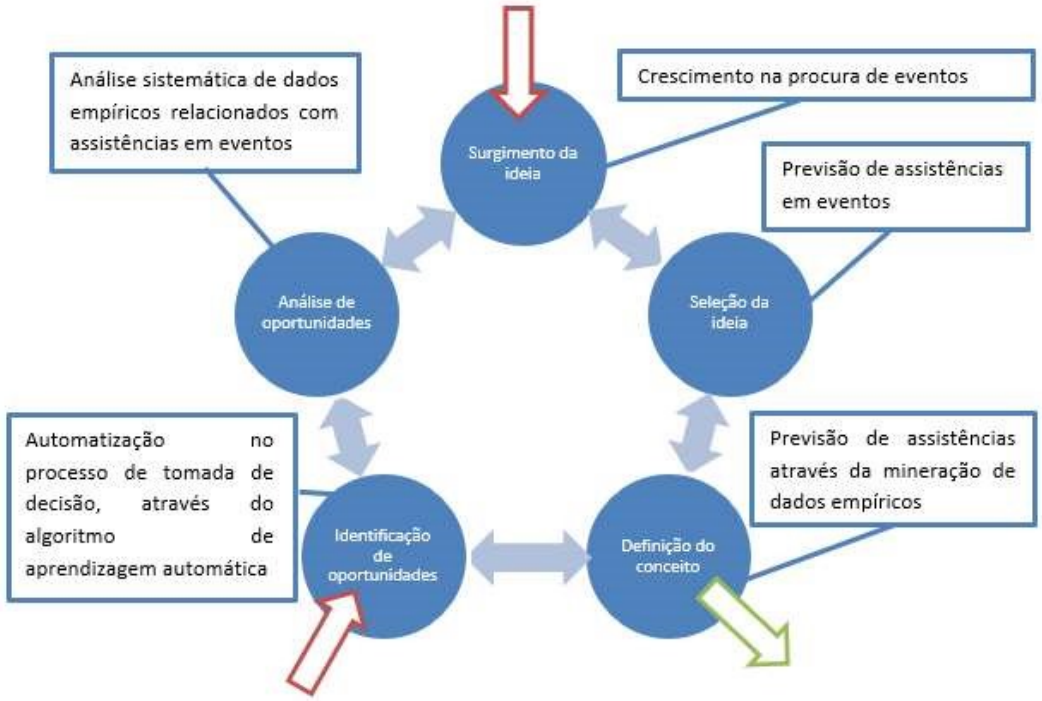


Figura 14 – Identificação dos cinco elementos chave do modelo NCD

Para analisar e identificar cada elemento foram utilizados técnicas/métodos, como a análise SWOT, que permite analisar e identificar as oportunidades existentes; Brainstorming que, tal como o próprio nome indica, é referente ao aparecimento de uma tempestade de ideias para conseguir definir um conceito.

2.4.2 Benefícios e sacríficos do cliente

Segundo Ludwig Von Miss o valor é “a forma de como o homem reage às condições do meio envolvente”, dependendo das “necessidades pessoais, interesses, atitudes e preferências” - Susana Nicola Eduarda Pinto Ferreira e J.J. Pinto Ferreira.

O valor para o consumidor são todas as vantagens que ele retira do produto/serviço que a organização lhe oferece, emitindo os valores da empresa pela qual o consumidor se identifica, ficando então fidelizado (valor percebido).

De acordo com as definições descritas acima, o cliente tem um pequeno sacrifício ao despender um pouco do seu tempo a fornecer informação, de modo a que possa ser analisado o seu comportamento, beneficiando, assim, a curto, médio e longo prazo de promoções adequadas ao seu perfil, bem como de atendimento personalizado.

2.4.3 Proposta de valor do produto

Este produto pretende demonstrar que com menos trabalho se consegue analisar mais informação, num curto espaço de tempo e bastante otimizado. Por exemplo, caso uma pessoa se dedicasse a analisar os dados de um utilizador, esta despenderia de algum tempo para conseguir classificar o utilizador e, além disso, os resultados podem induzi-la em erro.

Com a afirmação de Verna Allee, "People naturally network as they work so why not model itself as networks", podemos aferir que é possível construir e analisar o valor, ouvindo a opinião das pessoas, mapear informação e agir em conformidade com esta.

2.4.4 Modelo de negócio *Canvas*

O modelo de negócio *Canvas* é uma ferramenta de gestão estratégica na que permite desenvolver modelos de negócio novos ou existentes. Este modelo foi proposto por Alexander Osterwalder sustentado pelo seu trabalho sobre *Business Model Ontology*. Esta metodologia é um mapa pré- formatado composto por nove elementos que compõem um negócio, facilitando assim a visualização das atividades de uma empresa. Está dividido em duas grandes secções, a da direita (*Front Stage*) que contem elementos mais ligados aos clientes e a da esquerda (*Back Stage*) contém os elementos mais relacionados com os recursos necessários para suportar o negócio. Cada elemento contém um espaço para ser preenchido por respostas de modo a satisfazer os objetivos de cada um, que apesar de estarem separados todos eles se relacionam (Moura, 2014) (ISEP, 2016/2017).

Os elementos da secção da direita, como o Segmento de Clientes (*Customer Segments*) onde se pretende identificar para quem se cria valor e quem são os clientes mais importantes; as Relações com os Clientes (*Customer Relationships*) na qual se descrevem as relações existente com os clientes e que tipo de relação se espera que exista; os Canais de Distribuição (*Channels*) tem o intuito de demonstrar os meios para chegar aos clientes; a Proposta de Valor (*Value Proposition*) onde se identificam os problemas os clientes possuem que a empresa pode ajudar a resolver, que produtos se vão oferecer a cada segmento e as necessidades de cada um. Os elementos da secção da esquerda, como os Recursos-Chave (*Key Resources*) na qual se identificam os recursos necessários para o desenvolvimento do produto; nas Atividades-Chave (*Key Activities*) são descritas as atividades que a proposta de valor exige; os Parceiros-Chave

(*Key Partners*) permitem visualizar que fornecedores e parcerias a empresa terá. Os dois elementos em baixo estão relacionados com a área financeira do negócio, onde a Estrutura de Custos (*Costs Struture*) apresenta os custos relativos a atividades, recursos e custos mais importantes para o negócio e as Fontes de Receita (*Revenue Stream*) que descrevem os valores que os segmentos de clientes estão dispostos a pagar, quanto é que pagam atualmente e a contribuição de cada fonte de receita para a receita total (ISEP, 2016/2017). A Figura 15 apresenta o modelo de negócio *Canvas* do projeto.

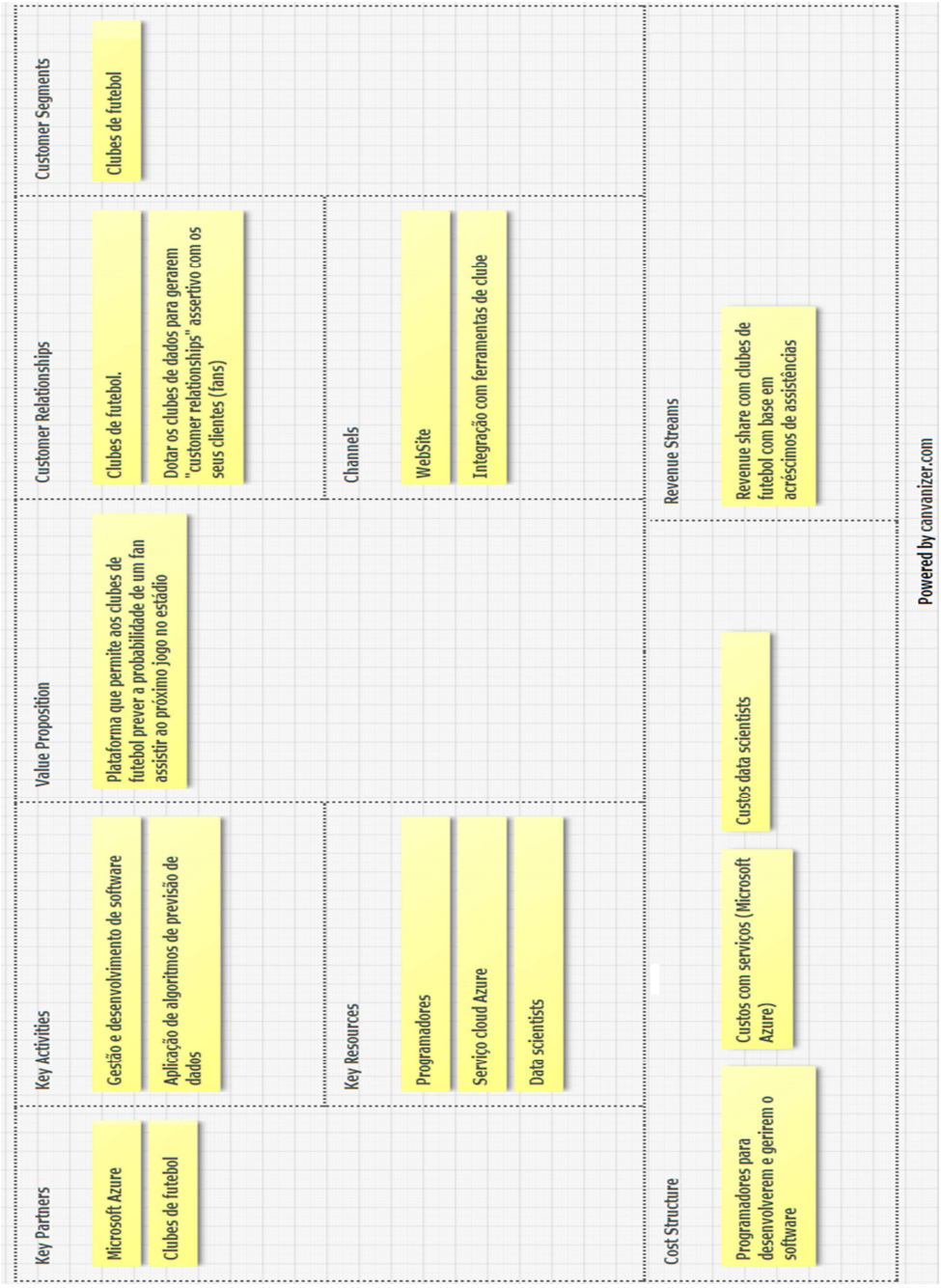


Figura 15 – Modelo de negócio Canvas do projeto

3 Descrição técnica

Este capítulo descreve todo o trabalho desenvolvido, em prol de uma solução que satisfaça as necessidades dos objetivos propostos. Por isso, será descrito detalhadamente o modo de funcionamento da solução encontrada e sua estrutura. É de salientar também a demonstração de testes realizados, bem como a sua análise, que resultarão na avaliação da solução.

3.1 Análise e desenho

Como foi referido anteriormente no planeamento, o projeto foi dividido por fases, onde a primeira se refere à análise e desenho da aplicação. O presente capítulo descreverá a primeira fase do projeto, que envolve a análise de requisitos, arquitetura da aplicação e análise de projetos relacionados.

3.1.1 Análise de requisitos

Segundo Ian Sommerville (Sommerville, 2013), os requisitos de um sistema são as descrições que o sistema deve realizar, os serviços que oferece e as restrições do seu funcionamento, refletindo as necessidades dos clientes. Os requisitos são normalmente divididos em dois tipos, funcionais e não funcionais. Os requisitos funcionais descrevem os serviços que o software deve fornecer, como este reage e se comporta em determinadas situações, isto é, o que um sistema deve fazer quando um utilizador interage com este. Os requisitos não funcionais são requisitos que não estão ligados diretamente com as funcionalidades oferecidas pelo sistema aos utilizadores, estando mais relacionados com as propriedades emergentes do sistema, como segurança, desempenho, usabilidade, entre outros. A Figura 16Figura 16 apresenta os tipos de requisitos não funcionais definidos por Ian Sommerville.

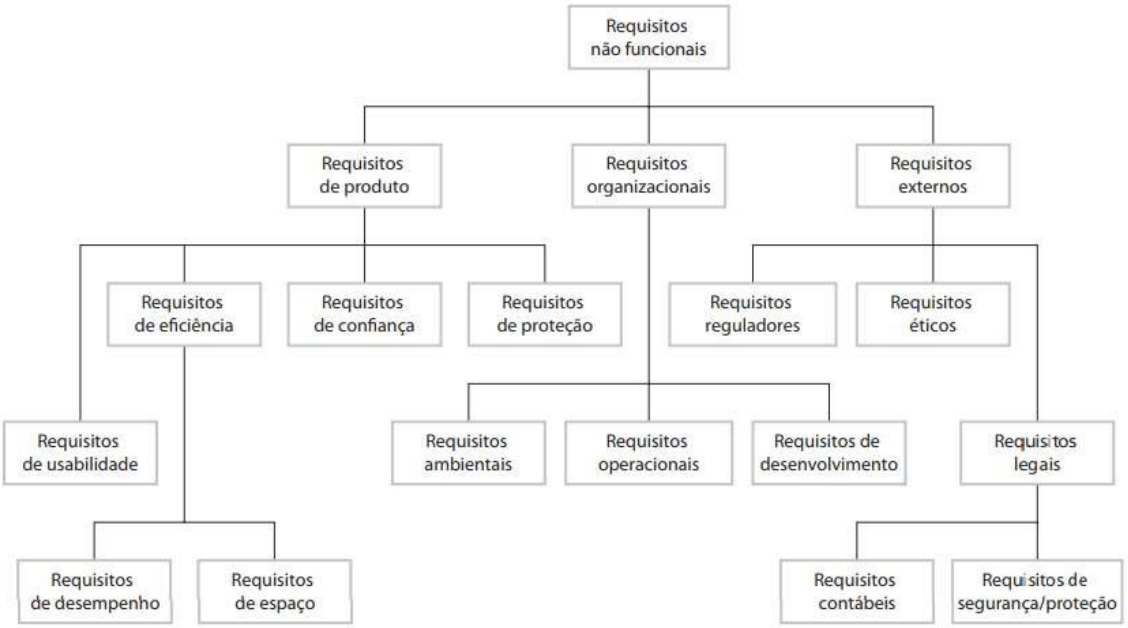


Figura 16 – Tipos de requisitos não funcionais (Sommerville, 2013)

No que diz respeito aos requisitos não funcionais, quando o ambiente de desenvolvimento passa pela *cloud* Azure, alguns destes requisitos são garantidos pela *cloud*. O mesmo não se pode dizer quando existe a necessidade de efetuar manutenção do sistema completo (servidores, latência, software, etc).

3.1.1.1 Requisitos funcionais

De acordo com a definição de requisitos funcionais anteriormente enunciada, foram definidos os requisitos e apresentados num diagrama de casos de uso, para ir ao encontro das necessidades do cliente (supervisor). O diagrama de casos de uso é uma forma de representar requisitos funcionais. O termo fora introduzido por Ivar Jacobson (Jacobson et al., 2011), onde definiu um caso de uso como todas as formas de utilização de um sistema por parte de um utilizador, para que este possa alcançar os seus objetivos, ou seja, tudo o que o utilizador poderá fazer com o sistema.

Os requisitos funcionais foram definidos de acordo com as necessidades do cliente, salientando os casos de uso 5. Visualizar todos os adeptos, 6. Visualizar detalhes de um adepto, 7. Visualizar percentagem de adeptos que vão assistir ou não ao próximo jogo, 8. Visualizar histórico de assistências de um adepto, 9. Visualizar adeptos com probabilidade personalizada e por ultimo o caso de uso 10. Visualizar a probabilidade de um adepto assistir ao próximo jogo, de acordo com o diagrama de casos de uso definido ilustrado na Figura 17.

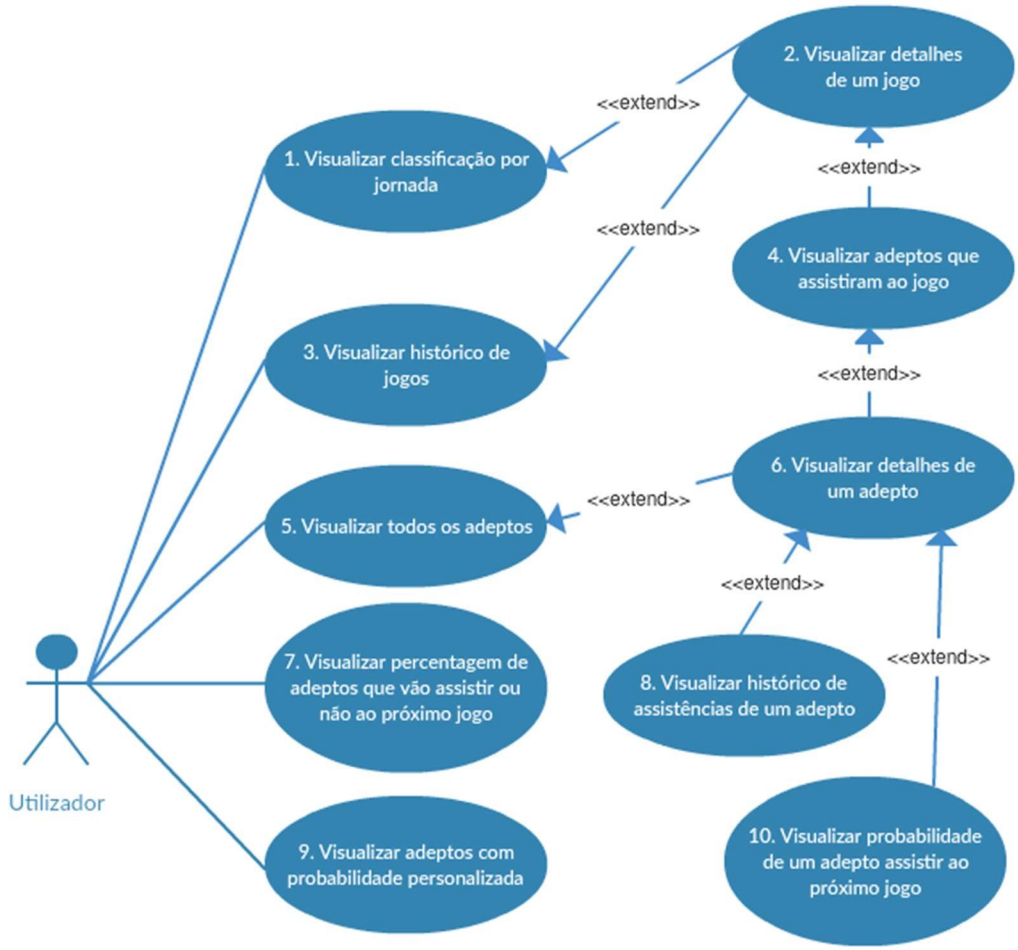


Figura 17 – Diagrama de casos de uso

3.1.1.2 Descrição de casos de uso

Tabela 2: Descrição do caso de uso 1. Visualizar classificação por jornada

Caso de uso: Visualizar classificação por jornada

Actor:	Utilizador
Pré-Condição:	Nenhuma
Descrição:	O sistema apresenta um quadro com a classificação da jornada por defeito, juntamente com os jogos dessa jornada, o utilizador pode selecionar a jornada pretendida, e o sistema apresenta-lhe a jornada e os jogos desta.
Variações:	Os jogos poderão conter ou não conter resultados
Pós-Condição:	Nenhuma

Tabela 3: Descrição do caso de uso 2. Visualizar detalhes de um jogo

Caso de uso: Visualizar detalhes de um jogo

Actor:	Utilizador
Pré-Condição:	Jogo disponível para visualização
Descrição:	O utilizador seleciona um jogo que pretende visualizar e o sistema apresenta os detalhes do jogo selecionado
Variações:	O jogo selecionado poderá apresentar ou não apresentar detalhes ou apresentar apenas alguns detalhes
Pós-Condição:	Nenhuma

Tabela 4: Descrição do caso de uso 3. Visualizar histórico de jogos

Caso de uso: Visualizar histórico de jogos

Actor:	Utilizador
Pré-Condição:	Equipa associada ao utilizador
Descrição:	O sistema apresenta uma lista de todos os jogos realizados pela equipa que está associado
Variações:	A equipa poderá conter ou não conter histórico de jogos
Pós-Condição:	Nenhuma

Tabela 5: Descrição do caso de uso 4. Visualizar adeptos que assistiram ao jogo

Caso de uso: Visualizar adeptos que assistiram ao jogo

Actor:	Utilizador
Pré-Condição:	Visualizar detalhes de um jogo
Descrição:	Depois do utilizador selecionar o jogo pretendido, o sistema apresenta uma lista com todos os adeptos que assistiram ao jogo selecionado
Variações:	O jogo poderá conter ou não conter informação
Pós-Condição:	Nenhuma

Tabela 6: Descrição do caso de uso 5. Visualizar detalhes de um adepto

Caso de uso: Visualizar todos os adeptos

Actor:	Utilizador
Pré-Condição:	Adeptos existentes na base de dados
Descrição:	O sistema apresenta uma lista com todos os adeptos
Variações:	Nenhuma
Pós-Condição:	Nenhuma

Tabela 7: Descrição do caso de uso 6. Visualizar detalhes de um adepto

Caso de uso: Visualizar detalhes de um adepto

Actor:	Utilizador
Pré-Condição:	Adepto selecionado
Descrição:	Depois do utilizador selecionar o adepto pretendido o sistema verifica se o adepto está disponível para visualização, se estiver disponível o sistema apresenta os detalhes do adepto seleceionado.
Variações:	O adepto selecionado poderá não conter informação, ou não estar disponível para visualização.
Pós-Condição:	Nenhuma

Tabela 8: Descrição do caso de uso 7. Visualizar percentagem de adeptos que vão assistir ou não ao próximo jogo

Caso de uso: Visualizar percentagem de adeptos que vão ou não assistir ao próximo jogo

Actor:	Utilizador
Pré-Condição:	Nenhuma
Descrição:	O sistema apresenta um gráfico circular com a percentagem de utilizadores que assistem e não assistem ao jogo
Variações:	Os adeptos poderão ou não apresentar previsão de assistir ao próximo jogo
Pós-Condição:	Nenhuma

Tabela 9: Descrição do caso de uso 8. Visualizar histórico de assistências de um adepto

Caso de uso: Visualizar histórico de assistências de um adepto

Actor:	Utilizador
Pré-Condição:	Adepto selecionado
Descrição:	Depois do utilizador selecionar o adepto pretendido o sistema apresenta uma lista com o histórico de assistência do adepto selecionado
Variações:	O adepto poderá ter ou não um histórico de assistências
Pós-Condição:	Nenhuma

Tabela 10: Descrição do caso de uso 9. Visualizar adeptos com probabilidade personalizada

Caso de uso: Visualizar adeptos com probabilidade personalizada

Actor:	Utilizador
Pré-Condição:	Nenhuma
Descrição:	O ajusta o *slicer* de valores de probabilidades, disponibilizado pelo sistema que posteriormente irá apresentar os adeptos com as probabilidades compreendidas entre os valores ajustados pelo utilizador
Variações:	Nenhuma
Pós-Condição:	Nenhuma

Tabela 11: Descrição do caso de uso 10. Visualizar probabilidade de um adepto assistir ao próximo jogo

Caso de uso: Visualizar a probabilidade de um adepto assistir ao próximo jogo	
Actor:	Utilizador
Pré-Condição:	Nenhuma
Descrição:	O sistema apresenta ao utilizador uma probabilidade prevista
Variações:	O adepto poderá não conter uma probabilidade prevista
Pós-Condição:	Nenhuma

3.1.2 Arquitetura da aplicação

Como em todos os projetos, é necessário definir a arquitetura da aplicação para que permita aos programadores a facilidade no desenvolvimento. O Azure é o ponto de partida para a construção da aplicação, tendo que seguir algumas regras de utilização de cada serviço. A arquitetura da aplicação a construir terá que ser baseada nas necessidades da aplicação, onde fazem parte os serviços de base de dados, *Machine Learning* e apresentação de resultados - o PowerBi. Com isto, foi adotada uma arquitetura já existente e utilizada por muitos dos programadores que utilizam o serviço de *Machine Learning* do Azure. Esta arquitetura está dividida em três camadas: a camada de dados, a camada lógica e a camada de apresentação. A Figura 18 apresenta a arquitetura de todo o sistema da aplicação (markga, 2017).

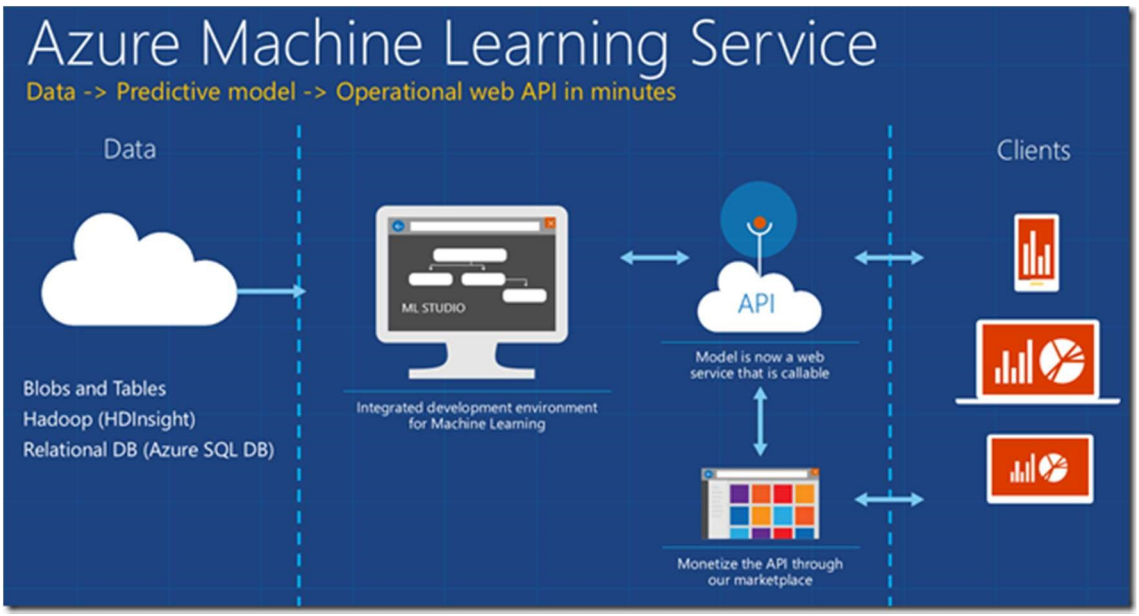


Figura 18 – Diagrama de fluxo do Azure ML (markga, 2017)

3.2 Gestão de base de dados

A segunda fase do projeto visa efetuar a gestão de base de dados, para que esta alimente os modelos de *Machine Learning* com dados de treino e permita guardar os dados necessários para efeitos de resultado. Este capítulo ajuda a perceber como a base de dados foi construída até ficar disponível para utilização por parte dos modelos preditivos.

3.2.1 Construção da base de dados

Antes de partir para a elaboração das tabelas necessárias para levar a cabo os objetivos do projeto, foi fornecido um ficheiro de *script* SQL com informação para a criação de uma base de dados (tabelas, dados e relações).

Iniciou-se o processo de criação de um servidor de base de dados no portal do Azure e, consequentemente, a execução do *script* para construir a base de dados. O acesso ao servidor poderá ser feito através de *software* ou linha de comandos, tendo sido escolhido o Microsoft SQL Server Management Studio (SSMS), pois oferece um ambiente integrado para efetuar a gestão de qualquer infraestrutura SQL, satisfazendo, assim, todas as necessidades de gestão da base de dados. Depois de efetuar as configurações necessárias de acesso ao servidor, abriu-se o ficheiro com o script, para dar início à construção da base de dados, no entanto, ocorreu um erro (Figura 19) que não permitiu iniciar a construção desta.

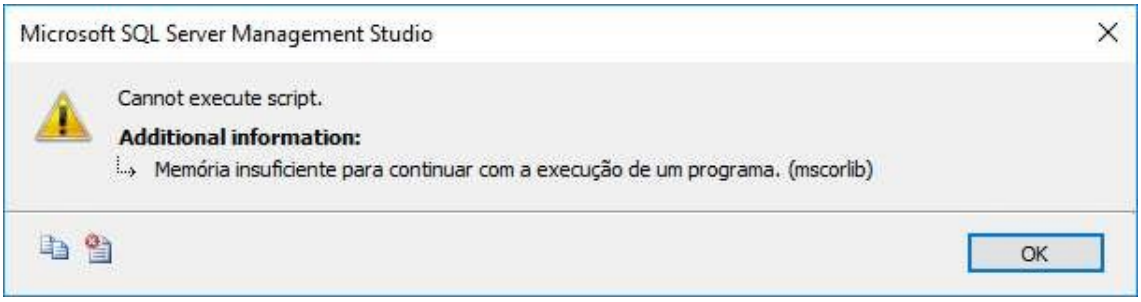


Figura 19 – Erro de memória insuficiente no SSMS

Depois de investigar a causa do erro (Microsoft, 2017d), verificou-se que o SSMS é um processo de 32 bits e está limitado a dois GB de memória. Este não tem memória suficiente para atribuir a resultados extensos, e impõe um limite artificial da quantidade de texto que pode ser apresentado por campo de base de dados na janela dos resultados. Foi, então, necessário encontrar soluções alternativas, pelo que o documento disponibilizava três soluções alternativas. Optou-se pela utilização da ferramenta `sqlcmd`, versão de 64 bits, que evita a restrição de memória provocada pelo processo do SSMS de 32 bits. O `sqlcmd` permite a execução de *querys* e ficheiros *script* SQL através da linha de comandos. O Anexo A contém uma lista de parâmetros permitidos pelo `sqlcmd`, que, conjugados, formam um comando a ser

executado pelo servidor de base de dados. Para ser possível a execução do ficheiro de *script* SQL, utilizou-se o comando da Figura 20Figura 20.

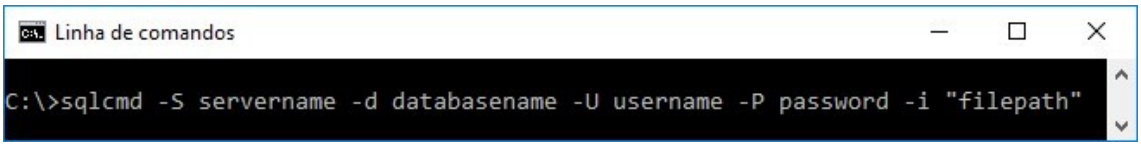


Figura 20 – Comando utilizado para executar o *script* para a criação da base de dados

A execução do *script* levou algum tempo até finalizar, um total de vinte e seis horas de execução. De seguida, efetuou-se uma análise à estrutura de dados, relações e dados de todas as tabelas.

3.2.2 Estrutura de dados (modelo relacional de dados)

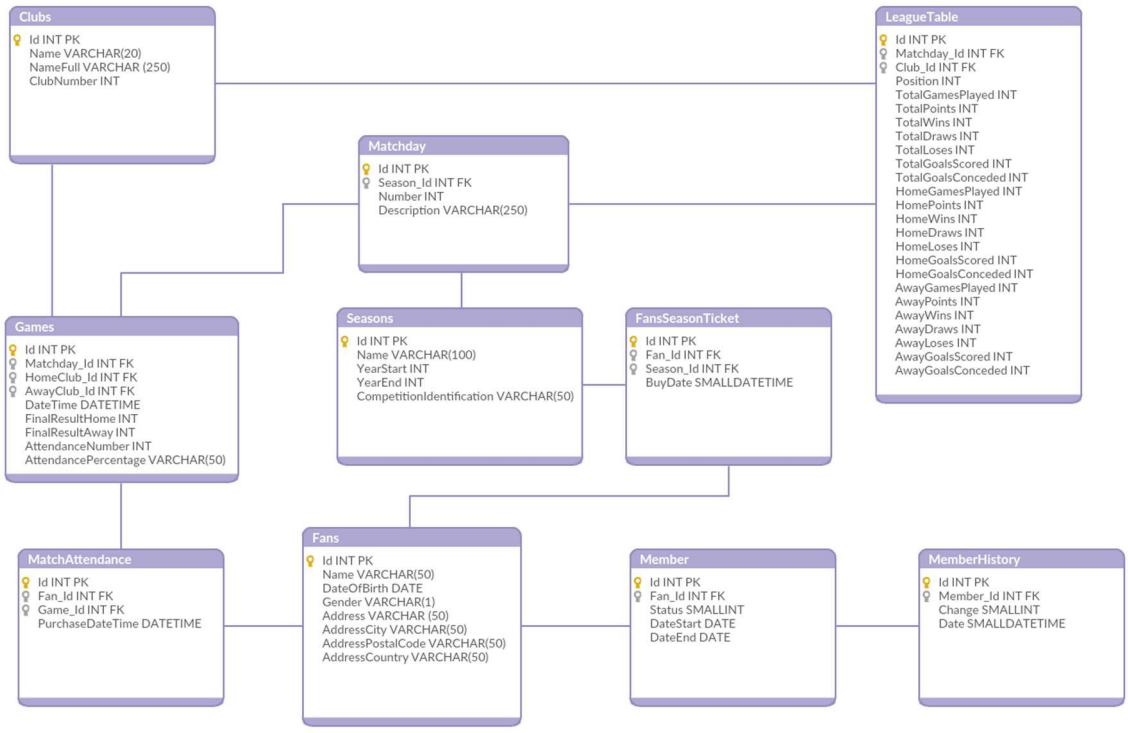


Figura 21 – Modelo relacional da base de dados gerada

A base de dados gerada através do *script* resultou na estrutura de dados presente na Figura 21. Esta é constituída por dez tabelas e cada uma tem o seu objetivo; a Tabela 12 resume a informação extraída da construção da base de dados.

Tabela 12: Resumo dos dados da base de dados gerada

Tabela	Registos	Descrição
Clubs	28	Clubes que estão envolvidos nas principais competições
Games	119	Jogos realizados pelas equipas (presentes na tabela Clubs)
MatchAttendance	3849607	Tabela com informação referente à assistência de adeptos em jogos
MatchDay	252	Jornada da temporada
Seasons	8	Tabela com informação referente ás temporadas do inicio e fim das competições
Fans	100000	Tabela referente a todos os adeptos para efeitos de previsão
FansSeasonTicket	0	A identificação dos adeptos que possuem bilhete anual
Member	100000	Identificação dos sócios pagantes do clube
LeagueTable	4236	Tabela para efeitos de representação da classificação detalhada de cada clube por jornada
MemberHistory	0	Tabela com alterações de estatuto de sócios

3.2.3 Modificações iniciais à estrutura

Como se pode verificar através da Figura 21, todos os nomes de tabelas e colunas seguem uma prática de escrita de palavras compostas, denominada *CamelCase*, onde cada palavra é iniciada com letra maiúscula e unida sem espaços. Apesar de ser muito utilizada em linguagens orientadas a objetos, esta torna-se menos percetível quando são utilizadas em contexto de base de dados, dependendo das preferências do gestor. Porém, o aspeto mais importante a considerar na escrita de palavras será a consistência, independentemente do esquema utilizado para a escrita.

O principal motivo para a modificação da escrita de palavras será, então, a facilidade de leitura destas. O *underscore* que também é uma prática de escrita muito utilizada foi o escolhido para aplicar nos nomes de colunas e tabelas da base de dados. Esta prática permite uma leitura rápida e coesa das palavras, pois consiste na utilização do "_" como agregador e onde todas as letras se encontram em minúsculo.

Também se verificou que não existiam colunas tanto para a obtenção, como para a escrita de resultados. Assim sendo, foram adicionadas as colunas `next_game_probability`, `next_game_result`, `next_game_result_timestamp` à tabela `fans`, com a finalidade de guardar a probabilidade de um adepto assistir ao próximo jogo, se ele vai ou não e a data e hora do resultado da previsão.

3.3 Desenvolvimento do modelo Machine Learning

A terceira fase do projeto envolve todo o processo de desenvolvimento de um modelo de *Machine Learning*. Este capítulo tem como objetivo demonstrar todos os passos efetuados para a criação de um modelo de *Machine Learning* capaz de efetuar a previsão de assistência dos adeptos.

3.3.1 Construção de um modelo no Azure Machine Learning Studio

Como referido anteriormente, o Azure Machine Learning Studio (ML Studio) é um ambiente de desenvolvimento que permite construir, testar e publicar soluções de *Machine Learning*, pelo que será demonstrado, de uma forma generalista, como se poderá construir um modelo através do ML Studio.

A construção de um modelo de *Machine Learning* no ML Studio divide-se normalmente em quatro partes: leitura de dados, tratamento de dados, aplicação de algoritmo e avaliação de resultados, conforme apresenta a Figura 22.

Figura 22 – Módulos para construção de um modelo padrão de Machine Learning Studio

A Leitura de dados poderá ser feita através de várias fontes, nomeadamente base de dados, ficheiros Excel e outros disponibilizados pelo ML Studio. Após a disponibilização dos dados para consulta, inicia-se a fase de tratamento de dados, uma das fases mais importante e complexa do modelo. Nesta fase, é efetuado um tratamento específico aos dados, como limpeza de dados vazios, seleção de tabelas para treino, aplicação de *querys* sobre dados, entre outros, de modo a criar as condições ideais para que o algoritmo possa efetuar os cálculos sem problemas.

O módulo da Divisão de Dados tem o mesmo propósito do nome, a divisão de dados, onde são divididos em percentagem, ou seja, uma percentagem parte para treino e a remanescente para avaliação. O Algoritmo e Módulo de treino formam um só, onde o módulo de treino depende sempre de um algoritmo, contendo a metodologia de aprendizagem dos dados. Nesta fase, é efetuada a aprendizagem da máquina, onde os dados serão treinados, para se conseguir extrair informação dos dados. Posteriormente, os dados treinados serão avaliados no Módulo de avaliação, comparando-os com os dados anteriormente divididos, resultando, assim, na previsão pretendida.

A Figura 23 apresenta uma visão mais abrangente de todos os módulos disponíveis no Azure Machine Learning Studio, onde também é referenciada a sequência destes para a construção de um modelo padrão.

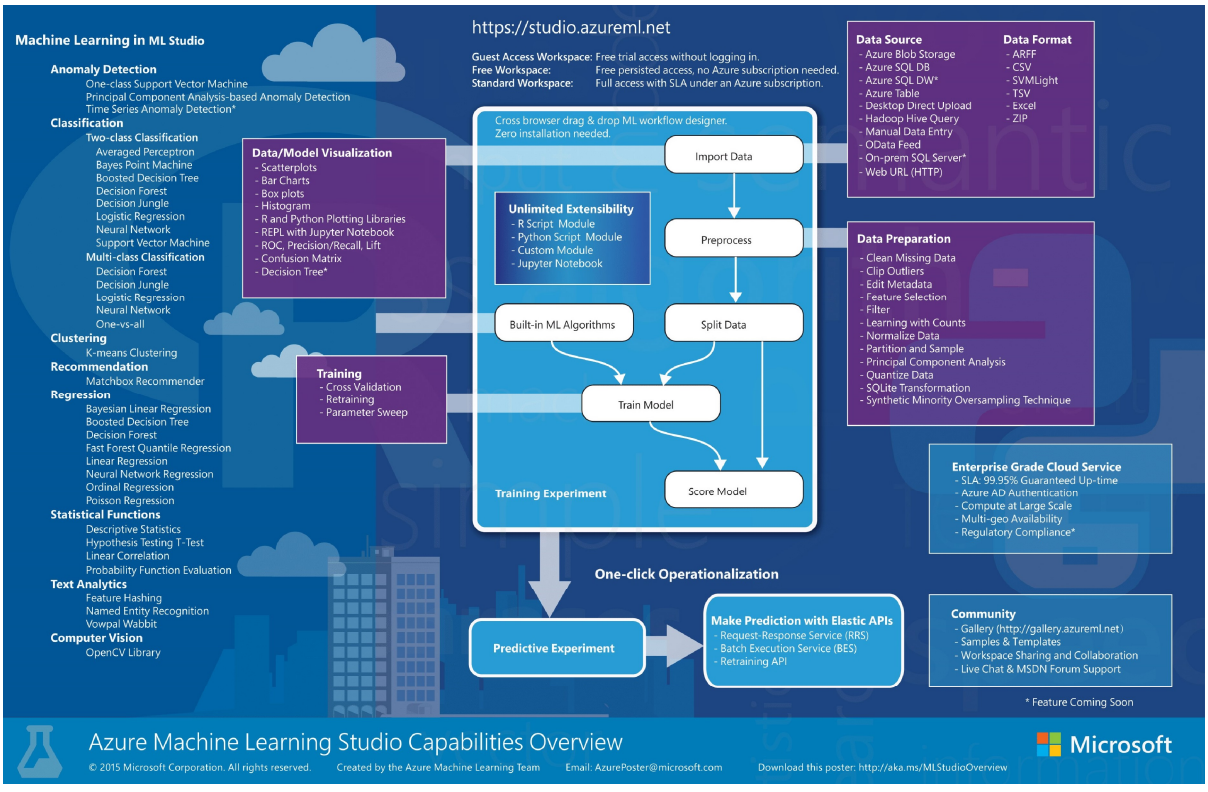


Figura 23 – Módulos disponíveis no Azure Machine Learning Studio (Ning et al., 2015)

3.3.2 Escolha de algoritmo

O processo de escolha do algoritmo não é tão linear quanto parece, este depende de alguns aspetos como o tamanho, a qualidade e a natureza dos dados. Passa também principalmente pelo desempenho deste face aos dados, o que, por sua vez, ter-se-á que experimentar alguns algoritmos, com o intuito de conseguir efetuar a comparação de acordo com as métricas pretendidas. Para além destes aspetos, o mais importante de todos será a finalidade do modelo e qual resultado se requer da execução deste. (Ericson et al., 2017b)

Começando então pelos dados disponíveis, podemos verificar que possuímos dados treinados, ou seja, a tabela `match_attendance` da base de dados indica que o adepto X foi ao jogo Y, ao qual remete a finalidade do modelo: a previsão da probabilidade de um adepto ir ao próximo jogo, através de um histórico de assistências. Com isto e com a definição dos tipos de AA, podemos confirmar que se trata da categoria aprendizagem supervisionada. O próximo passo será, então, decidir que algoritmo utilizar consoante o objetivo do projeto, que, neste caso se pretende obter uma probabilidade, concluindo que será necessário a utilização de algoritmos de regressão, contudo, poderemos também utilizar algoritmos de classificação, prevendo apenas se o adepto vai ou não assistir ao jogo ou mesmo os dois combinados (se vai ou não e a probabilidade).

A Figura 24 apresenta um fluxo para a escolha de um algoritmo no Azure Machine Learning Studio, na qual permite selecionar um algoritmo dependendo da natureza dos dados e o objetivo da utilização deste.

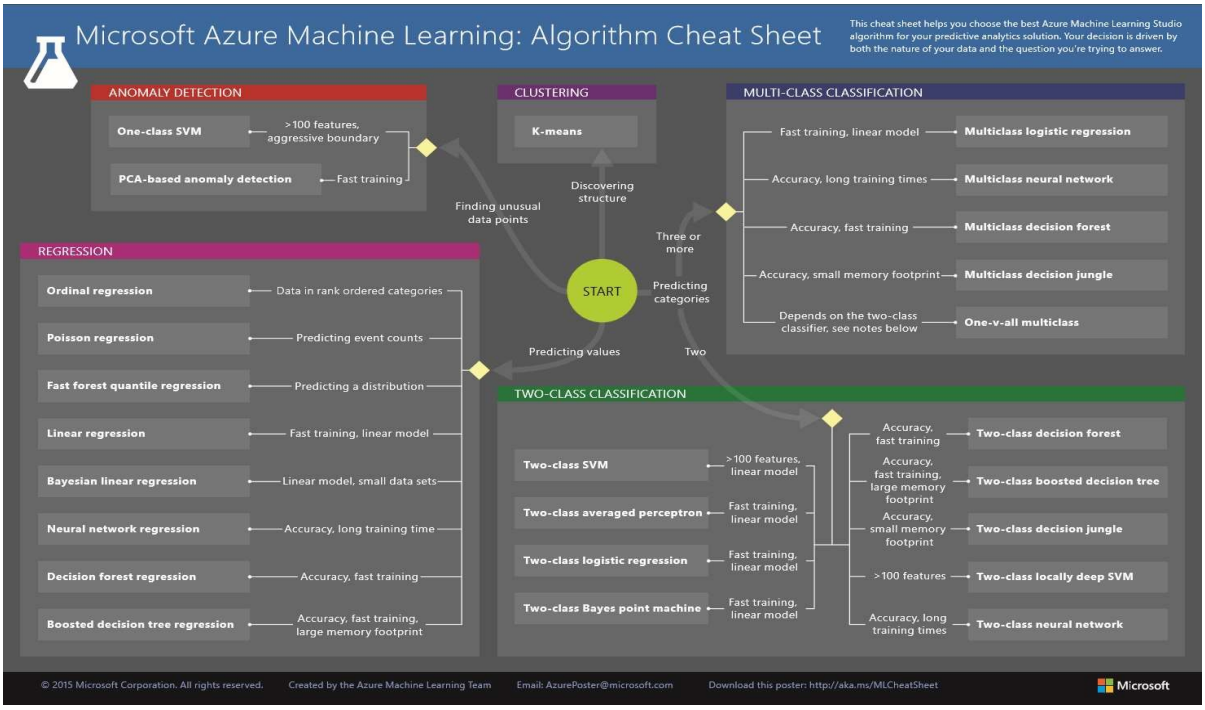


Figura 24 – Fluxo para a escolha de um algoritmo no Azure Machine Learning Studio (Ericson et al., 2017c)

3.3.3 Modelo desenvolvido

O modelo desenvolvido foi construído com base nos projetos anteriormente mencionados. O facto de o ambiente de desenvolvimento ser modular, necessita de um conhecimento prévio das metodologias de *Machine Learning*, bem como da sua utilização.

3.3.3.1 Leitura/Entrada de dados

O processo de construção de um modelo inicia-se sempre com a leitura dos dados de entrada. Desta forma, foram selecionadas as tabelas necessárias para a realização da previsão de assistência de cada adepto, com base no seu histórico. As tabelas `games`, `match_attendance` e `fans` foram adicionadas através de módulos ao modelo, como apresenta a Figura 25.

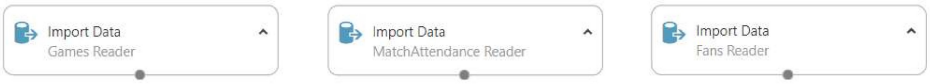


Figura 25 – Módulos para leitura de dados através da Azure SQL Database

Para importar os dados das tabelas pretendidas é necessário efetuar uma pequena configuração que é realizada através do separador lateral reservado para as propriedades de cada módulo. Para o módulo em causa, e como a origem dos dados é uma base de dados SQL (Azure SQL Database), são necessários os dados de acesso, como o nome do servidor, o nome de base de dados, utilizador e *password* da base de dados e ainda a *query* para retornar os dados das tabelas pretendidas.

A linguagem das *querys* deste módulo é diferente da base de dados, utilizando, assim, outra muito semelhante - o SQLite. As *querys* utilizadas nestes módulos são *querys* básicas de `selects` das tabelas em causa. Nestas propriedades ainda é permitida a utilização de resultados em cache, ou seja, permitem que se utilize dados carregados em memória, para que não seja necessário efetuar o acesso e execução de *querys* à base de dados. É importante salientar que, para outras origens de dados, as configurações poderão ser diferentes ou um pouco semelhantes. A Figura 26 apresenta a forma de configurar o acesso à base de dados e introdução de *query* para obtenção de dados.

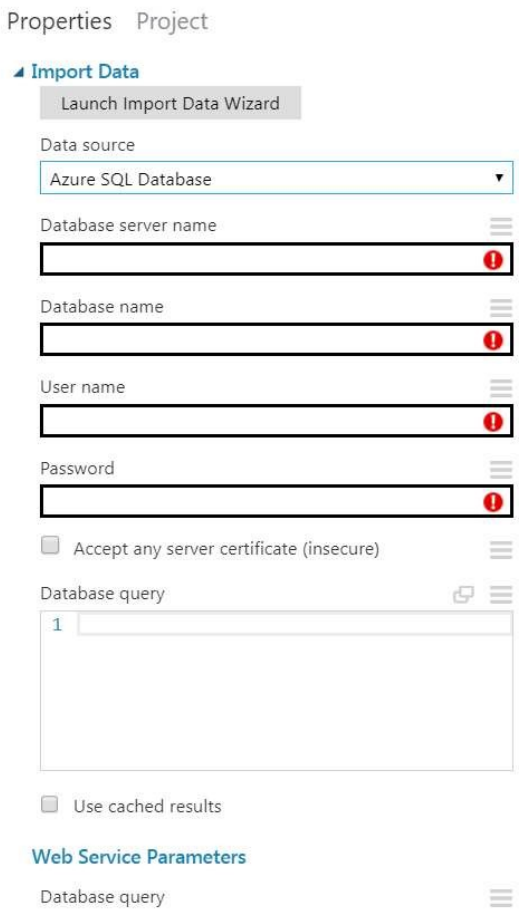


Figura 26 – Configuração do módulo de leitura de dados através de Azure SQL Database

3.3.3.2 Tratamento de dados

O tratamento de dados é a parte mais importante da construção dos modelos de *Machine Learning*. É nesta parte que os dados são tratados/transformados para que o algoritmo os possa analisar e treinar sem qualquer problema. Com isto, os dados de entrada sofrem algumas alterações antes de chegar à parte de treino, como a seleção de, apenas, colunas relevantes para a realização dos cálculos, remoção de valores a branco ou a vazio, realização de cálculos, entre outros.

Figura 27 – Modelo construído até a parte de transformação de dados

Conforme a sequência de acontecimentos na Figura 27Figura 27, a primeira transformação a realizar é a criação de uma tabela combinada entre `fans` e `games`, onde cada `fan` terá associado todos os jogos existentes. Este módulo (`Apply SQL Transformation`) permite a execução de *querys* em SQLite, mas, em algumas situações, dificulta a escrita de *querys*, devido à limitação da sua sintaxe. Um exemplo da sua limitação é esta transformação, onde existe a necessidade da utilização de um `FULL JOIN`, sintaxe que não é permitida. Este módulo permite a transformação de, no máximo, três tabelas, identificadas como t1, t2 e t3, sendo que t1 é a ligação mais à esquerda, t2 a do meio e t3 a mais a direita, onde a Figura 28 apresenta a transformação em causa.

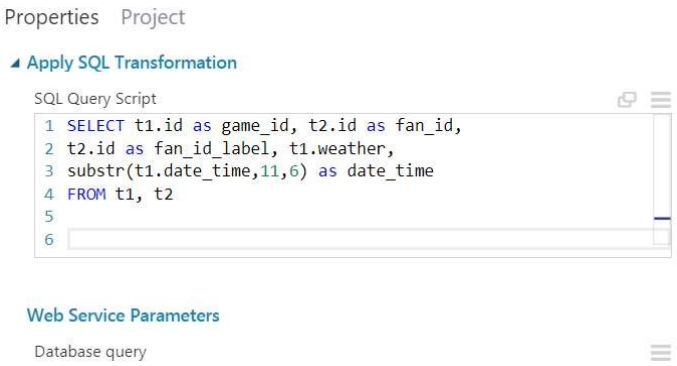


Figura 28 – Transformação dos dados, junção das tabelas `games` e `fans`

A transformação descrita na Figura 29 tem o objetivo verificar se o `fan` foi ao jogo "x", de acordo com a tabela de histórico de assistências (`match_attendance`), onde estão presentes apenas as idas aos jogos, finalizando, então, a junção das três tabelas.

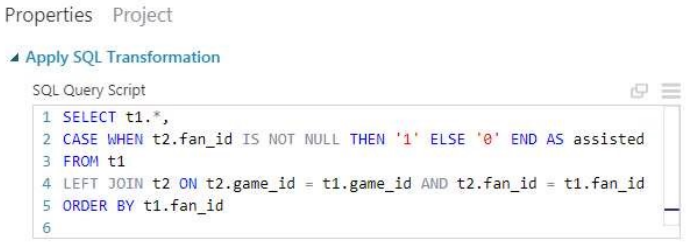


Figura 29 – Transformação dos dados, verificação se adepto foi ao jogo

Para a realização da previsão será necessário que os dados sejam divididos, uma percentagem para treino e outra para efetuar a previsão. Os próximos dois módulos de transformação SQL poderiam ser substituídos por um módulo, designado `Split Data`, que divide os dados em percentagem. Este não foi utilizado devido ao facto da divisão dos dados ser automática, e não permitir o controlo dos dados a serem distribuídos, optando, assim, por dois módulos de transformação SQL, onde o da esquerda obtém os dados para efetuar a previsão e o da direita os dados para efetuar o treino. A previsão dos dados será efetuada através de um jogo, ou seja, houve a necessidade de criar alguns jogos apenas para o propósito da previsão. Estes jogos estão com `id` menor que zero e o valor `assisted` a um (significa que vai a jogo), pois o pretendido é prever se o adepto vai a jogo (valor um). A Figura 30 e Figura 31 apresentam as *querys* para a divisão dos dados.

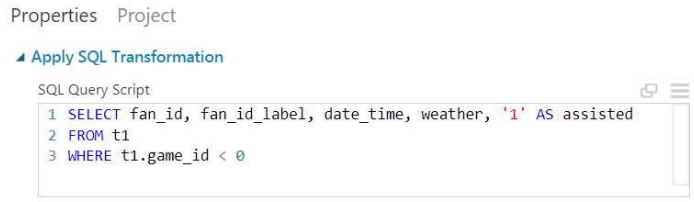


Figura 30 – Transformação dos dados, divisão dos dados para previsão

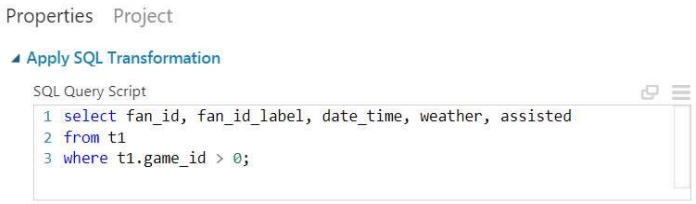


Figura 31 – Transformação dos dados, divisão dos dados para treino

Os dados transformados estão quase prontos para que o algoritmo os possa treinar sem qualquer problema, no entanto, se este os treinar identificá-los-ia como um só, ou seja, o resultado seria igual para todos os `fans`, o que não vai ao encontro dos objetivos. Desta forma, foi adicionado outro módulo, o `Build Count Transform`, que efetua uma análise aos dados e cria uma tabela de contagem destes, de forma a agrupá-los, efetuando, assim, a contagem de dados correspondente a cada `fan_id`. A Figura 32 apresenta a configuração para este módulo.

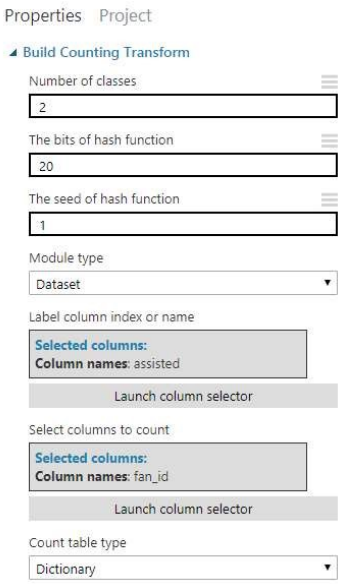


Figura 32 – Configuração do módulo `Build Count Transform`

O módulo `Modify Count Table Parameters` altera a forma como os recursos são gerados a partir de uma tabela de contagem que por sua vez transmite ao módulo `Apply Transformation` o modo de alteração a efetuar nos dados de entrada, agrupando-os assim por `fan_id`.

A última transformação a realizar de modo a que o algoritmo utilize apenas valores essenciais para o cálculo da previsão, foi introduzido o módulo `Edit Metadata`, que coloca uma *flag* na coluna que se pretende excluir dos cálculos a realizar. Desta forma, o módulo foi configurado para excluir a coluna `fan_id_label` do cálculo como apresenta a Figura 33.

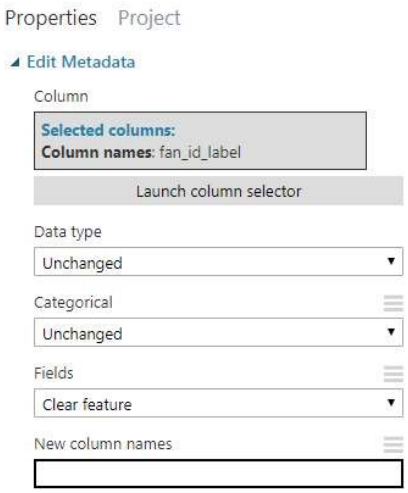


Figura 33 – Configuração do módulo `Edit Metadata`

3.3.3.3 Algoritmo e modelo final

A escolha de um algoritmo foi baseado em testes realizados aos vários algoritmos disponíveis (Figura 23), que poderiam corresponder às expectativas. Depois de testados alguns algoritmos, foram, então, considerados os módulos dos algoritmos `Two-Class Logistic Regression`, `Two-Class Support Vector Machine` e `Two-Class Averaged Perceptron`. Todos estes módulos seguem as características da aprendizagem supervisionada, ou seja, apenas aceitam conjuntos de dados, rotulados/classificados de modo a efetuar o treino através destes.

O módulo `Two-Class Logistic Regression` permite criar modelos de regressão logística, onde se pretende efetuar uma previsão de probabilidades, com apenas duas classes/categorias de resultado final. A regressão logística é uma metodologia estatística muito conhecida e é adotada para modelar muitos problemas (Microsoft, 2017e).

O módulo `Two-Class Support Vector Machine` é baseado no algoritmo *Support Vector Machine*, útil para efetuar a previsão entre dois resultados possíveis que, por sua vez, dependem de variáveis contínuas ou categóricas (Microsoft, 2017f).

O módulo `Two-Class Averaged Perceptron` é uma versão simples de uma rede neuronal. Os dados são classificados em várias saídas possíveis com base numa função linear, que posteriormente são combinados com um conjunto de pesos derivados do vetor de características. Este modulo é adequado para aprender padrões linearmente separáveis, enquanto as redes neuronais podem identificar padrões de classes mais complexos (Microsoft, 2017g).

A seleção do algoritmo para o cálculo da previsão foi efetuada com base nos testes realizados aos algoritmos. Para estes testes foram considerados, fundamentalmente, os aspetos relevantes para o cliente, nomeadamente tempo de execução, grau de acerto e probabilidades mais próximas da percentagem de assistência. Apesar do algoritmo apresentado no modelo final (`Two-Class Logistic Regression`), ainda não foi possível conseguir perceber qual será o mais indicado para efetuar a previsão.

Depois dos dados serem treinados pelo `Train model` com as regras do algoritmo, o `Score Model` recebe, então, esses dados treinados, onde fará a previsão para cada "fan". Consequentemente, os dados previstos são utilizados pelo módulo `Export Data` que efetua o armazenamento dos mesmos na base de dados. Este módulo necessita de algumas configurações a realizar, para que os dados sejam armazenados, como colunas a enviar, na tabela que receberá os dados e respetivas colunas, conforme se apresenta na Figura 34Figura 34.

Figura 34 – Configuração do módulo `Export Data`

De forma a constatar se a previsão foi efetuada corretamente, o módulo `Evaluate Model` efetua uma comparação dos dados reais com dados previstos, resultando, assim, no grau de acerto do algoritmo. Concluída a construção do modelo, este é apresentado numa versão final na Figura 35.

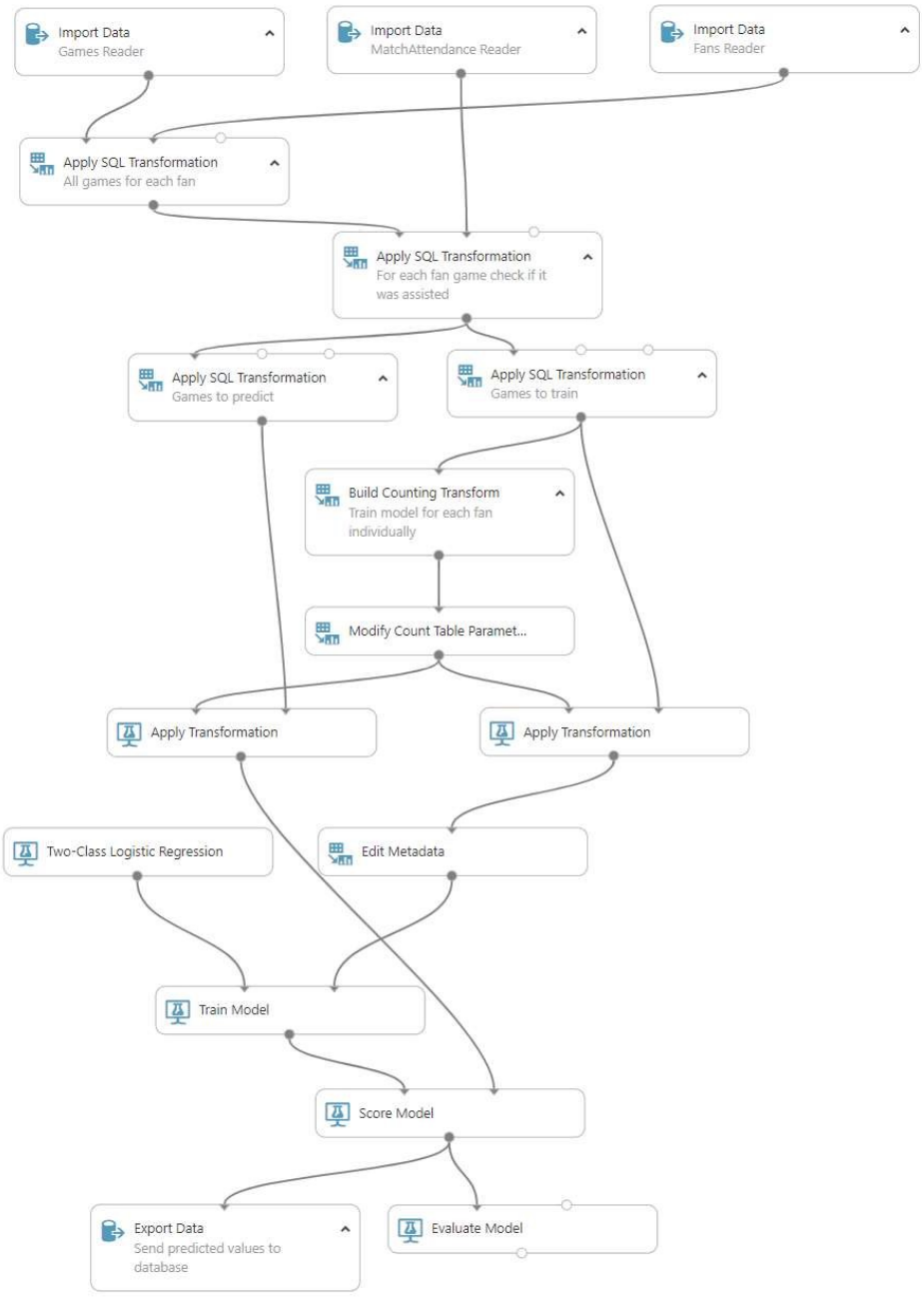


Figura 35 – Modelo final desenvolvido para a realização da previsão de assistências

3.3.4 Modificações na estrutura da base de dados

Com o desenrolar do desenvolvimento do projeto, a estrutura da base de dados foi sofrendo algumas alterações, de modo a satisfazer as necessidades do modelo de Machine Learning ou por indicações do cliente (supervisor).

As alterações envolveram as tabelas `games`, `fans`, `match_attendance` e ainda a criação da tabela `tmp_ml_result`.

A primeira alteração efetuada na estrutura da base de dados envolve a tabela "games" que, de acordo com o estudo "Fatores de influência em adeptos de desporto" presente no capítulo 2.1.2, verificou-se que existiam alguns fatores que seriam muito importantes a ser considerados para a previsão de assistências, nomeadamente o preço dos bilhetes, a meteorologia e a importância do jogo. Foram então adicionadas as colunas `ticket_price`, `weather` e `importance` à tabela `games` e gerados valores aleatórios para estas três colunas.

A tabela `games` sofreu ainda mais uma alteração, com o objetivo de permitir ao utilizador da aplicação a visualização da forma da equipa anfitriã, mais especificamente o resultado dos últimos cinco jogos. Foi, então, adicionada a coluna `sequence`, que, por sua vez, não contém os dados pretendidos. Para preencher todos os jogos foi desenvolvido um *script* SQL, que efetua a leitura de todos os jogos e classifica com "W" (vitória), "D" (empate) e "L" (derrota) cada jogo antes do atual, até ao máximo de cinco jogos e, de seguida, atualiza a tabela `games`. A Figura 36Figura 36 demonstra o script desenvolvido para o preenchimento de dados da coluna `sequence`.

```
DECLARE @cnt INT = 119
DECLARE @curr_id INT = 14
DECLARE @sequences AS NVARCHAR(9) = ''

WHILE @cnt >0
BEGIN

    SELECT @sequences = @sequences + result FROM (
        SELECT TOP (5)
            CASE
                WHEN g.final_result_home > g.final_result_away
                THEN 'W-'
                ELSE
                    CASE
                        WHEN g.final_result_home = g.final_result_away
                        THEN 'D-'
                        ELSE 'L-'
                    END
            END AS result
        FROM games g
        WHERE g.id < @curr_id and g.id >0
        ORDER BY g.id DESC) t

    IF LEN(@sequences)<9 AND LEN(@sequences)>0
    SET @sequences = SUBSTRING(@sequences, 1, LEN(@sequences)-1);

    UPDATE games SET sequence = @sequences WHERE id = @curr_id

    Print(CAST(@curr_id AS VARCHAR) + '->' + @sequences);

    SET @cnt = @cnt -1;
    SET @curr_id = (SELECT TOP (1) id FROM games WHERE id > @curr_id);
    SET @sequences = '';
END;
```

Figura 36 – Script para o preenchimento de dados da coluna `sequence`

A alteração relativa à tabela "fans" está ligada às alterações efetuadas na tabela `match_attendance`, visto que foi requisitada a percentagem de assistência de cada adepto (Figura 37) para efeitos de comparação com a probabilidade calculada. Desta forma, foi adicionada a coluna `assistance_percentage` à tabela `fans`, e a adição de um *trigger* à tabela `match_attendance`, que atualiza o valor existente na coluna `assistance_percentage` da tabela `fans` quando é efetuada alguma operação à tabela

`match_attendance` (*insert*, *update* ou *delete*). O valor subtraído na contagem de jogos refere-se à existência de um jogo que não entra para contabilização de jogos (jogo utilizado para efetuar a previsão com id = -1).

```
CREATE TRIGGER UPDATE_FANS_ASSIST_PERCENTAGE
   ON  [dbo].[match_attendance]
   AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    declare @games_num int;

    set @games_num = (SELECT Count(*) FROM games) -1;

    UPDATE fans SET assistance_percentage =
    ROUND((CAST(
        (SELECT count(*)
        FROM match_attendance
        WHERE fan_id = fans.id)
        as FLOAT)
        /CAST(@games_num as FLOAT))*100,2)
END
GO
```

Figura 37 – T*rigger* para recalcular a percentagem de jogos assistidos por cada adepto

A última alteração realizada foi a criação da tabela `tmp_ml_result` com as colunas `fan_id`, `scored_result` e `scored_probability`. Esta tabela tem o propósito de guardar temporariamente os resultados da execução do algoritmo de *Machine Learning*. Como foi referido anteriormente, o módulo de exportação de dados necessita de uma tabela para inserir dados e não realizar a atualização de dados já existentes. Os dados exportados pelo Azure sofrem uma alteração imediata quando são introduzidos na tabela. Isto deve-se ao facto de estes serem "transportados" para a tabela `fans`, pois o "transporte" dos dados é efetuado através de um *trigger* (Figura 38) desenvolvido para o efeito, que obtém os dados de cada linha da tabela e efetua uma atualização aos dados existentes na tabela `fans`. Depois, a linha em causa será eliminada, para que não haja incoerência de dados.

```
CREATE TRIGGER UPDATE_FANS_NEXT_GAME_COLUMNS
   ON  [dbo].[tmp_ml_result]
   AFTER INSERT
AS
BEGIN
    declare @tmp_fan_id int;
    declare @tmp_scored_probability float;
    declare @tmp_scored_result int;

    select @tmp_fan_id=input.fan_id from inserted input
    select @tmp_scored_probability=input.scored_probability from inserted input
    select @tmp_scored_result=input.scored_result from inserted input

    UPDATE Fans SET next_game_result = CASE WHEN @tmp_scored_result = 0 THEN 'NO' ELSE 'YES' END,
                    next_game_probability = ROUND(CAST(@tmp_scored_probability*100 as FLOAT),2),
                    next_game_result_timestamp = CURRENT_TIMESTAMP WHERE id = @tmp_fan_id
    DELETE FROM tmp_ml_result WHERE fan_id = @tmp_fan_id
END
GO
```

Figura 38 – *Trigger* que atualiza o valor da probabilidade calculada pelo algoritmo

3.4 Power BI

Como referido anteriormente o Power BI foi a ferramenta selecionada para a visualização de resultados. Assim foi desenvolvido um *dashboard* de modo a satisfazer os casos de uso apresentados no capítulo 3.1.1, onde esta apresenta uma lista de todos os adeptos com a probabilidade prevista e a percentagem de assistências, um gráfico circular com valores de assistências, tabela com histórico de jogos e uma secção de detalhes do adepto. De modo a filtrar os resultados, estão disponíveis um slicer para o ajuste das probabilidades e também um filtro para apresentar apenas os adeptos que assistem ou não assistem ao próximo jogo. A Figura 39 apresenta o *dashboard* desenvolvido, destacando o adepto com o `Id = 1`.

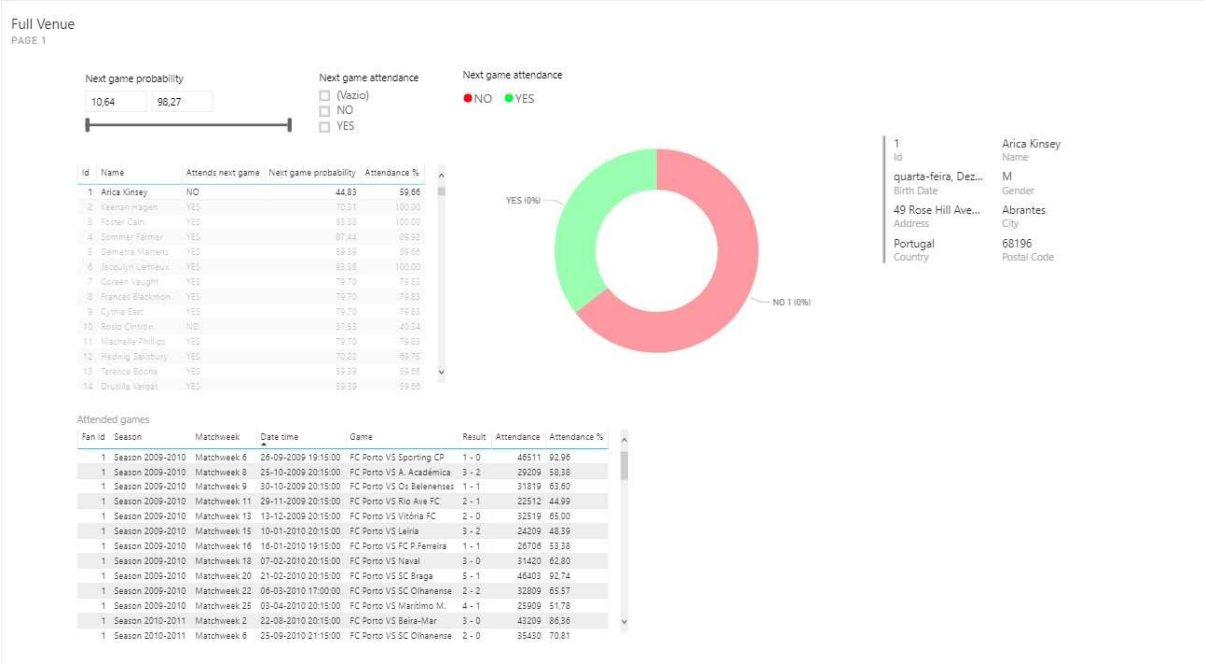


Figura 39 – *Dashboard* desenvolvido através do Power BI

4 Testes computacionais

Este capítulo tem como objetivo mostrar os resultados obtidos do trabalho realizado. Os resultados de qualquer pesquisa/estudo são a reflexão de todo o trabalho desenvolvido, que visa demonstrar a comparação entre as várias soluções possíveis. Assim, serão apresentadas análises aos resultados obtidos do impacto dos diferentes fatores na previsão e também qual o melhor algoritmo de desempenho, de acordo com métricas definidas.

Os testes apresentados estão divididos em dois aspetos, o primeiro utiliza os fatores (Tabela 13) meteorologia e hora de jogo para efeitos de aprendizagem, enquanto que o segundo utiliza os mesmos mais o preço do bilhete e a importância do jogo. Ambos os testes utilizam os algoritmos `Two-Class Logistic Regression`, `Two-Class Averaged Perceptron` e `Two-Class Support Vector Machine` para realizar a previsão.

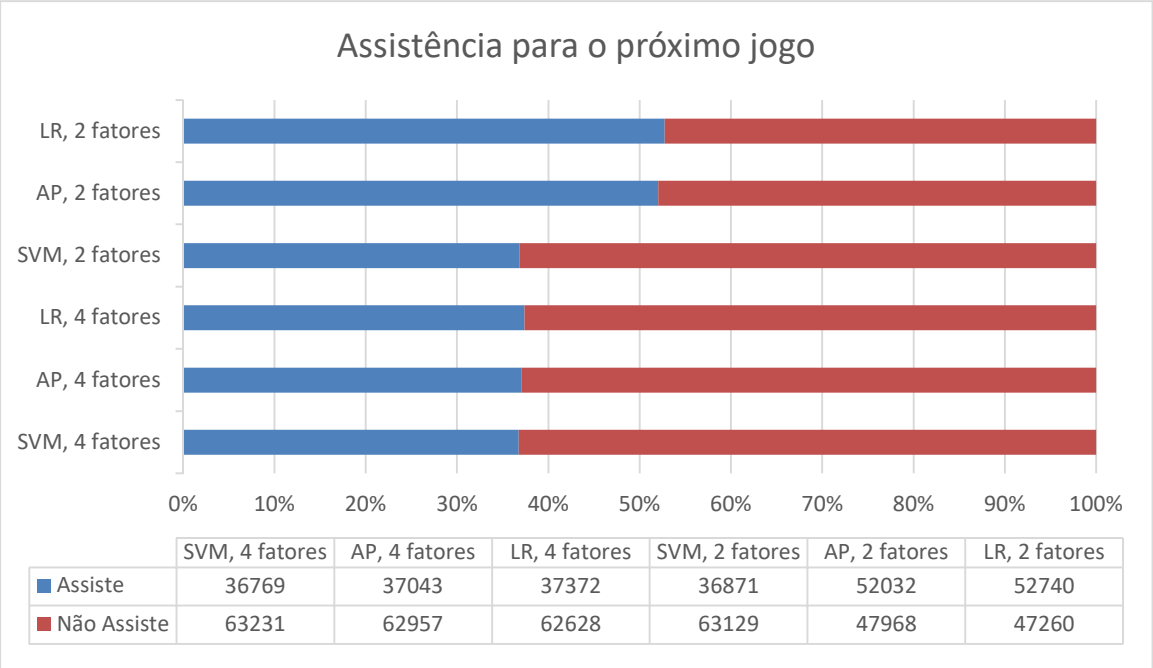
Em todos os testes, serão apresentadas a percentagem de acerto do algoritmo e quantas pessoas assistem ou não ao próximo jogo, para um total de 100 000 `fans`. Para a percentagem de acerto do algoritmo será utilizado o último jogo que contêm informações de assistência, com as características: hora de jogo = 20:15, preço do bilhete = `LOW`, meteorologia = `SUN`, importância do jogo = `LOW`. De modo a sumarizar o capítulo, será efetuada uma análise dos resultados obtidos.

Tabela 13: Fatores de aprendizagem e valores possíveis

Fator	Valores possíveis
Meteorologia	Pleasant, Rain, Sun, Wind
Hora de jogo	Hora do jogo com o formato HH:MM
Preço do bilhete	High, Normal, Low
Importância do jogo	High, Normal, Low

Para facilitar a leitura dos gráficos, foram definidas as abreviações dos algoritmos e fatores. Os fatores meteorologia e hora de jogo são definidos como 2 fatores, os fatores meteorologia, hora de jogo, preço do bilhete e importância do jogo são definidos como 4 fatores. Relativamente aos algoritmos, o `Two-Class Logistic Regression` foi definido como LR, o `Two-Class Averaged Perceptron` foi definido como AP e o `Two-Class Support Vector Machine` como SVM.

O Gráfico 1 apresenta a assistência para o próximo jogo, organizado por algoritmo e respetivo fator e ainda uma tabela com os resultados mais específicos. Os resultados apresentados neste gráfico, apenas demonstram os `fans` que assistem ao próximo jogo, no entanto é considerado que um `fan` assiste ao próximo jogo se a probabilidade gerada for superior a 50%. Neste gráfico pode-se verificar que existe uma diferença entre os algoritmos LR e AP utilizando dois fatores e os restantes.



	SVM, 4 fatores	AP, 4 fatores	LR, 4 fatores	SVM, 2 fatores	AP, 2 fatores	LR, 2 fatores
Assiste	36769	37043	37372	36871	52032	52740
Não Assiste	63231	62957	62628	63129	47968	47260

Gráfico 1 – Assistência para o próximo jogo por algoritmo e fator

O Gráfico 2 apresenta a percentagem de acerto - o valor previsto é igual ao valor real - dos algoritmos com os respetivos fatores, podendo-se verificar a existência de uma diferença dos algoritmos LR e AP com dois fatores para com os restantes. Para completar este gráfico, a Tabela 14 apresenta a divisão da percentagem de acerto pelo número de `fans`, que assiste ou não ao próximo jogo, apresentando também as probabilidades médias para cada aspeto.

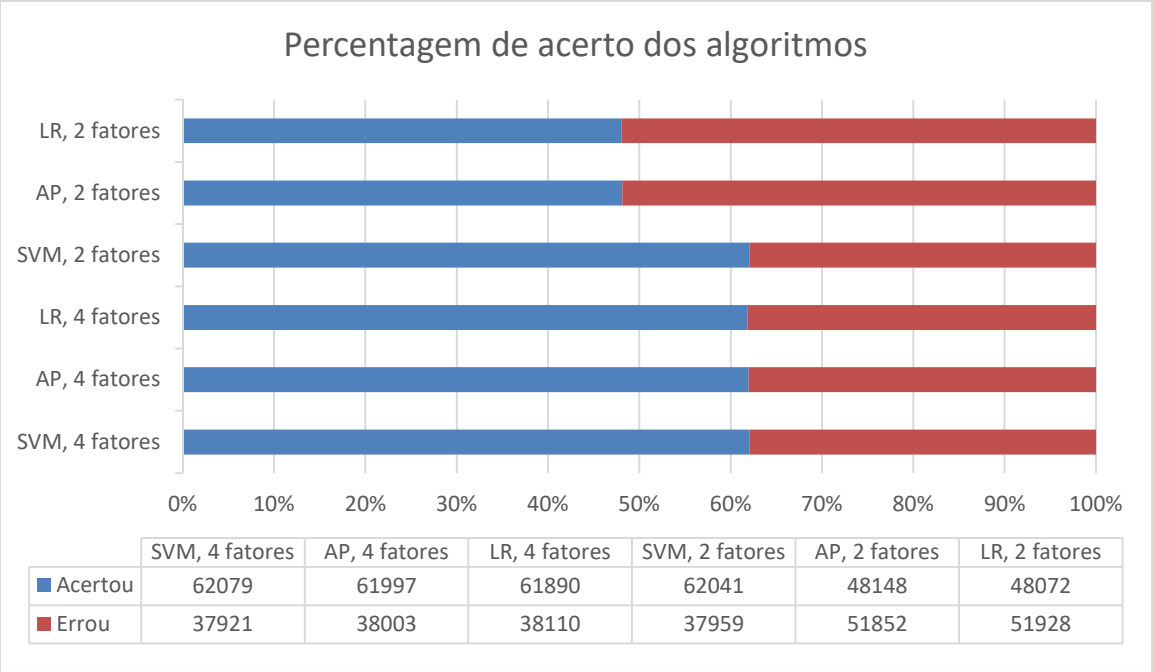


Gráfico 2 – Percentagem de acerto dos algoritmos com respetivos fatores

Tabela 14: Divisão da percentagem de acerto pelo número de `fans` que assiste ou não ao próximo jogo e respetivas probabilidades médias

Algoritmo e fator	**Acertou e assiste (Probabilidade média)**	**Acertou e não assiste (Probabilidade média)**	**Errou e assiste (Probabilidade média)**	**Errou e não assiste (Probabilidade média)**
LR, 2 fatores	21636 (87.23%)	26436 (41%)	20824 (41%)	31104 (62%)
AP, 2 fatores	21320 (87.26%)	26828 (40%)	21140 (39.90%)	30712 (60.5%)
SVM, 2 fatores	20686 (86.81%)	41355 (38.36%)	21774 (35%)	16185 (66.7%)
LR, 4 fatores	20861 (87.5%)	41029 (41.33%)	21599 (37.9%)	16511 (67.55%)
AP, 4 fatores	20750 (86.9%)	41247 (40.4%)	21710 (37.1%)	16293 (66.77%)
SVM, 4 fatores	20654 (86%)	41425 (37.73%)	21806 (34%)	16155 (65.82%)

Conforme os resultados apresentados anteriormente, podemos verificar que os testes realizados apresentam bastantes semelhanças nos algoritmos SVM com dois e quatro fatores, LR com quatro fatores e AP com quatro fatores. Os valores relativos à percentagem de acerto e o número de assistências, apresentam valores muito aproximados, com aproximadamente 62% de acertos e aproximadamente 37% de assistência para o próximo jogo. Os algoritmos LR e AP com dois fatores possuem semelhanças entre si, mas apresentam diferenças para os resultados dos algoritmos mencionados anteriormente, onde a percentagem de acerto é aproximadamente 48% e terá uma assistência para o próximo jogo de aproximadamente 52%.

De acordo com as probabilidades médias apresentadas, estas não diferem muito de algoritmo para algoritmo, quando estes efetuam uma previsão certa, mas pelo contrário, quando a previsão é errada verifica-se uma diferença um pouco mais acentuada.

Através da percentagem de acerto, verifica-se que os algoritmos têm valores mais elevados quando possuem mais fatores, à exceção do SVM, que apresenta ligeiras diferenças quando utiliza dois ou quatro fatores, mas notando-se também que quando este utiliza quatro fatores a percentagem de acerto aumenta. Com isto, podemos sintetizar que o nível de acerto é mais elevado com mais fatores, no entanto deve-se também considerar a utilização de mais fatores para efeitos de aprendizagem, não tornando esta síntese como definitiva, de modo a perceber a tendência do aumento de fatores.

Sem esquecer outro aspeto importante, o tempo de execução do modelo com os respetivos algoritmos, também não possuem grandes diferenças, na qual o SVM possui o melhor registo com dezoito minutos, seguido do LR com dezanove minutos e por último o AP com vinte e um minutos. Durante o tempo total de execução, o modelo despende aproximadamente quinze a dezasseis minutos para efetuar a preparação dos dados, um processo moroso devido a grande quantidade destes.

Finalizando a análise aos testes realizados, podemos verificar que o algoritmo SVM destaca-se em relação aos restantes, que independentemente do número de fatores apresenta uma percentagem de acerto mais alta.

5 Conclusão

Neste capítulo será descrito uma síntese de todo o trabalho realizado, se todos os objetivos foram atingidos, de que modo este trabalho poderá contribuir para o desporto, limitações do trabalho desenvolvido e a proposta de melhorias para o futuro. Apresentando uma proposta reflexiva e problematizante, partiu-se do pressuposto de que as questões iniciais e aquilo que fora planeado numa primeira fase não eram elementos estáticos, pois foram reajustados no decorrer do processo.

Este trabalho surgiu através da necessidade de dar suporte aos clubes de futebol de modo a prever o comportamento dos seus adeptos. Assim o desenvolvimento deste contribui para uma melhoria no desporto, através de dar informação aos clubes sobre as intenções dos adeptos assistirem aos jogos. Através de estratégias de marketing os clubes poderão atuar sobre os adeptos, aumentando assim a taxa de assistências, o que irá contribuir para uma melhoria do desporto.

Este trabalho de mestrado tem como objetivo efetuar a previsão de assistência nos estádios de futebol. Para tal, foi necessário recorrer à aprendizagem das metodologias e tecnologias que o sistema iria requerer para ser desenvolvido. Este trabalho começou então com a inicial aprendizagem de metodologias e técnicas de Aprendizagem Automática, passando pela análise de algoritmos e suas aplicações. Foram analisados projetos relacionados sobre o tema, assistência em eventos desportivos, e posteriormente iniciar a fase de desenvolvimento. Todos os componentes definidos na arquitetura foram implementados na *cloud* Azure, um serviço *online* que permite alojar várias ferramentas/tecnologias.

O aspeto mais importante de todo o trabalho desenvolvido é o modelo de Machine Learning desenvolvido, a componente que permite efetuar as previsões de assistências. Este é composto por uma unidade de leitura de dados, processamento dos dados e calculo de resultados, uma estrutura matemática capaz de classificar novos dados. Nesta estrutura foram

considerados alguns algoritmos para a realização de cálculos matemáticos, nomeadamente o `Two-Class Logistic Regressi`on, `Two-Class Averaged Perceptron` e `Two-Class Support Vector Machine`, estando disponível um guia detalhado de como contruir um modelo de *Machine Learning* para este trabalho.

Um dos objetivos deste trabalho passa também pela análise de qual o melhor algoritmo para efetuar a previsão. De acordo com o estudo computacional apresentado, o algoritmo Support Vector Machine destacou-se dos outros, tendo este melhores percentagens de acerto e mais consistência. No entanto devem ser tomados em conta outros algoritmos não considerados para esta primeira versão do projeto.

Em último, a apresentação dos resultados através do Power Bi, na qual é possível vislumbrar as probabilidades previstas pelo modelo de *Machine Learning* desenvolvido, bem como histórico de jogos e informação de cada adepto, concluindo assim que todos os objetivos propostos foram atingidos, apesar de terem sido atingidos com alguma dificuldade.

5.1 Limitações e trabalho futuro

Conforme o trabalho documentado anteriormente, este apresenta algumas limitações, nomeadamente a nível de apresentação de resultados através do Power BI, uma ferramenta que permite a análise estatística de dados. É considerada uma limitação devido ao facto de está não ser dinâmica no que toca á apresentação de resultados, estando limitado ao que a ferramenta oferece. A limitação mais significativa é o facto de não se poder chegar a uma conclusão definitiva em relação aos resultados apresentados, pois os dados existentes, apesar de serem coerentes e terem a possibilidade de se assemelharem, não são dados reais, mas sim gerados.

Este trabalho é a primeira versão do projeto em causa, pelo que existe ainda muito trabalho a ser desenvolvido. Deste modo deve-se considerar desenvolver uma página web personalizada, e a não utilização o Power Bi, de modo a que deixem de existir barreiras no que toca à apresentação de dados ao utilizador. Outra sugestão passa pelo melhoramento do modelo de *Machine Learning* desenvolvido, aumentando o máximo possível o número de fatores a ser utilizado pelo algoritmo, pois quantos mais fatores forem considerados mais preciso será o cálculo da previsão, aproximando-se assim da realidade. Também deve ser tomado em conta a realização de testes a mais algoritmos, de modo a perceber, juntamente com mais fatores, qual o algoritmo que melhor desempenho apresenta.

Referências

(Bezerra, 2016)	ResearchGate, Bezerra, S. (2016) Reservoir Computing com Hierarquia para Previsão de Vazões Médias Diárias, Figura 2.2: Representação de um neurônio biológico [Online]. Disponível em: https://www.researchgate.net/figure/307578398_fig1_Figura-22-Representacao-de-um-neuronio-biologico-Fonte-extraido-e-adaptado-de (acedido a 17 de outubro de 2017)
(Brownlee, 2016)	Machine Learning Mastery, Brownlee, J. (2016) Supervised and Unsupervised Machine Learning Algorithms [Online]. Disponível em: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (acedido a 2 de março de 2017)
(Carvalho, sem data)	Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, Carvalho, A. (sem data) Redes Neurais Artificiais [Online]. Disponível em: http://conteudo.icmc.usp.br/pessoas/andre/research/neural/ (acedido a 15 de fevereiro de 2017)
(Champandard, 2001-2002)	Alex J. Champandard, A.J. (2001-2002) Reinforcement Learning[Online]. Disponível em: http://reinforcementlearning.ai-depot.com/ (acedido a 3 de março de 2017)
(DesenvolvimentoÁgil, 2013/2014)	DesenvolvimentoÁgil (2013/2014) SCRUM [Online]. Disponível em: http:// www.desenvolvimentoagil.com.br/scrum/ (acedido a 8 de junho de 2017)
(Douvis, 2014)	University of Peloponnese, Department of Sport Management, Douvis, J. (2014) Advances in Sport Management Research, What makes fans attend professional sporting events? A review [Online]. Disponível em: http://sparti.uop.gr/~toda/asmrj/Vol1_c.pdf (acedido a 11 de maio de 2017)
(Ericson et al., 2017a)	Microsoft, Microsoft Azure, Ericson et al., G. (2017) O que é o Azure Machine Learning Studio?[Online]. Disponível em: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-ml-studio (acedido a 15 de junho de 2017)
(Ericson et al., 2017b)	Microsoft, Microsoft Azure, Ericson et al., G. (2017) How to choose algorithms for Microsoft Azure Machine Learning [Online]. Disponível em: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice (acedido a 1 de março de 2017)
(Ericson et al., 2017c)	Microsoft, Microsoft Azure, Ericson et al., G. (2017) Machine learning algorithm cheat sheed for Microsoft Azure Machine Learning Studio [Online]. Disponível em: https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-cheat-sheet (acedido a 20 de março de 2017)
(Ferman, 2015)	Microsoft, Microsoft Azure, Ferman, J. (2015) Hybrid Cloud without the Hassle, Simply Connect to Azure with Availability on Demand [Online]. Disponível em: https://azure.microsoft.com/en-us/blog/hybrid-cloud-without-the-hassle-simply-connect-to-azure-with-availability-on-demand/ (acedido a 21 de março de 2017)
(Freitas, sem data)	Universidade Técnica de Lisboa, Instituto Superior Técnico. Freitas, A. (sem data) Árvores de Decisão Disponível em: http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/indexf23d.html?id (acedido a 14 de fevereiro de 2017)

(Gonçalves, sem data) Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial, Gonçalves, A. (sem data) Máquina de Vectores Suporte [Online]. Disponível em: https://www-users.cs.umn.edu/~agoncalv/arquivos/pdfs/svm.pdf (acedido a 1 de agosto de 2017)

(Gronlund et al., 2017) Microsoft, Microsoft Azure, Gronlund et al, C.J. (2017) Introdução ao Machine Learning na cloud Azure [Online]. Disponível em: https://docs.microsoft.com/pt-pt/azure/machine-learning/machine-learning-what-is-machine-learning (acedido a 15 de junho de 2017)

(Hansen et al., 1989) Journal of Sport Management, University of Ottawa, Hansen, J. e Gauthier, R. (1989) Factors Affecting Attendance at Professional Sport Events [Online]. Disponível em: http://journals.humankinetics.com/doi/pdf/10.1123/jsm.3.1.15 (acedido a 14 de maio de 2017)

(Hubbard et al., 2017) Microsoft, Hubbard et al., H. (2017) sqlcmd Utility [Online]. Disponível em: https://docs.microsoft.com/en-us/sql/tools/sqlcmd-utility (acedido a 11 de maio de 2017)

(ISEP, 2016/2017) EINOV, ISEP (2016/2017) Pensar o Negócio, Modelo Canvas (acedido a 20 de fevereiro de 2017)

(Izumi e Lopes, sem data) Alexandre Izumi, A. e Lopes, D. (sem data) Cloud Computing, Vantagens e Desvantagens [Online]. Disponível em: https://sites.google.com/site/ec1096da428411/home (acedido a 15 de outubro de 2017)

(Jacobson et al., 2011) Jacobson et al., I. (2011) Use-Case 2.0 [Online]. Disponível em: https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/use-case_2_0_jan11.pdf (acedido a 20 de agosto de 2017)

(markga, 2017) Microsoft, Azure in Education, markga (2017) How can I get started with Azure Machine Learning? [Online]. Disponível em: https://blogs.msdn.microsoft.com/azureedu/2017/03/18/how-can-i-get-started-with-azure-machine-learning/ (acedido a 1 de março de 2017)

(Marr, 2016) Forbes, Marr, B. (2016) What Is The Difference Between Artificial Intelligence And Machine Learning? [Online]. Disponível em: https://www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/#42125b8c2742 (acedido a 16 de junho de 2017)

(Matteucci, sem data) Politecnico Milano 1863, Dipartamento Di Elettronica Informazione e Bioingegneria, Matteucci, M. (sem data) A Tutorial on Clustering Algorithms [Online]. Disponível em: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/index.html (acedido a 2 de agosto de 2017)

(Microsoft, 2017a) Microsoft, Microsoft Azure (2017) O que é o Azure? [Online]. Disponível em: https://azure.microsoft.com/pt-pt/overview/what-is-azure/ (acedido a 20 de março de 2017)

(Microsoft, 2017b) Microsoft, Microsoft Azure (2017) O que é a informática na cloud? [Online]. Disponível em: https://azure.microsoft.com/pt-pt/overview/what-is-cloud-computing/ (acedido a 20 de março de 2017)

(Microsoft, 2017c) Microsoft, Microsoft Azure (2017) Documentação da Base de Dados SQL do Azure [Online]. Disponível em: https://docs.microsoft.com/pt-pt/azure/sql-database/ (acedido a 26 de março de 2017)

(Microsoft, 2017d) Microsoft (2017) Excepção de "System.OutOfMemoryException" quando executa uma consulta no SQL Server Management Studio [Online]. Disponível em: https://support.microsoft.com/pt-pt/help/2874903 (acedido a 10 de março de 2017)

(Microsoft, 2017e) Microsoft, Microsoft Azure (2017) Two-Class Logistic Regression [Online]. Disponível em: https://msdn.microsoft.com/library/azure/b0fd7660-eeed-43c5-9487-20d9cc79ed5d (acedido a 10 de setembro de 2017)

(Microsoft, 2017f) Microsoft, Microsoft Azure (2017) Two-Class Logistic Regression [Online]. Disponível em: https://msdn.microsoft.com/library/azure/12d8479b-74b4-4e67-b8de-d32867380e20 (acedido a 10 de setembro de 2017)

(Microsoft, 2017g) Microsoft, Microsoft Azure (2017) Two-Class Logistic Regression [Online]. Disponível em: https://msdn.microsoft.com/library/azure/5ed44caa-5360-407d-ae6c-7a88c491474a (acedido a 10 de setembro de 2017)

(Microsoft, 2017h) Microsoft, Power BI (2017) Documentação do Power BI, Serviço do Power BI [Online]. Disponível em: https://powerbi.microsoft.com/pt-br/documentation/powerbi-azure-and-power-bi/ (acedido a 20 de agosto de 2017)

(Microsoft, 2017i) Microsoft, Microsoft Azure (2017) O que é SaaS? [Online]. Disponível em: https://azure.microsoft.com/pt-pt/overview/what-is-saas/ (acedido a 20 de março de 2017)

(Moura, 2014) Moura, L.A.R (2014) Oficina com Alexander Osterwalder, Canvas Model [Online]. Disponível em: https://s3.amazonaws.com/academia.edu.documents/35803187/Memoria_Seminario_sobre_Canvas_Model_com_Alexander_Osterwlader_-_by_Luiz_Rolim.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1508456708&Signature=yGt4LpsOFSx5ZSnqUBv7QTAC0cA%3D&response-content-disposition=inline%3B%20filename%3DMemoria_Seminario_sobre_Canvas_Model_com.pdf (acedido a 20 de fevereiro de 2017)

(Ning et al., 2015) Microsoft, Ning et al., H. (2015) Diagrama de descrição geral das funcionalidades do Machine Learning Studio [Online]. Disponível em: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-studio-overview-diagram (acedido a 20 de março de 2017)

(Ray, 2017) Analytics Vidhya, Ray, S. (2017) Understanding Support Vector Machine algorithm from examples (along with code) [Online]. Disponível em: https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/ (acedido a 15 de fevereiro de 2017)

(Schapire, 2008) Schapire, R. (2008) Theoretical Machine Learning [Online]. Disponível em: https://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf (acedido a 18 de outubro de 2017)

(Seaters, sem data) Seaters (sem data) [Online]. Disponível em: https://www.seaters.com/ (acedido a 20 de fevereiro de 2017)

(Sommerville, 2013) Sommerville, I. (2013). Engenharia de Software. 9. Disponível em: http://www.ifc-camboriu.edu.br/~catia/IA16/Engenharia_Software_3Edicao.pdf (acedido a 20 de agosto de 2017).

(Stein et al., 2017) Microsoft, SQL Server Management Studio (SSMS), Stein et al., S. (2017) What is SSMS? [Online]. Disponível em: https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms (acedido a 26 de março de 2017)

Anexo A

Este anexo apresenta uma lista de parâmetros que a ferramenta `sqlcmd` permite utilizar com a respetiva descrição. (Hubbard et al., 2017)

```
sqlcmd
   -a packet_size
   -A (dedicated administrator connection)
   -b (terminate batch job if there is an error)
   -c batch_terminator
   -C (trust the server certificate)
   -d db_name
   -e (echo input)
   -E (use trusted connection)
   -f codepage | i:codepage[,o:codepage] | o:codepage[,i:codepage]
   -g (enable column encryption)
   -G (use Azure Active Directory for authentication)
   -h rows_per_header
   -H workstation_name
   -i input_file
   -I (enable quoted identifiers)
   -j (Print raw error messages)
   -k[1 | 2] (remove or replace control characters)
   -K application_intent
   -l login_timeout
   -L[c] (list servers, optional clean output)
   -m error_level
   -M multisubnet_failover
   -N (encrypt connection)
   -o output_file
   -p[1] (print statistics, optional colon format)
   -P password
   -q "cmdline query"
   -Q "cmdline query" (and exit)
   -r[0 | 1] (msgs to stderr)
   -R (use client regional settings)
   -s col_separator
   -S [protocol:]server[instance_name][,port]
   -t query_timeout
   -u (unicode output file)
   -U login_id
   -v var = "value"
   -V error_severity_level
   -w column_width
   -W (remove trailing spaces)
   -x (disable variable substitution)
   -X[1] (disable commands, startup script, environment variables, optional exit)
   -y variable_length_type_display_width
   -Y fixed_length_type_display_width
   -z new_password
   -Z new_password (and exit)
   -? (usage)
```

Figura 40 – Lista de parâmetros do `sqlcmd`