isec
Engenharia

MESTRADO EM INFORMÁTICA E SISTEMAS

DEFINITIVO

Sistema de apoio à decisão para apostas desportivas: Previsão dos resultados de jogos de futebol

Autor

Anabela Dias Carreira

Orientador

Teresa Raquel Corga Teixeira Rocha

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, março de 2021

isec
Engenharia

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

Sistema de apoio à decisão para apostas desportivas: Previsão dos resultados de jogos de futebol

Relatório de Estágio para a obtenção do grau de Mestre em Informática e Sistemas

Especialização em Tecnologias da Informação e do Conhecimento.

Autor

Anabela Dias Carreira

Orientador

Teresa Raquel Corga Teixeira Rocha

Supervisor na empresa Mythical Technologies

Miguel Monteiro

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, março de 2021

“We can't solve problems by using the same kind of thinking we used when we created them.”

Albert Einstein

Dedico este trabalho aos meus pais por todo o apoio dado ao longo da minha vida. Sem eles nada teria sido possível.

AGRADECIMENTOS

Agradeço profundamente a todas as pessoas ou entidades que, de alguma forma, contribuíram e me incentivaram para a conclusão deste trabalho e do meu percurso académico.

Em primeiro lugar, à minha orientadora Professora Doutora Teresa Rocha, do departamento de Engenharia Informática e Sistemas, por toda a disponibilidade, atenção, acompanhamento e orientação prestados no decorrer de todo estágio e por todo o apoio e compreensão demonstrados.

Ao meu supervisor da *Mytical Tecnologies*, Mestre Miguel Monteiro, pela orientação ao longo deste último ano na empresa, mostrando-se sempre disponível e compreensivo. Todas as sugestões e correções prestadas foram indispensáveis para a conclusão deste trabalho.

À *Mythical Tecnologies* e a toda a equipa, por me terem tão bem acolhido e garantirem que o trabalho corria sempre de forma agradável. Sempre se disponibilizaram para me transmitir os melhores conhecimentos e acompanhar o projeto, assegurando o melhor resultado possível.

Ao Instituto Superior de Engenharia de Coimbra e a todos os professores do departamento de Engenharia Informática e Sistemas, por todos os conhecimentos e aprendizagens que me foram transmitidas.

À minha família por todo a motivação e apoio prestados ao longo da minha vida académica e, em especial, aos meus pais, que sem eles nada teria sido possível. Ao meu irmão Dennis Carreira por toda a ajuda e conselhos que me dispensou.

Aos meus amigos e colegas de Engenharia, em especial a minha amiga Joana Marques, por toda a amizade, apoio, entreajuda e companheirismo ao longo desta caminhada.

Por último, um agradecimento especial ao meu namorado, por toda a motivação e incentivo que me deu nos momentos em que mais precisei.

RESUMO

Nos últimos anos tem-se assistido a um enorme crescimento do mercado das apostas desportivas *online*, um pouco por todo o mundo, incluindo Portugal. Segundo dados do Serviço de Regulação e Inspeção de Jogos (SRIJ), o futebol é a modalidade desportiva com maior volume de apostas em território nacional.

Sendo cada vez maior a concorrência de casas de apostas *online*, é natural que estas procurem disponibilizar cada vez mais e melhor informação aos seus apostadores, no sentido de lhes facilitar a tomada de decisão.

Seguindo essa tendência, a empresa *Mythical Technologies*, proprietária do *Traderline* que é uma ferramenta de apostas desportivas *online*, desenvolvida em Portugal, decidiu propor o desenvolvimento do presente trabalho, cuja finalidade era testar diferentes abordagens para prever os resultados de eventos que ocorrem durante os jogos de futebol, através de diversos algoritmos de *Machine Learning*, seguindo a metodologia *Cross Industry Standard Process for Data Mining* (*CRISP-DM*).

Os eventos selecionados para o estudo foram, o *resultado final do jogo*, o *número de golos* e o *número de cantos*, tendo a escolha dos algoritmos recaído sobre *Decision Tree*, *Random Forest*, *K-Nearest Neighbors*, *Neural Network*, *Support Vector Machine*, *Logistic Regression* e *XGBoost*. Para o treino e validação dos modelos foram usados dados históricos das primeiras ligas portuguesa e inglesa, disponibilizados pela fonte *Footbal-Data.co.uk*.

Como resultado do estudo, o melhor desempenho na previsão dos referidos eventos foi obtido pelos seguintes algoritmos: *XGBoost* no resultado dos jogos, com 60,09% de *accuracy*; *XGBoost* no número de golos, com 55,37% de *accuracy*; Regressão Logística no número de cantos, com uma *accuracy* de 53,92%.

Concluindo, não obstante os resultados poderem vir a ser alvo de melhoria, este estudo constituiu um primeiro passo no desenvolvimento de um sistema de apoio à decisão para os utilizadores do *Traderline*.

Palavras-Chave: Apostas desportivas; Futebol; Sistema de apoio à decisão; Previsão; *Machine Learning*.

ABSTRACT

In recent years there has been an enormous growth in the online sports betting market, all over the world, including Portugal. According to data from the *Serviço de Regulação e Inspeção de Jogos* (*Game Regulation and Inspection Service*), football is the sport with the highest volume of bets in the national territory.

As competition from online bookmakers is increasing, it is natural that they seek to provide more and better information to their bettors, in order to facilitate their decision making.

Following this trend, the company *Mythical Technologies*, owner of *Traderline*, which is an online sports betting tool, developed in Portugal, decided to propose the development of this work, whose purpose was to test different approaches to predict the results of events that occur during football games, through several Machine Learning algorithms, following the *Cross Industry Standard Process for Data Mining* (*CRISP-DM*) methodology.

The events selected for the study were the *final result of the game*, the *number of goals* and the *number of corners*, with the choice of algorithms falling on *Decision Tree*, *Random Forest*, *K-Nearest Neighbors*, *Neural Network*, *Support Vector Machine*, *Logistic Regression* and *XGBoost*. For the training and validation of the models, historical data of the first portuguese and english leagues, retrieved from the source Footbal-Data.co.uk, were used.

As a result of the study, the best performance in the prediction of the mentioned events was obtained by: XGBoost in the result of the games, with 60.09% accuracy; XGBoost in the number of goals, with 55.37% accuracy; Logistic regression in the number of corners, with an accuracy of 53.92%.

In conclusion, although the results may be subject to improvement, this study was a first step in the development of a decision support system for Traderline users.

Keywords: Sports betting; Football; Decision Support System; Prediction; Machine Learning.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE QUADROS

SIGLAS E ACRÓNIMOS

ANN – Artificial Neural Network

API – Application Programming Interface

BN – Bayesian Networks

CRISP-DM – Cross Industry Standard Process for Data Mining

DM – Data Mining

F1 – F1-Score

FN – Falsos Negativos

FP – Falsos Positivos

ISEC – Instituto Superior de Engenharia de Coimbra

KDD – Knowledge Discovery in Database

KNN – K-Nearest Neighbors

LR – Logistic Regression

ML – Machine Learning

PPI – Player Performance Index

RF – Random Forest

RNNs – Recurrent Neural Networks

SAD – Sistema de Apoio à Decisão

SVM – Support Vector Machine

TE – Taxa de Erro

VN – Verdadeiros Negativos

VP – Verdadeiros Positivos

CAPÍTULO 1

INTRODUÇÃO

O presente relatório foi elaborado no âmbito do estágio curricular, com vista à conclusão do mestrado em Informática e Sistemas do Instituto Superior de Engenharia de Coimbra.

O estágio desenvolveu-se na empresa *Mythical Technologies*, durante os dois semestres do ano letivo 2019/2020, o primeiro a tempo parcial e o segundo a tempo integral. De referir que, devido à pandemia COVID-19, a partir de finais de março, o trabalho passou a regime remoto.

Este documento pretende, pois, descrever as atividades realizadas no contexto do estágio. O presente capítulo começa pela contextualização do problema que justifica a pertinência do estágio, em seguida faz uma breve apresentação das entidades envolvidas, prossegue com a definição dos objetivos do trabalho e termina com a descrição da forma como o documento se encontra organizado.

1.1. Contextualização do problema

O mercado das apostas desportivas é um setor em grande crescimento a nível mundial. No continente Europeu calcula-se que sejam gerados por este mercado, anualmente, mais de 11 biliões de euros (Matos, 2013). Em Portugal, segundo o relatório apresentado pelo Serviço de Regulação e Inspeção de Jogos (SRIJ), no primeiro trimestre de 2020, houve 157,4 milhares de novos jogadores registados e estima-se que haja cerca de 425,8 mil apostadores com prática de jogo, ou seja, aproximadamente 5% da população adulta no país. Durante o primeiro trimestre de 2020, foi produzido um valor de 34,5 milhões de euros (Figura 1), gerando um aumento de 39,2% da receita bruta face aos 24,8 milhões de euros produzidos no período homólogo de 2019 (SRIJ, 2020).

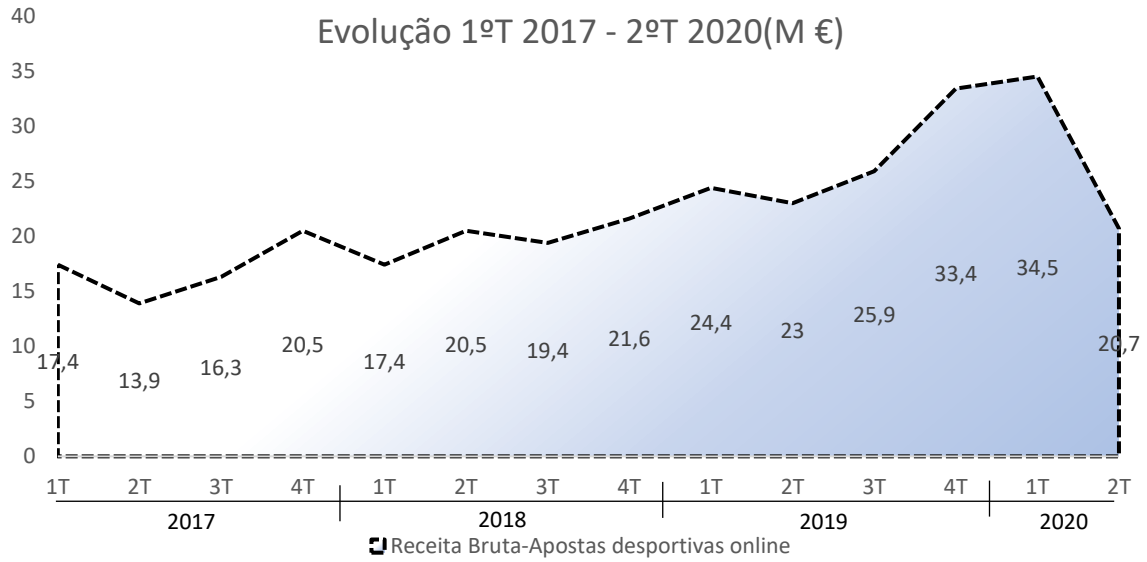


Figura 1 - Evolução da receita bruta das apostas desportivas online durante o período de 2017-2020

Dentro das apostas desportivas, o futebol é de facto o desporto mais praticado mundialmente e no qual os portugueses mais apostam o seu dinheiro, verificando-se que uma taxa de 74,7% das apostas desportivas totais são realizadas neste desporto (Coimbra, 2018).

Considerando o grande número de indivíduos envolvidos neste mercado e as quantias monetárias que este negócio movimenta, é do interesse destes apostadores ter uma previsão do resultado final de um confronto, para que, dessa forma, possam aumentar as probabilidades de as suas apostas serem bem sucedidas (Zaan, 2017). Neste contexto, as casas de apostas *online* procuram acompanhar e apoiar da melhor forma possível os apostadores nas suas sessões de jogo, desenvolvendo ferramentas que permitam, entre outros, estabelecer prognósticos confiáveis. Existem várias técnicas para tentar realizar a previsão dos resultados, mas não existe uma técnica que consiga prevê-los com exatidão, dada a grande quantidade de variáveis que podem influenciar esses resultados (Herbinet, Predicting Football Results Using Machine Learning Techniques, 2018).

É, pois, neste enquadramento, que surge a proposta de estágio da empresa *Mythical Tecnologies*. Sendo proprietária do *Traderline*, que é uma ferramenta de apostas desportivas *online*, decidiu propor o desenvolvimento do presente trabalho, cuja finalidade é testar diferentes abordagens para prever os resultados de eventos que ocorrem durante os jogos de futebol, recorrendo a diversos algoritmos de *Machine Learning*. O objetivo é que os melhores algoritmos venham a integrar um sistema de apoio à decisão para auxiliar os apostadores na plataforma *Traderline,*

1.2. Entidade de acolhimento

Como já referido, este trabalho foi desenvolvido na *Mythical Techologies* (Technologies, 2019), uma empresa de desenvolvimento de *software*, criada a 25 de maio de 2015, pelos fundadores Rui Sales e Ricardo Margalho. Esta *startup*, encontra-se sediada no bloco C, do Instituto Pedro Nunes, em Coimbra.

A empresa surgiu da necessidade de conceber uma aplicação na área das apostas desportivas *online* que permitisse aos utilizadores realizar vários tipos de apostas de uma maneira rápida e eficaz, capaz de responder tanto às necessidades dos apostadores mais experientes, como às dos que pretendiam estrear-se nesse mundo. Nasceu assim o seu produto principal, o *Traderline*, uma aplicação que permite efetuar apostas *online* nos principais desportos, de forma fácil, rápida e intuitiva, estando disponível nas plataformas *Windows*, *Andriod* e *iOS*. Este produto está certificado pela *Betfair*[1], o que lhe permite utilizar uma grande bolsa de apostas desportivas.

Paralelamente, a *Mythical Techologies* presta serviços aos clientes da *Betfair* e desenvolve soluções *web*, bem como *software* para *IOS, MacOS e Android*.

De referir que no período em que decorreu o estágio a empresa se encontrava a desenvolver uma nova versão do *Traderline*, como resposta ao crescimento exponencial que ocorreu neste mercado e para adaptar-se às novas necessidades dos utilizadores. Por outro lado, é de salientar que tendo começado a operar em países como Portugal e Brasil, presentemente já vende este produto para diversos países como, Emirados Árabes Unidos, Inglaterra, Espanha e Itália.

Figura 2 - Logotipo da empresa.
Fonte: (Technologies, 2019)

1.3. Instituto Superior de Engenharia de Coimbra

O Instituto Superior de Engenharia de Coimbra (ISEC, 2021), é uma instituição com quase 50 anos de história, acolhendo anualmente mais de 3300 alunos nas suas instalações, das mais diversas áreas da engenharia.

Este instituto inclui, na sua oferta formativa, doze licenciaturas, dez mestrados, bem como diversos cursos técnicos superiores profissionais, encontrando-se dividido nos departamentos de Engenharia Civil, Engenharia Eletrotécnica, Engenharia Mecânica,

[1] A maior casa de apostas online (Betfair, 2021)

Engenharia Química e Biológica, Física e Matemática, e de Engenharia Informática e de Sistemas (DEIS).

O DEIS, além de três licenciaturas em Engenharia Informática (regime diurno, regime pós-laboral e curso europeu), oferece o mestrado em Informática e Sistemas, no âmbito do qual se desenvolve o presente trabalho de estágio.

1.4. Objetivos e plano de trabalho

Tal como anteriormente referido, o mercado das apostas desportivas *online* é um setor em grande crescimento a nível mundial e Portugal acompanha essa tendência. A Mythical Technologies, proprietária do Traderline, tal como as suas concorrentes, pretende apoiar da melhor forma os seus apostadores, nomeadamente, proporcionando-lhes prognósticos que os ajudem na tomada de decisão.

Embora esta empresa pretenda integrar modelos preditivos para vários tipos de desportos na plataforma *Traderline,* sendo o futebol, o desporto que comprovadamente é alvo do maior número de apostas, nomeadamente em Portugal, foi selecionado como primeiro e único caso de estudo a considerar no estágio.

De facto, perante a complexidade do problema a resolver, dada a quantidade de variáveis que podem influenciar o resultado de um jogo e que vão desde as condições climatéricas, até à motivação e condições psicológicas da equipa, acordou-se que os outros desportos seriam alvo de trabalho futuro.

Desta forma, o objetivo do presente trabalho é desenvolver um sistema de apoio à decisão para auxiliar os apostadores de futebol a obter o maior lucro possível das suas apostas, mediante a previsão de eventos específicos que têm lugar durante os confrontos. Mais especificamente, deverão ser testados vários modelos de previsão com recurso a diferentes algoritmos de *Machine Learning*, seguindo a metodologia *Cross Industry Standard Process for Data Mining* (*CRISP-DM*).

O intuito é que os modelos com os melhores resultados, venham a integrar a plataforma *Traderline*.

Com base nestas considerações, programou-se o desenvolvimento do trabalho em diversas fases. Primeiramente, há que compreender bem o problema a resolver e investigar as possíveis fontes de dados a utilizar no estudo. Depois de escolher as fontes, há que selecionar e recolher os dados relevantes, avaliar a sua qualidade, bem como realizar a preparação dos mesmos. Depois, surge a criação dos modelos de previsão, seguindo-se o teste e avaliação dos mesmos. Posteriormente, há que selecionar os melhores algoritmos para integrar o sistema de apoio à decisão. A etapa final será a integração do sistema no Traderline, mas condicionada à duração das fases anteriores.

1.5. Estrutura do documento

Este relatório encontra-se dividido em sete capítulos, cujo conteúdo se resume em seguida.

No primeiro capítulo, é feita a contextualização do problema a resolver no âmbito do estágio, são apresentadas as instituições envolvidas e são especificados os objetivos a atingir.

No capítulo dois, é feita uma apresentação dos principais conceitos teóricos, que se consideram necessários à compreensão do presente trabalho. Nomeadamente, introduzem-se os conceitos de sistema de apoio à decisão, *Data Mining,* metodologia *CRISP-DM*, alguns algoritmos de *Machine Learning*, bem como metodologias e métricas para avaliação de desempenho dos modelos.

No capítulo três, é efetuada a contextualização do tema do trabalho, nomeadamente futebol e apostas desportivas *on-line*, é apresentado o *Traderline* e as suas principais concorrentes, e são revistos alguns trabalhos científicos realizados na área.

No capítulo quatro, procede-se à apresentação do contexto tecnológico que esteve na base do trabalho desenvolvido.

O capítulo cinco, descreve as fases que envolveram a recolha dos dados, a identificação das variáveis, a análise uni e multivariada dos dados, a limpeza e transformação dos mesmos, bem como a seleção das variáveis a incluir no *dataset* provisório. Apresenta ainda a construção dos vários *datasets* a serem utilizados na previsão dos diferentes eventos de futebol.

O capítulo seis, apresenta o processo de criação dos modelos para a previsão dos eventos de futebol, bem como a avaliação do seu desempenho e análise/discussão dos resultados.

No capítulo sete, são apresentadas as conclusões do trabalho realizado e algumas sugestões para trabalho futuro.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Ao longo deste capítulo, serão apresentados alguns dos conceitos que se consideram importantes para compreensão do trabalho realizado. Desta forma, primeiramente procede-se a uma breve explicação do que é um sistema de apoio à decisão. Em seguida, apresenta-se alguns conceitos de *Data Mining,* nomeadamente a metodologia *CRISP-DM* adotada para a realização do estudo, bem como os algoritmos de *Machine Learning* utilizados no mesmo. Por fim, introduzem-se as metodologias e métricas para avaliação de desempenho dos modelos criados.

2.1. Sistema de apoio à decisão

Não existe, até ao momento uma definição aceite para o conceito de sistemas de apoio à decisão (SAD). No entanto, de acordo com vários autores, um SAD pode ser considerado como um sistema informático que fornece aos seus utilizadores informações baseando-se em atributos e objetivos, para ajudar/apoiar a tomada de decisão relativamente ao problema inicial (Pinho de Sá, 2016) (Leitão Gomes, 2015).

Antes de tomar uma decisão pode adotar-se duas abordagens, sendo que a primeira segue a intuição e o conhecimento do decisor, ou seja, *ad-hoc*, enquanto a segunda segue um processo em que está definido o que deve ser executado em cada fase.

De acordo com Bohanec (Bohanec, 2009), o processo para tomar uma decisão deve passar pelas seguintes fases:

- Identificação do problema;
- Identificação de alternativas relevantes;
- Decomposição e modelação do problema;
- Avaliação e análise das alternativas;
- Decisão de qual a melhor alternativa;
- Implementação da alternativa selecionada.

Depois de um SAD ser implementado, este deve ser capaz de responder a problemas semiestruturados e não estruturados. Os dados recebidos dos primeiros problemas tanto podem ser bem estruturados como não serem estruturados, sendo que os últimos, como o nome indica, são dados que quando obtidos não são organizados. O sistema, no final, deve apoiar o utilizador nas diversas fases de decisão, aumentar a eficácia da tomada de decisão e ainda permitir prever/avaliar as diferentes opções.

A arquitetura de um sistema, como representado na Figura 3, é constituída por três principais componentes: um sistema de gestão de bases de dados (SGBD), um sistema de gestão de modelos (SGM) e uma interface. O sistema de gestão de base de dados, permite o acesso às bases de dados que contêm os dados relevantes para o problema em questão, permitindo assim distinguir os relacionamentos e processamentos dos dados da aplicação em si. O sistema de gestão de modelos utiliza os dados do SGDB e aplica-lhe os modelos de modo a extrair soluções para o problema inicial. Assim como o SGBD, o SGM permite separar os modelos que serão utilizados do resto da aplicação. A interface, como o nome evidencia, é o componente que é apresentado ao utilizador fornecendo-lhe as informações que foram extraídas da aplicação do modelo ajudando a decidir sobre a melhor solução do problema.

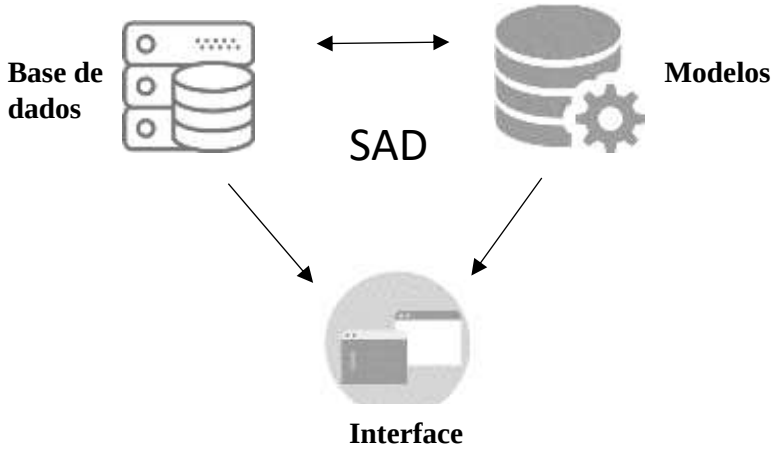


Figura 3 - Arquitetura de um SAD.
Fonte: Adaptado do (dsssystem, 2010)

2.2. Data Mining

Atualmente, existe uma grande preocupação em armazenar / analisar a informação que é recolhida diariamente pelas empresas. Com a evolução para bases de dados que permitem armazenar grandes volumes de informação e com o reconhecimento de que esta assume um valor cada vez maior no apoio à tomada de decisão, tornou-se necessário redefinir as tradicionais técnicas de análises de dados. De facto, este progresso levantou o problema de ser praticamente impossível analisar os dados com recurso a métodos estatísticos, como no passado, o que deu origem ao aparecimento do conceito de *Data Mining* (DM).

Não havendo uma definição única aceite para este conceito, de entre as várias que se encontraram, as que se acharam mais relevantes foram:

- *Data Mining* é a procura de informações valiosas em grandes bases de dados, é um esforço de cooperação entre homens e computadores. Os homens

projetam as bases de dados, descrevem o problema e definem os seus objetivos. Os computadores verificam os dados e procuram padrões semelhantes às metas estabelecidas pelos homens. (Weiss, Indurkhya, & Zhang, 2015)

- *Data Mining* é um processo lógico que é utilizado para procurar qual é a informação útil numa grande quantidade de dados (Ramager, 2010).
- *Data Mining* é um processo de negócio para explorar grandes quantidades de dados de forma a descobrir padrões e regras importantes (Linoff & A. Berry, 2011).
- *Data Mining*, é a extração não trivial de informação implícita, previamente desconhecida e potencialmente útil, a partir de bases de dados (Padhy, Mishra, & Panigrahi, 2012).
- *Data Mining* pode ser entendido como a aplicação de algoritmos de *Machine Learning* para extração semiautomática ou automática de informação que está armazenada em bases de dados (Konjevoda & Štambuk , 2011).

Realizando um resumo de todas as definições propostas, podemos retirar que o principal objetivo do DM é procurar padrões ou regras nos dados que estão armazenados em grandes bases de dados, de forma a obter informação útil. É necessário ter em conta que este processo pode ser realizado de forma automática ou semiautomática.

Na Figura 4, podem ser observadas as várias disciplinas que o conceito de *Data Mining* relaciona.

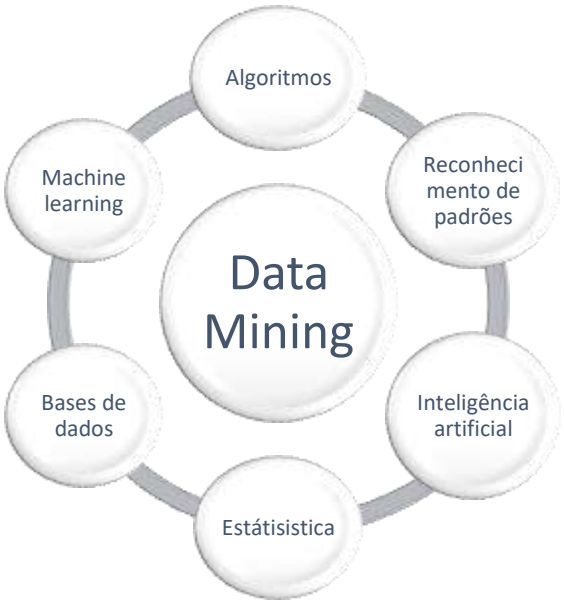


Figura 4 - Áreas envolvidas no Data Mining.
Fonte: Adaptado de (Haldurai. L, 2016)

A aplicação de técnicas de DM trouxe muitas vantagens em diversas áreas (*Marketing*, Banca, Saúde, entre outras), nomeadamente, na previsão de tendências futuras, no auxílio á tomada de decisão, na redução do aparecimento de situações não planeadas, bem como na análise mais rápida e rentável dos dados. Tornou possível a aplicação de vários algoritmos ao mesmo conjunto de dados (*dataset*) de modo a obter-se o melhor desempenho possível.

Não obstante, também trouxe algumas desvantagens para as empresas, pois levou-as a despender custos extraordinários na formação e na compra de ferramentas para o tratamento de dados e a aplicação de algoritmos. Outro problema que surgiu foi a possibilidade de serem retiradas conclusões erradas das análises realizadas, levando a tomada de decisões que na realidade não deveriam ser tomadas (Gomes T. , 2014).

2.3. Metodologia CRISP-DM

A metodologia *CRISP-DM (Cross Industry Standard Process for Data Mining)* foi concebida no ano de 1996, por um grupo de especialistas na área de *Data Mining.*

Esta é apresentada como um modelo hierárquico de processos que descreve o ciclo de vida que um projeto de *Data Mining* deve seguir, contendo as fases e as tarefas que devem ser executadas dentro de cada etapa, bem como o *output* que cada uma deve produzir (CRISP-DM: Towards a Standard Process Model for Data Mining, 2000).

Esta metodologia encontra-se ilustrada na Figura 5.

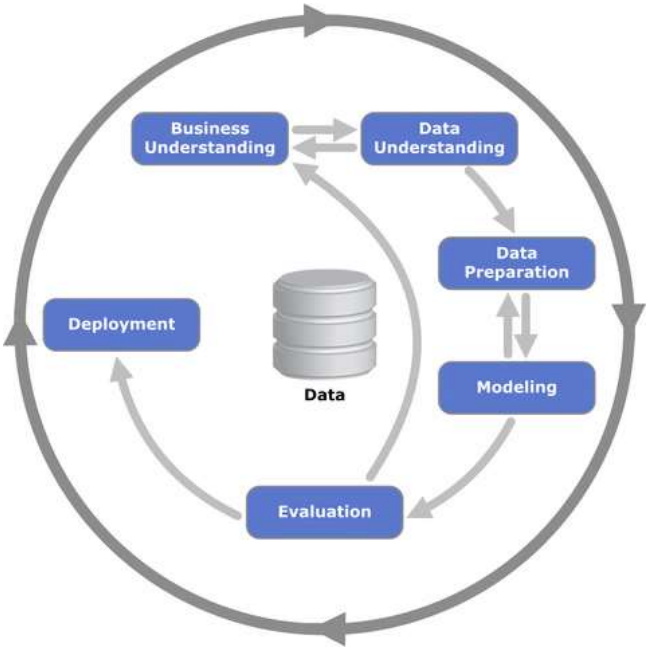


Figura 5 - Fases da metodologia CRISP_DM.
Fonte: (Vorhies, 2016)

Fase1 – Compreender o negócio

Esta primeira fase, tem como objetivo definir a questão inicial como um problema de DM. Para isso, é necessário declarar os objetivos do projeto, compreender o que é esperado, quais os requisitos necessários e ainda especificar o contexto do problema. É também nesta fase que é realizado um plano geral do projeto, onde são definidas as ferramentas e técnicas a adotar, é estudado o orçamento total do projeto, o esforço e os recursos que serão necessários para a sua execução.

Fase 2 – Compreender os dados

A fase de compreensão dos dados deve ser adaptada a cada mercado pois é necessário escolher o tipo de dados mais relevante para o estudo em causa. Esta fase pode ser divida em 4 subprocessos. Primeiramente é indispensável recolher os dados importantes, de seguida esses dados necessitam de ser descritos para que possam ser identificados *subsets* interessantes a explorar posteriormente e, no final, é verificada a qualidade dos dados.

Fase 3 – Preparar os dados

Nesta fase preparam-se os dados que foram encontrados na fase anterior, como os mais relevantes para o caso em estudo, de forma a obter-se o *dataset* final. Segundo o guia para esta metodologia, que foi desenvolvido pela IBM (IBM, 2012), nesta fase devem ser percorridas as seguintes etapas:

- Realizar a fusão do conjunto de dados;
- Selecionar o subconjunto de dados;
- Agregar os registos;
- Desenvolver novos atributos (ou features);
- Ordenar os dados a modelar;
- Eliminar ou substituir valores nulos/inexistentes;
- Separar os dados nos diversos *datasets* (treino, teste e validação);

É ainda de notar que esta fase é a que requer mais tempo de execução, pois é uma etapa fulcral que, caso não seja executada de forma correta, irá influenciar o sucesso da resolução do problema.

Fase 4 – Modelar

A fase de modelar os dados requer que sejam tomadas decisões sobre quais as melhores técnicas e ferramentas a utilizar na análise dos dados preparados anteriormente. Para decidir entre os vários tipos de algoritmos e métodos existentes,

é fundamental que se tenha algum conhecimento, tanto do domínio do problema em questão, como dos algoritmos. Particularmente neste trabalho, usaram-se algoritmos de *Machine Learning*.

De modo a obter-se os melhores resultados possíveis, devem ser testados os vários algoritmos que podem funcionar para o tipo de problema e devem ser ajustados os seus parâmetros, encontrando-se assim o modelo que nos irá fornecer maior qualidade, de entre os existentes.

Fase 5 – Avaliar

Nesta fase, é realizada a avaliação do modelo que foi escolhido anteriormente. Para ter uma boa avaliação, é importante que o modelo consiga descobrir padrões relevantes nos dados. Para avaliar se o modelo está a funcionar de acordo com o esperado, deve haver uma interligação entre o analista dos dados e o responsável pelo negócio, para assim perceber se este está de acordo com os objetivos e com o problema do negócio. É ainda necessário, nesta fase, realizar uma revisão do processo que foi efetuado até a fase corrente e decidir quais serão os próximos passos que devem ser efetuados. Ou seja, se é possível avançar para a fase de *deployment*, ou se é necessário recuar e repetir o processo.

Fase 6 – *Deployment*

Esta é a fase final do processo, em que é pressuposto que as etapas anteriores foram executadas com sucesso. Na fase de *deployment* é realizado o planeamento da sua implementação, a monitorização e a manutenção. Deve ainda ser produzida documentação de como tudo deve ser efetuado, já que na maioria dos casos não é o analista que irá realizar a implementação do modelo.

Na Figura 6, é apresentado um quadro resumo de todas as fases e tarefas que devem ser realizadas ao longo do processo descrito por esta metodologia (Alsultanny, 2011).

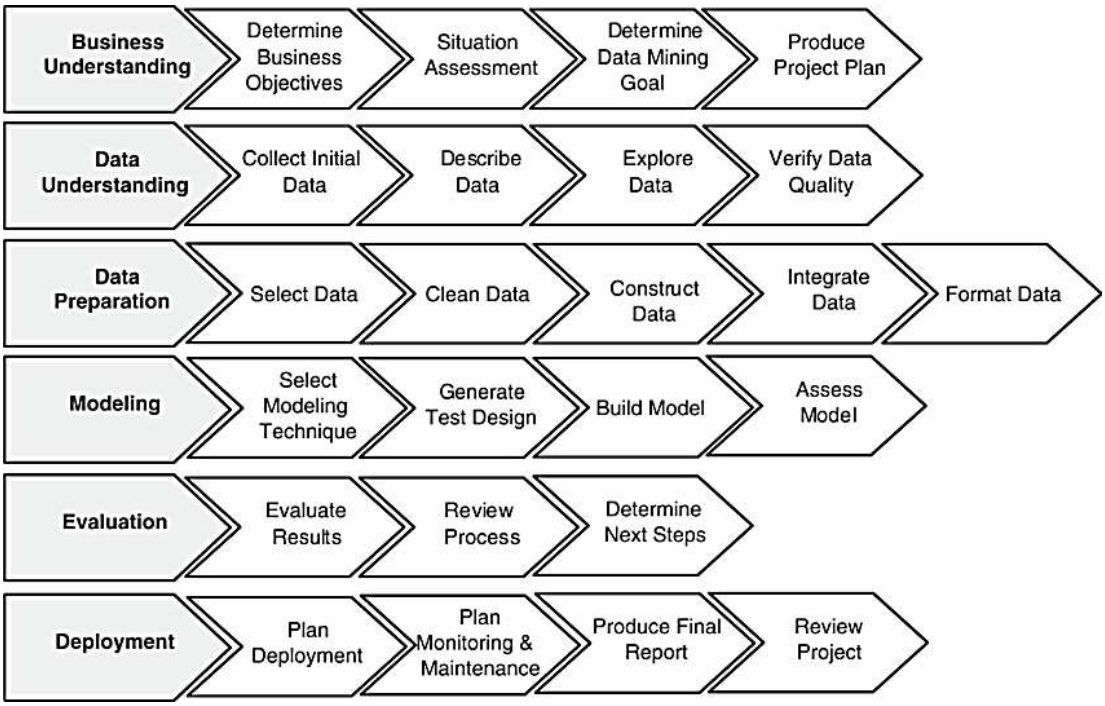


Figura 6 - Fases e tarefas da metodologia CRISP-DM.
Fonte: (Alsultanny, 2011)

2.4. Machine Learning

Inteligência artificial (IA) e *Machine Learning* (ML), são dos assuntos com maior relevância no século XXI. A IA é considerada como a atribuição de "inteligência" a uma máquina / sistema, ou seja, é dada a possibilidade a uma máquina de conseguir "pensar" e tomar decisões de maneira autónoma. A construção de um sistema inteligente engloba diferentes áreas como o processamento de linguagem natural, planeamento automatizado, *Machine Learning*, robótica, entre outros (Simeone, 2018).

A ML utiliza uma variedade de algoritmos que aprendem iterativamente com os dados para tentar melhorar a análise da informação e prever o melhor resultado possível. A aprendizagem pode ser supervisionada ou não supervisionada. Na primeira abordagem, os dados são conhecidos e sabe-se qual deve ser a classificação final dos mesmos. Na segunda abordagem, não existe um *output* associado a cada exemplo, ou seja, pretende-se que seja realizado um estudo de padrões e relações entre as *features* de forma a tentar obter grupos que possam distinguir os vários dados (Bunker & Thabtah, 2017).

Nos problemas de aprendizagem supervisionada, podem ser seguidos métodos diferentes para a sua resolução, dependendo do tipo de variável que se pretende prever. É necessário considerar que é um problema de regressão, sempre que o atributo de saída for do tipo numérico. Se a variável que se pretende prever, for uma classe, uma categoria ou um diagnóstico, nesse caso é um tratado como um problema de classificação. Esta distinção está representada na Figura 7.

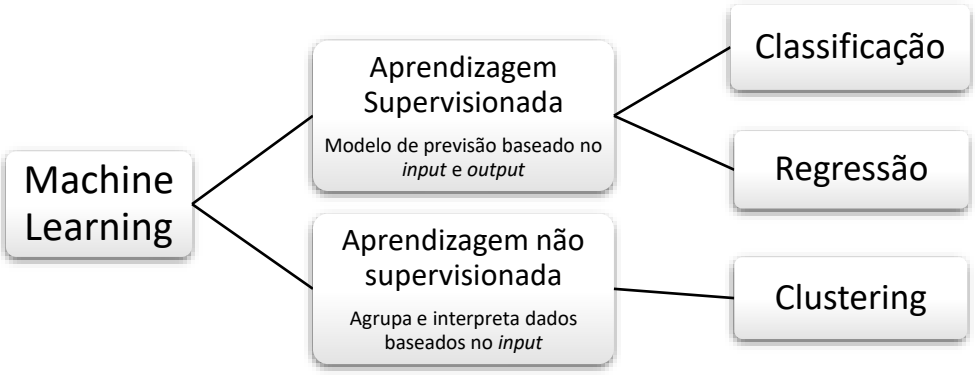


Figura 7 - Aprendizagem supervisionada e aprendizagem não supervisionada.
Fonte: Adaptado de (Bunker & Thabtah, 2017)

2.4.1. Algoritmos implementados

Seguidamente, descrevem-se os algoritmos de ML implementados neste trabalho.

Árvores de decisão

O algoritmo de árvores de decisão funciona de maneira análoga a um fluxograma. Este fluxograma em árvore de decisão é composto por nós e folhas, sendo que em cada nó são testadas regras a cada atributo (Campos R. , 2017). O resultado do teste feito a cada atributo cria uma ramificação, no final composta por folhas que indicam a classe à qual o exemplo pertence. As árvores possuem um nó principal denominado nó raiz, sendo este o que ocupa o grau hierárquico mais elevado e liga-se a outros nós, denominados nós filhos. Estes, por sua vez, podem possuir nós filhos e este ciclo continua até terminar num nó que não possua filhos, denominado de nó folha ou nó terminal. No final, a decisão, é tomada testando todas as condições desde o nó raiz até chegar ao nó folha onde está especificada a classe prevista. Para elucidar, na Figura 8 temos uma árvore de decisão na qual se decide se deve ser concedido crédito a um cliente. Suponhamos que para o conjunto de dados referentes a clientes de um banco pretendemos classificar uma nova instância. Primeiramente irá ser testado o atributo idade e, consoante o valor da instância, serão realizados mais testes de forma a ser atribuída uma classe à amostra testada.

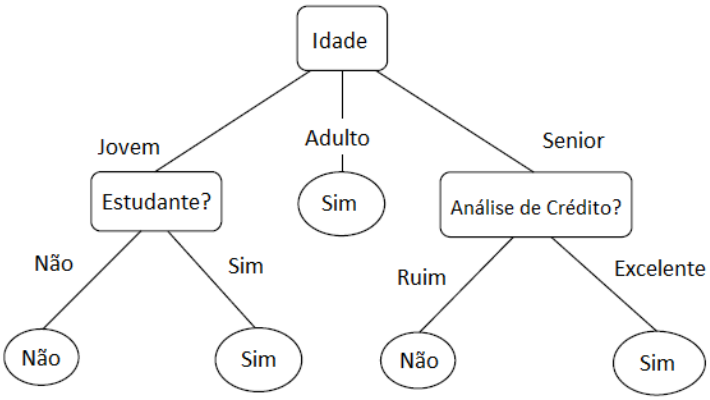


Figura 8 - Representação de uma árvore de decisão de um problema para crédito bancário
Fonte: (Ramos, 2016)

As árvores de decisão são algoritmos de aprendizagem supervisionados e não paramétricos, ou seja, os dados devem conter o valor a prever e nada é presumível antecipadamente, uma vez que é considerado que as amostras não têm dependências nos seus parâmetros. Este algoritmo é muito utilizado na resolução de problemas de classificação e de regressão.

As principais valências deste algoritmo são a fácil capacidade de interpretação e representação, pois as mesmas são facilmente visualizadas. Este algoritmo funciona bem para problemas com grande quantidade de dados, consegue lidar com dados categóricos e numéricos, e não necessita de ser realizado um pré-processamento, uma vez que ele consegue trabalhar com valores não normalizados. É um algoritmo de fácil implementação, pois não requer muitos parâmetros de configuração. As principais desvantagens deste classificador são: a ocorrência de *overfitting* em que as árvores se adaptam demasiado ao conjunto de treino; a sua instabilidade que leva a que, se os dados sofrerem pequenas alterações, as árvores produzidas sejam completamente diferentes e se os dados possuírem muitos atributos estas possam tornar-se complexas (Simões, 2008).

Random Forest

O algoritmo *Random Forest* (RF), como o próprio nome sugere, é uma combinação de outros algoritmos (*ensemble*). Este classificador combina diferentes árvores de decisão independentes, que operam em conjunto para prever a classe. Este utiliza técnicas de reamostragem de forma a reduzir a variância associada aos algoritmos de aprendizagem supervisionada (Fernandes, 2019).

O RF utiliza a técnica de *bagging* em que, a partir do *dataset* de treino, são gerados diferentes subconjuntos dos dados de treino aleatoriamente e com reposição, de forma a que seja criado um modelo (árvore de decisão) diferente para cada um desses subconjuntos de dados. Para realizar a classificação de uma nova instância, utiliza-se um estimador como a média dos valores no caso de problemas de regressão, ou

usa-se um sistema de votos no caso de problemas de classificação. No sistema de voto, cada um dos modelos prevê a classe para cada instância. A classe final, será a que tiver obtido mais votos (Yiu, 2019). Esta lógica encontra-se descrita na Figura 9.

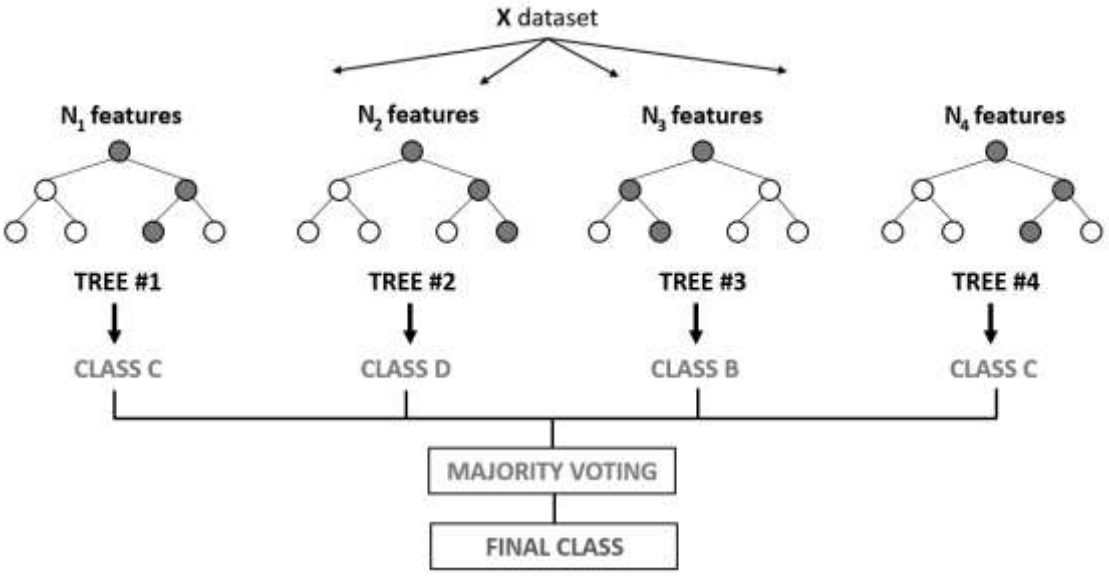


Figura 9 - Representação esquemática do algoritmo Random Forest.
Fonte: (TechTour, 2019)

As principais vantagens associadas a este algoritmo são: a sua fácil capacidade de perceção e implementação; a existência de poucos parâmetros de configuração; a previsão com boas taxas de acerto dos parâmetros *standard*; a grande versatilidade, uma vez que tanto pode ser utilizado em problemas de classificação como de regressão. Este algoritmo funciona bem com *datasets* grandes, com um grande número de atributos e ainda com classes não balanceadas (Silva , 2018). As RF, vêm resolver o problema de *overfitting* associado às árvores de decisão, com a construção de diferentes árvores, evitando que o modelo fique demasiado ajustado aos dados. As principais desvantagens ocorrem quando são geradas grandes quantidades de árvores, levando a que o algoritmo se torne lento e ineficiente, sendo difícil fazer previsões em tempo real. Este algoritmo não funciona muito bem com variáveis categóricas e costuma apresentar taxas de acerto mais baixas do que classificadores lineares para dados que contêm características independentes (Dmitrievsky, 2018).

K-Nearest Neighbors

O algoritmo *K-Nearest Neighbors* (KNN), é dos mais simples de compreender, pois simplesmente analisa a semelhança que existe entre os diferentes dados (vetores) mais próximos e a nova instância a classificar. Baseia-se na ideia de que um ponto no espaço está rodeado, na sua maioria, por pontos que pertencem à mesma classe, uma vez que os pontos são colocados no espaço de acordo com os seus atributos. Para classificar uma nova instância, é necessário calcular a distância desta aos pontos para os quais já se conhece a classe a que pertencem. Na Figura 10, está

exemplificado o processo de classificação com o KNN para um valor de *k* igual a 3. Na figura, a instância assinalada corresponde à observação a ser classificada. Como o valor de *k* é igual a 3, são escolhidos os três exemplos mais próximos. Neste caso, como são dois exemplos de classe "escura" e um exemplo de classe "clara", a classificação seria "escura", uma vez que na vizinhança a classe predominante é a "escura" (Italo, 2018).

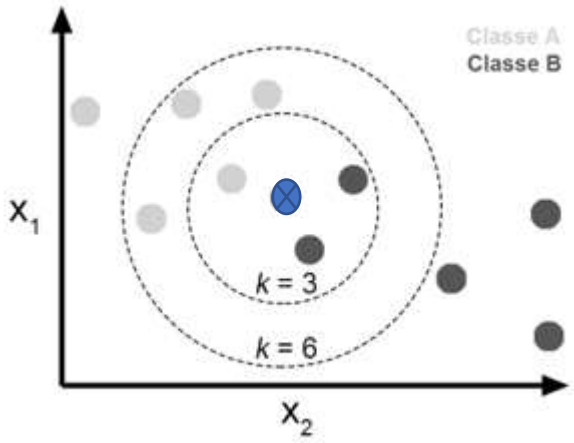


Figura 10 - Exemplo do classificador KNN, com o K igual a 3.
Fonte: (Vaish, 2018)

Para calcular a distância entre dois pontos existem diferentes métricas disponíveis como: distância Euclidiana, distância de *Manhattan*, distância de *Hamming*, etc.

A métrica mais utilizada é a distância Euclidiana que é calculada pela equação 1.

$$\mathrm{D}(\mathrm{y},\mathrm{p}) = \sqrt{\sum_{i=1}^{n} (\mathrm{yi} - \mathrm{pi})2} \qquad (1)$$

Na equação, para um conjunto de *n* instâncias, y é a instância a classificar, yi o valor de cada variável de entrada dentro do somatório e p um ponto do conjunto de treino.

No fim de ser calculada a distância em relação aos pontos do *dataset* de treino, é necessário definir o valor de *k* a considerar. Esta tarefa é muito importante, já que, um *k* muito elevado leva a que o critério de similaridade seja descartado, enquanto que um *k* muito pequeno pode levar a que seja considerado ruído, comprometendo a eficácia do classificador. Este valor deve ser ímpar para descartar situações de empate entre classes e deve ainda ser validado usando o *dataset* de validação, de modo a obter o melhor desempenho (Fernandes, 2019).

As principais vantagens a realçar deste algoritmo são a sua simplicidade, quer na sua teoria quer na implementação, a sua capacidade de apreender funções complexas e não perder informação (Igawa, 2015). Como desvantagens, são de realçar, a facilidade de ser enganado por um atributo irrelevante, poder ser difícil achar o vizinho mais próximo quando o *dataset* possuir uma grande dimensão, ser um algoritmo que funciona preferencialmente com variáveis normalizadas (Dihanster, 2020).

Redes Neuronais

O desenvolvimento de redes neuronais inspira-se na forma de funcionamento dos neurónios do cérebro humano. Esta rede é descrita como uma estrutura conexionista, em que o processamento é distribuído por um conjunto de unidades de processamento simples que se encontram densamente interligadas. Cada unidade realiza um mapeamento das variáveis de entrada, numa variável de saída. As variáveis de entrada na rede correspondem aos atributos ou *features* que o conjunto de dados disponibiliza, enquanto que a variável de saída corresponde à *label* que se pretende prever. Na Figura 11, é possível verificar o funcionamento de uma rede neuronal com duas camadas intermédias.

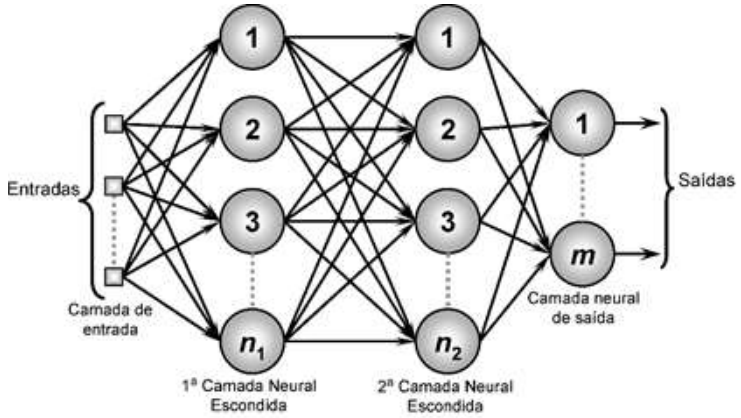


Figura 11 - Esquema de funcionamento de uma rede neuronal com múltiplas camadas. Fonte: (Rijo, 2017)

As redes neuronais podem ser simples ou profundas *(deep learning*), *d*ependendo do número de camadas existentes entre a camada de entrada e a de saída (Rijo, 2017). Para realizar a classificação, este algoritmo considera os atributos (x_m) que constituem o *dataset* na camada de entrada. Na segunda camada, as entradas correspondem aos valores obtidos da primeira camada. Na primeira camada oculta, é realizado o processamento em que são combinados os dados da camada de entrada com um conjunto de coeficientes que constituem os pesos (w_{km}) atribuídos às diferentes entradas, sendo estes ajustados consoante a aprendizagem da rede. Os resultados destas avaliações são somados. Posteriormente a soma passa por uma função de ativação, que determina como o sinal deve progredir na rede de forma a afetar o resultado. Esta função pode utilizar como ponderação o *bias* (b_k) que serve para aumentar ou diminuir a saída da função de ativação (por exemplo para entradas nulas, permite uma saída não nula). Por fim, é transmitido à camada de saída, o resultado de classificação (y_k) (Zampieri, 2006).

O modelo de funcionamento de um neurónio artificial está exemplificado na Figura 12.

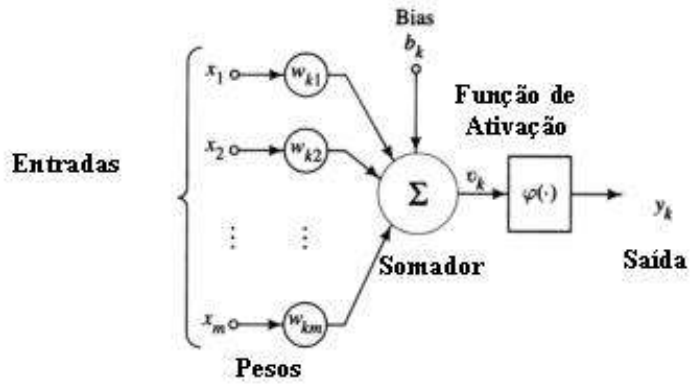


Figura 12 - Modelo de funcionamento de um neurónio artificial.
Fonte: (Zampieri, 2006)

Existem diversas funções de ativação, sendo as que mais se destacam a função sigmoide, linear, linear retificada e tangente hiperbólica. As principais vantagens das redes neuronais são: a capacidade de lidar com dados incompletos, imprecisos e com ruído; a grande adaptabilidade, pois uma vez criadas podem ser utilizadas em aplicações de tempo real sem necessidade de alterar a sua estrutura a cada atualização de dados; e a capacidade de mapear relações entre as variáveis de entrada e saída, mesmo quando estas são não lineares e complexas (Ambrósio, 2012). Como desvantagens, as redes apresentam uma lentidão no processo de aprendizagem/ treino, a necessidade de um grande volume de dados e a preparação previa destes.

Support Vector Machines

As *Support Vector Machines* (SVM) ou máquinas de vetores de suporte, são fundamentadas pela teoria de aprendizagem estatística desenvolvida pelo matemático Vapnik (Vapnik & Cortes, 1995).

O objetivo da SVM é classificar os dados de entrada entre uma de duas classes. Para tal utiliza um conjunto de dados de entrada e saída para treinar um modelo que depois é usado para classificar novos dados. Para desenvolver o modelo, mapeia as entradas de treino no espaço multidimensional e utiliza regressão para encontrar o hiperplano que divide com margem máxima as classes a prever.

Este classificador começa por procurar os pontos que se encontram mais próximos das diferentes classes para serem os vetores de suporte. Depois de se encontrarem os vetores de suporte, estes são conectados por uma linha. O melhor hiperplano é determinado pela reta perpendicular à linha de conexão entre os dois vetores de suporte (Lorena & Carvalho, 2007). Na Figura 13, encontra-se representado o esquema de funcionamento do classificador SVM para encontrar o melhor hiperplano entre duas classes a prever.

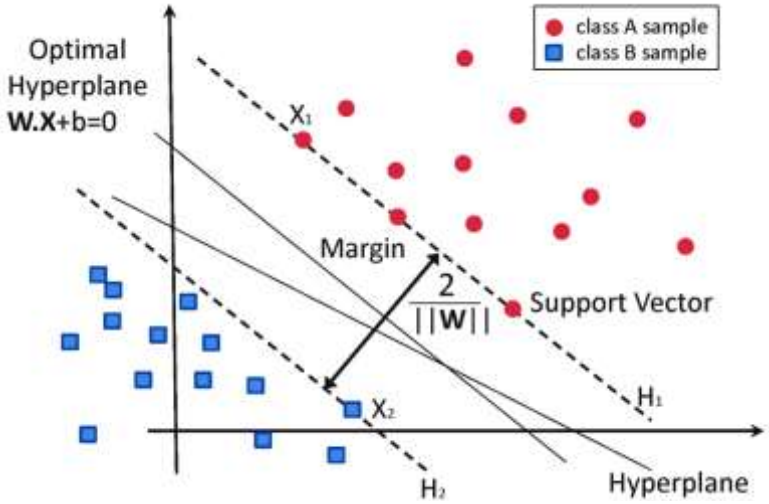


Figura 13 - Representação da escolha do hiperplano pelas SVM.
Fonte: (Gonzalo, Miniz, Nieto, Sanchez, & Fernandez, 2016)

Quando o problema em questão é multi-classe é necessário dividi-lo em classificações binárias, sendo que esta divisão pode ser *one-vs-one* ou *one-vs-all*. Na versão *one-vs-one*, é realizada a divisão em diferentes pares. Se existirem as classes A, B e C serão formados os pares (A,B), (A,C) e (B,C), e em cada caso o classificador irá votar na classe, sendo a classificação final a classe mais votada. A outra divisão *one-vs-all*, consiste em comparar cada classe com a junção das restantes classes (A, B+C), (B, A+C) e (C, A+B). A classe que ganhar será a escolhida como resultado da classificação.

Existem anda situações em que se torna necessária a escolha de uma SVM não linear, como quando não é possível a separação das classes por um hiperplano. Nesses casos, é preciso realizar uma transformação não-linear do espaço para que depois se possam separar os grupos por uma SVM linear.

As principais vantagens deste modelo são: a classificação de dados não regulares; encontrar a melhor solução possível para o problema de otimização; reduzir o risco de ocorrer *overfitting*; e permitir que seja utilizado em dados com grande quantidade de atributos (Coutinho, 2019). Como desvantagens, este revela uma dificuldade para o utilizador interpretar os cálculos efetuados quando o *dataset* é grande, sendo mais indicado para atributos numéricos (Campello & Mello, 2018).

Extreme Gradient Boosting

O algoritmo *Extreme Gradient Boosting* ou *XGBoost*, é baseado em árvores de decisão que utilizam uma estrutura de aumento de gradiente (Chen & Guestrin, 2016). Nesta técnica, o classificador constrói muitos modelos e, à medida que os vai construindo, fá-lo com base nos erros que os anteriores cometeram, de forma a corrigir essas deficiências.

No *XGBoost* cada novo modelo corresponde a uma nova árvore de decisão que é adicionada (Aderaldo, 2019). Este modelo dispõe de diversos módulos que realizam a regularização para deste modo evitar a ocorrência de *overfitting*, também executam nativamente a validação cruzada e trabalham com variáveis com valores em falta sem que seja necessário realizar o pré-processamento. Este modelo começa por realizar uma primeira aproximação da predição, sendo obtidos erros residuais que correspondem à diferença entre o valor observado e o valor previsto. De seguida, com esses erros, são treinados novos modelos para tentar prever o erro residual do primeiro modelo. Posteriormente, são somadas as predições de cada modelo para obter a previsão final e desta forma corrigir os valores da primeira aproximação. Este processo é repetido várias vezes até que os erros residuais sejam cada vez menores (Grover, 2017). Na Figura 14, está descrito o processo de classificação anteriormente referido.

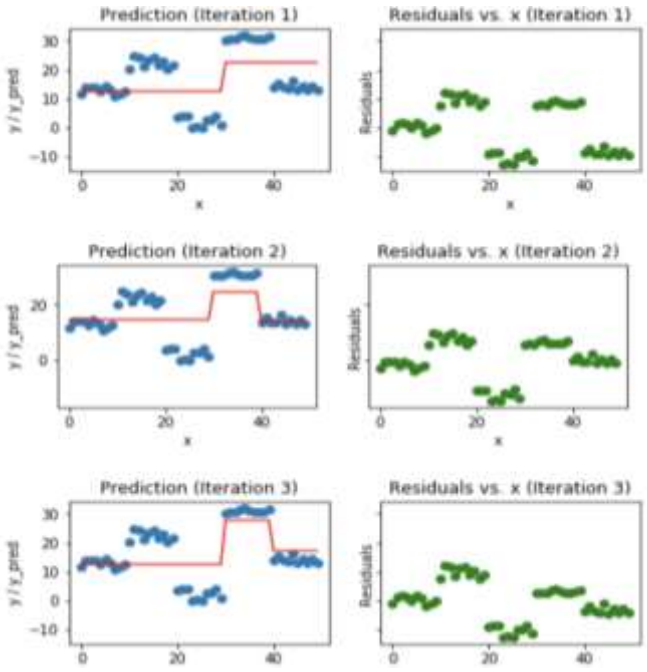


Figura 14 - Esquema representativo da modelo XGBoost.
Fonte: (grover, 2017)

As principais vantagens deste modelo são a capacidade de lidar eficientemente com uma grande variedade de dados, ser extremamente flexível devido ao número de hiperparâmetros e os bons resultados que obtém para problemas de regressão e classificação. A principal desvantagem deste método é a sua complexidade e a difícil perceção do que está a acontecer, conhecido como uma “caixa negra”, sendo também computacionalmente exigente.

Logistic Regression

Na *Logistic Regression* (LR) ou regressão logística, são utilizadas técnicas estatísticas para estimar as probabilidades associadas à ocorrência de cada evento. Este modelo é utilizado maioritariamente para problemas binominais, apesar de ser possível utilizá-lo em problemas multinominais. Para problemas multinominais a classe é prevista com base na fórmula 2.

$$E(Y) = \frac{e^{g(X)}}{1 + e^{g(X)}} \tag{2}$$

Na fórmula, *g(x)* é dado pela equação $\beta\mathbf{0} + \beta\mathbf{1}\boldsymbol{X}\mathbf{1} + \beta\mathbf{2}\boldsymbol{X}\mathbf{2} + \cdots + \beta\boldsymbol{p}\boldsymbol{X}\boldsymbol{p}$**,** correspondendo *Xp* às variáveis independentes que constituem o conjunto de dados.

Este algoritmo tem como vantagens ser extremamente simples e também dos mais conhecidos nesta área, servindo de base para outros algoritmos como é o caso das redes neuronais. Apesar da sua simplicidade, este modelo consegue obter bons valores de *accuracy*.

Como desvantagens, este algoritmo, devido à sua linearidade, tenta encontrar a separação entre as fronteiras como uma reta e, caso esta não se verifique, o modelo não consegue concretizar a separação. Outra limitação vem do fato de a regressão logística não garantir que a reta encontrada seja a que melhor separa as classes e que garante os melhores resultados possíveis (Facure, 2017).

2.5. Metodologias para avaliação do desempenho

Para avaliar o desempenho de um classificador é necessário que o *dataset* seja divido em dados para realizar o treino, em que a classe dos mesmos é conhecida, e em dados de teste, sendo estes diferentes dos de treino e desconhecidos, para validar o modelo. Esta divisão é muito importante para evitar que o modelo se ajuste exageradamente aos dados de treino. Quando tal acontece, os modelos acabam por memorizar os dados de treino e quando são submetidos a dados novos, os resultados que apresentam são bastante insatisfatórios.

Para realizar a divisão dos dados nos dois conjuntos, existem várias técnicas que podem ser aplicadas.

2.5.1. Holdout

A técnica *Holdou*t divide o conjunto de dados de tamanho *N* em dois subconjuntos, um a ser utilizado para treinar o algoritmo (*p*N*) e o restante conjunto para testar ((*1-p*)**N*). Estes subconjuntos são divididos aleatoriamente segundo uma

percentagem fixa, definida pelo utilizador. Os valores característicos utilizados são *p*=2/3 para o *dataset* de treino e *1-p*=1/3 para teste (Baranauskas, Métodos de Amostragem, 2019). O *Holdout* deve ser utilizado em conjuntos de dados que apresentem grandes quantidades, dado que poucos dados para treino e teste podem ser prejudiciais para o modelo. Outro problema que advém desta abordagem, é o facto de ao acontecer a divisão do *dataset,* esta poder ser desequilibrada, levando a que exista um maior número de exemplos de uma determinada classe num dos subconjuntos e menor no outro. Na Figura 15, está representada a divisão dos dados com esta abordagem.

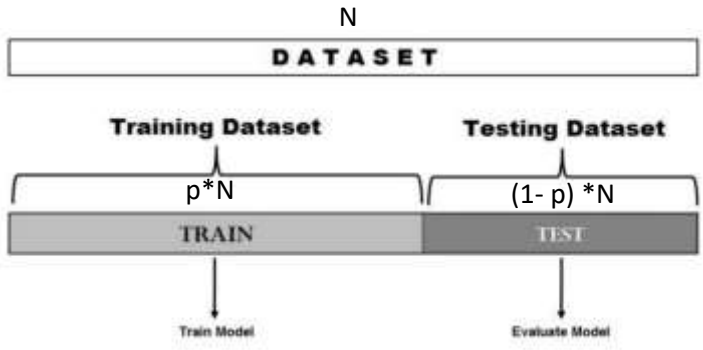


Figura 15 - Técnica Holdout para divisão do dataset.
Fonte: Adaptado de (Baranauskas, Métodos de Amostragem, 2019)

2.5.2. Amostragem aleatória estratificada

A amostragem aleatória pressupõe que o conjunto de dados é dividido em vários grupos (estratos) diferentes e que posteriormente, em cada um desses grupos, é realizada uma amostragem (Santana F. , 2020). Normalmente é utilizada a abordagem *Holdout*, em que, para cada subconjunto, são selecionados dados para teste na percentagem p e para treino na percentagem (1-p). Esta abordagem está ilustrada na Figura 16.

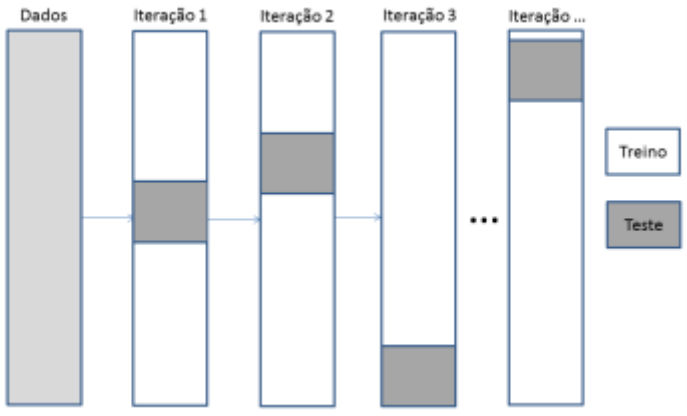


Figura 16 - Divisão dos dados segundo uma amostragem aleatória estratificada.
Fonte: (Duarte, 2015).

2.5.3. Validação cruzada

A validação cruzada realiza uma divisão aleatória dos dados em *K* subconjuntos, idealmente com a mesma dimensão ou semelhante, sendo comum utilizar o valor *K*=10 (Schneider, 2018).

Em cada iteração são selecionados os dados para treino, (*K*-1) conjuntos, e é selecionado um conjunto para teste sendo que este só pode ser utilizado uma vez. Este processo é realizado ao longo das *K* iterações. No final, o erro da classificação é calculado como a média dos erros em cada iteração (Santana R. , 2020). O processo de validação cruzada está representado na Figura 17.

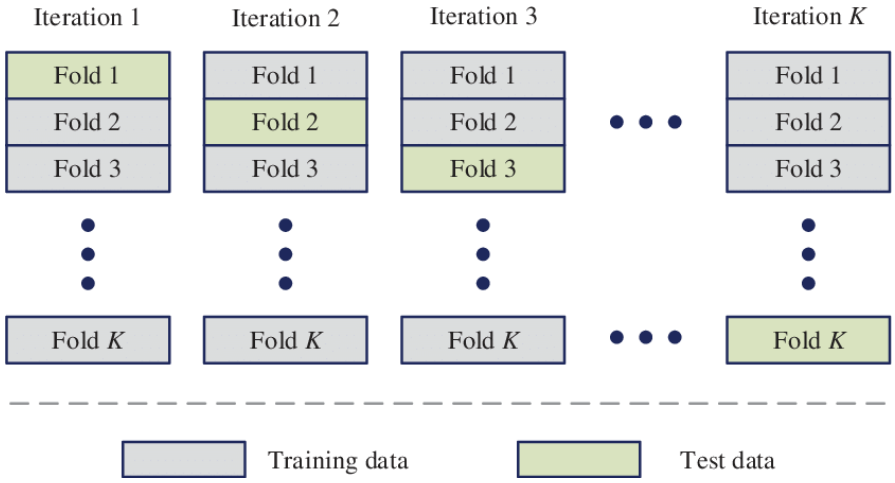


Figura 17 - Divisão do dataset utilizando a validação cruzada.
Fonte: (Qiubing, Mingchao, & Han, 2019)

2.5.4. Leave-one-out

A abordagem *Leave-one-out* é uma variação da validação cruzada. Neste método, como o próprio nome diz, é selecionado um exemplo para teste e os restantes exemplos são utilizados para treino. Este processo é realizado até que todos os exemplos sejam utilizados como teste, ou seja, o mesmo número que a quantidade de exemplos disponíveis no *dataset* original. O desempenho é calculado pela soma do desempenho em cada teste individual (SCHREIBER, et al., 2017). É indicado para *datasets* que possuem poucos dados, pois de outra forma é computacionalmente muito pesado. Esta abordagem está descrita na Figura 18.

Figura 18 - Abordagem de divisão de dados Leave-one-out.
Fonte: (DataCamp, 2020)

2.5.5. Bootstrap

O método *Bootstrap* pressupõe que são construídos K subconjuntos de dados (*bootstraps*) do *dataset* original. Estes conjuntos são gerados de forma aleatória e com reposição, contendo o *dataset* de treino o mesmo número de exemplos do *dataset* original, devido à existência de exemplos repetidos. O *dataset* de teste é construído com os exemplos que não foram considerados no conjunto de treino (Brownlee, 2018). Este processo de construção dos conjuntos de dados de treino e teste está representado na Figura 19.

Figura 19 - Abordagem Bootstrap para divisão de dados.
Fonte: (Brownlee, 2018)

2.6. Métricas para avaliação de desempenho

Quando se desenvolvem modelos de ML é fundamental avaliar o seu desempenho, quantificando a forma como cada modelo se comporta na resolução de um problema. De forma a ser possível realizar uma análise comparativa dos classificadores, existem algumas métricas que permitem uma correta avaliação dos resultados obtidos. Dependendo do tipo de classificador que se utiliza para o problema, as métricas utilizadas são distintas.

Quando o problema em questão é de classificação e binominal, ou seja, o classificador classifica entre as duas classes possíveis, as métricas utilizadas baseiam-se nos

valores retirados da matriz de confusão (Leal, 2017). Quando o problema em questão é multi-classe, como o resultado do jogo (empate, vitória, derrota), é necessário transformar a matriz de confusão, de forma a agrupar as diferentes classes, passando a ser um problema binominal, como é demonstrado na Figura 20.

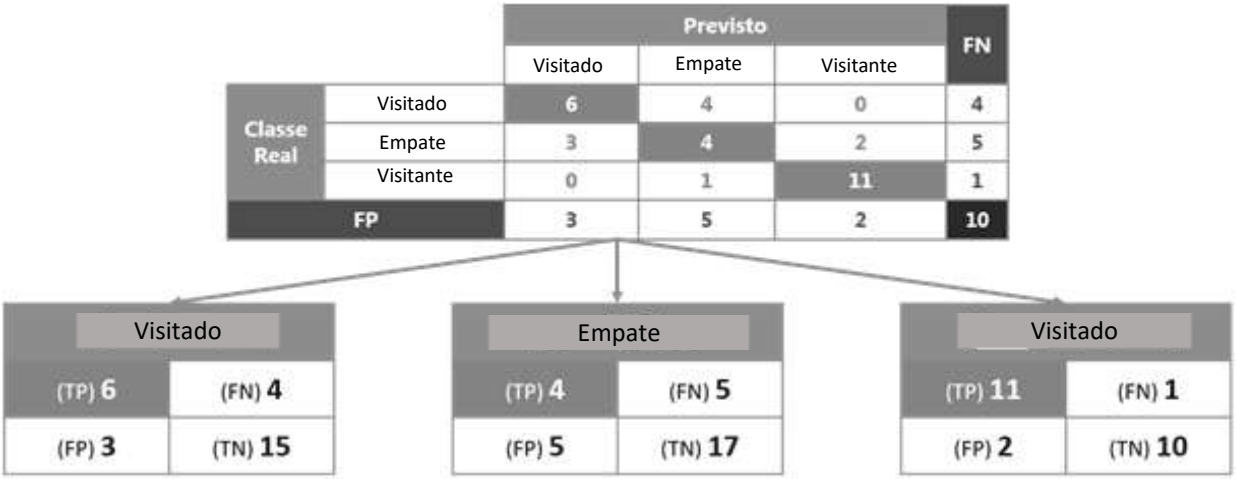


Figura 20 - Matriz de confusão para classificação multi-classe.
Fonte: Adaptado de (Leal, 2017)

Esta matriz é construída a partir de alguns valores retirados da classificação que o modelo realiza.

- Verdadeiros Positivos (VP) – este valor representa os exemplos que foram classificados corretamente pelo classificador como sendo positivos.
- Verdadeiros Negativos (VN) – este valor representa as instâncias que foram classificadas corretamente como negativas.
- Falsos Positivos (FP) – este valor representa os exemplos que foram erradamente classificados como sendo positivos, mas que eram exemplos negativos.
- Falsos Negativos (FN) – este valor representa os exemplos positivos e que foram classificados incorretamente como sendo instâncias negativas.

Utilizando este conjunto de valores retirados da matriz de confusão, existe um conjunto de métricas que podem ser calculadas para nos ajudar a perceber quais os classificadores que apresentam a melhor performance segundo cada dessas métricas e qual o que apresenta o melhor desempenho global (Schade, 2019).

Accuracy

A métrica *accuracy* ou taxa de acerto, representa a percentagem de exemplos que foram corretamente classificados. Esta varia entre 0 e 1, sendo que quanto mais perto de 1 estiver o valor, melhor o classificador está a classificar os exemplos. A *accuracy* é dada pela equação 3.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (3)$$

Taxa de erro

A taxa de erro (TE), representada na equação 4, fornece informação sobre a proporção de exemplos que foram classificados incorretamente pelo modelo, ou seja, o número de exemplos classificados como FP e FN em relação ao total de amostras classificadas.

$$TE = \frac{FP + FN}{VP + VN + FP + FN} \quad (4)$$

A taxa de erro também pode ser calculada em relação à métrica *accuracy*, apresentando uma relação inversa. Esta relação é dada pela equação 5.

$$TE = 1 - Accuracy \quad (5)$$

Precision

A métrica *precison* dá-nos a relação entre os exemplos positivos e classificados corretamente como tal e todas os exemplos classificados como positivos. Esta métrica permite ter informação sobre os falsos positivos. A equação 6 representa a fórmula que é utilizada para calcular a *precision*.

$$Precision = \frac{VP}{VP + FP} \quad (6)$$

Recall

O *recall* ou *sensitivity* é uma métrica que nos permite perceber a taxa de verdadeiros positivos. Este valor é utilizado para medir a fração de exemplos corretamente classificados como positivos em relação a todos os exemplos que são realmente positivos (VP e FN). Esta permite entender se o classificador está a identificar corretamente os valores positivos e pode ser calculada utilizando a fórmula 7.

$$Recall = \frac{VP}{VP + FN} \quad (7)$$

Specificity

A *specificity* permite aferir sobre a taxa de verdadeiros negativos. Este valor corresponde à relação entre os exemplos que foram corretamente classificados como negativos e todos os exemplos que são mesmo negativos (VN e FP). Esta métrica permite entender se o classificador está a identificar os exemplos negativos corretamente. A *specificity* é calculada a partir da equação 8.

$$Specificity = \frac{VN}{VN + FP} \tag{8}$$

F1-Score

A F1-Score ou F-*measure*, consiste numa média harmónica das métricas anteriormente referenciadas, *precision* e *recall*. Esta métrica é muito importante para ajudar a entender a qualidade geral do modelo. Este valor pode ser calculado a partir da equação 9.

$$F1 = \frac{2 * Recall * Precison}{Recall + Precison} \tag{9}$$

AUC

A métrica AUC, do inglês, Area Under the Curve, corresponde ao valor da área bidimensional que se encontra sob a curva ROC (Receiver Operating Characteristic Curve) que é obtida pela representação gráfica da taxa de verdadeiros positivos versus a taxa de falsos positivos. O seu valor varia entre 0 e 1, possuindo o valor de 0 quando o modelo classificar todos os exemplos erradamente e de 1 quando todas as previsões forem classificadas corretamente.

CAPÍTULO 3

CONTEXTUALIZAÇÃO DO TEMA EM ESTUDO E REVISÃO DA LITERATURA

Este capítulo inicia-se com uma breve introdução aos tópicos do futebol e das apostas desportivas *on-line,* à qual se segue uma pequena apresentação do *Traderline*. Prossegue com uma análise às aplicações concorrentes disponíveis no mercado e termina com a revisão de alguns trabalhos científicos realizados na área de interesse do presente estudo.

3.1. Futebol

O futebol é um desporto que já é praticado desde 3000 a.C. De facto, os registos mais antigos remontam a essa época e descrevem que os soldados chineses depois das guerras formavam duas equipas, cada uma com oito jogadores, e utilizavam a cabeça dos inimigos como bola, que passavam de pé em pé, até colocá-la dentro de duas estacas que simbolizavam a baliza do adversário, sendo uma prática muito semelhante ao futebol. Só muito mais tarde, no século XVII, quando este chegou a Inglaterra, é que começou a ser organizado, sistematizado e com regras para a sua prática (Reinke, A origem do futebol, 2019).

Este desporto, que na atualidade é dos que mais interesse suscita mundialmente, consiste em duas equipas, cada uma de 11 jogadores, que utilizam o corpo com exceção dos braços para levar a bola até à baliza do adversário (Herbinet , Predicting Football Results Using Machine Learning Techniques, 2018).

O jogo tem uma duração de 90 minutos, ocorrendo a meio do tempo um intervalo de 15 minutos que permite aos jogadores descansar. O vencedor do confronto é a equipa que atingir o maior número de golos. Caso ambas as equipas obtenham igual número de golos, o confronto acaba com um empate. Porém, neste último caso, se for um jogo decisivo é necessário realizar um prolongamento de 30 minutos para obter uma equipa vencedora. Se mesmo assim não houver um vencedor, ambas as equipas marcam penaltis, ganhando a que marcar mais golos nas 5 tentativas disponíveis. Durante um jogo de futebol podem ocorrer vários eventos, sendo os mais importantes:

Passes: ocorrem sempre que um jogador passar a bola a outro elemento da sua equipa, sem utilizar os membros superiores, para realizar essa troca.

Golos: é considerado golo sempre que a bola passa a linha da baliza da equipa adversária.

Remates: é o ato de uma das equipas tentar colocar a bola na baliza adversária, normalmente através de um pontapé, podendo ocorrer a várias distâncias da baliza.

Faltas: é assinalada falta sempre que o árbitro determina que foi realizada uma irregularidade. Estas podem ser colisões entre os jogadores com intenção de atingir o jogador adversário, troca de ofensas entre os mesmos, jogadas perigosas, toque com a mão na bola, etc.

Cartões: existem dois tipos de cartões que podem ser mostrados aos jogadores dependendo da gravidade da falta cometida por parte dos mesmos. Os cartões amarelos são mostrados quando ocorrem infrações com pouca gravidade, mas em que é necessário advertir o jogador. São exemplos deste tipo de infrações, a prática de atitudes antidesportivas, tais como desrespeitar as decisões tomadas pelo árbitro, abandonar o campo, entre outras. Os cartões vermelhos são apresentados sempre que existir bastante gravidade na infração cometida. Exemplos de infrações que levam à atribuição deste cartão são, obter dois cartões amarelos no mesmo jogo, faltas cometidas com intenção de magoar o adversário (entradas a pé juntos, agressões), entre outras.

Penáltis: este evento é assinalado sempre que a falta seja cometida dentro da zona da grande área. Após ser sinalizado pelo árbitro, a bola é colocada na zona de marcação de penálti, sendo que os únicos jogadores que podem estar dentro da grande área são o marcador e o guarda-redes da equipa que cometeu a falta.

Cantos: consiste em realizar um remate à baliza ou um passe a partir do canto do campo. Este tipo de evento ocorre quando a bola ultrapassa a linha de fundo e é tocada em último lugar por um jogador da equipa defensora.

3.1.1. Primeira Liga Inglesa

A Primeira Liga Inglesa, mais conhecida por *Premier League,* é considerada por muitos como o melhor campeonato do mundo, devido ao facto de ser a mais transmitida em todos os países e ao elevado número de atletas que jogam no Campeonato Mundial de Futebol. Esta liga é o nível mais alto do sistema de ligas existente no futebol Inglês e o começo da mesma aconteceu na temporada de 1992/1993 (Reinke, Entenda como funciona a Premier League, 2018).

Este campeonato é disputado por 20 equipas, ocorrendo em cada época (de agosto a maio) 38 jornadas. Em cada temporada, as equipas jogam entre elas, havendo um total de 38 jogos por cada equipa e acontecendo duas vezes a disputa entre as mesmas, em estádios diferentes, ou seja, no estádio do adversário e no estádio da casa. A *Premier League* utiliza o sistema de pontos acumulados para atribuir a classificação às equipas. A pontuação de cada equipa é regida pelo número de vitórias e empates que cada uma acumula. Cada vitória atribui 3 pontos ao vencedor e zero pontos à equipa derrotada não terá. No caso de empate, é atribuído 1 ponto a ambas as equipas.

No final das 38 jornadas, as equipas são organizadas pelo número de pontos que possuem e a que estiver em primeiro lugar será considerada campeã. No caso de

empate, serão considerados os pontos no confronto direto entre as equipas envolvidas. Se mesmo assim persistir o empate, será analisada a diferença entre os golos marcados e sofridos, bem como o número de golos marcados. Se ainda assim não conseguirem chegar a um vencedor, será agendado pela direção da *Premier League,* um jogo em campo neutro para determinar o campeão.

As equipas que ficarem nos 4 primeiros lugares garantem acesso direto à Liga dos Campeões (*UEFA Champions League*), enquanto o 5º e 6º classificados se qualificam para a Liga Europa (*UEFA Europa League*). As equipas que obtiverem as três piores classificações passarão a jogar na segunda divisão inglesa, ou seja, na *English Football League Championship* (Prasad, Kurian, Bhatia, & Tan, 2020).

3.1.2. Primeira Liga Portuguesa

A Primeira Liga, foi criada na época de 1934/1935 pela Federação Portuguesa de Futebol, passando a ser gerida pela Liga Portuguesa de Futebol Profissional na época de 1995/1996. Este campeonato corresponde à liga mais alta existente no sistema do futebol português.

Atualmente, na época de 2019/2020, a primeira liga é formada por um conjunto de 18 clubes, que se qualificam segundo um sistema de pontos que é análogo ao da *Premier League.* Ou seja, os pontos são atribuídos consoante o resultado da disputa: se a equipa vencer ganha 3 pontos, se empatar recebe 1 ponto e se acabar o confronto derrotada, não recebe nenhum ponto.

Ao longo das 34 jornadas da época, as equipas defrontam-se entre si, ou seja, uma equipa realiza jogos contra todas as outras equipas. As mesmas equipas defrontam-se duas vezes na mesma época, sendo realizados dois jogos, um no estádio da equipa e outro no estádio da equipa adversária, sendo a ordem dos jogos estabelecida por sorteio.

No final dos 306 jogos, é considerada campeã a equipa que acumular mais pontos ao longo do campeonato. Existem ainda regras definidas pela liga que devem ser seguidas no caso de ocorrer igualdade de pontos (LigaPortugal, 2018). A equipa que se sagrar campeã, atualmente garante acesso direto à *UEFA Champions League*, enquanto a que ficar em segundo lugar terá que disputar a 3ª pré-eliminatória da prova milionária, o terceiro e quarto lugar classificam-se para competir, respetivamente, na 2ª e 3ª pré-eliminatórias da *UEFA Europa League.* Estes acessos variam de ano para ano, pelo que estas regras são válidas para a época 2019/2020. Em oposição, os dois clubes que obtiverem as duas pontuações mais baixas descem para o nível inferior, a Segunda Liga, e os que obtiverem os dois primeiros lugares na Segunda Liga sobem para a Primeira Liga. Esta última é a que mais adeptos angaria nas ligas do futebol português (Portugal, 2020).

3.2. Apostas desportivas online

O conceito de Casa de Apostas tem origem na língua inglesa, sendo tradicionalmente conhecida por *Bookmarker*. A primeira abordagem de apostas a dinheiro aconteceu em corridas de cavalos, sendo o seu objetivo prever o cavalo que terminaria em primeiro lugar. As primeiras casas de apostas surgiram na Europa, na década de 1960, após o *Gambling Act*, um ato do parlamento britânico que legalizou formas adicionais de jogos de azar no Reino Unido. Foi a partir dessa data que as casas de apostas profissionais começaram a operar, sendo as pessoas a controlar o processo de aceitação de propostas e prémios.

A versão das apostas desportivas *online* só começou a funcionar na década de 90 do século XX, tendo sido lançada em 1995 a primeira casa de apostas da Grã-Bretanha "*William Hill*" (Moura Ferraz, 2011). O aparecimento das apostas *online* tornou possível a qualquer pessoa apostar através de um simples clique, durante 24 horas por dia e em *real-time*, ou seja, durante o decorrer de um jogo (Gomes T. , 2019).

Uma casa de apostas tem como principal objetivo ganhar dinheiro, com a colocação de um valor previamente definido numa determinada aposta para a previsão de um resultado de um evento desportivo. O valor atribuído a cada aposta é designado por *odd* e corresponde ao montante que irá ser pago no caso de acerto da mesma, sendo este estipulado com base na probabilidade associada a cada acontecimento do evento desportivo. As *odds* chegam a ser disponibilizadas até uma semana antes do jogo e, como consequência disso e do número de apostas num determinado resultado, podem sofrer variações (M., 2017).

Os desportos que despertam maior interesse nas apostas desportivas são o futebol, basebol e basquetebol.

Existem atualmente dois tipos de casa de apostas. No primeiro, mais convencional, o utilizador realiza uma aposta contra a casa: caso esta ganhe, fica com o lucro; caso perca, é o utilizador que ganha o lucro. São exemplos destas casas a *Bet365* e a *Bwin*. O segundo tipo envolve o conceito de *trading*, mais complexo, o qual funciona de maneira análoga à bolsa de valores, sendo um mercado de compra e venda. Ou seja, é um intercâmbio de apostas (*betting exchanges*), em que os apostadores não apostam contra ou a favor das *odds* estipuladas pelas casas de apostas, mas sim contra outros apostadores, negociando entre si e com os valores determinados pelos mesmos. Neste caso, as casas de apostas beneficiam cobrando uma taxa pelos lucros obtidos, sendo que esta varia consoante o montante e a regularidade com que utilizam a aplicação. São exemplos destas casas de apostas, a Betdaq e a Betfair. Esta modalidade é muito apreciada por traders2 que utilizam este mercado como fonte de rendimento (Moura Ferraz, 2011).

[2] Profissional que investe na bolsa de apostas

Existe uma enorme quantidade de apostas que podem ser realizadas, sendo que, no caso de um jogo de futebol, os principais mercados de apostas são:

- Resultado final – o utilizador aposta no empate ou na vitória da equipa visitante/visitada;
- Resultado ao intervalo – o utilizador aposta na vitória da equipa visitada/visitante ou empate ao intervalo;
- Resultado correto – o utilizador aposta no número de golos marcados por cada equipa, no final do confronto;
- Total de golos – o utilizador aposta se o total de golos marcados no confronto é superior ou inferior a um determinado número;
- Total de cantos – o utilizador aposta se o total de cantos marcados no confronto é superior ou inferior a um determinado número;
- Total de cartões – o utilizador aposta se o total de cartões assinalados no confronto é superior ou inferior a um determinado número;
- Ambas as equipas marcarem – o utilizador aposta em ambas as equipas marcarem golos ou na condição oposta;
- Primeira equipa a marcar – o utilizador aposta na equipa visitante/visitada para ser a equipa a marcar o primeiro golo;
- *Handicap* Asiático – é dada uma vantagem ou desvantagem de golos às equipas para equilibrar os valores das *odds* e trazer uma maior probabilidade de vencer a aposta.

É ainda possível realizar uma aposta múltipla, em que é executada uma conjugação de vários mercados de apostas e o sucesso da aposta só acontece caso ganhe as apostas de todos os mercados selecionados.

Em Portugal, a legalização das apostas desportivas ocorreu no final de 2015 com a criação de regulamentação e legislação para este mercado, obrigando a que muitas das casas que operavam em Portugal ficassem suspensas (M., 2017). Foi em maio de 2016 que a *BetClic.pt* obteve a primeira licença para poder operar legalmente em Portugal, seguindo-se a *Bet.pt*. Atualmente existe um total de 9 casas de apostas legais em Portugal: *Betclic*, *Bet*, *Betano*, *Betway*, *NossaAposta*, *Moosh*, *EscOnline*, *Luckia* e *Solverde*.

3.3. O Traderline

Tal como referido anteriormente neste documento, o *Traderline* é um software de apostas certificado pela *Betfair* (Betfair, 2021)(a empresa que opera a maior bolsa de apostas *online* do mundo), tendo sido desenvolvido pela Mythical Technologies para responder às necessidades dos apostadores que procuravam uma ferramenta de apostas e de

trading completa, simples e prática. Com efeito, o Traderline é o primeiro *software* totalmente Português, que permite aos utilizadores fazerem *trading* na *Betfair* de forma automática. Este começou a ser desenvolvido no final de 2012, tendo sido lançada a sua primeira versão em janeiro de 2013 (Serafin, 2016). Este *software* foi inicialmente lançado no mercado português, com a possibilidade de afiliação a quem realizasse a sua divulgação e aumentasse o seu renome nas comunidades de apostadores. Esta modalidade fez com que o *Traderline* crescesse de forma exponencial e em apenas alguns meses passasse a ser uma das melhores plataformas para *trading* desportivo, a ser utilizada em mais de 60 países. Após a certificação deste por parte da *Betfair* através da utilização da *API (Application Programming Interface)* da mesma, foi garantido que o processamento das apostas era realizado por esta e que era possível realizar qualquer tipo de ação sobre a bolsa de apostas desta empresa (Gonçalves, 2019).

Atualmente, o Traderline encontra-se disponível para ser utilizado em *Windows*, *Mac OSX, Android e IOS*, podendo ser escolhidos diversos planos de subscrição, desde planos mensais a planos anuais e vitalícios por diferentes preços (Traderline, 2021). Na Figura 21, é possível ver um exemplo da intuitiva *interface* deste produto.

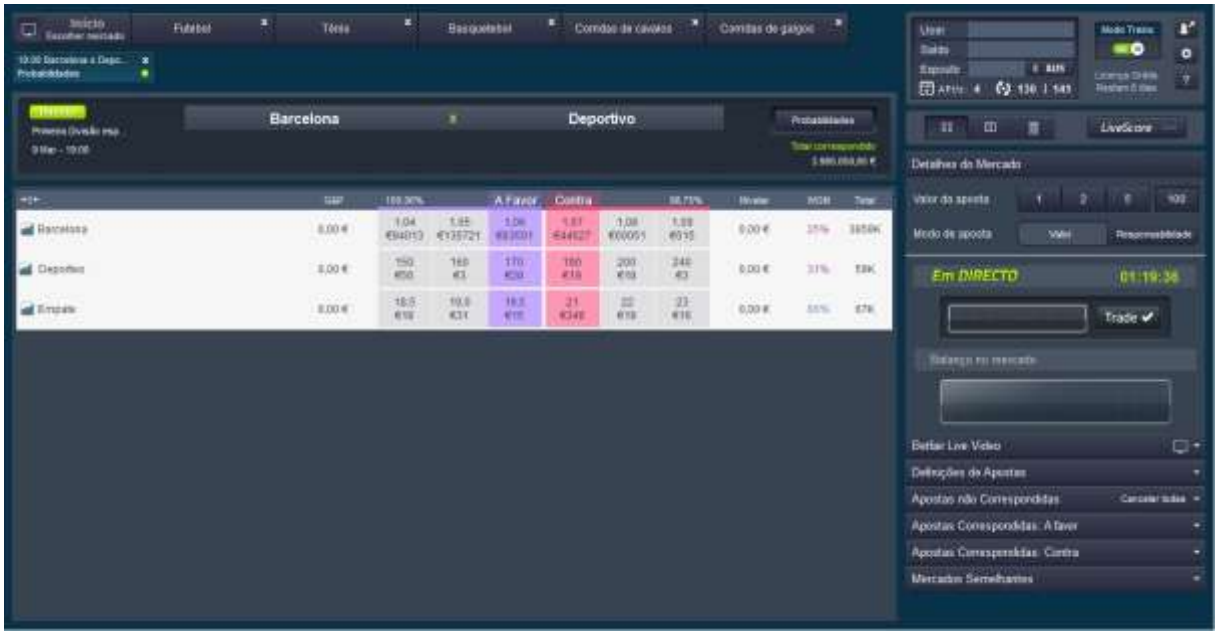


Figura 21 - Interface da janela principal do Traderline
Fonte: (Traderline, 2021)

3.4. Produtos concorrentes

Atualmente existe uma grande quantidade de casas de apostas e ferramentas que nos permitem realizar apostas desportivas, mas a grande maioria não está credenciada para poder operar em Portugal (ApostaLegal, 2020). No âmbito do presente trabalho de estágio, foi realizada uma análise aos principais produtos concorrentes com o *Traderline*, cujo resultado se apresenta em seguida. A ideia era perceber o posicionamento deste último, relativamente à oferta existente no mercado.

Na Tabela 1, é apresentada uma lista com os *websites* mais utilizados mundialmente e algumas das suas características, tais como: se estão disponíveis em Portugal, se apresentam estatísticas sobre os confrontos e se facultam previsões/ prognósticos dos resultados ao utilizador. A informação diz apenas respeito à modalidade de futebol.

Tabela 1 - Tabela comparativa dos websites de apostas mais utilizados no mundo

Website	Disponível em Portugal	Estatísticas	Previsões/ Prognósticos
Betano (Betano, 2020)	*X*	*X*	-
Betway (Betway, 2020)	*X*	*X*	-
Esc Online (EscOnline, 2020)	*X*	*X*	-
Betclic (Betclic, 2020)	*X*	-	-
Sportingbet (SportingBet, 2020)	-	*X*	-
Pinnacle (Pinnacle, 2020)	*X*	-	-
Betfair (Betfair, 2020)	-	*X*	-
Rivalo (Rivalo, 2020)	-	X	-
Suprabets (SupraBets, 2020)	-	X	-
SuperAposta (SuperAposta, 2020)	X	-	-
888Sport (888Sport, 2020)	-	X	-
Tipbet (TipBet, 2020)	-	X	-
Sportytrader (SportyTrader, 2020)	X	-	X
Traderline ((Traderline, 2021)	-	X	-

Analisando a tabela, percebe-se que quase todas disponibilizam ao utilizador estatísticas sobre os confrontos mas apenas uma, a *Sportytrader*, fornece prognósticos ou previsões dos jogos. Da pesquisa efetuada, verificou-se que os dados estatísticos disponibilizados incluem: informação histórica comparativa dos confrontos das duas equipas, a forma das equipas, informação acumulativa relativa aos golos marcados, sofridos, número de vitórias, empates, derrotas e pontos no campeonato.

Para além da plataforma Sportytrader já referida, existem outros websites menos populares que disponibilizam ao utilizador informação com prognósticos dos jogos, sugerindo a aposta que é considerada ideal. Essas ferramentas efetuam uma previsão em termos de probabilidades matemáticas, apresentando ao utilizador as probabilidades associadas a cada resultado e a cada evento. Alguns exemplos dessas outras aplicações são: *Statarea* (Statarea, 2021), *Vitibet* (Vitibet, 2021), Forebet (Forebet, 2021), *PredictZ* (PredictZ, 2021) e *SoccerVista* (Soccervista, 2021).

Verificou-se que todos estes *websites* utilizavam cálculos matemáticos para apresentar a previsão do resultado ao utilizador, distinguindo-se dos modelos de *Machine Learning* que se pretende implementar no *Traderline*.

3.5. Trabalho científico relacionado

O estudo realizado por Buursma (Buursma, 2010), tinha como principal objetivo desenvolver um sistema para prever resultados de jogos de futebol que superasse as *odds* das casas de apostas. Com base em dados históricos, foram selecionados os seguintes atributos ou *features*: golos marcados pela equipa nos últimos x jogos, golos sofridos pela equipa nos últimos x jogos, número médio de pontos obtidos pela equipa nos últimos x jogos. Depois da escolha dos atributos referidos, foram implementados 7 classificadores com base nos seguintes algoritmos: regressão linear, regressão logística, árvores de decisão, *Boosting*, *Bayesian networks*, *NaiveBayes* e um que permitia prever as vitórias da equipa da casa. Para a implementação o autor usou a plataforma *WEKA* que permitia visualizar as correlações entre os vários atributos e utilizar os algoritmos de *Data Mining*. No estudo, foi ainda testada a influência que o número de jogos tinha no desempenho dos classificadores, e a conclusão foi que, quanto maior fosse o número de jogos, melhor seria o desempenho do classificador. Foi então decidido que o classificador devia utilizar 20 jogos, para ter um desempenho razoável. Por fim, para os 20 jogos, foram selecionados diferentes conjuntos de *features* e treinados os vários classificadores com esses subconjuntos. Os melhores resultados foram obtidos pelo classificador de regressão linear com todas as *features* utilizadas, atingindo 55% de *accuracy*.

Num outro estudo realizado por Duarte (Duarte, 2015) foram aplicadas técnicas de *Data Mining* a dados históricos da primeira liga portuguesa, para realizar a previsão de resultados de jogos de futebol. Neste estudo, o autor utilizou os algoritmos *C5.0*, *Random Forest*, *KNN*, *Jrip*, *SVM*, redes neuronais e *Naive Bayes* para realizar a previsão do resultado. Para a implementação dos algoritmos de *Machine Learning* foi utilizada a biblioteca *Scikit-learn*. Os atributos usados nesta pesquisa foram: a identificação do jogo, a identificação da equipa visitada e da visitante, a jornada, o histórico do número de golos marcados por cada equipa, a forma das equipas e o fator casa. Este autor realizou quatro testes para perceber a interferência da correlação entre os atributos no *dataset*. Primeiramente, construiu o *dataset* completo com todos os atributos, de seguida construiu 3 *datasets*, um com todos os atributos que possuíssem correlações inferiores a 0.9, outro com correlações inferiores a 0.8 e, por fim, com os atributos que tivessem menos de 0.7 de correlação. O melhor resultado foi obtido para os algoritmos *KNN* e *SVM* com uma *accuracy* de 59%.

Por seu lado, Tavares (Tavares, 2013) realizou uma pesquisa em que tentava prever o resultado (vitória, empate ou derrota) dos jogos de futebol do campeonato Brasileiro. O autor utilizou os dados desde a época de 2002/2003, até à primeira metade (início do mês de dezembro) da época de 2013/2014, para realizar o treino do modelo e usou

os dados da segunda metade desta última época, para realizar o teste. Este autor implementou dois modelos utilizando *Matlab*: o *Maximum Likelihood Estimator* (MLE) e *Multilayer Perceptron* (MP), para posteriormente comparar os resultados entre si e averiguar qual o modelo mais adequado para a previsão de jogos de futebol. Foi utilizado um vetor de *features* que continha os dados dos últimos 10 anos do campeonato Brasileiro, com informação sobre: o local onde o jogo se tinha realizado para considerar o fator casa, o resultado do jogo (vitória, derrota ou empate) e a data do evento para que fosse atribuído maior peso aos eventos que tivesse ocorrido recentemente. Como resultados, obteve um valor de *accuracy* de 53,16% para o classificador MLE e de 55,79% para o modelo MP. O estudo concluiu que os modelos apresentavam pior desempenho na classificação da classe empate.

No trabalho desenvolvido por Stefan Samba (Samba, 2019), foi investigada a capacidade de um sistema de redes neuronais prever o resultado de jogos de futebol da *English Premier League*. Neste estudo, as variáveis utilizadas foram classificadas consoante fossem dependentes ou independentes das equipas que estavam a disputar o evento. Como *features* dependentes, foram utilizadas a força da equipa e a soma cumulativa de diferentes parâmetros como: os pontos, os golos, as faltas, os cartões e os cantos. As *features* independentes consideradas foram o árbitro, a época, a divisão, a forma, os dias entre os diferentes jogos e as *odds*. Como *target*, foram utilizadas as três hipóteses possíveis para o desfecho do confronto: empate, vitória da equipa que joga em casa ou vitória da equipa que joga fora. Este autor testou a interferência da alteração de diversos parâmetros na rede neuronal, tais como a profundidade da rede, o número de neurónios e o número de épocas de treino. O valor mais alto para a melhor configuração de rede, obteve uma *accuracy* de 48 %.

No trabalho desenvolvido por Ulmer e Fernandez (Ulmer & Fernandez , 2014) foram utilizados algoritmos de *Machine Learning* para prever as classes ganhar, perder ou empatar. Neste estudo, foram empregues os dados dos resultados dos jogos da *Premier League*, sendo que utilizaram 10 temporadas (2002/2003 até 2011/2012) para construir o *dataset* de treino e 2 temporadas (2012/2013 e 2013/2014) para serem utilizadas como teste. Como *features* de entrada, estes autores utilizaram o *ranking* de cada equipa, tendo em conta a diferença entre pontos das equipas. Foi também utilizada a forma atual das equipas, sendo esta calculada pelos resultados obtidos nas últimas 7 partidas e definidos pesos para atribuir menor importância às partidas mais antigas. Além do desempenho, tiveram em consideração o local onde o jogo era realizado, para ser considerada a vantagem de a equipa estar a jogar em casa. Com estes conjuntos de dados, foram criados os *modelos Stochastic Gradient Descendent, Naive Bayes, Hidden Markov Model, Support Vector Machine e Random Forest*. De entre os diversos classificadores, os que apresentaram melhor performance foram o *Support Vector Machine*, o *Random Forest* e o *Stochastic Gradient Descendent* na configuração *one-vs-all*. Estes classificadores apresentaram valores de *accuracy* entre 48% e 52%, tendo este último sido obtido pelo modelo *Stochastic Gradient Descendent* na configuração *one-vs-all*.

Num dos estudos mais recentemente realizados, os investigadores *Denny et al* (Palinggi, Ramos , Torres-Sospedra, Trilles, & Huerta, 2019) utilizaram técnicas de *Machine Learning* para tentar prever os resultados de jogos de futebol das ligas espanholas. Neste trabalho, foram considerados como atributos as condições meteorológicas e os dados históricos estatísticos das equipas. O estudo foi realizado com os dados relativos às épocas de 2013-2014 até 2016-2017, perfazendo um total de 3830 jogos, disputados entre as equipas da La Liga e da Segunda Divisão do campeonato espanhol. Foram testados três tipos de algoritmos de classificação para perceber qual iria obter melhor resultados. Os algoritmos utilizados foram o RF, KNN e SVM . Estes algoritmos foram implementados em linguagem *Python*, utilizando a biblioteca *scikit-learn*. Para testar a influência que os dados meteorológicos tinham no *dataset*, foram realizados dois casos de estudo: no primeiro foram consideradas exclusivamente as condições meteorológicas e no segundo foram utilizadas as condições meteorológicas e os dados históricos das estatísticas de cada equipa. Tendo em conta os dois casos, os que apresentaram melhores resultados, foi o segundo com o SVM, obtendo uma percentagem de 49,51% de *accuracy*. Para o primeiro caso, a *accuracy* de 47,59% quando utilizado o mesmo algoritmo.

A Tabela 2, apresenta um resumo da informação mais relevante dos estudos descritos anteriormente, tais como as *features* selecionadas, os algoritmos implementados, as ferramentas/bibliotecas utilizadas e os melhores resultados obtidos.

Tabela 2 - Resumo dos trabalhos científicos realizados na área em estudo

Nome do estudo	*Features*	Algoritmos	Resultados obtidos (accuracy)
Predicting sports events from past results (Buursma, 2010)	• Golos marcados pela equipa da casa/visitante nos últimos x jogos; • Golos sofridos pela equipa da casa/visitante nos últimos x jogos; • Número médio de pontos obtidos pela equipa da casa/visitante nos últimos x jogos.	• Regressão linear; • Regressão logística; • Árvores de decisão; • *Boosting;* • *Bayesian networks*; • *NaiveBayes;* • Algoritmo que prevê as vitórias da casa.	55%
Previsão de resultados de jogos de futebol (Duarte, 2015)	• Identificação da equipa visitada e visitante; • Jornada; • Histórico do número de golos marcados por cada equipa; • Forma; • Fator casa.	• C 5.0; • *Random Forest*; • *K-Nearest Neighbors*; • Jrip; • *Support Vector Machine*; • *Naive Bayes*; • Redes neuronais.	59%

Tabela 2 - Resumo dos trabalhos científicos realizados na área em estudo (cont.)

Nome do estudo	*Features*	Algoritmos	Resultados obtidos (accuracy)
Predicting Results of Brazilian Soccer League Matches (Tavares, 2013)	• Dados históricos dos últimos 10 anos com: local dos jogos;resultado dos jogos;data dos jogos.	• *Maximum Likelihoodestimator;* • *Multilayer Perceptron.*	55,79%
Predicting Soccer Match Results in the English Premier League (Ulmer & Fernandez , 2014)	• *Ranking*; • Forma; • Força da equipa; • Fator casa.	• *Stochastic Gradient Descendent;* • *Naive Bayes;* • *Hidden Markov Model;* • *Support Vector Machine;* • *Random Forest.*	52%
Football Result Prediction by Deep Learning algorithms (Samba, 2019)	• Árbitro; • *Odds*; • Época; • Divisão; • Força da equipa; • Somas acumulativas dos eventos; • Forma; • Dias entre os jogos.	• *Multilayer Perceptron.*	48%
Using Weather Condition and Advanced Machine Learning Methods to Predict Soccer Outcome (Palinggi, Ramos , Torres-Sospedra, Trilles, & Huerta, 2019)	• Condições meteorológicas; • Dados históricos estatísticos.	• *Support Vector Machine.* • K-*Nearest Neighbours;* • *Random Forest.*	49,51%

A revisão dos trabalhos anteriormente apresentados, permitiu perceber a diversidade das *features* selecionadas pelos vários autores, bem como a variedade dos algoritmos testados. Por outro lado, permitiu concluir que, em regra geral, os melhores resultados, em termos de *accuracy*, se situam na gama de valores 48-59%. Apesar de estes serem relativamente baixos, facilmente se explicam pela complexidade do problema em causa.

Resta dizer que toda a informação recolhida neste estudo, se revelou de grande utilidade no desenvolvimento do trabalho que é apresentado nos capítulos seguintes.

CAPÍTULO 4

CONTEXTO TECNOLÓGICO

Neste capítulo, procede-se à apresentação do contexto tecnológico que esteve na base do trabalho desenvolvido.

4.1. Python

O *Python* (Python, 2021) é uma linguagem funcional, de programação de alto nível e orientada a objetos.

Neste trabalho, foi utilizado o *python* como linguagem de programação na versão 3.8 e como ambiente de desenvolvimento, o *Pycharm* (JetBrains, 2021) e o *Jupyter Notebook* (Jupyter, 2021).

Figura 23 - Logótipo do Python
Fonte: (Isaac, 2015)

4.2. Bibliotecas python

Algumas das bibliotecas *Python* para M*achine Learning* utilizadas foram:

Numpy (NumPy, 2021) – biblioteca matemática que permite realizar diversas operações com os dados sobre a forma de *arrays.*

Pandas (Pandas, 2021) – utilizada para realizar o processamento/análise dos dados, principalmente para realizar a junção dos dados.

Scikit-learn (Scikit-Learn, 2021)– disponibiliza diversos algoritmos de classificação, regressão e *clustering*. Foi empregue para utilizar os algoritmos de classificação.

Figura 24 - Logótipo da scikit-learn
Fonte: (Muller, 2015)

4.3. Tensorflow

O *tensorflow* (Tensorflow, 2020) é uma biblioteca para M*achine Learning* que foi desenvolvida pela *Google* Brain e ficou disponível em 9 de novembro de 2015, sobre a licença de código aberto Apache 2.0. Neste trabalho, esta biblioteca foi utilizada para ajudar no processo de adquirir os dados, treinar os modelos, fornecer as previsões e refinar/demonstrar os resultados de maneira mais intuitiva.

Figura 25 - Logótipo da TensorFlow
Fonte: (Tensorflow, 2020)

A biblioteca *Tensorflow* também utiliza a biblioteca *Keras* (Keras, 2021) para desenvolver o modelo de redes neuronais.

4.4. RapidMiner

O RapidMiner (Rapidminer, 2020) é uma plataforma de análise de dados desenvolvida pela empresa com o mesmo nome. Fornece um ambiente integrado para preparação de dados, M*achine Learning, Deep Learning,* análise preditiva, etc. Esta ferramenta foi utilizada em algumas fases do pré-processamento dos dados e no teste de alguns algoritmos de predição ao longo da realização deste trabalho.

Figura 26 - Logótipo RapidMiner. Fonte: (Rapidminer, 2020)

4.5. MongoDB

O *mongoDB* (MongoDB, 2020) é uma base de dados NOSQL *open-source*, orientada a documentos no formato *JSON*. Este software é muito flexível, permitindo gravar grandes quantidades de dados, na forma que melhor se adequar para que estes sejam aplicados. Neste trabalho o *MongoDB* foi utilizado para guardar os dados.

Figura 27 - Logótipo do MongoDB. Fonte: (MongoDB, 2020)

4.6. PostMan

O *PostMan* (Postman, 2021) é uma ferramenta que permite ao utilizador testar os pedidos que são realizados a uma *API* de uma forma mais rápida e eficiente. Esta foi utilizada para visualizar os dados que eram disponibilizados pela *Football API* (API, 2021) ao realizar determinados pedidos.

Figura 28 - Logótipo do PostMan. Fonte: (San, 2018)

4.7. Anaconda

A *Anaconda* (Anaconda, 2020) é uma plataforma *open-source* que disponibiliza uma simples instalação de diferentes packages, permitido ainda a criação de diversos ambientes virtuais. Estes permitem realizar uma gestão adequada das funcionalidades pretendidas. No presente trabalho, esta plataforma foi utilizada para a criação de diferentes ambientes, contendo estes a instalação das diferentes bibliotecas de *Machine Learning* para as diferentes tarefas.

Figura 29 - Logótipo da Anaconda. Fonte: (Anaconda, 2020)

CAPÍTULO 5

COMPREENSÃO E PREPARAÇÃO DOS DADOS

Este capítulo descreve as fases que envolveram a recolha dos dados, a identificação das variáveis, a análise uni e multivariada dos dados, a limpeza e transformação dos mesmos, bem como a seleção das variáveis a incluir no dataset provisório. Por fim, apresenta a construção dos vários datasets a serem utilizados na previsão dos eventos de futebol.

5.1. Seleção da fonte de dados

Numa primeira fase foi efetuada uma análise comparativa das principais fontes de dados encontradas *online*, para selecionar a que melhor se adequava a este estudo. Para este efeito, foi criada uma tabela com as diferentes fontes de dados e assinaladas as *features* que estas apresentavam, de entre um conjunto considerado como o mais relevante para o problema em questão. Nomeadamente: o número de toques, o número de remates falhados, a posse de bola, o número de golos, o número de remates, o número de passes, o número de passes para o adversário, o número de desmarcações com sucesso e insucesso, cruzamentos de sucesso e insucesso, comprimento dos passes, número de *dribles*, tipo de competição, categoria da equipa, sucesso na época, fator casa, número de cartões, número de faltas, bloqueios, substituições e ainda as condições meteorológicas. De entre estas, algumas *features* foram assumidas como imprescindíveis e para as quais era obrigatória a existência de dados, como é o caso de: número de golos, número de cantos, número de cartões, número de faltas, substituições realizadas, qual o tipo de competição, qual a categoria e o sucesso na época. A informação referida encontra-se na Tabela 3.

Tabela 3 - Tabela comparativa dos features disponibilizadas por cada fonte de dados

Features	Fontes de Dados							
	SportsData.io	Betfair	Eoddsmaker	Betradar	RunningBall	SportRadar	API football	Football Data
Nº de toques						X		
Nº de remates falhados					X	X	X	X
Posse de bola						X	X	
Nº de golos	X	X	X	X	X	X	X	X
Nº de remates					X	X	X	X
Nº de passes					X		X	
Nº de cantos					X		X	X
Nº de passes para o adversário					X			
Desmarcações com sucesso							X	

Tabela *3* - Tabela comparativa dos features disponibilizadas por cada fonte de dados (*cont.)*

Features	Fontes de Dados							
	SportsData.io	Betfair	Eoddsmaker	Betradar	RunningBall	SportRadar	API football	Football Data
Cruzamentos com sucesso					X			
Cruzamentos com insucesso					X			
Nº de dribles					X		X	X
Tipo de competição	X	X	X	X	X		X	X
Categoria da equipa	X	X	X	X	X		X	X
Sucesso na época	X	X	X	X	X		X	X
Tempo de jogo	X		X	X	X		X	X
Fator casa	X				X		X	X
Nº de cartões					X		X	X
Nº de Faltas					X		X	
Bloqueios					X		X	
Nº de substituições					X		X	
Condições meteorológicas					X			

Posteriormente, foram consideradas características mais gerais, como, em qual suporte é que a fonte fornecia os dados, qual o número de ligas que estavam disponíveis, qual o período de tempo em que os dados eram atualizados, qual a cobertura de desportos que eles forneciam e ainda quais os preços e os diferentes tipos de planos que podiam ser cobrados. Estas informações podem ser consultadas na Tabela 4.

Tabela 4 - Tabela comparativa das features gerais de cada fonte de dados

Fontes de dados	Tipo de suporte	Nº de ligas	Período de atualização dos dados	Nº de desportos	Preço
SportsData.io (Sportsdata, 2020)	JSON e XML	10	3-5 minutos	8	Pago
Betfair (Betfair, 2020)	JSON	+100	1 minuto	7	Gratuito
Eoddsmaker (Eoddsmaker, 2020)	JSON e XML	200	30 segundos-5 minutos	30	Pago
Betradar (Betradar, 2020)	JSON e XML	+30	1 segundo	35	Pago
RunningBall (RunningBall, 2020)	XML e CSV	115	live	15	Pago

Tabela 4 - Tabela comparativa das features gerais de cada fonte de dados (cont.)

Fontes de dados	Tipo de suporte	Nº de ligas	Período de atualização dos dados	Nº de desportos	Preço
SportRadar (Sportsradar, 2020)	JSON	95	Dias	10	Pago
API football (football, 2020)	JSON	425	15 segundos	Futebol	Gratuito até 100 pedidos/dia
Football Data (Data, 2020)	CSV	22	Dias	Futebol	Gratuito

Apesar de a *RunningBall* ser a fonte de dados que mais *features* disponibilizava ao utilizador, depois de uma reunião com a empresa, esta não foi a escolhida, por questões monetárias. Das fontes de dados restantes, as melhores candidatas eram a *Football Api* e a *Football Data*, por disponibilizarem os dados de forma gratuita e por serem as que forneciam mais *features*, incluindo as consideradas imprescindíveis. No entanto, constatou-se que *a Football Api* apenas facultava os dados das épocas mais recentes, ao contrário da *Football Data* que disponibilizava os dados referentes às 20 épocas passadas. Foi então escolhida a *Football Data*, como fonte de dados para o presente trabalho.

5.2. Extração e Armazenamento dos dados

A fase que se sucedeu à escolha da fonte de dados, consistiu na extração dos dados e consequentemente no seu armazenamento. Inicialmente, esta fase foi executada para as duas fontes de dados, *Football API* e *Football Data*.

Relativamente à primeira fonte, foi analisada a documentação da API para compreender a arquitetura da mesma, representada na Figura 30, bem como quais os *requests* a serem executados para extrair a informação. Foi ainda utilizada a ferramenta *PostMan* para que fosse mais simples perceber o tipo de informação que era disponibilizada para cada *request*. De seguida, foi criado um pequeno *script* que realizava a autenticação na *API* e extraía a informação relevante para o problema. Depois de extraída, esta informação foi sendo guardada numa base de dados não relacional, MongoDB.

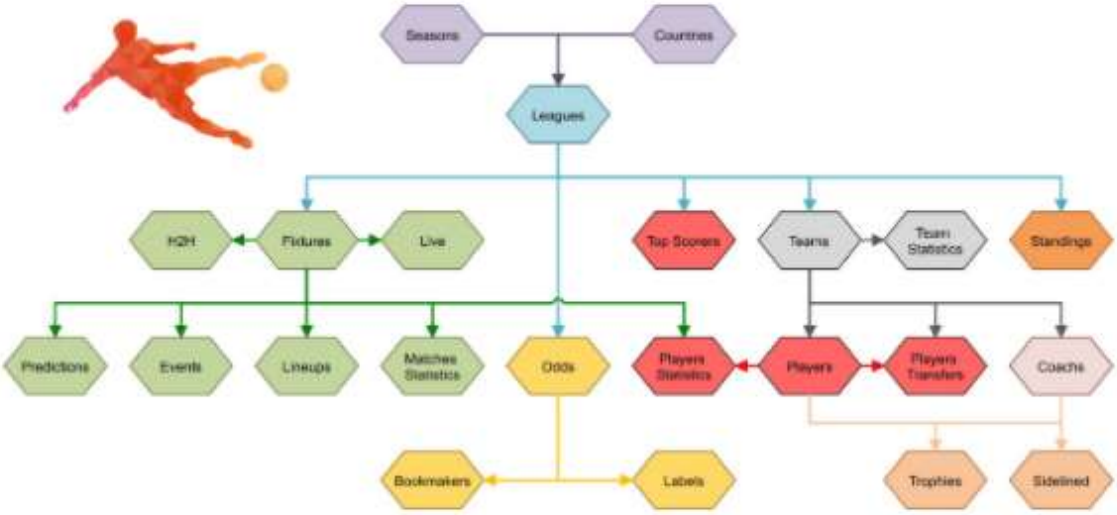


Figura 30 - Arquitetura da Football-API
Fonte: (football, 2020)

A escolha de uma base de dados não relacional deveu-se ao facto de ser de mais simples implementação, uma vez que neste caso eram criados documentos no formato JSON, ao invés de uma estrutura em tabelas com relações entre si, como pode ser observado na Figura 31.

Com o aumento da quantidade de dados, a complexidade das suas relações em bases de dados relacionais e das posteriores *queries* torna-se num problema muito complexo, sendo uma desvantagem em relação às não relacionais (Fernandes H. M., 2020).

O MongoDB torna-se apropriado para grande quantidade de dados e para dados não estruturados, facilitando a sua análise em tempo real.

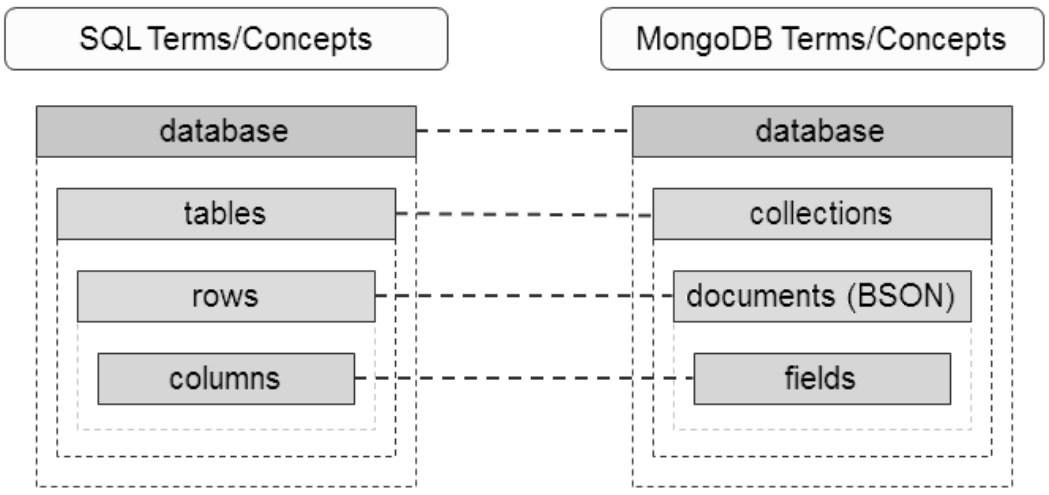


Figura 31 - Bases de dados relacionais vs bases de dados não relacionadas
Fonte: (Guedes, 2017)

No final de serem executados alguns pedidos à API, verificou-se que esta só continha dados estatísticos dos jogos das últimas épocas para as ligas que tinham sido selecionadas para estudar.

Na Figura 32, encontra-se representado as fases executadas até que a informação da *Football API* fosse armazenada na base de dados.

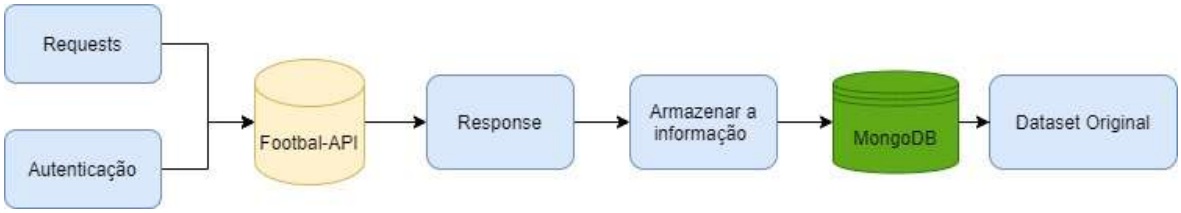


Figura 32 - Diagrama explicativo do processo de extração e armazenamento da Football-API

Decidiu-se então avançar para a segunda melhor opção, a fonte de dados *Football Data*, verificando-se que esta continha os dados desde as épocas de 1993/1994 até à época atual, mas que só a partir da época de 2010/2011 havia informações estatísticas dos jogos. Esta fonte disponibilizava os dados no formato *CSV*, tendo sido desta forma mais fácil a sua extração. Foi necessário construir um pequeno *script* que lesse os ficheiros *CSV* e extraísse os campos relevantes para que essa informação fosse posteriormente armazenada no *MongoDB* e que desta forma pudesse ser construído o *dataset*.

Na Figura 33, encontra-se representado o diagrama explicativo do processo que foi executado, desde a extração de informação da fonte *Football Data*, até ao armazenamento dos dados para que irão constituir o *dataset* original.

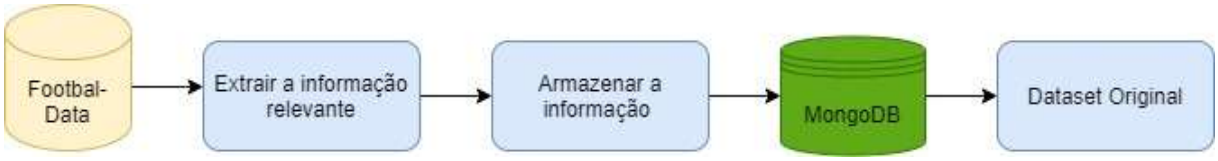


Figura 33 - Diagrama explicativo do processo de extração e armazenamento da Football-Datra

5.3. Identificação das variáveis

Depois de selecionada a fonte *Football Data*, foram recolhidos dados históricos através do *site football-data.co.uk*, o qual disponibiliza informação das principais ligas de futebol. Particularmente, foram considerados os dados das épocas de 2014/2015 até 2019/2020, para a Primeira Liga Inglesa e para a Primeira Liga Portuguesa.

Na Tabela 5, são apresentados os atributos que constituem o *dataset* original, o qual contém os dados fornecidos pela fonte de dados, bem como uma breve descrição sobre o que cada atributo representa. Cada atributo é rotulado como nominal ou numérico, a saber: o atributo é numérico quando a variável assume valores contínuos e quando é possível realizar operações aritméticas com o mesmo; os atributos

nominais ou categóricos não possuem nenhuma ordem entre si, sendo normalmente associados a categorias que fornecem informação qualitativa.

Tabela 5 - Tabela descritiva dos atributos que constituem o dataset original

Atributo	Tipo	Descrição do atributo
HT	Nominal	Nome da equipa visitada
AT	Nominal	Nome da equipa visitante
FTR	Nominal	Resultado do jogo
HS	Numérico	Número de remates da equipa visitada
AS	Numérico	Número de remates da equipa visitante
HST	Numérico	Número de remates à baliza da equipa visitada
AST	Numérico	Número de remates à baliza da equipa visitante
FTHG	Numérico	Golos marcados pela equipa visitada no final do jogo
FTAG	Numérico	Golos marcados pela equipa visitante no final do jogo
HF	Numérico	Número de faltas cometidas pela equipa visitada
AF	Numérico	Número de faltas cometidas pela equipa visitante
HC	Numérico	Número de cantos marcados pela equipa visitada
AC	Numérico	Número de cantos marcados pela equipa visitante
HY	Numérico	Número de cartões amarelos mostrados à equipa visitada
AY	Numérico	Número de cartões amarelos mostrados à equipa visitante
HR	Numérico	Número de cartões vermelhos mostrados à equipa visitada
AR	Numérico	Número de cartões vermelhos mostrados à equipa visitante
B365H	Numérico	*Odd* da *Bet365* para a equipa que está a jogar em casa
B365D	Numérico	*Odd* da *Bet365* para ocorrer o empate
B365A	Numérico	*Odd* da *Bet365* para a equipa que está a jogar fora
IWA	Numérico	*Odd* da *Interwetten* para a equipa visitante
IWH	Numérico	*Odd* da *Interwetten* para a equipa visitada
IWD	Numérico	*Odd* da *Interwetten* para o empate
WHA	Numérico	*Odd* da *William Hill* para a equipa visitante
WHD	Numérico	*Odd* da *William Hill* para o empate
WHH	Numérico	*Odd* da *William Hill* para a equipa visitada
M2_5	Numérico	*Odd* para haver mais de 2,5 golos no jogo
L2_5	Numérico	*Odd* para haver menos de 2,5 golos no jogo

5.4. Análise Univariada

A análise univariada revela-se uma tarefa de enorme importância, uma vez que é o primeiro contacto com os dados, permitindo explorar as variáveis e perceber se existem dados que necessitam de ser corrigidos ou transformados. Neste tipo de análise, cada variável é tratada isoladamente antes de ser cruzada com as outras.

Na Tabela 6, é realizada uma análise com métricas de tendência central e de dispersão para as diferentes *features* nominais que constituem o *dataset*. São calculados os valores médios, os valores máximos e mínimos, a variância e o desvio padrão de cada atributo.

Tabela 6 - Análise univariada com features de entrada

Atributo	Média	Mínimo	Máximo	Variância	Desvio padrão
HS	13,99	0	43	32,14	5,67
AS	11,26	0	30	23,20	4,82
HST	4,69	0	17	6,94	2.63
AST	3.85	0	15	5.22	2.28
FTHG	1.53	0	8	1.66	1.29
FTAG	1.18	0	9	1.36	1.17
HF	10.54	0	24	11.74	3.43
AF	11.06	1	26	12.54	3.54
HC	5.81	0	19	9.78	3.13
AC	4.72	0	16	7.24	2.69
HY	1.57	0	7	1.52	1.23
AY	1.75	0	9	1.60	1.27
HR	0.06	0	2	0.06	0.24
AR	0.08	0	2	0.07	0.27
B365H	2.95	1.06	23	5.21	2.28
B365D	4.24	3	17	2.09	1.44
B365A	5.04	1.12	41	20.57	4.54
IWA	NAN	1.12	30	NAN	NAN
IWH	NAN	1.07	20.0	NAN	NAN
IWD	NAN	2.95	12	NAN	NAN
WHA	4.91	1.12	46	19.88	4.46
WHD	3.96	2.80	15	1.65	1.28

Tabela *6* - Análise univariada com features de entrada *(cont.)*

Atributo	Média	Mínimo	Máximo	Variância	Desvio padrão
WHH	2.91	1.05	21	4.73	2.17
M2_5	1.88	1.21	2.91	0.08	0.29
L2_5	2.03	1.40	4.45	0.16	0.40

Desta análise foi possível extrair informação importante para ajudar a realizar a analise exploratória dos dados. Tendo sido verificado algumas relações como é o caso de que são realizados em média aproximadamente 13 remates, sendo que apenas aproximadamente 4 são realizados à baliza e apenas acorre em média 1 golo, sendo estes valores para cada equipa.

5.4.1. Análise exploratória dos dados

A análise exploratória dos dados (EDA, do ingles *Exploratory Data Analysis*), é uma abordagem que existe para analisar o conjunto de dados, predominantemente através de técnicas gráficas para a melhor perceção dos mesmos, permitindo encontrar as características principais, as relações ocultas e detetar os *outliers*. Esta análise foi realizada para o *dataset* original.

Casa vs visitante vs empate

A primeira análise realizada foi a distribuição do *target* resultado final ao longo das diferentes épocas do *dataset*. As siglas H, A e D correspondem respetivamente à vitória da equipa da casa, à vitória da equipa visitante e ao empate. Os valores apresentados na Figura *34*, correspondem às percentagens das diferentes classes do *target* ao longo das épocas de constituem o *dataset* original.

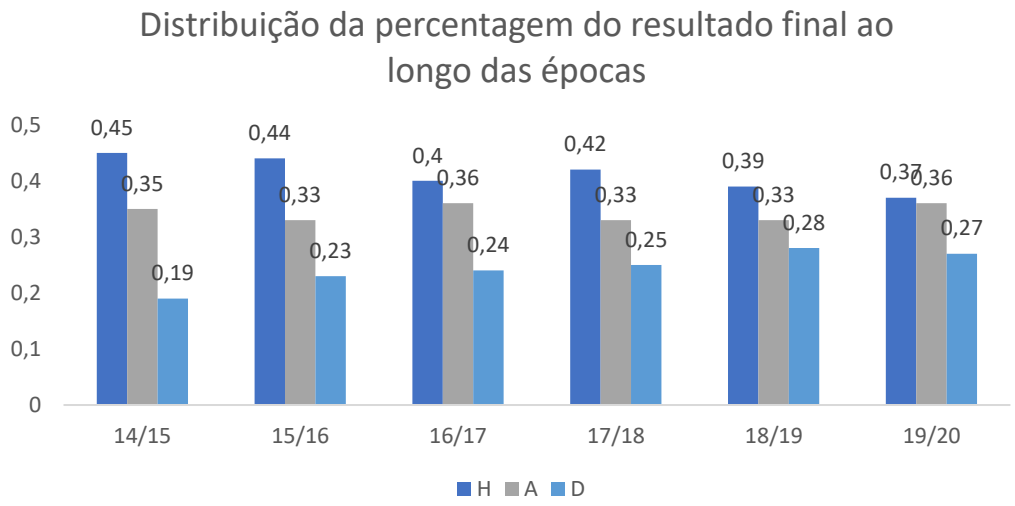


Figura 34 - *Distribuição do target FTR ao longo das épocas do dataset.*

É possível verificar que a percentagem de empates e vitórias da equipa visitante é sempre inferior à percentagem de vitórias da equipa que joga em casa. Também é possível concluir que ao longo dos anos as percentagens das classes têm vindo a ficar cada vez mais semelhantes, não se salientando tanto a diferença entre a equipa estar a jogar em casa ou no campo do adversário.

Golos marcados/ Média de golos marcados

De seguida, foi realizada para a época de 19/20 a comparação da distribuição do número de golos para a equipa visitante (A) e para a visitada (H) em cada jogo.

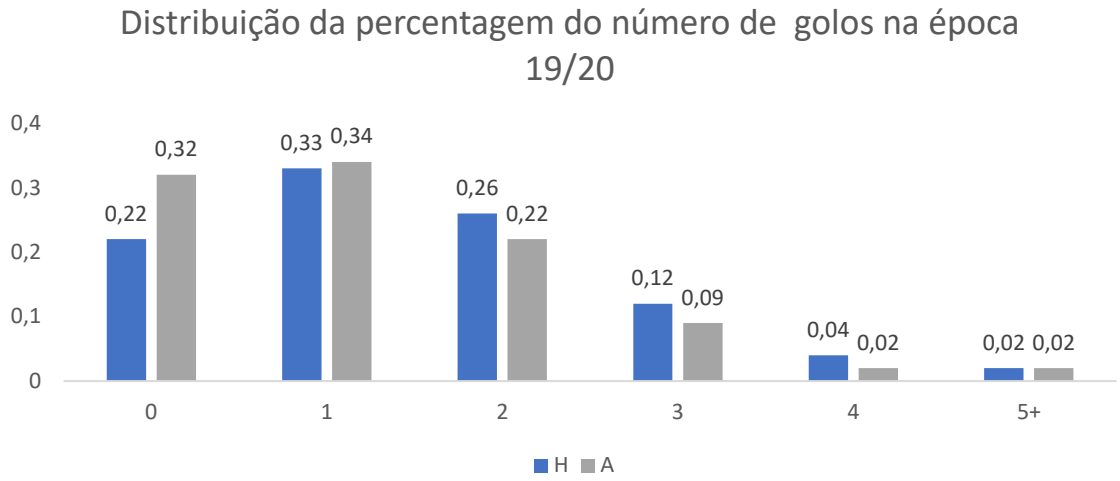


Figura 35 - Distribuição do número de golos ao longo dos jogos da época 19/20.

Na Figura 35, é possível verificar que a equipa que joga em casa é a que mais golos marca, sendo a que mais vezes marca mais do que 1 golo. Existe uma discrepância de 576 golos para a equipa da casa contra 458 golos da equipa visitante. Observa-se também que a existência de menos do que 2 golos num jogo, acontece em cerca de 50 % dos jogos, tanto para a equipa que joga fora como para a que joga em casa. Foram ainda comparados os valores médios de golos que são marcados pelas equipas visitantes e visitadas ao longo das épocas que constituem o dataset original.

Na Figura 36, é possível conferir que ao longo das épocas a equipa que joga em casa obtém sempre valores médios de golos mais altos do que a equipa que joga fora. É de salientar ainda que, em média, por jogo são marcados entre 1 à 2 golos.

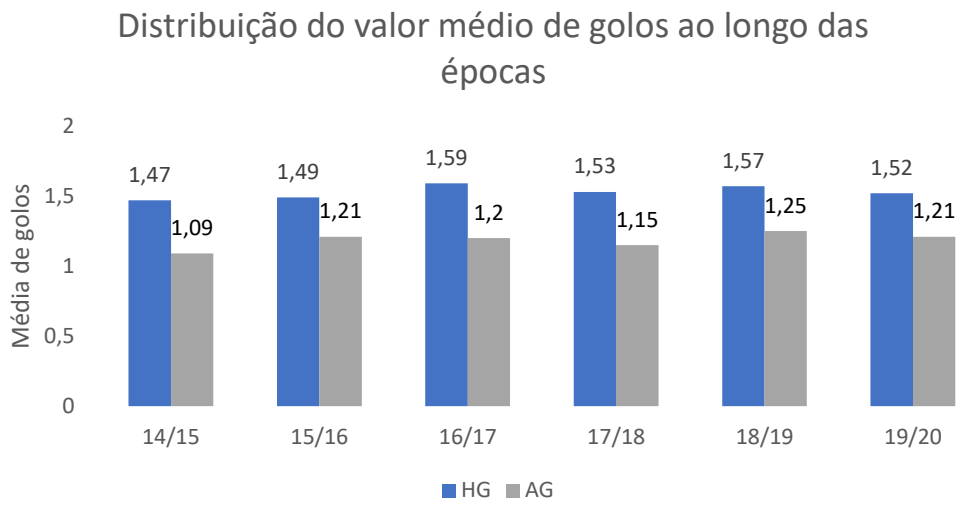


Figura 36 - Distribuição do número médio de golos ao longo das épocas.

Média de cantos marcados

A terceira análise foi efetuada ao número médio de cantos que eram assinalados durante cada jogo nas épocas do conjunto de dados, comparando a equipa da casa com a equipa que joga fora.

Na Figura 37, é possível verificar mais uma vez que o número de cantos é superior para a equipa que joga em casa, marcando esta aproximadamente mais um canto do que a equipa que joga fora. Constata-se ainda que, em média, são assinalados entre 5 à 6 cantos por jogo.

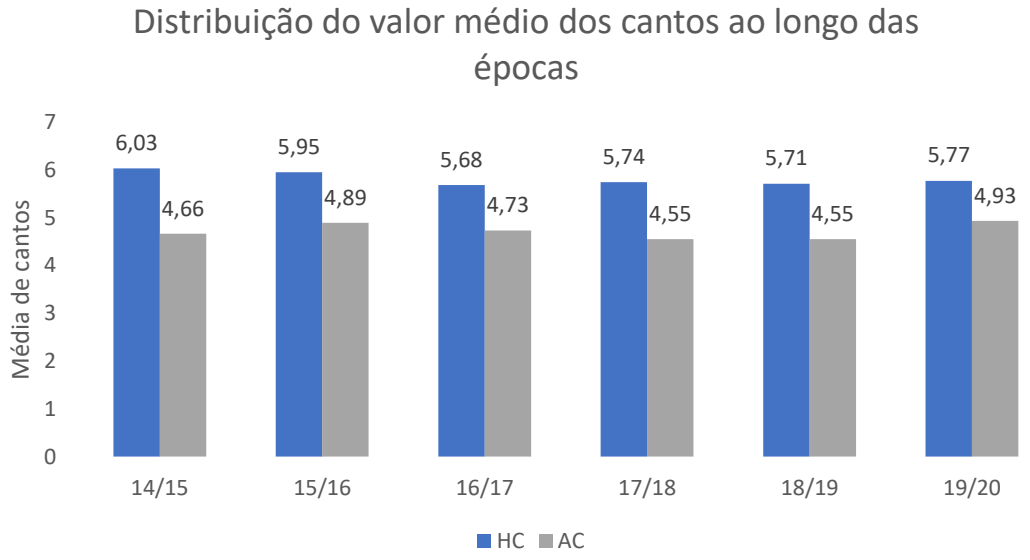


Figura 37 - Distribuição do número médio de cantos ao longo das épocas.

5.5. Análise Multivariada

De forma a perceber quais as relações existentes entre as diferentes variáveis que constituem o *dataset*, foram aplicados alguns métodos estatísticos para determinar as correlações que estas apresentam. Para tal, foram utilizados gráficos de dispersão e a matriz de correlação, que nos indica a correlação de *Pearson* entre os atributos, facilitando desta forma a interpretação das diversas relações entre os mesmos. Esta matriz pode ser observada na Figura 38.

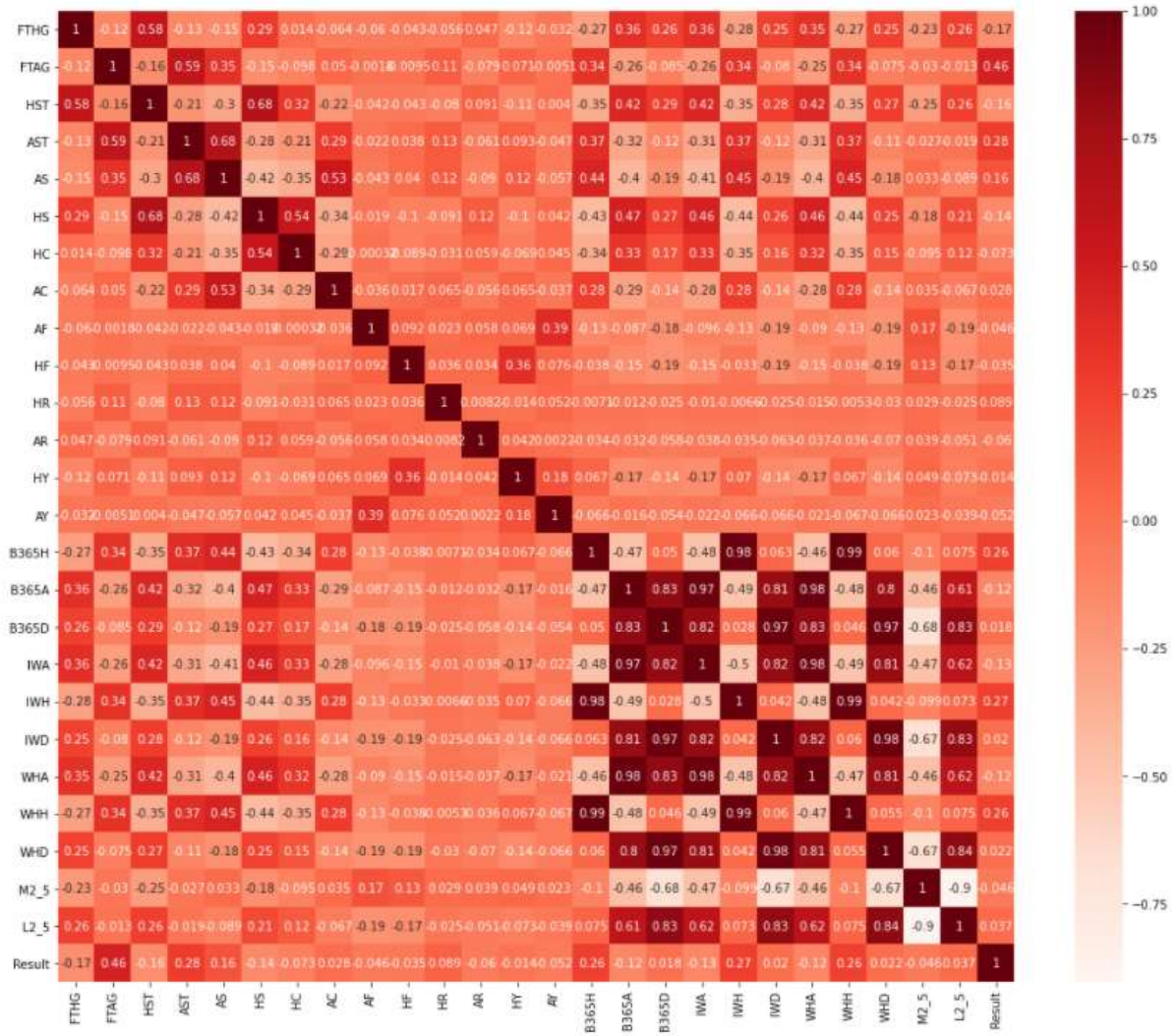


Figura 38 - Matriz de correlação do dataset

Da análise efetuada, verificou-se, por exemplo, que as *features* que se demonstravam mais relevantes para prever o *target Result* eram: as *Odds* para as diferentes classes, o número de remates, o número de remates realizados à baliza e o número de golos concedidos no jogo. Os gráficos de dispersão dos atributos que constituem o *dataset* podem ser consultados na secção de Anexos no anexo A.

5.6. Limpeza dos dados

Outra tarefa importante a realizar numa primeira fase do pré-processamento de dados, é verificar se o *dataset* construído possui dados: em falta, incorretos, incompletos ou até mesmo se existem *outliers* (dados que diferem drasticamente dos restantes).

Para corrigir estes valores, podem ser adotadas algumas abordagens, sendo que a mais simples, consiste em selecionar as instâncias que contêm dados desse tipo e descartá-las do conjunto de dados. No entanto, isto só é possível quando a informação a ser libertada não é relevante e o conjunto é constituído por uma grande quantidade de dados. Outra opção, é esses valores serem substituídos por outros, com base nos que foram observados nesse atributo. Normalmente, são substituídos pelo valor médio, valor mais frequente, mas também é possível utilizar um modelo preditivo para estimar esse valor.

De modo a verificar se existiam *outliers* entre os dados do *dataset* foram utilizados gráficos de *box-plot* que nos permitiram visualizar a sua existência e que podem ser consultados no anexo B da secção de anexos. Da análise destes *box-plots* foi possível verificar que os dados não se encontravam bem distribuídos, levando a pensar que existiam muitos *outliers*. No entanto, devido à natureza do problema, percebeu-se que os dados eram provenientes de eventos raros que ocorrem nos jogos e que fogem à normalidade tal como, a ocorrência de 8 golos num jogo.

Para verificar a existência de dados em falta foi utilizado o operador *isnull* da biblioteca Pandas, que nos demonstrou a existência de dados em falta referentes aos atributos das *odds* (IWH, IWD e IWA). Uma vez que existiam dados de outras casas de apostas para o mesmo jogo, optou-se por retirar esses atributos do *dataset*.

5.7. Transformação dos dados

Devido à existência de classificadores que unicamente aceitam dados numéricos, como é o caso do algoritmo KNN, surgiu a necessidade de converter as características que estivessem num outro tipo de dados. As *features* HT, AT e FTR encontravam-se nesta situação, ou seja, foi necessário transformar os valores desses atributos em valores numéricos. Para resolver esta situação foi construído um dicionário em que era atribuído ao nome de cada equipa um código diferente, para o caso dos atributos HT e AT. No atributo FTR, foi realizada uma conversão semelhante, ou seja, o valor "*Home*" foi substituído pelo valor 1, o valor "*Away*" foi substituído pelo valor 2 e o "*Draw*" foi substituído pelo valor 0.

Outra transformação que foi necessário realizar foi a normalização dos dados, devido a existirem dados de grandezas diferentes. Para que os dados fossem convertidos para uma escala comum, sem que perdessem as diferenças que existiam, foi realizada a alteração dos atributos utilizando a fórmula 10. Para cada exemplo foi

calculada a diferença entre o valor e a média do atributo, sendo esta dividida pelo desvio padrão do atributo.

$$Xs = \frac{X - mean}{std} \tag{10}$$

5.8. Seleção de variáveis

Depois de verificada a ligação que as variáveis apresentavam em relação aos *targets*, e de se concluir que não existiam atributos que estivessem bastante correlacionados com os saídas a serem previstas, optou-se por implementar novos, de forma a que fossem criadas novas variáveis. Essas atributos foram adicionados na tentativa de tornar mais evidente os possíveis relacionamentos entre os atributos existentes e os targets, e, desta forma facilitar a tarefa de classificação ao classificador. Os novos atributos implementados que deram origem ao *dataset* provisório, estão descritos na Tabela 7. Foram também transformadas as variáveis que só eram possíveis de obter no decorrer do jogo, sendo estas utilizadas como valores acumulados ao longo da época e valores médios. Por fim, removeram-se as variáveis que não apresentavam informações relevantes para o problema em questão, nem para nenhum dos targets a prever, como era o caso dos cartões (amarelos, vermelhos) e das faltas.

Tabela 7 - Descrição das novas features implementadas.

Atributo	Tipo	Descrição do atributo
Mw	Numérico	Jornada do jogo
HTGS	Numérico	Número de golos marcados pela equipa visitada na época
ATGS	Numérico	Número de golos marcados pela equipa visitante na época
HTGC	Numérico	Número de golos sofridos pela equipa visitada na época
ATGC	Numérico	Número de golos sofridos pela equipa visitante na época
ATST	Numérico	Número de remates à baliza na época pela equipa visitante
HTST	Numérico	Número de remates à baliza na época pela equipa visitada
ATC	Numérico	Número de cantos na época marcados pela equipa visitante
HTC	Numérico	Número de cantos na época marcados pela equipa visitada
ATS	Numérico	Número de remates na época da equipa visitante
HTS	Numérico	Número de remates na época da equipa visitada
ATF	Numérico	Número de faltas na época da equipa visitante
HTF	Numérico	Número de faltas na época da equipa visitada
Avg_HGS	Numérico	Média de golos marcados pela equipa visitada
Avg_AGS	Numérico	Média de golos marcados pela equipa visitante

Tabela 7 - Descrição das novas features implementadas. (cont.)

Atributo	Tipo	Descrição do atributo
HAS	Numérico	Força de ataque da equipa visitada
AAS	Numérico	Força de ataque da equipa visitante
HDS	Numérico	Força defensiva da equipa visitada
ADS	Numérico	Força defensiva da equipa visitante
HTP	Numérico	Pontos da equipa visitada
ATP	Numérico	Pontos da equipa visitante
ATWinStreak3	Binominal	Assume o valor “1” se a equipa visitante tiver vencido os últimos 3 jogos, caso contrário assume o valor “0”
HTWinStreak5	Binominal	Assume o valor “1” se a equipa visitada tiver vencido os últimos 5 jogos, caso contrário é “0”
ATWinStreak5	Binominal	Assume o valor “1” se a equipa visitante tiver vencido os últimos 5 jogos, caso contrário é “0”
HTLossStreak3	Binominal	Assume o valor “1” se a equipa visitada tiver perdido os últimos 3 jogos, caso contrário assume o valor “0”
ATLossStreak3	Binominal	Assume o valor “1” se a equipa visitante tiver perdido os últimos 3 jogos, caso contrário assume o valor “0”
HTLossStreak5	Binominal	Assume o valor “1” se a equipa visitada tiver perdido os últimos 5 jogos, caso contrário é “0”
ATLossStreak5	Binominal	Assume o valor “1” se a equipa visitante tiver perdido os últimos 5 jogos, caso contrário é “0”
M2_5_golsAT	Binominal	Assume o valor “1” se a equipa visitante tiver marcado mais do que 2 golos no jogo, caso contrário assume o valor “0”
M2_5_golsHT	Binominal	Assume o valor “1” se a equipa visitado tiver marcado mais do que 2 golos no jogo, caso contrário assume o valor “0”
HTGD	Numérico	Diferença entre os golos sofridos e marcados pela equipa visitada.
ATGD	Numérico	Diferença entre os golos sofridos e marcados pela equipa visitante
GoalsDiff	Numérico	Diferença de golos marcados aos longo da época pelas duas equipas
DiffPts	Numérico	Diferenças de pontos obtidos pelas duas equipas

Tabela 7 - Descrição das novas features implementadas (cont.)

Atributo	Tipo	Descrição do atributo
Last_games_away	Numérico	Pontos obtidos nos confrontos diretos anteriores com a mesma equipa pela equipa que joga fora
Last_games_home	Numérico	Pontos obtidos nos confrontos diretos anteriores com a mesma equipa pela equipa que joga em casa

5.9. Construção dos Datasets

Nesta secção será apresentada a construção dos vários *datasets* a serem utilizados na previsão dos eventos de futebol. Por questões de tempo, apenas foram selecionados três eventos: resultado final do jogo (vitória, derrota ou empate), número de golos e número de cantos marcados por ambas as equipas. Para cada uma das experiências (correspondentes aos eventos referidos), são especificadas as *features* que se consideraram, o *target* que se pretendia prever e é apresentada a distribuição das classes, de forma a perceber o seu balanceamento.

5.9.1. 1ª Experiência – Previsão "resultado final do jogo"

O conjunto de dados do dataset provisório não se encontrava adequado ao problema. Como primeira abordagem, foram utilizadas as variáveis criadas a partir dos dados fornecidos e que apresentavam uma maior correlação com o resultado que se pretendia prever.

O *target* que se pretendia prever, era um atributo nominal que representava o resultado que um jogo de futebol podia assumir no final do confronto, sendo descrito como o "resultado final do jogo", do inglês, *Full Time Result* (FTR).

Este atributo pode assumir 3 valores: *Home* quando a equipa que está a jogar em casa ganha o jogo, *Away* se a equipa que está a defrontar o jogo no campo do adversário *ganha o confronto* e *Draw* se o jogo entre as duas equipas acaba num empate.

As *features* utilizadas para realizar a previsão deste problema podem ser consultadas na Tabela 8.

De forma a perceber como as classes se encontravam distribuídas no *dataset*, foi construído o gráfico ilustrado na Figura 39. Verificou-se que 45,70% dos jogos correspondiam à taxa de vitória para a equipa visitada, em oposição, 30,35% dos confrontos acabavam em vitória para a equipa visitante e os restantes 23,95% terminavam com um empate. Desta forma, concluiu-se que as classes não se encontram balanceadas, uma vez que existiam aproximadamente o dobro de exemplos da classe "Home", em relação à classe "Draw".

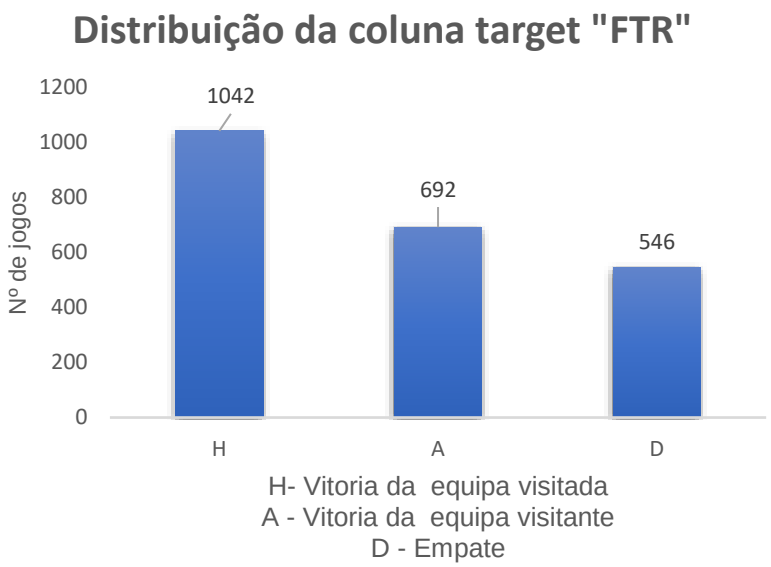


Figura 39 - Distribuição da coluna target "FTR"

Isto demonstra que se o classificador considerar a hipótese de prever o resultado futuro sempre com a classe maioritária, ou seja, Vitória (equipa de casa), irá acertar na maioria dos casos mesmo não estando a realizar a classificação das restantes classes.

Existem duas técnicas de amostragem que pretendem solucionar este problema. A primeira técnica realiza uma subamostragem (*Undersampling*) do conjunto de dados, a qual consiste em excluir alguns dos dados da classe maioritária do conjunto de treino.

A segunda técnica realiza uma super-amostragem (*Oversampling*) em que são criadas cópias dos dados da classe minoritária para que ambas as classes possam apresentar a mesma quantidade no conjunto de treino. A forma como esta divisão é realizada encontra-se descrita na Figura 40.

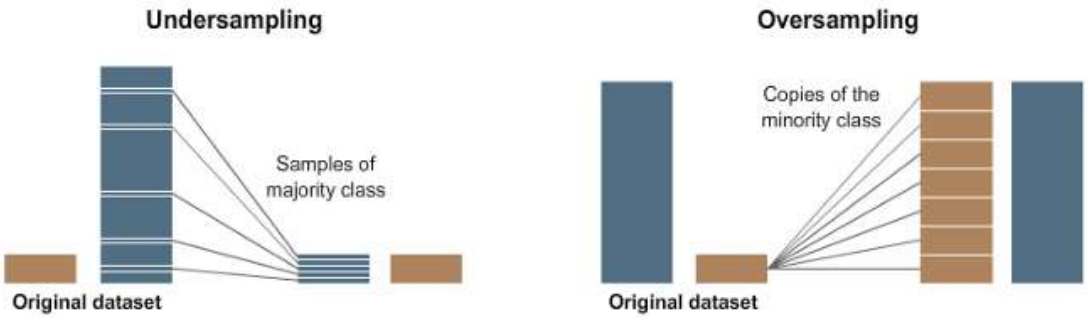


Figura 40 - Técnicas de amostragem de dados
Fonte: (Vaz, 2019)

No caso em questão, optou-se por realizar uma subamostragem, de forma a que o conjunto de dados de treino passasse a ter a mesma quantidade que a classe minoritária, nas três classes.

5.9.2. 2ª Experiência – Previsão de "mais/menos 2.5 golos"

Numa segunda abordagem, pretendia-se prever se, na totalidade, seriam marcados por ambas as equipas, mais do que 2 golos no final do jogo.

O *target* que se pretendia prever neste caso, era um atributo binominal, que representava o número de golos totais que iriam ocorrer até ao final do jogo de futebol.

Esse atributo pode assumir dois valores: *1* - quando no final do confronto foram marcados mais do que um total de dois golos por ambas as equipas; *0* - quando a condição não se verificou, ou seja, quando ocorreram menos do que dois golos.

Os atributos que foram utilizados neste *dataset* encontram-se apresentados na Tabela 8.

Para verificar se as classes se encontravam igualmente distribuídas, foi construído o gráfico representado na Figura 41, que nos demonstra que as classes se encontravam balanceadas, uma vez que existiam 52,06% da classe "1" e 47,93% da classe "0".

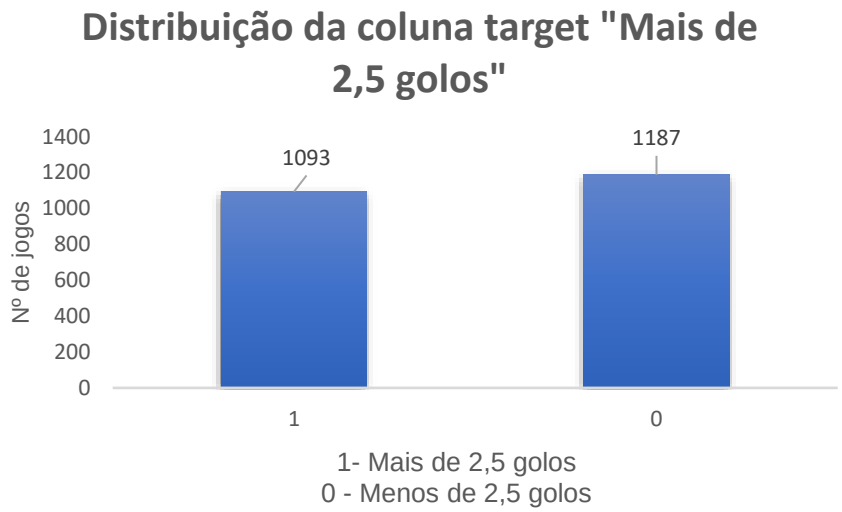


Figura 41 – Distribuição da coluna target" M2_5", ao longo dos jogos.

5.9.3. 3ª Experiência – Previsão de "mais/menos 10.5 cantos"

Numa terceira abordagem pretendia-se prever se seriam marcados, no total, por ambas as equipas, mais do que 8/10 cantos no final do jogo.

O *target* que se pretendia prever era um atributo binominal que representava o número de cantos totais que ocorreram até ao final do jogo de futebol.

Este atributo pode assumir 2 valores: *1* - quando no final do confronto foram marcados, por ambas as equipas, um total de mais do que oito/dez cantos; *0* - quando não ocorreram pelo menos oito/dez cantos.

Os atributos utilizados no conjunto de dados que permitiram realizar a previsão podem ser consultados na Tabela 8.

Para as classes “mais de 8 cantos” e “menos de 8 cantos”, foi também verificada a proporção destas no *dataset*. Averiguou-se que quando o target era ”mais de 8 cantos”, as classes se encontravam desbalanceadas, uma vez que existia uma maior quantidade de dados da classe “0”. A Figura 42, mostra que em mais de 72% dos jogos, existem mais de 8 cantos, evidenciando esse desequilíbrio.

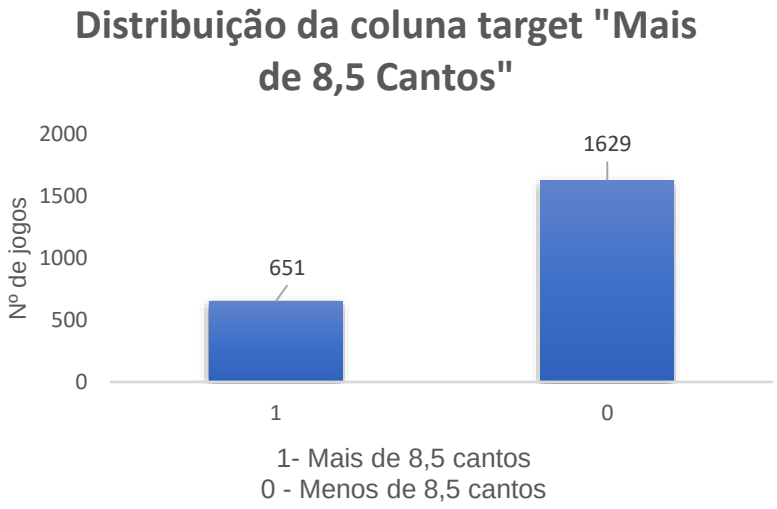


Figura 42 - Distribuição do target mais de 8,5 cantos ao longo dos jogos

Foi então testado o aumento do número de cantos para 10, uma vez que o número final é o total de cantos cometidos pelas duas equipas e, como se verificou na análise exploratória de dados, são marcados, em média, 5 cantos por cada equipa em cada jogo. A distribuição das classes para o caso de “mais de 10 cantos” encontra-se representada na Figura 43. Para este caso, constata-se que as classes se encontram aproximadamente balanceadas, com cerca de 50% cada uma, tendo sido assim resolvido o problema existente.

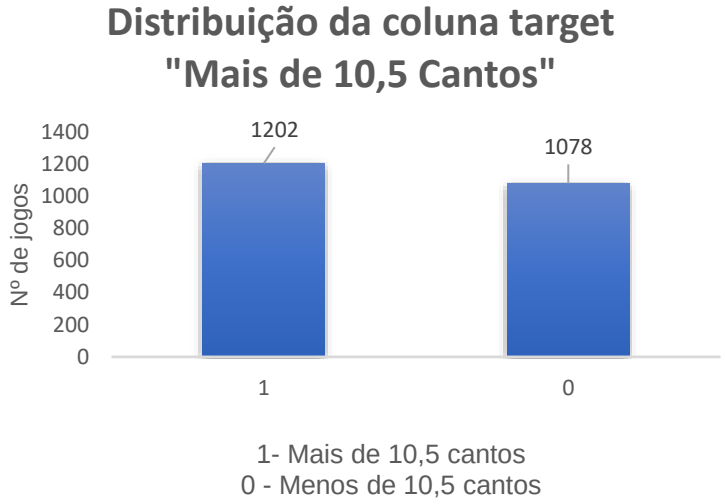


Figura 43- Distribuição do target mais de 10,5 cantos ao longo dos jogos

Na Tabela 8, encontram-se apresentadas as *features* que foram utilizadas em cada experiência.

Tabela 8 - Tabela com as features utilizadas para cada experiência

Features	Experiência 1	Experiência 2	Experiência 3
HT	X	X	X
AT	X	X	X
Odds Resultado	X	X	X
Past AST	X	X	
Past HST	X	X	
Past HS	X	X	
Past AS	X	X	
HAS	X		X
HDS	X		X
ADS	X		X
AAS	X		X
HTFormPts	X		
ATFormPts	X		
MW	X	X	X
Last_Games_Away	X	X	
Last_Games_Home	X	X	
HTGS	X	X	
ATGS	X	X	
HTGC	X	X	
ATGC	X	X	
Odds >2.5 goals		X	
HTP		X	
ATP		X	
GoalsDiff		X	
HTGD		X	
ATGD		X	
Goals_mean_away		X	
Goals_mean_home		X	
SotDiff		X	X
Past HC			X
Past AC			X

Tabela 8 - Tabela com as features utilizadas para cada experiência (cont.)

Features	Experiência 1	Experiência 2	Experiência 3
Mean_AC			X
Mean_HC			X
CornersDiff			X
ShotsDiff			X
DiffPts			X

CAPÍTULO 6

MODELAÇÃO E DISCUSSÃO DE RESULTADOS

Este capítulo apresenta o processo de criação dos modelos de *Machine Learning* para a previsão dos eventos de futebol anteriormente identificados, bem como a avaliação do seu desempenho. O critério para a escolha dos modelos, baseou-se nos mais utilizados em trabalhos científicos realizados na área, como os analisados no Capítulo 3. Para cada um dos eventos, os modelos são testados com os conjuntos de dados criados para o efeito, sendo os resultados analisados e discutidos.

Para a implementação, foram usadas as bibliotecas *Scikit-learn* e *Tensorflow*, introduzidas no Capítulo 4.

De referir que ao longo deste capítulo os termos ingleses "*Home*", "*Draw*" e "*Away*" correspondem à vitoria da equipa visitada, ao empate e à vitoria da equipa visitante, respetivamente.

6.1. Fluxo de trabalho

O esquema representado na Figura 44, exemplifica o fluxo de trabalho seguido para realizar a previsão dos resultados.

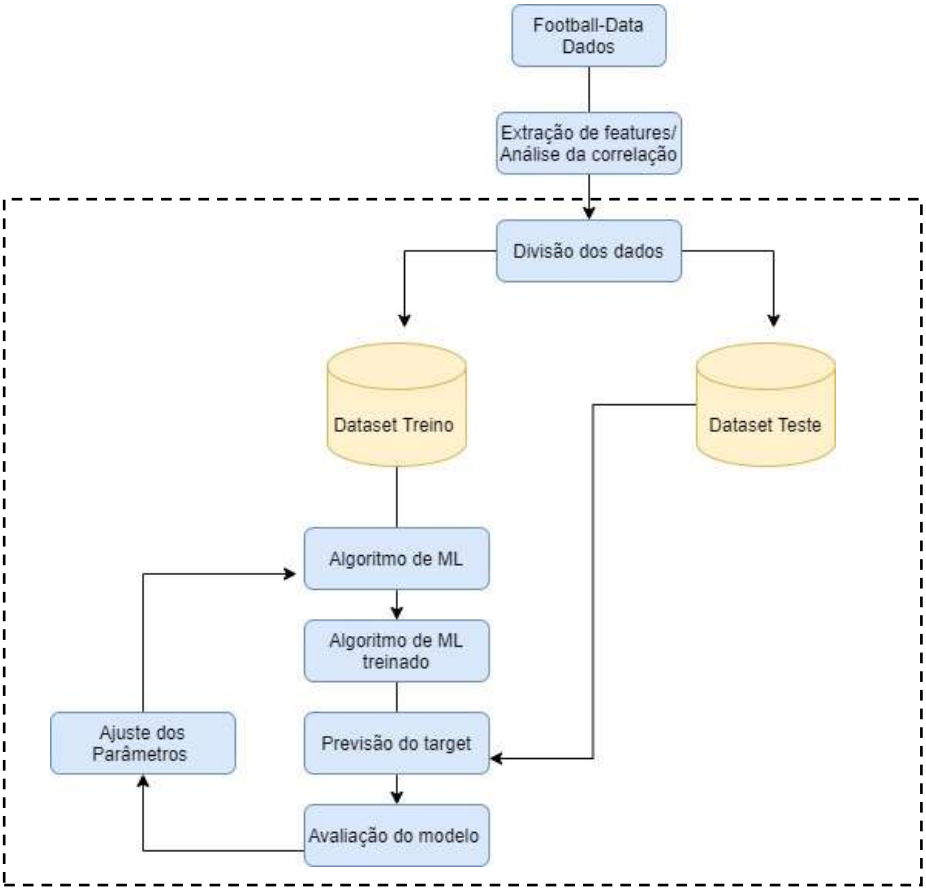


Figura 44 – Fluxograma utilizado para a previsão do resultado

Este esquema foi utilizado nos três *datasets* criados e descritos no capítulo anterior: o primeiro *dataset* será usado para a previsão do resultado final do jogo com os dados da liga inglesa, o segundo conjunto para prever se haverá mais do que 2 golos, o terceiro para a previsão da existência de mais do que 8/10 cantos e, por último, o *dataset* igual ao primeiro, mas desta vez com dados da liga portuguesa. A estes *datasets* foram aplicados os algoritmos de *Machine Learning* que se encontram descritos na secção 2.4.1.

6.2. Divisão em dados de treino / dados de teste

Os conjuntos de dados foram construídos com os dados das épocas de 2014/2015 à de 2019/2020 da Liga Inglesa, tendo sido também utilizados os dados da Primeira Liga Portuguesa para as mesmas épocas. Os dados da primeira jornada de cada época não foram considerados, uma vez que os atributos nessa jornada não possuíam valores por falta de histórico.

Para avaliar o desempenho dos modelos é necessário realizar uma divisão dos dados recolhidos, para que seja possível validar a forma como estes se comportam quando confrontados com novos conjuntos de dados. O *dataset* é dividido em duas partes que são, o conjunto de treino e o conjunto de teste. No primeiro conjunto, os dados são utilizados para a construção do modelo, enquanto que o segundo conjunto servirá para avaliar o modelo em instâncias diferentes das que foram usados no treino.

A divisão dos dados pode ser realizada utilizando várias abordagens, algumas das quais se encontram descritas na secção 2.5. No caso do problema em questão, o fator tempo é importante, pois os dados têm uma sequência temporal. Desta forma, a abordagem de validação escolhida foi o *Holdout*, sendo que, cada um dos *datasets,* foi dividido em aproximadamente 70% dos dados para treino e 30% para teste. Esta divisão encontra-se exemplificada na Figura 45.

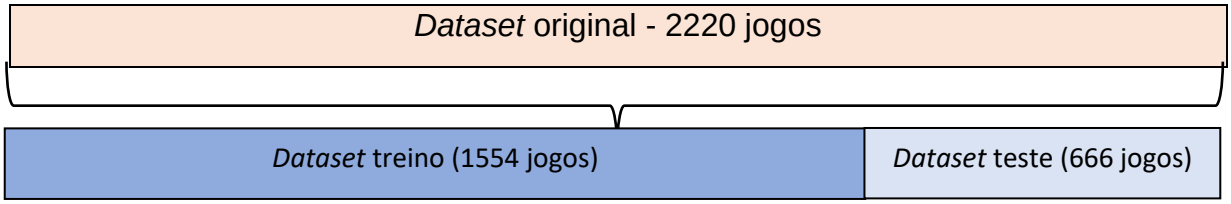


Figura 45 - Esquema da divisão dos datasets treino e teste

6.3. 1º Experiência – Previsão do "resultado final do jogo" para a Liga Inglesa

Na primeira experiência foram aplicados os diversos algoritmos de *Machine Learning* ao primeiro *dataset*, com o objetivo de realizar a previsão do resultado final da Primeira Liga Inglesa.

Nesta secção, são apresentados os hiperparâmetros que obtiveram o melhor desempenho para cada um dos classificadores. De forma, a encontrar os melhores parâmetros para cada um dos modelos, foi utilizada a técnica de *Grid Search* que executou o teste com as várias combinações possíveis e no final apresentou os parâmetros que melhor realizaram a classificação. O desempenho é demonstrado pelas matrizes de confusão, apresentadas para cada um dos classificadores, sendo assim possível analisar a forma como cada um se comportou na previsão de cada classe.

6.3.1. Redes Neuronais

O classificador de Redes Neuronais, que foi o primeiro a ser construído, apresentava muitos parâmetros a serem configurados, tais como o número de camadas, o número de neurónios e a função de ativação. Foi primeiramente testado um modelo de redes neuronais simples, mas os resultados não foram tão satisfatórios optando-se pelo *Deep Learning*.

Os melhores resultados foram obtidos com o modelo a dispor de 4 camadas. A primeira camada correspondente à camada de entrada, composta pelo número de neurónios igual ao número de atributos, duas camadas intermédias compostas por 50 neurónios cada uma e com a função de ativação linear retificada (ReLu), e a camada de saída com a função de ativação *Softmax* e com 3 unidades de saída correspondendo a cada uma das classes.

O modelo utilizou os atributos das *odds* como os de maior peso para realizar a previsão. Na Tabela 9 é possível observar a matriz de confusão obtida quando o algoritmo de *Deep Learning* foi submetido ao conjunto de teste.

Tabela 9 - Matriz de confusão do modelo de redes neuronais

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	263	103	123	53,78
Previsto A	34	80	35	53,69
Previsto D	2	9	3	21,43
Recall (%)	87,96	41,67	1,86	

Da matriz de confusão contruída para este classificador, é possível verificar que este tem uma taxa de acerto na classe "*Home*" superior à das restantes, sendo que isto se deve ao fato de esta classe ter uma maior presença no conjunto de dados, constituindo cerca de 40% dos mesmos. A classe "*Draw*" foi a que o classificador teve mais dificuldade em classificar, tendo acertado somente em 1,86% das classificações.

6.3.2. Decision Tree

De seguida, foi criado um modelo baseado em árvores de decisão para realizar a previsão do resultado. Os parâmetros escolhidos foram construir uma árvore com o máximo de profundidade de 10 ramificações e com um critério de construção de "*gini*". O modelo criou uma árvore com uma profundidade de 4 e com o atributo de número de golos marcados pela equipa da casa a ser utilizado como nó principal. Para realizar as tomadas de decisões, este utilizou como atributos mais importantes: os relativos às *odds* do resultado final, o valor da diferença entre os remates à baliza e o valor médio de golos, ao longo da época, para cada equipa.

Para entender como o modelo se comportou na classificação de cada classe, foi construída a tabela de confusão, apresentada na Tabela 10.

Tabela 10 - Matriz de confusão para a modelo Decison Tree

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	185	38	54	66,79
Previsto A	110	162	99	43,67
Previsto D	1	0	1	50,00
Recall (%)	62,50	81,00	0,65	

Da análise à tabela, é possível verificar que este modelo teve uma taxa de acerto na classe" *Away*" superior à das restantes, tendo também um menor valor de *precision*, o que nos indica que existe uma grande quantidade de instâncias que foram mal classificadas nesta classe.

Também se verifica que este modelo meramente classificou duas instâncias como sendo da classe "*Draw*", indicando que tem um mau desempenho na classificação desta classe.

6.3.3. Random Forest

De seguida, foi construído o classificador *Random Forest*, com uma função para medir a qualidade da divisão baseada no critério "*Gini*", que realiza a divisão tendo em conta a diminuição da impureza na divisão. Ou seja, é escolhido o atributo que traga uma melhor separação das classes. O número de estimadores utilizados foi 30, o que indica que foram construídas 30 árvores. Como o máximo de profundidade para cada árvore, foi utilizado o valor de 15 ramificações.

Os atributos que tiveram maior peso na criação das árvores, foram os valores das *odds* para cada classe, o valor da diferença entre pontos obtidos na época por cada equipa e os pontos obtidos nos últimos 5 jogos para cada equipa.

Na Tabela 11, é possível encontrar os valores obtidos para cada classe, quando o modelo foi submetido a um novo conjunto de dados.

Tabela 11 - Matriz de confusão do modelo RF

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	287	138	147	50,17
Previsto A	11	58	11	72,50
Previsto D	0	0	0	0,00
Recall (%)	93,31	29,59	0,00	

Da análise da matriz, é possível retirar mais uma vez a conclusão de que a classe "*Draw*" foi a pior classificada, não existindo nenhum exemplo classificado como sendo desta classe. Em oposição, a classe "*Home*" foi prevista com uma *accuracy* de 93%.

6.3.4. K – Nearest Neighbors

De seguida, foi construído o modelo *K - Nearest Neighbors*. Este algoritmo utilizou os seguintes parâmetros: o número de vizinhos com o valor dos 5 mais próximos; a distância Euclidiana como métrica; e os valores dos pesos de forma uniforme. Este modelo conferiu maior peso aos atributos referentes às *odds* de cada classe, à diferença entre golos e ao número de remates à baliza.

Os valores obtidos por este classificador estão apresentados na Tabela 12.

Tabela 12 - Matriz de confusão do modelo KNN

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	253	84	122	55,12
Previsto A	20	71	24	61,74
Previsto D	30	33	14	18,18
Recall (%)	83,50	37,77	8,75	

Da matriz de confusão do classificador KNN, verifica-se que a classe "*Home*" foi aquela em que o modelo apresentou uma maior taxa de acerto e, em oposição, a classe "*Draw*" foi aquela em que o classificador teve maior taxa de erro. Por sua vez, a classe "*Away*" foi a que o modelo classificou com uma melhor precisão.

6.3.5. Logistic Regression

O modelo *Logistic Regression* foi criado no formato *one-vs-rest* para este problema de multi-classe e utilizou como parâmetro de função de perda, a entropia cruzada. Os atributos que o modelo considerou como devendo ter maior peso na classificação

foram: o valor das *odds* para o resultado final, o valor médio de golos cometidos por cada equipa na época e o valor das *odds* para mais/menos de 2 golos e meio.

Na Tabela 13, encontra-se representada a matriz de confusão resultante da aplicação do modelo aos dados de treino.

Tabela 13 - Matriz de confusão para o modelo LR

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	251	108	120	52,40
Previsto A	32	83	33	56,08
Previsto D	12	10	3	12,00
Recall (%)	85,08	41,29	1,92	

Ao observar a matriz de confusão produzida por este modelo, observa-se que mais uma vez o modelo teve uma grande dificuldade em classificar os exemplos que pertenciam a classe" *Draw*" e que a classe "*Home*" foi, em oposição, a que melhor taxa de acerto apresentou na sua classificação.

6.3.6. Support Vector Machine

O modelo *Support Vector Machine* foi implementado com uma configuração *one-vs-one*, devido à existência de três classes como resultado final, funcionando neste esquema para problemas de muti-classe.

Os atributos que demonstraram ser mais importantes para ajudar na classificação foram: a diferença entre os pontos obtidos por ambas as equipas na época, o valor das *odds* para o resultado final e a diferença entre os pontos obtidos nos últimos 5 jogos.

Na Tabela 14, é apresentada a matriz de confusão e os valores de *recall* e *precision* obtidos para este classificador.

Tabela 14 - Matriz de confusão para o modelo SVM

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	298	196	158	45.71
Previsto A	0	0	0	0.00
Previsto D	0	0	1	0,00
Recall (%)	100.00	0.00	0,00	

Da análise da tabela, observa-se que as classes "*Draw*" e "*Away*" não conseguiram ser classificadas. Ou seja, o classificador previu todas as amostras como sendo da

classe "*Home*" tendo assim obtido uma taxa de acerto de 100% para esta classe e de 0% para as restantes.

6.3.7. XGBoost

O último modelo a ser construído foi o *XGBoost*, tendo sido implementado com o valor de taxa de aprendizagem de 0.1, com o valor máximo de profundidade em cada árvore criada de 3 ramificações e com o número de estimadores, ou seja, de árvores criadas, igual a 40.

Este modelo utilizou como features principais os valores : *odds*, diferença pontual das equipas ao longo da época, pontos obtidos nos últimos 5 jogos e média de golos marcados na época.

Na Tabela 15, encontra-se apresentada a matriz de confusão que resultou da aplicação do classificador *XGBoost* ao *dataset* de treino.

Tabela 15 - Matriz de confusão para o modelo XGBoost

	Real H	**Real A**	**Real D**	**Precision (%)**
Previsto H	275	110	140	52.30
Previsto A	26	77	21	62.10
Previsto D	1	0	2	66.67
Recall (%)	91.06	41.18	1.23	

Da análise dos resultados obtidos por este classificador, pode concluir-se que a classe que obteve melhor taxa de acerto foi a "*Home*", com um valor de 91,06%. Contrariamente ao sucedido para esta classe, na previsão da classe "*Away*" o modelo acertou só em 41,18% dos casos e relativamente à classe "*Draw*", teve um mau desempenho acertando em, aproximadamente, 2% dos casos.

6.3.8. Comparação dos modelos

De forma a que a comparação entre os diversos modelos fosse realizada de uma forma mais simples, foi construída a Tabela 16 que apresenta os valores da taxa de acerto ou *accuracy*, e a taxa de erro obtidos por cada classificador.

Tabela 16 - Tabela comparativa dos algoritmos na previsão do target "FTR"

Métrica	**RN**	**Decision Tree**	**RF**	**KNN**	**LR**	**SVM**	**XGBoost**
Accuracy (%)	**53,07**	**53,54**	**52,91**	**51,92**	**51,69**	**45,70**	**54,30**

Pela análise da tabela, percebe-se que o modelo que apresentou uma *accuracy* mais elevada foi o *XGBoost*, acertando em 54,3% dos exemplos do *dataset* de treino. O segundo melhor resultado, foi obtido pelo classificador *Decision Tree*, com um valor de 53,54% de *accuracy*. Em oposição, o modelo que teve a performance mais baixa foi o classificador SVM que só conseguiu classificar corretamente 45,7% do *dataset*.

Conclui-se ainda que todos os classificadores obtiveram melhores resultados na classificação da classe "*Home*" e que tiveram muita dificuldade em classificar a classe "*Draw*", o que vai de encontro aos resultados dos trabalhos reportados na literatura.

6.4. 2º Experiência – Previsão de "mais/menos 2.5 golos" para a Liga Portuguesa

Na segunda experiência pretendia-se prever se, num dado jogo, iriam ocorrer mais do que 2 golos, no total dos golos marcados por ambas as equipas.

Para realizar esta previsão, foram aplicados os diversos algoritmos ao *dataset* de treino para criar os modelos e, posteriormente, com os dados do *dataset* de teste, foi realizada a avaliação dos mesmos. Como métricas para medir o desempenho que os vários classificadores alcançaram, foram utilizados os valores de: *accuracy* (ou taxa de acerto), *precision*, *recall*, *F1-Score* e AUC.

Na Tabela 17, está exposta a comparação do desempenho obtido pelos vários modelos.

Tabela 17 - Tabela comparativa dos classificadores para a previsão do target "M2_5Goals"

Modelo	Accuracy (%)	Precision (%)	Recall (%)	F1	AUC
RN - Keras	53,06	53,53	77,28	0,63	0,54
Decision Tree	53,38	52,50	99.13	0,69	0,52
RF	49,85	50,69	91,65	0,65	0,44
KNN	48,46	53,29	35,83	0,43	0,51
LR	51,53	51.59	97.92	0,68	0,56
SVM	51,54	51.54	100,00	0,68	0,50
XGBoost	55.37	54.33	85.87	0,67	0,55

Da análise da tabela, verifica-se que o classificador *XGBoost* foi o que apresentou uma maior taxa de acerto para este problema de classificação, acertando em 55,37% dos dados de teste. O valor de *precision* foi de 54,33%, o que indica que só 54% dos exemplos positivos pertenciam realmente a essa classe. A métrica *recall* obteve um valor de 85%, revelando que este modelo conseguiu classificar corretamente as classes. Da combinação dos dois valores anteriores, surge o valor do *F1-Score* que

varia entre 1 e 0, e que indica a qualidade geral do modelo, o qual foi de 0,67. O valor AUC, como anteriormente referido, indica a área sob a linha e conseguiu um valor de 0,55.

O classificador que obteve piores resultados neste problema de classificação foi o KNN, conseguindo acertar somente em 48,46% dos casos. Tendo em conta que a *baseline* de acertos é de 50%, quando se aposta aleatoriamente, conclui-se que este modelo tem um mau desempenho na classificação dos exemplos de teste.

6.5. 3º Experiência – Previsão de "mais/menos de 10.5 cantos" para a Liga Portuguesa

Na terceira experiência, pretendia-se realizar uma previsão sobre a existência de mais do que um total de 8 cantos marcados, por ambas as equipas, até ao final do jogo. Para este efeito, foram utilizados os mesmos classificadores para realizar a comparação da forma como se comportavam na classificação de novos exemplos. As métricas utilizadas para avaliar os modelos foram as mesmas que na abordagem anterior.

Na Tabela 18, encontra-se representada a comparação do desempenho alcançado pelos diversos modelos quando aplicados ao conjunto de teste.

Tabela 18 - Tabela comparativa dos classificadores para previsão do target"M8_5Corners"

Modelo	Accuracy (%)	Precision (%)	Recall (%)	F1	AUC
RN - Keras	71,63	71,63	100,00	0,83	0,55
Decision Tree	71,63	71,63	100,00	0,83	0,49
RF	71,78	71,74	100,00	0,84	0,54
KNN	71,27	72,28	97,66	0,83	0,51
LR	52,15	76,62	49,58	0,60	0,58
SVM	71,63	71,63	100,00	0,83	0,49
XGBoost	71,63	71,63	100,00	0,83	0,57

Verificou-se que na previsão de "mais do que 8 cantos", os resultados eram bastante melhores do que para a classificação de "mais do que 2 golos" por jogo. Foi também apurado que os algoritmos no geral obtiveram valores próximos de 70%, tendo o modelo *Random Forest* obtido o melhor valor com uma taxa de acerto de 71,78%. O classificador *Logistic Regression* foi o que obteve piores resultados, tendo acertado somente em 52,15% dos casos.

Estes valores mais elevados deveram-se ao facto de as classes do *dataset* se encontrarem em proporções diferentes, tendo a classe “mais de 8 cantos” uma percentagem de 72% das amostras do conjunto de dados. Como forma de solucionar este problema, foram testados os mesmos algoritmos neste conjunto de dados, mas desta vez, para realizar a previsão de se existirá, no tempo total do jogo e por ambas as equipas, um valor superior ou inferior a 10 cantos.

Como foi anteriormente referido no documento, neste conjunto de dados as classes já se encontravam mais balanceadas, uma vez que existiam 52% das amostras pertencentes à classe 1 e 47% das amostras pertencentes à classe 0.

Na Tabela 19, encontra-se realizada a mesma comparação entre os classificadores, mas com o objetivo de efetuar a classificação da existência de mais ou menos de 10 cantos.

Tabela 19 - Tabela comparativa dos classificadores para previsão do target"M10_5Corners"

Modelo	Accuracy (%)	Precision (%)	Recall (%)	F1	Auc
RN - Keras	52,22	52,37	97,06	0,68	0,49
Decision Tree	52,22	52,24	99,42	0,68	0,50
RF	52,22	52,22	100,00	0,69	0,47
KNN	52,68	52,46	100,00	0,69	0,53
LR	53,92	53,21	97,06	0.69	0,57
SVM	50,54	51,83	97,06	0.68	0,49
XGBoost	52,68	54,27	90,19	0.68	0,58

Da análise da tabela, é possível verificar que as métricas apresentam valores menores de *accuracy*, o que confirma que os valores mais elevados desta taxa no teste anterior seriam devidos ao desequilíbrio do número de amostras pertencente a cada classe. Para a previsão de mais ou menos de 10 cantos, os classificadores obtiveram todos um desempenho muito semelhante, rondando os 52% de taxa de acerto. O melhor resultado foi obtido para o classificador *Logistic Regression* com o valor de 53,92% de *accuracy*.

Os segundos melhores resultados foram obtidos pelos classificadores *XGBoost* e *KNN*, com os valores de 52,68% de *accuracy* e *F1-Score* de 0,69.

O modelo que apresentou os piores resultados foi o *SVM* que somente acertou em, aproximadamente, metade das amostras do *dataset* de teste.

6.6. Comparação de Liga Portuguesa com a Liga Inglesa

De forma a perceber qual a influência que a liga tinha na previsão do resultado final do jogo, foi realizado um teste comparativo utilizando os mesmos modelos, em dados diferentes. Para o efeito, utilizaram-se os dados das épocas de 2014/2015 até 2019/2020, mas desta vez da primeira liga portuguesa. Neste segundo caso, o conjunto de dados era relativamente menor, uma vez que a liga portuguesa é disputada por 18 equipas, enquanto que a liga inglesa tem 20 equipas. O *dataset* foi então constituído por 1836 exemplos no total, sendo cada época composta por 306 jogos. De forma a perceber como os dados se encontravam distribuídos, foi analisada a distribuição dos resultados em termos de cada *target* a prever, como se mostra na Figura 46.

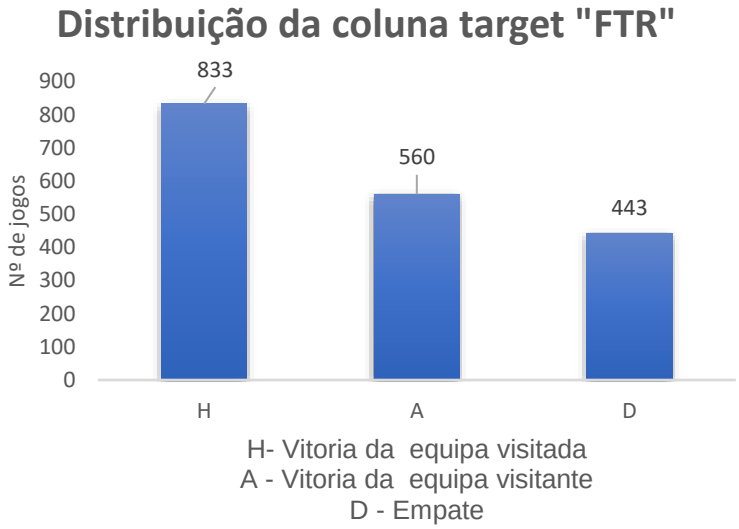


Figura 46 - Distribuição das classes do "FTR" para o dataset da liga portuguesa

O mesmo que se constatou na liga inglesa, também se verificou na liga portuguesa, em que a classe "*Home*" apresenta uma maior percentagem de jogos ganhos do que a classe "*Away*", sendo a classe "*Draw*" a que apresenta menos exemplos no conjunto de dados.

Ao conjunto de dados criado, foi realizado o mesmo pré-processamento que aos restantes, foram implementadas as mesmas métricas que tinham sido aplicadas ao dataset da liga inglesa e foi efetuada a divisão dos dados de forma análoga à que foi realizada para a outra liga. Ou seja, 70% dos dados para treino do modelo e 30% para teste e avaliação do mesmo. Depois de todas estas fases, foram então testados os modelos com os novos exemplos do novo conjunto de dados.

Na Tabela 20 , encontra-se a comparação dos resultados obtidos pelos modelos nas duas ligas, com a percentagem dos exemplos que foram corretamente classificados, em relação ao número total de exemplos classificados.

Tabela 20 - Tabela comparativa da Liga Portuguesa vs Liga Inglesa para o target "FTR"

Modelo / Liga	RN	Decision Tree	RF	KNN	LR	SVM	XGBoost
Liga Portuguesa	58,29	48,38	58,75	56,45	58,78	49,80	60,09
Liga Inglesa	53,07	53,54	52,91	51,92	51,69	45,70	54,30

Da análise desta tabela foram retiradas algumas conclusões. Primeiramente, concluiu-se que os valores da taxa de acerto eram superiores para a liga portuguesa, devendo-se este facto a questões relacionadas com o desempenho das equipas que constituem a liga. De facto, a liga inglesa é mais variável, tornando-se, por consequência, mais difícil de prever. A liga portuguesa é mais estável, no sentido em que já se sabe, de antemão, que equipas são candidatas à vitória dos jogos e do campeonato.

Por outro lado, verificou-se que o classificador que apresentava melhores resultados continuava a ser o *XGBoost*, com uma taxa de acerto de aproximadamente 60%, sendo seguido pelo classificador *K- Nearest Neighbors*. Contrariamente ao que tinha acontecido com a liga inglesa, o classificador *Decison Tree* foi o que obteve o pior desempenho para este *dataset*, conseguindo acertar somente em 48,38% dos casos.

CONCLUSÕES E TRABALHO FUTURO

Tal como referido anteriormente no documento, o mercado das apostas desportivas *online* é um setor em grande crescimento a nível mundial e, inclusivamente, em Portugal. Como tal, a Mythical Technologies, proprietária do *Traderline*, pretende apoiar da melhor forma os seus apostadores, proporcionando-lhes prognósticos que os ajudem na tomada de decisão.

Nesse sentido foi criado o presente estágio, cujo objetivo era criar um sistema de apoio à decisão para auxiliar os utilizadores do Traderline, mediante o desenvolvimento de vários modelos para prever eventos que ocorrem nos jogos de futebol, com recurso a diferentes algoritmos de *Machine Learning*. Dada a grande diversidade de possíveis eventos e o tempo limitado do estágio, os selecionados foram: o resultado final do jogo (vitória, empate ou derrota), se seriam marcados mais do que 2 golos (no total pelas duas equipas), se seriam marcados mais do que 10 cantos (também no total pelas duas equipas).

Para atingir esse objetivo, dividiu-se o trabalho em várias fases. Primeiramente, foram analisadas as principais casas de apostas *on-line* disponíveis no mercado e alguns dos trabalhos científicos desenvolvidos na área, os quais permitiram perceber quais os atributos e algoritmos mais importantes a considerar. De seguida, foram investigadas as fontes de dados existentes e selecionadas as que mais se adequavam ao estudo, tendo a escolha final recaído na *Football Data*. Posteriormente, foram analisados os dados extraídos das fontes, realizado o seu pré-processamento e construídos os *datasets* a usar no treino e teste dos modelos. Foram consideradas 3 experiências que diferiram no *target* a prever. Na primeira experiência, cujo objetivo era classificar os jogos em vitória para a equipa visitada, vitória para a equipa visitante, ou empate, dos vários algoritmos testados, o que obteve melhores resultados foi o *XGBoost* com uma *accuracy* de 54,3%. Nesta foram usados apenas dados da primeira liga inglesa. Na segunda experiência, em que se pretendia que o modelo previsse se iriam existir mais do que 2 golos, ou menos do que 2 golos, no total, para ambas as equipas, o melhor resultado foi conseguido pelo classificador *XGBoost*, com um valor de accuracy de 55,37%. Na terceira experiência, cuja finalidade era prever se iriam existir mais do que 10 cantos, ou menos do que 10 cantos, no total, para ambas as equipas, o melhor resultado foi obtido para o classificador *Random Forest* com um valor de *accuracy* de 53,92%. Um 4ª e última experiência, teve como finalidade verificar a influência que a liga teria nos resultados de previsão do resultado final do jogo. Foram então usados os mesmos modelos da 1ª experiência, mas nos dados da primeira liga portuguesa, sendo que o melhor valor foi novamente obtido pelo classificador *XGBoost*, mas com um valor de 60% de *accuracy*, confirmando que os resultados dependem efetivamente da liga em causa.

Terminado o estágio, conclui-se que o objetivo foi cumprido no sentido em que foram desenvolvidos e testados vários modelos de *Machine Learning* para prever alguns dos eventos que ocorrem nos jogos de futebol. Relativamente ao sistema de apoio à decisão previsto, já existem alguns modelos, já existe uma base de dados, mas não foi desenvolvida a interface por falta de tempo.

Convém referir que houve algumas dificuldades ao longo de todo o processo, tais como, o desconhecimento da estagiária sobre o tema futebol, a complexidade do problema de previsão em si, as inúmeras iterações necessárias para chegar aos resultados finais apresentados, e também a COVID-19 que teve consequências a nível pessoal e que obrigou ao tele-trabalho e ao consequente distanciamento dos orientadores e colegas da empresa.

Não obstante, o balanço final é muito positivo e a estagiária agradece uma vez mais à Mythical Technologies por esta experiência enriquecedora, quer a nível técnico, quer a nível pessoal. Além de lhe ter possibilitado consolidar conhecimentos e adquirir novos, também lhe permitiu experimentar o ambiente de trabalho em contexto empresarial, numa entidade que tão bem a acolheu.

Trabalho futuro:

Apesar de os resultados de previsão se situarem ao nível dos que se encontram na literatura, é sempre um desafio tentar melhorá-los.

Desta forma, algumas das sugestões para trabalho futuro seriam:

- Considerar novas *features*, tais como as referentes a eventos meteorológicos e desempenho dos jogadores, bem como as ligadas a fatores psicológicos e físicos da equipa.
- Implementar novos modelos de previsão.
- Testar o aumento da dimensão do *dataset*.
- Alterar os *target* a prever na primeira abordagem, utilizando somente duas classes. Ou seja, classificar entre a equipa visitada ganhar, ou não ganhar. Alternativamente, criar um modelo diferente para prever cada classe, ou seja, para prever se existirá empate ou não, se a equipa da casa irá ganhar ou não, se equipa de fora irá ganhar ou não.
- Desenvolver modelos para outros tipos de eventos que ocorrem nos jogos de futebol.

REFERÊNCIAS BIBLIOGRÁFICAS

Bohanec, M. (December de 2009). DECISION MAKING: A COMPUTER-SCIENCE AND INFORMATION-TECHNOLOGY VIEWPOINT . *DECISION MAKING: A COMPUTER-SCIENCE AND INFORMATION-TECHNOLOGY VIEWPOINT* .

Leitão Gomes, J. C. (2015). *Sistema Inteligente de Apoio à Decisão de.* Minho.

Moura Ferraz, D. M. (2011). *Casas de apostas online .* Porto.

Ramager, B. M. (December de 2010). DATA MINING TECHNIQUES AND APPLICATIONS. *DATA MINING TECHNIQUES AND APPLICATIONS*, pp. 301-305.

888Sport. (2020). *888Sport*. Obtido de 888Sport: https://www.888sport.com/pt/#/home

Aderaldo, S. (14 de 10 de 2019). *Algoritmo XGBoost utilizando computação GPU, aplicação no Google Colaboratory e personalização dos hiperparâmetros.* Obtido de Medium: https://medium.com/@steferson_69089/algoritmo-xgboost-utilizando-computa%C3%A7%C3%A3o-gpu-aplica%C3%A7%C3%A3o-no-google-colaboratory-e-personaliza%C3%A7%C3%A3o-dos-c3aa4065783f

Alsultanny, Y. (31 de October de 2011). Selecting a suitable method of data mining for successful forecasting . *Selecting a suitable method of data mining for successful forecasting* .

Ambrósio, P. E. (2012). *Redes neurais artificiais no apoio ao diagnóstico diferencial de lesões intersticiais pulmonares .* Ribeirão preto .

Anacleto, A. C. (2009). *Aplicação de Técnicas de Data Mining em Extracção de Elementos de Documentos Comerciais.* Porto.

Anaconda. (2020). Obtido de Anaconda: https://www.anaconda.com/

API, F. (2021). *Football API*. Obtido de Football API: https://www.api-football.com/

ApostaLegal. (2020). *Casas de apostas online legais - 2020*. Obtido de ApostaLegal: https://apostalegal.pt/

Baranauskas, J. A. (2016). *Clustering.*

Baranauskas, J. A. (2019). *Métodos de Amostragem.* Obtido de Universidade São Paulo: http://dcm.ffclrp.usp.br/~augusto/teaching/ami/AM-I-Metodos-Amostragem.pdf

Betano. (2020). *Betano*. Obtido de Betano: https://www.betano.pt/

Betclic. (2020). *Betclic*. Obtido de Betclic: https://www.betclic.pt/

Betfair. (2020). *Betfair*. Obtido de Betfair: https://www.betfair.com/pt

Betfair. (2021). *Betfair*. Obtido de Betfair: https://www.betfair.com/

Betradar. (2020). *Betradar*. Obtido de Betradar: https://www.betradar.com/

Betway. (2020). *Betway*. Obtido de Betway: https://www.betway.pt/desportos/

Brownlee, J. (25 de Maio de 2018). *Uma introdução suave ao método bootstrap.* Obtido de Machine Learning Mastery: https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/

Bunker, R., & Thabtah, F. (September de 2017). A Machine Learning Framework for Sport Result Prediction. *A Machine Learning Framework for Sport Result Prediction*.

Buursma, D. (2010). Predicting sports events from past results. *Predicting sports events from past results*.

Campello, R., & Mello, R. (27 de 02 de 2018). *SVM.* Obtido de Facom: http://www.facom.ufu.br/~backes/pgc204/Aula08-SVM.pdf

Campos, L. M. (22 de Outubro de 2014). Mineração de Dados com Detecção de Outliers em Tarefas de Predição de Séries Temporais. *Mineração de Dados com Detecção de Outliers em Tarefas de Predição de Séries Temporais*.

Campos, R. (28 de Nov. de 2017). Árvores de Decisão. *Árvores de Decisão*.

Chen, T., & Guestrin, C. (13-17 de 08 de 2016). XGBoost: A Scalable Tree Boosting System. *XGBoost: A Scalable Tree Boosting System*.

Coimbra, N. d. (27 de Novembro de 2018). *Futebol responsável pelo aumento das apostas desportivas online em Portugal*. (Noticias de Coimbra) Obtido em 10 de 01 de 2020, de https://www.noticiasdecoimbra.pt/futebol-responsavel-pelo-aumento-das-apostas-desportivas-online-em-portugal/

Coutinho, B. (28 de 7 de 2019). *Modelos de predição | SVM.* Obtido de Medium: https://medium.com/turing-talks/turing-talks-12-classifica%C3%A7%C3%A3o-por-svm-f4598094a3f1

CRISP-DM: Towards a Standard Process Model for Data Mining. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*.

Data, F. (2020). *Football Data*. Obtido de Football Data : https://www.football-data.co.uk/

DataCamp. (2020). *Leave-one-out*. Obtido de https://campus.datacamp.com/courses/model-validation-in-python/cross-validation?ex=10

Debugeverything. (2021). *Qual a diferença entre base de dados relacional e não relacional?* Obtido de Debugeverything: https://debugeverything.com/diferenca-base-de-dados-relacional-e-nao-relacional/

Dihanster, W. (2020). KNN-Introduction to machine learning algorithms. *Medim*.

Dmitrievsky, M. (31 de Julho de 2018). *FLORESTA DE DECISÃO ALEATÓRIA NA APRENDIZAGEM POR REFORÇO*. Obtido de MQL5: https://www.mql5.com/pt/articles/3856

dsssystem. (01 de 2010). Obtido de dsssystem: http://dsssystem.blogspot.com/2010/01/architecture-of-decision-support.html

Duarte, L. (2015). *1X2 – PREVISÃO DE RESULTADOS DE JOGOS DE FUTEBOL.* Porto.

Eoddsmaker. (2020). *Eoddsmaker*. Obtido de Eoddsmaker: https://eoddsmaker.co.uk/

EscOnline. (2020). *Apostas Desportivas*. Obtido de EscOnline: https://www.estorilsolcasinos.pt/pt/apostas

Facure, M. (2017). *Logistic Regression .*

Fernandes, F. A. (2019). *Previsão de resultados no futebol por meio de técnicas de aprendizado de máquina.* Belo Horizonte: Fevereiro.

football, A. (2020). *Api football*. Obtido de Api football: https://www.api-football.com/

Forebet. (2021). Obtido de Forebet: https://www.forebet.com/

Freitas, J. A. (2006). *Uso de Técnicas de Data Mining para Análise de Bases de Dados Hospitalares com Finalidades de Gestão .* Porto.

Gomes, T. (Dezembro de 2014). Ferramentas Open Source de Data Mining. *Ferramentas Open Source de Data Mining*.

Gomes, T. (01 de 06 de 2019). *O mundo das apostas .* Obtido de PalmeirasOnline: https://palmeirasonline.com/2019/06/01/o-mundo-das-apostas-desportivas/

Gonçalves, R. A. (2019). *Aplicações de Betting e Trading.* Coimbra.

Gonzalo, E., Miniz, Z., Nieto, P., Sanchez, A., & Fernandez, M. (29 de 06 de 2016). Hard-Rock Stability Analysis for Span Design inEntry-Type Excavations with Learning Classifiers.

Grover, P. (9 de 12 de 2017). *Gradient Boosting from scratch.* Obtido de Medium: https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

grover, P. (9 de 12 de 2017). *Gradient boosting from scratch.* Obtido de Medium: https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

Guedes, M. (2017). *SQL vs NoSQL.* Obtido de TreinaWeb: https://www.treinaweb.com.br/blog/sql-vs-nosql-qual-usar/

Haldurai. L, R. R. (June de 2016). Data Mining Algorithms and its Applications in Health Care Sectors. *Data Mining Algorithms and its Applications in Health Care Sectors*.

Herbinet , C. (2018). *Predicting Football Results Using Machine Learning Techniques.* London: Imperial College London.

Herbinet, C. (2018). *Predicting Football Results Using Machine Learning Techniques.* DEPARTMENT OF COMPUTING IMPERIAL COLLEGE OF SCIENCE,TECHNOLOGY AND MEDICIN.

IBM. (2012). *IBM SPSS Modeler CRISP-DM Guide.* USA.

Igawa, R. A. (16 de Novembro de 2015). K - Nearest Neighbours. Londrina, Brazil.

Igiri, C. P., & Nwachukwu, E. O. (December de 2014). An Improved Prediction System for Football a Match Result . *An Improved Prediction System for Football a Match Result* .

Isaac. (2015). *Las tres mejores IDEs Python de código abierto*. Obtido de linuxdictos: https://www.linuxadictos.com/las-tres-mejores-ides-python-de-codigo-abierto.html

ISEC. (2021). *Instituto Superior de Engenharia de Coimbra* . Obtido de Apresentação Instituto Superior de Engenharia de Coimbra : https://www.isec.pt/pt/instituto/#lnkApresentacao

Italo, J. (2018). KNN. *Medium*.

JetBrains. (2021). *PyCharm*. Obtido de JetBrains: https://www.jetbrains.com/pycharm/

Jupyter. (2021). *Jupyter*. Obtido de Jupyter: https://jupyter.org/

Keras. (2021). *Keras*. Obtido de Keras: https://keras.io/

Konjevoda, P., & Štambuk , N. (4 de july de 2011). Open-Source Tools for Data Mining in Social Science. *Open-Source Tools for Data Mining in Social Science*.

Leal, R. S. (2017). Métricas Comuns em Machine Learning. *Medium*.

LigaPortugal. (2018). *REGULAMENTO DAS COMPETIÇÕES.*

Linoff, G. S., & A. Berry, M. J. (2011). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management.* Canada: Wiley Publishing, Inc.

Lorena, A. C., & Carvalho, A. C. (2007). Uma Introdução às Support Vector Machines. *Uma Introdução às Support Vector Machines*.

M., V. (08 de 02 de 2017). *Apostas desportivas online para iniciantes* . Obtido de Pplware: https://pplware.sapo.pt/internet/apostas-desportivas-iniciantes/

Matos, J. M. (2013). *APOSTAS DESPORTIVAS ONLINE COMPORTAMENTO E PERFIL DO APOSTADOR PORTUGUÊS.* Lisboa: ISCTE.

MongoDB. (2020). Obtido de MongoDB: https://www.mongodb.com/

Muller, A. (2015). Obtido de amueller: http://amueller.github.io/

NumPy. (2021). *Numpy*. Obtido de NumPY: https://numpy.org/

Owramipu, F., Eskandarian, P., & Mozneb , F. S. (October de 2013). Football Result Prediction with Bayesian Network in Spanish League-Barcelona Team. *Football Result Prediction with Bayesian Network in Spanish League-Barcelona Team*.

Padhy, N., Mishra, P., & Panigrahi, R. (June de 2012). The Survey of Data Mining Applications. *The Survey of Data Mining Applications*.

Palinggi, D. A., Ramos , F., Torres-Sospedra, J., Trilles, S., & Huerta, J. (June de 2019). Using Weather Condition and Advanced Machine Learning Methods. *Using Weather Condition and Advanced Machine Learning Methods*, pp. 17-20.

Pandas. (2021). *Pandas*. Obtido de Pandas: https://pandas.pydata.org/

PETTERSSON , D., & NYQUIST, R. (2017). *Football Match Prediction using Deep Learning* . Gothenburg, Sweden : CHALMERS UNIVERSITY OF TECHNOLOGY.

Pinho de Sá, L. H. (2016). *SISTEMA DE APOIO À DECISÃO PARA AVALIAÇÃO TÉCNICA DE.* Rio de Janeiro: Universidade Federal do rio de Janeiro.

Pinnacle. (2020). *Pinnacle*. Obtido de Pinnacle: https://www.pinnacle.com/pt/

Portugal, L. (2020). *Sobre a Liga*. Obtido em 24 de 08 de 2020, de Liga Portugal: https://www.ligaportugal.pt/pt/paginas/conteudos/apresentacao-da-liga/

Postman. (2021). *Postman* . Obtido de Postman: https://www.postman.com/

Prasad, A., Kurian, R., Bhatia, R., & Tan, X. (2020). *THE ENGLISH PREMIER LEAGUE* .

PredictZ. (2021). Obtido de PredictZ: https://www.predictz.com/

Python. (2021). *Python*. Obtido de Python: https://www.python.org/

Qiubing, R., Mingchao, L., & Han, S. (19 de 02 de 2019). Tectonic discrimination of olivine in basalt usingdata mining techniques based on major elements:a comparative study from multiple perspectives. *Tectonic discrimination of olivine in basalt usingdata mining techniques based on major elements:a comparative study from multiple perspectives*.

Ramos, J. (2016). Learning Management System .

Rapidminer. (2020). Obtido de Rapidminer: https://rapidminer.com/

Reinke, B. (17 de 11 de 2018). *Entenda como funciona a Premier League*. Obtido em 25 de 08 de 2020, de Leitura do Jogo : https://www.leituradejogo.com.br/entenda-como-funciona-a-premier-league/

Reinke, B. (07 de 03 de 2019). *A origem do futebol*. Obtido em 20 de 08 de 2020, de leitura do jogo .

Rijo, S. M. (2017). *Técnicas de Deep Learning para detecção de eventos em áudio .* Évora: Universidade de Évora.

Rivalo. (2020). *Rivalo*. Obtido de Rivalo: https://www.rivalo.com/pt/apostas/

RunningBall. (2020). *Running ball database*. Obtido de Rballdb: http://www.rballdb.com/

Samba, S. (2019). *Football Result Prediction by Deep Learning Algorithms.* Tilburg.

San, F. (10 de 12 de 2018). Obtido de Cision: https://www.prweb.com/releases/postman_2018_state_of_the_api_survey_containers_and_serverless_architecture_gain_developer_interest/prweb15812110.htm

Santana, F. (3 de Março de 2020). *Data Analysis*. Obtido de MinerandoDados: https://minerandodados.com.br/tres-tecnicas-de-amostragem-de-dados-utilizando-python/

Santana, R. (6 de Fevereiro de 2020). *Validação Cruzada: Aprenda de forma simples como usar essa técnica*. Obtido de MinerandoDados: https://minerandodados.com.br/validacao-cruzada-aprenda-de-forma-simples-como-usar-essa-tecnica/

Schade, G. (12 de Marco de 2019). Machine Learning: Métricas para Modelos de Classificação. *Machine Learning: Métricas para Modelos de Classificação*.

Schneider, C. F. (2018). *MACHINE LEARNING APLICADO NA PREVISÃO DE RESULTADOS DE PARTIDAS DE FUTEBOL: UM ESTUDO DE CASO PARA COMPARAÇÃO DE DIFERENTES CLASSIFICADORES.* Porto Alegre: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL.

SCHREIBER, J. N., BESKOW, A. L., MÜLLER, J. C., NARA, E. O., SILVA, J. I., & REUTER, J. W. (2017). *TÉCNICAS DE VALIDAÇÃO DE DADOS PARA SISTEMAS INTELIGENTES: UMA ABORDAGEM DO SOFTWARE SDBAYES.* ARGENTINA: COLÓQUIO INTERNACIONAL DE GESTÃO UNIVERSITÁRIA.

Scikit-Learn. (2021). *Scikit-Learn*. Obtido de Scikit-Learn: https://scikit-learn.org/stable/

Serafin, B. L. (2016). *Ações de marketing como ferramenta do aumento do brand equity : um estudo de caso da marca Traderline.* Coimbra.

Silva , J. C. (2018). Understand Random Forest. *Medium*.

Simeone, O. (05 de 11 de 2018). A Very Brief Introduction to Machine Learning. *A Very Brief Introduction to Machine Learning*.

Simões, A. C. (2008). *Mineração de dados baseada em árvores de decisão para análise do perfil de contribuintes* . Recife: Universidade Federal de Pemambuco.

Soccervista. (2021). Obtido de Soccervista: https://www.soccervista.com/

SportingBet. (2020). *SportingBet*. Obtido de SportingBet: https://sports.sportingbet.com/pt-br/sports

Sportsdata. (2020). *Sportsdata*. Obtido de Sportsdata: https://www.sportsdata.ag/

Sportsradar. (2020). *Sportsradar*. Obtido de Sportsradar: https://www.sportradar.com/

SportyTrader. (2020). *SportyTrader*. Obtido de SportyTrader: https://www.sportytrader.pt/

SRIJ. (2020). *Relaório do 1º trimestre de 2020.* Lisboa: SRIJ.

Statarea. (2021). Obtido de Statarea: http://www.statarea.com/

SuperAposta. (2020). *SuperAposta*. Obtido de SuperAposta: https://br.superaposta.com/

SupraBets. (2020). *SupraBets*. Obtido de SupraBets: https://www.suprabets.com/

Tavares, A. T. (2013). Predicting Results of Brazilian Soccer League Matches. *Predicting Results of Brazilian Soccer League Matches*.

Technologies, M. (2019). Obtido de Mythical Technologies: http://mythicaltechnologies.com/

TechTour. (04 de 04 de 2019). *Logistic Regression vs Random Forest Classifier.* Obtido de https://i.ytimg.com/vi/goPiwckWE9M/maxresdefault.jpg

Tensorflow. (2020). Obtido de Tensorflow: https://www.tensorflow.org/about

TipBet. (2020). *TipBet*. Obtido de TipBet: https://www.tipbet.com/en/online-sport-betting/

Traderline. (01 de 2021). *Sobre o Traderline* . Obtido de Traderline : https://beta.traderline.com/About

Ulmer, B., & Fernandez , M. (2014). Predicting Soccer Match Results in the english premier League. *Predicting Soccer Match Results in the english premier League*.

Vaish, P. (2018). Obtido de A data analyst: https://adataanalyst.com/machine-learning/knn/

Vapnik, V., & Cortes, C. (9 de 1995). Support vector machine . pp. 273-297.

Vaz, A. L. (06 de 06 de 2019). *Como lidar com dados desbalanceados em problemas de classificação.* Obtido de Medium: https://medium.com/data-hackers/como-lidar-com-dados-desbalanceados-em-problemas-de-classifica%C3%A7%C3%A3o-17c4d4357ef9

Vitibet. (2021). Obtido de Vitibet: https://www.vitibet.com/index.php?lang=pt

Vorhies, W. (26 de 07 de 2016). *CRISP-DM – a Standard Methodology to Ensure a Good Outcome.* Obtido de DataScienceCentral: https://www.datasciencecentral.com/profiles/blogs/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome

Weiss, S. M., Indurkhya, N., & Zhang, T. (2015). *Fundamentals of Predictive Text Mining.* Springer.

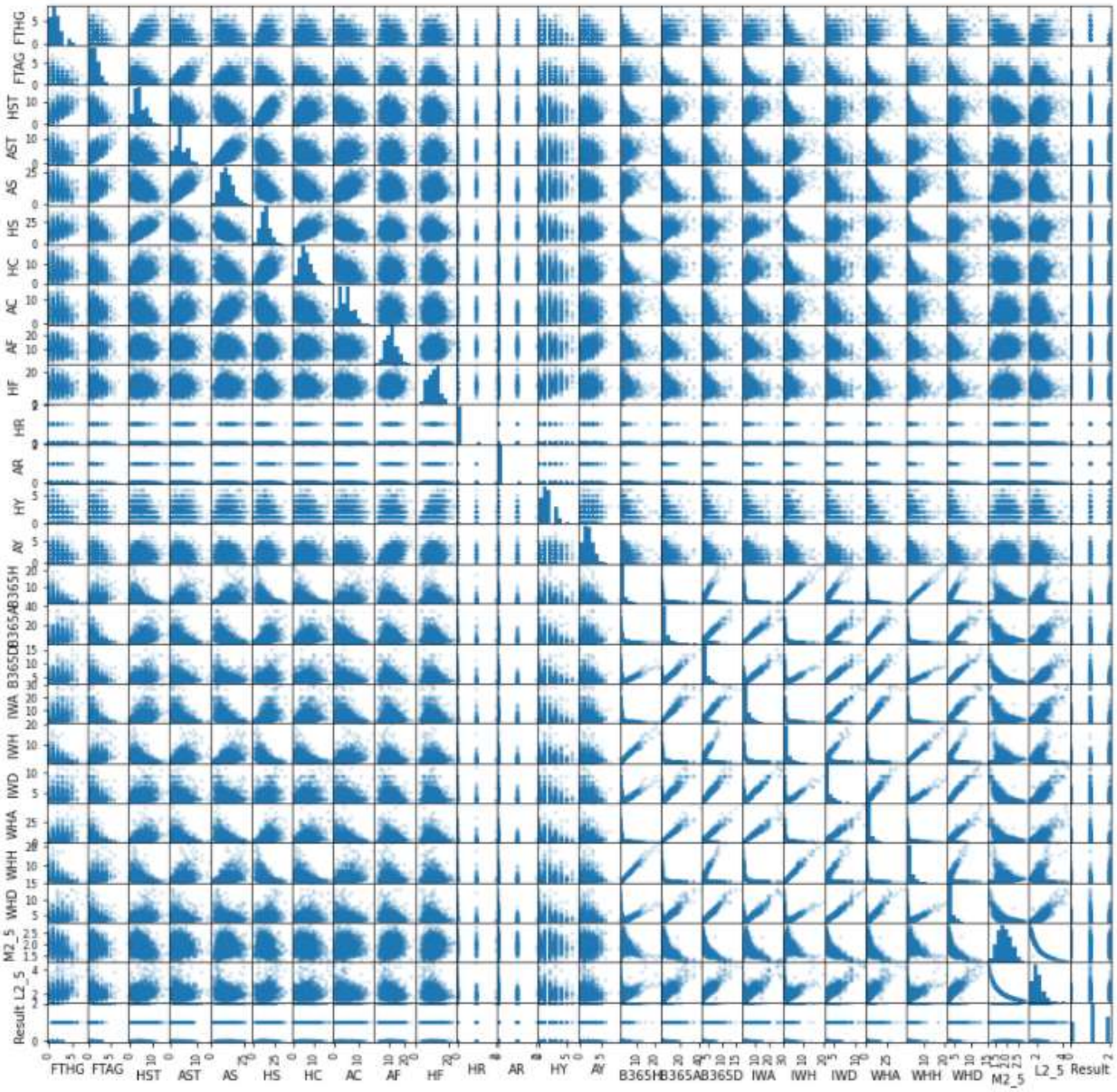
Yiu, T. (2019). Understanding Random Forest. *Towards Data Science*.

Zaan, T. (2017). *Predicting the outcome of soccer matches in order to make money with betting.* Erasmus University Rotterdam.

ANEXOS

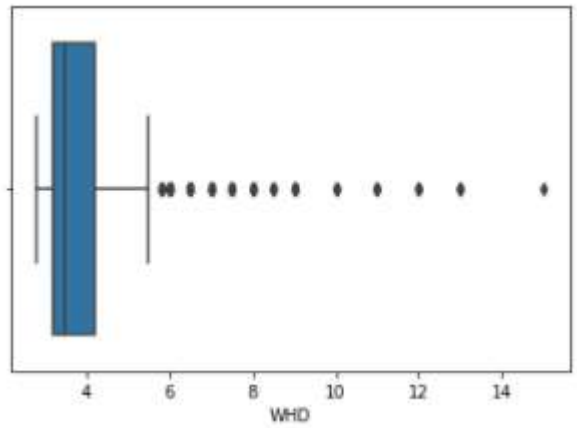
Anexo A

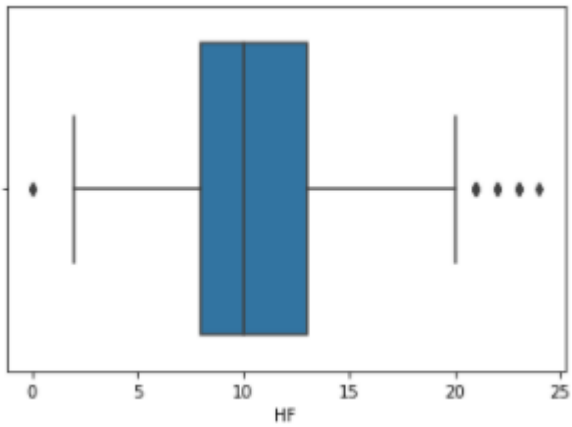
Scatter matrix do dataset original

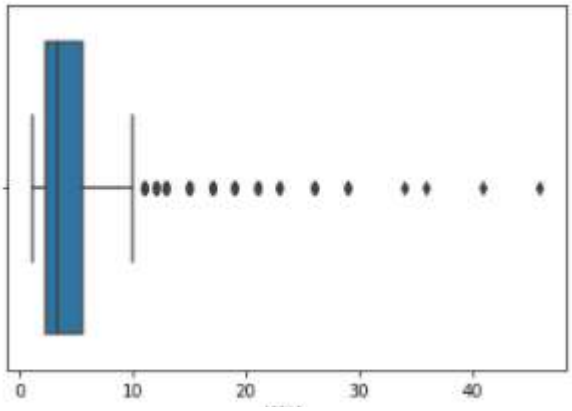


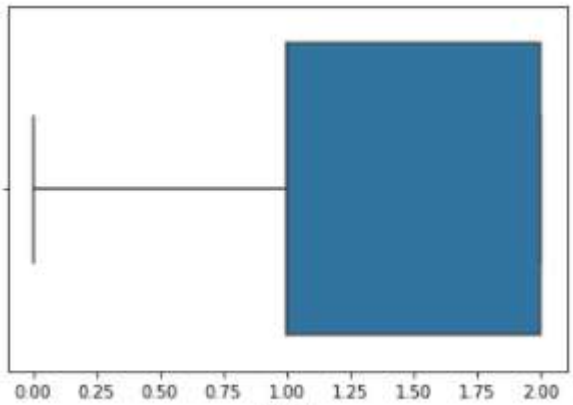
Anexo B

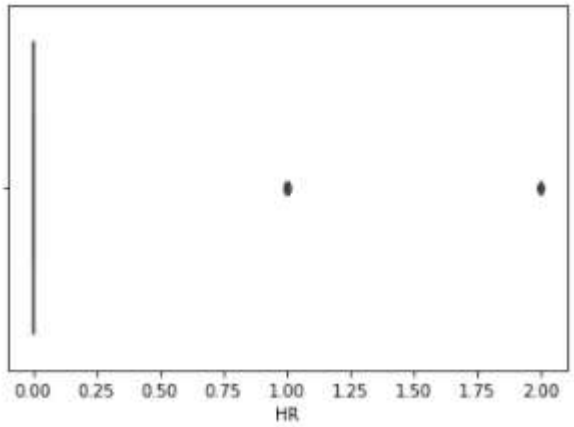
Box-plots dataset original

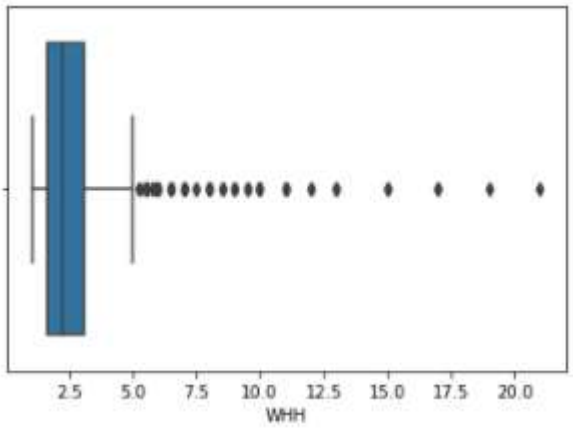


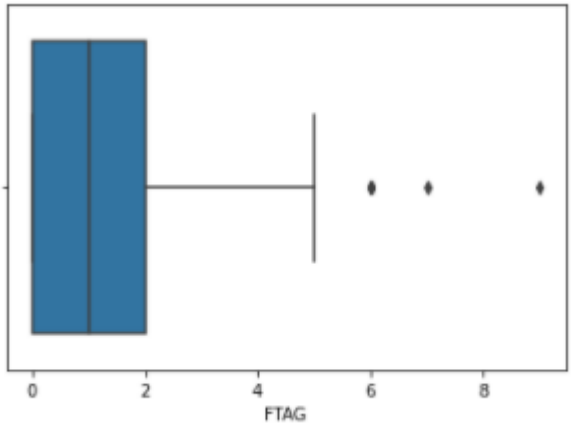


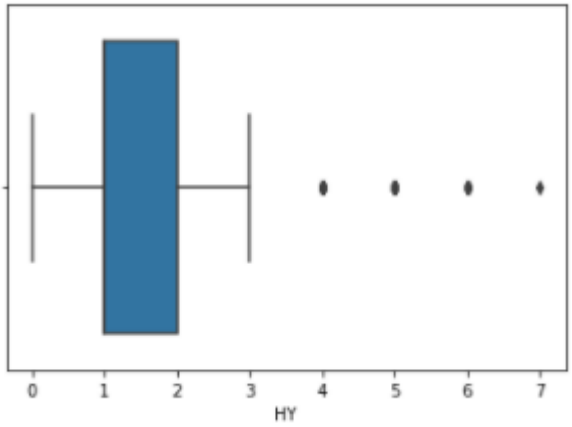


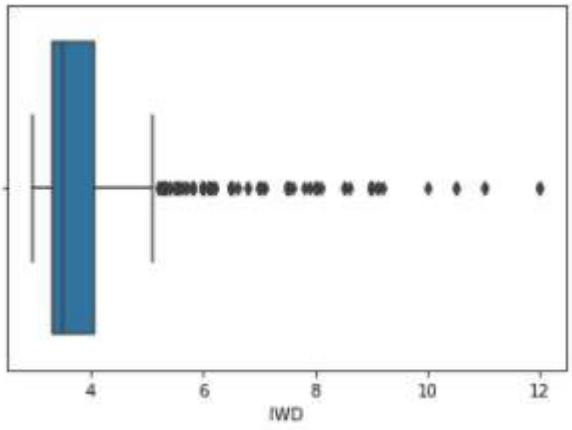


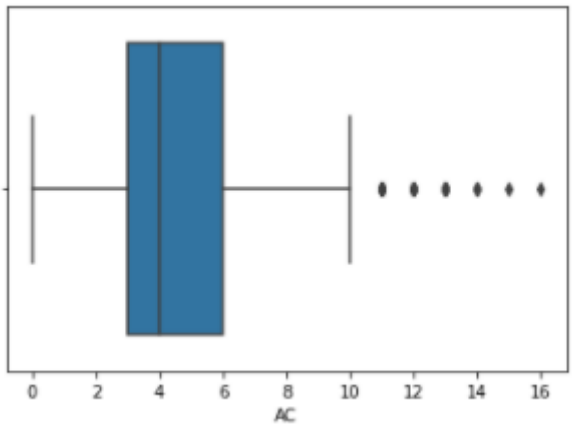


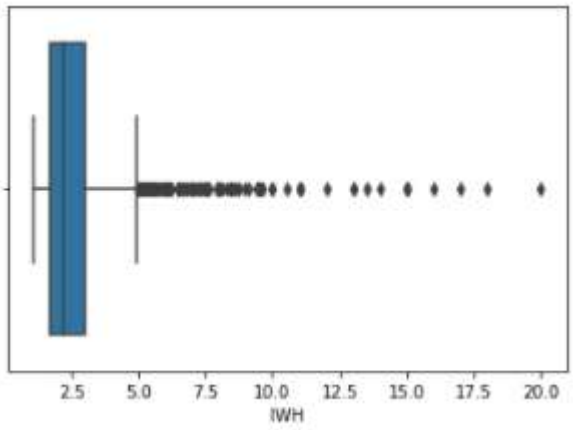


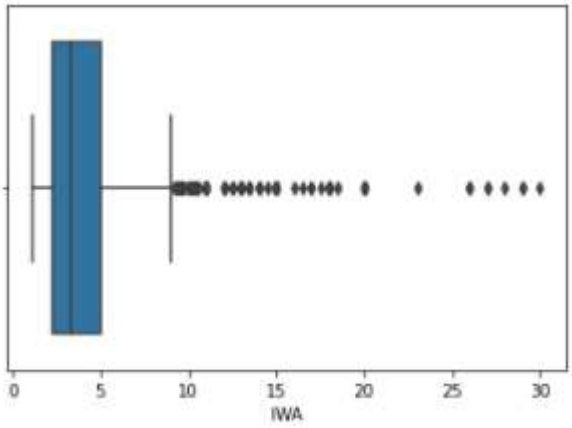


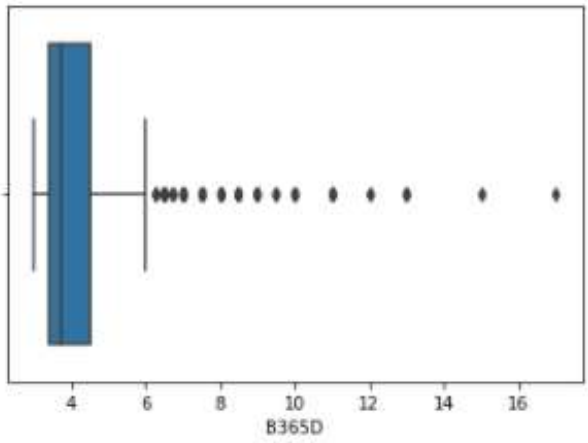


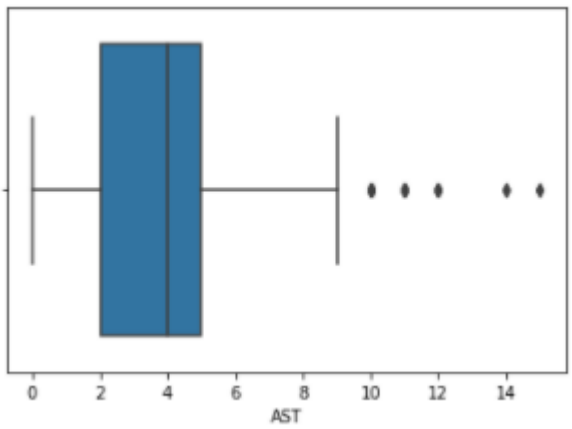

4
6
8
10
12
IWD

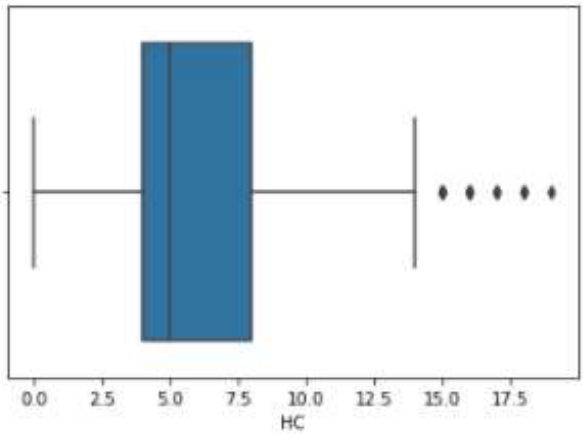

0
2
4
6
8
10
12
14
16
AC

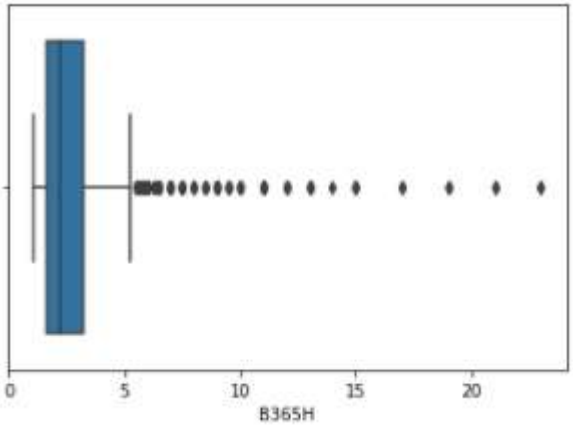

2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
IWH

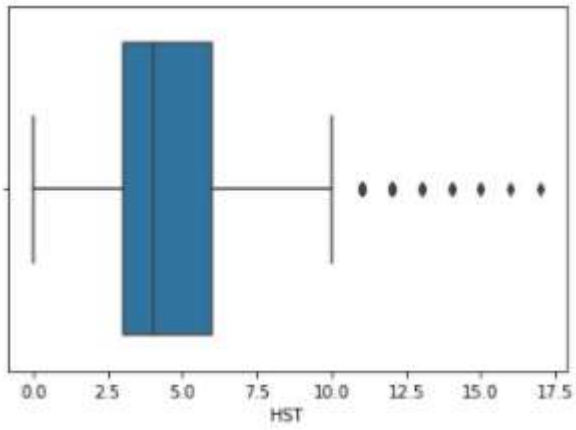

0
5
10
15
20
25
30
IWA

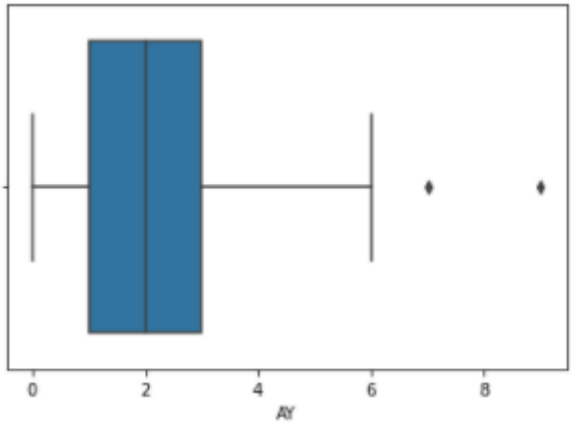

4
6
8
10
12
14
16
B365D

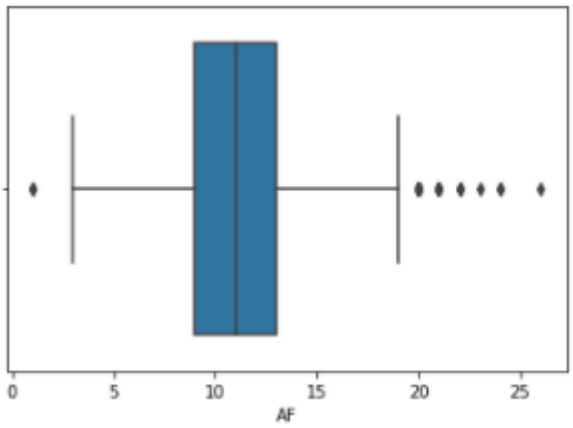

0
2
4
6
8
10
12
14
AST

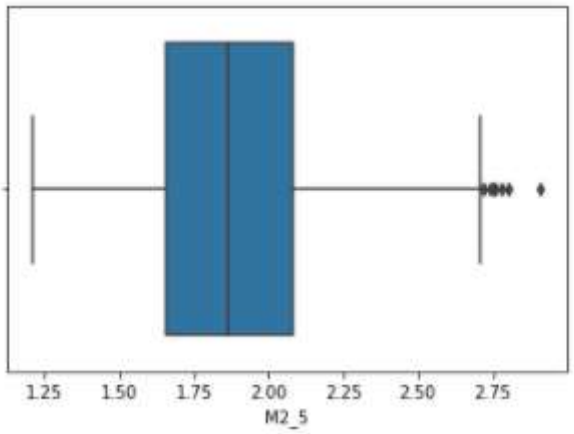

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
HC

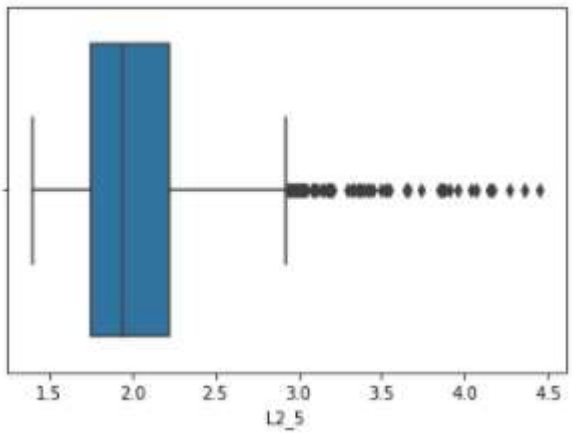

0
5
10
15
20
B365H

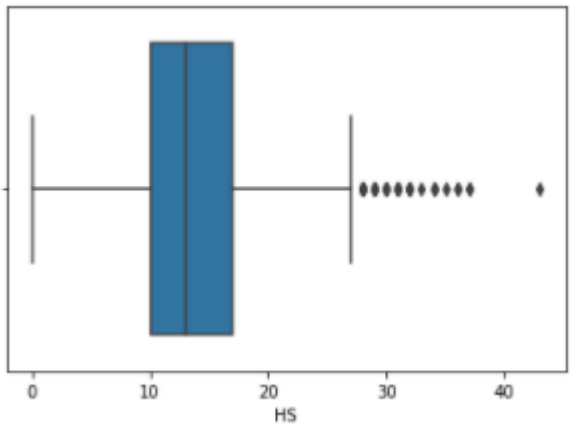

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
HST

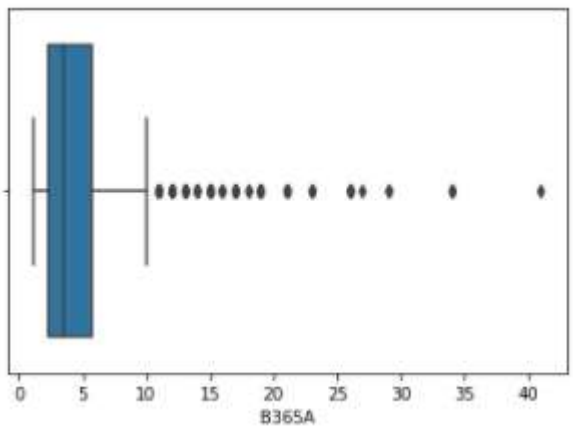

0
2
4
6
8
AY

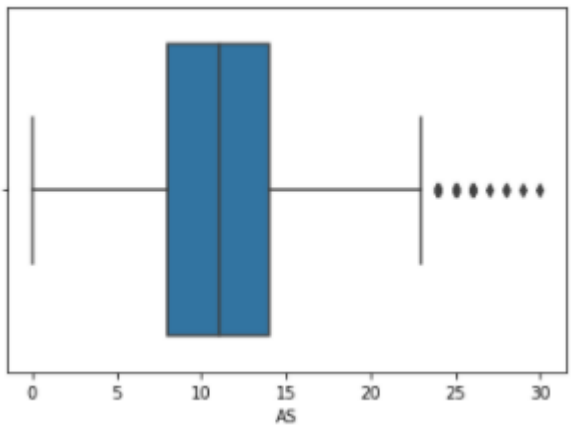

0
5
10
15
20
25
AF

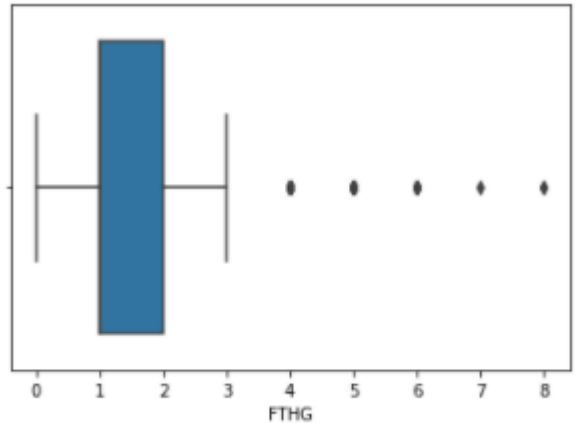

1.25
1.50
1.75
2.00
2.25
2.50
2.75
M2_5


1.5
2.0
2.5
3.0
3.5
4.0
4.5
L2_5


0
10
20
30
40
HS


0
5
10
15
20
25
30
35
40
B365A


0
5
10
15
20
25
30
AS


0
1
2
3
4
5
6
7
8
FTHG