INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Machine Learning na Previsão e Classificação de Dados no Futebol

TIAGO SOEIRA PINTO
Setembro de 2024

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Machine Learning na Previsão e Classificação de Dados no Futebol

Tiago Soeira Pinto

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

Orientador: Professora Ana Maria Madureira

Supervisor: Paulo Campos

Porto, setembro 2024

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 15 de setembro de 2024

Resumo

Com a globalização do desporto e a crescente importância atribuída à análise de dados recolhidos durante o jogo, a avaliação do desempenho individual e coletivo no futebol tornou-se uma prática comum. No entanto, esta análise frequentemente carece do uso de ferramentas baseadas em aprendizagem automática (ML).

Nesse sentido, esta dissertação explora o uso de técnicas de ML e pretende alcançar dois objetivos principais. O primeiro consiste na resolução de um problema de regressão, destinado a prever a eficácia dos remates do jogador com base em indicadores de desempenho como a sua posição, os minutos jogados, o total de remates e os golos marcados. O segundo visa resolver um problema de classificação, que classifique os níveis de desempenho do jogador com base na eficácia dos seus remates.

Para ambas as tarefas, foram examinados e comparados vários métodos de aprendizagem automática; a Árvore de Decisão e o *Gradient Boosting* foram considerados os mais eficazes. Estes modelos demonstraram resultados superiores na previsão da eficácia dos jogadores e na classificação do seu desempenho, oferecendo uma nova abordagem à análise do futebol que vai para além da análise estatística convencional.

Palavras-chave: Aprendizagem Automática, futebol, modelos de regressão, modelos de classificação, *GoalRatio*, desempenho do jogador, Árvore de Decisão, *Gradient Boosting*.

Abstract

With the globalization of sport and the growing importance attached to the analysis of data collected during the game, the evaluation of individual and collective performance in soccer has become common practice. However, this analysis often lacks the use of tools based on machine learning (ML).

This dissertation explores the use of ML techniques and aims to achieve two main objectives. The first is to solve a regression problem aimed at predicting the effectiveness of a player's shots based on performance indicators, such as position, minutes played, total shots and goals scored. The second aims to solve a classification problem, which classifies the player's performance levels based on the effectiveness of their shots.

For both tasks, various machine learning methods were examined and compared; Decision Tree and Gradient Boosting were found to be the most effective. These models demonstrated superior results in predicting players' effectiveness and classifying their performance, offering a new approach to soccer analysis that goes beyond conventional statistical analysis.

Keywords: Machine Learning, football, soccer, regression models, classification models, GoalRatio, player’s performance, Decision Tree, Gradient Boosting.

Agradecimentos

As primeiras palavras de agradecimento são inevitavelmente dedicadas aos meus pais. A escrita e defesa desta dissertação dita o culminar do meu percurso académico, o qual sem dúvida alguma não teria acontecido se não fosse o seu apoio e suporte. Foram muitos os desafios que este mestrado me colocou, desde logo conciliar a vida profissional com a vida académica e com a vida familiar. Sei que tudo fizeram para me apoiar nos momentos menos bons e que foram os primeiros a desejar que tudo me corresse pelo melhor, sem nunca faltar com uma palavra de incentivo, carinho ou amor.

Agradeço também à minha família, principalmente aos meus avós, que apesar da distância que se impôs e do contacto menos frequente, sempre estiveram no meu pensamento. Contribuíram imenso na formação da pessoa que sou hoje e estar-lhes-ei eternamente grato por isso.

Agradeço à minha namorada, a Ana, que surgiu a meio desta caminhada como uma lufada de ar fresco e que sem dúvida que a tornou menos difícil, sempre com as suas gargalhadas, a sua paciência, compreensão e excelentes conselhos.

Deixo um agradecimento muito especial à professora Ana Maria Madureira, por ter aceitado ser minha orientadora nesta dissertação, por toda a disponibilidade demonstrada e principalmente nos conselhos que prestou na revisão do documento.

Ao Paulo, agradeço pela supervisão ao longo da parte prática do projeto e pelo constante interesse em que eu evoluísse o máximo possível e produzisse um bom trabalho.

Por fim, agradeço aos meus amigos. Aos meus amigos engenheiros, a Bea, o Pedro, a Petra, o Rodrigo e o Zé, que sentiram na pele as mesmas dores que eu senti nestes anos de mestrado, mas que contribuíram para que custasse um pouco menos, quer dentro quer fora do DEI. Aos meus amigos *mediciners* (com menção especial ao Gonçalo), agradeço pelos bons momentos de diversão que vivemos juntos e que foram um bom escape para o *stress* sentido. Aos meus amigos de São João da Madeira, por serem casa e por serem onde estão as minhas raízes, onde tudo principiou.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

AD	Árvore de Decisão
CE	Critério de Exclusão
CI	Critério de Inclusão
CRISP-DM	*Cross Industry Standard Process for Data Mining*
DVC	*Data Version Control*
DIMEI	Dissertação
DP	*Dangerous Pass*
GB	*Gradient Boosting*
ISEP	Instituto Superior de Engenharia do Porto
kNN	*k-Nearest Neighbours*
KPI	*Key Performance Indices*
MAE	Erro Médio Absoluto
MEI	Mestrado em Engenharia Informática
ML	*Machine Learning*
MLP	*Multilayer Perceptron*
MSE	Erro Quadrático Médio
PREPD	Preparação para Dissertação
PRISMA	*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*
PSG	Paris Saint-Germain
QP	Questão de Pesquisa
QSL	*Qatar Stars League*
RF	*Random Forest*
RLin	Regressão Linear

RLog	Regressão Logística
RMSE	Raiz do Erro Quadrático Médio
RO	*Random Oversampling*
SO	*Synthetic Oversampling*
SPADL	*Soccer Player Action Description Language*
SRIJ	Serviço de Regulação e Inspeção de Jogos
SVM	*Support Vetor Machine*
VAEP	*Valuing activities by Estimating Probabilities*
xG	*Expected Goals*
WoS	*Web of Science*

1 Introdução

Neste capítulo é feita uma apresentação do contexto em que este documento se insere, é interpretado analítica, crítica e eticamente o problema e são identificados os objetivos a alcançar. É apresentada a metodologia de recolha de informação relativa ao estado da arte e, por fim, é descrita a estrutura e os tópicos abordados nos capítulos seguintes do documento.

1.1 Contexto

Este documento foi desenvolvido no âmbito da unidade curricular Dissertação (DIMEI), do Mestrado em Engenharia Informática (MEI), área de especialização de Engenharia de Software, lecionado no Instituto Superior de Engenharia do Porto (ISEP).

A ideia para o desenvolvimento desta dissertação surgiu em contexto empresarial, na ITSector - Sistemas de Informação, S.A.[1] (ITSector). O contexto de negócio que motivou a escrita deste documento está relacionado com um cliente cuja área de atuação é a gestão de competições futebolísticas. No ecossistema aplicacional desenvolvido para este cliente não existe nenhuma aplicação que potencialize os dados recolhidos em cada jogo das competições e que os analise estatisticamente.

Relativamente às restrições impostas para o desenvolvimento desta dissertação, as restrições financeiras estão diretamente ligadas com as tecnológicas, no sentido em que foi solicitado que apenas fossem utilizadas ferramentas e *frameworks open source*. No que toca às linguagens de programação não foram impostas restrições. Foi requerido que se fossem analisadas e avaliadas diferentes alternativas e que fosse escolhida a linguagem que melhor se adequasse e satisfizesse as necessidades e objetivos do projeto.

[1] https://www.itsector.pt/

1.2 Problema

A análise de dados em tempo real tornou-se numa das competências mais importantes numa conjuntura caracterizada pela geração de grandes volumes de dados. A capacidade de processar e analisar dados em tempo real pode fornecer informação e conhecimento valiosos para empresas e organizações em diferentes setores de atividade.

Tendo em conta a importância e dimensão do desporto à escala mundial e a crescente necessidade de informação para suportar o processo de planeamento e decisão, ou ainda na avaliação do desempenho desportivo individual, ou coletivo, diversos são os desafios que se colocam para manter as organizações desportivas competitivas.

Surge a necessidade de recorrer e desenvolver novas abordagens que agilizem e suportem esta área, recorrendo a ferramentas e técnicas mais adequadas para superar os desafios inerentes, permitindo a tomada de decisão mais eficiente e informada, a deteção de desvios e problemas em tempo útil assim como a melhorar a experiência do cliente.

Pretende-se o desenvolvimento de uma plataforma que suporte a análise de dados em tempo real, permitindo que as organizações futebolísticas monitorizem a dinâmica dos jogos e dos seus intervenientes e possam ajustar as suas estratégias na preparação e análise dos jogos em tempo real.

1.3 Objetivos

Tendo em conta o problema apresentado, esta dissertação foca-se no que é mais importante num jogo de futebol: o golo. Pretende-se recorrer a técnicas e modelos *Machine Learning* (ML) que sejam capazes de prever e calcular a eficácia dos remates dos jogadores e de classificar o seu desempenho com base em dados relativos aos seus remates e golos.

Os resultados obtidos no desenvolvimento desta dissertação podem simultaneamente ser o ponto de partida para o alargamento da integração de técnicas de ML a outros dados recolhidos durante um jogo de futebol, bem como ser posteriormente usados na implementação de um sistema de suporte à decisão utilizado pelos diversos profissionais da área.

1.4 Metodologia

Foram adotadas duas abordagens de recolha de informação. Numa primeira fase pretendeu-se contextualizar historicamente a análise estatística no futebol e apresentar algumas das estatísticas e suas representações mais comuns. Foi efetuada uma pesquisa na biblioteca B-On[2], mas esta revelou-se infrutífera, na medida em que não foram encontrados artigos publicados. Procedeu-se à pesquisa em páginas *Web* dedicadas à análise estatística e de análise de

[2] https://www.b-on.pt/

performance de jogador e foi possível encontrar e sistematizar informação, como é possível verificar nas secções 2.1 e 2.2.

Posteriormente, para a análise de alternativas de linguagens de programação a utilizar, foram pesquisados artigos científicos publicados em bibliotecas como a *IEEE Xplore*[3], a B-On, *Web of Science*[4] (WoS) ou recorrendo ao motor de busca *Google Scholar*[5]. No que toca à recolha de informação relativa à aplicação de técnicas de ML no contexto do futebol, seguiu-se a metodologia PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses)* (Page *et al.*, 2021), que será descrita na secção 1.4.1.

Relativamente à metodologia adotada para a preparação e avaliação do modelo ML a utilizar, foi seguido processo CRISP-DM (*Cross Industry Standard Process for Data Mining*). Este processo caracteriza-se por ser robusto e flexível, estruturando a abordagem de projetos de *data mining* em seis fases: compreensão do negócio, compreensão dos dados, preparação dos dados, modelagem, avaliação e implementação/*deployment* (Chapman *et al.*, 2000). Todos os passos deste processo, com exceção do último, serão descritos nos capítulos 3 e 4. Esta metodologia foi escolhida devido à sua natureza iterativa e adaptável, que permite revisões constantes e refinamentos em cada fase do processo, garantindo que o modelo de dados final está alinhado com os objetivos definidos e com as características específicas dos dados utilizados. Além disso, a sua abordagem estruturada e passo a passo facilita a documentação e a replicabilidade dos resultados, elementos essenciais para a validade científica e a transparência do projeto (Wirth and Hipp, 2000). Outra característica que suporta a escolha desta metodologia é a sua popularidade, quer ao nível da indústria, quer ao nível académico, evidenciada pelo seu uso em diversos estudos e pela sua capacidade em reduzir a complexidade e incertezas associadas a projetos de ciências de dados (Marban *et al.*, 2009).

A Figura 1 (Chapman *et al.*, 2000) é ilustrativa dos passos de CRIPS-DM.

Figura 1 - Passos do CRISP-DM (Chapman *et al.*, 2000)

[3] https://ieeexplore.ieee.org/Xplore/home.jsp
[4] https://www.webofscience.com/wos/
[5] https://scholar.google.com/

1.4.1 Metodologia PRISMA

A designação para a metodologia PRISMA provém da expressão: *Preferred Reporting Items for Systematic Reviews and Meta-Analyses*. Esta metodologia é amplamente reconhecida e adotada em revisões sistemáticas por proporcionar um conjunto claro de diretrizes que ajudam os investigadores a conduzir e relatá-las de forma rigorosa e padronizada (Moher *et al.*, 2009). Com o seu uso, pretende-se garantir a qualidade e a transparência no processo de recolha de dados e a inclusão de estudos relevantes e de alta qualidade, maximizando a reprodutibilidade e a credibilidade dos resultados obtidos (Liberati *et al.*, 2009).

Esta metodologia é composta por várias etapas fundamentais, como demonstra a Figura 2, incluindo a identificação de estudos relevantes através de pesquisas em base de dados, a seleção desses mesmo estudos com base em critérios de elegibilidade previamente definidos e a extração e síntese de dados obtidos da leitura dos estudos incluídos. Cada passo deste processo foi concebido para minimizar o viés e aumentar a fiabilidade dos resultados, facilitando uma abordagem estruturada da revisão da literatura (Page *et al.*, 2021). Adicionalmente, a utilização do PRISMA enfatiza a importância de documentar a lógica por trás de cada decisão tomada durante o processo de revisão, o que aumenta a transparência e a reprodutibilidade das revisões sistemáticas (Higgins *et al.*, 2019).

Como referido na secção anterior, o PRISMA foi utilizado na recolha de estudos que se enquadrassem no contexto da aplicação de técnicas de ML no futebol. Nas secções seguintes serão descritos os passos adotados em todo o processo.

1.4.1.1 Questões de Pesquisa

O procedimento inicial na utilização desta metodologia é a definição de questões de pesquisa (QP). Como se pretendia inferir sobre a utilização de modelos ML aplicados às estatísticas no futebol, foi elaborada a QP presente na Tabela 1, com vista a formular os termos de pesquisa nos passos seguintes.

Tabela 1 - Questões de Pesquisa

Questões de Pesquisa (QP)	
QP1	Quais são os algoritmos de ML mais eficazes na previsão de desempenho individual de jogadores de futebol com base em dados históricos?

1.4.1.2 Fontes de Informação

Determinar as fontes de informação a utilizar na revisão do estado da arte constitui o segundo passo do PRISMA. Uma pesquisa livre pela Internet seria demasiado abrangente, sendo necessário garantir a integridade dos dados, utilizando uma quantidade limitada de fonte de dados. Dessa forma, e tendo em conta a existência de bases de dados específicas para a publicação de estudos e artigos científicos, foram utilizadas as fontes de informação presentes na Tabela 2. Foram utilizados os mesmos operadores e consultas nas duas bases de dados, levando à duplicação de alguns resultados, posteriormente eliminados.

Tabela 2 - Fontes de Informação

Sigla	**Designação**	**Url**
WoS	Web of Science	https://www.webofscience.com/wos/
B-On	Biblioteca do Conhecimento Online	https://www.b-on.pt/

1.4.1.3 Termos de Pesquisa

Tendo em conta o âmbito e a questão de pesquisa da revisão do estado da arte, foram identificados dois domínios principais. Os termos de pesquisa consistem não só nestes próprios domínios como também na sua combinação, como é possível verificar na Tabela 3.

Tabela 3 - Domínios e Termos de Pesquisa

Domínios	**Termos de Pesquisa**
Machine Learning	(“Machine Learning”)
Football Statistics	(“Football Statistics”) or (“Soccer Statistics”)

Em virtude do tempo disponível para a recolha do conhecimento produzido nesta área e posterior sistematização neste documento, os termos de pesquisa foram aplicados ao nível do *abstract* nas duas bases de dados.

Depois de observada a vasta aplicabilidade e referências à utilização de técnicas ML em vários contextos que não o do futebol, ambos os domínios tiveram de ser cruzados na pesquisa a fim de direcionar o âmbito dos artigos encontrados para o objetivo da pesquisa. Todas as combinações de termos de pesquisa utilizadas e respetivos resultados são apresentados na Tabela 4.

O domínio *Soccer Statistics* foi incluído uma vez que em certos países o futebol é denominado de *soccer*, sendo por isso necessário garantir que estudos que mencionassem esse termo (ao

invés de *football*) seriam incluídos. Pela escassez de resultados quando intersetados os domínios de ML e *Football Statistics*/*Soccer Statistics*, a pesquisa foi alargada para incluir referências a ML num contexto mais alargado na modalidade. Sendo assim, foram analisados os resultados utilizando a última combinação apresentada na Tabela 4.

Tabela 4 - Combinação entre os Domínios e Respetivos Resultados

Domínios	**Resultados**	
	B-On	**WoS**
Machine Learning	381824	249484
Football Statistics	155	7
Soccer Statistics	43	2
Machine Learning OR Football Statistics	381834	249490
Machine Learning OR Soccer Statistics	381830	249486
Machine Learning AND Football Statistics	2	1
Machine Learning AND Soccer Statistics	0	0
Machine Learning AND Football	176	167
Machine Learning AND Soccer	115	156
Machine Learning AND (Football Statistics OR Soccer Statistics OR Football OR Soccer)	289	300

1.4.1.4 Avaliação da Qualidade dos Resultados: critérios de inclusão e exclusão

Para que fosse utilizada apenas a informação que melhor se adequasse à QP realizada, surgiu a necessidade de definir conjuntos de critérios para incluir e excluir dados desta revisão. Este método de definição de regras foi extremamente importante para não só minimizar o tempo de leitura de toda a informação obtida na consulta, mas também para obter um conjunto de dados mais fiável e adequado ao problema. Estes critérios podem estar relacionados com o ano de publicação da fonte, o tipo de fonte, a sua língua, se está direcionada para a área, entre outros. Os critérios de inclusão estão definidos na Tabela 5 e os critérios de exclusão estão representados na Tabela 6.

Tabela 5 - Critérios de Inclusão

Critérios de Inclusão (CI)	
CI1	A fonte pertence ao domínio do conhecimento da engenharia informática
CI2	A fonte descreve uma contribuição significativa para os domínios de estudo

Tabela 6 - Critérios de Exclusão

Critérios de Exclusão (CE)	
CE1	A fonte foi publicada há mais de 10 anos
CE2	A fonte não é um artigo ou um artigo de revisão
CE3	A fonte não está escrita em inglês
CE4	A fonte diz respeito a futebol americano em vez de futebol

1.4.1.5 Extração de Resultados

O processo de gestão dos dados obtidos nas bibliotecas começou por filtrar por data apenas os artigos publicados nos últimos 10 anos (desde 2014) para restringir a recolha de dados a investigações mais recentes. Pela constante mudança e evolução existe no que toca à aplicação de técnicas de ML, é importante aplicar o CE1 para obter resultados mais adequados para as necessidades atuais.

Foram encontrados alguns artigos que não se adequavam aos objetivos da pesquisa, pelo que foram acrescentados mais alguns filtros, tais como a necessidade de serem escritos em inglês e de serem identificados como artigo escrito no âmbito das Ciências da Computação. Com a aplicação destes critérios de exclusão, o número de resultados foi progressivamente reduzindo e ajustando-se à QP1.

A Figura 2 é ilustrativa dos passos seguidos neste processo e descritos nas subsecções anteriores, bem como evidencia o processo de seleção de resultados elegíveis para serem utilizados na revisão do estado da arte.

1.4 Metodologia

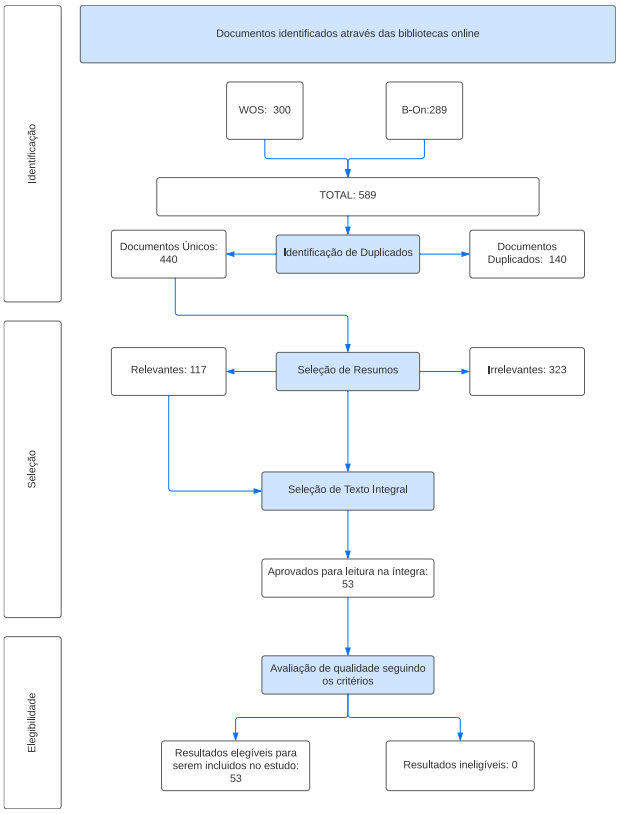

Figura 2 - Processo de Seleção de Resultados

Existem quatro etapas fundamentais neste processo descrito na Figura 2. A primeira consiste na identificação de registos duplicados, tendo sido por isso utilizado o *software* JabRef[6] para os eliminar.

Posteriormente, foram lidos os resumos dos artigos e eliminados aqueles que diziam respeito à utilização de técnicas ML no âmbito do futebol americano/futebol australiano, uma vez que não se enquadravam do âmbito da pesquisa. Restaram então 117 artigos considerados relevantes, que passaram à fase seguinte.

Os 117 artigos considerados relevantes foram ainda alvo de uma filtragem e foram removidos artigos que, apesar de se enquadrarem no âmbito do futebol, visavam outros aspetos como a melhoria da experiência no adepto nos estádios, otimização de infraestruturas, entre outros. Resultaram então 53 artigos que foram aprovados para serem lidos na íntegra.

Após a sua leitura, concluiu-se que estavam elegíveis na sua totalidade para serem incluídos na revisão do estado da arte.

[6] https://www.jabref.org/

1.5 Estrutura

Este documento está estruturado em cinco capítulos:

- Introdução, em que é descrito o propósito da escrita deste documento, apresentando o seu contexto, o problema, os objetivos propostos a alcançar e a metodologia utilizada para os atingir.
- Estado da arte, capítulo onde são contextualizadas a análise de dados recolhidos em jogos de futebol e a utilização de técnicas de ML nesta modalidade.
- Design e Desenvolvimento, capítulo onde serão descritas as decisões de design tomadas, bem como o processo de desenvolvimento seguido.
- Avaliação dos Modelos, onde serão comparados os desempenhos dos diferentes modelos avaliados para atingir os objetivos propostos.
- Conclusão, que consiste na apresentação das conclusões retiradas do trabalho, dos objetivos alcançados e quais devem ser os próximos passos a seguir.

2 Revisão do Estado da Arte

Neste capítulo é realizada a revisão do conhecimento produzido nas áreas relacionadas com o trabalho, tais como: dados estatísticos relacionados com o futebol e a sua representação; ML e os seus algoritmos; aplicação de técnicas de ML no futebol.

2.1 Perspetiva Histórica da Análise de Dados no Futebol

2.1.1 1950 – 1992

Os primeiros trabalhos identificados na área da análise de dados relativos ao desempenho coletivo e individual no futebol remontam ao ano de 1950, quando Charles Reep, um antigo comandante da Força Aérea Real Britânica, decidiu registar manualmente as ações da equipa da sua terra natal, o Swindon Town. Descontente com o futebol que a equipa praticava, Reep começou a tirar notas durante os jogos no sentido de perceber como é que o desempenho da equipa poderia melhorar, estabelecendo-se como o primeiro analista de performance no futebol profissional (Arastey, 2019).

Durante as décadas de 1950 e 1960, Reep desenvolveu a teoria de que a melhor forma de alcançar o sucesso no futebol seria chegar à baliza adversária com o menor número de passes possíveis. Alicerçou a sua teoria nos mais de 500 jogos que estudou e em todas as suas anotações durante esse período, que o tornaram convicto que a sua análise deu origem à melhor fórmula para o sucesso.

Reep tornou-se um modelo e um dos principais influenciadores do futebol praticado em Inglaterra. Nas duas décadas seguintes, procurou fundamentar a sua teoria recorrendo a conhecimentos no campo da probabilidade e publicou as suas notas e métodos analíticos em revistas inglesas, partilhando-as com o público geral.

Porém, apesar da sua popularidade, a teoria de Reep não foi imune à crítica e o inglês foi várias vezes acusado de se basear em *datasets* em que os intervenientes eram favoráveis às suas premissas e de interpretar os dados de forma errónea. Foi acusado de ignorar fatores como os diferentes níveis competitivos da modalidade e o acrescido cansaço do adversário em momentos que se encontra sem a posse da bola. Foi precisamente o primeiro ponto que levou a que a sua teoria fosse desacreditada na década de 90, quando a seleção inglesa fracassou mais uma vez numa competição internacional e alguns dos seus principais seguidores foram afastados de cargos de relevo nas instituições desportivas inglesas.

Contudo, Charles Reep deixa no futebol o legado de ter sido o pioneiro na análise estatística e da performance do atleta. Pese embora os seus métodos arcaicos e algumas más interpretações, deixou contributos e padrões, principalmente ao nível da recolha de dados, que se revelaram importantes para as metodologias utilizadas a partir de 1990.

2.1.2 1992 – Presente

O ano de 1992 marca o início do que se considera ser a "era moderna" do futebol. Algumas leis do jogo mudaram e as competições mais importantes foram reformuladas, destacando-se a fundação da *Premier League*[7]. Este último ponto, associado às inovações tecnológicas da época (principalmente a gravação de vídeo) trouxe consigo mais capital investido e, consequentemente, mais inovação ao desporto.

Em 1996 surgiu a Opta Consulting (agora StatsPerform (Stats Perform, 2022)). Esta empresa, financiada por um dos patrocinadores da *Premier League*, começou a recolher dados quantitativos relativos a diversos eventos decorrentes de um jogo futebol. Esta recolha foi facilitada pelas transmissões dos jogos e proporcionou a criação de diversos rankings, posteriormente fornecidos aos clubes. Ainda que básicos, a existência destes dados revolucionou por completo a abordagem ao jogo e novos parâmetros de análise eram introduzidos com regularidade (xfb Analytics, 2022).

Em 2001, e motivado pela crescente influência da análise de dados dos jogos, Alex Ferguson, treinador do Manchester United, tomou a inesperada decisão de transferir o seu atleta Jaap Stam para a Lazio. Ao examinar os dados de desempenho do jogador que tinha aos seu dispor, Ferguson constatou que a frequência de desarmes de Stam diminuiu, o que o levou a concluir que o jogador, aos vinte e nove anos, estava a sofrer uma deterioração do seu desempenho. Consequentemente, esta decisão constituiu um marco significativo, por ter sido a primeira transferência de um jogador impulsionada principalmente por análise e conhecimentos estatísticos. No entanto, mais tarde, foi revelado que a dependência de Ferguson de certos números se revelou uma avaliação errónea, sublinhando os perigos associados a uma dependência excessiva dos dados.

[7] https://www.premierleague.com/

O constante avanço e progresso da tecnologia assumiu preponderância no registo de dados relacionados com o jogo e permitiu que este deixasse de ser exclusivamente manual. Ainda que atualmente exista uma mistura entre ambas as metodologias de recolha de dados, o recurso à tecnologia permite não só avaliar diferentes âmbitos, como por exemplo o desempenho físico do atleta, como também permite o desenvolvimento de diversos de sistemas de apoio à decisão, com base nesses dados e posterior análise estatística.

2.2 Principais Indicadores de Desempenho no Futebol

A recolha de dados relativos ao desempenho durante um jogo de futebol é cada vez mais completa e mais abrangente a diversos níveis do jogo. Se nos anos imediatamente após o seu surgimento, esses dados abrangiam poucas métricas e eram maioritariamente dedicados aos clubes e aos seus treinadores e analistas, a proliferação de dados pormenorizados que fornecem informações sobre as inúmeras ocorrências de um jogo de futebol revolucionou os métodos utilizados para analisar e examinar o futebol, tanto nos próprios clubes, como nos meios de comunicação social.

O crescente aumento da granularidade de dados analisados também teve impacto no crescimento no setor das apostas desportivas, que se tem revelado um dos maiores público-alvo da análise estatística que a recolha destes dados proporciona. Segundo o relatório publicado pelo Serviço de Regulação e Inspeção de Jogos (SRIJ) (SRIJ, 2023), o futebol representou 68,4% do total das apostas efetuadas.

Os dados a ser analisados dependem do contexto em que a análise se insere, que pode inferir sobre uma equipa e o seu desempenho como um todo, ou ser mais pormenorizada e focar-se no indivíduo/jogador.

2.2.1 Dados Coletivos

No que diz respeito à análise das equipas, segundo (FutbolLab, 2023), os dados mais importantes são: golos marcados; remates na direção da baliza; número de faltas cometidas; passes completados; percentagem de tempo na posse da bola.

Em 2018 foi introduzida uma métrica que ganhou preponderância na análise das equipas e denomina-se xG (golos esperados, do inglês *expected goals*). Os principais fatores tradicionalmente incluídos na maioria dos modelos de cálculo do xG são: distância para a baliza, ângulo para a baliza, parte do corpo utilizada para o remate e o tipo de assistência ou ação anterior (como bola cruzada, cruzamento, bola parada, drible, etc.). O modelo xG atribui um valor de 0 a 1 a cada remate, com base em dados históricos de remates com características comparáveis. Este valor representa a probabilidade de o remate resultar em golo (Statsbomb, 2018).

No entanto, por se focar maioritariamente na última ação antes do golo, ou seja, no remate, constatou-se que o cálculo do valor de xG ignorava muitos momentos do jogo que também têm impacto na probabilidade de um remate resultar em golo. Foi então desenvolvido um modelo, recorrendo a dados históricos, denominado *Non-Shot Based xG* (Infogol, 2018). Este modelo vai além da simples medição do sucesso de um remate final e mede, em vez disso, a forma como as ações de cada jogador contribuem ou impedem os processos criativos e defensivos do seu clube.

A comparação entre as equipas também constitui uma boa forma de analisar o seu desempenho. A Figura 3 (XValue, 2024) é um exemplo representativo disso mesmo. Para além de apresentar alguns dos dados acima mencionados, é possível verificar a existência de *rankings* relativos à "percentagem de acerto do remate" e à "percentagem de remates que resultam em golo".

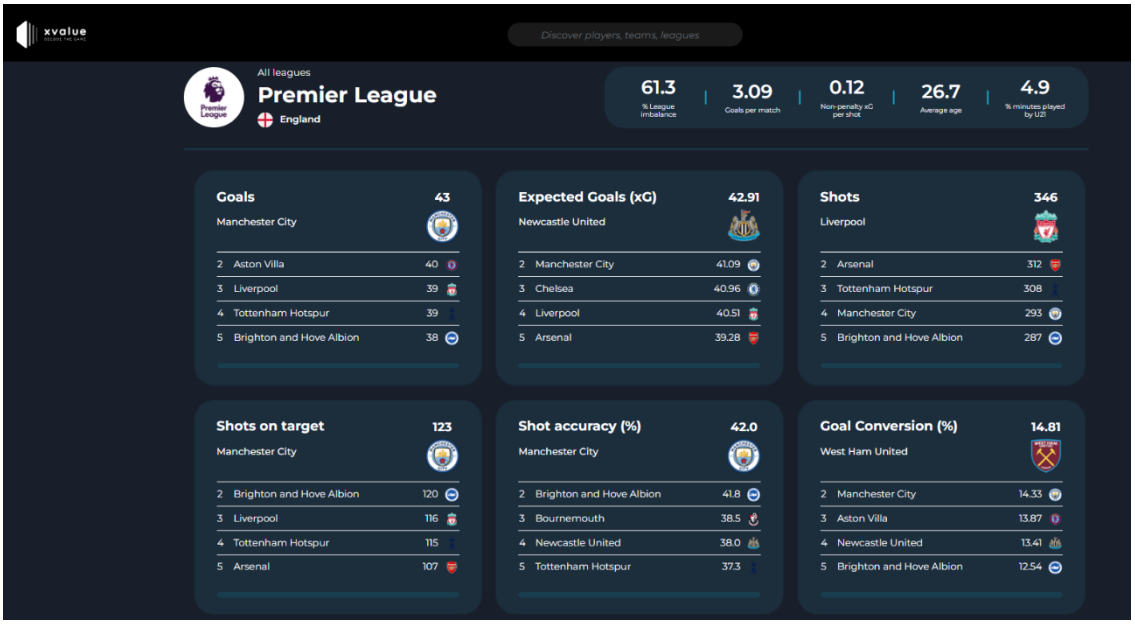

Figura 3 - Ranking comparativo de dados estatísticos de equipas (XValue, 2024)

O conceito de *Key Performance Indices* (KPI) é apresentado em (Soccermatics, 2022b) como um conceito segundo o qual as equipas deveriam guiar a sua cultura desportiva e empenhar-se em cultivá-lo. Para descobrir componentes do estilo de jogo da equipa que sejam mensuráveis e possam ser disseminados por todo o clube, estes KPI devem ser estabelecidos pela equipa técnica, em colaboração com os cientistas de dados e a administração do clube. No exemplo estudado, para além do xG, foram introduzidas métricas como o "número de chegadas ao último terço do terreno" e as "transições".

2.2.2 Dados Individuais

As métricas definidas para a recolha dos dados individuais durante um jogo de futebol variam, tipicamente, tendo em conta a posição do jogador (DataMB, 2023). À semelhança dos dados coletivos, é habitual os dados de cada jogador serem comparados, a fim de se tentar inferir aquele que possui melhor desempenho.

As Figura 4, Figura 5 e Figura 6 (DataMB, 2023) são exemplificativas da representação e comparação dos dados de jogadores. Constata-se que os parâmetros que são avaliados são diferentes nas várias posições, em função das principais características de cada uma e das principais ações expectáveis que um jogador naquela posição execute.

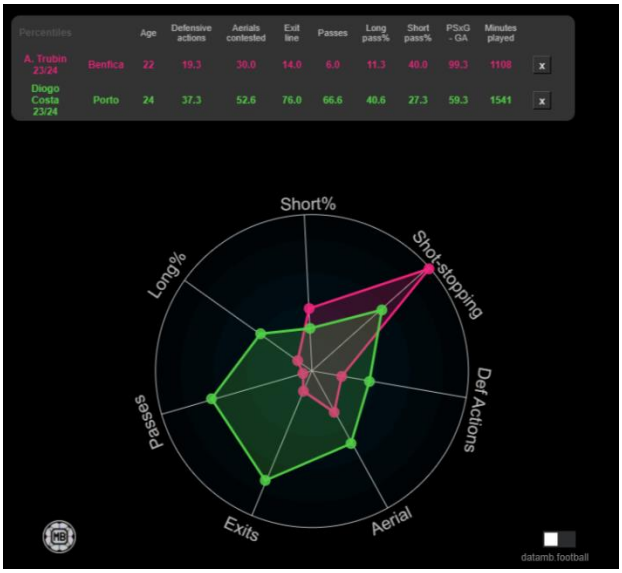

Figura 4 - Comparação de estatísticas de Guarda-Redes (DataMB,2023)

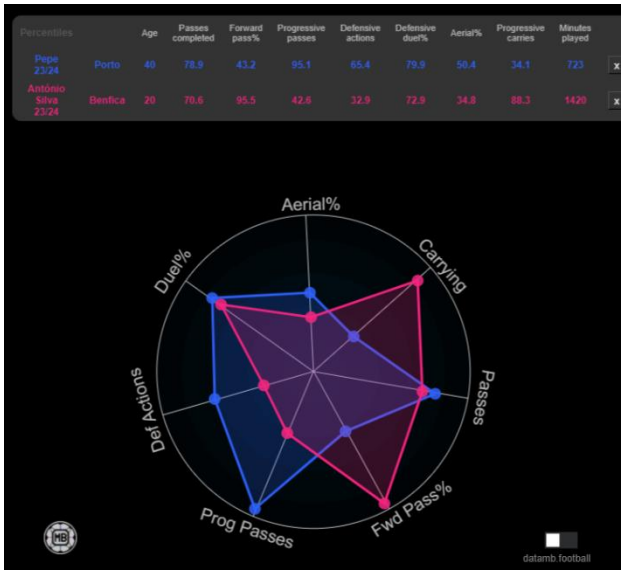

Figura 5 - Comparação de estatísticas de Defesas (DataMB,2023)

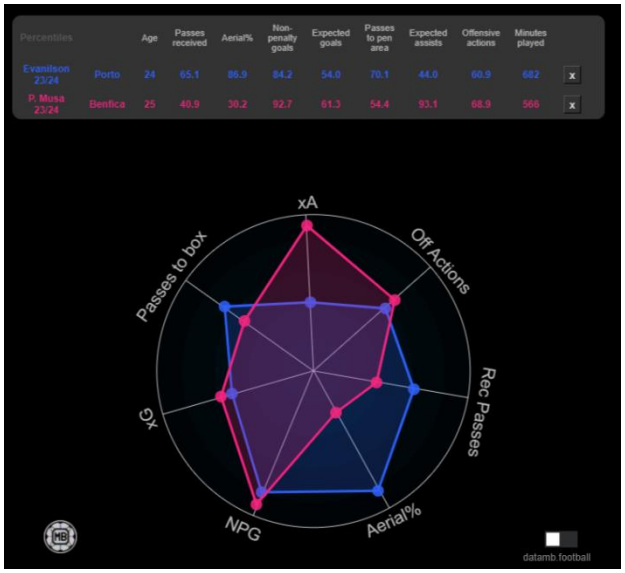

Figura 6 - Comparação de estatísticas de Avançados (DataMB,2023)

As diferentes métricas utilizadas para avaliar o desempenho do atleta devem ser tidas em conta em conjunto com o contexto de onde foram retiradas, como referido em (Soccermatics, 2022a). Por exemplo, consideremos os passes completados por um atleta. A percentagem de passes bem-sucedidos pode ser vista como um indicador da capacidade de manter a posse da bola. No entanto, também pode avaliar um estilo de jogo da equipa orientado para a posse de bola, caracterizado pela execução de passes simples. Um exemplo ilustrativo desta afirmação foi documentado num relatório realizado pelo Transfermarkt[8], que visava quais os jogadores com as maiores taxas de conclusão de passes (Transfermarkt, 2020). Descobriu-se que os jogadores

[8] https://www.transfermarkt.pt/

do clube Paris Saint-Germain (PSG), Thiago Silva e Presnel Kimpembe, têm as melhores taxas de conclusão de passes nas principais ligas europeias, com percentagens de 95,5% e 95,0%, respetivamente. Estes dois indivíduos desempenharam a função de defesas centrais numa equipa que tinha um controlo significativo da bola durante a maior parte do jogo. Esta estatística indica apenas que eles passavam repetidamente a bola um para o outro, não avaliando o real impacto que essa percentagem de acerto de passe tinha no desempenho da equipa.

2.3 Aprendizagem Automática (ML)

ML é um campo da inteligência artificial que se centra no desenvolvimento de algoritmos que permitem que computadores aprendam a partir de dados e melhorem constantemente o seu desempenho ao longo do tempo em tarefas específicas, sem serem explicitamente programados para tal (Mitchell, 1997). Essa abordagem baseia-se em técnicas estatísticas e matemáticas que capacitam os modelos a reconhecer padrões, fazer previsões e tomar decisões com base em grandes quantidades de dados. Ao contrário dos métodos tradicionais de programação, onde a especificação das regras específicas requer codificação manual, ML permite que os sistemas se adaptem e evoluam automaticamente à medida que são expostos a novos dados (Mitchell, 1997).

Os modelos ML podem ser distinguidos entre duas categorias: modelos de aprendizagem supervisionada e modelos de aprendizagem não supervisionada. Cada uma destas categorias apresenta diferentes propósitos e métodos, sendo que a utilização de cada uma se baseia na natureza dos dados disponíveis e nos objetivos definidos (Goodfellow *et al.*, 2016).

Os modelos ML de aprendizagem supervisionada têm como principal característica serem projetados para aprenderem a partir de um conjunto de dados tipicamente rotulados, que são usados para obter os resultados pretendidos. Dividem-se principalmente em dois tipos de problema:

- Classificação: É usado quando o objetivo é categorizar novas observações em classes pré-definidas, com base nas suas características. São exemplos da utilização deste tipo de algoritmo a deteção de spam em emails (Sahami *et al.*, 1998) e o reconhecimento de imagens (Krizhevsky, Sutskever and Hinton, 2012).
- Regressão: Tem como objetivo a previsão de determinado valor com base nos dados de entrada, quer seja, por exemplo, a previsão de preços de imóveis ou a previsão da procura de determinado artigo de *stock* (Hastie, Tibshirani and Friedman, 2009).

A Figura 7 (Terra, 2014) apresenta uma comparação entre os dois algoritmos acabados de descrever.

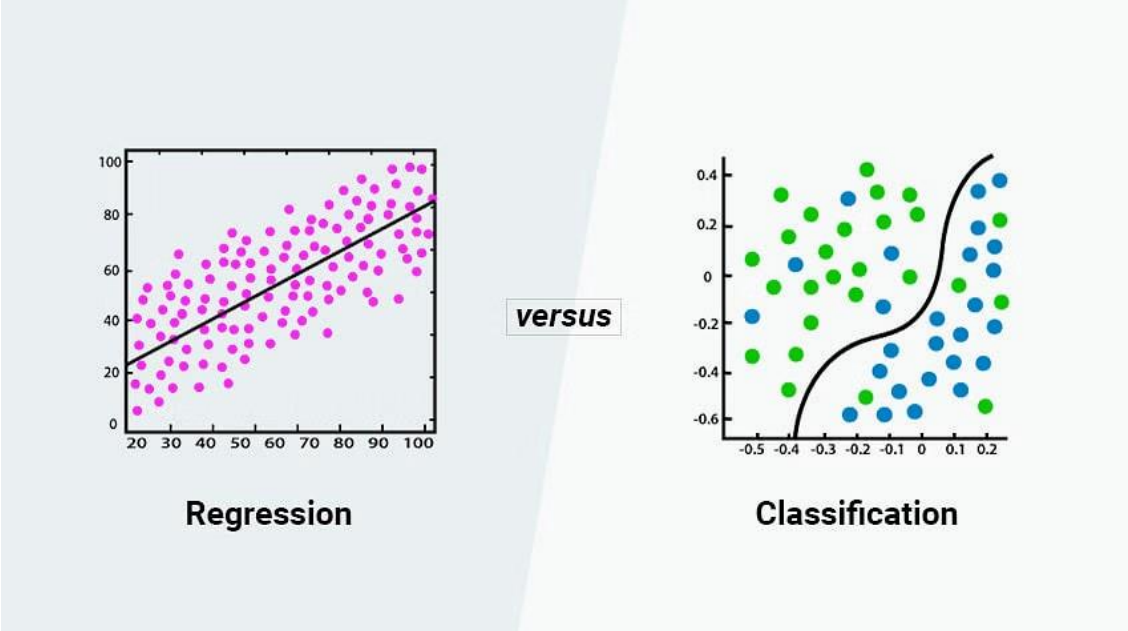

Figura 7 - Comparação entre Regressão e Classificação (Terra, 2014)

Por outro lado, os modelos ML de aprendizagem não supervisionada são utilizados em dados que possuem rótulos pré-definidos e o objetivo passa por padronizar e/ou identificar estruturas subjacentes dentro do conjunto de dados. A sua principal área de atuação é na análise exploratória de dados e possui os seguintes exemplos:

- *Clustering*: É um método que agrupa os dados em *clusters*, com base em características similares. É comum a sua utilização na segmentação de clientes de um determinado tipo de produto, permitindo a identificação de diferentes perfis sem a existência de rótulos prévios (Jain, Murty and Flynn, 1999).
- Redução de dimensionalidade: O objetivo deste método é reduzir o número de variáveis de um determinado conjunto de dados, mas mantendo o máximo de informação possível, como é observado em casos de compressão de imagens ou visualização de dados de elevada dimensão (Maaten, Postma and Herik, 2008).

A Figura 8 (Ezugwu *et al.*, 2021) exemplifica o resultado da utilização de algoritmos de *clustering*.

Tendo em conta os objetivos descritos na secção 1.3 e pelo exposto nos parágrafos anteriores, a escolha do método de ML recai na utilização de algoritmos de aprendizagem supervisionada. Na secção seguinte serão apresentados e descritos exemplos desses algoritmos.

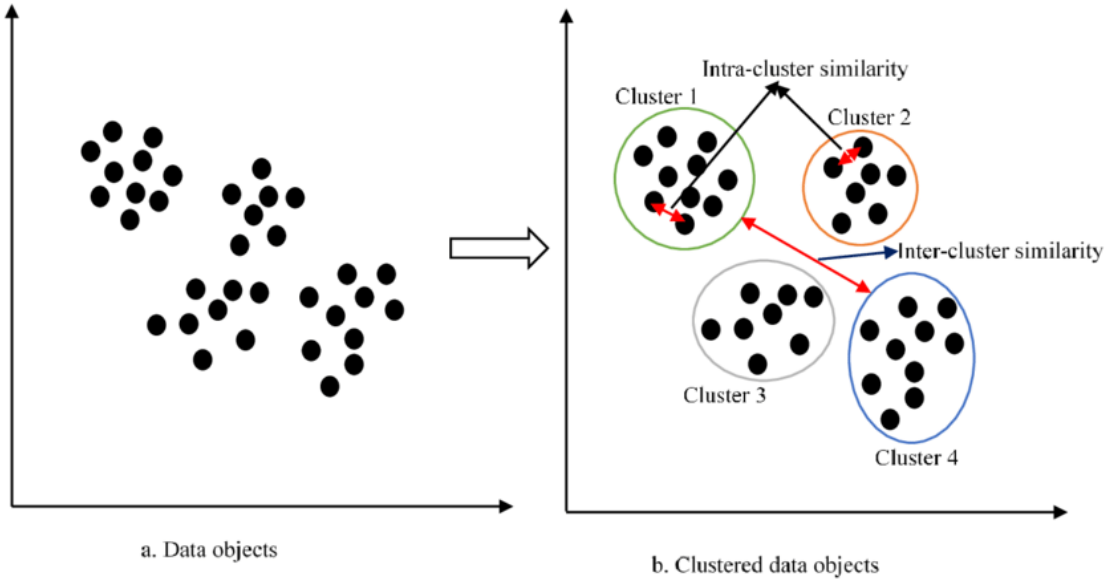

Figura 8 - Utilização de algoritmos de *clustering* (Ezugwu *et al.*, 2021)

2.3.1 Aprendizagem Supervisionada – Regressão

No conjunto dos algoritmos de aprendizagem supervisionada, os algoritmos de regressão são comummente utilizados quando o objetivo é efetuar previsões tendo em conta um registo prévio de observações. A sua utilização permite modelar relações entre variáveis dependentes e independentes, tornando-se úteis não só em tarefas de previsão bem como na análise de tendências (Hastie *et al.*, 2006).

Entre os diversos tipos de algoritmos de regressão, destacam-se pela sua eficiência e facilidade de interpretação (mesmo quando os contextos são bastante diferentes entre si) os seguintes (Hastie, Tibshirani and Friedman, 2009):

- Regressão Linear (RLin)
- Regressão Logística (RLog)
- Árvore de Decisão (AD)
- *Support Vetor Machine* (SVM)
- *K-Nearest Neighbours (kNN)*

A Tabela 7 apresenta as vantagens e limitações de cada um dos algoritmos bem como o contexto tipicamente mais favorável à utilização de cada um, permitindo compreender a sua influência e aplicabilidade no contexto da aprendizagem supervisionada (Hastie *et al.*, 2006), (Breiman *et al.*, 1984), (Cortes, Vapnik and Saitta, 2020), (Cristianini and Shawe-Taylor, 2000), (Duda, Hart and Stork, 2000).

Analisando os dados apresentados na Tabela 7, realça-se uma vantagem comum à maioria dos algoritmos: o facto de serem de fácil implementação os seus resultados serem facilmente interpretados. É um ponto importante que justifica a sua utilização na secção 3.2.3. As suas limitações possuem igual importância e estão diretamente associadas ao *datasets* a utilizar para treinar e avaliar o modelo, sendo que serão apresentadas técnicas para minimizar o seu impacto. No que toca ao contexto de utilização dos modelos, constituem exemplos onde a implementação de determinados modelos mostrou ser adequada.

Tabela 7 – Comparação de algoritmos de regressão

Algoritmo	**Vantagens**	**Limitações**	**Contexto Ideal**
AD	• Fácil de visualizar e interpretar; • Pode lidar com dados categóricos e numéricos; • Não requer normalização dos dados.	• Propenso a *overfitting*; • Sensível a pequenas variações nos dados; • Pode criar modelos complexos e difíceis de interpretar, caso não seja devidamente ajustada.	Útil para problemas onde a interpretabilidade (a capacidade de ser compreensível para o ser humano) é crucial, como decisões relacionadas com crédito ou triagem médica.
kNN	• Simples de entender e implementar; • Eficaz com grandes conjuntos de dados; • Não requer treino do modelo.	• Sensível a ruído nos dados; • Requer grande quantidade de memória; • Desempenho reduzido em casos de alta dimensionalidade (*curse of dimensionality*).	Adequado para problemas onde a interpretação não é prioritária e o volume de dados é grande, como recomendação de produtos.
RLin	• Simples de interpretar e implementar; • Exige menos recursos computacionais; • Boa performance com dados lineares.	• Desempenho baixo em dados com relações não lineares; • Sensível a *outliers*; • Assume independência entre variáveis preditivas	Ideal para problemas de regressão com relações lineares entre variáveis, especialmente em análise financeira e económica.

RLog	• Fácil de interpretar; • Eficaz para problemas de classificação binária; • Fornece probabilidades de previsão.	• Limitada a problemas de classificação binária ou multinomial; • Assume independência das variáveis explicativas.	Adequada para classificação binária, como deteção de fraude, e problemas na área da medicina, como como diagnóstico e deteção de doenças.
SVM	• Eficaz em espaços de alta dimensionalidade; • Robusto ao *overfitting*, especialmente em classificações; • Pode ser usado para classificação e regressão.	• Custo computacional elevado para grandes datasets; • Difícil de interpretar; • Performance limitada em datasets ruidosos.	Apresenta bom desempenho em problemas de classificação complexa com limites claros, como reconhecimento de imagem e bioinformática.

2.3.2 Aprendizagem Supervisionada – Classificação

Os algoritmos de classificação também possuem importância dentro do campo da aprendizagem supervisionada, tendo sido desenvolvidos para dividir os dados em classes ou grupos tendo em conta a interpretação de determinada característica. São algoritmos capazes de categorizar quaisquer novos dados de entrada, permitindo a constante relação entre eles e a classe mais apropriada onde se inserem (Kotsiantis, 2007).

Dos algoritmos de classificação analisados, três apresentam a mesma base (mas resultados diferentes) dos algoritmos apresentados em 2.3.1: AD, kNN e SVM. Adicionalmente, foram também analisados os algoritmos *Random Forest* (RF) e *Gradient Boosting* (GB) (Hastie, Tibshirani and Friedman, 2009).

- O algoritmo RF caracteriza-se por construir múltiplas árvores de decisão na fase de treinos do modelo, com o objetivo de aumentar a performance do modelo e controlar o seu *overfitting*. Cada árvore é independente das restantes e apresenta variedade de dados, uma vez que lhe é atribuída uma amostra aleatória do *dataset* de treino.
- Em relação ao algoritmo GB, tem em comum com o algoritmo anterior o facto de utilizar árvores de decisão, porém distingue-se pela abordagem que adota. Segue um processo iterativo que gera uma nova árvore com a correção dos erros cometidos pela árvore anterior. Esta abordagem permite uma constante otimização do modelo e a obtenção de elevados valores de precisão (Chen and Guestrin, 2016).

A Tabela 8 sintetiza as vantagens, limitações e o contexto ideal para a utilização destes dois algoritmo (Hastie, Tibshirani and Friedman, 2009)(Chen and Guestrin, 2016). Verifica-se a diferença de comportamento dos algoritmos no que toca ao *overfitting*, constituindo uma limitação do modelo GB caso não seja devidamente regularizado. Se, por um lado, o algoritmo RF torna-se mais complexo e difícil de interpretar quando comparado com uma única árvore de decisão, por outro consegue contrariar essa limitação apresentando robustez em *datasets* com dados ruidosos e/ou quando identificado o fenómeno de *missing data*.

Finalizada a análise dos algoritmos, é importante averiguar sobre qual a tecnologia a utilizar na sua implementação. A próxima secção é destinada a essa comparação.

Tabela 8 – Comparação de algoritmos de classificação

Algoritmo	Vantagens	Limitações	Contexto Ideal
RF	• Em comparação com AD simples, o seu *overfitting* é mais reduzido; • Robusto contra dados ruidosos; • Possibilita o cálculo da importância das features.	• A sua interpretação é mais complexa em relação a AD; • Carga computacional mais elevada, com muitas árvores e características.	Problemas onde se priorizem a robustez e a precisão, em detrimento da interpretabilidade.
GB	• Excelente desempenho preditivo; • Flexível e ajustável; • Captura relações complexas e não-lineares.	• Mais propenso a overfitting; • Processo de treino mais lento.	Cenários com dados complexos e features relacionadas de forma não linear.

2.4 Tecnologias e Ferramentas para o Desenvolvimento de ML

Tendo em conta a rápida evolução no domínio de ML, a escolha das tecnologias de implementação desempenha um papel crucial no desenvolvimento e implementação de modelos. Nesta secção são exploradas as várias ferramentas e linguagens de programação que permitem aos investigadores e programadores criar soluções eficazes de aprendizagem automática.

2.4.1 Linguagens de Programação

A seleção da linguagem de programação tem influência direta na eficiência do modelo desenvolvido, na facilidade de *deployment* e ainda na eficácia apresentada pela solução. Na

lista de linguagens de programação mais populares no contexto de ML, destacam-se: Python, R, Java e Julia (Géron, 2017).

O Python assume-se como uma escolha popular quer para principiantes, quer para profissionais já experientes neste contexto, apresentando como principais vantagens o reconhecimento generalizado da sua simplicidade, as vastas bibliotecas que possui e uma comunidade de suporte bastante ativa (van Rossum and Drake, 2009). O recorrente *feedback* de utilizadores e programadores contribuem continuamente para a documentação, suporte e melhoria das bibliotecas. Para além de providenciar atualizações constantes, esse papel ativo facilita a resolução de problemas e a integração de novas funcionalidades, tornando o Python uma escolha robusta e eficiente em projetos de ML em larga escala.

A linguagem R é uma das mais utilizadas pelos programadores em projetos cujo foco é a análise estatística e a visualização de dados. Possui variados pacotes (*packages*) especialmente desenvolvidos para modelação e análise estatística avançada e representação gráfica, permitindo a implementação de análise de dados exploratória e a construção de modelos preditivos fiáveis (Kuhn, 2008). Devido à sua robusta integração com a computação estatística, o R tornou-se uma ferramenta vital para analistas e investigadores cujo trabalho exige um elevado grau de precisão e detalhe (Ihaka and Gentleman, 1996).

Por seu lado, a sua capacidade de integração em sistemas de larga escala e o seu desempenho eficiente fazem com que o Java seja uma linguagem com preponderância em soluções desenvolvidas em ambientes empresariais (Gosling *et al.*, 2005). A combinação entre o seu paradigma de programação orientada a objetos e a utilização de *big data frameworks* (Apache Hadoop[9] and Apache Spark[10]), tornam o Java adequado para a implementação de modelos ML em ambientes de computação distribuídos com elevado desempenho (Dean and Ghemawat, 2008).

Por fim, a mais recente entrada na lista apresentada. Julia tem vindo a destacar-se pelas suas capacidades de alto desempenho no que toca à computação numérica (alcançando valores relativamente próximos à velocidade de execução registada na linguagem C) e facilidade com que permite o desenvolvimento de código curto e conciso, mas eficiente (Power *et al.*, 2017).

2.4.1.1 Escolha da Linguagem de Programação

Ambos os problemas de regressão e classificação apresentados não têm como principal objetivo a representação gráfica dos seus resultados, mas sim o desempenho eficaz dos modelos desenvolvidos e a facilidade de suporte e ajuste aos modelos uma vez implementados em ambiente de produção. Pelo exposto, pelas características das diferentes linguagens de programação apresentadas e pelo conhecimento do autor desta dissertação nas linguagens mencionadas, foi escolhido o Python como linguagem de programação de desenvolvimento dos

[9] https://hadoop.apache.org/
[10] https://spark.apache.org/

modelos. A subsecção seguinte apresenta algumas bibliotecas e *frameworks* existentes no Python.

2.4.2 Bibliotecas e *Frameworks*

Tendo em conta a escolha do Python como linguagem de programação a utilizar, é importante analisar as bibliotecas e *frameworks* existentes no sentido de perceber qual a que possui as características mais adequadas ao contexto do problema. A característica que as diferentes alternativas possuem em comum é o facto de possuírem funções *built-in*, ferramentas e processos implementados que aceleram e facilitam o desenvolvimento do modelo e o seu *deployment*.

As *frameworks* TensorFlow [11] e PyTorch [12] estão entre as mais populares *frameworks* de aprendizagem profunda ("*deep learning*"), ambas caracterizando-se pela sua flexibilidade e escalabilidade na conceção de redes neuronais complexas (Abadi *et al.*, 2016)(Paszke *et al.*, 2019).

- TensorFlow foi desenvolvida pela Google e é-lhe reconhecida como mais-valia o facto de se integrar com API's de alto nível, como, por exemplo, a API "Keras". Esta integração simplifica a criação e o treino das redes neuronais e confere à *framework* robustez e desempenho adequados a ambientes de produção (Keras-Team, 2017).
- PyTorch foi desenvolvida pelo Facebook e ganhou popularidade por se destacar no dinamismo ao nível da computação gráfica, que permite mais flexibilidade na construção do modelo e mais facilidade no *debug* do código (Paszke *et al.*, 2019).

Em complementaridade com as *frameworks* utilizadas em aprendizagem profunda, bibliotecas como Scikit-Learn[13] e XGBoost[14] tornaram-se escolhas comuns no âmbito execução dos mais variados e tradicionais algoritmos de ML.

- A biblioteca Scikit-Learn foi desenvolvida com base em outras três bibliotecas: NumPy, SciPy, and Matplotlib. É valorizada por aliar a sua simplicidade à posse de ferramentas eficientes nos campos de análise de dados e de mineração de dados (*"data mining*") e ainda ao facto de apresentar bom desempenho em contacto com aplicações complexas. Por tudo isto, torna-se bastante acessível a quem contacta pela primeira vez com ML (Pedregosa *et al.*, 2011).
- A biblioteca XGBoost caracteriza-se pela sua escalabilidade e pela eficiência no que toca ao método *gradient boosting*. A sua adoção generalizou-se no desenvolvimento de modelos ML pela seu desempenho com dados estruturados, tipicamente em tabelas e bastante comum em problemas de classificação, providenciando ao

[11] https://www.tensorflow.org/?hl=pt
[12] https://pytorch.org/
[13] https://scikit-learn.org/stable/
[14] https://xgboost.readthedocs.io/en/stable/

desenvolvedor/programador implementações de modelos bastante otimizadas (Chen and Guestrin, 2016).

2.4.2.1 Escolha das Bibliotecas e *Frameworks*

Como descrito anteriormente, ambas as *frameworks* apresentadas são indicadas no desenvolvimento de redes neuronais profundas ou de modelos com maior grau de complexidade, que envolvam o processamento de grandes volumes de dados. Como referido na secção 2.3, esta dissertação foca-se em algoritmos de aprendizagem supervisionada, não possuindo complexidade que exija a utilização quer de TensorFlow quer de Pytorch. A própria dimensão do volume de dados, que será detalhado no capítulo 3, também não constitui um entrave à utilização das bibliotecas apresentadas. Pelo exposto, pelas suas características e pela variedade de algoritmos que apresenta, será utilizada a biblioteca Skicit-Learn.

2.5 Trabalhos Prévios

A convergência da tecnologia e do desporto sofreu uma transformação notável, particularmente no domínio do futebol, onde é cada vez mais recorrente o recurso à tecnologia em diversos sistemas de informação e no auxílio da tomada de decisão. A adoção de técnicas de ML no domínio do futebol, permitiu a integração de algoritmos de aprendizagem automática, o que constitui uma perspetiva inovadora ao nível da análise do desempenho individual do atleta e ao nível da melhoria das táticas coletivas, entre outros. Esta secção apresenta e examina vários exemplos da aplicação da aprendizagem automática no futebol, centrando-se em estratégias específicas que convertem dados não processados em conhecimento significativo. Assim, uma nova era de compreensão do jogo inicia e o desempenho atlético é melhorado consistentemente.

2.5.1 Previsão de Resultados

Em 2020 foi publicado um artigo (Almulla and Alam, 2020) que tinha como principal objetivo construir modelos de ML que fossem capazes de prever o vencedor de jogos na *Qatar Stars League* (QSL). Não obstante, procurou também identificar as principais métricas de desempenho individual que contribuíssem para que uma equipa vencesse o jogo.

A pesquisa centrou-se no desempenho dos jogadores em 864 jogos desta competição entre 2012 e 2019. Primeiramente foram descartados os dados relativos a jogos onde não existiu vencedor e de seguida os dados foram agrupados tendo em conta três categorias, cada uma delas correspondente à posição do jogador (avançado, médio, defesa). Foram analisados 16 índices relativos à performance técnica e 6 índices relativos à performance física do atleta, nas três diferentes posições, resultando num *dataset* de 69 características. Seguiu-se a realização

de testes de hipóteses ("Anderson-Darling" (Engmann and Cousineau, 2011), "Student T-Test" e "Mann-Whiteny" (Fay and Proschan, 2010)).

Procedeu-se então ao desenvolvimento de vários modelos de ML de classificação, recorrendo a vários algoritmos de ML presentes na biblioteca *scikit-learn* do *Python* (*scikit-learn: machine learning in Python — scikit-learn 1.5.0 documentation*, no date): *k-Nearest Neighbors* (kNN), *Random Forest* (RF), Logistic Regression (LR), Árvores de Decisão, *Multilayer Perception* (MLP), *Support Vector Machine* (SVM) com *kernel* linear (L-SVM), SVM com *kernel* polinomial (P-SVM), SVM com *kernel* de função de base radial (RBF-SVM) e XGBoost.

Almulla e Alam (Almulla and Alam, 2020) testaram os modelos e analisaram a sua performance em quatro parâmetros distintos: exatidão, sensibilidade, especificidade e área sob a curva de representação gráfica de cada um. O modelo que obteve melhor performance na previsão do vencedor de um jogo foi o LR, com 80.1% de exatidão, 79.9% de sensibilidade e 80.4% de especificidade. A explicação dada pelos autores do estudo prende-se com o facto do modelo LR assumir que a transformação logarítmica da variável de saída (rótulo/*label* da classe) tem uma relação linear com as variáveis independentes (vetor de características). Este modelo também requer pouca ou nenhuma multicolinearidade entre as variáveis independentes (vetor de características) e o desempenho de previsão do seu classificador depende do facto de o conjunto de dados subjacente manter o modelo assumido.

O estudo também utilizou técnicas de seleção de características para descobrir as medidas de desempenho dos jogadores que desempenharam um papel na determinação do resultado do jogo. O modelo de ML sugerido identificou características, denominadas qualidades-chave, que consistem em critérios que favorecem uma equipa no sentido de obter a vitória no jogo da competição em análise. Estas incluem o número de remates à baliza por parte dos avançados, a distância percorrida a alta velocidade por parte dos avançados e dos médios e o número de passes bem-sucedidos. Além disso, o estudo realçou a importância dos defesas na obtenção de um resultado positivo e sublinhou que a adoção dos princípios do *fair play* aumenta a probabilidade de vitória no QSL.

2.5.2 Análise do Desempenho Individual

No que toca à análise do desempenho do jogador (ação também denominada de prospeção ou *scouting*), foi proposto o desenvolvimento de uma página *Web* que integrasse um sistema de *scouting* automatizado recorrendo a algoritmos de *data science* (Ghar, Patil and Arunachalam, 2021).

O objetivo deste sistema é recomendar jogadores que se adequem aos critérios do treinador. O treinador pode introduzir os seus parâmetros e, consequentemente, o sistema fornecerá uma lista dos N melhores jogadores que satisfazem esses requisitos. No contexto do futebol, por diversas vezes é sublinhada a importância da química entre os membros da equipa, tendo um impacto substancial no sucesso global da equipa. No entanto, os autores do artigo mencionam que a maioria dos algoritmos (à data da publicação) não têm em conta o conceito de química e

bom entrosamento da equipa ao fazer previsões sobre o desempenho dos jogadores, concentrando-se apenas em medições individuais. A adoção deste comportamento como um padrão pode resultar na proposição de indivíduos que se destacam individualmente, mas que não se alinham estrategicamente com as suas equipas.

Para ultrapassar estes constrangimentos, a investigação sugere uma metodologia em três fases: Pré-processamento; Avaliação dos jogadores melhor classificados de acordo com as suas capacidades individuais e as necessidades estratégicas do treinador; Avaliação dos melhores jogadores com base na sua compatibilidade com o plantel. O sistema utiliza a *framework* Flask para oferecer uma interface *Web* e recorre a duas ferramentas, a SPADL (*Soccer Player Action Description Language*) e a VAEP (*Valuing activities by Estimating Probabilities*), para analisar e avaliar as performances dos jogadores, melhorando assim a acessibilidade e a forma de interpretar os dados.

Os *datasets* utilizados foram extraídos de três fontes diferentes. Para categorizar as informações pessoais de cada jogador (ao nível físico, técnico, tático) foi utilizado um *dataset* de 5 anos de registos do videojogo FIFA (*FIFA 18 Complete Player Dataset*, no date). Este *dataset* disponibilizou 104 atributos diferentes dos jogadores. Para recorrer a dados reais das performances e de dados relacionados com a capacidade de passe dos jogadores em jogos anteriores, este estudo correu ao *dataset* FBref (*Football Statistics and History | FBref.com*, no date), que contém 11 ficheiros CSV. O terceiro *dataset* é proveniente da plataforma Wyscout (*Homepage - Wyscout FootballData*, no date) que contém informação variada, no formato JSON, sobre todos os eventos do jogo e respetivos jogadores, competições e equipas.

Depois de efetuado o pré-processamento dos dados, em que foram eliminados eventuais valores nulos, uniformizados dados comuns a vários *datasets* e a sua normalização, seguiu-se a filtragem dos dados tendo em conta atributos individuais do atleta. Enquanto o estudo apresentado em 2.5.1 agrupou o *dataset* tendo em conta apenas três posições dos jogadores, este vai mais longe e categorizou os dados tendo em conta 15 tipos de jogador, que considerou como sendo "especializações" dessas mesmas posições. Esta categorização deu-se com a combinação dos dois primeiros *datasets*, que permitiu calcular uma pontuação ponderada e normalizada para cada um destes tipos de jogadores e gerar uma lista ordenada dos que melhor se adequam a cada tipo.

Finalizadas as fases de pré-processamento e consolidação dos dados, passou-se à escolha do algoritmo. Inicialmente, foi utilizada a regressão logarítmica como linha de base, seguida da regressão *random forest*. Tanto a regressão logarítmica como a *random forest* com 100 árvores forneceram resultados quase semelhantes. Mais tarde, ao alterar o número de árvores, ou seja, ao aumentar gradualmente para 500, o algoritmo *random forest* mostrou uma melhoria na precisão. Além disso, foi referido que este algoritmo possui bons resultados com dados de entrada não equilibrados e não lineares, como era o caso do *dataset* final que possuía características independentes. Assim, os autores do estudo concluíram que o algoritmo *random forest* era superior em comparação com os algoritmos baseados na regressão linear.

2.5.3 Prevenção de Lesões

A prevenção de lesões também constitui um âmbito em que podem ser utilizadas técnicas de ML, como constatado por (Rommers *et al.*, 2020). Este estudo apresenta e discute os resultados obtidos através da utilização de um modelo de ML para avaliar a probabilidade de lesões em jogadores de futebol jovens e altamente qualificados, utilizando medidas relacionadas com o corpo do atleta, as suas capacidades físicas e características pessoais.

Utilizando dados recolhidos em testes efetuados na pré-temporada, o objetivo principal consistia em avaliar a precisão de um modelo de aprendizagem automática na previsão de lesões durante a época. O segundo objetivo centrou-se na utilização de uma metodologia comparável para categorizar com precisão tipos distintos de lesões, incluindo lesões por uso excessivo e lesões agudas.

Os dados presentes no *dataset* inicial foram obtidos através de um conjunto de testes que abrangiam medidas antropométricas, de coordenação motora e de aptidão física. No total, foram recolhidos dados de 734 atletas do sexo masculino, com idades compreendidas entre os 10 e os 15 anos. Para todas as variáveis alvo de medição, foram calculadas a sua média e desvio padrão. Os autores do estudo procederam à análise estatística destes dados recorrendo à biblioteca XGBoost (*Python Package Introduction — xgboost 2.0.3 documentation*, no date), do Python, justificando a escolha por permitir a afinação do modelo e a inserção de parâmetros de regularização.

Foi utilizada uma amostra aleatória de 80% dos dados recolhidos para formar os dados de treino do modelo. Para aumentar a precisão da previsão, o processo de *boosting* combina um grupo de *weak learners*. Para minimizar a função de custo utilizou-se a descida de gradiente, parametrizada por um parâmetro de modelo específico. Finalmente, para otimizar a função de custo, foi utilizada a perda de articulação. A perda de articulação penaliza as previsões, tanto quando estão incorretas como, quando estão corretas, mas não confiantes. Por fim, o modelo com melhor performance foi testado nos restantes 20% da amostra recolhida, que não foram utilizados no treino do modelo.

Os autores do estudo concluíram que através do modelo desenvolvido foi possível prever lesões ao longo da época com 85% de precisão nos dados de teste. Foram observados apenas 15% de falsos positivos e 15% dos jogadores lesionados foram erradamente classificados como não lesionados com base nas medições efetuadas na pré-temporada. Além disso, foi possível distinguir as lesões por uso excessivo das lesões agudas com uma precisão ligeiramente inferior, utilizando os mesmos resultados dos testes de pré-temporada. Os resultados dos modelos auxiliam na tomada de decisão e na avaliação dos jogadores que mais necessitam de iniciativas de gestão do risco de lesões. A informação suplementar produzida pelo modelo de ML tem o potencial de ajudar os clubes a otimizar a gestão do seu tempo e dos seus recursos para um rastreio abrangente e dessa forma precaver a eventual existência de lesões.

2.5.4 Aprendizagem Automática utilizando rastreamento de dados

Diversos estudos que abordam o desenvolvimento de modelos de ML utilizando rastreamento de dados foram apresentados por (Herold *et al.*, 2019), que refere terem como vantagem possibilitar aos analistas efetuar uma comparação quantitativa entre os jogadores e entre as equipas. Os autores dividiram os modelos desenvolvidos por 4 secções: Reconhecimento e classificação de padrões de passe; Comportamento da equipa com a posse da bola; Comportamento da equipa em função do tempo, espaço e oportunidades de golo criadas; Estratégias defensivas relacionadas com comportamento ofensivo.

2.5.4.1 Reconhecimento e classificação de padrões de passe

Na primeira secção, o autor identificou e analisou um modelo 2D baseado na distância de Frechet e nas distâncias euclidianas médias entre trajetórias para agrupar sequências de passes entre jogadores distintos (Gudmundsson and Wolle, 2014). O *dataset* inicial consistiu em 23 trajetórias possíveis de uma bola de futebol e em 22 jogadores e o *output* do modelo foi a descrição de todos os passes possíveis. Este *output* permitiu aos autores do estudo avaliar a qualidade de execução de um passe, a disponibilidade e capacidade de receção do mesmo e inferir sobre a capacidade de decisão do jogador, quer em situações que o jogador passou a bola, quer quando não efetuou o passe. No entanto, referiram que este modelo necessitaria de ser melhorado com um esquema de categorização de qualidade do passe, antes de ser utilizado como ferramenta pelos treinadores e analistas.

A análise efetuada foi complementada com a referência a um esquema de classificação, que incorporou no modelo 2D um modelo de regressão logística multinomial (Horton *et al.*, 2015). Este modelo de aprendizagem supervisionada utilizou dados de 4 jogos, recolhendo 2932 observações. Embora tenha atingido elevado nível de exatidão (85,6%), os níveis de precisão e *recall* foram relativamente baixos, significando que foram encontrados elevados resultados "falsos negativos" e "falsos positivos". Os autores apontaram como maiores limitações desse estudo o facto dos dados se resumirem a apenas 4 jogos e terem sido registados apenas por 2 observadores, não sendo suficiente para chegar a um consenso.

Foi também referido um modelo de ML supervisionado que tinha como objetivo classificar automaticamente a qualidade dos passes efetuados (Chawla *et al.*, 2017). Os autores utilizaram algoritmos de aprendizagem automática mais complicados para agrupar (não supervisionado) ou classificar (supervisionado) os passes, que foram categorizados como '*Good*', '*Ok*' e '*Bad*'. A precisão obtida, comparando a classificação atribuída pelo modelo com a registada pelo observador humano, foi de 90.2%. Não obstante o elevado grau de precisão obtida, foi mencionado que é preciso ter conta as limitações que as classificações subjetivas possuem, ao nível da medição quantitativa da eficácia do passe, da velocidade da bola e da sua trajetória.

2.5.4.2 Comportamento da equipa com a posse da bola

Este tópico remete para uma análise do jogo de uma perspetiva tática, que muitas vezes tem tendência a tornar-se subjetiva e a afastar-se da objetividade dos dados recolhidos. Pretendendo contornar essa desconcordância, em (Rathke, 2017) foi proposta uma investigação que visava a utilização de inteligência artificial na análise tática. Pretendia-se investigar o quão imprevisível é a estratégia de passes das equipas, sem ter em conta qualquer fio condutor do seu jogo, apenas utilizando os dados relativos a esses mesmo passes. Para isso, foram medidas as trajetórias dos passes efetuados em 380 jogos da *Premier League*, resultando na elaboração de um mapa de imprevisibilidade dos mesmos. Adicionalmente, com o objetivo de classificar o estilo de jogo das equipas, foi utilizado o algoritmo *k-nearest neighbours.* Por fim, combinando esta análise com técnica de representação dos passes e com a recolha de outros 23 dados do jogo, os autores do estudo obtiveram uma precisão de 47% na classificação dos passes e estilo de jogo.

Ainda neste âmbito, foi apresentada uma outra abordagem (Goes *et al.*, 2019) que tinha como premissa fundamental o facto da avaliação da qualidade de um passe deve ser feita em função do efeito que o passe tem sobre a equipa adversária. Partindo de pesquisas anteriores relacionadas com a dispersão e superfície do terreno ocupada pela equipa no seu processo defensivo (Frencken *et al.*, 2011), definiram uma métrica D-Def, que quando calculada representa a alteração na organização defensiva resultante de um passe do adversário. Ou seja, com base na diferença entre as posições dos jogadores quando o passe é efetuado e as posições registadas três segundos após esse momento, é possível verificar o efeito que o passe teve na organização da equipa. Para além disso, foi possível ainda classificar os passes e quais os jogadores com mais eficácia de passes, constatando que um passe mais rápido e preciso está associado a uma situação de sucesso e, consequentemente, maior valor de D-Def. Esta abordagem distinguiu-se da seguida por outros estudos neste âmbito por considerar todas as direções em que passes são efetuados e não apenas passes frontais. No entanto, tem como falha comum o facto de considerar apenas passes realizados com sucesso, o que não permite inferir na totalidade sobre a capacidade de passe e de decisão do atleta.

2.5.4.3 Comportamento da equipa em função do tempo, espaço e oportunidades de golo criadas

A aplicação de técnicas de ML baseadas em dados de localização permitiu a implementação de estratégias ofensivas relacionadas com a gestão do espaço e do tempo. Em (Lucey *et al.*, 2014), foram analisados os 10 segundos imediatamente anteriores ao momento em que ocorre a tentativa de remate, com o objetivo de encontrar o valor de *xG*.

Foram analisados vários elementos espaciais, tais como a posição do campo onde o remate ocorreu, a posição dos defesas em redor do jogador e ainda a velocidade com que a jogada decorreu. Assim, utilizando estes dados e um modelo ML seguindo regressão logística, foi possível estimar a probabilidade que cada remate tem em resultar em golo. Por ter em conta os fatores espácio-temporais e não apenas a indicação se o remate foi bem sucedido ou não,

este modelo foi apontado por (Herold *et al.*, 2019) como mais completo comparado com os demais modelos desenvolvidos com o mesmo objetivo.

Também no âmbito da análise da janela temporal de 10 segundos antes do remate, em (Power *et al.*, 2017) foram utilizados dados de monitorização para classificar os passes com base no risco (representado pela probabilidade do passe ser executado em cada circunstância), e na recompensa (representada pela probabilidade de um passe resultar numa oportunidade de golo). Foi criada a métrica *Dangerous Pass* (DP) definida como uma tentativa de passe que tem uma probabilidade superior a seis por cento de resultar num remate durante os dez segundos seguintes.

O *dataset* utilizado consistia em dados de duas épocas da *Premier League*, entre 2014 e 2016, resultando numa amostra de 352466 dados de teste. Estes dados foram utilizados para testar o modelo, desenvolvido tendo por base a regressão logística. Os autores do estudo apresentaram como justificação a escolha deste modelo o facto de, ao contrário de um modelo não linear, este poder ser mais facilmente interpretável e, consequentemente, permitir melhor análise dos dados que conduzem a níveis mais altos da performance do atleta. Aplicando técnicas de *cluster* k-*nearest neighbours*, em que cada *cluster* representa uma zona do campo, foi possível constatar para cada equipa analisada a zona do campo com maior valor de DP, verificando ainda que, na grande maioria das equipas, os passes que resultaram na maior recompensa (DP) foram aqueles que aconteceram perto do limite da área de grande penalidade. Este tipo de análise proporcionado por este modelo foi apontado pelos autores do estudo como essencial no que toca à preparação e estudo do adversário.

3 Design e Desenvolvimento

Neste capítulo é apresentado o design da solução desenvolvida e é descrito todo o processo de desenvolvimento seguido.

3.1 Design

Relativamente ao design da solução desenvolvida, é importante realçar as tecnologias utilizadas, a arquitetura seguida na conceção da solução e ainda uma proposta de arquitetura de sistema a utilizar posteriormente.

3.1.1 Tecnologias

A solução foi desenvolvida e implementada na linguagem Python, versão 3.10.1, recorrendo ao seguintes módulos e bibliotecas:

- Matplotlib, versão 3.9.0.
- Numpy, versão 1.26.4.
- Pandas, versão 2.2.2.
- Pyodbc, versão 5.1.0.
- Skicit-Learn, versão 1.5.0.

3.1.2 Pipeline da Solução

A pipeline da solução consiste numa sequência organizada de operações, que parte de dados não processados até atingir o objetivo do estudo, sejam previsões, classificações, etc. A adoção desta abordagem sequencial e modular facilita não só a manutenção do código e a sua leitura, como também aumenta a reprodutibilidade e escalabilidade do modelo (Géron, 2017). A Figura

9, adaptada de (Nazarov, 2023) representa o processo seguido e que será descrito a partir da secção 3.2. As únicas fases que não foram implementadas nesta dissertação foram a implementação do modelo em produção e consequente *feedback*.

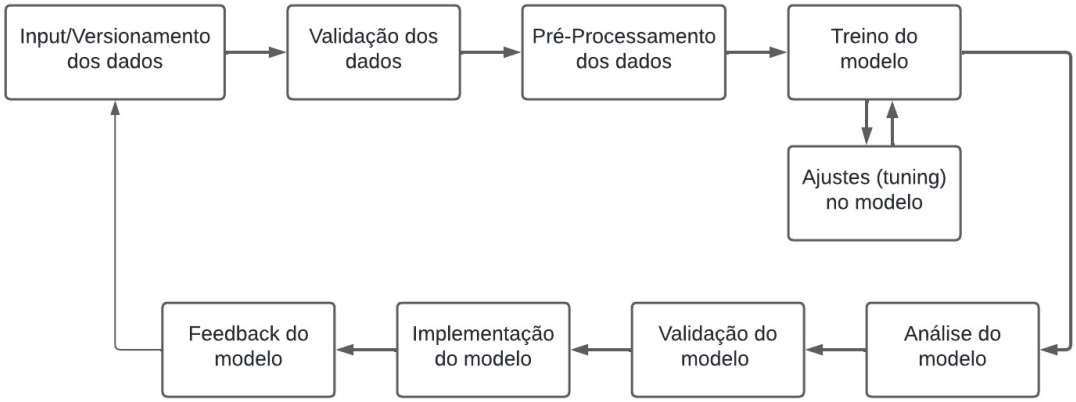

Figura 9 - Fluxo de Implementação do Modelo

Este fluxo tem início no Input/Versionamento dos dados, que consiste na recolha de dados de diversas fontes (bases de dados, ficheiros CSV, entre outros) e na utilização de ferramentas como o *Data Version Control* (DVC) (dvc.ai, no date) para garantir o rastreio e reprodução de qualquer alteração nos dados. A fase seguinte é a validação dos dados, onde é verificada a sua integridade, avaliando a existência de valores duplicados, inconsistentes ou até mesmo ausentes. Neste passo também é analisada a distribuição dos dados, inferindo sobre a presença de *outliers*.

A terceira etapa, o pré-processamento dos dados, pode dividir-se em três momentos: primeiramente é efetuada a limpeza do *dataset*, onde são removidos ou corrigidos dados cujos valores sejam inconsistentes com o restante conjunto de dados; segue-se a transformação dos dados, que consiste em diversas técnicas, tais como a normalização e o seu balanceamento ou a conversão de variáveis categóricas em numéricas; termina com a divisão dos dados, que visa separar o *dataset* em conjuntos de treino, validação e teste para avaliar o desempenho do modelo.

Na fase de treino do modelo é selecionado o algoritmo ML mais adequado ao problema proposto, através de uma análise comparativa das alternativas existentes. Posteriormente, o conjunto de treino é utilizado no modelo, permitindo a aprendizagem dos padrões existentes nos dados. Paralelamente a esta fase, ocorre a etapa de ajustes do modelo, que consiste na otimização dos hiper parâmetros para melhorar o seu desempenho e na utilização de técnicas de validação cruzada para avaliar a sua robustez e evitar o *overfitting*.

Relativamente à fase de análise, consiste na avaliação do desempenho do modelo seguindo diferentes métricas, dependendo se se trata de um problema de regressão ou de classificação. Dessa análise podem resultar ajustes aos dados e/ou aos modelos. Segue-se a etapa de validação do modelo, onde se pretende testar a sua capacidade em obter bom desempenho

quando confrontado com dados desconhecidos, simulando o ambiente de produção. Nesta fase são efetuados os últimos ajustes ao modelo antes de passar para a fase de implementação.

Na fase de implementação, o modelo desenvolvido é de facto colocado num ambiente de produção e encontra-se disponível para ser utilizado em tempo real. Tipicamente associada a esta fase está a utilização de ferramentas (como o *Docker*[15] ou *Kubernetes*[16]) para garantir a escalabilidade do modelo e que o seu desempenho não diminui com o aumento do volume de dados.

Por fim, a fase de *feedback*, que consiste na recolha e monitorização dos resultados obtidos pelo modelo em produção, garantindo que o seu desempenho não é afetado negativamente e continua a dar resposta às necessidades do sistema. Qualquer ajuste proveniente da fase de *feedback* será integrado na fase de Input/Versionamento dos dados, evidenciando a natureza cíclica deste processo.

3.1.3 Arquitetura do Sistema

Foi discutida e definida uma arquitetura de sistema a utilizar caso se pretenda, a partir do modelo desenvolvido e do estudo efetuado, implementar um sistema de suporte à decisão a ser utilizado pelos profissionais técnicos/analistas das equipas de futebol.

Na Figura 10 encontra-se representada a arquitetura do sistema proposta.

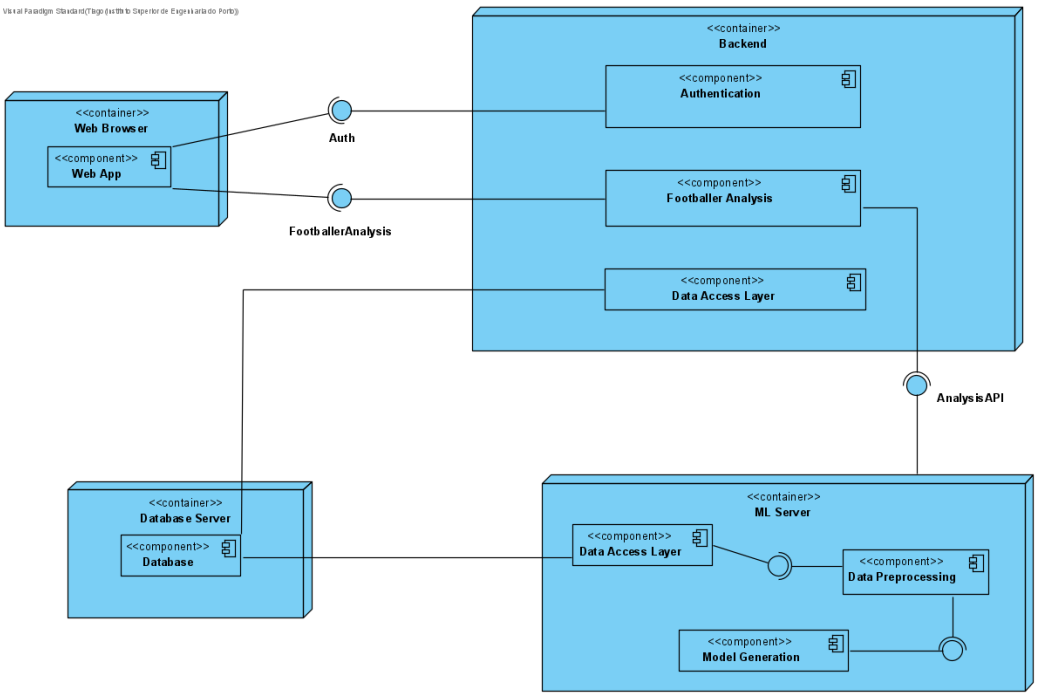

Figura 10 – Arquitetura do Sistema

[15] https://www.docker.com/
[16] https://kubernetes.io/

Atualmente, todos os *containers* representados com exceção do *container* "MLServer" já existem. Será necessário proceder à sua implementação e à implementação do *component* "FootballerAnalysis", que comunicará com o *container* "ML Server". Este *container* terá duas responsabilidades principais: alojar os modelos desenvolvidos no âmbito desta dissertação e, através dos seus diversos componentes, lidar com todos os pedidos destinados à análise de dados dos jogadores, tendo em conta as implementações ML efetuadas.

O *container* "WebApp" é a interface utilizada pelos utilizadores do sistema. Desempenha a função de porta de entrada para os utilizadores se autenticarem e efetuarem as demais requisições que pretendam, que passam pelo *container* "Backend" (desempenhando o papel de m*iddleware*) para serem redirecionadas para o *container* "ML Server".

Todos os dados utilizados pelos modelos (e outros dados do sistema) estão armazenados no *container* "Database".

3.2 Desenvolvimento

3.2.1 Processamento dos Dados

Todos os dados utilizados resultam de recolhas efetuadas pela empresa X (nome fictício por razões de confidencialidade) durante jogos de futebol realizados nas últimas duas épocas desportivas, sendo posteriormente armazenados numa base de dados SQL Server. Esses dados estão tipicamente associados a eventos que acontecem no jogo, tais como um passe, um remate, uma falta, um golo, entre outros, e ainda ao interveniente responsável por esse evento, o jogador. Como este estudo pretende analisar o desempenho dos jogadores no que diz respeito à marcação de golos, os dados foram filtrados para se obter a informação relativa à totalidade dos remates e de golos de cada jogador.

Primeiramente, foi analisada e interpretada a informação armazenada na base de dados SQL. As distribuições de frequência dos remates e golos dos jogadores em várias posições são examinadas nas tabelas seguintes. Inicialmente são apresentadas as frequências absolutas e relativas no que diz respeito aos remates e, de seguida, as frequências absolutas e relativas no que toca aos golos. Analisando as colunas relativas à frequência absoluta, é possível observar onde é que os dois tipos de evento se concentram em cada posição, enquanto a análise da frequência relativa confere uma compreensão proporcional dos eventos e permite comparações entre as diferentes posições.

Tabela 9 - Frequência absoluta e relativa de remates por posição

Posição	Frequência Absoluta	Frequência Relativa (%)
Guarda-Redes	1	0,01%
Defesa	2395	18,08%
Médio	3833	28,94%
Avançado	7014	52,96%

Tabela 10 - Frequência absoluta e relativa de golos por posição

Posição	Frequência Absoluta	Frequência Relativa (%)
Guarda-Redes	0	0%
Defesa	196	13,96%
Médio	301	21,44%
Avançado	907	64,60%

Como é possível verificar pela Tabela 9 e pela Tabela 10, o peso dos dados recolhidos para a posição de guarda-redes na globalidade da amostra é praticamente nulo. Por essa razão, esses dados foram descartados dos passos seguintes do estudo. Constata-se também que, quer nos remates, quer nos golos, as amostras em que a posição do jogador é "Avançado" são as que possuem maior frequência absoluta.

Nesse sentido, foi construído um *dataset* para armazenar os dados relativos aos remates dos jogadores e outro para armazenar os dados relativos aos seus golos. O *dataset* com informação dos remates dos jogadores continha 619 amostras que estavam organizadas de acordo com a Figura 11. O *dataset* dos golos era composto por 381 amostras, sendo que a sua informação estava estruturada de acordo com a Figura 12.

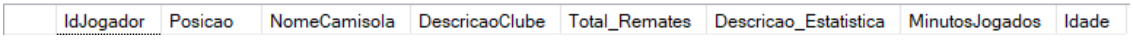

Figura 11 - Colunas do dataset relativo aos remates

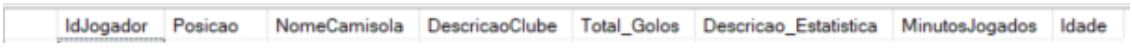

Figura 12 - Colunas do *dataset* relativo aos golos

Uma vez que se pretendia estudar a eficácia dos remates dos jogadores, e esse valor é dado pela expressão $GoalRatio = TotalGolos/TotalRemates$, foi necessário calcular este rácio através da combinação dos dois *datasets* apresentados na Figura 11 e na Figura 12, somando os valores de remates e golos associados a cada jogador e posteriormente efetuando o quociente entre as duas somas. O *dataset* final é composto pelas colunas apresentadas na Figura 13.

NomeCamisola	Posicao	Idade	TotalRemates	TotalGolos	MinutosJogados	GoalRatio

Figura 13 - Colunas do *dataset* final

A presença das colunas "Posicao", "MinutosJogados" e "Idade" no *dataset* deve-se ao facto de estarem diretamente ligadas ao desempenho individual do atleta, sendo por isso importante avaliar o seu impacto. Na Tabela 11 são apresentadas algumas características do *dataset*. Constata-se que a dimensão do *dataset* diminuiu comparativamente com os dados apresentados nas Tabela 9 e Tabela 10, apresentando agora 361 registos. Essa diminuição deve-se precisamente ao agrupamento de dados mencionado no parágrafo anterior.

Tabela 11 - Características do *dataset* com *GoalRatio* calculado

Posição	**Frequência Absoluta**	**Frequência Relativa (%)**	**Min – Max Golos**	**Média Golos**	**Min – Max GoalRatio**	**Média GoalRatio**
Guarda-Redes	0	0%	-	-	-	-
Defesa	98	27,14%	1 – 10	2,02	0,02 – 1	0,17
Médio	99	27,42%	1 – 23	3,01	0,02 – 0,43	0,11
Avançado	164	45,43%	1 – 41	5,54	0,02 - 1	0,14

Para inferir sobre a presença de *outliers* e, caso existam, determinar a sua percentagem na amostra, foi utilizado o método *Interquartile Range* (IQR) (Dekking *et al.*, 2006). A aplicação deste método tem como objetivo analisar a distribuição dos dados e perceber se existem valores que se desviam de forma significativa dos restantes, denominados de *outliers*. A Figura 14 ilustra a distribuição dos valores de *GoalRatio* do *dataset*.

Os seis pontos situados acima do limite superior do *boxplot* são os denominados *outliers*. Porém, não significa que sejam erros de medição ou valores com os quais não se deve ter conta. Vão ser considerados no estudo para perceber quais as características ou atributos que contribuíram para que apresentassem esses mesmos valores. A presença dos *outliers* no *dataset* a utilizar no modelo de ML depende sempre do contexto do estudo, como abordado em (Hleap, 2022).

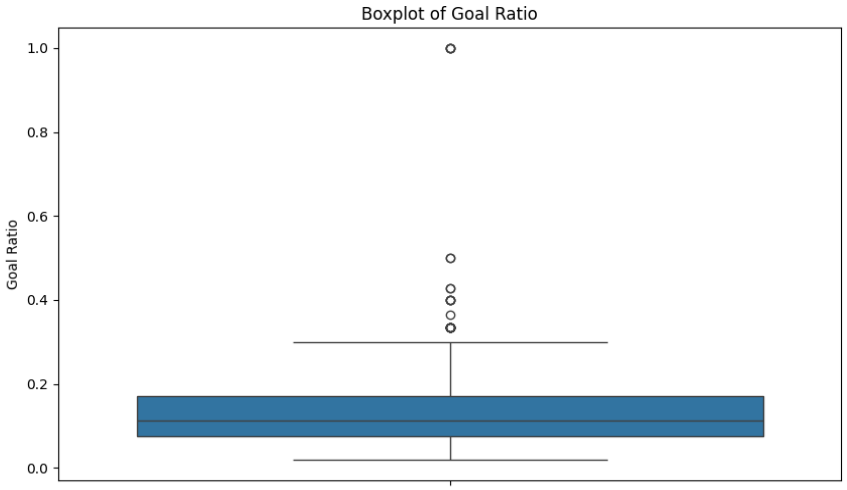

Figura 14 - Distribuição dos dados relativos a GoalRatio

O passo seguinte foi a codificação (*encoding*) dos dados. Este processo consiste em transformar dados, tipicamente não numéricos, em dados numéricos e compatíveis com os algoritmos ML (Baruah, 2023). Neste caso, apesar de existirem duas variáveis com valores não numéricos (o "NomeCamisola" e "Posicao"), apenas foi codificada a variável "Posicao", por se tratar de uma variável categórica e que se pretendia utilizar no algoritmo ML.

3.2.2 Normalização dos Dados

A normalização do *dataset* é uma operação a efetuar antes do treino e validação dos modelos, quer em problemas de regressão, quer em problemas de classificação. No contexto do modelo de regressão, tendo em conta que a variável representativa do número de remates (que varia entre as dezenas e as centenas) é distinta da que representa a eficácia do atleta (que varia entre 0 e 1), faz sentido normalizar os dados para que estejam na mesma escala e não enviesem a estimativa dos coeficientes.

Relativamente ao modelo de classificação, ainda que modelos como AD e RF sejam menos sensíveis à existência de variáveis em diferentes escalas (Breiman, 2001), o mesmo não acontece com algoritmos baseados em distância, como o kNN e o SVM, em que tipicamente o desempenho do modelo é melhor quando as *features* possuem o mesmo peso (Singh and Singh, 2020).

As técnicas de normalização usadas neste processamento, pela sua vasta aplicabilidade, referência em vários trabalhos (Singh and Singh, 2020) e se adequarem ao contexto, foram *Min-max Normalization* e *Z-score Normalization*. Ambas as técnicas constam da biblioteca *scikit-learn*.

No capítulo 4 serão apresentadas comparações dos resultados dos modelos com e sem normalização dos dados.

3.2.2.1 Min-max Normalization

Esta técnica, cuja fórmula se encontra representa pela Equação (1), aplica uma transformação linear ao *dataset* original com o intuito que todos os dados fiquem num determinado intervalo (tipicamente entre 0 e 1) (Akanbi, Amiri and Fazeldehkordi, 2014).

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

Em que x é o valor original do *dataset*, $min(x)$ o valor mínimo e $max(x)$ o valor máximo.

3.2.2.2 Z-score Normalization

Esta técnica de normalização de dados baseia-se na média aritmética e no desvio padrão dos dados do *dataset* para uniformizar as *features* utilizadas, garantido que possuem a mesma importância, sendo processadas pelo modelo (Jain, Nandakumar and Ross, 2005). O seu cálculo é dado pela Equação (2).

$$x' = \frac{x - \mu}{\sigma} \quad (2)$$

Em que x é o valor original do *dataset*, μ é a média e σ o desvio padrão.

3.2.3 Escolha dos Algoritmos ML

Foi adotada uma abordagem metódica na seleção do modelo de aprendizagem automática a utilizar para solucionar o problema de regressão, considerando vários algoritmos e dando ênfase à eficiência da previsão e no ajuste aos dados fornecidos. Recorrendo à biblioteca *scikit-learn* da linguagem *Python* (*scikit-learn: machine learning in Python — scikit-learn 1.5.0 documentation*, no date), foram testados diversos modelos de ML.

Devido à sua facilidade de utilização e interpretação, foi inicialmente utilizado um modelo de Regressão Linear para estabelecer uma linha de base para a previsão. No entanto, modelos como kNN, Árvores de Decisão e SVM (identificados na secção 0) foram também utilizados e avaliados para garantir a precisão e a robustez das previsões. Possuem características distintas, o que releva a sua comparação: O kNN baseia-se na proximidade dos dados e o seu uso é intuitivo, a Árvore de Decisão caracteriza-se por captar relações não lineares e a SVM apresenta bons resultados na classificação em ambientes de elevada dimensão (Hastie, Tibshirani and Friedman, 2009). Por último, foram testados os modelos *Multilayer Perceptron* (MLP) e GB. Com exceção do último modelo mencionado, que foi utilizado recorrendo à biblioteca XGBoost, todos os restantes modelos encontram-se na biblioteca Scikit-Learn.

Relativamente ao modelo de classificação, a análise efetuada em 2.3.2 suportou a escolha dos algoritmos a utilizar. Serão testados e comparados os modelos RF, AD, SVM, kNN, *GuassianNB* e GB. Distinto do modelo de regressão, o objetivo do modelo de classificação passa por prever o desempenho dos jogadores com base no *GoalRatio*, categorizando esta variável em cinco classes, cada uma representando um nível de desempenho.

3.2.4 Divisão do *Dataset*

Segundo o processo apresentado em (Gallatin and Albon, 2023), dividir o *dataset* antes de treinar e avaliar o modelo constitui uma boa prática, evitando problemas tipicamente relacionados com *overfitting*, tais como (*machine learning - Why is it wrong to train and test a model on the same dataset? - Data Science Stack Exchange*, no date):

- Falta de generalização, que consiste na excessiva especialização e consequente incapacidade para generalizar diferentes tipos de dados. Sabendo de cor o conjunto de dados de treino, o modelo pode não funcionar corretamente quando confrontado com dados com que nunca contactou.
- Otimização de casos específicos, dada pela capacidade do modelo em se personalizar tendo em conta determinado dado ou característica que observa. Existindo novos dados ou diversificando os cenários, o modelo pode não funcionar corretamente.
- Falha em detetar *overfitting*, que se pode manifestar pela excelente performance nos mesmos dados, mas uma performance muito pior em novos dados que surjam no *dataset*.

Pelo exposto, procedeu-se à divisão do *dataset*. No entanto, existem diferentes métodos para realizar esta ação. Sendo os mais comuns o *holdout* e *k-fold cross-validation*, foram identificados e analisados outros métodos (variantes do segundo) no sentido de averiguar o que melhor se adequava ao *dataset*, encontrando-se descritos nas subsecções seguintes.

3.2.4.1 Holdout

Este método foi sugerido como uma solução para o problema *overfitting* que surgia com a utilização dos mesmos dados, quer para treinar o modelo, quer para o testar. O método consiste na divisão do *dataset* em duas partes distintas, uma com a responsabilidade de treinar o modelo e outra de o testar (Yadav and Shukla, 2016).

3.2.4.2 K-Fold Cross-Validation

Esta técnica tem início na divisão equitativa em k segmentos iguais. Considerando estes segmentos, são efetuados treinos e testes no mesmo número de k iterações, usando apenas

um dos segmentos para testar e os restantes para treinar o modelo. A precisão do modelo consiste na média das precisões obtidas em cada uma das iterações (Yadav and Shukla, 2016).

3.2.4.3 Stratified K-Fold Cross-Validation

Sendo uma variação do método *K-Fold Cross-Validation*, este método destaca-se pela sua utilidade na divisão de *datasets* em que seus dados não estão distribuídos de forma equilibrada pelas classes. Aplicando-o, cada *fold* apresentará aproximadamente a mesma percentagem de dados de cada classe, aumentando a fiabilidade dos dados de teste e de avaliação do modelo (Olamendy, 2024). A Figura 15 (Pramod, 2023) é exemplificativa da aplicação deste método.

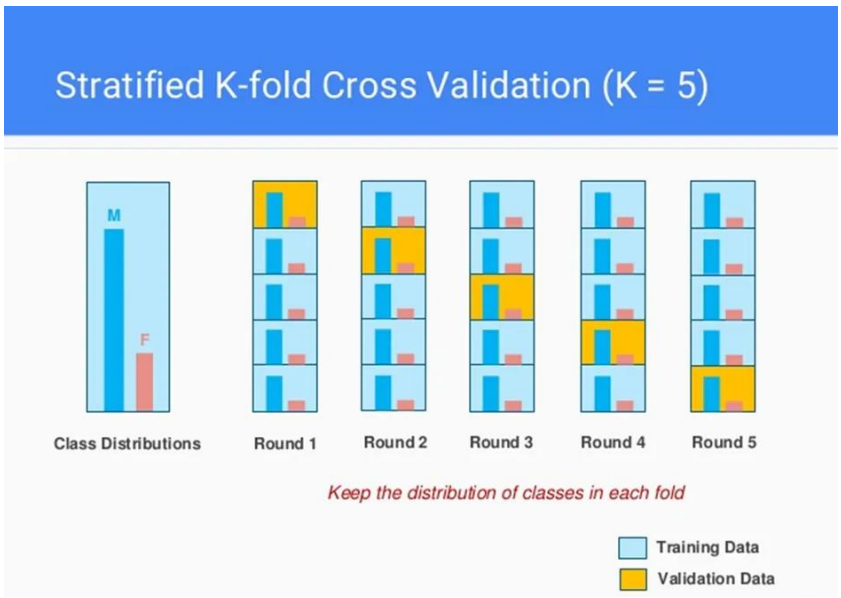

Figura 15 - Exemplo da aplicação do *Stratified K-Fold Cross Validation* (Pramod, 2023)

3.2.4.4 Leave-One-Out Cross-Validation (LOOCV)

Como o seu nome indica, o método LOOCV também tem origem no método *K-Fold Cross-Validation*. Distingue-se pela particularidade de o número de *k* ser igual ao número de exemplos que constam do *dataset*. É um método que exige muitos recursos computacionais, não sendo por isso apropriado para *datasets* na ordem das dezenas de milhares de exemplos. De todos os *folds* gerados, apenas um é utilizado como teste, sendo os restantes destinados a treinar o modelo. Isto constitui uma vantagem deste método em relação aos restantes, uma vez que confere maior robustez na estimativa de desempenho do modelo (Brownlee Jason, 2020).

3.2.4.5 Shuffle-Split Cross-Validation

Esta técnica de divisão de *dataset* tem como premissa a divisão aleatória em conjuntos de treino e testes ao longo de várias iterações (que não têm um valor fixo). A ocorrência de sucessivas divisões aleatórias leva a que possa existir sobreposição dos conjuntos de testes ao longo das várias iterações (Abhigyan, 2021). A Figura 16 exemplifica a aplicação desta técnica.

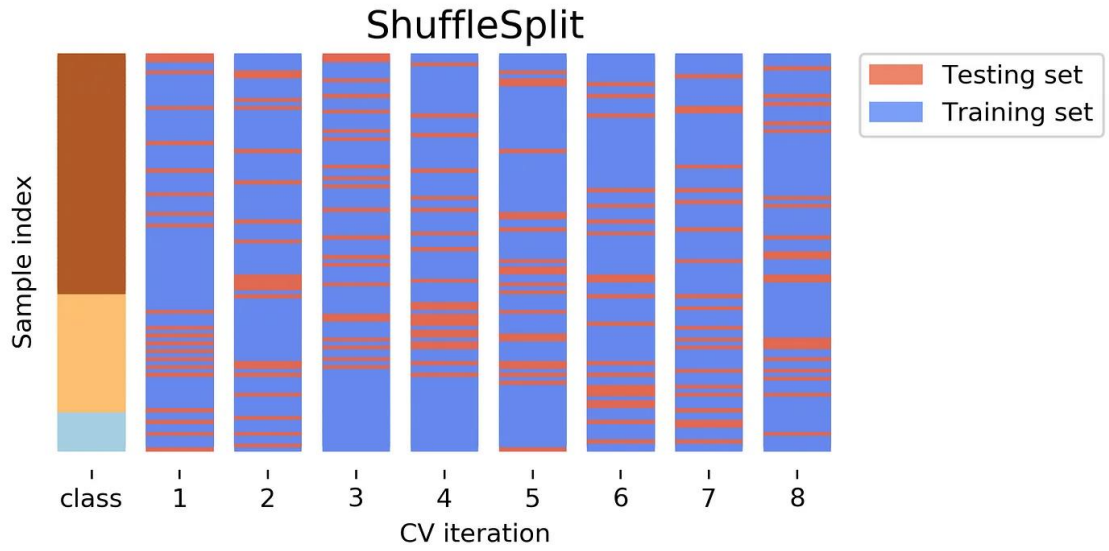

Figura 16 - Exemplo da aplicação de *Shuffle-Split Cross-Validation* (Abhigyan, 2021)

Efetuada a análise das diferentes técnicas, a escolha foi orientada pelas características do *dataset*, apresentadas na Tabela 11. Tendo em conta que a classe "Guarda-Redes" foi descartada, verifica-se que a divisão entre as restantes classes é desigual, com a classe "Avançado" a representar aproximadamente metade da amostra. Pelo que foi descrito na secção 3.2.4.3, seria plausível a escolha do método *Stratified K-Fold Cross-Validation*. No entanto, é preciso ter em conta o contexto em que o *dataset* se insere e o problema que se pretende resolver.

As classes deste *dataset* representam uma característica importante durante um jogo de futebol, a posição do jogador. Sendo algo que distingue o jogador, é importante que o modelo não retire o peso a essa característica e mantenha a distribuição do *dataset*. Por outro lado, para o modelo de classificação é importante garantir que não existem classes cujos dados estão desbalanceados. A combinação de técnicas de *undersampling*/*oversampling* com a adoção do método *Stratified K-Fold Cross-Validation* garante a distribuição dos dados de forma proporcional pelas classes.

Pelo exposto, para o modelo de regressão foi escolhido o método de divisão *K-Fold Cross-Validation* (com k = 5) enquanto para o modelo de classificação foi escolhido o método de divisão *Stratified K-Fold Cross-Validation* com (k = 5). Para este último, foi comparado o desempenho do modelo quando treinado com um *dataset* alvo de técnicas de *oversampling*: *random oversampling* (RO) e *synthetic oversampling* (SO). A comparação foi efetuada não só

por se tratar de um problema de classificação (onde é prática comum), mas também pelo facto de algumas classes possuírem menos de 5 registos. Os dados relativos à comparação do desempenho do modelo sob as diferentes circunstâncias mencionadas são detalhados na secção 4.3, na Tabela 16.

3.2.5 Escolha de Hiper parâmetros

Ao contrário dos parâmetros do modelo, que são aprendidos pelo algoritmo ML com base nos dado de treino, os hiper parâmetros são parâmetros que são definidos antes da fase de treino. A sua otimização tem como objetivo controlar e melhorar o desempenho do modelo (Géron, 2017). Existem várias técnicas amplamente utilizadas para este propósito, cada uma com suas características e vantagens:

- *Grid Search*: Consiste numa das abordagens mais populares, onde é efetuada uma busca exaustiva por todas as possibilidades de combinações de hiper parâmetros, num espaço definido pelo utilizador. Apresenta resultados eficazes, tendo como contrapartida um elevado custo computacional caso seja aplicada em *datasets* de elevada dimensão (Bergstra, Ca and Ca, 2012).
- *Random Search*: Como o nome indica, esta técnica seleciona as combinações dos hiper parâmetros de forma aleatória. É uma abordagem que pode encontrar a melhor solução em menos tempo do que *Grid Search* (Bergstra, Ca and Ca, 2012).
- *Bayesian optimization*: Esta técnica é mais recente que as anteriores e apresenta diferenças no que toca à busca que efetua. Os hiper parâmetros são avaliados por um modelo probabilístico, o que possibilita a otimização do processo de busca de forma iterativa (Snoek, Larochelle and Adams, 2012).

Tendo em conta a dimensão do *dataset* e a facilidade com que se encontra implementado na linguagem Python, foi escolhido o método *GridSearch*. Os seus resultados são detalhados na subsecção 4.2.1.

4 Avaliação dos Modelos e Análise dos Resultados

Finalizado o desenvolvimento dos modelos, é necessário testá-los, validá-los, analisar os resultados obtidos e efetuar as devidas comparações, com o objetivo de averiguar quais os que tiveram o melhor desempenho. Primeiramente, são apresentadas as métricas de desempenho tipicamente utilizadas para avaliar modelos de regressão e de classificação. Seguidamente, é apresentado o desempenho do modelo de regressão desenvolvido para dar resposta ao problema de regressão, que consiste na previsão de *GoalRatio*, o rácio entre golos marcados e a quantidade de remates de cada jogador. Por fim, é apresentado o desempenho do modelo desenvolvido no âmbito do problema de classificação, que visa classificar o desempenho do jogador com base no seu valor de *GoalRatio*.

4.1 Métricas de Desempenho

Para avaliar o desempenho de cada um dos modelos, foi efetuada uma análise comparativa das métricas de desempenho, apresentada na Tabela 12 e na Tabela 16. Tendo em conta os dois contextos de problema de aprendizagem supervisionada (regressão e classificação) em que os modelos foram utilizados, foram analisadas diferentes métricas de desempenho.

4.1.1 Métricas de Desempenho em Modelos de Regressão

Segundo (Botchkarev, 2018), as principais métricas de desempenho para avaliar os modelos de regressão são: o Erro Médio Absoluto (MAE), o Erro Quadrático Médio (MSE) e a Raiz do Erro Quadrático Médio (RMSE).

4.1.1.1 MAE

O MAE, calculado através da Equação (3), mede a média das diferenças absolutas entre os valores previstos e os valores reais.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (3)$$

Onde $\boldsymbol{y_i}$ representa o valor real, $\boldsymbol{\hat{y}_i}$ o valor previsto e **n** o número de observações.

4.1.1.2 MSE

O MSE, calculado através da Equação (4), representa a média dos quadrados das diferenças entre os valores reais e os valores previstos.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (4)$$

Onde $\boldsymbol{y_i}$ representa o valor real, $\boldsymbol{\hat{y}_i}$ o valor previsto e **n** o número de observações.

4.1.1.3 RMSE

O RMSE, calculado através da Equação (5), consiste no cálculo da raiz quadrada do MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (5)$$

Onde $\boldsymbol{y_i}$ representa o valor real, $\boldsymbol{\hat{y}_i}$ o valor previsto e **n** o número de observações.

4.1.2 Métricas de Desempenho em Modelos de Classificação

Entre as métricas para avaliar a performance dos modelos de classificação mais usadas, encontram-se as seguintes: *Accuracy* (ACC), *Precision* (PRE), *Recall* (RCL) e *F1-Score* (F1) (Sokolova and Lapalme, 2009).

4.1.2.1 Accuracy

O cálculo da *Accuracy* é dado pela Equação (6) e representa a proporção de previsões corretas em relação ao total de previsões.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}} \quad (6)$$

Em que TP representa os verdadeiros positivos (*true positives*), TN representa os verdadeiros negativos (*true negatives*), FP representa os falsos positivos (*false positives*) e FN representa os falsos negativos (*false negatives*).

4.1.2.2 Precision

O valor da *Precision* é calculado através da Equação (7) e mede a proporção de previsões TP em relação ao total de previsões positivas feitas pelo modelo.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}} \quad (7)$$

4.1.2.3 Recall

Obtido pela fórmula apresentada na Equação (8), o cálculo de *Recall* representa a proporção de verdadeiros positivos medidos em relação ao total de verdadeiros positivos existentes.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \quad (8)$$

4.1.2.4 F1-Score

Esta métrica une a *Precision* e *Recall*, calculando a média harmónica ao invés da média aritmética para existir o devido equilíbrio entre as duas métricas de desempenho. É calculada através da fórmula apresentada na Equação (9)

$$\boldsymbol{F1\ Score} = \frac{\mathbf{2} \times \boldsymbol{TP}}{\mathbf{2}\ \times \boldsymbol{TP} + \boldsymbol{FN} + \boldsymbol{FP}} \quad (9)$$

4.2 Previsão do *GoalRatio* dos Jogadores

Esta secção visa apresentar a comparação dos modelos ML utilizados para dar resposta ao problema de regressão colocado, que consistia na previsão do *GoalRatio* dos jogadores.

Partindo do *dataset* mencionado no final da subsecção 3.2.1, procedeu-se ao processamento dos seus dados pelos diferentes modelos ML de regressão identificados. Tendo em conta a utilização de técnicas de normalização e balanceamento de dados, foi inferido sobre a sua influência nos resultados obtidos, comparando a qualidade do modelo em dados de controlo

(sem aplicação de qualquer técnica) com dados normalizados e balanceados. Foram selecionados os dois modelos com melhor desempenho para a fase de ajuste e otimização dos hiper parâmetros. A Tabela 12 apresenta os resultados obtidos pelos diferentes modelos nas métricas de desempenho definidas em 4.1.1.

Tabela 12 - Resultados obtidos pelos Modelos de Regressão

Modelo	**Dados de Controlo**	**Normalização Min-max**	**Normalização Z-Score**
RLin	MAE: 0,061 MSE: 0,011 RMSE: 0,104	MAE: 0,062 MSE: 0,012 RMSE: 0,106	MAE: 0,062 MSE: 0,011 RMSE: 0,106
kNN	MAE: 0,076 MSE: 0,014 RMSE: 0,118	MAE: 0,055 MSE: 0,010 RMSE: 0,099	MAE: 0,047 MSE: 0,009 RMSE: 0,094
SVM	MAE: 0,072 MSE: 0,013 RMSE: 0,112	MAE: 0,064 MSE: 0,011 RMSE: 0,105	MAE: 0,059 MSE: 0,009 RMSE: 0,093
AD	MAE: 0,022 MSE: 0,00091 RMSE: 0,029	**MAE: 0,012** **MSE: 0,00076** **RMSE: 0,027**	MAE: 0,012 MSE: 0,00077 RMSE: 0,027
MLP	MAE: 0,097 MSE: 0,030 RMSE: 0,165	MAE: 0,070 MSE: 0,014 RMSE: 0,116	MAE: 0,056 MSE: 0,010 RMSE: 0,098
GB	MAE: 0,009 MSE: 0,0003 RMSE: 0,017	MAE: 0,0089 MSE: 0,00027 RMSE: 0,016	**MAE: 0,0088** **MSE: 0,00026** **RMSE: 0,016**

Os resultados apresentados na Tabela 12 permitem verificar o impacto que as técnicas de normalização tiveram no desempenho nos modelos. Enquanto alguns modelos, como RLin, AD e GB, são relativamente indiferentes à utilização de técnicas de normalização e o seu desempenho não apresenta variação significativa, no caso do kNN e SVM e MLP o impacto é maior levando a melhorias no seu desempenho. No entanto, os modelos que apresentaram melhores valores de métrica de desempenho foram: o AD (com min-max *normalization*), que apresentou erro médio absoluto = 0,0012, erro quadrático médio = 0,00076 e raiz do erro quadrático médio = 0,027; e o GB (com z-score *normalization*), que apresentou erro médio absoluto = 0,0088, erro quadrático médio = 0,00026 e raiz do erro quadrático médio = 0,016.

4.2.1 Hiper parâmetros

Seguidamente, foi aplicado o método de otimização de hiper parâmetros *GridSearch* nestes dois modelos (AD e GB), com o objetivo de perceber quais os que promovem melhores resultados nas métricas de desempenho.

A Tabela 13 apresenta os parâmetros fornecidos ao método *GridSearch*.

Tabela 13 - Parâmetros fornecidos ao método *GridSearch* nos modelos de regressão

Modelo	Hiper parâmetros
AD	'*max_depth*': [3, 5, 10], '*min_samples_split*': [2, 5, 10]
GB	'*n_estimators*': [50, 100, 150], '*learning_rate*': [0.01, 0.1, 0.5], '*max_depth*': [3, 5, 8]

A Tabela 14 apresenta uma comparação entre o desempenho dos modelos selecionados na Tabela 12 e o desempenho dos modelos quando executados com os melhores hiper parâmetros, encontrados através da utilização do método *GridSearch*. A Figura 17 apresenta essa comparação, mas de forma gráfica, facilitando a visualização dos diferentes valores.

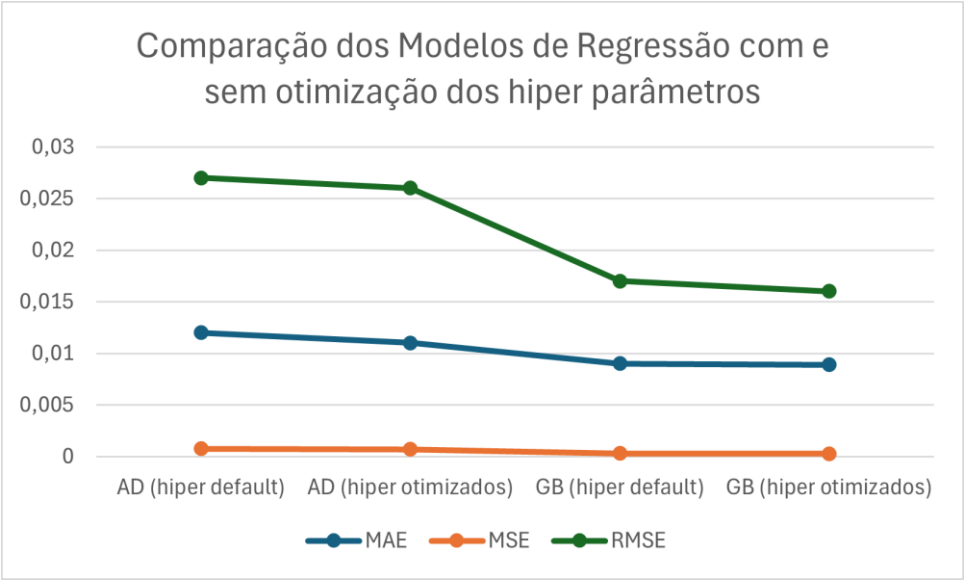

Figura 17 - Comparação dos Modelos de Regressão com e sem otimização dos hiper parâmetros

Da análise da Figura 17, constata-se que, apesar das diferenças entre as métricas serem reduzidas, a otimização dos hiper parâmetros levou à redução dos erros calculados, o que se traduz na melhoria do desempenho dos modelos ML desenvolvidos.

Tabela 14 - Comparação de resultados dos Modelos de Regressão com otimização dos hiper parâmetros e respetivos hiper parâmetros encontrados

Modelo	**Com Hiper parâmetros *default***	**Com Hiper parâmetros otimizados**	**Melhores hiper parâmetros**
AD	MAE: 0,012 MSE: 0,00076 RMSE: 0,027	MAE: 0,011 MSE: 0,0007 RMSE: 0,026	max_depth: 10 min_sample_split: 2
GB	MAE: 0,009 MSE: 0,0003 RMSE: 0,017	MAE: 0,0089 MSE: 0,00027 RMSE: 0,016	learning_rate: 0,5 max_depth: 3 n_estimators:150

Para o modelo AD, foi encontrado o hiper parâmetro "*max_depth*" com valor igual a 10. Este valor representa a profundidade ideal da árvore para evitar *overfitting* dos dados e ao mesmo tempo garantir precisão nas previsões efetuadas. O hiper parâmetro "*min_sample_split*" com valor igual a 2 fornece informações sobre o ajuste efetuado na árvore para se tornar sensível a pequenas alterações ou diferenças nos dados.

Relativamente ao modelo GB, são apresentados três hiper parâmetros. O "*learning rate*" com valor igual a 0,5 indica que o modelo adotou uma estratégia de aprendizagem em que foi equilibrado o tempo de convergência do modelo e o seu desempenho. Relativamente a "*max_depth*", o facto de possuir valor igual a 3 indica que o modelo foi ajustado para gerar árvores simples em cada uma das suas iterações, prevenindo o *overfitting* e promovendo a correção gradual dos erros. O terceiro hiper parâmetro foi o "*n_estimators*", com valor igual a 150. Este valor é representativo do valor ideal de árvores necessárias para capturar os padrões existentes nos dados sem provocar *overfitting*.

Com este passo de otimização de hiper parâmetros e a combinação dos mesmos, foi possível chegar à melhores versões dos modelos ML implementados.

4.3 Classificação do Desempenho dos Jogadores

Semelhante à comparação efetuada para o modelo de regressão, também foi comparada a performance dos diferentes modelos de classificação testados. Neste caso, as métricas de desempenho a avaliar são diferentes, tendo sido analisadas as métricas definidas na subsecção 4.1.2. O objetivo principal do desenvolvimento deste modelo foi classificar o desempenho do jogador, com base no seu valor de *GoalRatio*, em cinco classes de desempenho: 'Muito Baixo', 'Baixo', 'Médio', 'Alto' e 'Muito Alto'. A Tabela 15 apresenta a relação entre os valores de *GoalRatio* e a sua respetiva classe.

Tabela 15 - Relação entre as Classes de Desempenho e os valores de *GoalRatio*

Classe de Desempenho	**Intervalo de valores de *GoalRatio***
Muito Baixo	[0 ; 0,2[
Baixo	[0,2; 0,4[
Médio	[0,4; 0,6[
Alto	[0,6; 0,8[
Muito Alto	[0,8; 1]

De todos os modelos de classificação testados, foram selecionados os dois com melhor performance para posteriormente passarem à fase de otimização do seus hiper parâmetros. O valor das métricas de desempenho dos modelos de classificação encontra-se na Tabela 16.

Tabela 16 - Resultados obtidos pelos Modelos de Classificação

Modelo	Dados de Controlo	Normalização Min-max	Normalização Z-Score	RO	SO
RF	ACC: 91,69% PRE: 89,94% RCL: 90,82% F1: 90,38%	ACC: 88,37% PRE: 86,67% RCL: 87,89% F1: 87,28%	ACC: 88,65% PRE: 87,16% RCL: 88,22% F1: 87,69%	ACC: 90,86% PRE: 89,96% RCL: 90,73% F1: 90,16%	ACC: 91,96% PRE: 91,08% RCL: 91,24% F1: 91,16%
AD	**ACC: 92,80%** **PRE: 92,74%** **RCL: 92,48%** **F1: 92,61%**	ACC: 91,13% PRE: 91,14% RCL: 90,78% F1: 90,96	ACC: 91,41% PRE: 91,53% RCL: 90,93% F1: 91,23%	ACC: 90,30% PRE: 90,33% RCL: 89,79% F1: 90,06	ACC: 91,41% PRE: 92,20% RCL: 90,95% F1: 91,57%
SVM	ACC: 83,38% PRE: 69,52% RCL: 83,37% F1: 75,82%	ACC: 61,47% PRE: 82,69% RCL: 60,06% F1: 69,58%	ACC: 80,60% PRE: 89,68% RCL: 79,74% F1: 84,42%	ACC: 55,40% PRE: 78,74% RCL: 53,44% F1: 63,67%	ACC: 62,04% PRE: 77,51% RCL: 59,62% F1: 67,40%
kNN	ACC: 84,21% PRE: 79,39% RCL: 82,17% F1: 80,73%	ACC: 78,67% PRE: 83,98% RCL: 77,91% F1: 80,83%	ACC: 81,99% PRE: 85,88% RCL: 81,08% F1: 83,41%	ACC: 64,55% PRE: 74,20% RCL: 63,65% F1: 68,52%	ACC: 58,17% PRE: 75,21% RCL: 57,03% F1: 64,87%

GaussianNb	ACC: 79,76% PRE: 80,25% RCL: 77,40% F1: 78,80%	ACC: 65,38% PRE: 82,74% RCL: 63,95% F1: 72,14%	ACC: 65,93% PRE: 82,79% RCL: 64,52% F1: 72,52%	ACC: 65,38% PRE: 81,21% RCL: 64,29% F1: 71,77%	ACC: 73,12% PRE: 81,34% RCL: 71,60% F1: 76,16%
GB	ACC: 94,74% PRE: 94,69% RCL: 94,39% F1: 94,54%	ACC: 94,19% PRE: 93,88% RCL: 93,96% F1: 93,92%	ACC: 94,74% PRE: 94,51% RCL: 94,57% F1: 94,54%	ACC: 93,63% PRE: 92,96% RCL: 93,56% F1: 93,26%	**ACC: 95,57%** **PRE: 95,26%** **RCL: 95,46%** **F1: 95,36%**

Analisando os resultados obtidos, constata-se que a aplicação das técnicas de normalização e de balanceamento de dados provocou melhorias na performance dos modelos RF e GB. Nos restantes modelos, o impacto dessas técnicas não foi sentido. Relativamente às métricas de desempenho, os dois modelos com melhor performance foram AD (sem aplicação de técnicas de normalização e balanceamento de dados) e GB (com aplicação de *synthetic oversampling*).

O modelo AD apresentou os seguintes resultados:

- *Accuracy*: 92,80%
- Precisão: 92,74%
- *Recall*: 92,48%
- *F1-Score*: 92,61%

Esses resultados sugerem que o modelo AD está bem ajustado, apresentando um equilíbrio entre todas as métricas, sem mostrar grandes variações que indicariam problemas de *trade-off* entre precisão e *recall*. A alta *accuracy* é consistente com as outras métricas e o facto de estarem todas próximas de 92% sugere que a performance do modelo é consistente, sem grandes variações entre os diferentes aspetos do seu desempenho. Outro dado a retirar é o facto destes valores indicarem que o modelo é robusto e generaliza bem quando confrontado com novos dados.

Relativamente ao modelo GB, o seu melhor desempenho foi com a aplicação de *synthetic oversampling*, apresentando os seguintes resultados:

- *Accuracy*: 95,57%
- Precision: 95,26%
- *Recall*: 95,46%
- *F1-Score*: 95,36%

À semelhança do modelo AD, também este modelo apresenta elevados valores de desempenho. Após a aplicação da técnica *synthetic oversampling*, os resultados foram superiores quando comparado com os dados de controlo, sugerindo que a técnica foi eficaz para lidar com possíveis problemas de desbalanceamento das classes. O desempenho consistente em todas as métricas aponta para um modelo bem ajustado, com uma excelente capacidade de generalização.

Esses resultados indicam que o modelo é capaz de prever com precisão, recuperando a maioria dos exemplos relevantes e minimizando erros, tanto falsos positivos, quanto falsos negativos. Portanto, o uso desta abordagem de *oversampling* foi fundamental para alcançar esse nível de performance, tornando o modelo GB uma escolha sólida e confiável.

4.3.1 Hiper parâmetros

Seguidamente, foi aplicado o método de otimização de hiper parâmetros *GridSearch* nestes dois modelos (AD e GB), com o objetivo de perceber quais os que promovem melhores resultados nas métricas de desempenho. A Tabela 17 apresenta os parâmetros fornecidos ao método *GridSearch*.

Tabela 17 - Parâmetros fornecidos ao método GridSearch nos modelos de classificação

Modelo	Hiper parâmetros
AD	'*max_depth*': [1,5,8,10,15,20], '*min_samples_split*': [2, 6,8, 10,15,20,30]
GB	'*n_estimators'*: [50, 100, 150], '*learning_rate*': [0.05, 0.1, 0.5], '*max_depth*': [3, 5, 7]

A Tabela 18 visa apresentar os resultados da comparação entre o desempenho dos modelos selecionados da Tabela 16 com o desempenho dos modelos quando executados com os hiper parâmetros encontrados pelo método *GridSearch*. A Figura 18 apresenta essa comparação, mas de forma gráfica, permitindo visualizar de que forma os valores das diferentes métricas variaram com otimização dos hiper parâmetros.

Tabela 18 - Comparação de resultados dos Modelos de Classificação com otimização dos hiper parâmetros

Modelo	Com Hiper parâmetros ***default***	Com Hiper parâmetros otimizados	Melhores hiper parâmetros
AD	ACC: 92,80% PRE: 92,74% RCL: 92,48% F1: 92,61%	ACC: 91,69% PRE: 91,25% RCL: 91,27% F1: 91,26%	max_depth: 5 min_sample_split: 8
GB	ACC: 95,57% PRE: 95,26% RCL: 95,46% F1: 95,36%	ACC: 96,31% PRE: 95,77% RCL: 95,83% F1: 95,80%	learning_rate: 0,5 max_depth: 3 n_estimators: 150

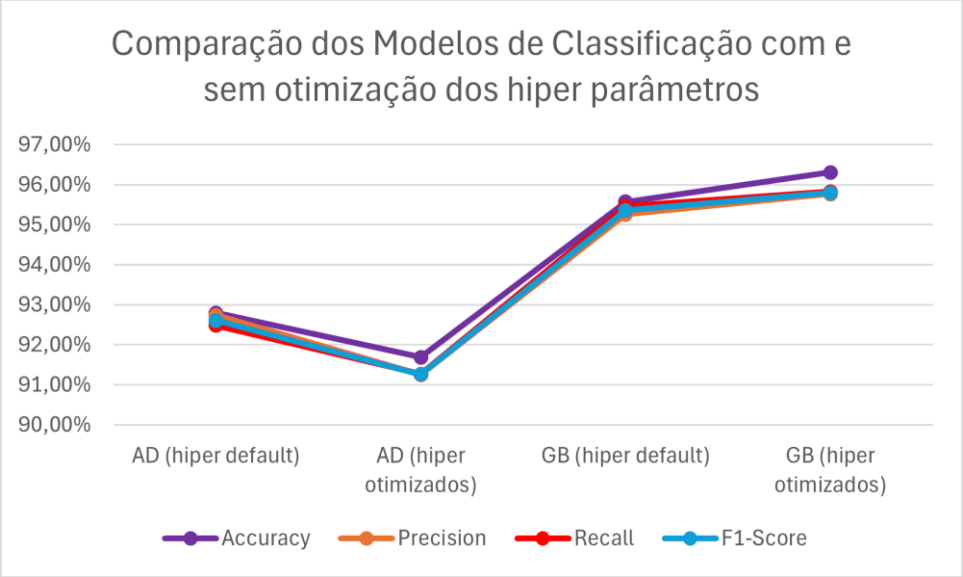

Figura 18 - Comparação dos Modelos de Classificação com e sem otimização dos hiper parâmetros

No que toca à otimização de parâmetros nos modelos de classificação, o comportamento foi distinto entre os modelos. Se, por um lado, o modelo GB apresentou melhorias nas suas métricas de desempenho, por outro, esta otimização não surtiu efeitos na melhoria da performance do modelo AD. Desta forma, foi possível encontrar a versão do modelo GB com melhor performance.

5 Conclusão

Esta dissertação tinha como objetivo tratar dois problemas: um problema de regressão, que consistia na previsão da eficácia de remates (*GoalRatio)*, e um problema de classificação, que que categorizasse os níveis de desempenho do jogador com base na eficácia dos seus remates. Ambos os objetivos tinham em comum o facto de requererem a o estudo e desenvolvimento de modelos ML.

Depois da recolha e apresentação de informação relativa a ML e à sua utilização no contexto do futebol, foram recolhidos os dados relativos a informações dos jogadores em todos os jogos de duas competições das últimas duas épocas desportivas. Foi necessário efetuar pré-processamento dos dados, selecionando as características que tipicamente impactam no cálculo de *GoalRatio*.

Em ambos os problemas de regressão e de classificação foram estudados diversos modelos ML. No que toca à previsão do valor de *GoalRatio*, foram comparados os seguintes modelos: Regressão Linear, kNN, SVM, AD, MLP e GB. Relativamente à classificação do desempenho do atleta, foram comparados os seguintes modelos: RF, AD, SVM, kNN, *GaussianNb* e GB. O desempenho dos modelos foi avaliado sob diferentes circunstâncias, com e sem aplicação de técnicas de normalização e balanceamento de dados.

Para prever o *GoalRatio* de um jogador, os modelos ML que apresentaram melhores métricas de desempenho foram AD e GB. Os valores obtidos nas métricas calculadas são bons indicadores para a previsão acertada da eficácia, contribuindo para o desenvolvimento e constante melhoria do atleta.

Em relação à classificação do desempenho do atleta, foram criadas cinco classes para categorizar o desempenho do atleta: ‘Muito Baixo’, ‘Baixo’, ‘Médio’, ‘Alto’ e ‘Muito Alto’. Dos diversos modelos avaliados, os que apresentaram melhor desempenho foram também os modelos AD e GB. Com uma percentagem de *Accuracy* acima dos 90% em ambos os modelos, os resultados apresentados revelam que os modelos construídos são capazes de classificar o desempenho dos atletas.

5.1 Trabalho Futuro

O trabalho desenvolvido no âmbito desta dissertação pode ter continuidade em diversos tópicos:

- No caso da previsão do *GoalRatio*, seria interessante perceber o impacto de outras características/eventos do jogador no jogo nessa previsão.
- Testar os modelos com um *dataset* mais alargado, com dados de outras competições futebolísticas.
- Explorar e avaliar a utilização de métodos de aprendizagem não supervisionada, com o objetivo de agrupar os jogadores com base nas suas características.
- Desenvolver uma plataforma de suporte à decisão, partindo da arquitetura apresentada em 3.1.3, onde será possível aos diferentes profissionais do futebol utilizar os modelos desenvolvidos no âmbito desta dissertação.

Referências

Abadi, M. *et al.* (2016) 'TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems', *GPU Computing Gems Emerald Edition*, (November), pp. 277–291. Available at: www.tensorflow.org. (Accessed: 31 August 2024).

Abhigyan (2021) *Cross-Validation Techniques. This article aims to explain different… | by Abhigyan | Geek Culture | Medium*, *medium.com*. Available at: https://medium.com/geekculture/cross-validation-techniques-33d389897878 (Accessed: 21 August 2024).

Akanbi, O. A., Amiri, I. S. and Fazeldehkordi, E. (2014) *A Machine-Learning Approach to Phishing Detection and Defense*, *A Machine-Learning Approach to Phishing Detection and Defense*. Elsevier Inc. doi: 10.1016/c2014-0-03762-8.

Almulla, J. and Alam, T. (2020) 'Machine Learning Models Reveal Key Performance Metrics of Football Players to Win Matches in Qatar Stars League', *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 8, pp. 213695–213705. doi: 10.1109/ACCESS.2020.3038601.

Arastey, G. M. (2019) *History of Performance Analysis: The Controversial Pioneer Charles Reep*, *Sports Performance Analysis*. Available at: https://www.sportperformanceanalysis.com/article/history-of-performance-analysis-the-controversial-pioneer-charles-reep (Accessed: 19 December 2023).

Baruah, I. (2023) 'All you need to know about encoding techniques! _ by Indraneel Dutta Baruah _ ANOLYTICS _ Medium', *Medium*. Available at: https://medium.com/anolytics/all-you-need-to-know-about-encoding-techniques-b3a0af68338b (Accessed: 18 August 2024).

Bergstra, J., Ca, J. B. and Ca, Y. B. (2012) 'Random Search for Hyper-Parameter Optimization Yoshua Bengio', *Journal of Machine Learning Research*, 13, pp. 281–305. Available at: http://scikit-learn.sourceforge.net. (Accessed: 7 September 2024).

Botchkarev, A. (2018) 'Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology', *Interdisciplinary Journal of Information, Knowledge, and Management*. Informing Science Institute, 14, pp. 45–76. doi: 10.28945/4184.

Breiman, L. *et al.* (1984) *Classification and regression trees*, *Classification and Regression Trees*. CRC Press. doi: 10.1201/9781315139470.

Breiman, L. (2001) 'Random forests', *Machine Learning*. Springer, 45(1), pp. 5–32. doi: 10.1023/A:1010933404324/METRICS.

Brownlee Jason (2020) *LOOCV for Evaluating Machine Learning Algorithms - MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/ (Accessed: 21 August 2024).

Chapman, P. *et al.* (2000) *CRISM-DM 1.0: Step-by-step data mining guide*.

Chawla, S. *et al.* (2017) 'Classification of Passes in Football Matches Using Spatiotemporal Data', *ACM Transactions on Spatial Algorithms and Systems (TSAS)*. ACM PUB27 New York, NY, USA , 3(2). doi: 10.1145/3105576.

Chen, T. and Guestrin, C. (2016) 'XGBoost: A scalable tree boosting system', in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, pp. 785–794. doi: 10.1145/2939672.2939785.

Cortes, C., Vapnik, V. and Saitta, L. (2020) *Support-Vector Networks Editor*, *Machine Leaming*. Kluwer Academic Publishers.

Cristianini, N. and Shawe-Taylor, J. (2000) 'An Introduction to Support Vector Machines and Other Kernel-based Learning Methods', *An Introduction to Support Vector Machines and Other Kernel-based*

Learning Methods. Cambridge University Press. doi: 10.1017/CBO9780511801389.

DataMB (2023) *DataMB | Compare Players*. Available at: https://datamb.football/radars/ (Accessed: 3 January 2024).

Dean, J. and Ghemawat, S. (2008) 'MapReduce: Simplified data processing on large clusters', *Communications of the ACM*. ACMPUB27New York, NY, USA, 51(1), pp. 107–113. doi: 10.1145/1327452.1327492/SUPPL_FILE/P107-DEAN.JP.PDF.

Dekking, F. *et al.* (2006) 'A Modern Introduction to Probability and Statistics: Understanding Why and How', *Journal of the American Statistical Association*, 101(473), pp. 393–394. doi: 10.1198/jasa.2006.s72.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000) *Pattern Classification*. Wiley-Interscience.

dvc.ai (no date) 'Data Version Control · DVC', *Data Version Control · DVC*. Available at: https://dvc.org/ (Accessed: 13 September 2024).

Engmann, S. and Cousineau, D. (2011) 'Quantitative Methods Inquires 1 COMPARING DISTRIBUTIONS: THE TWO-SAMPLE ANDERSON-DARLING TEST AS AN ALTERNATIVE TO THE KOLMOGOROV-SMIRNOFF TEST'.

Ezugwu, A. E. *et al.* (2021) 'Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature', *Neural Computing and Applications*. Springer Science and Business Media Deutschland GmbH, pp. 6247–6306. doi: 10.1007/s00521-020-05395-4.

Fay, M. P. and Proschan, M. A. (2010) 'Wilcoxon-Mann-Whitney or T-test? on assumptions for hypothesis tests and multiple interpretations of decision rules', *Statistics Surveys*. Institute of Mathematical Statistics, 4, pp. 1–39. doi: 10.1214/09-SS051.

FIFA 18 Complete Player Dataset (no date). Available at: https://www.kaggle.com/datasets/thec03u5/fifa-18-demo-player-dataset (Accessed: 25 March 2024).

Football Statistics and History | FBref.com (no date). Available at: https://fbref.com/en/ (Accessed: 26 March 2024).

Frencken, W. *et al.* (2011) 'Oscillations of centroid position and surface area of soccer teams in small-sided games', *European Journal of Sport Science*. Taylor & Francis Ltd, 11(4), pp. 215–223. doi: 10.1080/17461391.2010.499967.

FutbolLab (2023) *Statistical analysis of a football game: Keys to understand how this sport works - FutbolLab*. Available at: https://www.futbollab.com/en/news/the-statistical-analysis-of-a-football-match-how-numerical-information-improves-team-performance (Accessed: 28 December 2023).

Gallatin, K. and Albon, C. (2023) *Machine Learning with Python Cookbook, 2nd Edition*. Available at: https://books.google.pt/books?hl=pt-PT&lr=&id=kIhQDwAAQBAJ&oi=fnd&pg=PT107&dq=Machine+Learning+with+Python+Cookbook:+Practical+Solutions+from+Preprocessing+to+Deep+Learning&ots=OoStUQilIU&sig=hxnBeprtUhNSRRpb5zJ4XK__tWI&redir_esc=y#v=onepage&q=Machine Lea (Accessed: 19 August 2024).

Géron, A. (2017) *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, *Hands-On Machine Learning with R*. O'Reilly Media. Available at: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ (Accessed: 2 September 2024).

Ghar, S., Patil, S. and Arunachalam, V. (2021) 'Data Driven football scouting assistance with simulated player performance extrapolation', *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021*. Institute of Electrical and Electronics Engineers Inc., pp. 1160–1167. doi: 10.1109/ICMLA52953.2021.00189.

Goes, F. R. *et al.* (2019) 'Not Every Pass Can Be an Assist: A Data-Driven Model to Measure Pass Effectiveness in Professional Soccer Matches', *Big data*. Big Data, 7(1), pp. 57–70. doi: 10.1089/BIG.2018.0067.

Goodfellow, I. *et al.* (2016) 'Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning'. Deep learning The MIT Press, 800, p. 262035618. doi: 10.1007/s10710-017-9314-z.

Gosling, J. *et al.* (2005) *The Java Language Specification, Third Edition*, *Java Language Specification Third Edition*. Available at: https://www.researchgate.net/publication/200040359_The_Java_Language_Specification_Third_Edition (Accessed: 30 August 2024).

Gudmundsson, J. and Wolle, T. (2014) 'Football analysis using spatio-temporal tools', *Computers, Environment and Urban Systems*. Pergamon, 47, pp. 16–27. doi: 10.1016/J.COMPENVURBSYS.2013.09.004.

Hastie, T. *et al.* (2006) *An Introduction to Statistical Learning, Springer Texts*, *Springer Texts*. New York, NY: Springer US (Springer Texts in Statistics). doi: 10.1007/978-1-0716-1418-1.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) 'The Elements of Statistical Learning'. New York, NY: Springer New York (Springer Series in Statistics). doi: 10.1007/978-0-387-84858-7.

Herold, M. *et al.* (2019) 'Machine learning in men's professional football: Current applications and future directions for improving attacking play', *International Journal of Sports Science and Coaching*. SAGE Publications Inc., 14(6), pp. 798–817. doi: 10.1177/1747954119879350/FORMAT/EPUB.

Higgins, J. P. T. *et al.* (2019) *Cochrane handbook for systematic reviews of interventions*, *Cochrane Handbook for Systematic Reviews of Interventions*. wiley. doi: 10.1002/9781119536604.

Hleap, S. (2022) *Unmasking the Outliers: Exploring the Interquartile Range Method for Reliable Data Analysis*, *Procogia*. Available at: https://procogia.com/interquartile-range-method-for-reliable-data-analysis/ (Accessed: 13 August 2024).

Homepage - Wyscout FootballData (no date). Available at: https://footballdata.wyscout.com/ (Accessed: 26 March 2024).

Horton, M. *et al.* (2015) 'Automated classification of passing in football', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, 9078, pp. 319–330. doi: 10.1007/978-3-319-18032-8_25.

Ihaka, R. and Gentleman, R. (1996) 'R: A Language for Data Analysis and Graphics', *Journal of Computational and Graphical Statistics*, 5(3), pp. 299–314. doi: 10.1080/10618600.1996.10474713.

Infogol (2018) *Introducing A Non-Shot Expected Goals Model at Infogol. | Analysis | Infogol*. Available at: https://www.infogol.net/en/blog/analysis/introducing-non-shot-expected-goals-model (Accessed: 29 December 2023).

Jain, A. K., Murty, M. N. and Flynn, P. J. (1999) 'Data clustering', *ACM Computing Surveys (CSUR)*. ACMPUB27New York, NY, USA, 31(3), pp. 264–323. doi: 10.1145/331499.331504.

Jain, A., Nandakumar, K. and Ross, A. (2005) 'Score normalization in multimodal biometric systems', *Pattern Recognition*, 38, pp. 2270–2285. doi: 10.1016/j.patcog.2005.01.012.

Keras-Team (2017) 'keras: Deep Learning for humans', *Github*. Available at: https://keras.io/ (Accessed: 31 August 2024).

Kotsiantis, S. B. (2007) 'Supervised Machine Learning: A Review of Classification Techniques', *Informatica*, 31, pp. 249–268.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012) 'ImageNet classification with deep convolutional neural networks', *Communications of the ACM*, 60(6), pp. 84–90. doi: 10.1145/3065386.

Kuhn, M. (2008) 'Building Predictive Models in R Using the caret Package', *Journal of Statistical Software*. American Statistical Association, 28(5), pp. 1–26. doi: 10.18637/JSS.V028.I05.

Liberati, A. *et al.* (2009) 'The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration', *PLOS Medicine*. Public

Library of Science, 6(7), p. e1000100. doi: 10.1371/JOURNAL.PMED.1000100.

Lucey, P. *et al.* (2014) '"Quality vs Quantity": Improved Shot Prediction in Soccer using Strategic Features from Spatiotemporal Data', *Proc. 8th Annual MIT Sloan Sports Analytics Conference*, pp. 1–9. Available at: http://www.sloansportsconference.com/?p=15790 (Accessed: 5 January 2024).

Maaten, L., Postma, E. and Herik, J. (2008) 'Dimensionality Reduction: A Comparative Review'.

machine learning - Why is it wrong to train and test a model on the same dataset? - Data Science Stack Exchange (no date). Available at: https://www.geeksforgeeks.org/why-is-it-wrong-to-train-and-test-a-model-on-the-same-dataset/ (Accessed: 21 August 2024).

Marban, O. *et al.* (2009) 'A Data Mining & Knowledge Discovery Process Model', *Data Mining and Knowledge Discovery in Real Life Applications*. IntechOpen. doi: 10.5772/6438.

Mitchell, T. (1997) 'Machine Learning (McGraw-Hill International Editions Computer Science Series): Tom M. Mitchell: 9780071154673'. McGraw-Hill Pub. Co. (ISE Editions), p. 414.

Moher, D. *et al.* (2009) 'Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement', *PLoS Medicine*. Public Library of Science, p. e1000097. doi: 10.1371/journal.pmed.1000097.

Nazarov, K. (2023) *Overview of the Steps in a Machine Learning Pipeline*, *Linkedin*. Available at: https://www.linkedin.com/pulse/overview-steps-machine-learning-pipeline-khalid-nazarov (Accessed: 2 September 2024).

Olamendy, J. (2024) *A Comprehensive Guide to Stratified K-Fold Cross-validation for Unbalanced Data*, *medium.com*. Available at: https://medium.com/@juanc.olamendy/a-comprehensive-guide-to-stratified-k-fold-cross-validation-for-unbalanced-data-014691060f17 (Accessed: 21 August 2024).

Page, M. J. *et al.* (2021) 'The PRISMA 2020 statement: An updated guideline for reporting systematic reviews', *The BMJ*. BMJ Publishing Group, 372. doi: 10.1136/BMJ.N71.

Paszke, A. *et al.* (2019) 'PyTorch: An Imperative Style, High-Performance Deep Learning Library', *Advances in Neural Information Processing Systems*, 32.

Pedregosa, F. *et al.* (2011) 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research*, 12, pp. 2825–2830. Available at: http://arxiv.org/abs/1201.0490 (Accessed: 31 August 2024).

Power, P. *et al.* (2017) '"Not all passes are created equal:" Objectively measuring the risk and reward of passes in soccer from tracking data', *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, Part F129685, pp. 1605–1613. doi: 10.1145/3097983.3098051.

Pramod, O. (2023) *Cross Validation. Cross-validation is a technique for… | by om pramod | Medium*, *medium.com*. Available at: https://medium.com/@ompramod9921/cross-validation-623620ff84c2 (Accessed: 21 August 2024).

Python Package Introduction — xgboost 2.0.3 documentation (no date). Available at: https://xgboost.readthedocs.io/en/stable/python/python_intro.html (Accessed: 1 April 2024).

Rathke, A. (2017) 'An examination of expected goals and shot efficiency in soccer', *Journal of Human Sport and Exercise*. Universidad de Alicante Servicio de Publicaciones, 12(Proc2). doi: 10.14198/jhse.2017.12.proc2.05.

Rommers, N. *et al.* (2020) 'A Machine Learning Approach to Assess Injury Risk in Elite Youth Football Players', *Medicine and science in sports and exercise*. Med Sci Sports Exerc, 52(8), pp. 1745–1751. doi: 10.1249/MSS.0000000000002305.

van Rossum, G. and Drake, F. L. (2009) 'PYTHON 2.6 Reference Manual'. CreateSpace.

Sahami, M. *et al.* (1998) 'A Bayesian approach to filtering junk e-mail', *Learning for Text Categorization: Papers from the AAAI Workshop*, WS-98-05(Cohen), pp. 55–62. Available at: http://research.microsoft.com/en-us/um/people/horvitz/junkfilter.htm (Accessed: 27 August 2024).

scikit-learn: machine learning in Python — scikit-learn 1.5.0 documentation (no date). Available at: https://scikit-learn.org/stable/ (Accessed: 23 March 2024).

Singh, D. and Singh, B. (2020) 'Investigating the impact of data normalization on classification performance', *Applied Soft Computing*. Elsevier, 97, p. 105524. doi: 10.1016/J.ASOC.2019.105524.

Snoek, J., Larochelle, H. and Adams, R. P. (2012) 'Practical Bayesian Optimization of Machine Learning Algorithms', *Advances in Neural Information Processing Systems*, 25.

Soccermatics (2022a) *Statistical scouting — Soccermatics documentation*. Available at: https://soccermatics.readthedocs.io/en/latest/lesson3/ScoutingPlayers.html (Accessed: 3 January 2024).

Soccermatics (2022b) *Visualising football — Soccermatics documentation*. Available at: https://soccermatics.readthedocs.io/en/latest/lesson1/VisualisingFootball.html (Accessed: 29 December 2023).

Sokolova, M. and Lapalme, G. (2009) 'A systematic analysis of performance measures for classification tasks', *Information Processing & Management*. Pergamon, 45(4), pp. 427–437. doi: 10.1016/J.IPM.2009.03.002.

SRIJ (2023) *Relatório 2º trimestre 2023: Registo da atividade de jogo online em Portugal*. Available at: www.srij.turismodeportugal.pt (Accessed: 26 December 2023).

Stats Perform (2022) *Opta data from Stats Perform*. Available at: https://www.statsperform.com/opta/ (Accessed: 28 December 2023).

Statsbomb (2018) *What is xG? How is it calculated? | StatsBomb | Data Champions*, *Soccer Metrics*. Available at: https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/ (Accessed: 29 December 2023).

Terra, J. (2014) 'Regression vs. Classification in Machine Learning for Beginners', pp. 1–2. Available at: https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article (Accessed: 28 August 2024).

Transfermarkt (2020) *Pass completion rates: PSG dominate top 20 - Otamendi best in Premier League with 92.4%*. Available at: https://www.transfermarkt.com/pass-completion-rates-psg-dominate-top-20-otamendi-best-in-premier-league-with-92-4-/view/news/358096 (Accessed: 3 January 2024).

Wirth, R. and Hipp, J. (2000) 'CRISP-DM : Towards a Standard Process Model for Data Mining', *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, (24959), pp. 29–39.

xfb Analytics (2022) *The history of football analytics - Part 2*, *xfb Analytics*. Available at: http://www.xfbanalytics.hu/blog/blog-post/32 (Accessed: 19 December 2023).

XValue (2024) *Premier League 2023/2024 top stats and rankings by teams | Soccerment*. Available at: https://xvalue.ai/stats/en/league/premier_league/teams (Accessed: 29 December 2023).

Yadav, S. and Shukla, S. (2016) 'Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification', *Proceedings - 6th International Advanced Computing Conference, IACC 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 78–83. doi: 10.1109/IACC.2016.25.