UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LUIZA STRINGHINI LINHARES

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA ANÁLISE DE PARTIDAS DE FUTEBOL

PATO BRANCO

2024

LUIZA STRINGHINI LINHARES

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA ANÁLISE DE PARTIDAS DE FUTEBOL

APPLICATION OF MINING TECHNIQUES DATA FOR ANALYSIS OF FOOTBALL MATCHES

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Dalcimar Casanova

Coorientador: Prof. Dr. Érick Oliveira Rodrigues

PATO BRANCO

2024

LUIZA STRINGHINI LINHARES

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS PARA ANÁLISE DE PARTIDAS DE FUTEBOL

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: Maio/2024

Dalcimar Casanova
Dr. Física
Universidade Tecnológica Federal do Paraná

Jefferson Tales Oliva
Dr. Ciência da Computação e Matemática Computacional
Universidade Tecnológica Federal do Paraná

Viviane Dal Molin
Dra. Informática
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2024

Dedico essa monografia aos meus Pais, por todo incentivo e apoio ao longo dessa caminhada.

AGRADECIMENTOS

Primeiramente a Deus, por ter me concedido saúde e força para superar as dificuldades e chegar até aqui.

À minha família, pelo apoio constante, pela ajuda inestimável e por proporcionarem um ambiente adequado para meus estudos.

Aos meus amigos, especialmente a Edinei Martinello, cuja bondade infinita e companheirismo foram essenciais para eu alcançar este objetivo.

Ao meu orientador, Prof. Dr. Dalcimar Casanova, e ao meu coorientador, Prof. Dr. Érick Oliveira Rodrigues, pela orientação dedicada, pelo apoio nos momentos de dificuldade e por acreditarem no potencial deste trabalho.

À Universidade Tecnológica Federal do Paraná (UTFPR), por possibilitar a realização da minha graduação em uma instituição pública de qualidade.

Se podemos sonhar, também podemos tornar
nossos sonhos realidade - Walt Disney

RESUMO

O interesse em prever os resultados de partidas de futebol permanece relevante atualmente, especialmente devido ao crescente mercado de apostas esportivas no Brasil e no mundo. Para determinar o favoritismo em um confronto, uma série de fatores é considerada, incluindo estatísticas dos jogadores, histórico de confrontos e desempenho na temporada. No entanto, a distância percorrida pelos times ao longo da temporada para jogos fora de casa é frequentemente negligenciada. Neste estudo, é investigada a influência da distância percorrida pelos times nos resultados das partidas da *Premier League*, utilizando técnicas de mineração de dados. O processo envolveu a criação de um data *warehouse* com informações de desempenho e dados geográficos, complementado com dados de competições de meio de semana. Foram definidos cenários e aplicado o algoritmo de regressão logística para treinar o modelo. Realizando a análise dos resultados, foi observado que a distância percorrida pelos times não teve um impacto significativo nos resultados das partidas. A comparação com os dados das casas de apostas mostrou uma acurácia semelhante, indicando que outros fatores podem ser mais relevantes na determinação dos resultados. Conclui-se que, embora a distância percorrida pelos times seja relevante na logística das competições, não parece ser um fator determinante nos resultados da *Premier League*.

Palavras-chave: mineração de dados; futebol; aprendizado de máquina.

ABSTRACT

The interest in predicting the outcome of soccer matches remains relevant today, especially given the growing sports betting market in Brazil and around the world. To determine favoritism in a match, a number of factors are considered, including player statistics, history of matches and performance during the season. However, the distance traveled by teams throughout the season for away games is often overlooked. In this study, we investigated the influence of the distance traveled by teams on the results of Premier League matches, using data mining techniques. The process involved creating a data warehouse with performance information and geographical data, supplemented with data from midweek competitions. Scenarios were defined and the logistic regression algorithm was applied to train the model. Analysis of the results showed that the distance traveled by the teams had no significant impact on match results. The comparison with bookmaker data showed similar accuracy, indicating that other factors may be more relevant in determining results. In conclusion, although the distance traveled by the teams is relevant to the logistics of competitions, it does not seem to be a determining factor in Premier League results.

Keywords: data mining; soccer; machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Abreviaturas

FA Cup	*Football Association Cup*

Siglas

API	*Application Programming Interface*
B365A	Bet365 *away win odds*
B365D	Bet365 *draw odds*
B365H	Bet365 *home win odds*
FTAG	*Full Time Away Goals*
FTHG	*Full Time Home Goals*
FTR	*Full Time Result*
HTAG	*Half Time Away Goals*
HTHG	*Half Time Home Goals*
HTR	*Half Time Result*
IQR	*Interquartil range*
KDD	*Knowledge Discovery in Databases*
KNN	*k-Nearest Neighbors*
SVM	*Support Vector Machines*
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO

O desporto de equipe mais difundido globalmente é o futebol, tendo sua origem na Inglaterra ao longo do século XIX. Contudo, registros históricos sugerem a existência prévia de atividades desportivas semelhantes. A formação do futebol moderno resultou da colaboração entre clubes descontentes com certas regulamentações do rugby, os quais optaram por se unir na criação de uma modalidade em que a bola não fosse manipulada pelas mãos. As diretrizes desse novo desporto, denominado futebol, foram formalizadas pela Universidade de Cambridge em 1846 (Silva, 2024).

Quando o futebol se tornou profissional, em 1933, o interesse em tentar antecipar os resultados das partidas cresceu significativamente. A busca por conhecer os desfechos dos eventos esportivos antes de sua conclusão tornou-se evidente para a indústria de apostas, ao passo que a tecnologia experimentou um rápido avanço, proporcionando uma compreensão aprimorada. A Inteligência Artificial, por sua vez, está transformando a maneira como encaramos a previsão de resultados esportivos (Fialho, 2021).

Mineração de dados é parte integrante de um processo mais amplo, conhecido como descoberta de conhecimento em bases de dados, *Knowledge Discovery in Databases* (KDD). Embora muitos usem mineração de dados como sinônimo de KDD, foi proposto que a terminologia descoberta de conhecimento em bases de dados se referisse a todo o processo de extração de conhecimentos a partir de dados. Foi proposto também que a terminologia mineração de dados fosse empregada exclusivamente para a etapa de descoberta do processo de KDD, que inclui a seleção e integração das bases de dados, limpeza da base, transformação dos dados e a avaliação dos dados (Ferrari; Silva, 2016).

Para fazer a análise da partida, levando em consideração que pode ocorrer, vitória do mandante, vitória do visitante ou empate, é comum serem levados em conta os aspectos como: estatísticas dos jogadores, lesões ou histórico de confrontos. Pouco se fala da quantidade excessiva de viagens que os jogadores são submetidos para realizar as partidas fora de casa.

Neste sentido, o objetivo desse trabalho é investigar a influência da distância percorrida pelos times nos resultados das partidas da *Premier League*, utilizando técnicas de mineração de dados, como: coleta dos dados, pré-processamento, aprendizado de máquina, métodos de avaliação, entre outras; para chegar a uma conclusão do problema a ser explorado.

Para isso, é coletada uma base de dados da *Premier League* e executadas as etapas de mineração de dados, para a criar um *data warehouse* com informações de desempenho e dados geográficos, complementado com dados de competições de meio de semana, os quais foram definidos cenários, aplicado o algoritmo de regressão logística para treinar o modelo e medido o desempenho através da acurácia para avaliar o impacto que a distância tem sobre as partidas.

1.1 Objetivos

1.1.1 Objetivo geral

Investigar a influência da distância percorrida pelos times em resultados de partidas de futebol da *Premier League*.

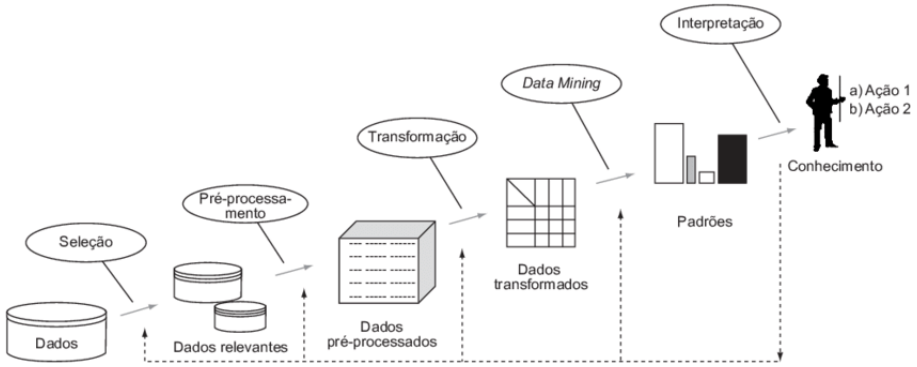
1.1.2 Objetivos específicos

- Montar um *data warehouse* com informações de desempenho e dados geográficos;
- Determinar a influência da distância percorrida pelos times entre partidas, no resultado final do jogo.

2 REFERENCIAL TEÓRICO

Neste capítulo, busca-se uma análise detalhada das diversas etapas que compõem o processo de mineração de dados, conforme ilustrado na Figura 1. A Seção 2.1 explora os variados métodos de seleção de dados, destacando suas implicações e aplicabilidades. Em seguida, na Seção 2.2, são discutidas as estratégias de pré-processamento dos dados para aumentar a qualidade dos dados para a etapa subsequente. Por fim, na Seção 2.3, são apresentados os fundamentos do aprendizado de máquina, delineando os conceitos fundamentais que serão empregados na resolução do problema em questão.

Figura 1 – Etapas da mineração de dados



Fonte: (Fayyad; Piatetsky-Shapiro; Smyth, 1996).

Na Figura 1 é ilustrado, o processo de mineração de dados, iniciando com a seleção dos dados, seguindo com o pré-processamento dos mesmos, já com a base de dados estruturada podem ser realizadas transformações na base de dados, como, por exemplo, o cálculo de novas características, para então ser aplicados métodos de aprendizado de máquina e os resultados serem analisados conforme os valores obtidos.

2.1 Seleção de dados

Coletar os dados é uma tarefa exaustiva onde existem diversos desafios desde a descoberta da fonte a ser escolhida, pois dados ausentes, duplicados ou com erros de medição podem levar a uma análise equivocada e que não será condizente com a realidade. A localização dos dados pode estar em diversas fontes, dentre elas: Banco de dados encontrados na *internet*, de uma máquina ou até mesmo dados coletados à mão, o que precisa estar bem definido é a finalidade esperada. Das fontes citadas anteriormente, sites da internet são as mais utilizadas, e a principal dificuldade é dissemelhança, pois existem diversos tipos de páginas tais como: documentos, artigos técnicos e planilhas. Para a realização da coleta neste ambiente, é comum a utilização de ferramentas de apoio, como Motores de Busca Baseados em Robô (*Robotic Internet Search Engines*) e Diretórios de Assunto (*Subject Directories*) (Carrilho, 2007).

Quando se deseja extrair dados de um site, é comum criar programas chamados de *Crawler* ou *Web Crawler*, cujo objetivo é coletar automaticamente os dados armazenados em uma página. No entanto, essa coleta pode incluir dados irrelevantes, resultando em desperdício de processamento (Mortari, 2022).

2.2 Pré-Processamento

Na grande maioria dos estudos a serem realizados os dados brutos devem ser processados para torná-los adequados para análise. Embora o objetivo seja melhorar a qualidade dos dados, diversas técnicas podem ser aplicadas sobre os dados para que eles se ajustem melhor a uma técnica ou ferramenta de mineração de dados específica. (Tan *et al.*, 2006).

Neste capítulo são apresentados técnicas que podem ser aplicadas para melhorar a qualidade dos dados, dentre elas: limpeza, integração e transformação dos dados.

2.2.1 Limpeza de dados

Essa etapa envolve uma verificação da consistência das informações, excluindo valores nulos, redundantes e atípicos, os dados são identificados e caso sejam encontrados valores duplicados, são removidos da base. A limpeza de dados é um problema que afeta a maior parte das bases de dados reais e pode ser feita caso os dados sejam ausentes ou pouco detalhados. Essa etapa é essencial para reduzir o tempo de processamento, eliminar consultas irrelevantes e evitar resultados não confiáveis (Goldschmidt; Passos, 2005).

Valores ausentes

O tratamento incorreto ou a eliminação de objetos com valores ausentes pode ocasionar erros das ferramentas de análise, como forma de evitar isso, a substituição de valores ausentes, também chamada de imputação, que pode ser estimada ou atribuída, tem como objetivo estimar valores ausentes com base nas informações disponíveis no conjunto de dados. A imputação de valores ausentes assume que essa ausência de valor implica a perda de informação relevante de algum atributo, mas que o valor a ser inserido não deve somar nem subtrair informação à base, em outras palavras, não deve enviesar a base, pois pode vir a afetar os resultados do algoritmo de mineração (Ferrari; Silva, 2016).

Para lidar com esse tipo de problema, podem ser aplicadas as seguintes técnicas de acordo com Han, Pei e Kamber (2011):

- Descartar a entrada: Essa abordagem é comum quando falta a etiqueta de classe (assumindo que a tarefa de mineração é de classificação). No entanto, essa técnica não é eficaz, a menos que a entrada contenha muitos atributos com valores ausentes;

- Preenchimento manual dos valores ausentes: Geralmente, essa abordagem é demorada e pode não ser viável em conjuntos de dados grandes com muitos valores ausentes;
- Substituição por uma constante global: Todos os valores de atributos ausentes são substituídos pela mesma constante como "Desconhecido", essa abordagem pode ser simples mas o programa de mineração pode interpretar erroneamente que formam uma conceito interessante pois todos têm um valor em comum;
- Substituição pela média ou mediana dos valores pertencentes à mesma classe da entrada
- Utilização do valor mais provável para preencher o valor ausente: Isso pode ser determinado através de regressão, ferramentas baseadas em inferência usando formalismo bayesiano ou árvores de decisão.

Dados Ruidosos

O termo “dados ruidosos” refere-se a valores inconsistentes, imprecisos ou irrelevantes que estão presentes nos conjuntos de dados. Esses valores podem ser resultado de erros de entrada, falhas no sistema de medição que podem distorcer a análise e prejudicar a precisão do modelo (Han; Pei; Kamber, 2011).

Para lidar com dados ruidosos, existem técnicas de suavização, dentre as apresentadas por Han, Pei e Kamber (2011) estão:

- *Binning*: Suaviza um valor de dados classificado consultando sua “vizinhança”, ou seja, os valores ao seu redor. Os valores classificados são distribuídos em várias caixas. Como os métodos de *binning* consultam a vizinhança dos valores, eles realizam suavização local;
- Regressão: uma técnica que adapta os valores dos dados a uma função. A regressão linear envolve encontrar a “melhor” linha para ajustar dois atributos (ou variáveis) para que um atributo possa ser usado para prever o outro.

Dados Inconsistentes

A inconsistência de um dado está diretamente relacionada à sua discrepância em relação aos outros, como uma informação preenchida com uma palavra diferente, mas que possui o mesmo significado. Outro caso é quando o valor está fora do domínio correspondente, do mesmo modo que os dados ruidosos a inconsistência pode ser facilmente encontrada através da representação gráfica dos atributos (Goldschmidt; Passos, 2005).

2.2.2 Redução da dimensionalidade

Na maioria das vezes, imagina-se que quanto maior a quantidade de atributos e objetos melhor, mas a redução da dimensionalidade oferece alguns benefícios como um desempenho aprimorado de algoritmos de mineração de dados, isso ocorre devido à capacidade de eliminar características irrelevantes e reduzir o ruído nos dados, e também devido à chamada "maldição da dimensionalidade". Outro benefício é que a redução da dimensionalidade pode resultar em modelos mais compreensíveis e também possibilita uma visualização mais fácil dos dados, ajudando a identificar padrões, tendências e *outliers* de maneira mais clara (Tan *et al.*, 2006).

Amostragem

A amostragem pode ser empregada como uma técnica de redução de dados, pois permite que um extenso conjunto de dados seja representado por uma amostra (ou subconjunto) consideravelmente menor e aleatória (Tan *et al.*, 2006).

Dentre os modos de amostragem pode-se destacar os seguintes, segundo Goldschmidt e Passos (2005):

- Amostragem de *Clusters*: fundamenta-se em organizar a base de dados em X *clusters* de modo que possa ser realizada uma amostragem entre os *clusters* de forma aleatória;
- Amostragem aleatória sem reposição: em que uma amostra com X objetos distintos é retirada aleatoriamente da base;
- Amostragem estratificada: para esse tipo de redução a base de dados deve ser dividida em grupos disjuntos. Esse tipo de amostra fundamenta-se em selecionar aleatoriamente um subconjunto de amostras de cada grupo, auxiliando na obtenção de amostras representativas em casos que os dados forem enviesados ou tendenciosos.

Discretização

A discretização dos dados é uma técnica útil para reduzir a complexidade de um atributo, como a idade, que pode ser substituída por intervalos de valores (por exemplo, 0-10, 11-20) ou rótulos conceituais (como jovem, adulto, sênior). Estes rótulos podem então ser organizados de forma recursiva em conceitos mais amplos, formando uma hierarquia de conceitos para o atributo numérico. Esta abordagem torna a representação do conhecimento mais concisa e facilita a interpretação dos resultados da mineração, resultando em aprendizado mais eficiente e resultados mais claros (Ferrari; Silva, 2016).

2.2.3 Transformação dos dados

Durante esta fase de pré-processamento, os dados são remodelados ou ajustados para aprimorar a eficiência do processo subsequente de mineração. Esta abordagem visa facilitar a compreensão dos padrões identificados, tornando-os mais acessíveis para análise (Han; Pei; Kamber, 2011).

Normalização

Essa etapa tenta dar a todos os atributos um peso igual, ou seja, após a normalização, todos os atributos contribuem de maneira mais equilibrada para o processo de análise. A normalização é útil para algoritmos de classificação envolvendo redes neurais ou medições de distância, como classificação e agrupamento do vizinho mais próximo. Para métodos baseados em distância, a normalização ajuda a evitar que atributos com intervalos inicialmente grandes superem atributos com intervalos inicialmente menores. A normalização pode ocorrer de diversas formas, dentre elas: normalização min-max, normalização z, normalização pelo escalonamento decimal e normalização pelo range interquartil (Han; Pei; Kamber, 2011).

Para a normalização min-max é feita uma transformação linear dos dados originais. Sabendo que max_a é o valor máximo e min_a é o valor mínimo do atributo A. A normalização min-max mapeia um valor a em um valor a' no domínio $[novomin_a, novomax_a]$, fazendo com que a equação fique da seguinte forma:

$$a' = \frac{a - min_a}{max_a - min_a} * (novomax_a - novomin_a) + novomin_a \quad (1)$$

A normalização min-max preserva as relações entre os valores de dados originais, isto é, encontrará um erro “fora dos limites” caso um dado de entrada futuro para normalização cair fora do intervalo de dados original para a (Han; Pei; Kamber, 2011).

Segundo Ferrari e Silva (2016) a normalização z também é denominada por normalização de média zero ou normalização por desvio-padrão, onde os valores de um atributo A são normalizados tendo como referência a média e o desvio padrão do atributo, na equação a seguir $\bar{a}$ é a média, e σ o desvio padrão do atributo a ser normalizado.

$$a' = \frac{(a - \bar{a})}{\sigma} \quad (2)$$

Comparando com o método anterior, esse método é útil quando há *outliers* ou os valores mínimos e máximos não são conhecidos (Ferrari; Silva, 2016).

Outro tipo de normalização é a pelo escalonamento decimal, que fundamenta-se em mover a casa decimal dos valores do atributo a. O número de casas movidas depende do valor máximo absoluto do atributo a, na equação, j é o menor inteiro de modo que $max(|a'|) < 1$ (Han; Pei; Kamber, 2011).

Dessa forma, a equação que representa a normalização pelo escalonamento decimal é definida da seguinte maneira:

$$a' = \frac{a}{10^j} \tag{3}$$

O último método de normalização definido por Ferrari e Silva (2016), é o pelo intervalo interquartil (*Interquartil range* (IQR)) que toma cada valor do atributo, subtrai a mediana e divide pelo intervalo interquartil, só quartis de um atributo ordenado são os três pontos que dividem o domínio do atributo em quatro grupos de cardinalidades iguais, o quartil Q1, é definido como o ponto central entre o menor valor e a mediana, o quartil Q2 é a mediana que divide o conjunto de dados ordenados em dois subconjuntos, cada um com metade da quantidade total dos dados, o terceiro quartil, Q3, é o valor do meio entre a mediana e o maior valor do atributo. Logo, a normalização pelo intervalo interquartil é dada por:

$$a' = \frac{a - Q_2}{Q_3 - Q_1} \tag{4}$$

2.3 Aprendizado de máquina

Com o avanço tecnológico, a interação com dados passou por uma transformação. Hoje em dia, os dados estão intrinsecamente presentes na vida cotidiana, seja em cada transação de compra *online*, em postagens nas redes sociais ou em pesquisas realizadas, resultando na geração de novos dados. Esse crescimento na produção de dados não apenas apresenta desafios, mas também oferece oportunidades significativas para compreender e antecipar o comportamento humano (Alpaydin, 2014).

Nesse contexto, o aprendizado de máquina emerge como uma abordagem para lidar com a complexidade e o volume dos dados gerados. Em vez de depender unicamente de algoritmos predefinidos, o aprendizado de máquina capacita os sistemas computacionais a identificar padrões e *insights* diretamente dos dados, adaptando-se e aprimorando-se com a experiência acumulada, invés de exigir que humanos derivem as regras e construam modelos manualmente. Essa capacidade analítica e preditiva torna-se essencial em diversos domínios, desde o varejo até a medicina e a ciência, possibilitando tomadas de decisão mais precisas e informadas (Alpaydin, 2014).

Na área de aprendizado de máquina, o método geralmente se divide em três categorias principais: aprendizado supervisionado, não supervisionado e por reforço. O aprendizado supervisionado envolve o processo de aprender a partir de um conjunto conhecido de dados de entrada e saída, onde o modelo busca identificar padrões e relacionamentos entre as entradas e as saídas esperadas. Por outro lado, o aprendizado não supervisionado visa descobrir padrões intrínsecos nos dados de entrada, sem o auxílio de saídas rotuladas. Enquanto isso, o apren-

dizado por reforço visa adquirir conhecimento sobre a melhor maneira de agir ou se comportar em um ambiente dinâmico, com base em sinais de recompensa ou punição (Murphy, 2012).

Nesta seção, é dado um enfoque ao aprendizado supervisionado, explorando suas técnicas, algoritmos e aplicações relevantes para o contexto da pesquisa.

2.3.1 Aprendizado supervisionado

Como introduzido na Seção 2.3, o aprendizado supervisionado é empregado sempre que há a necessidade de prever um resultado determinado a partir de uma entrada fornecida, contando com exemplos de pares de entrada-saída corretos (Murphy, 2012). De maneira geral, há dois tipos de problemas de aprendizado de máquina supervisionado: classificação e regressão.

Na abordagem de classificação, um algoritmo é utilizado para separar de maneira precisa os dados de teste em categorias específicas. Alguns exemplos de algoritmos de classificação incluem classificadores lineares (*Linear models*), máquinas de vetores de suporte (*Support Vector Machines* (SVM)), árvores de decisão (*Decision trees*), *k-Nearest Neighbors* (KNN) e floresta aleatória (*Random Forest*). Essencialmente, o algoritmo identifica entidades particulares no conjunto de dados e busca tirar conclusões sobre como essas entidades devem ser rotuladas ou definidas (IBM, 2024).

Diferente da abordagem de classificação, a regressão é o processo de estimar a relação entre uma variável dependente contínua e uma ou mais variáveis independentes, com o objetivo de prever ou explicar o valor da variável dependente com base nas variáveis independentes fornecidas. Por meio da análise dos dados, os modelos de regressão procuram identificar padrões e relações subjacentes nos dados de treinamento, permitindo que façam previsões ou inferências sobre os valores futuros da variável dependente quando novos conjuntos de variáveis independentes são introduzidos. Os algoritmos de regressão são desenvolvidos para aprender essas relações a partir dos dados de treinamento e, posteriormente, aplicar esse conhecimento para fazer previsões sobre novos dados. Dentre os principais algoritmos de regressão estão: regressão linear simples e múltipla, regressão polinomial e regressão logística (Müller; Guido, 2017).

A seguir é apresentado com mais detalhes o algoritmo de regressão logística que é utilizado para implementação do problema proposto.

2.3.2 Regressão Logística

Os métodos de regressão são agora fundamentais em qualquer análise de dados interessada em descrever a relação entre uma variável de resposta e uma ou mais variáveis explicativas (Hosmer; Lemeshow, 2000).

Na regressão é calculada a probabilidade de um evento acontecer, sendo que a variável a ser prevista pode ser de classe binária ou multiclasse (Raschka, 2015).

Regressão Logística Binária

Na regressão logística binária, quando a variável alvo tem duas classes, segundo (Bishop, 2006) a função sigmoide logística é utilizada para transformar a saída da regressão linear em uma probabilidade entre 0 e 1 é dada por:

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \tag{5}$$

Onde a é a combinação linear das características da amostra, multiplicadas pelos pesos do modelo. No cálculo da probabilidade de pertencer a uma classe $C1$ a equação é modelada utilizando a função sigmoide aplicada à combinação linear das características, conforme descrito na Equação 6 (Bishop, 2006).

$$p(C_1|\phi) = \sigma(w^T\phi) \tag{6}$$

Onde $\sigma(w^T\phi)$ é o produto escalar entre o vetor de pesos w e o vetor de características (ou variáveis independentes) ϕ (Bishop, 2006).

Na estimação dos parâmetros do modelo de regressão logística a máxima verossimilhança é um método estatístico utilizado, para isso a derivada da função sigmoide, que consiste na multiplicação da própria função sigmoide por $(1 - \sigma)$, representada na Equação 7, é necessária para este método (Bishop, 2006).

$$\frac{d\sigma}{da} = \sigma(1 - \sigma). \tag{7}$$

A função de verossimilhança é definida para um conjunto de dados σ_n, t_n, onde t_n é o rótulo da classe da amostra (que pode ser 0 ou 1) e σ_n é o vetor de características da amostra n. A Equação 8 representa a função de verossimilhança que é descrita como uma multiplicação das probabilidades das classes observadas nos dados (Bishop, 2006).

$$p(t|w) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n} \tag{8}$$

Onde $t = (t_1,...,t_N)$ e $y_n = p(C_1|\phi)$, para determinar os parâmetros do modelo, é comum definir uma função de erro que é o negativo do logaritmo da função de verossimilhança, o que resulta na função de erro de entropia cruzada (*cross-entropy*), definida na Equação 9, como a soma das diferenças entre os rótulos reais e as probabilidades previstas, ponderadas

pelo logaritmo das probabilidades previstas para a classe verdadeira e para a classe falsa, respectivamente, para cada amostra do conjunto de dados (Bishop, 2006).

$$E(w) = -\ln p(t|w) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \tag{9}$$

na qual $y_n = \sigma(a_n)$ e $a_n = w^T \phi_n$. Com o objetivo de otimizar os parâmetros do modelo de regressão logística durante o treinamento, é calculado o gradiente da função de erro para cada ponto de dados individualmente e, em seguida, somado para todos os pontos de dados (Bishop, 2006).

$$\nabla E(w) = \sum_{n=1}^{N} (y_n - t_n)\phi_n \tag{10}$$

onde:

- $(y_n - t_n)$ representa a diferença entre a probabilidade prevista y_n para a classe $C1$ e o rótulo verdadeiro t_n para o ponto de dados n;
- ϕ: vetor de variáveis independentes.

Regressão Logística Multiclasse

Nos problemas onde a variável preditiva é mais que duas classes, é denominada regressão logística multiclasse, a função *softmax* é utilizada para modelar a probabilidade de pertencer a cada classe em relação às variáveis independentes, onde para cada classe exista uma função logística criada. A função *softmax* normaliza as saídas dessas funções logísticas para que a soma das probabilidades de todas as classes seja igual a 1 (Bishop, 2006).

$$p(C_k|\phi) = y_k(\phi) = \frac{e^{a_k}}{\sum_j e^{a_j}}, \quad \text{onde } a_k = w_k^T \phi \tag{11}$$

Na Equação 11 é representada a probabilidade de pertencer à classe C_k dado um conjunto de variáveis independentes ϕ, onde w_k é o vetor de pesos associados à classe k (Bishop, 2006).

Para a determinação dos parâmetros w_k, Bishop (2006) cita que é possível utilizar máxima verossimilhança, contudo para utilizar esse método é necessário o cálculo das derivadas das probabilidades previstas y_k em relação às ativações a_j do modelo como pode ser visto na Equação 12, sabendo-se que I_{kj} são elementos da matriz identidade.

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j) \tag{12}$$

A etapa seguinte envolve a definição da função de verossimilhança utilizando o esquema de codificação $1 - of - K$. Neste esquema, o vetor alvo t_n para um vetor de características ϕ_n pertencente à classe C_k é um vetor binário com todos os elementos sendo zero, exceto o elemento k, que é igual a um. Na Equação 13 é apresentada a definição da função de verossimilhança (Bishop, 2006).

$$p(T|w_1, \ldots, w_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(C_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}} \tag{13}$$

Onde $y_{nk} = y_k(\phi_n)$ e T é uma matriz NXK de variáveis alvo com elementos t_{nk}.

A função de erro da regressão logística multiclasse, denominada função de erro de entropia cruzada (*cross-entropy*) é definida como o negativo logaritmo da função de verossimilhança, como mostra a Equação 14 (Bishop, 2006).

$$E(w_1, \ldots, w_K) = -\ln p(T|w_1, \ldots, w_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk} \tag{14}$$

Na sequência é definido o gradiente da função de erro, que consiste em uma medida da inclinação da função de erro em relação a cada parâmetro do modelo, ou seja, é executado para atualizar os parâmetros durante o treinamento do modelo. que é calculado como a soma dos erros multiplicados pelas características correspondentes das amostras, conforme descrito na Equação 15 (Bishop, 2006).

$$\nabla_{w_j} E(w_1, \ldots, w_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \phi_n \tag{15}$$

Onde:

- N: número de amostras no conjunto de treinamento;
- y_{nj}: valor previsto da probabilidade da classe j para a amostra n;
- t_{nj}: Valor do alvo para amostra n na classe j;
- ϕ_n: vetor de características da amostra n.

Segundo Bishop (2006), o cálculo do gradiente segue um padrão semelhante encontrado em outros algoritmos de aprendizado de máquina como regressão linear e regressão logística binária.

2.4 Métodos de validação

Para que o modelo não apenas memorize os dados de treinamento, mas também aprenda padrões úteis que possam ser generalizados para novos dados, é necessário adotar métodos de validação apropriados (Raschka, 2015).

Nesta seção, é apresentado o método validação cruzada *K-Fold*, utilizado para a implementação do problema proposto.

2.4.1 Validação Cruzada *K-Fold*

O processo inicia-se com a aleatória divisão do conjunto de dados em k subconjuntos sem substituição, formando assim um conjunto de treinamento, onde k-1 subconjuntos são destinadas ao treinamento do modelo e uma dobra é designada para teste. Essa etapa é repetida k vezes, resultando em k modelos e suas respectivas estimativas de desempenho (Raschka, 2015).

Posteriormente, é realizado o cálculo da média de desempenho dos modelos com base em múltiplas subconjuntos independentes, proporcionando uma estimativa de desempenho que é menos sensível às variações na distribuição dos dados de treinamento quando comparada ao método *holdout*. A validação cruzada, de forma ampla, emerge como uma ferramenta robusta na avaliação de modelos de aprendizado de máquina, servindo também para o ajuste de parâmetros com o intuito de otimizar o desempenho de generalização (Raschka, 2015).

Uma vez identificados os hiperparâmetros ótimos, o modelo é recalibrado utilizando o conjunto de treinamento completo, e seu desempenho final é avaliado utilizando um conjunto de teste independente (Raschka, 2015).

A abordagem *k-fold* da validação cruzada, por sua característica de não substituição durante a reamostragem, apresenta a vantagem de incluir cada observação no conjunto de treinamento e teste exatamente uma vez, resultando em estimativas de desempenho com menor variabilidade em relação ao método holdout (Raschka, 2015).

Embora o valor padrão de k na validação cruzada seja 10, uma escolha comumente aceita, é possível ajustar esse valor segundo as características do conjunto de dados. Valores maiores de k diminuem o viés na estimativa do desempenho de generalização, mas aumentam o custo computacional e podem gerar estimativas mais instáveis. Em contrapartida, valores menores de k podem ser mais eficientes em termos computacionais, preservando a precisão na estimativa do desempenho médio do modelo (Raschka, 2015). Na Figura 2 é mostrado um exemplo de quando $k = 10$ e para cada iteração é calculado um valor de E_i que representa a acurácia.

Figura 2 – Método *K-Fold Cross Validation*

Fonte: (Raschka, 2015).

2.5 Métrica de avaliação

No contexto de aprendizado de máquina supervisionado, o conjunto de dados é dividido entre treinamento e teste e como forma de medir o desempenho do modelo, para garantir sua capacidade de fazer previsões precisas e úteis, são utilizadas métricas de avaliação (Müller; Guido, 2017).

Nesta seção é abordada a métrica de acurácia na qual é aplicada ao problema em questão.

2.5.1 Acurácia

Para realizar o cálculo da acurácia inicialmente, é calculado o erro de previsão que consiste em uma medida de quantas previsões estão erradas em relação ao total de previsões, como é mostrado na Equação 16 (Raschka, 2015).

$$\text{ERR} = \frac{FP + FN}{FP + FN + TP + TN} \tag{16}$$

Onde:

- FP: Falso positivo;
- FN: Falso negativo;
- TP: Verdadeiro positivo;
- TN: Verdadeiros negativos.

Enquanto a acurácia é a proporção de previsões corretas em relação ao total de previsões. É calculada como a soma de previsões corretas (TP e TN) dividida pelo total de previsões como é apresentada na Equação 17.

$$\text{ACC} = \frac{TP + TN}{FP + FN + TP + TN} = 1 - \text{ERR} \tag{17}$$

2.6 Técnicas de Regularização

A regularização refere-se a um conjunto de técnicas usadas no aprendizado de máquina para evitar o *overfitting* de um modelo. O *overfitting* ocorre quando um modelo se torna muito complexo devido ao ajuste muito preciso aos dados de treinamento. Isso significa que o modelo está capturando não apenas os padrões reais nos dados, mas também o ruído e a variação aleatória presentes nos dados de treinamento (Raschka, 2015).

Existem diversas técnicas de regularização disponíveis para suavizar o *overfitting* em modelos de aprendizado de máquina, nesta seção serão apresentadas duas abordagens, a regularização L1 e regularização L2.

2.6.1 Regularização L1

A regularização L1 produz modelos esparsos e reduz o volume de ruído no modelo aplicando uma restrição aos coeficientes menos importantes, levando-os a zero. A esparsidade pode ser útil na prática se tivermos um conjunto de dados de alta dimensão com muitos recursos irrelevantes, especialmente casos em que se tem mais dimensões irrelevantes do que amostras. Nesse sentido, a regularização L1 pode ser entendida como uma técnica de seleção de atributos. De modo geral, a regularização L1 é a soma dos valores absolutos dos pesos, que é definida da seguinte forma (Raschka, 2015):

$$L1 = \sum_{j=1}^{m} |w_j| \tag{18}$$

2.6.2 Regularização L2

Essa técnica adiciona uma penalidade à função de custo proporcional ao quadrado dos pesos do modelo, o que incentiva o modelo a produzir valores de peso menores (Raschka, 2015). A regularização L2 é definida pela fórmula abaixo:

$$||L2||^2 = \sum_{j=1}^{m} |{w_j}^2| \tag{19}$$

em ambas Equações, 28 e 29, m representa a quantidade de pesos e w_j o valor absoluto de cada peso.

2.7 Otimização do modelo

No aprendizado de máquina, com o objetivo de melhorar o desempenho do modelo em relação aos dados de teste ou validação, são definidos hiperparâmetros. Os hiperparâmetros,

diferente dos pesos que são aprendidos a partir dos dados de treinamento, são configurações ajustáveis que não são aprendidas a partir dos dados, mas sim determinam o comportamento e a capacidade de generalização do modelo (Raschka, 2015).

Uma forma de definir os melhores hiperparâmetros do modelo é utilizando *Grid Search* que é apresentado nessa seção.

2.7.1 *Grid Search*

Segundo Raschka (2015), Grid Search é uma abordagem de busca exaustiva e de força bruta, onde uma lista de valores para diferentes hiperparâmetros é especificada. O computador então avalia o desempenho do modelo para cada combinação de valores para encontrar a combinação ótima que maximiza o desempenho do modelo. Em suma, o Grid Search é uma técnica que automatiza a busca pelos melhores hiperparâmetros para um modelo de aprendizado de máquina, testando diferentes combinações de valores para otimizar o desempenho do modelo. No contexto da regressão logística, alguns do hiperparâmetros a serem definidos, segundo Raschka (2015) são:

- C: O parâmetro de regularização controla a força da regularização no modelo. Valores mais altos de C indicam menos regularização, o que pode levar a um ajuste excessivo (*overfitting*), enquanto valores mais baixos de C indicam mais regularização, o que pode ajudar a evitar o *overfitting*;
- Solver: O solver é o algoritmo usado para otimizar os parâmetros do modelo. Diferentes *solvers* podem ser mais adequados para diferentes conjuntos de dados e tamanhos de problema, podem ser: "liblinear", "newton-cg", "lbfgs"e "sag";
- $Max\ iter$: O número máximo de iterações permitidas para o solver convergir. Se o solver não convergir dentro do número máximo de iterações, pode retornar uma mensagem de erro;
- Penalty: Controla o tipo de regularização aplicada ao modelo, impedindo que o modelo se ajuste aos dados de treinamento.

3 TRABALHOS RELACIONADOS

No estudo de Mortari (2022), o objetivo foi desenvolver um modelo para classificar a influência de atributos climáticos e de distância no Campeonato Brasileiro. A base de dados final incluiu vitórias, empates, derrotas, saldo de gols, média de gols, atributos históricos, temperatura, umidade, velocidade do vento, entre outros, totalizando 89 atributos. Esses dados foram processados e analisados utilizando a biblioteca de aprendizado de máquina Scikit-Learn e o algoritmo Regressão Logística.

Durante o pré-processamento, foram removidos dados de jogos do Joinville e Santos por falta de informações climáticas, além das 5 primeiras rodadas de cada ano para calcular atributos históricos e garantir que não houvesse valores nulos. A base final consistiu em 5681 amostras de 41 equipes. A seleção do modelo incluiu etapas de treinamento, validação e teste, com a escolha dos melhores hiperparâmetros via *GridSearch* e validação cruzada *k-fold* para séries temporais.

O modelo foi então treinado, testado e validado em dois cenários: um com todos os atributos (resultando em acurácia média de 50,40%) e outro sem os atributos de clima e distância (acurácia média de 50,45%). Esses resultados foram comparados com as odds de casas de apostas, que têm uma média de acerto de 51,41%.

Concluiu-se que o modelo teve um desempenho aceitável, mas os atributos de clima e distância não tiveram impacto significativo.

Na pesquisa conduzida por Fialho (2021), o objetivo era prever o resultado de partidas de futebol utilizando redes neurais artificiais e dados provenientes de 13 ligas diferentes. O número de partidas em cada liga variava de 259 a 3638, totalizando aproximadamente 28 mil partidas.

Os dados extraídos das partidas incluíam: gols, avaliação dos jogadores titulares na partida, chutes, chutes na direção do gol, passes corretos, disputa aérea de bola, *dribles* bem-sucedidos, número de tentativas de roubada de bola, posse de bola, nome do técnico, qualidade geral da equipe, formação, percentual de *dribles* bem-sucedidos, percentual de roubadas de bola, número de interceptações no passe do oponente, números de escanteios, roubadas de bola e a liga.

Para o estudo, Fialho empregou uma rede neural *feedforward* com múltiplas camadas escondidas. A técnica de regularização utilizada foi a *Dropout*, visando evitar o *overfitting*. A saída da rede consistia em três valores: vitória do time da casa, empate e vitória do visitante. No treinamento, foram utilizadas 20712 partidas, enquanto para validação e teste, a mesma proporção de dados de treino foi empregada. Ou seja, 10% dos dados de treino pertenciam a uma determinada liga, e o conjunto de treino e validação também correspondia a 10%. Nesse cenário, obteve-se uma taxa de acerto de 51,8%.

Posteriormente, foram identificados os dados mais relevantes para o modelo, e novos dados foram criados, como as chances de uma equipe marcar um gol, a perda da bola, en-

tre outros. A escolha dos melhores parâmetros da rede foi realizada utilizando *Grid Search*, resultando em uma média de acerto de 52,8%.

Outro estudo relevante foi conduzido por Buursma (2011), que tinha como objetivo prever os resultados em forma de vitória mandante, vitória visitante ou empate nos jogos do campeonato holandês. Para isso, foram coletados dados de 15 temporadas do referido campeonato.

A base inicial de dados continha informações sobre: gols marcados pelos times nas últimas x partidas, gols sofridos pelo time mandante e pelo time visitante nas últimas x partidas, e a média de pontos ganhos pelos times nas últimas x partidas.

Para determinar a quantidade de partidas a serem consideradas, Buursma utilizou a aplicação *Waikato Environment for Knowledge Analysis* (WEKA), que oferece diversos classificadores, e variou o número de x partidas entre 4 e 75 partidas. Descobriu-se que a partir de um histórico de 20 partidas, a acurácia não variava significativamente.

Com base nesse número, foram adicionadas mais características à base de dados, e as partidas em que os times haviam completado menos de 20 foram descartadas. Para a divisão da base de treino e teste, foi empregada a validação cruzada *k-fold*, e seis algoritmos foram executados. Os dois que obtiveram o melhor desempenho foram o de regressão linear, com uma acurácia de 54,86%, e a regressão logística, com uma acurácia de 54,84%.

Com base nos trabalhos citados acima, o presente trabalho tem como finalidade avaliar o impacto que dados geográficos tem sobre o resultado final de partidas da *Premier League*, utilizando mineração de dados para montar um *data warehouse* com todas as informações relevantes para o estudo e aplicando o algoritmo de regressão logística para verificar o desempenho dos cenários montados através da métrica acurácia.

4 METODOLOGIA

Neste capítulo é detalhado os materiais, métodos e procedimentos adotados para realizar a pesquisa. Será apresentado como os dados foram coletados, processados e analisados.

4.1 Materiais

Os materiais utilizados para a realização desse trabalho foram:

- **Python:** Linguagem de programação utilizada para o desenvolvimento.
- **Colab:** Ambiente de desenvolvimento colaborativo baseado na nuvem, frequentemente usado com Jupyter Notebooks.
- **Power BI:** Ferramenta de visualização de dados da Microsoft.
- **Notebook:** Hardware onde o código e análises foram escritos.

Bibliotecas Python:

- **datetime:** Para lidar com datas e horários;
- **pandas:** Biblioteca para manipulação e análise de dados;
- **wikipedia:** Para acessar e obter informações da Wikipedia;
- **csv:** Para trabalhar com arquivos CSV;
- **geopy:** Para geocodificação de endereços;
- **matplotlib:** Biblioteca para criação de visualizações e gráficos;
- **sklearn:** Biblioteca para execução do aprendizado de máquina

4.2 Coleta de Dados

Para esse estudo será utilizado os jogos da *Premier League*, Liga da primeira divisão da Inglaterra, da FA Cup, Copa da Inglaterra e para comparação dos resultados obtidos dados referentes a casas de aposta.

Referente a *Premier League*, a base inicial dados foi obtida a partir do *website* Kaggle (2024), um repositório público de base de dados. O processo de extração dos dados envolveu o *download* direto da base de dados no *website*. O conjunto de dados de Lawson (2018) abrange as temporadas de 1993 a 2018 e compreende uma variedade de variáveis para a análise, as quais incluem:

- Div: Divisão da competição;
- *Date*: Data em que o jogo foi realizado;
- *HomeTeam*: Time Mandante;
- *AwayTeam*: Time Visitante;
- *Full Time Home Goals* (FTHG): Total de gols do time mandante após o fim da partida;
- *Full Time Away Goals* (FTAG): Total de gols do time visitante após o fim da partida;
- *Full Time Result* (FTR): Qual foi o resultado no fim da partida, onde 'H' representa que o time da casa venceu, 'A' time da casa vencedor e 'D' o empate;
- *Half Time Home Goals* (HTHG): Total de gols do time mandante no intervalo da partida;
- *Half Time Away Goals* (HTAG): Total de gols do time visitante no intervalo da partida;
- *Half Time Result* (HTR): Qual time foi vencedor no intervalo da partida, onde 'H' representa que o time da casa venceu, 'A' time da casa vencedor e 'D' o empate;
- *Season*: Temporada que o jogo ocorreu.

Como pode ser visualizada na Tabela 1:

Tabela 1 – Base de dados referente a *Premier League*

Div	*Date*	*HomeTeam*	*AwayTeam*	FTHG	FTAG	FTR	HTHG	HTAG	HTR	*Season*
E0	19/08/95	Aston Villa	Man United	3	1	H	3	0	H	1995-96
E0	19/08/95	Blackburn	QPR	1	0	H	1	0	H	1995-96
E0	19/08/95	Chelsea	Everton	0	0	D	0	0	D	1995-96

Fonte: (Lawson, 2018).

Para a base de dados da FA Cup, a coleta de dados foi realizada utilizando *Power BI*, (uma ferramenta de avaliação e visualização de dados), a partir de (Soccer24, 2024a).

Assim, abrangendo as temporadas de 1998 a 2018, possuindo as seguintes variáveis:

- *Date*: Data em que o jogo foi realizado;
- *HomeTeam*: Time Mandante;
- *AwayTeam*: Time Visitante;
- FTHG: Total de gols do time mandante após o fim da partida;
- FTAG: Total de gols do time visitante após o fim da partida;
- *Season*: Temporada que o jogo ocorreu.

Foi adicionada a essa base de dados a seguinte variável, "*Season*" (Temporada), pelo fato de estar ausente na base de dados original. Essa nova coluna tem como propósito identificar a temporada específica para a qual a extração de dados foi realizada, conferindo contexto temporal aos dados carregados no Power BI, como apresentado na Tabela 2.

Tabela 2 – Base de dados referente a FA Cup

Date	*HomeTeam*	*AwayTeam*	FTHG	FTAG	*Season*
01.01.	Aston Villa	Hull	3	0	1998-99
01.01.	Blackburn	Charlton	2	0	1998-99
01.01.	Bolton	Wolves	1	2	1998-99

Fonte: (Soccer24, 2024b).

Desta forma, temos duas bases de dados que abrange a maioria dos jogos disputados pelos times da *Premier League*. As bases de dados coletadas, estão estruturadas de maneira diferente, as quais necessitam de uma estruturação e equiparação dos dados.

Os dados de casa de aposta foram extraídos do website Data (2024), no qual contém as colunas Date, HomeTeam, AwayTeam, Bet365 *home win odds* (B365H), Bet365 *draw odds* (B365D), Bet365 *away win odds* (B365A) e FTR, sendo elas:

- *Date*: Data em que o jogo foi realizado;
- *HomeTeam*: Time Mandante;
- *AwayTeam*: Time Visitante;
- B365D: Probabilidade de vitória do time mandante;
- B365D: Probabilidade de empate;
- B365A: Probabilidade de vitória do time visitante;
- FTR: Qual foi o resultado no fim da partida, onde 'H' representa que o time da casa venceu, 'A' time da casa vencedor e 'D' o empate.

Como pode ser visualizada na Tabela 3:

Tabela 3 – *Odds* de casa de aposta

Date	*HomeTeam*	*AwayTeam*	FTR	B365H	B365D	B365A
14/08/10	Wolves	Stoke	H	2.3	3.25	3.2
15/08/10	Liverpool	Arsenal	D	2.5	3.25	2.88
16/08/10	Man United	Newcastle	H	1.25	5.5	15.0
21/08/10	Arsenal	Blackpool	H	1.17	7.0	19.0
21/08/10	Birmingham	Blackburn	H	2.2	3.25	3.4

Fonte: (Data, 2024).

4.3 Pré-Processamento

4.3.1 Estruturação dos Dados

Na base de dados *Premier League*, as datas foram padronizadas para o formato ANO-MES-DIA e as colunas Div, HTHG, HTAG, HTR foram removidas. A razão da remoção da coluna Div, é que apresenta o elemento 'E0', ao qual se refere a primeira divisão. As colunas HTHG, HTAG, HTR, representam informações relacionadas ao intervalo da partida, as quais a base de dados da FA Cup não possui, assim buscando a padronização das bases de dados, foi realizada a remoção dessas colunas. Ficando da seguinte forma, como apresentado na Tabela 4:

Tabela 4 – Base de dados *Premier League* após estruturação dos dados

Date	*HomeTeam*	*AwayTeam*	FTHG	FTAG	FTR	*Season*
1993-08-14	Arsenal	Coventry	0	3	A	1993-94
1993-08-14	Aston Villa	QPR	4	1	H	1993-94
1993-08-14	Chelsea	Blackburn	1	2	A	1993-94

Fonte: Autoria própria.

Para a base de dados da FA Cup, foram realizadas algumas equiparações, de modo a ficaram com o mesmo padrão que a base de dados *Premier League*. Foi acrescentada a coluna FTR que representa: qual time foi vencedor no fim da partida, onde 'H' representa que o time da casa venceu, 'A' time visitante vencedor e 'D' o empate, com base nas colunas FTHG e FTAG, que representam o total de gols do time mandante/visitante após o fim da partida, dessa forma, definindo como:

- 'H' quando a coluna FTHG maior que FTAG;
- 'A' quando a coluna FTHG menor que FTAG;
- 'D' quando FTHG é igual a FTAG.

Além disso, as datas também foram padronizadas no formato ANO-MES-DIA. Ficando da seguinte forma, como apresentado na Tabela 5:

Tabela 5 – Base de dados FA Cup após estruturação dos dados

Date	*HomeTeam*	*AwayTeam*	FTHG	FTAG	FTR	Season
1998-11-06	Notts Co	Hendon	3	0	H	1998-99
1998-11-06	Kidderminster	Plymouth	0	1	A	1998-99
1998-11-09	Darlington	Manchester City	1	1	D	1998-99

Fonte: Autoria própria.

No intuito de aprimorar a base de dados, adicionar a base de dados da FA Cup à nossa análise, reconhecendo a participação significativa dos times da *Premier League* nesta competição. Contudo, para manter o foco no escopo específico do trabalho, que é centrado nos times

da *Premier League*, foi realizada a exclusão de times da FA Cup que não possuem participação simultânea na *Premier League*. Posteriormente, realizando a ordenação da base de dados em ordem cronológica.

4.3.2 Engenharia de Atributos

Para calcular a distância ou deslocamento das equipes para cada partida, foram utilizados dados geográficos obtidos através da latitude e longitude dos estádios. Essa informação foi adquirida por meio da *Application Programming Interface* (API) do Wikipedia, resultando na criação da coluna "*Distance*".

Outras métricas importantes foram geradas, como o aproveitamento dos times nas últimas 5 partidas, refletido nas colunas “*PerfomanceHome*” para o time mandante e “*Perfomance-Away*” para o time visitante.

A soma das últimas 5 distâncias percorridas pelos times também foi calculada, resultando em “*SumDistanceHome*” e “*SumDistanceAway*” para os times mandante e visitante, respectivamente. Por fim, a diferença de dias entre jogos foi calculada, proporcionando as colunas “*DaysBetweenGamesHome*” para o time mandante e “*DaysBetweenGamesAway*” para o time visitante. A Tabela 6, 7 e 8 apresentam a base de dados final:

Tabela 6 – Base de dados final - Parte 1

Date	*HomeTeam*	*AwayTeam*	FTR	Season
2018-04-16	West Ham	Stoke	D	2017-18
2018-04-17	Manchester United	Tottenham	H	2017-18
2018-04-17	Brighton	Tottenham	D	2017-18
2018-04-18	Chelsea	Southampton	H	2017-18
2018-04-18	Bournemouth	Manchester United	A	2017-18

Fonte: Autoria própria.

Tabela 7 – Base de dados final - Parte 2

DistanceHome	*DistanceAway*	*PerformanceHome*	*PerformanceAway*
0	217.46	0.33	0.06
0	256.25	0.8	0.6
0	85.83	0.13	0.46
0	111.74	0.66	0.2
0	308.05	0.33	0.8

Fonte: Autoria própria.

Para o cálculo da taxa de acerto das casas de aposta, foi realizado da seguinte forma: o menor valor entre as colunas B365D, B365D e B365A é considerado como o mais provável de acontecer em relação ao resultado final da partida, sendo 1 quando a casa de aposta acerta e 0 quando erra, valor que é atribuido na coluna *Result*, como mostra a Tabela 9:

Tabela 8 – Base de dados final - Parte 3

SumDistanceHome	*SumDistanceAway*	*DaysBetweenGamesHome*	*DaysBetweenGamesAway*
265.32	434.93	8	9
5.68	748.55	2	3
403.54	565.48	3	0
254.83	640.96	4	4
455.51	308.05	4	1

Fonte: Autoria própria.

- Exemplo 1: Para um jogo A, a coluna B365D teve menor valor, sendo assim, foi considerada com mais probabilidade de acontecer, caso o resultado final do jogo, na coluna FTR seja 'H' a coluna *Result* terá valor 1;
- Exemplo 2: Para um jogo B, a coluna B365A teve menor valor, sendo assim, foi considerada com mais probabilidade de acontecer, caso o resultado final do jogo, na coluna FTR seja 'A' a coluna *Result* terá valor 1;
- Exemplo 3: Para um jogo C, a coluna B365D teve menor valor, sendo assim, foi considerada com mais probabilidade de acontecer, caso o resultado final do jogo, na coluna FTR seja 'D' a coluna *Result* terá valor 1;
- Exemplo 4: Para um jogo D, caso não se encaixe nas três situações acima a coluna *Result* terá o valor 0;

Tabela 9 – Dados de casa de aposta com o acerto

Date	*HomeTeam*	*AwayTeam*	FTR	B365D	B365D	B365A	*Result*
14/08/10	Wolves	Stoke	H	2.3	3.25	3.2	1
15/08/10	Liverpool	Arsenal	D	2.5	3.25	2.88	0
16/08/10	Man United	Newcastle	H	1.25	5.5	15.0	1
21/08/10	Arsenal	Blackpool	H	1.17	7.0	19.0	1
21/08/10	Birmingham	Blackburn	H	2.2	3.25	3.4	1

Fonte: Autoria própria.

4.4 Parametrização do modelo

No experimento foi utilizado o algoritmo de regressão logística para obter seus resultados. Para otimizar o modelo, foi aplicado o método *GridSearch* com o objetivo de encontrar os melhores hiperparâmetros. A métrica escolhida para avaliar o desempenho do modelo foi a acurácia.

Os dados de treinamento abrangem um período equivalente às dez temporadas anteriores à temporada de avaliação. Em outras palavras, para avaliar a temporada 2003-04, o modelo foi treinado com dados das temporadas 1993-94 até 2002-03.

4.5 Avaliação do modelo

Para realizar a avaliação do modelo, serão realizados diferentes cenários, estes que foram comparados com a taxa de acerto das casas de aposta. Foram considerados três cenários base, com: nome dos times, aproveitamento e dias de descanso; em seguida, foram adicionados mais atributos em cada cenário. São detalhados os cenários base:

- Cenário 1: *HomeTeam*, *AwayTeam*;
- Cenário 2: *HomeTeam*, *AwayTeam*, *PerformanceHome* e *PerformanceAway*;
- Cenário 3: *HomeTeam*, *AwayTeam*, *PerformanceHome*, *PerformanceAway*, *DaysBetweenGamesHome* e *DaysBetweenGamesAway*;

Posteriormente, são adicionados aos cenários 1, 2, 3 o atributo de distâncias percorridas por cada time, resultando em mais três cenários.

- Cenário 4: *HomeTeam*, *AwayTeam*, *DistanceHome* e *DistanceAway*;
- Cenário 5: *HomeTeam*, *AwayTeam*, *PerformanceHome*, *PerformanceAway*, *DistanceHome* e *DistanceAway*;
- Cenário 6: *HomeTeam*, *AwayTeam*, *PerformanceHome*, *PerformanceAway*, *DaysBetweenGamesHome*, *DaysBetweenGamesAway*, *DistanceHome* e *DistanceAway*;

Em seguida, é adicionado o atributo de descanso, resultando em mais um cenário:

- Cenário 7: *HomeTeam*, *AwayTeam*, *DistanceHome*, *DistanceAway*, *DaysBetweenGamesHome* e *DaysBetweenGamesAway*;

Depois é adicionado o atributo de soma das últimas cinco distâncias percorridas por cada time, ocasionando em mais três cenários:

- Cenário 8: *HomeTeam*, *AwayTeam*, *DistanceHome*, *DistanceAway*, *DaysBetweenGamesHome*, *DaysBetweenGamesAway*, *SumDistanceHome* e *SumDistanceAway*;
- Cenário 9: *HomeTeam*, *AwayTeam*, *PerformanceHome*, *PerformanceAway*, *DistanceHome*, *DistanceAway*, *SumDistanceHome* e *SumDistanceAway*;
- Cenário 10: *HomeTeam*, *AwayTeam*, *PerformanceHome*, *PerformanceAway*, *DaysBetweenGamesHome*, *DaysBetweenGamesAway*, *DistanceHome*, *DistanceAway*, *SumDistanceHome* e *SumDistanceAway*;

Após a estruturação dos diferentes cenários que foram comparados com o modelo da casa de aposta.

Sendo assim, considera-se um cenário bom, o qual, tenha taxa de acerto maior que o modelo de casas de apostas.

5 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos para verificar a influência da distância percorrida pelos times em partidas de futebol. O objetivo deste estudo foi analisar como a extensão das viagens impacta o desempenho dos times, considerando diversos cenários apresentados na Seção 4.5.

5.1 Parametrização do modelo

Na realização do *Grid Search* com validação cruzada os melhores hiperparâmetros encontrados foram[1]:

- *Solver*='lbfgs';
- $max\ iter$=10000;
- CV=10;
- C= 0.01;
- *penalty*: 'l2'.

5.2 Avaliação do modelo

Com base nos cenários descritos na Seção 4.5, os hiperpârametros definidos na seção 5.1 e realizada a regressão logística para classificar, a Tabela 10 e Tabela 11, mostram os resultados obtidos, onde os valores em negrito representam os melhores resultados para cada temporada.

Para realizar a análise de como a distância afeta as partidas de futebol, foram feitas as seguintes comparações:

1. Cenários 1, 4, 7 e 8: Estes cenários consideram sem atributos de aproveitamento.
2. Cenários 2, 5 e 9: Estes cenários consideram atributos de aproveitamento.
3. Cenários 3, 6 e 10: Estes cenários consideram atributos de aproveitamento e dias de descanso.

[1] O *Grid Search* é uma técnica que melhora o desempenho de modelos de aprendizado de máquina ao testar diferentes combinações de hiperparâmetros. O parâmetro C controla a força da regularização, o *solver* otimiza os parâmetros do modelo, $max\ iter$ define o número máximo de iterações permitidas para o *solver* convergir, a penalidade controla a regularização, e CV determina o número de subconjuntos na validação cruzada *k-fold* (Raschka, 2015).

Tabela 10 – Resultados Obtidos (Odds e Cenários 1 a 5)

Temporadas	Casa de Apostas	Cenário 1	Cenário 2	Cenário 3	Cenário 4	Cenário 5
2003-04	49,47	47,02	60,62	**61,10**	48,69	60,14
2004-05	51,32	53,01	**66,90**	66,87	52,08	64,35
2005-06	57,63	55,17	68,51	68,05	55,63	68,05
2006-07	52,37	53,16	65,11	65,81	53,86	65,34
2007-08	58,42	52,62	65,95	66,43	52,62	**66,67**
2008-09	53,95	54,15	65,44	**66,13**	54,15	65,21
2009-10	56,05	53,72	66,51	66,98	53,02	66,28
2010-11	50,79	50,24	61,56	**62,74**	49,29	61,56
2011-12	52,11	51,98	67,13	66,20	52,45	66,20
2012-13	53,16	49,18	62,06	63,00	50,82	62,76
2013-14	60,00	53,70	67,54	**67,78**	52,51	67,54
2014-15	53,95	51,54	65,72	65,72	51,06	**65,96**
2015-16	47,63	45,31	63,05	**63,15**	45,54	62,44
2016-17	61,05	60,28	70,69	71,63	57,45	70,69
2017-18	55,26	55,48	64,80	64,49	55,24	**65,97**
Média	**53,95**	**53,01**	**65,72**	**66,13**	**52,51**	**65,96**

Fonte: Autoria própria.

Tabela 11 – Resultados Obtidos (Cenários 6 a 10)

Temporadas	Cenário 6	Cenário 7	Cenário 8	Cenário 9	Cenário 10
2003-04	60,14	48,93	49,16	61,34	**61,10**
2004-05	65,05	51,62	51,62	66,20	66,44
2005-06	67,59	55,86	55,17	69,43	**69,66**
2006-07	65,57	53,63	52,93	**66,28**	**66,28**
2007-08	66,19	52,38	52,38	65,71	65,95
2008-09	65,21	53,92	53,92	65,67	65,90
2009-10	**67,21**	53,49	53,72	**67,21**	**67,21**
2010-11	62,03	48,82	49,76	62,03	**62,50**
2011-12	66,20	52,45	52,45	67,37	**67,60**
2012-13	**63,47**	50,12	51,29	61,83	61,59
2013-14	**67,78**	52,51	52,03	**67,78**	67,30
2014-15	**65,96**	51,06	50,83	65,72	65,72
2015-16	62,91	45,31	46,01	61,74	61,74
2016-17	**71,87**	57,92	57,45	70,21	70,69
2017-18	65,50	54,78	54,31	63,40	64,34
Média	**65,57**	**52,45**	**52,38**	**65,72**	**65,95**

Fonte: Autoria própria.

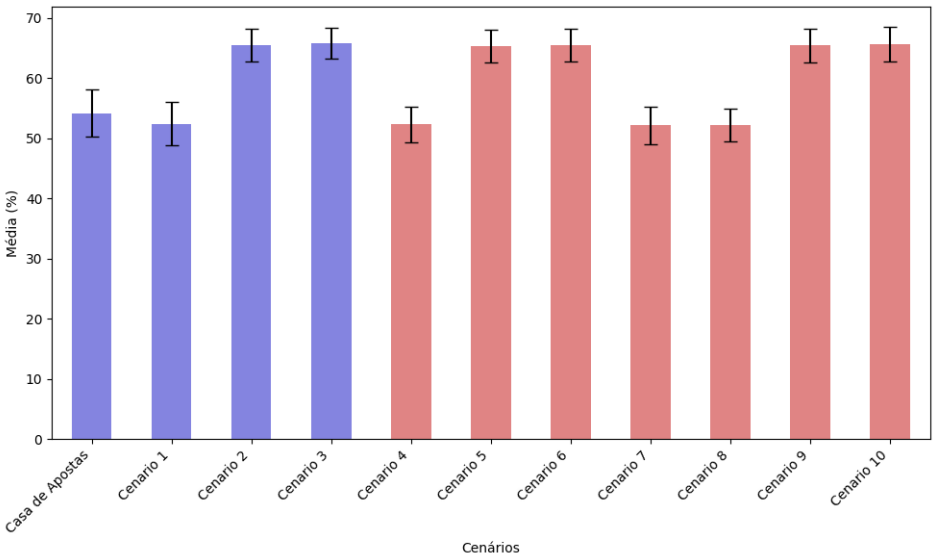
Análise 1: Analisando os cenários, com e sem atributos de descanso e distância, tem-se que comparando os cenários base 1, 2 e 3 com os cenários 4, 5 e 6 respectivamente, aos quais foi adicionado o atributo de distância: distância percorrida por cada time; obteve-se que na comparação do cenário 1 para 4, cenário 2 para 5 e cenário 3 para o 6 o valor obtido como acurácia foi semelhante, levando a concluir que o atributo de distância não influencia nos resultados das partidas.

Agora, levando em consideração a comparação dos cenários 7 para 8, 5 para o 9 e 6 para 10, o valor obtido nas comparações foi semelhante, levando a concluir que a variável distância não influencia nos resultados da partida.

Análise 2: Analisando os cenários em relação as casas de aposta tem-se que, os cenários que possuem apenas os atributos de descanso e distância (cenários 1,4,7,8) tiveram uma acurácia parecida com a casa de aposta, enquanto os cenários 2,3,5,6,9 e 10 obtiveram acurácias superiores as casas de aposta, pelo fato de levar em consideração o atributo de desempenho das equipes nos últimos jogos.

Na Figura 3 há uma representação dos dados em forma de gráfico de barras, que auxilia a compreensão das tendências ao longo do tempo, onde estão as médias de cada cenários e a média da casa de apostas, sendo que as cores azul e vermelho indicam, respectivamente, os casos sem o atributo de distância e com o atributo de distância.

Figura 3 – Comparação da média e desvio padrão entre cenários



Fonte: Autoria própria.

6 CONCLUSÃO

O presente trabalho tinha como objetivo verificar se a distância percorrida pelos times de futebol poderia ser considerada um fator determinante nos resultados das partidas, considerando as diferentes abordagens das casas de apostas para prever os vencedores.

Utilizando um conjunto de dados, que incluiu jogos da *Premier League* e da FA Cup, foi possível construir um *data warehouse* contendo informações, como a distância percorrida por cada time, histórico de distâncias anteriores e dias de descanso entre as partidas. Além disso, técnicas de aprendizado de máquina foram aplicadas, especificamente regressão logística com *GridSearchCV*, para analisar os dados e identificar padrões.

Os resultados revelaram que a variável de distância percorrida pelos times não teve um impacto significativo nos resultados das partidas de futebol. Após comparar os resultados obtidos com os dados das casas de apostas, observou-se uma acurácia semelhante, levando à conclusão de que outros fatores podem ser mais relevantes na determinação dos resultados das partidas.

Dessa forma, conclui-se que, embora a distância percorrida pelos times seja um aspecto relevante na logística das competições, não parece ser um fator determinante nos resultados das partidas da *Premier League*. Portanto, investir tempo e recursos na análise detalhada da distância percorrida pode não ser justificável em termos de previsão de resultados.

Em resumo, este estudo contribui para o campo de análise esportiva ao demonstrar que a distância percorrida pelos times não é um fator determinante nas partidas da *Premier League*. No entanto, ressalta-se que existem várias abordagens possíveis para prever resultados, e a distância pode ser apenas uma das muitas variáveis a serem consideradas.

Além disso, é importante considerar que em outras ligas ou campeonatos, dependendo do país e das condições de infraestrutura, como meios de locomoção e qualidade das estradas, a distância percorrida pode ter um impacto maior nos resultados das partidas. Esta possibilidade abre caminho para estudos futuros que poderiam investigar como esses fatores influenciam o desempenho dos times em diferentes contextos geográficos e logísticos.

REFERÊNCIAS

ALPAYDIN, E. **Introduction to machine learning**. Cambridge; London: MIT press, 2014.

BISHOP, C. M. **Pattern recognition and machine learning**. Cambridge, U.K.: [*s.n.*], 2006.

BUURSMA, D. Predicting sports events from past results. v. 21, 2011.

CARRILHO, J. R. J. Desenvolvimento de uma metodologia para mineração de textos. **Pontificia Universidad Catolica de Rio de Janeiro: Rio de janeiro, Brasil**, 2007.

DATA, F. **Data Files: England**. 2024. Disponível em: https://www.football-data.co.uk/englandm.php.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37–37, 1996.

FERRARI, D. G.; SILVA, L. N. D. C. **Introdução a mineração de dados**. São Paulo, SP, BRA: Saraiva Educaçao SA, 2016.

FIALHO, G. P. **Redes Neuronais para a Previsão do Resultado de Jogos de Futebol**. 2021. Tese (Doutorado) — Instituto Politecnico de Braganca (Portugal), 2021.

GOLDSCHMIDT, R.; PASSOS, E. **Data mining: um guia Prático**. New York, NY, USA: Elsevier Editora, 2005. ISBN 9788535218770. Disponível em: https://books.google.com.br/books?id=JJYHNrREwyEC.

HAN, J.; PEI, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. Elsevier Science, 2011. (The Morgan Kaufmann Series in Data Management Systems). ISBN 9780123814807. Disponível em: https://books.google.com.br/books?id=pQws07tdpjoC.

HOSMER, D. W.; LEMESHOW, S. **Applied logistic regression**. Danvers, MA, USA: John Wiley & Sons, 2000.

IBM. **Aprendizado Supervisionado**. 2024. Disponível em: https://www.ibm.com/br-pt/topics/supervised-learning.

KAGGLE. **Kaggle**. 2024. Disponível em: https://www.kaggle.com/.

LAWSON, S. **EPL Results 1993-2018**. 2018. Disponível em: https://www.kaggle.com/datasets/thefc17/epl-results-19932018.

MORTARI, G. H. d. A. Análise de dados relacionados a partidas de futebol utilizando técnicas de mineração. 2022. Disponível em: https://search.ebscohost.com/login.aspx?direct=true&db=ir00595a&AN=riut.1.29121&lang=pt-br&site=eds-live&scope=site.

MÜLLER, A. C.; GUIDO, S. **Introduction to Machine learning with Pyhton: A Guide for Data Scientists**. Sebastopol, CA, USA: O'Reilly Media, 2017.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. Cambridge; London: MIT press, 2012.

RASCHKA, S. **Python machine learning**. Birmingham,B3 2PB ,UK: Packt publishing ltd, 2015.

SILVA, D. N. **Futebol**. 2024. Disponível em: https://mundoeducacao.uol.com.br/educacao-fisica/futebol-2.htm.

SOCCER24. **Soccer 24: Fa Cup**. 2024. Disponível em: https://www.soccer24.com/england/fa-cup/.

SOCCER24. **Soccer 24: Fa Cup 1998/1999**. 2024. Disponível em: https://www.soccer24.com/england/fa-cup-1998-1999/results/.

TAN, P.-N. *et al.* **Introduction to Data Mining**. Arlington Street, Suite 300, Boston, MA: Pearson, 2006.