

ARTIGO ORIGINAL

Predição de resultados em partidas de futebol utilizando Deep Learning

Predicting results in soccer matches using Deep Learning

Edson Menin¹

¹Universidade de Passo Fundo (UPF)

*171947@upf.br;

Recebido: 2023-06-21. Revisado: 2023-06-26. Aceito: 2023-06-27.

Abstract

This paper aims to present a study on the prediction of results in Premier League football matches using Deep Learning techniques, specifically Multilayer Perceptron Neural Network (MLP), in order to contribute to sports data analysis and the prediction of football match outcomes, and to use them in online betting platforms. The prediction of results is carried out in two ways: in the first approach, the network predicts a home team win (0), a draw (1), or an away team win (2), achieving an accuracy of 53.65% in this context. In the second approach, only whether the home team wins (0) or not (1) is considered, resulting in an accuracy of 64.23%. Additionally, a simulation was conducted based on the pre-match odds of the games to analyze whether the bettor would profit or incur losses. In the first format, the simulation showed a loss of R\$ -266.41 for a bet of R\$ 548.00, while in the second format, a loss of R\$ -58.91 was incurred for the same amount. These results suggest that the use of the MLP algorithm can be effective only as an aid in decision-making for predicting football match outcomes, as considering the predicted data would result in losses for the bettor.

Keywords: Bets; Computer Science; Deep Learning; Neural Networks; Soccer.

Resumo

Este trabalho visa apresentar um estudo sobre a previsão de resultados de partidas de futebol da Premier League utilizando técnicas de Deep Learning, mais especificamente Rede Neural Perceptron Multicamadas (MLP), com o intuito de contribuir para a análise de dados esportivos e para a predição de resultados de partidas de futebol, e utilizá-las em casas de apostas online. A predição dos resultados é realizada de duas formas: na primeira, a rede prevê vitória do mandante (0), empate (1) ou vitória do visitante (2) onde, neste contexto, obteve-se uma acurácia de 53.65%; na segunda abordagem, é levado em consideração apenas se a equipe da casa ganha (0) ou se não ganha (1), sendo que neste caso obteve uma acurácia de 64.23%. Além disso, foi realizada uma simulação com base nas odds das partidas pré-jogo, com o intuito de analisar se o apostador obteria lucro ou prejuízo nas apostas, no qual, no primeiro formato, a simulação apresentou que o apostador teria um prejuízo de R\$ -266.41 perante a aposta de R\$ 548.00, e no segundo formato, um prejuízo de R\$ -58.91 perante o mesmo valor. Esses resultados sugerem que a utilização do algoritmo MLP pode ser eficaz apenas como auxílio na tomada de decisão para predição de resultados de partidas de futebol, já que levando em consideração os dados previstos, o apostador teria prejuízo.

Palavras-Chave: Apostas; Aprendizado Profundo; Ciência de dados; Futebol; Rede neural.

1 Introdução

Segundo Lira et al. (2018), a prática de jogos, dos mais variados tipos, é algo que existe desde a origem da humanidade. Está presente nas diversas sociedades que se constituíram ao longo dos séculos. O interesse constante por atividades de lazer e entretenimento sempre esteve ligada à humanidade e daí que surgiram os jogos, incluindo os de azar, que se difundiram ao longo das gerações sempre como atos de entretenimento ou atividades econômicas.

Com o advento da tecnologia, houve um aumento significativo no número de dados disponíveis sobre jogos de futebol, desde estatísticas de jogadores a de equipes. De acordo com Garey and Prado (2021), o mercado de apostas esportivas no Brasil alcançou R\$ 7 bilhões em 2020, mesmo com a pandemia que paralisou boa parte dos jogos. Entre 2018 e 2020, o setor cresceu de R\$ 2 bilhões para R\$ 7 bilhões. Ainda segundo Garey and Prado (2021), em 2020, o mercado global foi avaliado em US\$59.6 bilhões – podendo chegar até US\$127,3 bilhões em 2027.

Com isso, existe um interesse no estudo e na predição de resultados do esporte, como por ser visto nos trabalhos de Miljković et al. (2010), onde os autores utilizaram redes neurais para predição de resultados de partidas da Liga Nacional de Basquete (NBA); no trabalho de Tax and Joutstra (2015), com referências em outros estudos, utilizou técnicas de machine learning para predição de resultados de partidas do Campeonato Neerlandês de Futebol (Eredivisie); e no trabalho de Huang and Chang (2010), os quais realizaram estudos para predição de resultados na Copa do Mundo de 2006 utilizando rede neural perceptron multicamada (MLP).

Com isso, acredita-se que a utilização de técnicas de deep learning poderá trazer importantes contribuições para a análise de dados esportivos e para a predição de resultados de partidas de futebol. Neste artigo, adotamos a rede neural perceptron multicamada supervisionada (em inglês *Multilayer Perceptron* (MLP)) para prever o resultado de partidas de futebol na Premier League.

2 Revisão Bibliográfica

2.1 Premier League

Considerada a mais badalada competição nacional, a Premier League fez 30 anos em 2022 (Lois, 2022). De acordo com Lois (2022), a liga iniciada em Maio de 1992, foi criada após representantes de alguns clubes ingleses iniciaram uma separação da *English Football League*, entidade que organizava todo o futebol inglês desde 1888. Tal movimento ocorreu após tragédias relacionadas ao futebol que ocorreram no país, e visando também, o aumento das receitas provenientes dos direitos televisivos.

O campeonato segue um formato de pontos corridos, com um total de 38 rodadas e 10 jogos em cada rodada, totalizando 380 jogos por temporada. A temporada segue o calendário europeu, com duração de 10 meses, começando geralmente em agosto e terminando em maio do ano seguinte.

2.2 Redes neurais artificiais

Para de Leon F. de Carvalho (2009), Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

Haykin (2001), definiu redes neurais da seguinte forma em seu livro: Uma rede neural é um processador maciço, paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

- i. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
- ii. Forças de conexão entre neurônios conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

de Leon F. de Carvalho (2009), complementa que uma rede neural artificial é um sistema composto por várias unidades de processamento interconectadas, cujo funcionamento é baseado em princípios biológicos. Essas unidades, também chamadas de neurônios artificiais, são conectadas por canais de comunicação que possuem pesos associados. Cada unidade realiza operações locais em seus dados de entrada, recebidos por meio de suas conexões. A inteligência de uma rede neural artificial emerge das interações entre suas unidades de processamento.

A operação de um neurônio artificial (Fig. 1), proposta por McCullock e Pitts em 1943, pode ser resumida da seguinte maneira:

- i. Sinais são apresentados à entrada do neurônio.
- ii. Cada sinal é multiplicado por um peso que indica sua influência na saída do neurônio.
- iii. Os sinais ponderados são somados para produzir um nível de atividade.
- iv. Se o nível de atividade exceder um determinado limite (*threshold*), o neurônio produz uma resposta de saída.

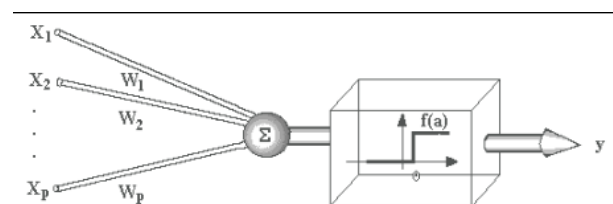


Figura 1: Esquema de unidade McCullock.

Suponha que tenhamos p sinais de entrada $X_1, X_2, \dots, X_p$ e pesos correspondentes $w_1, w_2, \dots, w_p$, além de um limite t . Se os sinais assumem valores booleanos (0 ou 1) e os pesos assumem valores reais, o nível de atividade a é calculado como:

$$a = w_1X_1 + w_2X_2 + \dots + w_pX_p$$

A saída y é determinada pela condição:

$$y = \begin{cases} 1 & \text{se } a \geq t \\ 0 & \text{se } a < t \end{cases}$$

Grande parte dos modelos de redes neurais utiliza alguma forma de regra de treinamento, na qual os pesos das conexões são ajustados com base nos padrões apresentados à rede. Em outras palavras, as redes neurais aprendem por meio de exemplos.

As arquiteturas neurais são tipicamente organizadas em camadas, onde as unidades de uma camada podem estar conectadas às unidades da camada seguinte. Essas camadas são classificadas em três grupos:

- i. **Camada de Entrada:** onde os padrões são apresentados à rede.
- ii. **Camadas Intermediárias ou Escondidas:** onde ocorre a maior parte do processamento, por meio de conexões ponderadas; essas camadas podem ser consideradas como extratoras de características.
- iii. **Camada de Saída:** onde o resultado final é produzido e apresentado.

A especificação de uma rede neural envolve principalmente sua topologia, as características dos nós e as regras de treinamento, como por exemplo o treinamento supervisionado do modelo de rede Perceptron.

2.3 Treinamento Supervisionado

Segundo de Leon F. de Carvalho (2009), o treinamento supervisionado do modelo de rede Perceptron, consiste em ajustar os pesos e os thresholds de suas unidades para que a classificação desejada seja obtida. Para a adaptação do threshold juntamente com os pesos podemos considerá-lo como sendo o peso associado a uma conexão, cuja entrada é sempre igual a -1 e adaptar o peso relativo a essa entrada. Quando um padrão é inicialmente apresentado à rede, ela produz uma saída. Após medir a distância entre a resposta atual e a desejada, são realizados os ajustes apropriados nos pesos das conexões de modo a reduzir esta distância.

2.4 Multilayer Perceptron

O Perceptron Multicamadas é uma rede neural variante do modelo original de Perceptron proposto por Rosenblatt em 1950. Ramchoun et al. (2016) definiu a MLP, como uma rede que possui uma ou mais camadas ocultas entre suas camadas de entrada e saída, os neurônios são organizados em camadas e as conexões são sempre direcionadas de camadas inferiores para camadas superiores. Ramchoun et al. (2016) explica que a rede MLP utiliza o algoritmo de Back-propagation para adaptar os pesos das conexões para obter uma diferença mínima entre a saída da rede e a saída desejada.

Haykin (2001), explica em seu livro, que durante o treinamento com o algoritmo Backpropagation, a rede opera em uma sequência de dois passos: a propagação e a retropropagação. No primeiro passo, um padrão é apresentado à camada de entrada da rede e a resposta é produzida pela camada de saída. Durante a propagação, os pesos sinápticos são fixos. No segundo passo, a saída obtida é comparada à saída desejada e, se houver erro, ele é propagado da camada de saída até a camada de entrada. Durante a retropropagação, os pesos sinápticos são ajustados de acordo com uma regra de correção de erro para fazer com que a resposta real da rede se mova para mais perto da resposta desejada. O objetivo é minimizar o erro em um sentido estatístico.

Optou-se por utilizar neste trabalho a rede MLP em razão do seu treinamento de forma supervisionada com o algoritmo de retropropagação de erro. Além disso, o estudo de Huang and Chang (2010) motivou a utilização desta rede neural, por apresentar uma taxa alta de acertos, chegando a uma acurácia de 76,6%, excluindo jogos de empate.

2.5 Abordagens utilizadas para predição de resultados em futebol

2.5.1 Jogos vencidos, perdidos e empatados

A análise do número de jogos vencidos, perdidos e empatados é uma informação relevante para a predição de resultados de partidas esportivas, como demonstrado em estudos anteriores. Por exemplo, Miljković et al. (2010) destacaram em seu artigo que as vitórias e derrotas das equipes de basquete são informações relevantes para a predição de resultados na Liga Nacional de Basquete (NBA). Da mesma forma, Aranda-Corral et al. (2013) classificaram o número de vitórias, derrotas e empates de equipes de futebol como uma variável de importância média a alta na predição dos resultados.

2.5.2 Gols marcados e sofridos

Miljković et al. (2010) também utilizaram o número de cestas marcadas e sofridas para a previsão de resultados de jogos de basquete. Isso ajuda a ressaltar que, assim como as cestas são usadas para prever resultados de jogos de basquete, os gols marcados e sofridos têm um papel semelhante na análise e avaliação. Huang and Chang (2010) consideraram a quantidade de gols marcados em sua pesquisa para predição de resultados na Copa do Mundo de 2006.

2.5.3 Promoção para a liga principal

Aranda-Corral et al. (2013) mesmo classificando como médio a promoção de equipes, e não utilizando em sua pesquisa, chegou à conclusão que, quando se joga numa divisão inferior ou superior, a dificuldade da divisão muda e os jogos não podem ser comparados da mesma forma. Por isso, é necessário lidar de forma diferente com uma equipe que jogava numa divisão diferente da atual.

2.5.4 Chutes a gol

Aranda-Corral et al. (2013) chegaram a conclusão que o número de chutes a gols associados ao número de gols da equipe, elevam a probabilidade de vitória da equipe. Huang and Chang (2010) utilizaram a quantidade de chutes a gol

em suas previsões. Além disso, foi gerado um gráfico de dispersão em que os pontos representam cada jogo e a posição de cada ponto no eixo x indica o número de chutes e no eixo y o número de gols marcados, demonstrando uma tendência de que quantos mais chutes a gol, mais gols são marcados. Na Fig. 2 são representados os dados da equipe da casa em cada jogo, e na Fig. 3 os dados da equipe visitante.

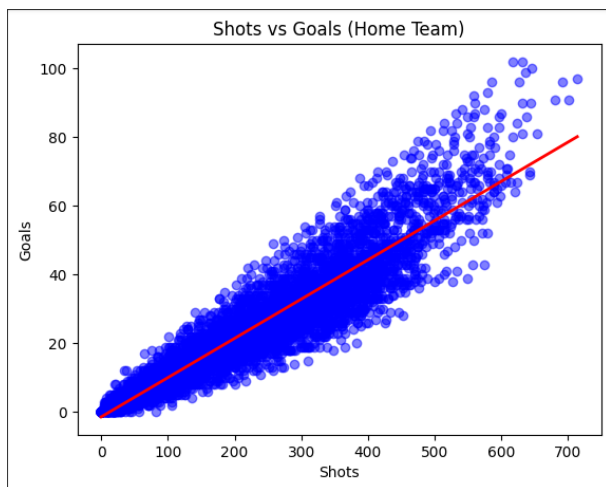


Figura 2: Chutes x gols por jogo equipe da casa.

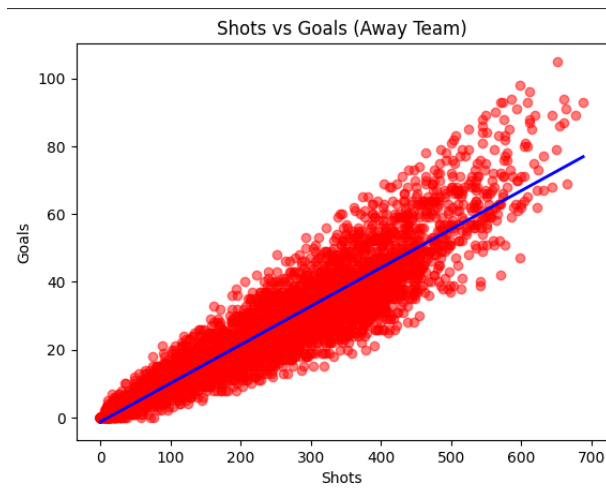


Figura 3: Chutes x gols por jogo equipe visitante.

2.6 Ferramentas utilizadas

2.6.1 Sklearn

Scikit-learn é uma biblioteca de aprendizado de máquina de código aberto que suporta aprendizado supervisionado e não supervisionado. Ele também fornece várias ferramentas para ajuste de modelo, pré-processamento de dados,

seleção de modelo, avaliação de modelo e muitos outros utilitários (Pedregosa and et al., 2011). Neste projeto foi essencial para pré-processamento dos dados, treinamento da rede e previsão dos dados.

2.6.2 Matplotlib

Criado por Hunter (2007), Matplotlib é uma biblioteca para fazer gráficos 2D de arrays em Python. Embora tenha sua origem na emulação dos comandos gráficos do MATLAB, é independente do MATLAB, podendo ser utilizado de forma Pythonica, orientada a objetos. Embora o Matplotlib seja escrito principalmente em Python puro, ele faz uso intenso do NumPy e de outros códigos de extensão para fornecer bom desempenho, mesmo para grandes matrizes. O Matplotlib auxiliou na geração de gráficos para análise dos dados no pré-processamento e no comparativo dos resultados.

2.6.3 Numpy

NumPy é um projeto de código aberto que permite computação numérica com Python. Foi criado em 2005 com base no trabalho inicial das bibliotecas Numeric e Numarray (Harris and et al., 2020). Foi utilizado especificamente na geração do gráfico de dispersão.

2.6.4 Pandas

Pandas é uma ferramenta de análise e manipulação de dados de código aberto rápida, poderosa, flexível e fácil de usar, construída sobre a linguagem de programação Python (Pandas Development Team, 2020). Esta ferramenta foi utilizada para leitura e manipulação do dataset.

3 Metodologia

3.1 Fonte de dados e descrição do dataset

A fonte de dados utilizada nesta pesquisa é o site [Football-data](https://www.football-data.co.uk/), que fornece dados históricos sobre partidas de futebol em várias ligas ao redor do mundo. O dataset utilizado consiste em informações sobre partidas de futebol realizadas nas últimas doze temporadas da liga principal da Inglaterra, a Premier League, tendo como data final 28/05/2023, no qual corresponde ao último jogo da temporada.

O dataset inclui informações como data e horário das partidas, equipes que jogaram em casa e fora, número de gols marcados e sofridos por cada equipe, resultado final (vitória para a equipe da casa, vitória para a equipe visitante ou empate), estatísticas de chutes a gol para cada equipe e valores das apostas pré-jogo, no qual nesta pesquisa, consideramos apenas do site [Bet365](https://www.bet365.com/).

As informações de cada coluna do dataset foram obtidas a partir do arquivo "notes.txt" disponível no seguinte link <https://www.football-data.co.uk/notes.txt>. As colunas utilizadas nesta pesquisa podem ser vistas na Tabela 1, com o conteúdo previsto em cada coluna.

3.2 Pré-processamento de dados

Inicialmente, realizamos uma validação para verificar se havia valores faltantes em algumas colunas consideradas

Tabela 1: Informações dataset.

Campo	Significado
Div	Divisão da liga em que a partida foi realizada
Date	Data em que a partida foi realizada (dd/mm/yy)
Time	Horário de início da partida
HomeTeam	Equipe que jogou em casa
AwayTeam	Equipe visitante
FTHG	Número de gols marcados pela equipe da casa na partida (90 minutos regulares)
FTAG	Número de gols marcados pela equipe visitante na partida (90 minutos regulares).
FTR	Resultado final da partida (H=vitória para a equipe da casa, A=vitória para a equipe visitante, D=empate)
HS	Número de chutes a gol da equipe da casa na partida (90 minutos regulares)
AS	Número de chutes a gol da equipe visitante na partida (90 minutos regulares)
B365H	Odds de vitória em casa da Bet365
B365D	Odds de empate da Bet365
B365A	Odds de vitória fora de casa da Bet365
B365AHH	Odds de handicap asiático da equipe da casa da Bet365
B365AHA	Odds de handicap asiático da equipe visitante da Bet365

importantes para a análise, como **Div**, **Date**, **HomeTeam**, **AwayTeam**, **FTHG**, **FTAG**, **FTR**, **HS** e **AS**. Felizmente, todas essas colunas estavam devidamente preenchidas, sem valores faltantes. Para as colunas **B365H**, **B365D**, **B365A**, **B365AHH** e **B365AHA** as colunas que faltavam informações foram preenchidas com o valor 1, no qual, de maneira satisfatória, todos valores que serão utilizados para simulação das apostas, estavam devidamente preenchidos.

Adicionamos uma coluna, chamada **Game_id**, para ser utilizada como referência para união de dados nas futuras análises. Uma padronização na coluna **Date** foi realizada, para após, ordenar o Dataset.

Em seguida, mapeamos os valores da coluna **FTR**, que indicam o resultado da partida (H = Home Win, D = Draw, A = Away Win), para valores numéricos (0 = Home Win, 1 = Draw, 2 = Away Win). Essa conversão numérica facilita a análise e o treinamento do nosso modelo.

Para aprimorar o treinamento do modelo, criamos duas colunas adicionais: **HomeTeam_id** e **AwayTeam_id**. Cada equipe no conjunto de dados recebeu um identificador único, permitindo uma melhor representação e compreensão da rede.

Também adicionamos uma nova coluna chamada **Season** para indicar a temporada correspondente a cada partida. Para determinar a temporada, consideramos a data da partida e o conhecimento geral de que as temporadas de futebol geralmente começam no segundo semestre do ano anterior e terminam no primeiro semestre do ano seguinte. Para isso, implementamos um loop que percorre as linhas do DataFrame, adicionando a temporada correspondente a cada partida.

Para identificar os clubes que foram promovidos para a Premier League em cada temporada de 2011 a 2022, criamos um dicionário. Em seguida, adicionamos duas colunas ao DataFrame, chamadas **HomePromoted** e **AwayPromoted**, ambas inicializadas com o valor 0. Depois, implementamos um loop que percorre cada partida no DataFrame. Para cada partida, verificamos se a equipe da casa ou a equipe visitante está presente no dicionário de clubes promovidos para a temporada correspondente. Se alguma das equipes estiver presente, a coluna correspondente (**HomePromoted** ou **AwayPromoted**) é atualizada com o valor 1, indicando que a respectiva equipe foi promovida

para a Premier League naquela temporada. Essa abordagem permite identificar automaticamente e com precisão quais equipes em cada partida foram recém-promovidas para a Premier League na temporada correspondente.

Por fim, adicionamos para cada equipe em cada temporada, o número de vitórias, derrotas e empates, o número de gols marcados e sofridos, o número de chutes a gol e o número total de partidas disputadas na temporada. Além disso, incluímos uma coluna chamada "Round" para categorizar a rodada em que cada partida foi realizada, o que será utilizado para representar graficamente os dados em nossa pesquisa. Esses dados serão utilizados para avaliar o desempenho das equipes ao longo da temporada.

Também foi realizado um segundo treinamento e teste, no qual foi feita uma alteração no dataset, utilizando a codificação 0 = Home Win e 1 = Draw ou Away Win. Neste teste, o objetivo era identificar apenas se o time da casa ganhou ou perdeu a partida, sem levar em consideração o empate.

4 Resultados e treinamento

4.1 Treinamento e validação do modelo

Para treinamento da rede, foram utilizados as seguintes informações sobre cada partida: os IDs dos times da casa e visitante, número de vitórias e derrotas, número de empates, gols marcados, gols sofridos, total de chutes, total de jogos de cada equipe e se a equipe foi promovida ou não para a Premier League. A lista Y, por outro lado, contém as saídas esperadas para cada partida, ou seja, o resultado final do jogo (FTR, na sigla em inglês).

O código e o dataset utilizado podem ser acessados através deste [repositório](#) no GitHub.

4.2 Treinamento e resultados no teste A

Para avaliar o desempenho do modelo de classificação de dados usando o algoritmo MLP (Perceptron Multicamadas), algumas técnicas foram empregadas. Primeiramente, normalizamos os dados usando a classe **MinMaxScaler** do **sklearn.preprocessing**, que transforma os dados

para que eles estejam em um intervalo específico (neste caso, entre 0 e 1). Em seguida, foram divididos os dados em conjuntos de treinamento e teste, sendo 88% do dataset original utilizado como dados de treinamento para ajustar o modelo, e os 12% restantes dos dados de teste para avaliar o desempenho do modelo.

Usando a classe `MLPClassifier` do `sklearn.neural_network`, foi criada uma instância do modelo de classificação de MLP e treinado o modelo usando os dados de treinamento normalizados. Depois disso, avaliamos o desempenho do modelo em dados de treinamento e teste, usando o método `.score()` para calcular a acurácia.

Para medir a eficácia do modelo, foram utilizadas métricas como F1 Score, que combina precisão e recall em uma única medida, e a função `accuracy_score`, que calcula a acurácia das previsões do modelo.

No final, foi obtido uma acurácia de 53.65% e F1 Score de 53.65% nos dados de teste, enquanto no conjunto de treinamento, o modelo obteve um desempenho com acurácia de 54.24%.

	Round	Game_id	HomeTeam	AwayTeam	Real	Predicted
4012	2021/22	4013	Leicester	Brighton	1	0
4013	2021/22	4014	Chelsea	Tottenham	0	0
4014	2021/22	4015	Burnley	Watford	1	0
4015	2021/22	4016	Newcastle	Everton	0	0
4016	2021/22	4017	West Ham	Watford	0	0
...	...	...	...	...	...	...
4555	2022/38	4556	Brentford	Man City	0	2
4556	2022/38	4557	Aston Villa	Brighton	0	2
4557	2022/38	4558	Arsenal	Wolves	0	0
4558	2022/38	4559	Chelsea	Newcastle	1	2
4559	2022/38	4560	Southampton	Liverpool	1	2

Figura 4: resultado real X previsto - teste A

Na Fig. 4, cada linha representa um jogo, com o time da casa, o time visitante, o resultado real (vitória do time da casa, vitória do time visitante ou empate) e a previsão da rede neural para aquele jogo. A tabela permite uma fácil comparação entre o resultado real e a previsão da rede neural.

Na Fig. 5, vemos as previsões da rede neural em relação aos resultados reais dos jogos, agrupados por 10 jogos cada barra. Para gerar o gráfico, primeiro foi criada uma coluna chamada "Resultado" que indica se a previsão da rede foi acertada ou não. Em seguida, os dados foram agrupados por rodada e pelo resultado (acerto ou erro) e contados. O resultado disso é uma tabela com duas colunas (acerto e erro) e várias linhas (uma para cada rodada). Por fim, foi gerado o gráfico de barras usando a biblioteca Matplotlib, com as barras empilhadas e coloridas de verde para acertos e vermelho para erros. O eixo x representa as rodadas e o eixo y representa o número de jogos.

4.3 Simulação de apostas para o teste A

Nesta etapa, foram calculados os ganhos e perdas com base nas odds de vitória fornecidas pelas apostas esportivas, com base nas colunas `B365H`, `B365D` e `B365A`. Para cada linha do dataframe `df_profit`, o cálculo foi realizado utilizando uma função lambda que verificou se a previsão (`Predicted`) e o resultado real (`Real`) coincidiam. Caso houvesse coincidência, o valor correspondente à aposta bem-sucedida era atribuído a coluna `Ganhos_Perdas`. Caso contrário, foi atribuído o valor -1, representando uma perda de R\$ 1.00 na aposta.

Além disso, foi adicionada uma coluna chamada `Valo_Apostado`, com o valor fixo de 1 para cada aposta. Em seguida, os ganhos e perdas foram multiplicados pelo valor apostado para obter o retorno.

Após esses cálculos, os resultados podem ser visualizados no dataframe `df_profit`.

Uma prévia dos resultados pode ser observada na Fig. 6. Ao final da simulação, o total de retorno foi igual a R\$ 281.59, sendo que o que total apostado foi R\$ 548.00, dando um prejuízo ao apostador de R\$ -266.41.

4.4 Treinamento e resultados no teste B

Para este teste, foram utilizadas as mesmas técnicas que foram utilizadas no Teste A, no qual a única alteração foi realizada no dataset, utilizando a codificação 0 = Home Win e 1 = Draw ou Away Win.

Neste caso, houve uma melhoria significativa nos resultados obtidos, no qual alcançou uma acurácia de 64.23% e F1 Score de 64.23% nos dados de teste, e no conjunto de treinamento, o modelo obteve o mesmo desempenho, com acurácia de 64.23%.

Na Fig. 7, cada linha representa um jogo, com o time da casa, o time visitante, o resultado real (vitória do time da casa, vitória do time visitante ou empate) e a previsão da rede neural para aquele jogo. A tabela permite uma fácil comparação entre o resultado real e a previsão da rede neural.

Na Fig. 8, vemos as previsões da rede neural em relação aos resultados reais dos jogos, agrupados por 10 jogos em cada barra. Para gerar o gráfico, primeiro foi criada uma coluna chamada "Resultado" que indica se a previsão da rede foi acertada ou não. Em seguida, os dados foram agrupados por rodada e pelo resultado (acerto ou erro) e contados. O resultado disso é uma tabela com duas colunas (acerto e erro) e várias linhas (uma para cada rodada). Por fim, foi gerado o gráfico de barras usando a biblioteca Matplotlib, com as barras empilhadas e coloridas de verde para acertos e vermelho para erros. O eixo x representa as rodadas e o eixo y representa o número de jogos.

4.5 Simulação de apostas para o teste B

Nesta etapa, foram calculados os ganhos e perdas com base nas odds de vitória fornecidas pelas apostas esportivas, com base nas colunas `B365AHH` e `B365AHA`. Para cada linha do dataframe `df_profit`, o cálculo foi realizado utilizando uma função lambda que verificou se a previsão (`Predicted`) e o resultado real (`Real`) coincidiam. Caso hou-

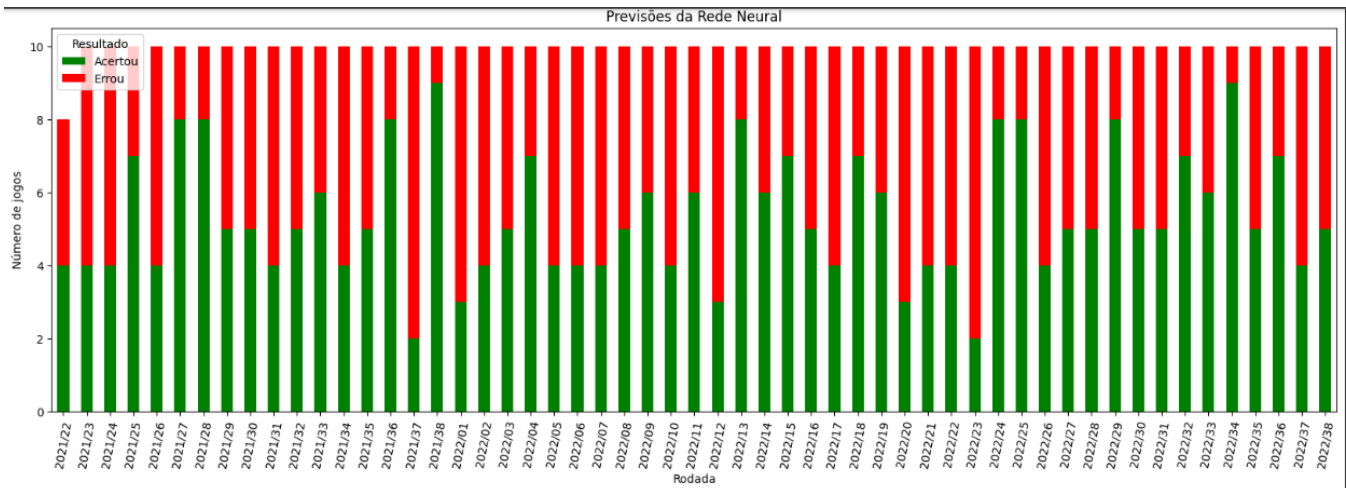


Figura 5: resultado real X previsto – teste A

	Round	Game_id	HomeTeam	AwayTeam	Real	Predicted	B365H	B365D	B365A	Ganhos_Perdas	Valor_Apostado	Retorno
0	2021/22	4013	Leicester	Brighton	1	0	2.50	3.25	2.90	-1.00	1	-1.00
1	2021/22	4014	Chelsea	Tottenham	0	0	1.61	4.20	5.00	1.61	1	1.61
2	2021/22	4015	Burnley	Watford	1	0	2.20	3.25	3.50	-1.00	1	-1.00
3	2021/22	4016	Newcastle	Everton	0	0	2.50	3.25	2.87	2.50	1	2.50
4	2021/22	4017	West Ham	Watford	0	0	1.55	4.20	5.75	1.55	1	1.55
...	...	...	...	...	...	...	...	...	...	...	...	...
543	2022/38	4556	Brentford	Man City	0	2	3.80	4.00	1.83	-1.00	1	-1.00
544	2022/38	4557	Aston Villa	Brighton	0	2	2.00	3.60	3.50	-1.00	1	-1.00
545	2022/38	4558	Arsenal	Wolves	0	0	1.36	5.25	8.00	1.36	1	1.36
546	2022/38	4559	Chelsea	Newcastle	1	2	2.90	3.50	2.38	-1.00	1	-1.00
547	2022/38	4560	Southampton	Liverpool	1	2	6.50	4.75	1.44	-1.00	1	-1.00

Figura 6: Previsão das Apostas – teste A

	Round	Game_id	HomeTeam	AwayTeam	Real	Predicted
4012	2021/22	4013	Leicester	Brighton	1	1
4013	2021/22	4014	Chelsea	Tottenham	0	0
4014	2021/22	4015	Burnley	Watford	1	1
4015	2021/22	4016	Newcastle	Everton	0	1
4016	2021/22	4017	West Ham	Watford	0	0
...	...	...	...	...	...	...
4555	2022/38	4556	Brentford	Man City	0	1
4556	2022/38	4557	Aston Villa	Brighton	0	1
4557	2022/38	4558	Arsenal	Wolves	0	0
4558	2022/38	4559	Chelsea	Newcastle	1	1
4559	2022/38	4560	Southampton	Liverpool	1	1

Figura 7: resultado real X previsto – teste B

vesse coincidência, o valor correspondente à aposta bem-sucedida era atribuído a coluna Ganhos_Perdas. Caso contrário, foi atribuído o valor -1, representando uma perda de R\$ 1.00 na aposta.

Além disso, foi adicionada uma coluna chamada **Valor_Apostado**, com o valor fixo de 1 para cada aposta. Em seguida, os ganhos e perdas foram multiplicados pelo valor apostado para obter o retorno.

Após esses cálculos, os resultados podem ser visualizados no dataframe `df_profit`.

Uma prévia dos resultados pode ser observada na Fig. 9. Ao final da simulação, o total de retorno foi igual a R\$ 489.09, sendo que o que total apostado foi R\$ 548.00, dando um prejuízo ao apostador de R\$ -58.91.

5 Conclusão

Na primeira abordagem, em que foram previstas vitórias do mandante, empates ou vitórias do visitante, a rede neural apresentou uma acurácia de 53.65%, demonstrando um desempenho abaixo do esperado na previsão dos resultados. Na simulação, o apostador teria um prejuízo de quase 50%. Já na segunda abordagem, onde foi previsto apenas se a equipe da casa ganha ou não, a acurácia foi melhor, alcançando 64.23%, indicando um desempenho muito bom, com um índice de acerto aceitável, porém, na simulação, o apostador teria ainda assim, um prejuízo.

Além disso, podemos observar que tanto na Fig. 5 como na Fig. 8, houve uma oscilação muito grande entre as previsões, no qual não manteve uma frequência de acertos.

Esses resultados sugerem que a utilização do algoritmo MLP em conjunto com técnicas de normalização de dados podem servir apenas como ferramenta de auxílio na tomada de decisão para predição de resultados de partidas de futebol, já que levando em consideração apenas os dados previstos, o apostador teria prejuízo.

As ferramentas utilizadas contribuíram significativamente para o sucesso e a qualidade deste estudo, permitindo uma análise abrangente dos dados e a obtenção de *insights* relevantes. Foi muito importante essa pesquisa

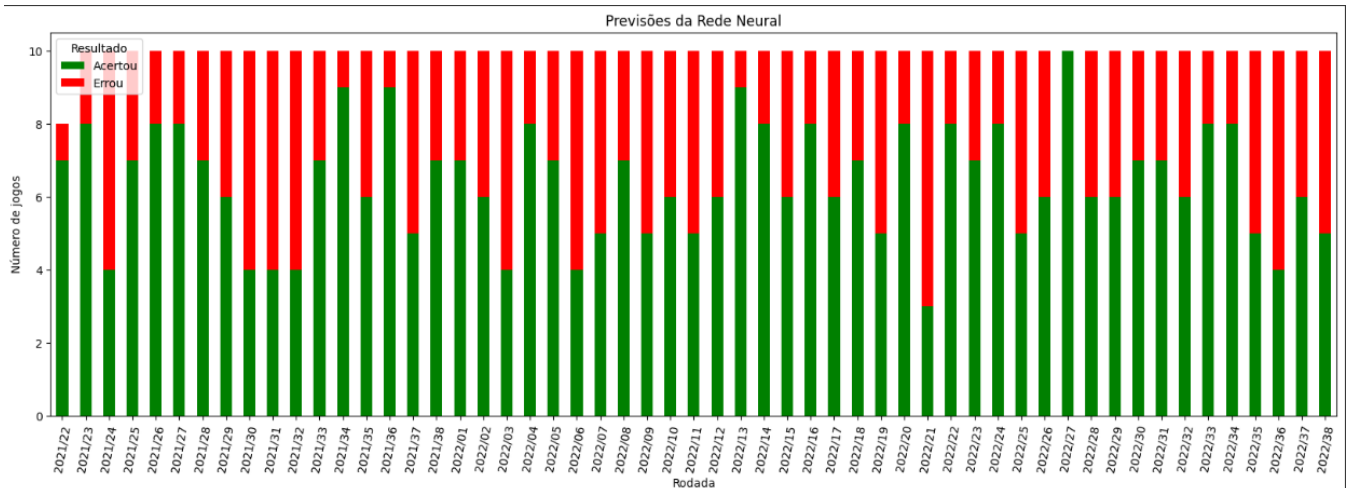


Figura 8: resultado real X previsto – teste B

	Round	Game_id	HomeTeam	AwayTeam	Real	Predicted	B365AMH	B365AMH	Ganhos_Perdas	Valor_Apostado	Retorno
0	2021/22	4013	Leicester	Brighton	1	1	1.91	2.02	2.02	1	2.02
1	2021/22	4014	Chelsea	Tottenham	0	0	1.93	2.00	1.93	1	1.93
2	2021/22	4015	Burnley	Watford	1	1	1.88	2.02	2.02	1	2.02
3	2021/22	4016	Newcastle	Everton	0	1	1.80	2.10	-1.00	1	-1.00
4	2021/22	4017	West Ham	Watford	0	0	1.93	1.97	1.93	1	1.93
...	...	...	...	...	...	...	...	...	...	...	...
543	2022/38	4556	Brentford	Man City	0	1	2.05	1.85	-1.00	1	-1.00
544	2022/38	4557	Aston Villa	Brighton	0	1	2.04	1.86	-1.00	1	-1.00
545	2022/38	4558	Arsenal	Wolves	0	0	2.01	1.89	2.01	1	2.01
546	2022/38	4559	Chelsea	Newcastle	1	1	1.83	2.07	2.07	1	2.07
547	2022/38	4560	Southampton	Liverpool	1	1	1.75	2.05	2.05	1	2.05

548 rows x 11 columns

Figura 9: Previsão das Apostas – teste B

para evolução pessoal, aumentando o conhecimento na análise de dados.

Em futuros estudos, podem-se levar em consideração maiores detalhes de uma partida, tais como diferença de dias entre uma partida e outra de uma equipe, porcentagem de lotação dos estádios, clima no local do estádio, a posição que a equipe se encontra no campeonato no momento do jogo ou um estudo direcionado para gestão de risco das apostas, ou seja, uma simulação de apostas, com os resultados entregues pelas rede, levando em consideração as odds das casas de apostas, para verificar a possibilidade de lucro ou não, ou gerando os resultados de previsão em porcentagem, para como mencionado anteriormente, auxiliar o apostador na tomada de decisão.

Referências

- Aranda-Corral, G. A., Borrego-Díaz, J. and Galán-Páez, J. (2013). Complex concept lattices for simulating human prediction in sport, *Journal of Systems Science and Complexity* 26(1): 117–136. <https://doi.org/10.1007/s11424-013-2288-x>.
- de Leon F. de Carvalho, A. P. (2009). Redes neurais artificiais. Disponível em <https://sites.icmc.usp.br/andre/research/neural>.
- Garey, R. and Prado, G. (2021). O mercado de apostas esportivas. Disponível em <https://gente.globo.com/o-mercado-de-apostas-esportivas>.
- Harris, C. R. and et al. (2020). Array programming with NumPy, *Nature* 585(7825): 357–362. <https://doi.org/10.1038/s41586-020-2649-2>.
- Haykin, S. (2001). *Redes neurais: princípios e prática*, Bookman Editora.
- Huang, K.-Y. and Chang, W.-L. (2010). A neural network method for prediction of 2006 world cup football game, *The 2010 international joint conference on neural networks (IJCNN)*, IEEE, pp. 1–8. <https://doi.org/10.1109/IJCNN.2010.5596458>.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment, *Computing in Science & Engineering* 9(3): 90–95. <https://dx.doi.org/10.1109/MCSE.2007.55>.
- Lira, P. E. M. d. et al. (2018). Os desafios para a regulamentação das apostas esportivas frente ao sistema jurídico brasileiro.
- Lois, R. N. (2022). Premier league chega a 30 anos como a liga mais forte do mundo. Disponível em <https://integrativos.ge.globo.com/futebol/futebol-internacional/futebol-ingles/materia/premier-league-chega-a-30-anos-como-a-liga-mais-forte-do-mundo>.
- Miljković, D., Gajić, L., Kovačević, A. and Konjović, Z. (2010). The use of data mining for basketball matches outcomes prediction, *IEEE 8th international symposium on intelligent systems and informatics*, IEEE, pp. 309–312. <https://doi.org/10.1109/SISY.2010.5647440>.
- Pandas Development Team, T. (2020). pandas-dev/pandas: Pandas. <https://doi.org/10.5281/zenodo.3509134>.
- Pedregosa, F. and et al. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12: 2825–2830. <https://dl.acm.org/doi/pdf/10.5555/1953048.2078195>.

- Ramchoun, H., Ghanou, Y., Ettaouil, M. and Idrissi, M. A. J. (2016). Multilayer perceptron: Architecture optimization and training. <https://dx.doi.org/10.9781/ijimai.2016.415>.
- Tax, N. and Joustra, Y. (2015). Predicting the dutch football competition using public data: A machine learning approach, *Transactions on knowledge and data engineering* 10(10): 1–13. <http://dx.doi.org/10.13140/RG.2.1.1383.4729>.