

REDES NEURONAIS PARA A PREVISÃO DO RESULTADO DE JOGOS DE FUTEBOL

GABRIEL PINTO FIALHO

Relatório Final da Dissertação apresentado à
Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Bragança

Para obtenção do grau de Mestre em
Engenharia Industrial
Ramo de Engenharia Eletrotécnica

Orientador do IPB:

Professor Doutor João Paulo Teixeira

Co-Orientadora do CEFET/RJ:

Professora Doutora Aline Gesualdi Manhães

Agradecimentos

Primeiramente gostaria de agradecer ao CEFET/RJ e ao Instituto Politécnico de Bragança pela oportunidade de fazer parte dessas renomadas instituições de ensino. Agradeço também à todos que fizeram parte direta ou indiretamente para a realização deste trabalho, em particular à minha família que me incentivou a seguir minhas escolhas.

RESUMO

Este trabalho tem como objetivo prever o resultado em forma de vitória/empate/derrota do confronto de duas equipas de futebol usando redes neuronais artificiais e dados estatísticos da performance das duas equipas e seus jogadores ao longo do tempo. É também um objetivo verificar se é realmente possível realizar essa previsão.

Foram coletadas estatísticas de cerca de 26 mil partidas de 13 ligas diferentes, entre elas o número de cantos, golos, avaliações dos jogadores. Depois, foi feita uma análise visual do conjunto de dados e realizado testes com diferentes formas de organizações históricas das estatísticas que comprovou que a média simples de 20 partidas anteriores apresenta a melhor performance, entre outras testadas.

Foi utilizada inicialmente uma rede neuronal “feedforward” com 100 neurônios em sua camada de entrada, 80 na camada seguinte, 5 e 3 nas camadas restantes. Além disso foi utilizada camadas de regularização do modo “dropout” entre todas as camadas de neurônios. Com esse modelo foi obtido 51.8% de exatidão no conjunto teste de 2500 partidas. Em seguida foi feita uma análise da relevância de cada dado estatístico e a criação de novos dados a partir dos dados iniciais. Além de realizar a otimização automática dos hiperparâmetros da rede neuronal utilizando a pesquisa em grade a fim de obter a melhor performance.

Após os procedimentos descritos, foi obtido, entre outros, 60,9% de exatidão nas partidas da Liga NOS de Portugal, 58,1% em partidas da Liga da Turquia e 57,2% nas partidas da Premier League da Inglaterra. Como resultado final, o modelo obteve 52,8% de exatidão no conjunto teste de 2500 partidas. Em alguns casos, por exemplo quando o modelo previu com uma probabilidade maior que 60% da equipa da casa vencer, a exatidão subiu para 80,3%. Esses resultados comprovam que é possível realizar a previsão de partidas de futebol utilizando redes neuronais artificiais.

Palavras-Chave: Inteligência Artificial, Redes Neuronais, previsão, resultados, futebol.

ABSTRACT

This study aims to predict soccer matches in the form of victory, draw or defeat and observe if soccer matches are predictable. Artificial Intelligence techniques were used in the form of Artificial Neuronal Networks in addition with statistics data from past matches such as players and team ratings, corners, goals. Data from more than 26,000 matches were collected from 13 different leagues throughout the world. Furthermore, graphics were plotted to visualize the distribution of the variables between the three different outcomes to conclude if the data, in fact, have correlation with the results.

In addition, the data was organized in historical form and tested in different ways such as weighted average of past 7 matches, average of past 20 matches and exponential weighted moving average with different alphas. Moreover, more statistics were generated with the initial dataset such as Pi-rating and Streak.

To test the different historical organization of the data, a feed-forward Neuronal Network with 100 neurons on the first layer, 80 on the second, 5 on the third and 3 on the last layer was used. Between the layers a Dropout layer was used to prevent the model of overfitting. The average of the 20 past matches had the best result with 51.8% of accuracy on the 2500 matches of the test dataset. Furthermore, more data was generated using the original ones to increase the accuracy. In addition, a grid-search was used to find the hyper-parameters which had the best performance.

Moreover, the model successfully predicted 60.9% matches of Portugal Liga NOS, 58.1% matches of Super Lig of Turkey, 57.2% matches of England Premier League. The artificial neuronal network successfully predicted the 2,500 matches of the test data with an accuracy of 52.8%. In addition, in some specific cases, for example when the network predicted that home team would win with more than 60% probability, the accuracy increased up to 80.3%. This conclude that, in fact, soccer matches can be predicted using neuronal networks.

Keywords: Artificial Intelligence, Machine Learning, Neuronal Network, predict, soccer outcomes.

ÍNDICE

1. Introdução.....	1
2. Ferramentas de Inteligência Artificial	3
2.1. <i>Aprendizado de Máquina</i>	<i>3</i>
2.2. <i>Sobreajuste e Subajuste</i>	<i>12</i>
2.3. <i>Hiper-parâmetros</i>	<i>13</i>
2.4. <i>Pré-processamento de Dados</i>	<i>14</i>
2.5. <i>Programas utilizados.....</i>	<i>17</i>
3. Estado da Arte	19
4. Recolha, Análise e Processamento da Base de Dados	23
4.1. <i>Base de dados</i>	<i>23</i>
4.2. <i>Processamento de Dados.....</i>	<i>31</i>
5. Modelo de Previsão de Resultados.....	37
5.1. <i>O Modelo</i>	<i>37</i>
5.2. <i>Resultados Iniciais</i>	<i>39</i>
5.3. <i>Melhorias.....</i>	<i>41</i>
6. Resultados e Discussão.....	51
7. Conclusões e Trabalhos Futuros.....	55
7.1. <i>Conclusões.....</i>	<i>55</i>
7.2. <i>Trabalhos Futuros.....</i>	<i>56</i>

ÍNDICE DE FIGURAS

Figura 2.1 – Diagrama de funcionamento de um algoritmo de aprendizado supervisionado	4
Figura 2.2 – Dados de grupos de flores e suas categorias	5
Figura 2.3 – Exemplo de categorização de um modelo supervisionado de classificação	6
Figura 2.4 – Gráfico que apresenta um conjunto de dados sobre carros.	7
Figura 2.5 – Exemplo de modelo de previsão por regressão.....	8
Figura 2.6 – Exemplo de um perceptron com 3 entradas (x_1 , x_2 , e x_3).....	9
Figura 2.7 – Rede Neuronal constituída por 3 perceptrons na primeira camada, 4 na segunda camada e um na saída	9
Figura 2.8 – Exemplo de árvore de decisão para classificar uma pessoa de acordo com sua altura	11
Figura 2.9 – Exemplos de subajuste, sobreajuste e um modelo correto.....	13
Figura 4.1 – Gráfico de números de partidas pelos resultados, respetivamente, equipa da casa venceu, equipa visitante venceu, ou empate.....	28
Figura 4.8 – Gráfico da distribuição de valores de golos.	33
Figura 4.9 – Gráfico da distribuição de valores de chutes.....	34
Figura 4.10 – Gráfico da distribuição de valores da posse de bola.	34
Figura 5.1 – Diagrama da Rede Neuronal proposta	39
Figura 5.2 – Importância dos dados no processo de previsão da vitória da equipa da casa.	42
Figura 5.3 – Importância dos dados no processo de previsão do empate.....	42
Figura 5.4 – Importância dos dados no processo de previsão da vitória da equipa visitante ...	42

ÍNDICE DE TABELAS

Tabela 2.1 – Hiper-parâmetros em uma rede neuronal	14
Tabela 4.1 – Dados estatísticos retirados das partidas e suas definições	25
Tabela 4.2 – Número de partidas coletadas por liga e temporadas.	26
Tabela 4.3 – Exemplo de organização dos dados estatísticos (incompletos) de 10 partidas....	27
Tabela 5.1 – Resultados dos diferentes tipos de organização temporal utilizando a uniformização como forma de processamento	40
Tabela 5.2 – Resultados dos diferentes tipos de organização temporal utilizando a normalização como forma de processamento	40
Tabela 5.3 – Resultado da análise da importância pela permutação das variáveis.	47
Tabela 6.1 – Exatidão do Modelo para cada Liga.	51
Tabela 6.2 – Concentração dos resultados no conjunto de teste.	52

1. Introdução

O desporto faz parte da vida de uma sociedade há milénios, mas não apenas como espectadores, o interesse em ser mais do que uma simples audiência nos remete a mais de 2000 anos atrás, quando gregos iam ao Coliseu apostar em gladiadores na época do Império Romano. Desde então, o interesse em tentar acertar resultados de jogos desportivos somente aumentou. Podemos ver comentadores em programas de televisão discutindo qual equipa irá ganhar o Super Bowl ou tentando descobrir o resultado de um jogo da Premier League, vemos pessoas em mídias sociais discutindo quem irá ganhar o Campeonato Mundial de Tênis, e vemos até mesmo comentadores fazendo palpites durante os jogos de quem será o vencedor. Ou seja, faz parte das nossas vidas.

O interesse por conhecer o resultado dos jogos desportivos antes de sua conclusão é agora claro e indispensável para a indústria de apostas. Simultaneamente, a tecnologia expandiu de uma forma exponencial e nos elevou a outro nível de compreensão, e a Inteligência Artificial está mudando a forma como vemos a previsão de resultados desportivos.

Os últimos anos foram muito importantes para as tecnologias de Aprendizado de Máquina (“Machine Learning”), tivemos uma expansão considerável de sua exatidão, e agora as redes neuronais artificiais já estão superando os seres humanos em muitas áreas (Steinberg, 2017). Por exemplo, como um experimento, Timnit Gebru (2017) capturou 50 milhões de imagens do Google Street View (recurso da Google que permite visão 360 graus ao nível do solo) e utilizou como entrada para uma rede de aprendizagem. O computador aprendeu a localizar e reconhecer veículos e detectou mais de 22 milhões de carros, suas marcas, modelos e tipos de carroceria. Com base na análise, ele concluiu que, se o número de sedans encontrados durante uma viagem de 15 minutos por uma cidade for maior do que o número de picapes, a cidade provavelmente votará em um democrata durante a próxima eleição presidencial (88% de exatidão).

Dentre as áreas em que computadores superam os humanos, a previsão é uma delas. O Aprendizado de Máquina é uma técnica dentro do ramo da inteligência artificial para prever resultados futuros com base em dados passados. O objetivo deste trabalho é utilizar o ramo de aprendizado de máquina, mais especificamente redes neuronais para prever resultados de jogos de futebol. Com isso, chegar à conclusão de que com a junção da tecnologia dos dias atuais e dados estatísticos passados é possível ou não prever resultados de jogos de futebol com uma exatidão satisfatória. Na próxima seção iremos explicar o que é o Aprendizado de Máquina

(*Machine Learning*) e suas técnicas mais eficazes para prever resultados futuros.

2. Ferramentas de Inteligência Artificial

2.1. Aprendizado de Máquina

O mundo está repleto de dados, fotos, músicas, textos, vídeos, armazenados nos celulares, computadores ou até mesmo em plataformas na nuvem. *Machine Learning* tem como promessa dar sentido a todas essas informações. Esta técnica é uma aplicação da inteligência artificial (IA) que fornece aos sistemas a capacidade de aprender e melhorar automaticamente a partir da experiência (dados), sem ser explicitamente programado. Aprendizado de Máquina se concentra no desenvolvimento de programas que acessam grande quantidade de informações e as utilizam para identificar padrões a fim de tomar decisões futuras. O objetivo principal é permitir que os computadores aprendam automaticamente sem intervenção ou assistência humana e ajustar as ações de acordo.

A técnica de Aprendizado de Máquina apresenta diferenças em relação à programação direta. A abordagem da programação convencional é alimentar o computador com um conjunto de instruções para cenários definidos. Assim, o computador utilizará seus recursos de computação para processar os dados de maneira mais rápida e eficiente. Enquanto no *Machine Learning*, uma enorme quantidade de dados é introduzida no computador, que por sua vez processa e cria a solução chamada de modelo treinado. Essa técnica apresenta vantagens em relação à programação direta, pois tem a capacidade de revisar grandes volumes de dados e descobrir tendências e padrões específicos que não seriam aparentes para os seres humanos. Também não é necessária intervenção humana, já que essa técnica proporciona às máquinas a capacidade de aprender, fazer previsões e aprimorarem os algoritmos por conta própria. À medida que eles ganham experiência, os algoritmos continuam melhorando em sua exatidão e eficiência, isso permite que o modelo tome melhores decisões ao longo do tempo.

A técnica de *Machine Learning* pode ser dividida em categorias de acordo com sua finalidade, discutiremos as principais a seguir. A primeira categoria é a de algoritmos supervisionados, que podem aplicar o que foi aprendido no passado a novos dados usando exemplos para prever eventos futuros. A partir da análise de um conjunto de dados conhecidos de treinamento, o algoritmo de aprendizado produz uma função para fazer previsões sobre os valores de saída de acordo com os dados de entrada. O sistema é capaz de fornecer saídas para qualquer nova entrada após treinamento suficiente. O algoritmo também pode comparar sua saída com a saída correta, encontrar erros e modificar o modelo de acordo. Em contraste, os algoritmos não supervisionados trabalham apenas com dados de entrada e nenhuma variável de

saída correspondente. Esses algoritmos têm como objetivo achar padrões e regras nos dados, trazendo uma visão melhor para os usuários. Já os algoritmos semi-supervisionados estão entre os dois citados acima. São utilizados em problemas onde há uma grande quantidade de dados de entrada e somente uma parte tem valores de saída, pois em algumas situações pode se tornar caro e consumir muito tempo.

O objetivo deste trabalho se encaixa perfeitamente na categoria de algoritmo supervisionado, sendo assim entraremos mais em detalhes nesse tema. Para essa técnica é necessário primeiro treinar um modelo de Aprendizado de Máquina utilizando atributos (variáveis independentes) de um determinado dado e seus resultados (variáveis dependentes), geralmente representados por vetores. O modelo então “aprende” a relação que as variáveis independentes têm com as variáveis dependentes. Finalmente, faz-se previsões com dados novos onde o resultado é desconhecido. A figura 2.1 ilustra os passos descritos.

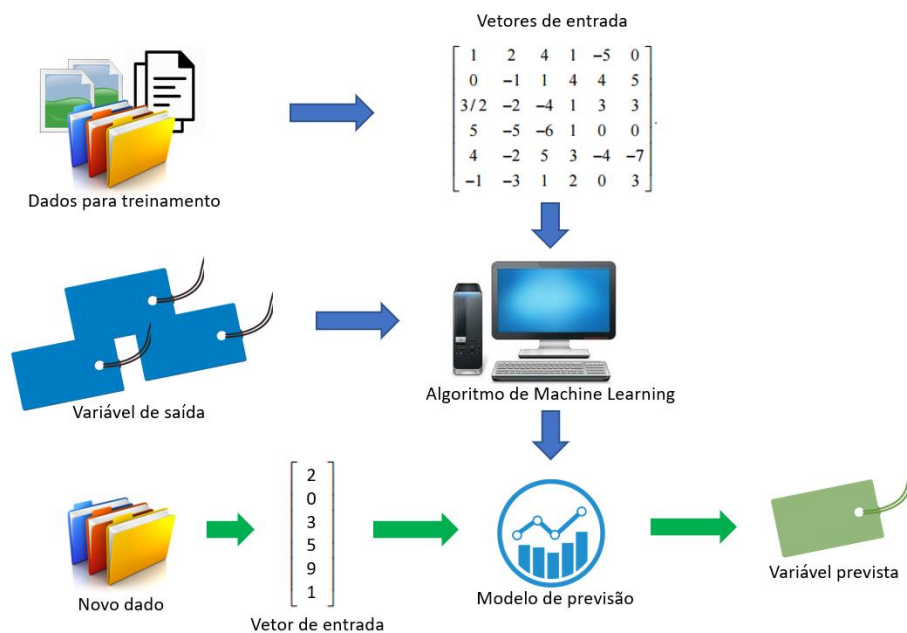


Figura 2.1 – Diagrama de funcionamento de um algoritmo de aprendizado supervisionado

Os algoritmos de aprendizado supervisionados são divididos em duas categorias: classificação e regressão.

2.1.1. Algoritmo Supervisionado de Classificação

Algoritmos supervisionados de classificação são utilizados em problemas onde se deseja categorizar dados em determinadas e distintas classes. Alguns exemplos de utilização dessa técnica são: reconhecimento de voz, identificação biométrica, classificação de documentos, classificação de e-mails não desejados. Nesse modelo, o conjunto de dados é uma coleção de exemplos categorizados $(x_i, y_i)^{(n)}$ onde cada elemento “x” é chamado de vetor de características, “y” é a classe de “x” e “n” um exemplo específico. Um vetor de características é um vetor onde cada dimensão $j = 1, \dots, P$ contém um valor que descreve um exemplo. Por exemplo se cada dado x em um determinado modelo representa uma pessoa, então a primeira característica x_1 pode conter a altura em centímetros, a segunda característica x_2 , pode conter o peso em quilos, x_3 o gênero da pessoa e assim por diante. Além disso, os dados devem ser uniformes. Assim, se x_2 contém o peso em quilos em algum exemplo $n=1$, então $x_2^{(2)}$ também deve conter o peso em quilos para qualquer exemplo $x_2^{(k)}$, $k = 1, \dots, N$. Nesse modelo, y_i pode representar os nomes das pessoas, sendo um conjunto finito de classes.

O conjunto de entrada de um algoritmo supervisionado de classificação pode ser numérica ou categórica. Podemos ver na figura 2.2 um exemplo de conjunto de dados utilizados de certos grupos de flores. Os eixos x e y representam especificações numéricas das flores e as cores verde, azul e vermelho representam as 3 diferentes categorias relacionadas aos parâmetros e cada ponto do gráfico representa o conjunto de dados de uma flor.

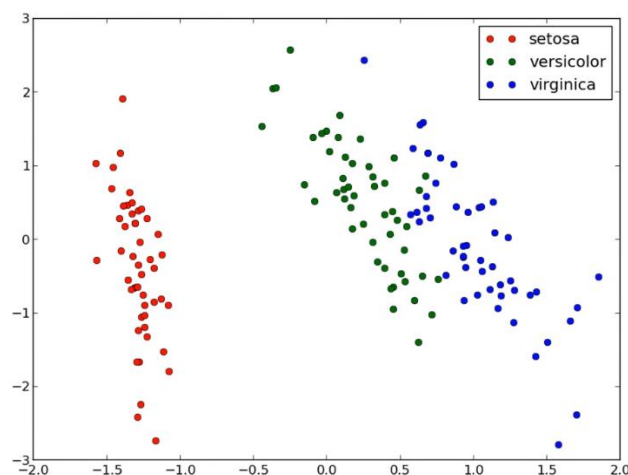


Figura 2.2 – Dados de grupos de flores e suas categorias

Fonte: Anderson (1936); Fisher (1936).

O conceito de um modelo de classificação é utilizar dados de entrada para treinar e

aprender fronteiras que separam as categorias dos dados para posteriormente prever futuras entradas. Nesse exemplo das flores, uma máquina de aprendizado supervisionado de classificação poderia separar as categorias de acordo com a Figura 2.3, assim se um determinado dado de entrada estiver dentro da área B, provavelmente será classificada como versicolor, e assim adiante.

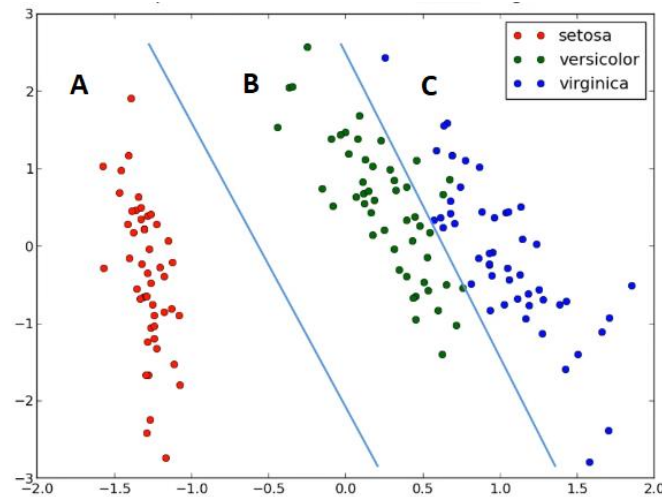


Figura 2.3 – Exemplo de categorização de um modelo supervisionado de classificação

A exatidão do modelo é calculada como a porcentagem de exemplos classificados corretamente dividido por todas as previsões feitas. Por exemplo, se um modelo de previsão de classificação realizou 8 previsões e 6 foram corretas, a exatidão será de 75% (6 divididos por 8).

Apesar de ser utilizada para classificar dados em valores discretos, a saída do modelo pode ser representada em valores contínuos onde pode-se extrair a probabilidade desse dado pertencer a cada classe. A seguir, essa probabilidade é convertida em uma classe, selecionando a categoria onde se tem o maior valor. Por exemplo, um e-mail pode ser classificado por 0,1 como “spam” e 0,9 como “não spam”. Assim, futuros e-mails serão classificados conforme a maior proximidade dos dois valores. Se a saída do modelo for 0.2, então provavelmente o e-mail será “spam”.

2.1.2. Algoritmo Supervisionado de Regressão

Algoritmos supervisionados de regressão são utilizados em problemas onde se deseja prever um valor contínuo, geralmente quantidades ou tamanhos. A saída desse modelo é

representada por um número real contínuo, como inteiro ou fracionado. Alguns exemplos da utilização dessa técnica são: prever o valor de uma casa pelas suas especificações, prever quantos metros um atleta de salto em distância conseguirá atingir, prever salários de empregados durante os anos.

Podemos ver na Figura 2.4 um exemplo de conjunto de dados representado por um gráfico onde o eixo horizontal é o peso de um determinado carro e o eixo vertical quantas milhas esse veículo faz por galão e cada ponto da figura representa um carro.

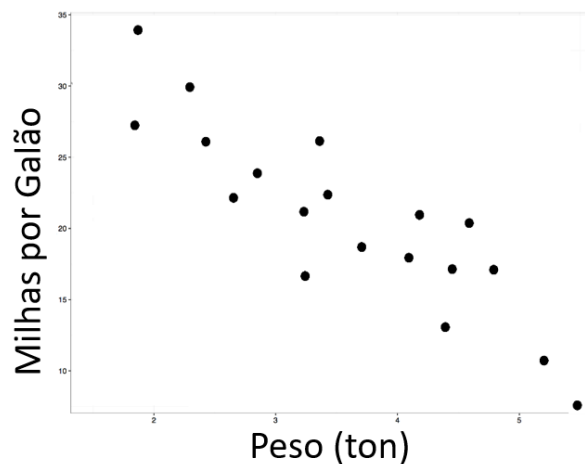


Figura 2.4 – Gráfico que apresenta um conjunto de dados sobre carros.

Fonte: Pacote R, “mtcars”.

Deseja-se prever quantas milhas por galão um determinado carro faz apenas sabendo seu peso. Assim, utilizamos algoritmos supervisionados de regressão, pois a variável dependente (milhas por galão) é contínua. Um exemplo de modelo de previsão para esse caso pode ser visto na Figura 2.5. A linha vermelha representa o padrão gerado para futuras previsões, os pontos pretos são o conjunto de dados reais, os pontos azuis representam a saída do sistema de previsão (dados previstos) e as linhas cinzas são as distâncias que os dados previstos têm sobre os dados corretos.

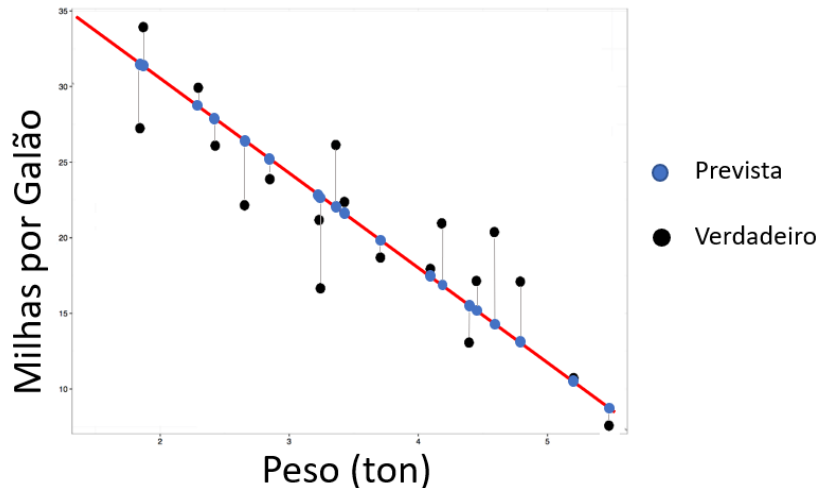


Figura 2.5 – Exemplo de modelo de previsão por regressão

Diferente do modelo de classificação, a exatidão é calculada levando em consideração o quão distante o resultado previsto está do real. Podem ser utilizadas diversas equações matemáticas para calcular essa distância, a mais comum é a Raiz do Erro Quadrático Médio (REQM) ou *Root Mean Square Error (RMSE)*. Essa técnica representa o desvio dos resíduos (erros de previsão). Resíduos são uma medida da distância dos pontos (dados) da linha de regressão, RMSE representa como se “espalham” esses resíduos. Em outras palavras, ele informa como os dados estão concentrados em torno da linha.

2.1.3. Redes Neurais Artificiais

Redes Neurais Artificiais se encontram dentro da categoria de Aprendizado de Máquina e são sistemas computacionais inspirados nas redes neuronais que constituem os cérebros dos seres vivos. É composto por uma rede artificial que, via algoritmo, permite que o computador aprenda incorporando dados.

Enquanto a unidade básica do cérebro é o neurônio, o essencial de uma rede neuronal artificial é um perceptron, realizando o processamento simples de um sinal, estes são então conectados para formar uma rede neuronal artificial. Perceptrons foram desenvolvidos nos anos 1950 e 1960 pelo cientista Frank Rosenblatt (1958), inspirado no trabalho de Warren McCulloch e Walter Pitts (1943). Um perceptron tem uma quantidade x de entradas binárias e produz uma única saída binária, um exemplo de perceptron com três entradas está representado na figura 2.6.

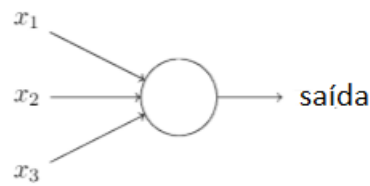


Figura 2.6 – Exemplo de um perceptron com 3 entradas (x_1 , x_2 , e x_3)

Fonte: Nielsen (2019), p. 3.

Para calcular a saída, Rosenblatt introduziu pesos (w_1, w_2, w_3, \dots) que são representados por números reais e expressam a importância que a determinada entrada tem para a saída. A saída do neurônio é então determinada pela soma ponderada, 0 caso essa soma for menor que um valor limite e 1 caso seja maior, como mostra a equação 2.1, onde j representa o número de entradas.

$$Saída = \begin{cases} 0 & \text{se } \sum_j w_j x_j \leq \text{limite} \\ 1 & \text{se } \sum_j w_j x_j > \text{limite} \end{cases} \quad (2.1)$$

Uma rede neuronal é constituída por diversos perceptrons, organizados em camadas, como na figura 2.7.

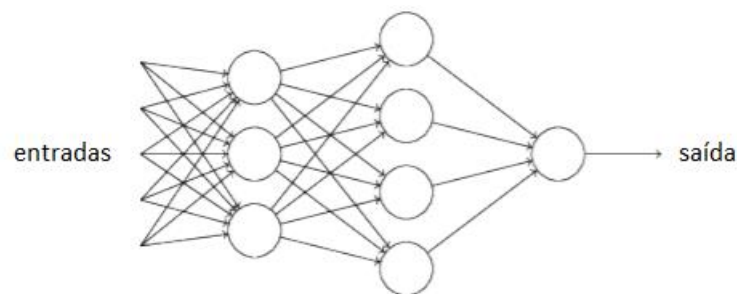


Figura 2.7 – Rede Neuronal constituída por 3 perceptrons na primeira camada, 4 na segunda camada e um na saída

Fonte: Nielsen (2019), p. 4.

Nessa rede, a primeira camada de perceptrons (primeira coluna) realiza as decisões mais simples. Os perceptrons da segunda camada chamada de camada escondida, pois ela não é entrada ou saída, realizam uma decisão pesando os resultados da primeira camada. Assim, eles

conseguem realizar decisões mais complexas e abstratas do que perceptrons da primeira camada. E decisões ainda mais complexas podem ser tomadas pelos perceptrons na terceira camada. Desta forma, uma rede de perceptrons de várias camadas pode realizar complexas e trabalhosas decisões.

Pequenas mudanças nos pesos devem causar pequenas mudanças na saída da rede neuronal, isso é o ideal para um modelo de aprendizado. Com isso, pode-se realizar pequenas modificações nos pesos para fazer com que a rede neuronal trabalhe da forma desejada. Porém isso não acontece em uma rede neuronal composta por perceptrons. Utilizando de exemplo uma rede neuronal para reconhecer imagens de animais e categorizá-los. A sua saída representa uma imagem como sendo um gato quando na verdade é um cachorro. Como dito anteriormente, pode-se realizar pequenas alterações nos pesos da rede para que ela consiga chegar mais perto de categorizar a imagem corretamente como “cachorro”, assim a rede estaria aprendendo. O problema é que isto não ocorre quando a rede contém perceptrons. Uma pequena mudança nos pesos de um único perceptron pode causar uma mudança drástica em sua saída, por exemplo de 0 para 1. Essa alteração pode fazer com que a rede neuronal tenha um comportamento totalmente diferente. A rede agora consegue acertar imagens de cachorros, porém o comportamento da rede com imagens de outros animais pode ter mudado drasticamente e de uma forma imprevisível. Isso dificulta o processo de modificar gradualmente os pesos para que a rede se aproxime do comportamento desejado. Esse problema foi controlado introduzindo um novo tipo de neurônio artificial chamado neurônio sigmóide. Os neurônios sigmóides são semelhantes aos perceptrons, mas modificados para que pequenas mudanças em seus pesos causem apenas uma pequena alteração em sua saída.

O neurônio sigmóide apresenta entradas como o perceptron, x_1, x_2, x_3, \dots Mas sua saída ao invés de ser apenas 0 ou 1 também pode assumir qualquer valor entre 0 e 1. E juntamente com perceptrons, apresenta pesos para cada entrada (w_1, w_2, w_3, \dots). Sua saída pode ser representada pela equação 2.2. Onde σ é a função sigmóide (chamada de função de ativação) representada pela equação 2.3.

$$\frac{1}{1 + \exp(-\sum_j w_j x_j - b)} \quad (2.2)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.3)$$

Hoje, temos diversas funções de ativação para situações específicas. Com isso, redes neurais se tornaram diversificadas e abrangendo um número maior de problemas.

Os exemplos mostrados anteriormente destacavam redes neurais onde a saída de uma camada é utilizada como entrada para a próxima camada. Esse modelo é chamado de *feedforward*, isso significa que a informação segue o caminho sempre da entrada para a saída, no caso da figura 2.7 da esquerda para direita.

2.1.4. Árvores de Decisão

Uma árvore de decisão é uma estrutura semelhante a um fluxograma na qual cada nó interno representa um “teste” em um atributo (por exemplo, se uma moeda aparece cara ou coroa), cada “galho” representa o resultado do teste e cada nó (“folha”) representa uma classe (decisão tomada após computar todos os atributos). Os caminhos da raiz para a folha representam regras de classificação.

Os algoritmos de aprendizado baseados em árvores de decisão são considerados um dos melhores e mais usados métodos de aprendizado supervisionados. Os métodos baseados em árvores permitem modelos preditivos com alta exatidão, estabilidade e facilidade de interpretação. Essa técnica pode ser utilizada para qualquer tipo de problema (classificação ou regressão). Apesar disso, a árvore de decisão é um tipo de algoritmo de aprendizado supervisionado que é usado principalmente em problemas de classificação. Um exemplo de uma árvore de decisão pode ser observado na figura 2.8, onde o objetivo é classificar uma pessoa entre pequena, média ou grande de acordo com sua altura.

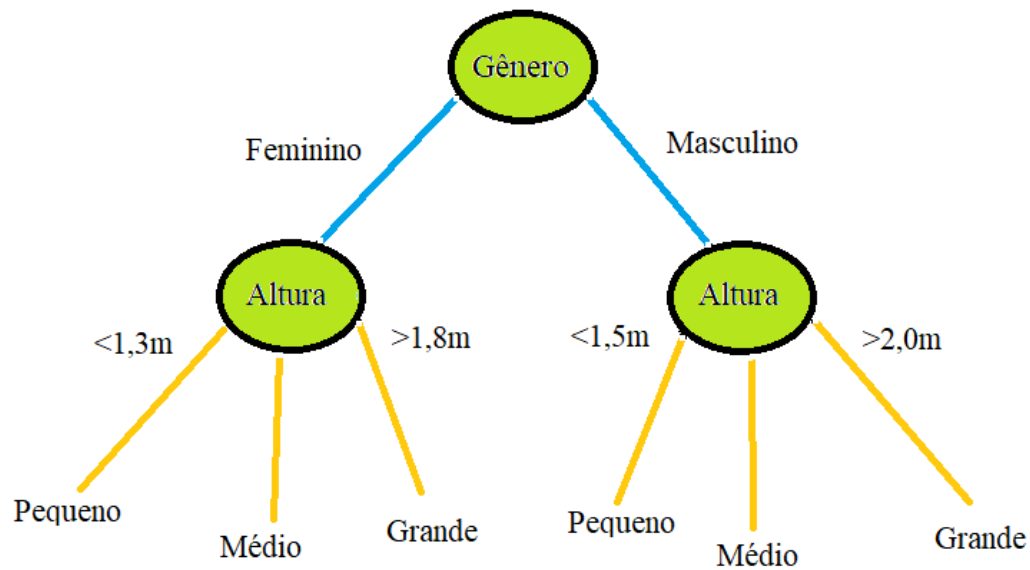


Figura 2.8 – Exemplo de árvore de decisão para classificar uma pessoa de acordo com sua altura

O algoritmo da árvore de decisão tenta resolver o problema usando a representação em árvore. Cada nó interno da árvore corresponde a um atributo e cada nó da folha corresponde a um rótulo de uma classe. Para prever uma classe, o processo é iniciado a partir da raiz da árvore. Os valores do determinado dado são comparados com o registro (obtido após o treinamento do modelo). Com base na comparação, o processo continua para a ramificação correspondente a esse valor e segue para o próximo nó. Esse processo é repetido até alcançar um nó de folha com o valor de uma classe prevista.

2.2. Sobreajuste e Subajuste

Para se utilizar a inteligência artificial é necessário ter alguns cuidados para evitar situações indesejadas, como o sobreajuste (“overfitting”) e subajuste (“underfitting”). Sobreajuste ocorre quando um modelo estatístico ou algoritmo de aprendizado de máquina captura o ruído dos dados. Ou seja, o modelo ajusta seu algoritmo especificamente de acordo com os dados de treinamento e não de acordo com os dados de validação ou teste. Assim, a exatidão para o conjunto de treino se torna extremamente elevada, porém a exatidão para o conjunto de validação ou teste se torna baixa. O sobreajuste é frequentemente o resultado de um modelo excessivamente complexo. Já o subajuste ocorre quando um modelo estatístico ou algoritmo de aprendizado de máquina não consegue capturar a tendência dos dados. Ou seja, o

subajuste ocorre quando o modelo ou o algoritmo não se ajusta bem aos dados e é frequentemente resultado de um modelo excessivamente simples. Exemplos de modelos com subajuste, sobreajuste e modelo correto estão representados na Figura 2.9 na qual deseja-se classificar uma variável entre dois estados representados pelo “X” e círculo e a linha azul é a saída do modelo onde delimita as duas áreas da classificação. Ambas situações, subajuste e sobreajuste, produzem uma baixa performance na previsão de dados novos.

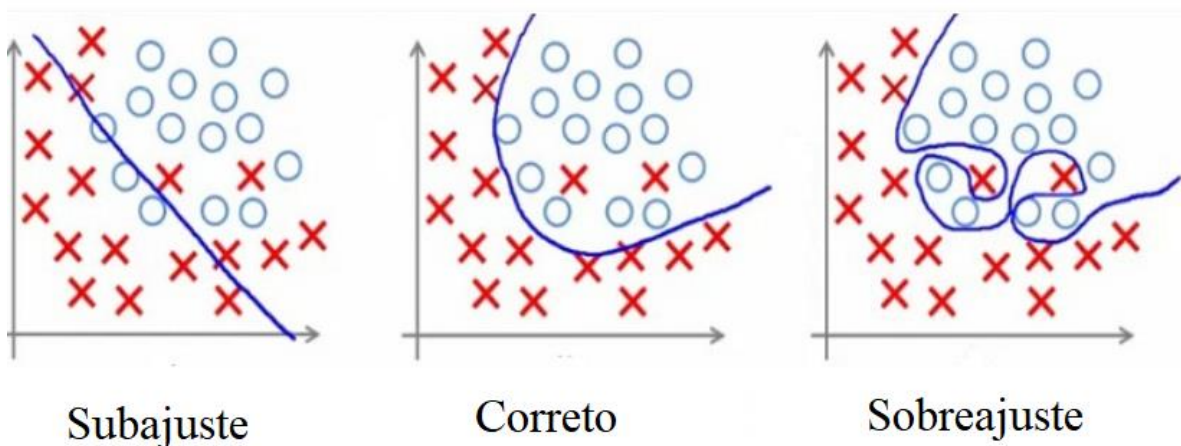


Figura 2.9 – Exemplos de subajuste, sobreajuste e um modelo correto

Fonte: <https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76>

2.3. Hiper-parâmetros

Um modelo de aprendizado de máquina é a definição de uma fórmula matemática com parâmetros que precisam ser aprendidos com os dados. Esse é o ponto crucial do aprendizado de máquina: ajustar um modelo aos dados. Isso é feito através de um processo conhecido como treinamento. Ou seja, treinando um modelo com dados existentes, a rede ajusta seus parâmetros. No entanto, há outros tipos de parâmetros que não podem ser aprendidos diretamente do processo de treinamento regular. Esses parâmetros expressam propriedades de “alto nível” do modelo, como sua complexidade ou quão rápido ele deve aprender. Eles são chamados de hiper-parâmetros. Os hiper-parâmetros geralmente têm de ser fixados antes do início do processo de treinamento real.

Em uma rede neuronal artificial há diversos hiper-parâmetros a serem dados antes do

treinamento do modelo. Alguns dos mais importantes estão representados na Tabela 2.1 abaixo.

Tabela 2.1 – Hiper-parâmetros em uma rede neuronal

Hiper-parâmetros	Explicação
Número de neurônios	Número de neurônios em cada camada da rede
Número de camadas escondidas	Número de camadas entre a camada de entrada e saída
Função de ativação	Função de ativação correspondente a cada camada da rede
Taxa de aprendizagem	Taxa em que a aprendizagem é feita no treinamento
Número de iterações	Número máximo de iterações feitas no treinamento do modelo
Número do lote	Tamanho de amostras fornecidas à rede no treinamento
Algoritmo de treinamento	Algoritmo utilizado para treinar a rede
Regularização	Técnica utilizada para evitar o sobreajuste, aumentando o poder de generalização da rede

2.4. Pré-processamento de Dados

Apesar de ser uma tecnologia inovadora, modelos de inteligência artificial não fazem o impossível. Não basta apenas introduzir qualquer conjunto de dados em sua entrada e esperar uma saída coerente. É necessário que haja um pré-processamento para que os dados sejam organizados de forma compreensível e carreguem informações relevantes para a rede neuronal. Esse processo é intensivo e demanda criatividade do analista de dados e, preferencialmente, conhecimento no assunto. Por exemplo, para transformar a interação de usuários com um determinado jogo de computador em um conjunto de dados, pode-se criar dados que contém informação dos jogadores, como o sexo do usuário, a frequência de conexão por dia, mês e ano, também pode-se computar o tempo de cada sessão ativa. Ou seja, tudo que for mensurável pode ser utilizado como um dado. O papel do analista de dados é gerar dados que contém informações úteis, ou seja, aquelas que proporcionam o algoritmo de aprendizado um modelo que realmente consegue prever um resultado com o conjunto de dados de entrada. Dados com muita relevância para o modelo são chamados também de dados com um alto poder de predição. Por exemplo, a média de duração da sessão do usuário tem um alto poder preditivo para o problema em que se deseja prever se o usuário continuará jogando o jogo no futuro.

Um conjunto de dados geralmente é composto de valores numéricos. Em uma rede

neuronal onde deseja-se obter uma alta performance, são utilizadas técnicas de processamento a fim de otimizar o aprendizado da rede e aumentar sua exatidão.

2.4.1. Técnicas de Processamento para Valores Numéricos

Uma técnica bastante utilizada no meio do aprendizado de máquinas é a normalização. Este processo converte o intervalo de valores de um dado para um intervalo definido, geralmente $[-1,1]$ ou $[0,1]$. Por exemplo, se um dado for composto de um intervalo de 1200 até 11 200, a normalização pode ser feita subtraindo 1200 e dividindo o resultado por 10 000, assim o dado estará agora no intervalo de $[0,1]$. A fórmula genérica para uma normalização nesse mesmo intervalo está demonstrada na equação 2.4, onde $\max^{(j)}$ e $\min^{(j)}$ são os valores máximo e mínimo do dado (j).

$$\bar{x}^{(j)} = \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}} \quad (2.4)$$

Normalizar os dados não é um fator obrigatório, porém, na prática, pode levar a uma velocidade de aprendizado maior. Imagine um conjunto de dados de duas dimensões, onde o primeiro dado tem um intervalo de valores de $[0, 100]$ e o segundo $[0, 0,0001]$. A primeira dimensão irá dominar a segunda e como consequência, a rede neuronal dará um peso maior ao primeiro dado, mesmo não sendo necessariamente mais importante que o outro.

Outra técnica bastante utilizada é a uniformização de valores. Esta técnica realiza o procedimento no qual os valores são processados para que tenham as propriedades de uma distribuição normal padrão com $\mu = 0$ e $\sigma = 1$, onde μ é o valor médio de todos os exemplos do dado e σ é o desvio padrão. Essa uniformização pode ser calculada utilizando a equação 2.5.

$$\bar{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}} \quad (2.5)$$

A uniformização é utilizada para comparar variáveis com unidades ou escalas diferentes e tende a otimizar o processo de treinamento, pois a condição numérica das variáveis é aprimorada.

Redes neuronais aceitam apenas valores numéricos como entrada, porém em muitos casos deseja-se também introduzir valores categóricos em uma rede devido à sua importância

para o modelo de aprendizado. Nessas ocasiões, são utilizadas técnicas de codificação dessas categorias em números.

2.4.2. *Técnicas de Codificação de Valores Categóricos*

Variáveis categóricas são variáveis que descrevem um certo atributo de um objeto, sistema ou entidade. Por exemplo as cores como vermelho, amarelo e verde representam o fato de que alguns objetos físicos irradiam uma distribuição específica do espectro eletromagnético. Essas variáveis são muito mais complexas do que uma simples distribuição numeral. Dados categóricos podem, mas não precisam, seguir uma ordem. Eles podem, muitas vezes, ser distintos ou até sobrepostos. Todos esses conceitos precisam ser levados em consideração antes de transferir um dado categórico para um conjunto vetorial numérico.

O modo mais simples de realizar essa codificação é assinalando cada categoria a um valor inteiro, por exemplo “1” para vermelho, “2” para amarelo e “3” para verde. Essa técnica é chamada de codificação ordinal e implica que há uma ordem entre os valores nesta categoria e essa ordem específica é importante para o processo de tomada de decisão. Se a ordem das categorias de uma característica não for importante, é provável que o uso de números ordenados como valores confunda o algoritmo de aprendizado, pois ele tentará encontrar um padrão onde não há. Para variáveis onde não existe uma relação ordinal, essa técnica não é suficiente. De fato, usar essa codificação e permitir que o modelo de aprendizado assuma que exista uma ordenação natural entre as categorias pode resultar em um baixo desempenho ou resultados inesperados da rede. Nesse caso pode-se utilizar a codificação *One-Hot*, que realiza a transformação das categorias em diversos números binários. Por exemplo codificar a categoria “gato” como o vetor binário [1, 0, 0], “cachorro” como [0, 1, 0] e “carro” como [0, 0, 1]. A representação de uma categoria tem o valor 1 na dimensão que representa essa categoria e 0 no restante. O tamanho do vetor é definido pelo número de categorias que a rede neuronal suportará. Cada dimensão tem o seu valor sendo 0 ou 1 e a soma de todos os vetores sempre será 1. Pode-se pensar neste vetor como sendo um vetor de probabilidades, isto significa que um “gato” é 100% gato, 0% cachorro e 0% carro ([1, 0, 0]). Então, por exemplo, se obtivermos como saída de uma rede neuronal [0,6, 0,1, 0,3] sobre [gato, cachorro, carro], podemos dizer que a rede neuronal classificou essa imagem como um gato com um nível de confiança de 60%.

2.5. Programas utilizados

Neste trabalho, foram utilizadas ferramentas com objetivos distintos em diferentes estágios para facilitar o processo. Na etapa de visualização dos dados foi utilizada a biblioteca “Matplotlib” (biblioteca de plotagem para a linguagem de programação Python), como também a função de plotagem de gráficos do Matlab (software interativo de alta performance voltado para o cálculo numérico). O pré-processamento de dados foi feito também pelo Matlab em conjunto com PyCharm (ambiente de desenvolvimento integrado usado em programação de computadores, especificamente para a linguagem Python) e Excel (editor de planilhas). Para a criação do modelo de rede neuronal foi utilizada a biblioteca “Keras” (biblioteca de rede neuronal de código aberto escrita em Python) juntamente com “Tensorflow” (biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas) e “Scikit-Learn” (biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python).

Pode-se observar que foram utilizados 3 principais programas ao longo deste trabalho: Matlab, PyCharm e Excel. Essa diversidade se dá pelo fato de que cada um apresenta pontos positivos diferentes para este projeto. O Matlab se mostrou ótimo para a visualização dos dados, o Pycharm pela sua rapidez de processamento e o Excel pela sua fácil organização.

3. Estado da Arte

O constante avanço da ciência permitiu que algumas tecnologias se desenvolvessem, tornando-as mais populares, como a Inteligência Artificial. Esse ramo vem revolucionando diversas áreas da ciência, mas agora se tornou oficialmente parte da vida cotidiana. Cada vez mais, pessoas do mundo todo estão estudando e realizando pesquisas ou trabalhos nesse tema. Nesta seção iremos falar sobre trabalhos realizados utilizando a Inteligência Artificial para prever resultados desportivos.

Purucker (1996) foi um dos primeiros a estudar a previsão de resultados desportivos usando Redes Neurais Artificiais (RNAs). O estudo foi feito coletando dados das oito primeiras jornadas da Liga Nacional de Futebol Americano (NFL), e cinco estatísticas: jardas ganhas, jardas ganhas rápidas, margem de rotatividade, tempo de posse e probabilidades das casas de apostas. Uma RNA foi usada, alcançando 61% de exatidão em comparação com 72% de exatidão de especialistas. Mais tarde, em 2003, Kahn (2003) continuou o trabalho de Purucker e acrescentou novos recursos, como diferencial total de jardas, diferencial de velocidade em jardas, diferencial de rotatividade, indicador de equipa de fora e indicador da equipa da casa. O problema foi tratado como um problema de classificação, com duas classes: vitória da equipa visitante e vitória da equipa da casa (-1 para perda e +1 para vitória). Dados sobre 208 partidas na temporada de 2003 foram utilizados. Ele alcançou 75% de exatidão, os resultados foram comparados com as previsões de oito comentadores da ESPN. Os especialistas previram uma média de 63% de partidas corretamente.

Adam Maszcyk (2011) mediu características de 140 lançadores de dardo da categoria júnior para prever os resultados dos lançamentos futuros. Entre as características estão a medição antropométrica, medidas de flexibilidade, medição de habilidades coordenativas, medição de eficiência especial, estresse, habilidade motora, habilidades de movimento, personalidade e temperamento. Os dados dos primeiros 96 atletas foram subdivididos em três tipos: 46 dados de aprendizado, 25 dados de validação e 25 dados de teste. A pesquisa foi feita usando investigações empíricas e preditivas, com o seguinte modelo: variável dependente Y (a distância do lançamento do dardo), e variáveis independentes. Para determinar o conjunto ideal de variáveis, o vetor R0 foi determinado para as variáveis independentes e o vetor R1 mostrando correlação significativa com a variável Y. As relações funcionais entre as variáveis foram encontradas por meio de técnicas de computação gráfica e aproximação quadrática de ponto médio. Essa abordagem permitiu determinar oito preditores que melhoraram significativamente

o modelo. Média e desvio padrão foram calculados para todas as variáveis. Redes neurais artificiais foram construídas, com arquitetura 4-3-1. Para generalização e predição dos resultados, foram utilizados modelos neurais de multicamadas de perceptron. As redes foram treinadas, ele então selecionou 20 novos lançadores de dardo e comparou os valores calculados com os resultados reais dos arremessos. Após 1 ano de treinamento para os 20 atletas, o erro total da rede era de apenas 2,50 m e, no caso de apenas oito lançadores, as previsões calculadas diferiam dos resultados reais na faixa de $\pm 0,01$ a $\pm 0,19$ m.

Igiri (2014) extraiu dados de 110 partidas disputadas na temporada de 2014/2015 da Premier League inglesa e os utilizou como entrada para um sistema de redes neurais. O sistema proposto foi implementado usando duas técnicas diferentes: Redes Neurais Artificiais e Regressão Logística (*Logistic Regression*). Os dados para entrada na rede neuronal escolhidos foram: golos, cantos, probabilidades de casas de apostas, potencial de ataque, performance dos jogadores, performance dos técnicos, vitórias dos técnicos, máximo de vitórias consecutivas. Todos esses dados foram colhidos para ambas as situações: jogando em casa e jogando fora de casa. Depois de coletados os dados, foi utilizado um operador específico para limpeza de dados. Este operador calcula o valor de dados ausentes utilizando os dados atuais como entrada em um modelo de aprendizagem que identifica as tendências e substitui os dados ausentes por valores reais. Em seguida, foi utilizado um módulo para otimização dos parâmetros a fim de melhorar o aprendizado da rede neuronal. Este módulo calcula os pesos dos recursos dos conjuntos de dados de futebol usando um algoritmo genético (GA). Quanto maior o peso de um atributo, mais relevante ele é considerado. Um GA é um algoritmo de busca que imita o processo natural de evolução. Ele é rotineiramente usado para criar soluções úteis de problemas de otimização e busca. Algoritmos genéticos são um aspecto de uma classe maior de algoritmos evolutivos, que produzem soluções para problemas de otimização usando técnicas como herança, mutação e seleção, bem como recombinação. Igiri descobriu com os resultados que uma alta previsão é encontrada quando nove dados são otimizados por peso. Sem a otimização, a exatidão da previsão foi de 75%, e com a otimização 85% para a temporada de 2014/2015 da Premier League. Além disso, descobriu-se que as técnicas de *logistic regression* só podiam prever resultados de vitória ou derrota. O sistema proposto poderia prever resultados de vitória, empate ou perda usando técnicas de Redes Neurais Artificiais.

Goddjin (2018) realizou um projeto para prever partidas de futebol da Premier League. Foram recolhidos dados no site *Kaggle* contendo informações de mais de 25 000 partidas de ligas europeias. As estatísticas escolhidas foram: posição de cada jogador da casa e de fora,

avaliação FIFA dos jogadores da casa e de fora e média de vitórias, derrotas e empates das equipas. As saídas desejadas eram em formas de números, sendo -1 para vitória da equipa de fora, 0 para empate e 1 para vitória da equipa da casa. Os dados finais recolhidos consistiam em 867 vitórias de equipas da casa, 1390 vitórias da equipa de fora e 783 empates, para um total de 3040 partidas de 8 temporadas diferentes. Para o treinamento, foram utilizadas 2432 partidas, já para a validação e teste foram utilizadas 304 partidas (80%, 10%, 10% respectivamente). Para o algoritmo de base, foi optado por executar uma regressão logística no conjunto de dados. Para isso, inicializou uma matriz de peso, de tamanho (3, 22) e multiplicou com o vetor de entrada que consistia em 22 características (todas as classificações de jogadores em casa e fora). A saída foi normalizada e calculada a diferença entre a saída prevista e a saída real. Em seguida, foi usado um gradiente descendente para atualizar os pesos, de modo que a diferença mencionada fosse minimizada. O segundo modelo experimentado foi uma rede neuronal de três camadas totalmente conectada. A primeira camada oculta continha 27 neurónios, a segunda continha 9 e na terceira, 3. Como resultado, o modelo de regressão logística obteve uma exatidão de 35% de resultados previstos corretamente e o modelo de redes neuronais obteve 54% de exatidão.

Podemos ver com os estudos mostrados acima que para conseguir prever o resultado de uma partida desportiva com uma exatidão relativamente alta são necessários modelos específicos para o problema, um grande conjunto de dados históricos coletados, diversas técnicas de análise e pré-processamento de dados e métodos computacionais para treino e previsão do modelo.

4. Recolha, Análise e Processamento da Base de Dados

A partir de trabalhos anteriores foi realizado um estudo sobre o tema e identificado potenciais pontos de melhorias, assim inicia-se a etapa de coleta de dados. Para isso foi feita uma pesquisa sobre os principais meios para obtenção de estatísticas de jogos de futebol. Chegou-se à conclusão de que há 3 principais meios: grupos de apoio na internet, empresas fornecedoras de dados em tempo real e sites especializados em futebol. O primeiro não se mostrou eficiente pois falta confiabilidade nos dados, visto que é feito por meio de colaboração de usuários. Já o segundo se torna bastante elevado o custo, visto que é necessário pagar mensalmente uma quantia alta para a obtenção dos dados. Portanto, foi escolhido o terceiro meio, pois apresentou uma quantidade suficiente de dados disponíveis e alta confiabilidade.

Para a extração dos dados estatísticos foi desenvolvido um programa em Python utilizando PyCharm onde se utilizou a biblioteca “Beautiful Soup” para extrair dados de arquivos HTML, a biblioteca “Selenium” para automatizar a aplicação web e “Xlwt” para converter os dados em arquivos do formato do Excel. Para a organização dos dados utilizou-se Matlab com suporte aos arquivos de Excel a fim de obter dois conjuntos de dados diferentes, um especificamente com as estatísticas das equipas e outro com as estatísticas dos jogadores. Já com os dados organizados, utilizou-se o Excel para o processamento das estatísticas na forma histórica.

4.1. Base de dados

4.1.1. Introdução

Um conjunto de dados que apresenta uma relevância para o modelo, que realmente carrega informações cruciais para a rede neuronal conseguir coletar padrões e fazer seu trabalho é a base para qualquer modelo eficiente de aprendizado de máquina. A rede neuronal necessita de uma direção para poder aprender, ou seja, de dados que realmente têm uma importância no tema e influenciam na variável de saída. Isto é, a qualidade dos dados está diretamente associada à qualidade da rede neuronal. No entanto, para uma base de dados ser de qualidade, além de conter informações importantes para a previsão, também deve-se ter uma relação positiva entre a quantidade de informações, a quantidade de amostras e o número de classes que se deseja prever. Geralmente o número de amostras deve ser maior que o número de variáveis

em um fator $x\%$ que pode variar conforme o objetivo da rede, para problemas complexos esse fator é elevado. Isso se dá pelo fato de a rede neuronal necessitar aprender a importância de cada variável para o resultado, quanto mais informações, mais difícil é essa tarefa, e conseqüentemente um maior número de amostras é essencial. Também deve-se ter um número y de amostras independentes para cada classe, onde y pode ser dezenas, centenas, ou milhares dependendo da complexidade do problema, para que o algoritmo consiga calcular todos os parâmetros que influenciam na escolha de uma determinada classe.

A criação da base de dados é sem dúvidas a principal etapa no processo de desenvolvimento de um modelo de aprendizagem e todas as informações contidas devem ser estudadas e serem proporcionais à complexidade do problema. No Estado da Arte, pode-se notar que uma grande dificuldade dos autores é manter uma balança entre uma quantidade relativamente grande de amostras com uma quantidade significativa de informações estatísticas. Isso se dá pelo fato do obstáculo que há em coletar dados de jogos de futebol.

4.1.2. Disponibilidade de Dados Sobre Jogos de Futebol

Alguns bancos de dados são disponibilizados gratuitamente em plataformas de colaboração como o Kaggle, porém não apresentam uma profundidade de dados estatísticos das partidas e muitas das vezes são produzidos por meio de colaboração dos usuários, não sendo totalmente confiáveis. Uma grande quantidade de dados de entrada é necessária para um modelo complexo. De fato, é importante ter registros históricos longos, e com estatísticas das partidas, para que a rede neuronal consiga, com esses dados como entrada, prever resultados futuros, esse é um fator chave para previsões confiáveis. No entanto, a maioria dos conjuntos de dados disponíveis não possui um número considerável de informações de datas passadas ou também são restritos apenas às principais ligas. Com isso em mente, é necessário manter um equilíbrio entre uma grande quantidade de amostras de partidas e uma boa quantidade de informações estatísticas.

4.1.3. Base de Dados 'Whoscored.com'

Neste projeto foram retirados dados do site “whoscored.com” para compor a base de dados de informações estatísticas de partidas de futebol. O site contém um equilíbrio entre uma grande quantidade de informações e histórico de partidas relativamente elevado. Além disso, apresenta uma vasta gama de ligas de diversos países, o que torna a previsão dos resultados possível para uma diversidade maior de partidas. Portanto, esses dados possibilitam manter a

balança positiva mencionada nos parágrafos anteriores. Outra vantagem deste site é que ele também apresenta seu próprio sistema de avaliação dos jogadores e equipas nesse jogo. As classificações do “WhoScored.com”, segundo o site, são consideradas os indicadores de desempenho mais precisos, respeitados e conhecidos no mundo do futebol. São baseados em seu algoritmo próprio estatístico e é calculado ao vivo durante o jogo, com mais de 200 dados para o cálculo da classificação de um jogador ou uma equipa. Todo o evento de importância é levado em conta, com um efeito positivo ou negativo sobre a classificação final. O sistema de pontuação tem seus valores entre 6 e 10 pontos, sendo 10 a pontuação mais alta. Portanto não é necessário calcular manualmente pontuações de equipas e jogadores, otimizando o processo de obtenção da base de dados. A Tabela 4.1 mostra os dados que foram obtidos e suas definições.

Tabela 4.1 – Dados estatísticos retirados das partidas e suas definições

	Dados Estatísticos	Definição
1	Golos	Número de golos feitos no tempo total da partida
2	<i>Ratings</i> dos jogadores titulares	Avaliação dos jogadores titulares na partida
3	Chutes	Remates realizados
4	Chutes à baliza	Remates na direção da baliza
5	Passes Sucedidos	Passes com sucesso feito pela equipa
6	Duelos Aéreos Sucedidos	Ganhar uma disputa aérea direta com um adversário
7	<i>Dribles</i> Sucedidos	<i>Dribles</i> com sucesso feito pela equipa
8	Tentativas de Recuperação de Bola	Número de tentativas de retirar a bola do adversário
9	Posse de Bola	Posse total de bola em percentagem
10	Rating da equipa	Qualidade geral da equipa
11	Nome do Técnico	Nome do técnico responsável pela equipa
12	Formação	Formação da equipa. Ex: 4-5-2
13	Percentagem de <i>Dribles</i> Sucedidos	Percentagem total de <i>dribles</i> com sucesso
14	Percentagem de Recuperações de Bola Sucedidas	Percentagem total de recuperações de bola com sucesso

15	Interceção	Intercetar o passe do oponente antes de chegar no destino
16	Retiradas de Posse de Bola	Perder a posse de bola sem tentar passar pelo oponente
17	Liga	Liga em que o jogo aconteceu
18	Cantos	Número total de cantos da equipa

Os dados mostrados na tabela anterior foram retirados de cerca de 28 mil partidas de 13 ligas diferentes, como pode ser observado na Tabela 4.2 abaixo.

Tabela 4.2 – Número de partidas coletadas por liga e temporadas.

Liga	Número de Partidas	Intervalo de Temporadas Coletadas
Argentina Primeira Divisão	1064	2016/2019
Campeonato Brasileiro Série A	2142	2013/2018
China Super League	407	2016/2018
Campeonato da Liga de Futebol Inglesa	933	2017/2019
Premier League Inglaterra	3507	2009/2019
Liga da França	3282	2009/2019
Bundesliga Alemã	3638	2009/2019
Série A Itália	3266	2009/2019
Campeonato Holandês Eredivise	259	2017/2019
Portugal Liga NOS	810	2016/2019
Premier League Rússia	1063	2013/2019
Liga Espanhola	3186	2009/2018
Super Lig Turquia	1203	2014/2018

Foram escolhidas ligas em que tenham uma semelhança entre elas. Todas apresentam um sistema de ranqueamento por pontos, diferente das competições em que a disputa é feita por chaves/eliminatórias, como quartas de final, semifinal, final. Competições dessa forma apresentam uma diferença muito grande dos campeonatos por pontos, pois em algumas fases não pode haver empate, ou seja, o jogo tem de ter um ganhador e conseqüentemente a rede neuronal terá que ser diferente. As partidas coletadas foram apenas as que continham todos os dados estatísticos da Tabela 4.1. O site não possui uma uniformidade quanto ao ano em que iniciou a contabilizar todas as estatísticas das partidas, por isso há uma diferença entre o primeiro ano coletado das ligas. Em algumas tem-se dados desde 2009, outras apenas desde 2017.

Os dados foram transformados para o formato xls e foi utilizada a ferramenta Excel para organizá-los em uma única tabela de forma que as partidas sejam separadas pelas linhas e os dados estatísticos por colunas. Abaixo temos um exemplo de como as partidas foram organizadas, com alguns dados estatísticos omitidos, devido à sua grande dimensão.

Tabela 4.3 – Exemplo de organização dos dados estatísticos (incompletos) de 10 partidas.

League	Date	TeamHome	TeamAway	GoalsHome	GoalsAway	H_Player1	H_Player1	H_Player2	H_Player2
Turkey Su	Fri, 22-Dec-17	Antalyaspor	Alanyaspor	3	1	Ferhat Kap	6.2	Nazim Sar	6.2
Turkey Su	Sat, 23-Dec-17	Bursaspor	Genclerbir	1	1	Harun Tek	6.29	Baris Yard	6.29
Turkey Su	Sat, 23-Dec-17	Sivasspor	Besiktas	2	1	Tolgahan A	7.48	Auremir27	7.48
Turkey Su	Sat, 23-Dec-17	Konyaspor	Fenerbahce	1	1	Serkan Kir	7.85	Nejc Skubi	7.85
Turkey Su	Sun, 24-Dec-17	Kayserispor	Yeni Malat	0	1	Silviu Lung	6.45	Cristian Sa	6.45
Turkey Su	Sun, 24-Dec-17	Karabukspor	Trabzonspor	1	1	Caglar Sal	6.03	Osman Ce	6.03
Turkey Su	Sun, 24-Dec-17	Kasimpasalar	Istanbul Bas	1	2	Ramazan H	7.46	Strahil Pop	7.46
Turkey Su	Sun, 24-Dec-17	Galatasaray	Goztepe	3	1	Fernando T	6.45	Martin Linr	6.45
Turkey Su	Mon, 11-Dec-17	Konyaspor	Karabukspor	2	0	Serkan Kir	6.72	Nejc Skubi	6.72

Na tabela acima pode-se ver um exemplo com 10 partidas da liga da Turquia, junto com a data, as equipas, os golos marcados, nomes dos jogadores titulares e seus Ratings dados pelo site. A tabela se estende até a coluna 80, contendo todas as informações da Tabela 4.1 separadas por colunas.

O conjunto de dados é formado por 46.27% de vitórias de equipas da casa, 28.08% vitórias de equipas jogando fora de casa e 25.63% de empates. Esses valores condizem com a realidade do esporte e comprovam a existência da superioridade da equipa jogando dentro de casa. Abaixo, a Figura 4.1 demonstra essa diferença, sendo no eixo vertical o número de partidas, “HOME” o resultado em que a equipa da casa vence, “AWAY” quando a equipa de fora vence e “DRAW” sendo empate. Pode-se observar que de todo o conjunto de dados, cerca

de 12 mil partidas foram ganhas pela equipa da casa. Portanto, já pode-se inferir sobre o modelo de aprendizado terá em sua saída uma distribuição parecida com a do gráfico, tendo em sua maioria, previsões de que a equipa da casa vencerá.

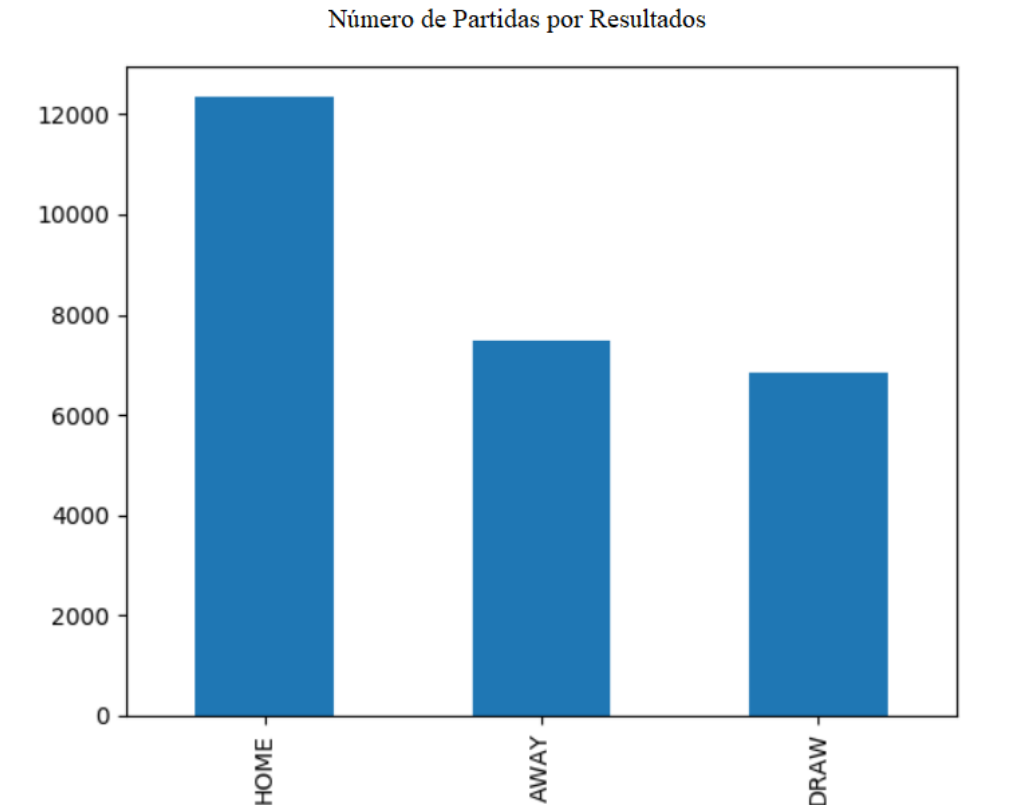


Figura 4.1 – Gráfico de números de partidas pelos resultados, respetivamente, equipa da casa venceu, equipa visitante venceu, ou empate.

4.1.4. Informações Adicionais Calculadas

Mesmo as equipas mais fortes podem ter um período de desempenho mais fraco durante uma temporada e vice-versa. Com o objetivo de captar essa periodicidade do desempenho, foi calculado o aproveitamento de cada equipa. Para o cálculo desse dado, foi utilizada a equação 4.1, onde é calculado o número total de pontos feitos pela equipa nas últimas 5 partidas utilizando o sistema mundialmente conhecido em que quando a equipa é a vencedora da partida ela ganha 3 pontos, caso empate 1 ponto e em caso de derrotas não ganha pontos. Os pontos ganhos nas partidas estão representados pelas variáveis P_1 até P_5 .

$$Aproveitamento = P_1 + P_2 + P_3 + P_4 + P_5 \quad (4.1)$$

Além do aproveitamento também foi calculada a diferença de golos feitos pelos golos sofridos nos últimos cinco jogos. Essa variável é importante, pois ela expressa a habilidade de ataque e de defesa em apenas um valor. Ou seja, caso esse número seja elevado, significa que a equipa faz mais golos do que sofre, e caso o valor seja negativo, a equipa sofre mais golos do que faz. Podemos ver a equação para a diferença de golos abaixo, onde G_m são os golos marcados em uma partida e G_s os Golos sofridos.

$$Dif_Gols = (G_{m1} - G_{s1}) + (G_{m2} - G_{s2}) + (G_{m3} - G_{s3}) + (G_{m4} - G_{s4}) + (G_{m5} - G_{s5}) \quad (4.2)$$

Outra variável adicionada foi o pi-rating de cada equipa. Esse dado tem a proposta de avaliar dinamicamente uma equipa apenas com base nas discrepâncias dos resultados. Essa estatística foi criada por Anthony C. Constantinou (2013) que comprovou ser extremamente eficaz na previsão de resultados de partidas de futebol e superando outras formas de avaliar equipas como o sistema ELO que foi originalmente inventado como um sistema aprimorado de classificação de xadrez, mas também é usado como um sistema de classificação para competição multijogador em videogames, futebol, futebol americano, basquete, baseball, tenis de mesa. Constantinou seguiu 3 regras para a criação desse sistema de avaliação: o fenómeno da vantagem quando a equipa está jogando dentro de casa; o fato em que resultados mais recentes são mais importantes do que antigos para estimar a habilidade de uma equipa; o fato em que ganhar é mais importante para uma equipa do que aumentar sua diferença de golos. Cada equipa apresenta um pi-rating para jogos dentro de casa e um para fora de casa e seu valor é atualizado a cada partida de acordo com o resultado, suas equações estão demonstradas abaixo.

$$\text{Pi-Rating da casa do time da casa} \quad \rightarrow \quad \widehat{R}_{\alpha H} = R_{\alpha H} + \psi_H(e) \times \lambda \quad (4.3)$$

$$\text{Pi-Rating de fora do time da casa} \quad \rightarrow \quad \widehat{R}_{\alpha A} = R_{\alpha A} + (\widehat{R}_{\alpha H} - R_{\alpha H}) \times \lambda \quad (4.4)$$

$$\text{Pi-Rating de fora do time visitante} \quad \rightarrow \quad \widehat{R}_{\beta A} = R_{\beta A} + \psi_A(e) \times \lambda \quad (4.5)$$

$$\text{Pi-Rating da casa do time visitante} \quad \rightarrow \quad \widehat{R}_{\beta H} = R_{\beta H} + (\widehat{R}_{\beta A} - R_{\beta A}) \times \lambda \quad (4.6)$$

À esquerda das equações são os pi-ratings recalculados para uma equipa α jogando dentro de casa (representado pelo H) e uma equipa β jogando como visitante (representado pelo A). $R_{\alpha H}$ representa o pi-rating antigo da equipa α jogando dentro de casa e $R_{\beta H}$ o mesmo, porém para a equipa β . Já $R_{\alpha A}$, o pi-rating antigo da equipa α jogando fora de casa e $R_{\beta A}$ o mesmo para a equipa β . O erro entre os golos previstos e observados está representado pela letra e nas equações, onde $\psi(e)$ é uma função que depende do erro, demonstrada na equação 4.7, γ e λ são as taxas de aprendizagem, onde Constantinou comprovou que os valores 0.7 e 0.035 são os ideais para o máximo desempenho do modelo, respectivamente.

$$\psi(e) = c \times \log_{10}(1 + e) \quad (4.7)$$

A diferença de golos esperada para cada time em particular é calculada na equação 4.8, onde G representa o campo, podendo ser visitante ou jogando em casa, b é uma constante 10 e $R_{\tau G}$ é o pi-rating da equipa τ jogando no campo G. Quando o rating de uma equipa for negativo, a diferença de golos esperados deve ser negativa.

$$\widehat{g}_{DG} = b^{\left| \frac{R_{\tau G}}{c} \right|} - 1 \quad (4.8)$$

Na equação acima, \widehat{g}_{DG} é a diferença de golos esperados contra o oponente para uma equipa que joga no campo G. Então temos \widehat{g}_{DH} para o time da casa e \widehat{g}_{DA} para o time visitante. A diferença de golos prevista final entre os dois adversários é mostrada na equação abaixo.

$$\widehat{g}_D = \widehat{g}_{DH} - \widehat{g}_{DA} \quad (4.9)$$

Por fim, o erro entre a diferença de golos previstos (\widehat{g}_D) e a diferença de golos observados (g_D) está representada na equação 4.10. Este erro é utilizado após a partida terminar para atualizar os valores de Pi-Ratings para as duas equipas.

$$e = |g_D - \widehat{g}_D| \quad (4.10)$$

Com as equações acima, foram calculados os Pi-Ratings de todas as equipas de cada partida da liga e inseridos no conjunto de dados.

4.2. Processamento de Dados

Todas as informações estatísticas obtidas foram calculadas após a partida pelo site, ou seja, todos os dados do conjunto explicados acima são dados de depois dos 90 minutos (ou mais) das partidas. Um modelo que deseja prever eventos futuros não terá essas respectivas informações, pois os acontecimentos ainda não ocorreram. Por exemplo, deseja-se prever o resultado de uma partida entre Porto e Benfica que ocorrerá em uma semana. Como a partida ainda não aconteceu, não terá dados como a posse de bola das equipas, as avaliações dos jogadores naquela partida, ou os golos marcados para introduzir no modelo de aprendizagem. Portanto o conjunto de dados demonstrado acima não pode ser utilizado como entrada para o modelo, pois “prever” o futuro com os dados do futuro não é previsão. Devem ser utilizadas informações que estejam disponíveis antes da partida começar como entradas para o modelo. Os únicos dados que estarão acessíveis antes dos jogos são os nomes das equipas, jogadores titulares, e a determinada liga. Porém, como pode-se ver na Tabela 4.1, temos diversas informações estatísticas adicionais que não vão estar disponíveis até o fim da partida. Nesses casos, utiliza-se técnicas onde dados passados são usados para prever dados futuros.

4.2.1 Organização Temporal

Existem diversos métodos para realizar esse cálculo, como a média simples, média ponderada ou a média exponencial de partidas anteriores. Douwe Buursma (2011) comprovou que utilizando uma média simples das estatísticas das últimas 20 partidas de uma equipa seria ideal para prever o resultado de uma partida de futebol com o conjunto de dados dele (caso a partida fosse antes da vigésima, utiliza-se a média das partidas existentes até ao momento). Já Gunjan Kumar (2013) apostou na média ponderada, onde atribuiu um maior peso para partidas recentes. A equação que ele utilizou está expressada abaixo para um determinado dado estatístico “x” em uma partida “t” para uma janela de “n” partidas.

$$x_t = \frac{(nx_{t-1} + (n-1)x_{t-2} + (n-2)x_{t-3} + \dots + 1x_{t-n})}{(n + (n-1) + (n-2) + \dots + 1)} \quad (4.11)$$

Kumar realizou testes e concluiu que com uma janela de 7 partidas, realizando a média ponderada, obteve um melhor resultado.

Outro método, bastante utilizado no ramo de finanças, é a média móvel ponderada exponencial. Essa técnica dá mais peso aos valores recentes por uma variável chamada de fator de suavização e, desta forma, torna-se mais sensível às novas informações, tendências e padrões. Esta é uma das principais diferenças com os outros métodos, pois responde mais rapidamente às mudanças de valores. A fórmula da média móvel ponderada exponencial está demonstrada na equação 4.12.

$$S = ax_{t-1} + (1 - a)S_{t-1} \quad (4.12)$$

A fórmula afirma que o valor da média móvel (S) no tempo t é uma mistura entre o valor do sinal bruto (x) no tempo t-1 e o valor anterior da própria média móvel, isto é, t-1. A variável a é o fator de ponderação entre a média anterior e o valor de x no instante t-1.

Baseado nos parágrafos anteriores, iremos testar os métodos mencionados e compará-los a fim de obter o melhor desempenho para o modelo.

4.2.2. *Separação entre jogos em casa e fora de casa*

Além de organizar de forma temporal, devido à vantagem do local da partida mencionada acima, os dados foram calculados separadamente, caso a equipa esteja jogando dentro de casa ou fora. Portanto, se uma equipa jogará dentro de casa, os dados estatísticos serão calculados utilizando as últimas partidas jogadas também dentro de casa. Quando ele for jogar fora de casa, os dados serão calculados utilizando as últimas partidas jogadas também fora de casa.

4.2.3. *Outliers*

Outliers ou dados anormais são dados que apresentam valores que são muito menores ou maiores que a maioria dos outros valores em um conjunto de dados. É muito importante que esses valores sejam tratados porque eles podem afetar como os dados são percebidos. Por exemplo, a equipa Werder Bremen sofreu uma derrota por 6 x 2 para o Bayer Leverkusen e a derrota acabou com a invencibilidade de 16 jogos do Werder. O Leverkusen marcar 6 golos fora é uma exceção que precisa ser resolvida ao calcular a média dos golos fora da equipa. O Bayer Leverkusen marcou no total 3 golos nos seus últimos 4 jogos fora e marcar 6 na sua

quinta é anomalia. A média de golos do Bayer nos últimos 4 jogos (excluindo o resultado de 6-2) é de 0,75 e, se somarmos o quinto jogo, a média deles será 1,8. Este é um exemplo muito bom que mostra como a média pode ser enganosa. A organização histórica neste trabalho é utilizada como forma de capturar tendências, por exemplo, um time que fez 5 golos nas últimas 5 partidas tem uma tendência de marcar 1 golo por partida. Por isso é necessário tratar os outliers antes de realizar a média, pois não são valores que estão contidos na tendência, são valores anormais ao comportamento do time.

Apesar desses valores serem reais, ou seja, não são erros do processo de coleta de dados, atrapalham o processo de aprendizagem da rede neuronal, pois afetam a distribuição das variáveis dificultando a rede neuronal de identificar padrões. Portanto, antes de realizar as médias mencionadas na seção 4.2.1, esses valores anormais serão tratados. É importante ressaltar que apenas as variáveis dependentes de entrada serão tratadas.

A fim de comprovar se há ou não anomalias, são plotados gráficos da distribuição dos valores para alguns dados do nosso conjunto. Em cada caixa, a marca central indica a mediana, e as bordas inferior e superior indicam os percentuais 25 e 75, respectivamente, dos dados. Os limites estendem-se até os pontos de dados mais extremos que não são considerados anormais, esses são plotados individualmente usando o símbolo '+'.
'+'.



Figura 4.8 – Gráfico da distribuição de valores de golos.

Pode-se ver na figura acima que a distribuição dos valores de golos feitos pela equipa da casa e da equipa de fora apresentam valores extremos, fora do intervalo normal, caracterizando-se valores anormais.

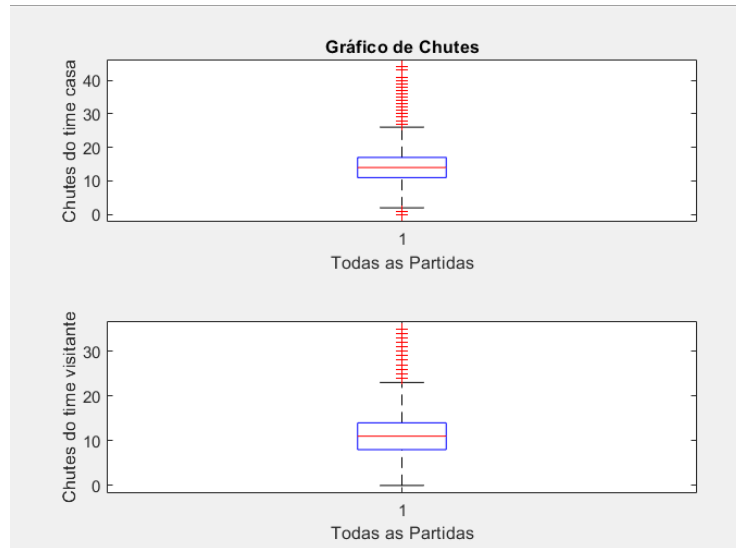


Figura 4.9 – Gráfico da distribuição de valores de chutes.

Outros dados que apresentam também valores anormais são os chutes e a posse de bola da equipa da casa e equipa visitante, como pode-se ver nas figuras 4.9 e 4.10.

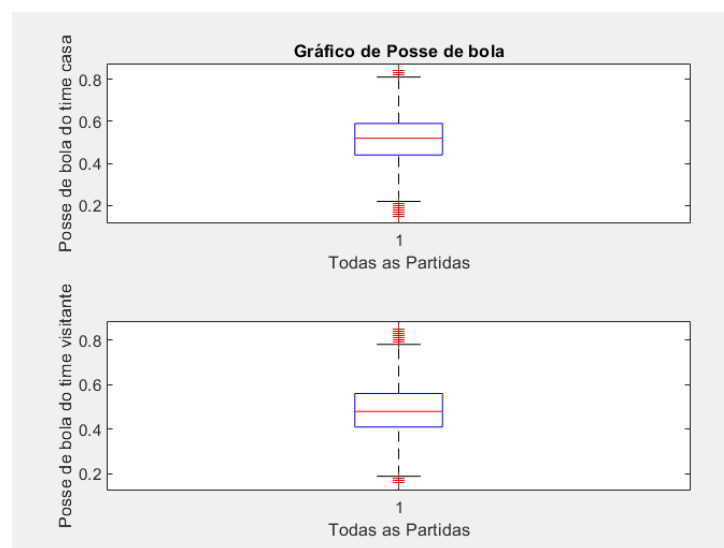


Figura 4.10 – Gráfico da distribuição de valores da posse de bola.

Após repetir esse procedimento para todas as estatísticas do conjunto de dados, foi comprovado que todas as variáveis apresentam dados anormais.

A fim de lidar com esse problema, os valores considerados extremos foram substituídos por valores máximos dentro da distribuição normal de cada dado com a função “filloutliers” do Matlab com o método “clip”. Essa função tem o objetivo de detectar e substituir valores

anormais de uma distribuição de dados.

4.2.4. *Ausência de Valores*

Pode-se notar que como os dados são calculados utilizando partidas anteriores, então nas primeiras partidas jogadas dentro de casa e fora de casa do campeonato, a equipa não terá nenhum dado disponível para o cálculo da média. Para corrigir isso, iremos utilizar as partidas que estão disponíveis para os cálculos. Por exemplo, na quarta partida de uma determinada equipa, utilizamos a média das 3 últimas partidas, já na terceira partida, utilizamos a média das últimas 2, e na segunda partida, os valores da primeira partida. Essa técnica será utilizada nos modelos apresentados na seção 4.2.1.

Com essa técnica, a quantidade de partidas sem dados se torna menor. Entretanto, ainda temos as primeiras partidas das temporadas que não terão nenhum dado estatístico. Nesses casos deve-se removê-las do conjunto de dados para que não haja erros de interpretação do modelo de aprendizagem. Por outro lado, a nível de jogadores o caso é mais complexo. Em um campeonato de futebol, a rotação de jogadores é grande, ou seja, em diversas ocasiões jogadores estarão jogando sua primeira partida pela equipa. Consequentemente não haverá dados estatísticos passados, como mencionado acima para a equipa, a única diferença é de que o número de partidas onde não haverá dados de pelo menos 1 jogador é muito maior. Com isso, se utilizarmos a mesma técnica acima, de remover as partidas, muita informação seria perdida, então é necessário utilizar outro método para lidar nessas situações.

Portanto, nesses casos onde as informações passadas dos jogadores não estão disponíveis, colocamos a média geral de todos os jogadores e, com isso, nenhuma partida será removida novamente. Por exemplo, caso o jogador de uma determinada equipa esteja jogando sua primeira partida, ele não terá uma avaliação com a média para aquela data. Ao invés de excluirmos esse jogo do conjunto de dados, colocamos a média geral de todas as avaliações dos jogadores em seu lugar. Essa técnica é bastante utilizada em situações como essa, e é chamada de substituição pela média. Em uma substituição pela média, o valor médio de uma variável é usado no lugar do valor de dados perdidos para essa mesma variável. Isso permite a utilização dos dados coletados em um conjunto de dados incompleto. A base teórica da substituição pela média é que a média é uma estimativa razoável para uma observação selecionada aleatoriamente a partir de uma distribuição normal.

4.2.5. *Normalização e Uniformização*

A normalização ou uniformização é necessária para que todas as entradas estejam em um intervalo comparável. Por exemplo, em uma rede neuronal onde se tenha duas entradas, x_1 e x_2 . A primeira variando de 0 a 0,5, já a segunda varia de 0 a 1000. Uma mudança em x_1 de 0,5 equivale a 100% de mudança, porém, uma alteração em x_2 de 0,5 é apenas uma mudança de 0,05%. Assim, esse processo é necessário para o funcionamento correto do modelo, visto que o conjunto de dados contém uma variação diferente no intervalo de valores para cada dado estatístico. Serão aplicadas essas duas técnicas explicadas no capítulo 2.4.1 a fim de compará-las e obter a melhor performance.

4.2.6. *Dados Categóricos*

O conjunto de dados contém um dado categórico que corresponde à liga em que a partida de futebol está acontecendo. Como redes neurais artificiais não suportam dados categóricos em sua entrada, será utilizada a técnica já explicada do vetor binário (“One-hot Encoding”). As ligas serão transformadas em vetores de números binários onde cada coluna representa uma liga específica. Por exemplo, caso o jogo for da Série A da Itália, que corresponde à segunda coluna do vetor binário, então o valor dessa coluna será 1 e o resto será 0.

5. Modelo de Previsão de Resultados

5.1. O Modelo

O modelo tem como objetivo representar o problema real em um universo matemático e assim, realizar previsões. Será utilizado uma rede neuronal artificial “feedforward” (do inglês, “alimentar para frente”).

A previsibilidade de um fenômeno é baseada em encontrar um padrão nas variáveis que o englobam. O futebol apresenta diversas variáveis envolvidas na decisão da partida e a relação entre elas e o resultado não é intuitiva. Além disso, é um esporte composto de seres humanos, onde o comportamento dos jogadores afeta o resultado. Portanto, o problema é de elevada complexidade e a rede neuronal deverá ser estruturada de forma que suporte isto.

Será utilizada uma rede neuronal profunda (múltiplas camadas escondidas), pois devido as camadas extras, a rede pode identificar relacionamentos não lineares complexos entre as variáveis, pois a entrada é expressa como uma composição em camadas primitivas em que cada camada da rede modela uma parte da entrada. A rede será composta por 100 neurônios na primeira camada, 80 na segunda, 5 na terceira e 3 na camada de saída. Além disso será utilizada uma técnica de regularização chamada de “dropout” (do inglês, sair fora) com a finalidade de reduzir os efeitos do sobreajuste. “Dropout” é uma técnica de regularização para modelos de redes neurais proposta por Srivastava (2014). Essa técnica faz com que neurônios selecionados aleatoriamente são ignorados durante o treinamento. Eles são “descartados” aleatoriamente, ou seja, os outros neurônios terão que intervir para fazer previsões para os neurônios ausentes. O efeito disso é que a rede se torna menos sensível aos pesos específicos dos neurônios. Isso, por sua vez, resulta em uma rede que é capaz de uma melhor generalização e tem menor probabilidade de se sobreajustar aos dados de treinamento, fenômeno conhecido como *overfitting*.

A saída da rede será composta por três valores binários (o primeiro para a vitória da equipa da casa, o segundo para o empate e o terceiro para a vitória da equipa visitante). Por exemplo, para a vitória da equipa da casa, o vetor será [1, 0, 0]; para empate: [0, 1, 0]; para a vitória da equipa visitante: [0, 0, 1]. Dessa forma, o vetor será igualmente distribuído para todos os casos de resultados mostrados acima.

Além disso, como falado anteriormente, os resultados das partidas estão desbalanceados, ou seja, há mais vitórias da equipa da casa do que empates e derrotas. Com isso, a rede tenderá a prever muito mais vitórias das equipas da casa e pouco para os outros

casos. A fim de corrigir isso, aplicaremos pesos a cada classe (vitória da equipa da casa, empate e vitória da equipa visitante) de modo que seja proporcional ao seu número de amostras. Assim, a função de erro torna-se ponderada, em que o peso de cada amostra é especificado. Com essa técnica, as saídas do modelo serão mais distribuídas e, conseqüentemente, haverá um aprendizado melhor.

O conjunto para o treinamento da rede será composto por 20 712 partidas, e o conjunto para a validação e teste terão 2 588 partidas cada, de todas as ligas citadas. Os conjuntos de validação e teste foram organizados de forma que a concentração de partidas das ligas seja a mesma, por exemplo: caso tenha 25% de partidas do Brasileirão no conjunto de treino, o conjunto de validação e teste também terão 25%. Além disso, a data em que ocorreram os jogos é organizada de forma crescente, do conjunto de treino para validação e teste, nesta ordem. Com isto, a rede irá prever partidas futuras utilizando dados passados.

A resposta da Rede Neuronal estará na forma de um vetor, no entanto, mesmo que ela tenha sido composta por valores binários no conjunto de dados do treinamento, a rede irá fornecer saídas numéricas fracionadas e as probabilidades dos acontecimentos podem ser extraídas desses valores. Por exemplo, se a resposta for [0.49, 0.31, 0.20], a equipa da casa terá 49% de chances de vencer a partida, 31% de empate e a equipa visitante terá 20% para vencer. Ou seja, para esta partida a equipa da casa provavelmente vencerá. A fim de manter a soma das saídas em 1 (100%), foi utilizada a função de ativação “softmax” na camada de saída da rede neuronal. *Softmax* é uma função que recebe como entrada um vetor de números reais e o normaliza em uma distribuição de probabilidades, ou seja, cada saída terá uma distribuição entre [0,1] e a soma de todas as saídas será sempre 1. O diagrama do modelo pode ser visto na figura abaixo.

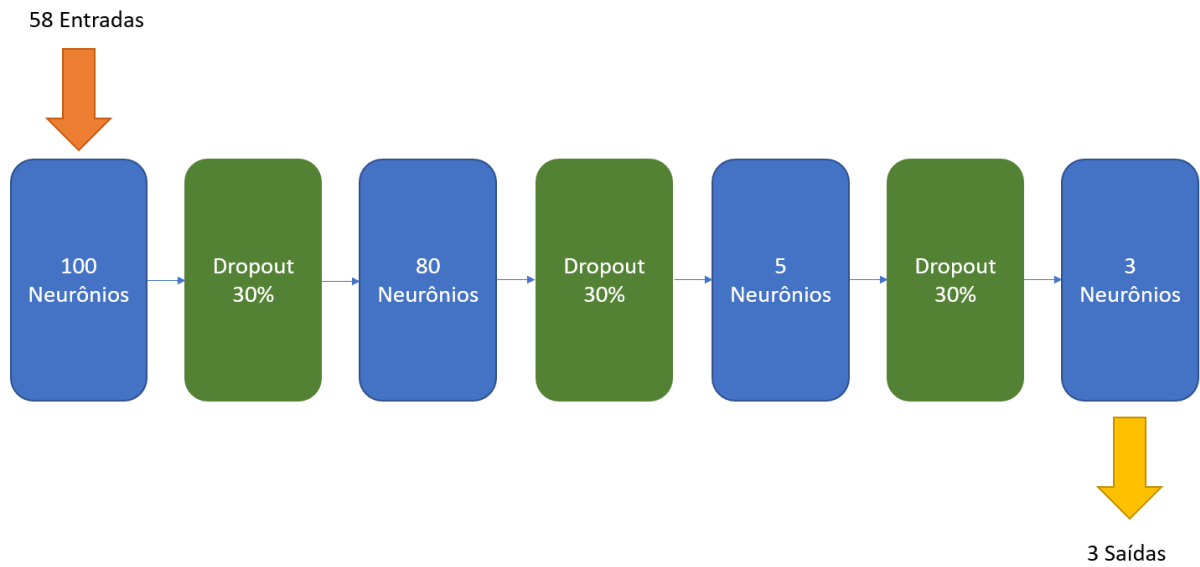


Figura 5.1 – Diagrama da Rede Neuronal proposta

5.2. Resultados Iniciais

O modelo foi testado com todas as variáveis explicadas na seção 4, de diferentes formas, a fim de determinar o conjunto com a melhor performance. Os resultados da exatidão para os conjuntos de teste serão mostrados e analisados nas tabelas seguintes. A exatidão foi calculada pela soma de partidas previstas corretas (p_c) divididos pelo número total de partidas analisadas (p_t), como visto na equação abaixo.

$$Exatidão = \frac{p_c}{p_t} \quad (5.1)$$

Tabela 5.1 – Resultados dos diferentes tipos de organização temporal utilizando a uniformização como forma de processamento

Organização Temporal	Alpha	Exatidão
Média Ponderada de 7 Partidas		49.3%
Média Simples de 20 Partidas		51.3%
Média Móvel Ponderada Exponencial	0.2	49.4%
Média Móvel Ponderada Exponencial	0.4	50.9%
Média Móvel Ponderada Exponencial	0.6	49.7%
Média Móvel Ponderada Exponencial	0.8	49.3%

Tabela 5.2 – Resultados dos diferentes tipos de organização temporal utilizando a normalização como forma de processamento

Organização Temporal	Alpha	Exatidão
Média Ponderada de 7 Partidas		49.9%
Média Simples de 20 Partidas		51.8%
Média Móvel Ponderada Exponencial	0.2	50.6%
Média Móvel Ponderada Exponencial	0.4	51.3%
Média Móvel Ponderada Exponencial	0.6	51.2%
Média Móvel Ponderada Exponencial	0.8	50.5%

Pode-se observar que a técnica da média ponderada das últimas 20 partidas apresenta uma melhor performance diante das outras. E, também, a normalização entre mínimos e máximos se mostrou melhor do que a uniformização. A normalização e uniformização são utilizadas para atingir um objetivo semelhante, que é criar recursos que tenham intervalos semelhantes entre si e amplamente utilizados para ajudar o modelo de aprendizagem a identificar padrões entre os dados. Porém, são técnicas que dependem do conjunto de dados, no nosso caso, a normalização obteve a melhor performance.

Esses dados são importantes para a escolha da organização temporal e técnica de processamento. Com o conjunto que obteve a maior performance podemos passar para a próxima etapa e investigar mais à fundo os dados estatísticos e realizar mais melhorias ao modelo.

5.3. Melhorias

A criação de um modelo de aprendizagem de máquina é geralmente um processo complexo com diversas etapas a serem concluídas. A definição da organização histórica dos dados e a técnica de processamento foi concluída no capítulo anterior. O modelo de melhor performance obteve 51,8% de acertos no conjunto de teste. O objetivo agora é melhorar a exatidão utilizando diversas técnicas em diferentes etapas da criação do modelo.

5.3.1. Conjunto de dados

Como dito anteriormente, o conjunto de dados é um dos principais componentes em aprendizado de máquina e afeta diretamente o desempenho do modelo. Os dados utilizados para treinar a rede têm uma grande influência no seu desempenho, recursos irrelevantes ou parcialmente relevantes podem impactar negativamente. Nessa seção analisaremos a relevância dos dados estatísticos para a previsão.

Árvores de decisão serão utilizadas para obter uma visão geral da importância dos dados para um modelo de previsão. Um benefício dessa técnica é que depois que as árvores são construídas, é relativamente fácil calcular a importância de cada atributo. Geralmente, a importância fornece uma pontuação que indica quão útil ou relevante cada dado foi na construção das árvores de decisão dentro do modelo. Quanto mais um dado é usado para tomar decisões nas ramificações da árvore, maior é sua importância relativa. Essa importância é calculada e quantificada para cada atributo no conjunto de dados, permitindo que eles sejam classificados e comparados entre si. A importância é calculada pela soma de todas as contribuições de um determinado dado nas ramificações da árvore, quanto maior esse valor, mais importante a variável será.

A importância será calculada separadamente para a previsão das três situações distintas: vitória da equipa da casa, empate e vitória da equipa visitante. Assim, é possível identificar quais variáveis são importantes para a previsão dos 3 casos. As figuras 5.2, 5.3 e 5.4 ilustram a relevância dos dados estatísticos para cada caso, sendo “H” correspondente à equipa da casa e “A” à equipa visitante.

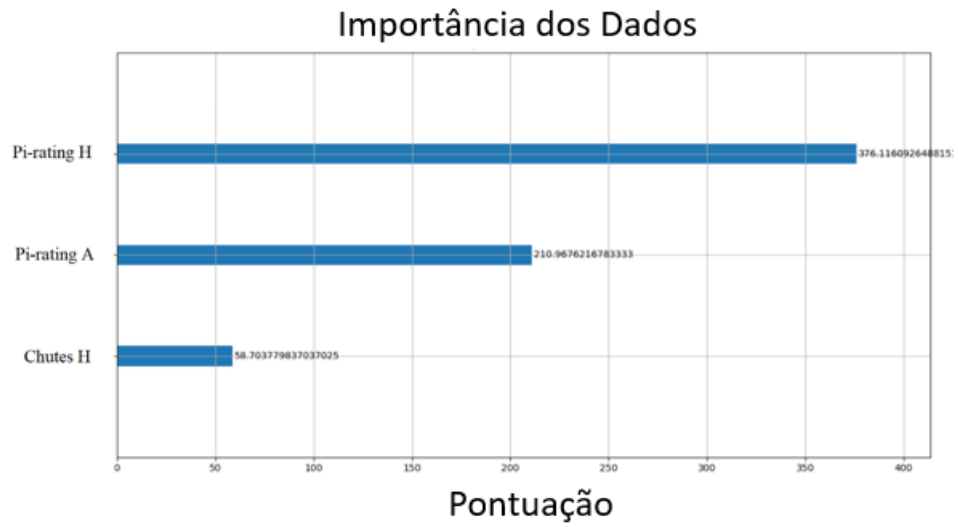


Figura 5.2 – Importância dos dados no processo de previsão da vitória da equipa da casa.

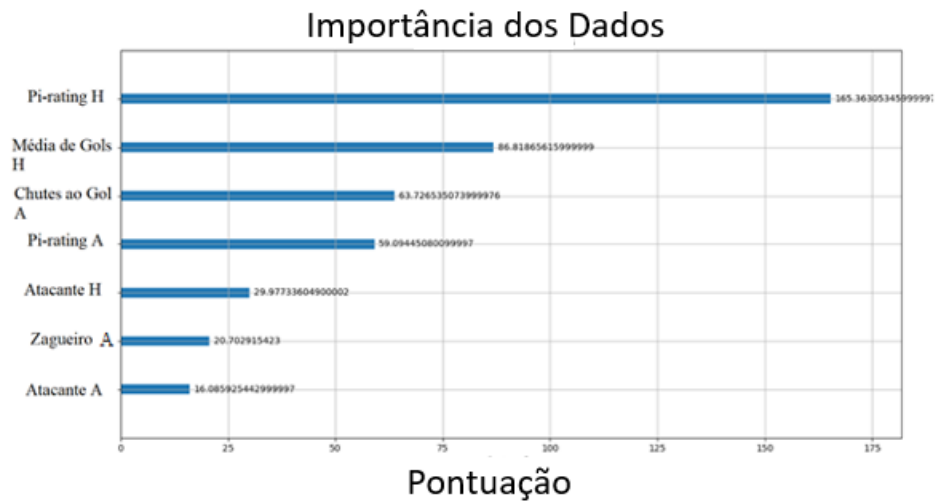


Figura 5.3 – Importância dos dados no processo de previsão do empate

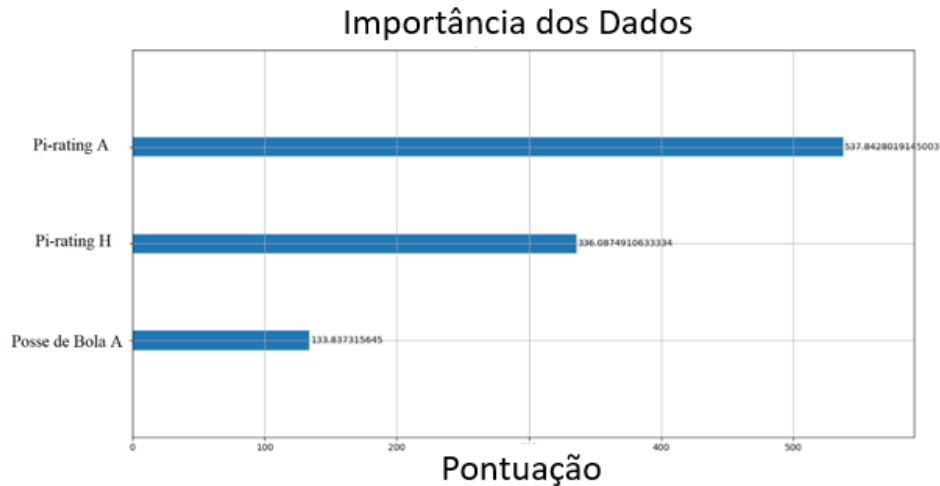


Figura 5.4 – Importância dos dados no processo de previsão da vitória da equipa visitante

Pode-se ver nos gráficos das figuras 5.2 e 5.4 que de 58 variáveis, apenas 3 se mostraram relevantes, já no gráfico da figura 5.3, utiliza-se mais variáveis para a previsão. Também pode-se observar que os Pi-ratings têm uma relevância muito maior que os dados estatísticos prontos, como cantos, chutes etc. Portanto, apenas com essa análise inicial pode-se observar que a rede provavelmente não está conseguindo interpretar a informação contida nesses dados. Por isso, é necessária a manipulação dessas variáveis a fim de melhorar a interpretação e o aprendizado da rede neuronal.

A fim de maximizar a habilidade da rede de identificar padrões e melhorar a performance, os dados serão transformados de uma maneira eficiente para o modelo. A transformação é um passo comum pois os dados podem conter informações importantes que apenas estarão visíveis no seu estado transformado. Portanto, inicialmente iremos subtrair as variáveis da equipa da casa pela equipa visitante. Ou seja, ao invés de ter uma variável por equipa, por exemplo escanteios/cantos da equipa A, escanteios/cantos da equipa B, agora teríamos apenas uma que engloba as duas equipas, ou seja, escanteios/cantos da equipa A menos escanteios/cantos da equipa B. Com isso, o número de entradas seria reduzido pela metade e a informação não seria perdida, seria até mesmo mais eficiente, pois já representaria uma relação entre as duas equipas.

Como falado anteriormente, alguns dados somente terão importância em sua forma transformada. O segundo passo na melhoria do conjunto de dados é a geração de novas variáveis. Elas serão criadas a partir da fusão das estatísticas com o objetivo de representar uma informação útil para a previsão do resultado. Abaixo explicaremos os novos dados criados.

- *Poder de Ataque, Defesa e Meio de Campo*

Esses dados têm como objetivo representar o poder das equipas em áreas diferentes do campo. Eles foram gerados a partir da avaliação dos jogadores. Ou seja, para o cálculo do poder de ataque da equipa, foram somadas as avaliações dos 3 jogadores mais à frente em sua formação. Já para o poder de meio de campo, foram somadas as avaliações dos jogadores do meio de campo, e para o poder de defesa, os zagueiros/defesas. Com esses três novos dados, também foram calculados os poderes relativos entre a equipa da casa com a equipa visitante e vice-versa. Assim, os poderes relativos de ataque da equipa “a” é gerado dividindo as avaliações dos jogadores atacantes da equipa “a” pela avaliação dos zagueiros/defesas da equipa “b”, também foram calculados os poderes relativos do meio de campo da mesma maneira. Podemos observar a equação do poder relativo de ataque da equipa “a” abaixo. “ Av_{aa1} ” significa avaliação do jogador atacante da equipa “a” e “ Av_{bd1} ” a avaliação do jogador da defesa da equipa “b”. Lembrando que não é necessário se preocupar com a grandeza dessas variáveis, pois depois será feita a sua normalização, como mencionado.

$$\text{Poder Relativo de Ataque do time } a = \frac{(Av_{aa1} + Av_{aa2} + \dots + Av_{aan})}{(Av_{bd1} + Av_{bd2} + \dots + Av_{bdn})} \quad (5.2)$$

- *Expectativa de Golo*

A fim de calcular as chances de uma equipa “a” fazer golo à equipa “b” dividimos os chutes ao gol da equipa “a” pela avaliação do goleiro/guarda-redes da equipa “b”. Assim, esse dado carregará a informação da expectativa de uma equipa fazer golo na outra. Sua equação está demonstrada abaixo. Foram calculadas as expectativas para ambas equipas.

$$\text{Expectativa} = \frac{\text{Chutes ao Golo da equipa } a}{\text{Avaliação do goleiro (guarda – redes) da equipa } b} \quad (5.3)$$

- *Exatidão de Chutes/Remates*

A exatidão dos chutes/remates é calculada dividindo a média de golos pela média de chutes/remates de uma equipa. Foram calculadas as precisões de ambas equipas e sua equação está demonstrada abaixo.

$$\text{Precisão} = \frac{\text{Media de Golos}}{\text{Media de Chutes}} \quad (5.4)$$

- *Poder Relativo do Escanteio/Canto*

Esse dado tem como objetivo carregar a informação do poder que uma equipa apresenta em seus escanteios/cantos. Para isso, foram utilizados o número de escanteio/cantos e a percentagem de duelos aéreos sucedidos. Sua equação está demonstrada abaixo.

$$\text{Poder Relativo de Escanteio} = \frac{\text{Escanteios}_a \times \text{Duelos Aéreos}_a}{\text{Escanteios}_b \times \text{Duelos Aéreos}_b} \quad (5.5)$$

- *Perda Relativa de Bola*

Essa variável traz o poder relativo da perda de bola para uma equipa “a” em relação à equipa “b”. Ela é calculada multiplicando o número de perda de bolas da equipa “a” pela percentagem de roubadas/recuperações de bolas da equipa “b”, em seguida fazemos o mesmo para a equipa “b” e dividimos os dois para obter o dado final. Sua equação está demonstrada abaixo.

$$\text{Perda Relativa} = \frac{\text{Perda de bolas time a} \times \text{Roubadas de bola time b}}{\text{Perda de bolas time b} \times \text{Roubadas de bola time a}} \quad (5.6)$$

- *Passes e Interceptações*

A última variável gerada representa o poder de criação de jogadas que uma equipa tem em relação à outro. Portanto, são utilizadas as taxas de acertos de passes e número de

interceptações de ambas as equipas. Sua equação está representada abaixo.

$$Criação = \frac{\frac{Porcentagem\ de\ Passes\ Sucedidos\ time\ a}{Interceptações\ time\ b}}{\frac{Porcentagem\ de\ Passes\ Sucedidos\ time\ b}{Interceptações\ time\ a}} \quad (5.7)$$

Após a criação dos novos dados explicados acima, será feita uma análise mais profunda do impacto que cada variável do conjunto de dados tem sobre a rede neuronal. A importância de um dado pode ser calculada pelo aumento do erro na previsão do modelo depois que esse dado é permutado aleatoriamente, pois a relação entre essa variável e o resultado é cortada. Ou seja, um dado é importante se depois de “embaralhado” aleatoriamente seus valores o erro do modelo aumenta, pois nesse caso o modelo dependia da variável para a previsão. Um dado não tem importância se o erro do modelo não se altera mesmo depois dos valores da variável serem permutados aleatoriamente, pois nesse caso o modelo não necessita do dado para a previsão.

Essa técnica foi introduzida por Breiman (2001) e mais tarde Fisher, Rudin e Dominici (2018) propuseram um modelo para o cálculo da importância de um dado para um modelo de previsão. Utilizaremos os seguintes passos para o cálculo das importâncias dos dados:

1. Calcular a exatidão utilizando o conjunto de dados completo (ϵ_c)
2. Permutar os valores de uma variável
3. Calcular a exatidão para esse novo conjunto de dados (ϵ_n)
4. Calcular a importância da variável com a seguinte equação: $\epsilon_c - \epsilon_n$

Os passos acima serão feitos para todas as variáveis do conjunto de dados a fim de calcular suas relevâncias para o modelo.

Redes neuronais apresentam uma característica de inicialização aleatória em sua fase de treinamento. Os pesos dos neurônios são iniciados de forma diferente em cada treinamento, assim a rede apresenta diferentes pontos de partida para cada simulação realizada. Portanto é normal uma rede neuronal artificial apresentar resultados semelhantes, porém não exatamente iguais a cada treinamento. Essa situação é essencial para se chegar na melhor performance do modelo. Portanto, a fim de descobrir a importância das variáveis, os passos de 2 a 4 serão repetidos 30 vezes para cada dado, e a maior performance será utilizada.

Após esse procedimento, foi concluído que 2 dados estatísticos não são relevantes para o

modelo, ou seja, o resultado não se alterou ou até mesmo melhorou com a permutação dessas variáveis, são eles: as avaliações dos jogadores zagueiros/defesas 3 e 5 de ambas equipas. Portanto eles serão retirados do conjunto de dados. Quando permutado os valores dos jogadores da posição 3 (sendo posição 1 goleiro/guarda-redes e posição 11 atacante) houve um aumento de 0.2% na exatidão do modelo, e quando permutado os jogadores 5 não houve melhoras nem pioras na exatidão. Também se chegou à conclusão de que as variáveis mais relevantes na ordem decrescente são: poder de meio de campo da equipa visitante, avaliação de ambas equipas, percentagem de dribles sucedidos de ambas equipas. Os resultados completos podem ser vistos na tabela abaixo.

Tabela 5.3 – Resultado da análise da importância pela permutação das variáveis.

Variáveis	Importância ($\epsilon_c - \epsilon_n$)
Meio de Campo Equipa Visitante	1.00
Avaliação das Equipas	0.97
Dribles Sucédidos	0.93
Defesa Equipa da Casa	0.89
Dribles Ganhos	0.81
Tentativas de Roubadas/Recuperações de Bola	0.81
Ataque Equipa Visitante	0.81
Passes e Interceptações	0.81
Poder de Meio de Campo Relativo	0.77
Expectativa de Golo Equipa da Casa	0.77
Jogadores 9	0.73
Pi-rating Equipa da Casa	0.73
Pi-rating Equipa Visitante	0.73
Jogadores 6	0.70
Passes Sucédidos	0.70
Expectativa de Golo Equipa Visitante	0.70
Jogadores 1	0.66
Chutes/Remates	0.62
RoubadasRecuperações de Bola Sucédidas	0.62
Exatidão de Chutes Equipa da Casa	0.62

Perda Relativa de Bola	0.62
Jogadores 8	0.58
Jogadores 4	0.54
Jogadores 11	0.54
Duelos Aéreos Sucedidos	0.54
Interceptações	0.54
Exatidão de Chutes Equipa Visitante	0.54
Chutes ao Golo	0.50
Defesa Equipa Visitante	0.50
Jogadores 7	0.46
Média de Golos	0.46
Posse de Bola	0.46
Forma das Equipas	0.46
Diferença de Golos	0.46
Ataque Equipa da Casa	0.46
Poder de Ataque Relativo	0.46
Média de Golos Sofridos	0.42
Perda de Bola	0.42
Escanteios/Cantos	0.42
Poder Relativo de Escanteio/canto	0.35
Jogadores 10	0.27
Meio de Campo Equipa da Casa	0.27
Jogadores 2	0.15
Jogadores 5	-0.04
Jogadores 3	-0.23

5.3.2. Seleção Automática de Hiperparâmetros

A fim de obter os melhores parâmetros da rede neuronal será utilizada a técnica de pesquisa em grade (“grid-search”). A pesquisa em grade é o processo de varredura dos dados para configurar os parâmetros ideais para um determinado modelo. Dependendo do tipo de modelo utilizado, certos parâmetros são necessários. Como mostrado anteriormente, redes

neurônios artificiais têm diversos hiperparâmetros para serem otimizados.

A pesquisa em grade pode ser aplicada no aprendizado de máquina para calcular os melhores parâmetros a serem usados em qualquer modelo. É importante observar que essa técnica pode ser extremamente pesada em termos de computação e levar muito tempo para que seja executada. O “Grid-Search” criará um modelo para cada combinação de parâmetros possível. Ele itera através de cada combinação de parâmetros e armazena um modelo para cada combinação. No fim, obteremos o melhor conjunto de hiperparâmetros possível.

6. Resultados e Discussão

Depois de realizadas todas as alterações do capítulo anterior o modelo obteve, entre outros, 60,9% de exatidão nas 82 partidas de 2019 da Liga NOS de Portugal, 58,1% nas 122 partidas da Super Liga da Turquia no ano de 2018 e 2019 e 57,2% nas 360 partidas da Premier League da Inglaterra, como visto na tabela abaixo.

Tabela 6.1 – Exatidão do Modelo para cada Liga.

Liga	Partidas Previstas	Exatidão
Liga NOS Portuguesa	82	60,9%
Super Liga da Turquia	122	58,1%
Premier League Inglaterra	360	57,2%
Série A Italiana	357	56,5%
Campeonato Inglês	94	56,3%
La Liga Espanhola	321	55,4%
Brasileirão	215	52,5%
Super Liga Chinesa	62	50,0%
Eredivise Holandesa	28	46,4%
Premier League Russia	114	45,6%
Liga Francesa	356	45,2%
Primeira Divisão Argentina	108	44,4%
Bundesliga Alemã	370	41,3%
Total	2589	52,8%

O modelo obteve no geral, 52,8% de exatidão nas 2589 partidas do conjunto teste. Porém, analisando os resultados, quando a rede neuronal preveu com uma probabilidade maior que 50% que a equipa da casa iria ser o ganhador, o modelo obteve uma exatidão de 78,8%. Já no caso onde a probabilidade era maior que 60% a exatidão subiu para 80,3%. Por outro lado, para a equipa visitante a exatidão foi de 76,6% para quando a probabilidade era maior que 50%. Na previsão de empates, o modelo não obteve uma boa performance, acertando apenas 32% das partidas em que previu o resultado como sendo empate. Portanto, pode-se concluir que o modelo não consegue prever empates entre as equipas.

Como pode-se observar na Tabela 6.1, o modelo apresentou uma diferença significativa entre as exatidões de diferentes ligas. Essa diferença pode ter duas explicações possíveis, há uma diferença na concentração dos resultados das partidas das ligas no conjunto teste ou

algumas ligas são realmente mais previsíveis do que outras, por fatores externos. Na primeira hipótese, se o modelo apresentasse uma tendência de prever mais resultados como vitória da equipa da casa, então em um conjunto de dados onde a maioria dos resultados foram de vitória da equipa da casa a performance do modelo seria elevada. Porém em conjuntos onde há uma concentração de empates ou vitória da equipa da casa maior, a performance do modelo seria pior. Por exemplo, suponha que um modelo apenas prevê o resultado como sendo empate, independentemente de sua entrada. Em um conjunto de dados de 100 partidas, onde 90 foram realmente empates, o modelo terá uma performance de 90%. Isto é um exemplo de um modelo de aprendizado falso, onde os resultados não condizem com a sua real performance.

A fim de testar essa hipótese, a concentração dos resultados das partidas do conjunto de teste utilizado para calcular a exatidão do modelo pode ser vista na tabela abaixo.

Tabela 6.2 – Concentração dos resultados no conjunto de teste.

Liga	Vitórias do Time da Casa	Empate	Vitória do Time Visitante
Liga NOS Portuguesa	45,1%	23,1%	31,8%
Super Liga da Turquia	49,1%	22,9%	28,0%
Premier League Inglaterra	49,1%	18,3%	32,6%
Série A Italiana	44,2%	26,3%	29,5%
Campeonato Inglês	50,0%	19,1%	30,9%
La Liga Espanhola	47,6%	23,6%	28,8%
Brasileirão	53,0%	28,8%	18,2%
Super Liga Chinesa	54,8%	20,9%	24,3%
Eredivise Holandesa	46,4%	28,5%	25,1%
Premier League Russia	39,4%	28,9%	31,7%
Liga Francesa	43,2%	29,2%	27,6%
Primeira Divisão Argentina	41,6%	32,4%	26,0%
Bundesliga Alemã	41,0%	29,7%	29,3%

As Tabelas 6.1 e 6.2 são comparadas a fim de verificar um padrão entre a concentração dos resultados no conjunto teste de cada liga e sua exatidão. Como pode ser observado, a concentração de vitórias do time da casa não está diretamente ligada à exatidão do modelo, por exemplo, a Liga Nos de Portugal apresenta uma concentração de 45,1% e uma exatidão de 60% enquanto que o Brasileirão apresenta uma concentração de 53% e exatidão de 52%, já a Primeira Divisão Argentina apresenta uma concentração de 41% e exatidão de 44%. O mesmo se repete para a análise da concentração da vitória do time visitante.

Em relação à concentração de empates no conjunto teste e a sua relação com a exatidão, pode-se observar que há um padrão de que quanto menor sua concentração, melhor a exatidão, esta característica pode ser explicada pelo fato de o modelo não obter uma performance boa na previsão de empates, como falado anteriormente, porém essa relação não é muito expressiva e não se mantém para todas as ligas.

Com as análises acima, a primeira hipótese foi descartada e a explicação da diferença de exatidão entre as ligas se dá pelo fato de que algumas ligas são realmente mais previsíveis do que outras, isso pode se dar ao fato de diversos fatores, como a diferença entre as jogabilidades das ligas, estilo de jogo, visibilidade da liga, e etc. Isto é positivo para o modelo, pois mostra que a sua performance não apresenta relação com a concentração dos dados do conjunto de teste.

7. Conclusões e Trabalhos Futuros

7.1. Conclusões

A principal característica de jogos desportivos é a sua imprevisibilidade. O fato de uma equipa poder fazer golo a qualquer momento é o que torna o desporto mundialmente conhecido. Este trabalho teve como objetivo verificar se é possível prever resultados de jogos de futebol utilizando redes neuronais artificiais.

Foram realizadas diversas etapas e técnicas para a criação do modelo final de previsão. O modelo é composto de uma rede neural profunda com 3 camadas escondidas e seus hiperparâmetros foram escolhidos automaticamente para sua melhor performance. Além disso, o conjunto de dados foi pré-processado de forma a otimizar o treinamento da rede e diversos novos dados foram calculados com o objetivo de melhorar sua performance.

O modelo proposto obteve 52,8% de exatidão no conjunto de cerca de 2500 partidas, chegando até 80,3% para casos específicos. Após a análise feita com a performance da rede para as diferentes ligas, foi verificado que a exatidão do modelo não é devido à concentração dos resultados do conjunto de teste. Fato que comprova a aprendizagem da rede neural a fim de prever resultados de partidas de futebol.

O futebol apresenta 3 resultados possíveis, vitória da equipa da casa, empate ou vitória da equipa visitante. Se esse esporte fosse totalmente imprevisível, qualquer método de previsão teria no máximo 33,3% de exatidão, visto que são 3 resultados possíveis para 1 escolha apenas ($1/3=0.33=33\%$). O modelo proposto obteve 52,8% de exatidão no conjunto de cerca de 2500 partidas de 2018 e 2019 com dados anteriores a essas datas. Ou seja, a rede neuronal conseguiu com uma margem positiva de 19,5% (52,8% previstos pelo modelo menos os 33% se o desporto fosse imprevisível) identificar os parâmetros principais que afetam o resultado e com isso criar uma equação matemática para prever partidas futuras com dados estatísticos passados. Analisando mais à fundo os resultados, como mostrado, essa margem pode aumentar até 47,5% para casos específicos (quando a rede neuronal previu que o time da casa iria ganhar com mais de 60% de chance).

Com os resultados obtidos, pode-se afirmar que é possível prever resultados de partidas de futebol com uma exatidão relativamente boa utilizando redes neuronais artificiais e dados estatísticos de partidas passadas.

7.2. Trabalhos Futuros

A fim de complementar este trabalho, será proposto um aplicativo onde os dados estatísticos de partidas de futebol serão extraídos automaticamente e o usuário poderá prever partidas futuras apenas escolhendo, na plataforma de interface, as equipas que jogarão. O aplicativo dará a resposta em forma de probabilidades de cada equipa ganhar e o empate acontecer. Esse aplicativo tornará o processo da previsão de partidas de futebol mais simples e poderá ser utilizado no dia a dia.

BIBLIOGRAFIA

AKHTAR, F., Hahne. **C.RapidMiner 5 Operator Reference**. Rapid-I GmbH, 2012, Disponível em: http://rapidminer.com/wp-content/uploads/2013/10/RapidMiner_OperatorReference_en.pdf.

ANDERSON, E. **The species problem in Iris**. Annals of the Missouri Botanical Garden, 23 (3): 457–509, 1936.

ANGELINI, Giovanni, Luca De Angelis. **Efficiency of online football betting markets**. 712–721, International Journal of Forecasting 35, 2019.

BABOOTA, Rahul e Harleen Kaur. **Predictive analysis and modelling football results using machine learning approach for English Premier League**. 741–755, International Journal of Forecasting 35, 2019.

BRADLEY, R. **16 Examples of Artificial Intelligence (AI) in Your Everyday Life**. The Manifest, 26 de setembro de 2018.

BREIMAN, Leo. Random Forests. Machine Learning 45 (1). Springer: 5-32, 2001.

BUHAGIAR, Ranier e Dominic Cortis e Philip W.S. Newall. **Why do some soccer bettors lose more money than others?** 85–93, Journal of Behavioral and Experimental Finance 18, 2018.

BUNKER, Rory P. e Fadi Thabtah. **A machine learning framework for sport result prediction**. 27–33, Applied Computing and Informatics 15, 2019.

BURKOY, Andriy. **The Hundred-Page Machine Learning Book**.

BUURSMA, Douwe. **Predicting sports events from past results**, Towards effective betting on football matches. University of Twente P.O. Box 217, 7500AE Enschede The Netherlands.

CHO, Yoonjae e Jaewoong Yoon e Sukjun Lee. **Using social network analysis and gradient boosting to develop a soccer win–lose prediction model.** 228–240, Engineering Applications of Artificial Intelligence 72.

CONSTANTINOU, Anthony C. e Norman E. Fentom. **Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries.** Vol. 9, Iss. 1, 37–50, Journal of Quantitative Analysis in Sports, 2013.

Disponível em: <http://deeplearningbook.com.br/entropia-cruzada-para-quantificar-a-diferenca-entre-duas-distribuicoes-de-probabilidade/>. Acesso em: 29 de setembro de 2018.

Disponível em: <http://deeplearningbook.com.br/funcao-de-ativacao/>. Acesso em: 29 de setembro de 2018.

Disponível em: <http://mathworld.wolfram.com/Levenberg-MarquardtMethod.html>. Acesso em: 18 de setembro de 2018.

Disponível em: <http://www.physics.csbsju.edu/stats/KS-test.html>. Acesso em: 26 de janeiro de 2019.

Disponível em: <https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/>. Acesso em: 31 de fevereiro de 2018.

Disponível em: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>. Acesso em: 28 de agosto de 2018.

Disponível em: <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>. Acesso em: 31 de agosto de 2018.

Disponível em: <https://towardsdatascience.com/an-overview-of-categorical-input-handling-for-neuronal-networks-c172ba552dee>. Acesso em: 25 de setembro de 2018.

Disponível em: <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>. Acesso em: 29 de dezembro de 2018.

Disponível em: <https://www.expertsystem.com/machine-learning-definition/>. Acesso em: 29 de novembro de 2018.

DOEDEN, Matt. **Legalized Gambling: Revenue Boom or Social Burst?** Twenty-First Century Books, 2010.

FISHER, Aaron, Cynthia Rudin, e Francesca Dominici. **Model Class Reliance: Variable importance measures for any machine learning model class, from the ‘Rashomon’ perspective.** 2018, disponível em: <http://arxiv.org/abs/1801.01489>

FISHER, R. **The use of multiple measurements in taxonomic problems.** *Annals of Eugenics.* 7 (2): 179–188, 1936.

GEBRU, Timnit, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, e Li Fei-Fei. **Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States.** *PNAS*, 2017.

GODDIJN, S., Evgeny Moshkovich, Rohan Challa. **A Sure Bet: Predicting Outcomes of Football Matches.** *CS230: Deep Learning*, Winter 2018, Stanford University.

HUBÁČEK, Ondrej, Gustav Sourek e Filip Zelezny. **Learning to predict soccer results from relational data with gradient boosted tree.** 108:29–47, *Mach Learn*, 2019.

IGIRI, R., C. Peace, A.Nwachukwu e E.Okechukwu. **An Improved Prediction System for a Football Match Result.** Vol.4, Issue 12, 12-20, *IOSR journal of Engineering*, 2014.

KAHN, J. **Neuronal network prediction of NFL football games.** 9–15, *World Wide Web Electronic Publication* 2003.

KAUL, Ambika, Saket Maheshwary, Vikram Pudi. **AutoLearn - Automated Feature Generation and Selection.** *Data Sciences and Analytics Center, Kohli Center on Intelligent*

Systems International Institute of Information Technology, Hyderabad, India, 2017.

KOTSIANTIS, S. B. e D. Kanellopoulos, e P. E. Pintelas. **Data Preprocessing for Supervised Learning**. Vol:1, No:12, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2007.

KUMAR, Gunjan, Prof. dr. Jesse Davis, Assessors: Prof. dr. ir. Gerda Janssens ir. Jan Van Haaren, Mentors: Daniele Alfarone ir. Jan Van Haaren. **Machine Learning for Soccer Analytics**. Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science, KU LEUVEN, 2013.

LEUNG, Carson K. e Kyle W. Joseph. **Sports data mining: predicting results for the college football games**. 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems – KES, 2014.

MASZCYK, A. e Adam Zajac e Igor Rygula. **A neuronal network model approach to athlete selection**. Springerlink.com, 2018.

MCCULLOCH, W. e Walter H. Pitts. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics, Vol. 5, p. 115-133, 1943.

NIELSEN, Michael. **Neuronal Networks and Deep Learning**. Disponível em: <http://neuronalnetworksanddeeplearning.com/>. Acessado em 02 de setembro de 2019.

PURUCKER, M. C. **Neuronal network quarterbacking**. 9–15, IEEE Potentials 15, 1996.

ROSENBLATT, F. **The Perceptron: A probabilistic model for information, storage and organization in the brain**. Psychological Review, Vol. 65, No. 6, 1958.

Sci-Kit Learn, Grid-Search. Disponível em: https://scikit-learn.org/stable/modules/grid_search.html. Acesso em: 03 de setembro de 2019.

SHEARER, C. **The CRISP-DM model: the new blueprint for data Mining**. 5, 13–

22, Journal of Data Warehousing, 2000.

SHRESTHA, Sumit. **PREMIER LEAGUE GAME RESULT PREDICTION**. DWIT College, DEERWALK INSTITUTE OF TECHNOLOGY, Tribhuvan University, 2016.

STEINBERG, Roman. **6 areas where artificial neuronal networks outperform humans**. UKIT AI, 2017. Disponível em: <https://venturebeat.com/2017/12/08/6-areas-where-artificial-neuronal-networks-outperform-humans/>

USMANI, Z. **What is Kaggle, Why I Participate, What is the Impact?** Getting Started. 2018, disponível em: <https://www.kaggle.com/getting-started/44916>.

Wei Gu Conceptualization; Data Curation; Formal Analysis , Krista Foster Roles/Writing - original draft; Writing - review & editing , Jennifer Shang Supervision; Roles/Writing - original draft; Writing - review & editing , Lirong Wei Visualization; Writing - review & editing . **A Game-Predicting Expert System Using Big Data and Machine Learning, Expert Systems With Applications**. 2019, disponível em: <https://doi.org/10.1016/j.eswa.2019.04.025>

