# ***Predicting football match result***

***1-Problem Statement: -***

5.1bn dollars, that is the number you should know of how much television broadcasting for the football rights to televise the matches and that is only for the premier league. According to BBC, sky sports have paid 4.176bn dollars to buy the rights of 126 matches from the premier league and BT have paid 960 million to have only 42 matches, that can be showed in the following graph that shows cost of premier league broadcasting rights.

Chart

Description automatically generated

Figure 1 the premier league coverage fees

What makes these companies pay that amount of money is that they know that the revenue of watching will cover these expenses, which we can conclude from the fact that world cup only in 2010 was being watched by 3.1 billion around the world according to (Writer, 2020) and according to (Richter, 2018) 3.2 billion have watched the world cup which took place in 201 which can be summarized in the following chart that shows the interest and participation in football globally.

![Table

Description automatically generated]()

Figure 2 The popularity of football around the world

This shows how much interest can be put in football only in sporting field. Alongside this a lot of people became interested in playing a mobile game called “Fantasy” which is a game that casts you in the role of a Fantasy manager of Premier League players. You must pick a squad of 15 players from the Premier League, who score points for your team based on their performances for their clubs in PL matches. It became very popular worldwide were many mini tournaments/leagues started taking place and everyone is competition to win. So we thought that implementing a ML model that is able to predict match results could be of great importance for any fantasy player and based on his performance inside the game we can evaluate the model’s accuracy.

Dataset:

Our goal is Predicting premier league match results. The data contains matches since (1993-202) .We use the past 5 years(20) dataset for predictive Analysis using deep learning models. To predict match results we used several data for prediction some of them as follows

|  |  |
| --- | --- |
| Season | Match Season |
| DateTime | Match Date and Time (yyyy-mm-dd hh:mm:ss) |
| HomeTeam | Home Team |
| AwayTeam | Away Team |
| FTHG | Full Time Home Team Goals |
| FTAG | Full Time Away Team Goals |
| FTR | Full Time Result (H=Home Win, D=Draw, A=Away Win) |
| HTHG | Half Time Home Team Goals |
| HTAG | Half Time Away Team Goals |
| HTR | Half Time Result (H=Home Win, D=Draw, A=Away Win) |
| Referee | Match Referee |
| HS | Home Team Shots |
| AS | Away Team Shots |
| HST | Home Team Shots on Target |
| AST | Away Team Shots on Target |
| HC | Home Team Corners |
| AC | Away Team Corners |
| HF | Home Team Fouls Committed |
| AF | Away Team Fouls Committed |
| HY | Home Team Yellow Cards |
| AY | Away Team Yellow Cards |
| HR | Home Team Red Cards |
| AR | Away Team Red Cards |

***2-Related work: -***

(Elsevier B.V,2022), stated that the football match result can be predicted by obtaining statistical data related to the previous games and others related to the team itself. the data was trained by four years of premier league seasons, the model predicted the games with 65.26%. For the future it is advised to integrated with a decision support system assessing the risk of bets.

(Awadalla,A) have concluded that LSTM was the most successful algorithm, and to avoid overfitting dropout was used. It is also advised to enlarge the data across different leagued that would help improving prediction.

***3-Model Architecture: -***

* Developed regression model using neural network models to predict class numeric value
* The model will be trained to minimize the mean squared error (MSE) loss function using the effective Adam version of stochastic gradient descent.
* Finally, after the model is trained, we will evaluate it on the holdout test dataset and report the mean absolute error
* Finally predicting match result

2-

* The model was developed based on the training data from which we imported it from the dataset
* The model that was used it was TensorFlow from which we depend on keras library to import the sequential model that allow layers as list with input, output and hidden layers.
* We used 330 neurons and input dimensions same as features number which is 86.
* Then the next layer was 10 neurons with the input dimension from the previous neuron of 330.
* We then used the softmax which because we have 3 classes that we created by the categorical.

3-

* Multi-Layer Perceptron model was implemented using 2 approaches.
* 1st approach was using built-in functions and changing certain parameters like the number hidden layers and neurons, learning rate and random state.
* 2nd approach was using Keras and TensorFlow libraries.
* Using sequential () we started building the model layers which were 5 layers that consisted of input layer, output layer and and 3 hidden layers.
* All layers except the output layer were built using relu activation method because we don’t accept negative numbers. The output layer used the softmax activation method where elements of the output vector are in range (0, 1) and sum to 1.
* The model was compiled with binary\_crossentropy loss function as it is supposed to be used for multi-class classification model where there are two or more output labels.
* Also, Adagrad optimizer as it uses different learning rates for each parameter base on iteration.
* While fitting the model we choose that the number of iterations while fitting the training data (epochs) = 100 and batch size = 80 to achieve better results.

4-

* The nodes of the same layer are not connected, put each node in the layer is connected to every node from the next layer
* The input layer will contain 261 unit which calculate by “unit = number of output channels \* (number of input channels + 1)”
* We have 3 nodes as output that represents the three columns "we want to get the highest prediction value from them".

5-

* Developed artificial neural network classification model used our data to predict the winner and give us the accuracy of the model
* The model used 3 layers to train the data
* In this model we used Adam optimizer which one if tensor library that has is the high efficiency rate and didn’t used much space, so it considers one of the best optimizers
* Also, because we used 3 classes in our project so to calculate the loss, we should use some function that dead with more than class, but we used binary\_crossentropy that only deal with two classes to only see the changes and how this will affect in our accuracy

6

* After developing neural network with a layers, a better efficient model was done
* Gaussian naïve bayes model, it was used because we wanted to teach each feature independently with no relation with other.
* The model was constructed with layers and Huber loss function to correct the mistakes of other loss functions and optimizer as SGD to correct GD optimizer

***4-Evaluation results: -***

* predict with the whole data: -

“yhat” variable contain the probability of the three class we have, and we will use argmax to get the highest accuracy and its index so we know which team will be predicted by the model.

* The models accuracy was between 60% percent as we used Adam and categorical cross-entropy as loss function
* The lowest accuracy came as 21% with using the linear as activation function
* The highest accuracy was the Gaussian model with 72% percent.

***5-evaluation strategies outputs: -***

* predict single input: -

We can predict single input by pathing array with the number of features and fill these features so it can predict it.

* The models were built to test different probabilities as we used cross categorical entropy for multiple class
* Adam optimizer was used for more optimizing to use the momentum of gradient descent and the learning rate more better
* Binary cross entropy was used to test the choice of two classes
* Adagrad was used as optimizer in different model to optimize the learning rate
* Relu was used as the activation function to make sure no negative values appear
* Leaky Relu was used in order to prevent the dying Relu problem

***6-Survey: -***

<https://docs.google.com/forms/d/e/1FAIpQLScIPnFZO-j8FtCo7KOuvfcHqgltnN7YfCCsriMp4bdejAW90w/viewform?usp=sf_link>

Chart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generatedChart, pie chart

Description automatically generated