AI for Football (Soccer) Analysis

Extended Abstract

Cristiano Gaudino - 117434292

Supervisor - Alejandro Arbelaez

Challenges

Dataset Creation

To make any predictions a suitable dataset is first needed, there are a number of requirements that the dataset must fit to. The dataset to be used in training and testing must have any relevant statistics pertaining to the given fixture, this includes statistics for the season for both the home and away teams such as number of wins, goals conceded, etc. It is also important that the player data is made available, this will allow for statistics on the starting eleven players of each team to be used in training. The data provided by the chosen dataset must also be available for multiple seasons, the models must have enough data available for training to avoid any issues with overfitting. The difficulty here comes from the leading football data providers not making their data publicly available, as such a lower quality dataset which does not meet all the requirements must be chosen, or a dataset must be created from publicly available information.

Feature Preprocessing

Once a suitable dataset is chosen, it must then be modified to better fit the problem. This may include removing any redundant features, for example those that have a very low standard deviation, as well as engineering new features for training purposes. These engineered features can be found by manually testing different equations, or by referring to other papers on a similar topic, for example Baboota and Kaur (2018) use Home Advantage and Goal Difference to improve their model's performance.

Model Selection

The result of a given football match can be a Home Win, a Draw, or an Away Win, for this reason a suitable multi-class classifier must be chosen. Instance based classifiers, model based classifiers and neural networks each have different attributes that make them suited to different tasks, though in this case it is unclear which is best. Baboota and Kaur (2018) achieved an accuracy of 57% using a Random Forest classifier and an accuracy of 55% using a Support Vector Machine, whereas Hucaljuk and Rakipović (2011) saw success using a Neural Network. To find the appropriate classifier for this problem, multiple will have to be trained on the dataset with the parameters of each being optimised based on preliminary results.

Solutions

Dataset Creation

To overcome the issue of datasets not being publicly available, a dataset was created for the purpose of this paper. This dataset was built using publicly available data provided by the website fbref.com, this data was chosen as the base for the dataset as there is extensive squad and player statistics available for multiple seasons.

Creating this dataset first required gathering the relevant files needed, fbref.com provides a large number of csv files which contain all of the data needed. Having gone through these files, the desired features were chosen based on what was expected to be correlated with a win or a loss, such as goals scored, while features such as expected goals were omitted as these are decided by experts or other AI systems. The desired features are stored in a number of csv files based on different subheadings such as Defensive Statistics or Scoring Statistics, all of which are based on the teams in the league.

The fixtures file provides a row for each fixture within a season as well as the result of said fixture, this file serves as the base for the parser code. For a given fixture, the parser will go through each of the csv files and gather the specified statistics for the home and away team. This data can then be concatenated and the fixture row can be updated with the new data, resulting in a dataset wherein the relevant data for each fixture is present. This also means that the parser code can be rerun on any given season once the relevant files have been provided.

A similar approach is taken with player data, once the relevant features were chosen, based on what was likely to be correlated to a win or loss, the dataset could be built. For each fixture, the starting 11 for both the home and away team have to be acquired, however this data is not available in the fixtures dataset as the lineups themselves are not stored there by fbref.com. Instead they are stored on a webpage specific to the fixture, the link to which is available on the fixtures webpage wherein the fixtures csv file was acquired. The solution to this was to use a library to get the raw HTML text from the webpage, this could then be converted to a string using the BeautifulSoup library at which point the fixtures table can be parsed out. The URL to each match report page is then easily accessible for each fixture. Doing this for every fixture is very inefficient, as a result this code is initially used by the parser to create a hash table for the match report URLs, this allows for efficient lookup for each fixture.

Similar to the above method, once the link is acquired the starting 11 can then be gathered from the web page. With this list of 22 players, the player statistics can be formed. A hash table is created from the player statistics file, this is to increase lookup times rather than opening the file for every player name. With this hash table, the specified features of the 22 players can be gathered and added to the fixtures dataset

Feature Preprocessing

The created dataset initially contained a large number of features, as such it was decided to reduce the number of features to reduce any potential overfitting. Many features were found to be largely redundant and as such were removed, these were features such as a player's minutes played. During this process it was also decided to remove the per 90 minute features, for example the goals per 90 minutes feature is linearly correlated with goals scored making it largely redundant.

It was soon discovered that there was an issue with the created dataset, there was a large number of features that were not recorded in previous seasons. These features were introduced in the 2017/18 season, but were not recorded prior to this, with the training data being seasons from 2015/16 to 2018/19, this meant that 2 of the four seasons were missing features. The two options of either removing the older 2 seasons in favour of keeping the features, versus removing the features from all seasons were tested. Removing the 2 seasons initially saw correlations increase, however there was no improvement to the accuracy at the time and due to the training data being halved there was now a large amount of overfitting. In contrast dropping the features saw no reduction in accuracy, and no issues with overfitting. As a result of this the missing features were removed.

Model Selection

Due to the large number of models and the relatively short time frame, it was not feasible to extensively grid search all of the hyperparameters for each model. In some cases, such as with the AdaBoost classifier, random search was used initially as the computation time was way too high to consider grid searching. As such the models were trained and optimised to the point where they were satisfiable, at which point the best models would be extensively grid searched.

Many different classifiers were trained and tested using the dataset, models such as Decision Tree, Random Forest and Support Vector Classifier were trained due to their prevalence in papers by Baboota and Kaur (2018) and other related papers. Once it became clear that many models were simply ignoring the Draws rather than trying to predict them, AdaBoost was trained as it is strong in binary classification problems, similarly Logistic Regression's use of One versus Rest meant that it was trained as well. Overall the result was 6 viable models that were all of roughly equal accuracy, Figure 1 illustrates the results of these 6 models. It's clear that both ensembles, Random Forest and AdaBoost, still suffer from overfitting as well as SVC, while the Decision Tree classifier and K-Nearest Neighbour are underperforming. However Logistic Regression has the highest accuracy, with no issues of over/underfitting.

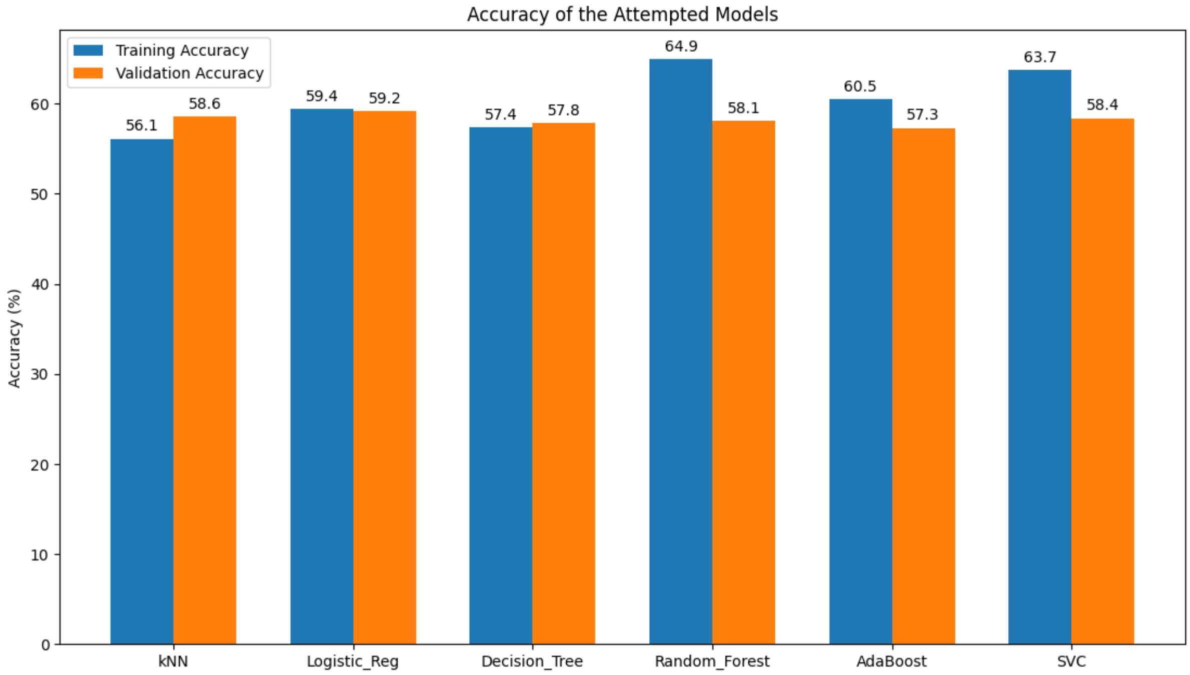

Figure 1.0

Preliminary Results

Bookmakers do not not make their prediction models, or the accuracy of them publicly available. As such to use the bookmakers accuracy as a baseline this had to be calculated from a dataset of odds provided by bookmakers for a given fixture. The odds given by Bet365, one of the largest British gambling companies, were used to calculate this baseline. The result of this when using the odds from the previous 4 seasons of Premier League football is an accuracy of 55%, whereas when using just the previous season the accuracy is 58%. In-line with the many models that were trained for this problem, the bookmaker never predicts draws and instead elects to never classify any predictions as draws.

Based on the Model Selection results, it was decided to proceed with Logistic Regression and Random Forest, both of these models were extensively grid searched over a number of days to get the best possible hyperparameters. Figure 2 illustrates the resulting confusion matrices for these models, it's clear that both still remain very close in terms of accuracy at 60%, however the Random Forest model was still overfitting slightly, whereas Logistic Regression had no issues with under/overfitting. Based on this Logistic Regression was chosen as the best model.

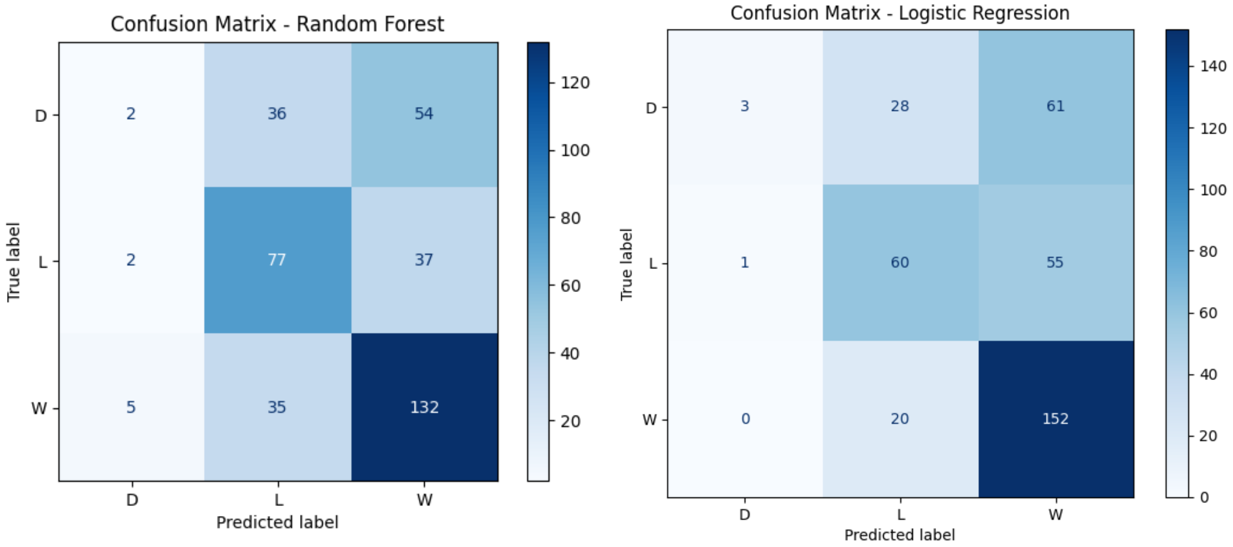

Figure 2.0

To compare the results of using the created dataset with the bookmakers, a dataset was created using the bookmakers odds. This dataset includes odds given by several top bookmakers such as Bet365, Ladbrokes and more, with three seasons being used for training/validation and one for testing. As the best model trained off the created dataset was

Logistic Regression, a Logistic Regression Classifier was also trained off the odds dataset. Depicted in Figure 3.0 is the resulting confusion matrix of the Logistic Regression model trained with the bookmaker odds dataset. With an accuracy of 58.5% and some slight overfitting it is worse than the best model, however not by much. It is also evident that draws are still disregarded by the bookmakers as illustrated by the confusion matrix.

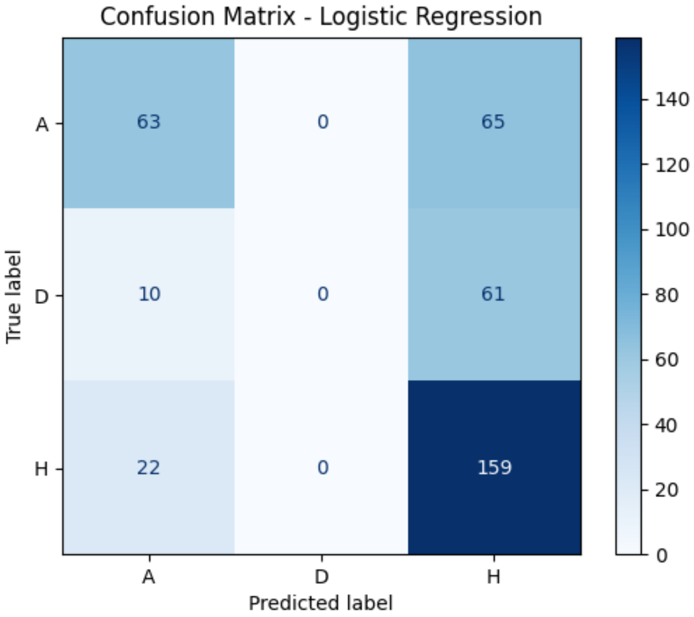

Figure 3.0

References

Baboota, R., & Kaur, H. (2018). Predictive analysis and modelling football results using machine learning approach for Premier League. *International Journal of Forecasting*

Hucaljuk, J., & Rakipović, A. (2011). Predicting football scores using machine learning techniques. *MIPRO, 2011 Proceedings of the 34th International Convention*

Football-Data. (2021). Football-Data.co.uk. European Football Results and Betting Odds [database file]. Retrieved from http://www.football-data.co.uk/englandm.php

FBRef. (2021). FBRef.com. Football/Soccer Statistics for both teams and players. Retrieved from https://fbref.com/en/comps/9/Premier-League-Stats