USING GOOGLE SEARCH TRENDS TO PREDICT ENGLISH PREMIER LEAGUE RESULTS

A PREPRINT

Group Name: GROUP A
Nanxi Zhang, Nour El Din El Sheikh, Pierre Steen, Hue Nam Yap, Amelia Baker, Gianluca Traversa
Department of Computer Science
University College London
London, WC1E 6BT

January 11, 2021

ABSTRACT

Predicting of the outcomes of sports is a popular application of machine learning techniques. In particular, the English Premier League (EPL) draws high interest and can offer significant financial rewards for accurate prediction of football games. In the EPL, due to the system of relegation and promotion of teams, it is possible to have matches between teams with very limited historical EPL match data available. This paper looks to address this issue by drawing predictions from Google Search Trends in addition to FIFA expert ratings and historical match data. Machine learning techniques have been implemented to use these datasets for the prediction of match outcomes. This paper presents an investigation of Bayesian networks, artificial neural networks, the support vector machine classifier with recursive feature elimination (RFE SVM), logistic regression, random forest and gradient boosting (xgboost). For teams without historical matches data, the method of highest accuracy was found to be RFE SVM, with 53.75% accuracy. For teams with sufficient historical matches data, the random forest method was most accurate, yielding 54.04% accuracy. The models were evaluated using k-fold cross validation (K=4) for a prediction of win, draw or lose for the match result.

Keywords Machine Learning · Football Prediction · Google Search Trends

1 Introduction

The EPL (English Premier League) is a yearly football league in England that draws worldwide attention. There is significant interest in predicting outcomes of games in the EPL in order to profit from betting on the winning team. Each year, 20 teams compete in 38 games of football (this is is known as a 'season'). Each team plays every other team twice, once at their home stadium (making them the 'home team') and once at the opponents home stadium (making them the 'away team'). In order to qualify for the competition, a team must be promoted from another lower ranked division, the EFL Championship. Teams may also be relegated each season, hence if they perform particularly badly, they will not be able to compete in the next EPL [1]. Winning a game in the EPL requires scoring more goals than the opponent by the end of the 90 minute tie. If the two teams score an equal number of points, the result is classed as a 'draw'.

The prediction of sport results is one of the many applications of data science and machine learning in modern times. Researchers combine cutting edge techniques with large data sets to achieve increasingly accurate predictions. With large financial incentives, these predictions are useful for many stakeholders, from sports betting companies to large corporate sport teams looking to gain an edge over their opponents.

Alongside historical match data, this research aims to evaluate whether Google Search Engine trends could be helpful in predicting the outcome of games in the EPL, especially in the cases where limited historical data is available. For example, it would be very difficult to predict the outcome of a match against 'Leeds United' in 2021, using historical match data alone. This is because the team has not played in EPL for 16 years[2], hence the relevant historical match

data is very limited. In cases like this, it is necessary to consider how data from other sources can be used to predict match outcomes.

Google Search Engine was first launched on 1998. Since it was founded, Google Search Engine has become the world's most used search engine, accounting for 88.41% of the global search engine market (according to October 2020 figures) [3]. There has previously been research into using Google Search Trends to predict a variety of events, such as financial markets[4], housing prices and sales[5], and oil consumption[6]. This suggests that Google Search trends can be used as a meaningful feature in prediction models across many different domains.

The methodology proposed in this report incorporates data from Google Search Engine trends along with historical match data. The importance of these data sources has been evaluated by using 3 separate data sets to train machine learning algorithms and then comparing their predictive ability. The first data set includes features generated from Google Search Trends and online FIFA scores, the second data set includes features generated from historical EPL match data and FIFA scores and the third data set includes all the features generated from Google Search Trend, historical EPL match data and online FIFA scores.

A variety of machine learning approaches have been considered for the prediction of the outcome of EPL games. Initially, the models were evaluated on their ability to make a binary prediction on football games (home team win or loss) and they used historical match data only.. Following this, the models that achieved a higher level of accuracy were trained and tested using all three data sets. In this case, the models were required to predict whether the match outcome would be a home team win, draw or loss. The accuracy of these predictions has been evaluated and compared. The algorithm and data set that yielded the most accurate predictions has been used to predict the outcomes of a set of matches in 2021.

1.1 Related works

Considering that the prediction of football match outcomes is of crucial importance to multi-billion dollar industries, a substantial amount of research has been conducted in attempt to tackle the results prediction of football match, or of some other similar sports matches.

A bivariate Poisson regression model with parameters of features obtained from team's past performance, was created by Dixon and Coles in 1997, to be used as a betting strategy with positive returns over bookmaker's odds [7].Another interesting method that has been previously explored is the Fuzzy Model approach [8]. The model used fuzzy knowledge base to identify nonlinear dependencies and tuned parameters using genetic algorithms and neural networks. The research showed that the tuning approach could improve the fuzzy logic model.

Lock and Nettleton used situational features such as current scores to build a Random Forest classifier in NFL games. The model reached a promising accuracy in match outcome prediction [9]. ELO rating, a rating approach that was originally used in ranking chess players was used in football prediction by [10].They assigned ELO ratings to the teams based on their past performance and predict match outcome based on team ELO ratings. Social media has also been used as source of training data, where collective knowledge from Twitter was used to predict the football match outcomes [11]. The Twitter-based model outperformed other models that used historical data. This research is the inspiration for using Google Search Trends to train football predicting models.

One implementation of a neural network for predicting football match outcome, by students at the University of Tartu [12], was able to achieved an impressive accuracy of over 60% in both the EPL and SLL (Spanish La Liga). Another approach at a broader level can be seen in [13] which tackles the application of NN's to sports predictions in general rather than a specific sport or league.

Bayesian networks are useful in football prediction as they can incorporate subjective expert judgement along with historical data. This approach has been shown to yield higher accuracy predictions for football matches than other approaches (MC4, naive Bayes, Bayesian learning and K-nearest neighbour) [14]. When implementing Bayesian Networks, it is possible to use the domain specific knowledge of rules of the EPL games to create the network structure. The parameters for this network can then be learned from the data. Major limitations of Bayesian networks arise when there are many nodes, as it can become difficult to create the structure using domain knowledge and computationally expensive to learn the parameters [15]. This limitation is not of considerable concern for the EPL prediction as there need not be an extremely high number of nodes used to predict the outcome of football games.

2 Data Transformation and Exploration

2.1 Data Origin

The features used in this research have been generated from the following 3 different sources of data:

- Google Search Trends
- Historical EPL data
- Online FIFA Database

The Google Search Trends data used in this research has been obtained using pytrends, an unofficial API for Google Trends [16]. In addition to EPL historical match data from 2008 to 2020, FIFA teams and players attributes have been scrapped from an online database [17]. The FIFA data represents expert opinions of the skills of players and teams as a whole.

2.2 Feature Engineering

According to previous studies, the selection of features and their ability to describe team skills has a significant influence on the resulting models's ability to produce high quality predictions [14]. For this reason, a major focus of this research is the generation of features that are optimal for building EPL predictive models.

2.2.1 Google Search Trends

For each match, the pytrends API was called to obtain the Google search trends for the home team's full team name and the away team's full team name, for one month before their match. The monthly searches for home team and away team respectively are then summed to get their total search number. Generally, for a team's match on date $y-m-d$ in the format of 'year−month−day', the Google search trend is calculated by:

$$ST_{y-m-d} = \sum_{j=y-\tilde{m}-d}^{y-m-d} ST_j, \tag{2.1}$$

where ST = Search Trend, $\tilde{m} = (m-1)\%12$.

2.2.2 Historical EPL Data

The first feature generated from the historical data is the cumulative goals difference. Goals difference has been shown to yield good performance in football prediction [18]. For each team's each match, the goals difference means goals scored by the team and minus the goals scored by the competitor. Cumulative valid goals represents the team's cumulative goal differences over previous games. The cumulative valid goals are reset to zero every 2 seasons. For a team's i_{th} match, the cumulative goal difference is defined as:

$$CGD_i = \sum_{j=1}^{i-1} GS_j - GC_j,$$

where CGD = goals difference, GS = goals scored, GC = goals conceded.

The second feature we generate is accumulated win score. For each of the team's historical matches, a win score is assigned based on the match outcome by the following rule:

$$Win \to 3, \quad Draw \to 1, \quad Loss \to 0$$

The teams win score is accumulated and then reset to zero every 2 seasons. For a team's i_{th} match, the cumulative win score is defined as:

$$CWS_i = \sum_{j=1}^{i-1} WS_j,$$

where CWS = cumulative win score, WS = win score.

The third feature is the trend – it describes the team's win score growth rate across recent matches m (this is also called the 'form'). For a team's i_{th} match, the formula to calculate trend could be written as:

$$WT_i = \sum_{j=i-1-m}^{i-1} \frac{WS_j}{m},$$

where WT = win trend. In our model, we set the parameter $m = 38$.

The sequence of results of the last three matches and winning rate in last three matches are also added as features. For each match, the home team and away team's respective last three matches results are found and the sequential results are added to the features. A winning rate is also computed from the results. For example, a team that has won two out of three matches in its last three matches will have winning rate of 0.67.

Another feature is the foul score. The foul score is generated for each team in each match based on the team's cumulative red card and yellow card amount from previous matches. Every team's foul score is reset to zero at the beginning of each season. For a team's i_{th} match, the foul score is generated using the following formula:

$$FS_i = \sum_{j=1}^{i-1} 3RC_j + 2YC_j + F_j,$$

where FS_i = foul score,

Historical corners and historical shots on target are also accumulated respectively for each team in each match. They are accumulated over a season and the value is set to zero at the start of each season. For a team's i_{th} match, the cumulative corner and cumulative historical shots on target are generated using the following formula:

$$CC_i = \sum_{j=1}^{i-1} C_j, \quad CSOT_i = \sum_{j=1}^{i-1} SOT_j$$

where CC_j is cumulative corners, C_j is corners, $CSOT_i$ is cumulative shots on target, SOT_j is shots on target.

2.2.3 Online FIFA Data

In addition to these features, team attribute and player attribute data has been scrapped from an online FIFA statistics database [17]. The team attribute includes team overall rating for each season and the player attribute includes each player's overall statistics score and age. Instead of using each player's data directly, the mean of players statistics score and the mean of players ages for each team have been computed.

2.3 Feature Selection

The final data set has 28 processed features, as shown in Table 2.1. This dataset has been separated into 3 broad feature datasets: Dataset A, with features from Google search trends and FIFA data; Dataset B with features from historical data and FIFA data, and Dataset C with features from Google search trends, FIFA data and historical data. The details of each of these datasetsm are shown in Table 2.2

In order to achieve good accuracy in models' predictions on Dataset B and C, the best performing features must be selected [19].

Several feature selection methods have been applied to the supervised machine learning models built on feature Datasets A, B and C.

For the Logistic Regression model, an embedded feature selection method, Lasso regularisation (L1 regularisation), is used to drop unimportant features while training. Lasso Regularisation only selects the features with best performance and sets other less important features to zero[20].

However, the regularization number C, which controls penalty strength, must be selected carefully. If it is too low, then the unimportant features will remain and if too high, important features will be eliminated. This has been solved by hyper-parameter tuning using grid-search, as further explained in the Model Training and Validation section.

For the SVM (Support Vector Machine) model, feature selection has been processed with RFECV (Recursive Feature Elimination with Cross-Calidation). RFE (Recursive Feature Elimination) first builds an estimator that can provide information on feature importance after fitting; in this case, the estimator is the SVM model. It then eliminates features by recursively pruning the least important feature from the original feature set based on validation score, which is evaluated by some scoring metric. For this particular model the scoring metric used is accuracy. Elimination does not stop until the best performing set of features is eventually reached [21]. To reach best result of SVM RFE, the hyper-parameter C and gamma for the SVM model has also been tuned, following the feature selection. This is considered in more depth in the sections that follow.

For the xgboost (Extreme Gradient Boosting) and Random Forest models, feature selection is automatically performed during the model fitting [22][23] .

Feature Name	Feature description
HTGPT	Home team Google Search Trend
ATGPT	Away team Google Search Trend
HVG	Home team cumulative valid goals
AVG	Away team cumulative valid goals
HWS	Home team cumulative win score
AWS	Away team cumulative win score
HWT	Home team winning trend
AWT	Away team winning trend
HR1/HR2/HR3	Home team's last 3 games results
H3WR	Home team last 3 games' winning rate
AR1/AR2/AR3	Away team's last 3 games results
A3WR	Away team last 3 games' winning rate
FHTOVA	Home team Overall Score from FIFA
FATOVA	Away team Overall Score from FIFA
FHTPSM	Mean of home team players statistics scores from FIFA
FATPSM	Mean of away team players statistics scores from FIFA
FHTPAM	Mean of home team players ages from FIFA
FATPAM	Mean of away team players ages from FIFA
HFS	Home team's cumulative foul score
AFS	Home team's cumulative foul score
HC	Home team's cumulative corners
AC	Away team's cumulative corners
HCST	Home team's cumulative shots on target
ACST	Away team's cumulative shots on target

Table 2.1: Demonstration of all features with description after feature engineering

Feature Dataset	Feature Source	Selected features
A	Google search trends, FIFA data	HTGPT, ATGPT, FHTOVA, FATOVA, FHPTSM, FAPTSM, FHTPAM, FATPAM
B	Historical EPL data, FIFA data	HVG, AVG, HWS, AWS, HWT, AWT, HR1/HR2/HR3, A3WR, H3WR, AR1/AR2/AR3, FHTPSM, FHTOVA, FATOVA, FATPSM, FHTPAM, FATPAM, HFS, AFS, HC, AC
C	Google search trends, FIFA data, Historical EPL data	Full Features in Table 2.1

Table 2.2: Demonstration of feature dataset A, B and C

The Neural Networks have been given the initial features data sets A, B and C directly, without further feature selection. This is because feature selection is implicitly performed during training, only if eliminating one or more features could improve the prediction outcome.

For the Bayesian Network approach, the features used are taken from the historical matches data only, with the full time result of each game simplified to a binary win or lose result. The Bayesian Network approach was ruled out after this initial phase due to low accuracy and slow evaluation process compared with other models, hence was not developed further for other datasets. The features used for the nodes of the network were chosen based on high correlation with the full time result of games.

3 Methodology Overview

After Data Engineering, in order to evaluate which machine learning model provides the most accurate prediction of outcomes of EPL games, several supervised machine learning models have been implemented. The outcomes of the various approaches have been compared in order to propose an optimal prediction algorithm.

The following machine learning approaches have been investigated:

- Bayesian Network
- Logistic Regression
- Support Vector Machine
- Extreme Gradient Boosting
- Random Forest
- Neural Network

The prediction of football match results is a multi-class classification problem as outcome of a game is identified as one the 3 classes: {Hometeam Win, Awayteam Win, Draw}.

In the first stage of this research, to experiment with the various available methods for prediction and to explore the historical EPL data, the multi-class problem was simplified to a binary prediction problem with only 2 classes: {Hometeam Win, Hometeam Lose}. Each of the methods mentioned were then tested on their ability to perform this binary classification task. During this phase, the models were trained with the features generated from original historical EPL dataset only. This report will discuss the performance of the Bayesian Network implemented during this exploratory phase, as it was not further developed for the multiclassification task (due to inefficient available tools). All the other models will be discussed in terms of their performance on the multiclassification problem using the datasets described in the data transformation section. The Bayesian network has been included as it could be an interesting model for future development using alternative tools.

Following the initial exploratory phase with a binary problem, multi-class classifiers were then implemented using Logistic Regression, SVM (Support Vector Machine), xgboost (Extreme Gradient Boosting), Random Forest and Neural networks. All the multi-class classifiers were trained on the three feature datasets (A, B and C), demonstrated in Table 2.2. Further feature selection was performed with Logistic Regression and SVM using L1 regularisation and RFECV respectively. Apart from neural networks, each of the model's hyperparameters were tuned using grid search or randomised search (for xgboost and Random Forest). The models have been evaluated using K-fold cross validation, where $K = 4$. Details of these models and their implementation have been further described below. The results of the model are presented in Results part.

3.1 General introduction to methods implemented

3.1.1 Bayesian Network

Bayesian networks are graphical probabilistic models that represent conditional dependencies between variables; they use directed acyclic graphs to represent the structure of the network in the form of nodes and directed edges which connect them. Each node represents a random variable in the model and the directed edges represent the conditional dependencies between them. Along with the structure of the network, Bayesian networks have associated parameters in the form of conditional probability tables (CPTs), which describe the probability of the outcomes of each node, conditional on its parent nodes.

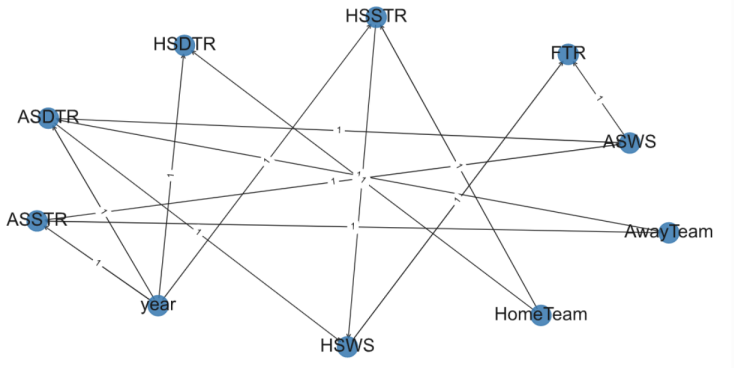

Figure 3.1: Structure of tested Bayesian Network

The structure proposed using historical EPL data consists of the following nodes: HSWS (cumulative wins of the home team for their past 38 games), ASWS (cumulative wins of the away team for their past 38 games), ASDTR (away team

average defensive ratio over the past 38 games), HSDTR (home team average defensive ratio over the past 38 games), ASSTR (away team average offensive ratio over the past 38 games) and HSSTR (home team average offensive ratio over the past 38 games) (fig 3.1). The offensive ratio is computed as the ratio of shots scored compared with shots taken for each game. The defensive ratio is the ratio of on-target shots to actual goals scored by the opposing team in a game. In this structure, it is assumed that ASWS and HSWS are conditionally dependent on the corresponding defensive and offensive ratios and the full time result of a game (FTR) is conditionally dependent on ASWS and HSWS.

After inputting this structure, bnlearn was used to learn the maximum a priori CPT for each node. Combining the network structure and learned parameters, inferences were performed on a test-set taken from the data using train_test_split from sklearn. This yielded 61.40% accuracy on a binary prediction of win or lose. Performing a large number of inferences on the Bayesian network was computationally intensive and thus a test size of 5% of the original dataset was used, with 95% used to learn the parameters of the network. The tools available for developing and implementing Bayesian networks in python are relatively limited compared with other machine learning algorithms, hence the inefficiency of this approach. Compared with other models at the binary prediction, this accuracy was relatively lower and it was decided not to develop this model further.

3.1.2 Logistics Regression

Logistic regression is one of the most common machine learning methods used to solve classification problems. An advantage of logistic regression is that the presence of weights allows us to evaluate the importance of different features in the model. At the same time, it is noted that if there is a feature that can completely separate the model, a logistic regression model would not converge and cannot be trained. Since there is no such feature in football match prediction, it remains a suitable method.

The Logistic Regression model has been implemented using scikit-learn library. The method chosen for feature selection is L1 regularisation. In this approach, hyperparameter selection is important for ensuring that the important features are better filtered. Grid-search has been used to select hyperparameter C from the following values:

$$C \in \{0.001, 0.005, 0.01, 0.09, 0.4, 5, 10, 25\}$$

3.1.3 Support Vector Machine

The SVM approach involves searching for an optimal separating hyperplane, which can serve as a decision boundary between classes of the data. The optimal separating hyperplane is that which provides the largest margin between itself and the datapoints. Support vectors from the data are used to define the discriminant function, which determines the separating hyperplane. While this approach can be relatively more time consuming, it tends to yield high accuracy predictions, is less prone to overfitting and is capable of building complex, non-linear decision boundaries.

When implementing SVM model, it is assumed that, regardless of the hyperparameters chosen, the important features for the model are similar. Therefore, the RFECV approach was first applied to eliminate unimportant features and then, with the filtered features, grid search was used to tune the hyperparameters of the model. In this way, the computation cost was minimised. Grid search was used with the following hyperparameter combination:

$$C \in \{0.3, 0.4, 0.5, 0.6\}, \quad \gamma \in \{0.05, 0.1, 0.15\}, \quad kernel \in \{\text{linear, radial basis function, polynomial}\}$$

The best combinations are: $C = 0.6, \gamma = 0.15$, kernel = radial basis function on dataset A; $C = 0.3, \gamma = 0.05$, kernel = linear on dataset B; $C = 0.6, \gamma = 0.05$, kernel = radial basis function on dataset C. The final outcomes of both SVM and SVM with RFECV are shown in the results section for comparison.

3.1.4 Extreme Gradient Boosting

Gradient boosting is a method that is based on several prediction models. It builds an ensemble model from weaker models (such as decision trees). A starting tree would be continually tuned by boosting methods, which reduces bias and variances, to generate a more powerful model. Gradient boosting is considered to be more effective on models with high bias and low variance. In general, however, gradient boosting takes more time to train. It is more prone to overfitting due to its tendency to emphasize outliers.

3.1.5 Random Forest

In contrast with gradient boosting which produces an improved model at each stage, random forest constructs a number of decision trees at once and uses bagging and feature randomness to improve them. The trees are trained with random samples from the dataset and the final output is be the mean of the predictions from the trees.

Random forest learning algorithms can efficiently handle large dataset, especially those with high dimensionality, and also reduces overfitting compared to a single decision tree through random sampling [24]. Despite this, random forest is considered a computationally inefficient learning algorithm due to the binomial nature of its structure and often yields a worse performance compared to gradient boosting. Random forest algorithms seek a function $\hat{f}$ which predicts the outcome of unseen input features $\hat{x}$ by:

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f_i(\hat{x}), \tag{3.1}$$

where f_b represents all the decision trees produced by the novel data point being averaged to give the final mapping, in a process called 'bagging'. In this use case, most likely due to reduced nature of the input features considered, we discovered that this learning algorithm was actually one of the faster performing.

The xgboost and Random Forest methods were first implemented using xgboost and scikit-learn library respectively. The hyperparameters were then tuned by applying RandomizedSearchCV from the scikit-learn implementation, where the CV part is Stratified K Fold with $K = 4$. By not using grid search, randomized search was able to save significant computation time.

3.1.6 Neural Network

Neural networks (NNs) are popular approach to machine learning when seeking complex non-linear relationship between the inputs and outputs.[25]. Various inputs are fed to the network and passed through various layers of neurons, interconnected via links consisting of activation functions and biases. Correctly predicting outcomes, given some inputs, is achieved by optimising the weights and biases associated with the artificial neurons, such that the output of the network matches the labelled outputs of the selected training inputs closely.

NNs are an example of supervised learning, meaning they require a labeled dataset to train (in this case the label is the outcome of each game). The major challenges with NNs are creating appropriate features, and choosing the optimal structure for the neural network, including layer number and size, optimization techniques and activation functions. Following this the training hyperparameters need to be tuned such that the network learns at a desirable rate while avoiding overfitting to the training data.

We made three network architectures and selected the one which performed best for each data set. The three network architectures had the following characteristics:

Input Layer Nodes:
- Dataset A: 8
- Dataset B: 26
- Dataset C: 28

Hidden Layer Nodes:
- modelNN0A: 6
- modelNN0B: 12
- modelNN0B: 13
- modelNN1_: 32, 8
- modelNN2_: 300, 128, 32, 8

Outputs: 3

Epochs: 1000

Activation functions:
- Input: Sigmoid
- Hidden: ReLU
- Output: Softmax

Given the innately sporadic-like relationship between technical analysis of football matches and the actual outcome of the game it is quite difficult to linearize the problem. The amount of data available is large enough to allow for intensive training of a non-linear model and

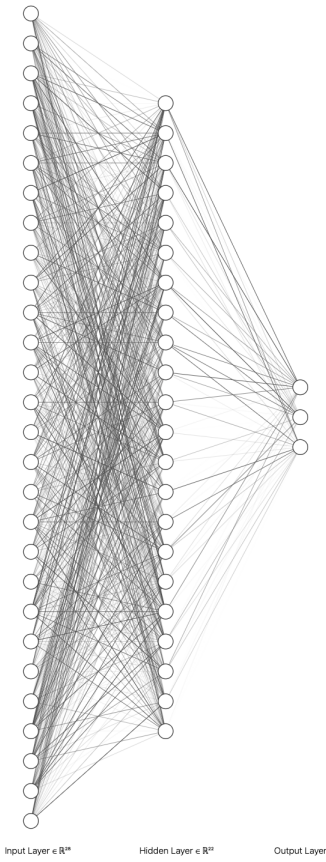

Figure 3.2: Sample structure of the ANN(28x22x3)

the available data per match also makes it possible to include a broad range of features to characterize the input (this also includes the possibility of the data being unnecessarily-granular hindering the learning). However, neural networks are well suited for this kind of predictive task and hence have been evaluated for all datasets in this research. The structure of the tested NN can be seen in Fig 3.2 and the activation functions associated with each layer are *Sigmoid, ReLU* and *Softmax* in that order. We tested models using the universal approximation theorem.

4 Final Model – Training and Validation

This section will discuss how the supervised machine learning models have been trained and validated using K-Fold cross validation method.

Initially, various models were trained on historical data only and evaluated as a binary predictor of football games (win or lose). In this evaluation the Bayesian Network proved to be extremely slow to test and returned the lowest accuracy. For this reason, the Bayesian network was rejected. Following this, all the remaining models were trained on three feature datasets respectively (see §2.3).

To evaluate these models, K-Fold Cross validation was used ($K = 4$). The various models have all been trained and validated on the same datasets. This poses a challenge since it is desirable to maximise the accuracy of the model by using as much of the dataset as possible for training, but there also needs to be a sufficiently large validation dataset.

Instead of using a traditional 75%-training, 25%-evaluation dataset split, the entire data set was divided into 'n' folds. Each of the models were then trained n-times - for each model iteration, the validation set was chosen as the corresponding nth fold and the training set as the rest of the dataset.

The performance of the model then evaluated as:

$$\text{Performance} = \frac{1}{n}\sum_{i=1}^{n}\text{Performance}_i \tag{4.1}$$

As a result, the models have been validated on the entire dataset, maximising both the accuracy and robustness in the process.

5 Results

Dataset	Algorithm	Accuracy	Precision	Recall	F1
A	Logistic Regression	0.5282	0.44	0.43	0.39
	SVM	0.5359	0.44	0.36	0.39
	RFE SVM	**0.5375**	0.36	0.45	0.39
	xgboost	0.5373	0.44	0.44	0.39
	Random Forest	0.5362	0.36	0.44	0.38
	Neural Networks	0.5338	0.47	0.53	0.46
B	Logistic Regression	0.5324	0.44	0.43	0.39
	SVM	0.5368	0.36	0.45	0.39
	RFE SVM	**0.5395**	0.36	0.45	0.39
	xgboost	0.5377	0.39	0.45	0.39
	Random Forest	0.5384	0.36	0.44	0.39
	Neural Networks	0.5369	0.49	0.54	0.47
C	Logistic Regression	0.5324	0.44	0.45	0.39
	SVM	0.5348	0.35	0.44	0.39
	RFE SVM	0.5362	0.35	0.45	0.39
	xgboost	0.5346	0.42	0.44	0.39
	Random Forest	**0.5404**	0.36	0.44	0.39
	Neural Networks	0.5327	0.48	0.53	0.46

Table 5.1: Model performance on feature dataset A, B and C

Table 5.1 presents the resulting performance of the trained various models for the three input datasets. We can see that among all models and feature datasets, the Random Forest model on feature Dataset C, which includes all of the

features generated, has the best accuracy of 54.04% in prediction. The best predictive model on feature Dataset A, which means without using features from EPL historical data, is the RFE SVM model. The RFE SVM model only used features from Google Search Trends and FIFA data, but also reached an accuracy of 53.75%, which is quite close to the accuracy of the best model (54.04%). Overall, all models trained on feature Dataset A have achieved accuracy around 53%, this suggests that, in absence of sufficient EPL historical data, Google Search Trends data and FIFA data can be used for English Premier League prediction. This is specifically important for the final prediction on test set, as the team 'Leeds United' in Test set has no recent years' EPL historical data available (having been excluded from the EPL competition for 16 years). Therefore, the RFE SVM model trained on feature dataset A will be used to predict the outcomes of matches played by Leeds United.

6 Final Predictions on Test Set

Date	Home Team	Away Team	RBF SVC	RF	Final Prediction
16/01/21	Arsenal	Newcastle	H	H	H
16/01/21	Aston Villa	Everton	A	A	A
16/01/21	Fulham	Chelsea	A	A	A
16/01/21	Leeds	Brighton	H	H	H
16/01/21	Leicester	Southampton	H	H	H
16/01/21	Liverpool	Man United	H	H	H
16/01/21	Man City	Crystal Palace	H	H	H
16/01/21	Sheffield United	Tottenham	A	A	A
16/01/21	West Ham	Burnley	H	H	H
16/01/21	Wolves	West Brom	H	H	H

Table 6.1: Demonstration of final predictions on epl-test data

For the final prediction, there is no historical data for team Leeds (Leeds United), so the prediction of matches played by team Leeds has been achieved with features Dataset A and best model 'RFE SVM'. First, feature selection has been applied to the test data features, using the selections RFE SVM model previously made on features in training set. Following this, the RFE SVM model has been used to predict on the features selected. For prediction of the other matches results, the method used is Random Forest with tuned hyperparameters on feature data set C, because this has proved to be the best performing model, if historical data is given. The final prediction is shown in Table 6.1 It is notable that here, that both models return the same predictions for the outcome of each of the test matches.

7 Conclusion

We have demonstrated that we are able to build six different learning algorithms which can all predict the outcome of a Premier League match to a higher degree of accuracy than a coin toss. Roughly 4% better than a $50/50$ does not seem like much; however, in high-level commercial sports prediction, this represents significant insight which may translate directly to capital gained. Each model has been able to extract some useful feature maps which relate our input data points X_all to match outcomes. The accuracy with which the mappings were learned varied over input datasets, training and validation methods. In the end, our most accurate predictor was found to be a Random Forest classifier with an overall accuracy of: 0.5404 when K-Fold validated over the entire training dataset. We have also discussed that certain input data points, for example Leeds United, would produce an inaccurate predictions when using the Random Forest classifier trained on dataset C since the newly promoted team lacked in head-to-head match data on which such an algorithm relies heavily to train. We saw that in this case, leveraging the Google Trends dataset to its advantage and producing an impressive accuracy of: 0.5395, the RFE SVM algorithm performed best. We therefore decide to utilise the accuracy of the Random Forest classifier to predict the outcome of matches involving teams with a large historic head-to-head dataset and to use the RFE SVM algorithm the predict the outcome of matches involving Leeds United. It is also worth noting that the Neural Networks performed well over the board, and whilst their accuracy was not quite as high as the Random Forest and RFE SVM classifiers, their precision, recall and F1 scores certainly indicate that there is room for improvement – potentially in the direction of additional feature expansion. Whilst the results of this project cannot serve as a definitive answer to the question of what is the superior model for use in football prediction, they can serve as an indication that each model serves its own purpose and when combined, they could yield a much more robust predictive algorithm that would accommodate for more novel inputs. In order to develop this research further, it would be interesting to explore how other data sources could be used to improve the accuracy of the models presented. For example, we could incorporate more detailed information on players' form and fatigue, or the financial

standing of the football teams. It could also be useful to use alternative programming languages, such as R, to develop the Bayesian network, where there are more advanced tools for efficient inference. Another interesting development would be to investigate how the predictive algorithms perform on other sports competitions. The predictive power of the models could also be evaluated in terms of profit from betting on the teams to better understand the usefulness of these techniques compared with the bookmakers.

References

[1] Premier league. `https://www.premierleague.com/`. [Onlines; Accessed: 10-01-2021].

[2] OneFootball. Leeds united back in the premier league for first time in 16 years.

[3] J. Clement. Search engine market share worldwide, Nov 2020.

[4] Min-Hsuan Fan, Mu-Yen Chen, and En-Chih Liao. A deep learning approach for financial market prediction: utilization of google trends and keywords. *Granular Computing*, 6(1):207–216, 2019.

[5] Lynn Wu and Erik Brynjolfsson. The future of prediction: How google searches foreshadow housing prices and sales, Apr 2015.

[6] Lean Yu, Yaqing Zhao, Ling Tang, and Zebin Yang. Online big data-driven oil consumption forecasting with google trends, Jan 2018.

[7] Mark J. Dixon and Stuart G. Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(2):265–280, 1997.

[8] A. P. Rotshtein, M. Posner, and A. B. Rakityanskaya. Football predictions based on a fuzzy model with genetic and neural tuning. *Cybernetics and Systems Analysis*, 41(4):619–630, 2005.

[9] Dennis Lock and Dan Nettleton. Using random forests to estimate win probability before each play of an nfl game. *Journal of Quantitative Analysis in Sports*, 10(2), 2014.

[10] Lars Magnus Hvattum and Halvard Arntzen. Using elo ratings for match result prediction in association football. *International Journal of Forecasting*, 26(3):460–470, 2010.

[11] Stylianos Kampakis and Andreas Adamides. Using twitter to predict football outcomes. 11 2014.

[12] Roland Shum. Neural networks football result prediction, Jun 2020.

[13] Rory P. Bunker and Fadi Thabtah. A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1):27–33, 2019.

[14] A. Joseph, N.E. Fenton, and M. Neil. Predicting football results using bayesian nets and other machine learning techniques, Jun 2006.

[15] K. B. Laskey and S. M. Mahoney. Network engineering for agile belief network models. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):487–498, 2000.

[16] pytrends.

[17] Khachin Borjigin. Players. `http://www.sofifa.com/`. [Online; accessed 10-01-2021].

[18] Dimitris Karlis and Ioannis Ntzoufras. Bayesian modelling of football outcomes: using the skellam's distribution for the goal difference, Sep 2008.

[19] Neda Abdelhamid, Fadi Thabtah, and Hussein Abdel-Jaber. Phishing detection: A recent intelligent machine learning comparison based on models content and features. *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2017.

[20] Seung-Jean Kim, K. Koh, M. Lustig, Stephen Boyd, and Dimitry Gorinevsky. An interior-point method for large-scale -regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[21] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1/3):389–422, Jan 2002.

[22] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016.

[23] M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, May 2004.

[24] Trevor Hastie, Jerome Friedman, and Robert Tisbshirani. *The Elements of statistical learning: data mining, inference, and prediction*. Springer, 2017.

[25] Sun-Chong Wang. Artificial neural network. *Interdisciplinary Computing in Java Programming*, page 81–100, 2003.