ARTICLE IN PRESS

International Journal of Forecasting ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Predictive analysis and modelling football results using machine learning approach for English Premier League

Rahul Baboota [a], Harleen Kaur [b,*]

[a] Guru Gobind Singh Indraprastha University, New Delhi, India
[b] Department of Computer Science and Engineering, School of Engineering Sciences and Technology, Jamia Hamdard, New Delhi, India

ARTICLE INFO

Keywords:
Machine learning
Feature engineering
Data mining
Predictive analysis
Random forest
Support vector machines (SVM)
Ranked probability score (RPS)
Gradient boosting

ABSTRACT

The introduction of artificial intelligence has given us the ability to build predictive systems with unprecedented accuracy. Machine learning is being used in virtually all areas in one way or another, due to its extreme effectiveness. One such area where predictive systems have gained a lot of popularity is the prediction of football match results. This paper demonstrates our work on the building of a generalized predictive model for predicting the results of the English Premier League. Using feature engineering and exploratory data analysis, we create a feature set for determining the most important factors for predicting the results of a football match, and consequently create a highly accurate predictive system using machine learning. We demonstrate the strong dependence of our models' performances on important features. Our best model using gradient boosting achieved a performance of 0.2156 on the ranked probability score (RPS) metric for game weeks 6 to 38 for the English Premier League aggregated over two seasons (2014–2015 and 2015–2016), whereas the betting organizations that we consider (*Bet365* and *Pinnacle Sports*) obtained an RPS value of 0.2012 for the same period. Since a lower RPS value represents a higher predictive accuracy, our model was not able to outperform the bookmaker's predictions, despite obtaining promising results.

1. Introduction

Data mining is the process of discovering patterns and extracting useful information from data for further application and analysis. The potential applications of data mining are tremendously broad, ranging from financial fraud detection (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011; Zhou & Kapoor, 2011) to the analysis of information and communication technologies (Kaur, Lechman, & Marszk, 2017; Kaur & Tao, 2014). In sports itself, predictive analysis and data mining have been applied to various sporting games such as basketball (Štrumbelj & Vračar, 2012; Vračar, Strumbelj, & Kononenko, 2016), horse-racing (Lessmanna, Sung, & Johnson, 2010) and cricket (Asif & McHale, 2016).

Football, sometimes referred to as soccer, is the most popular sport in the world. Several countries have many football teams competing in both regional and national championships. Of the numerous football championships across the world, our paper focuses specifically on the English Premier League. The English Premier League is the most watched sports league in the world, being broadcast to 643 million homes in 212 territories and with a potential TV audience of 4.7 billion people (https://en.wikipedia.org/wiki/Premier_League). Modelling a predictive system for football matches is not merely of interest in academia, but is also hugely significant in terms of economic value. A BBC article estimated the worth of the football betting market to be around $700 billion to $1 trillion (http://www.bbc.com/sport/football/24354124).

* Corresponding author.
E-mail address: harleen@jamiahamdard.ac.in (H. Kaur).

https://doi.org/10.1016/j.ijforecast.2018.01.003
0169-2070/

A major challenge encountered while modelling the results was the highly competitive nature of the English Premier League and the high incidence of upsets (where weak teams outscore strong teams). The world was awed when the football club Leicester City won the championship in 2016, against all odds, but this demonstrates the highly unpredictable nature of the game, and therefore the difficulty of the problem we are trying to tackle.

Our research focuses on the use of feature engineering to create and extract meaningful features, as well on as the application of advanced machine learning techniques, and achieves promising results. The problem of football match prediction is that it is a multi-class classification problem, with the result falling in one of three classes: {Home Win, Away Win, Draw}.

Many people in academia and industry have tackled the problem of football match prediction, owing to both its interesting nature and its economic importance. Previous research regarding this problem can be divided into two major categories: result-based studies and goal-based studies. Goal-based approaches aim to predict goals scored by each team in a particular match, whereas result-based approaches aim to predict the result class. Our work falls into the latter category.

A result-based study regarding this problem was done by Joseph, Fenton, and Neil (2006). The main focus of their study was on the creation of an expert Bayesian network for predicting the results of the team Tottenham Hotspurs for the time period 1995–1997. They demonstrated that their expert Bayesian Network outperformed other machine learning techniques such as *k*-nearest neighbors, a naive Bayesian learner and MC4 decision trees. Even though they were able to achieve an error rate of 40.79%, their predictive models focused only on one particular team, and that for a specific time period. Another use of Bayesian networks is the work done by Owramipur, Eskandarian, and Mozneb (2013). They proposed a Bayesian network for predicting the football results of the Spanish football team, FC Barcelona. Their research showcased the use of interesting features such as the weather conditions, psychological state of players, and whether or not any of the main players were injured. They reported very high accuracy results, with a mean accuracy of 92%, but they modelled their system for only one team and a single season, involving just 20 matches for the team under observation. Another Bayesian network for forecasting Association Football match outcomes was proposed by Constantinou and Fenton (2013) based on their previous study (Constantinou, Fenton, & Neil, 2012). They used their model to predict every match of the 2011–2012 EPL season and published their predictions online prior to each match. They demonstrated that they were able to generate profitable returns by using a less complex model than their previous study.

Dixon and Coles (1997) employed a bivariate Poisson distribution for the number of goals scored by each team, parameterized by features relating to the team's past performance. However, they focused on using the developed model as the basis for developing a betting strategy that would yield positive expected returns over the bookmaker's odds. Goddard (2005) used an ordered probit regression model for predicting the outcome of a football match. Their study involved the inclusion of various interesting predictors such as the geographical distance between the team's home town and the average attendance at a match, and they were able to show that these explanatory variables were significant for predicting matches. However, their study focused extensively on the analysis of economic gains and the price efficiency of the fixed odds betting market, rather than on match prediction. Karlis and Ntzoufras (2010) compared a weighted likelihood approach with the conventional maximum likelihood approach, and showed that their proposed method offers an efficient protection against outliers. Boshnakov, Kharrat, and McHale (2017) introduced a model which uses a count process based on Weibull inter-arrival times and a copula to produce a bivariate distribution for the numbers of goals scored by the home and away teams in a match.

Another attempt at predicting football outcomes was undertaken by Rue and Salvesen (2000). They took an applied statistics approach where they created a Bayesian dynamic generalized linear model. They also exploited the power of Markov chain Monte Carlo techniques for inference, as well as performing a retrospective analysis of a season. They extended their model to various applications, such as the analysis of betting profits, the prediction of final teams rankings and the detection of outlier matches. However, they created their model only for the season from 1997–1998, meaning that it is unlikely to perform well for long-term generalization. Crowder, Dixon, Ledford, and Robinson (2002) built upon the work of Dixon and Coles (1997) by replacing the original stochastic process model with an approximation that yielded a more tractable computation without comprising the predictive power too much. Koopman and Lit (2015) developed a statistical model using a bivariate Poisson distribution. Their non-Gaussian space state model used intensity coefficients for the competing teams that depended on the attack and defense strengths of each team, which change stochastically over time. They further demonstrated that their modelling system could produce significantly better returns than the bookmaker odds.

Rotshtein, Posner, and Rakityanskaya (2005) created a fuzzy model for football prediction where the parameters were tuned using a combination of genetic algorithms and neural networks. They applied this model to the tournament data for the championship of Finland and showed that these tuning techniques for model parameter selection improved the results of their fuzzy logic model.

An interesting approach to this problem was taken by Godin, Zuallaert, Vandersmissen, De Neve, and Van de Walle (2014) and Schumaker, Jarmoszko, and Labedz (2016), who made use of collective knowledge from social media platforms, especially Twitter. Schumaker et al. (2016) used the sentiment in the tweets that they collected for predicting the results of the match, as well as for making decisions regarding wagering. They showed that crowdsourcing yielded better results than using domain experts, especially in wagering decisions. Godin et al. (2014) tackled the problem using a hybrid of collective knowledge and traditional statistical learning techniques. However, their work focused more on improving the bookmaker predictive odds in order to realize a greater monetary profit.

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

ELO ratings were originally proposed by Elo (1978) for ranking international chess players, but were later adapted to the problem of football match prediction by Hvattum and Arntzen (2010), who used economical and statistical measures to compare the merits of the ELO ranking system applied to football match prediction with those of a set of six benchmark prediction methods. However, Constantinou, Fenton, and Neil (2013) subsequently proposed another rating technique, called the pi-rating, that has been shown to outperform the ELO adaptation proposed by Hvattum and Arntzen (2010).

The remainder of the paper is organized as follows. Section 2 discusses the engineering of the relevant features in order to perform the predictive analysis. Section 3 explains the different feature selection techniques that are employed for finding the best performing features. Section 4 describes the machine learning techniques that are used for performing the predictive analysis. Section 5 contains a thorough analysis of the results obtained by the machine learning algorithms. Finally, we conclude by comparing our results with existing baseline and state-of-the-art metrics and describing other avenues for future work.

2. Feature engineering

We obtained our data from a public United Kingdom based source, *Football UK* (http://www.bbc.com/sport/football/24354124). We used data from 2005 to 2016, spanning 11 seasons. Although we also had access to data from much earlier seasons, we chose not to use them due to the limited match statistics available in these earlier seasons. For the *Rating* statistics, we scraped the data from an online database (https://www.fifaindex.com).

Previous research (Joseph et al., 2006; Owramipur et al., 2013) has shown that the quality of the results in this prediction task is associated directly with the quality of the feature set used for modelling the system. Both our understanding of the problem domain and our predictive analysis suggest that the selection of the correct set of features is of paramount importance. Thus, one of the most crucial aspects of our research was the engineering of features that would be likely to help in predicting the outcomes of a given football match. This paper explains the ideas behind the features engineered thoroughly, along with their mathematical formulations.

Both prior research and our own intuition suggested that the home/away factor (whether a team is the home team or the away team) is an important characteristic of the problem, and, despite being deceptively simple, this factor did turn out to be of great importance. Thus, we treat the home/away factor as a global characteristic and compute the feature values for both the home and away teams for every feature that we engineer.

The features engineered below are computed across each season independently, and no feature values/statistics are inherited from the previous seasons. Thus, the values of the engineered features are computed at a season-wise level, as opposed to an entire database level.

Some of the models used previously have incorporated features such as the relative strengths of the attack and defense for the home and away teams. Influenced by these, we have also incorporated the *Attack*, *Midfield*, *Defense* and *Overall* ratings for the respective teams. We collected the data for the *Rating* statistics from an online database (https://www.fifaindex.com) that has a collection of individual team ratings (*Attack*, *Midfield*, *Defense*, *Overall*) for each season that are generated by the football video game series FIFA, which is released annually by Electronic Arts. The database also included the season-wise (static throughout a season) *Rating* statistics for each team, which account for variation in a team's strength across seasons. The ratings used are the ones that are determined by the algorithm used by EA Sports for their video game series FIFA. We scraped the *Ratings* (*Attack*, *Midfield*, *Defense*, *Overall*) statistics from the database directly, rather than using any kind of team player rating aggregation of our own to compute the different *Rating* statistics for each team. However, the introduction of these features raised the issue of their non-Gaussian distribution over the dataset, which dampened the results of both probabilistic and linear models. We toyed with various different approaches for dealing with this problem, and ultimately found that the optimal solution was to consider the differential of the *Rating* statistics.

For $R_i \in \{Attack, Midfield, Defense, Overall\}$:

$$R_i = R_i^H - R_i^A,$$

where $R^H = Home\ Team\ Rating$ and $R^A = Away\ Team\ Rating$.

The next feature that we have considered is the *Goal Difference*. The *Goal Difference* is of pivotal importance when building predictive models for football. The pi-rating system introduced by Constantinou and Fenton (2013) provides empirical proof that the *Goal Difference* works well as a feature for forecasting football match results. We used the traditional unweighted *Goal Difference*, which is a running sum of the numbers of goals scored differenced by the running sum of the numbers of goals conceded by each team before coming into a new match.

The *Goal Difference* (*GD*) for a team's kth match is defined as:

$$GD_k = \sum_{j=1}^{k-1} GS_j - \sum_{j=1}^{k-1} GC_j,$$

where $GS = Goals\ Scored$ and $GC = Goals\ Conceded$.

The dataset available to us was relatively rich in terms of match statistics, and we took advantage of this to engineer more useful features. After a careful analysis of the match statistics available, we decided that the most important metrics for assessing the performance of a team would be *Corners*, *Shots on Target* and *Goals*.

We chose *Corners* because a higher number of *Corners* indicates that a team is playing well by enforcing corner kicks from the opponent team. *Shots on Target* is an even better performance metric, as it displays the superiority of teams that have large numbers of *Shots on Target*. The reason for choosing *Goals* is self-evident, as it is what determines the final result of a match. These metrics provide us with mathematical insights regarding team performance, and the pressure exerted by each team on the other. When engineering features from the above performance metrics, we decided that it would be logical for the features to

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

reflect the teams' recent performances, so as to capture any increasing/decreasing trends in the performances; thus, we took their average values over the past k games, where k is a hyper-parameter.

For $\mu^i \in \{Corners, Shots\ on\ Target, Goals\}$, μ^i for a team's jth match is defined as:

$$\mu^i_j = (\sum_{p=j-k}^{j-1} \mu^i_p)/k.$$

When designing our model's features for the prediction of football results, we wanted to introduce a feature that could provide us with useful quantifiable insights for judging the recent performances of a team. Thus, we engineered a feature that we refer to as the *Streak*. This feature encapsulates the recent improving/declining trend in the performance of a team. The *Streak* value for a team is computed by assigning a score to each match result and taking the mean of the previous k scores, where k is the aforementioned hyper-parameter. We also included a temporal dimension to the *Streak* feature by placing time-dependent weights on the scores of the previous games of a team, obtaining a feature that we refer to as the *Weighted Streak* (greater weights for recent games, decreasing gradually for non-recent games). In the *Weighted Streak* feature, the weighting scheme is as follows:

- a weight of 1 on the oldest observation $(j-k)$
- a weight of k on the most recent observation $(j-1)$.

Mathematically, *Streak* (δ) for a team's jth match is defined as:

$$\delta_j = (\sum_{p=j-k}^{j-1} res_p)/3k.$$

Similarly, *Weighted Streak* (ω) for a team's jth match is defined as:

$$\omega_j = \sum_{p=j-k}^{j-1} 2\frac{p-(j-k-1)res_p}{3k(k+1)},$$

where $res_p \in \{0, 1, 3\}$ for *Loss*, *Draw* and *Win*, respectively.

The *Streak* and *Weighted Streak* features are normalized such that their values range between zero and one. The $3k$ factor in the formulation of *Streak* (δ) ensures that the maximum *Streak* value that can be attained by a team, which happens when a team has won its previous k matches, is 1. For the *Weighted Streak* (ω), the weights are assigned by the factor $(p-(j-k-1))$, which assigns a weight of 1 to the oldest match in our observation window $(j-k)$ and a weight of k to the most recent observation $(j-1)$. The normalization is done according to the weighting scheme defined above. In this case, the normalization factor is $3(k\,(k+1)\,/2)$, where the factor $(k\,(k+1)\,/2)$ is the sum of the k weights and the multiplication by three arises from the fact that the maximum value of res_p is three. This normalization factor in the computation of *Weighted Streak* (ω) again ensures that the maximum attainable value is one.

Even though the *Streak* and *Weighted Streak* and the set of μ^i (*Corners*, *Shots on Target*, *Goals*) features performed quite well, these features still present the issue of missing values for the first k weeks in a season. We tried to overcome this issue by filling in the initial missing values through setting the value of k equal to $(j-1)$ for the jth week (except for the first week) up to k weeks (i.e., $k=1$ for the 2nd week, $k=2$ for the 3rd week, and so on, up to the kth week). However, we did not end up adopting this approach because of some underlying problems with it. Firstly, this approach breaks the notion of using a uniform value of k across the season, as k is changed dynamically for game weeks 2 to k, then retains a constant value for the remainder of the season. Secondly, this method still doesn't provide feature values for a season's first game week. Thus, we dropped the first k game weeks of each season from our database, due to both the non-rigorous nature of the scaling approach mentioned above, and the fact that dropping the games for the first k weeks gave us better performance results than using the scaling procedure. Therefore, one shortcoming of our predictive system is that it is unable to predict the outcomes for a team's initial k matches in a season. We are currently researching better techniques for scaling these features to the start of a season in order to overcome this limitation of our predictive models.

We wanted to include another temporal dimension in our problem hyperspace as well, one which would encompass a team's performances in individual matches much more intricately. As a result, we introduced an additional temporal feature, *Form*. A team's *Form* provides great detail about its recent performances and stature. The *Form* coefficient values are updated after each match based on the final result, which shows how strongly it captures the notion of time dependency. Both the *Form* and *Streak* features are team factors; however, unlike the *Streak* feature, which serves as a measure of how a team is performing irrespective of the opponents that they have faced, the *Form* feature is able to incorporate a team's performance relative to their opponents. Our mathematical formulation of *Form* ensures that a greater coefficient update is provided if a weak team triumphs over a strong team, and vice-versa. In the case of a draw, the *Form* of a weak team increases while that of a strong team decreases.

The *Form* value of each team is initialized to one at the beginning of each season and then updated after each match according to the result of the match, such that a higher value indicates a better *Form*.

Suppose that a team α beats another team β; then, the *Form* (ξ) value for each team's jth match is updated as:

$$\xi^\alpha_j = \xi^\alpha_{(j-1)} + \gamma\xi^\beta_{(j-1)}$$
$$\xi^\beta_j = \xi^\beta_{(j-1)} - \gamma\xi^\beta_{(j-1)}.$$

In case of a *Draw* outcome, the *Form* (ξ) values for each team's jth match are updated as:

$$\xi^\alpha_j = \xi^\alpha_{(j-1)} - \gamma(\xi^\alpha_{(j-1)} - \xi^\beta_{(j-1)})$$
$$\xi^\beta_j = \xi^\beta_{(j-1)} - \gamma\left(\xi^\beta_{(j-1)} - \xi^\alpha_{(j-1)}\right),$$

where γ is the stealing fraction $\in (0, 1)$.

When one team α beats another team β, the *Form* updates can be viewed as the winning team (α) stealing a certain fraction (γ) of the form of the losing team (β). Since

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

Table 1
Numerical demonstration of the computation of *Form* updates.

Outcome	$\xi^{\alpha}_{(j-1)}$	$\xi^{\beta}_{(j-1)}$	Update to α	Update to β	ξ^{α}_{j}	ξ^{β}_{j}
α wins	1.4	0.8	+0.264	−0.264	1.664	0.536
β wins	1.4	0.8	−0.462	+0.462	0.938	1.262
Draw	1.4	0.8	−0.198	+0.198	1.202	0.998

Table 2
Feature description.

Feature name	Feature abbreviation	Class category
Home form	*HForm*	*Class A*
Away form	*AForm*	*Class A*
Home streak	*HSt*	*Class A*
Away streak	*ASt*	*Class A*
Past *k* home shots on target	*HSTKPP*	*Class A*
Past *k* away shots on target	*ASTKPP*	*Class A*
Past *k* home goals	*HGKPP*	*Class A*
Past *k* away goals	*AGKPP*	*Class A*
Past *k* home corners	*HCKPP*	*Class A*
Past *k* away corners	*ACKPP*	*Class A*
Home attack rating	*HAttack*	*Class A*
Away attack rating	*AAttack*	*Class A*
Home midfield rating	*HMidField*	*Class A*
Away midfield rating	*AMidField*	*Class A*
Home defence rating	*HDefence*	*Class A*
Away defence rating	*ADefense*	*Class A*
Home overall rating	*HOverall*	*Class A*
Away overall rating	*AOverall*	*Class A*
Home goal difference	*HTGD*	*Class A*
Away goal difference	*ATGD*	*Class A*
Home weighted streak	*HStWeighted*	*Class A*
Away weighted streak	*AStWeighted*	*Class A*
Form differential	*FormDifferential*	*Class B*
Streak differential	*StDifferential*	*Class B*
Past *k* shots on target differential	*STKPP*	*Class B*
Past *k* goals differential	*GKPP*	*Class B*
Past *k* corners differential	*CKPP*	*Class B*
Attack rating differential	*RelAttack*	*Class B*
Midfield rating differential	*RelMidField*	*Class B*
Defence rating differential	*RelDefense*	*Class B*
Overall rating differential	*RelOverall*	*Class B*
Goal difference differential	*GDDifferential*	*Class B*
Weighted streak differential	*StWeightedDifferential*	*Class B*

this fraction is directly proportional to the *Form* magnitude of a team, larger updates are given for beating a stronger team (high *Form* value) than for beating a weaker team (low *Form* value). In the case of a draw, a positive update is given to the weaker team and a negative update to the stronger team. For the *Draw* outcome, the update in *Form* is directly proportional to the difference between the two teams' *Form* values at the start of the match. Thus, the size of the update will be small when teams with similar *Form* values tie, and vice-versa. After cross-validating for different values of γ, we found that the optimal solution was to set $\gamma = 0.33$. The example in Table 1 demonstrates how updates are given to the teams after each match.

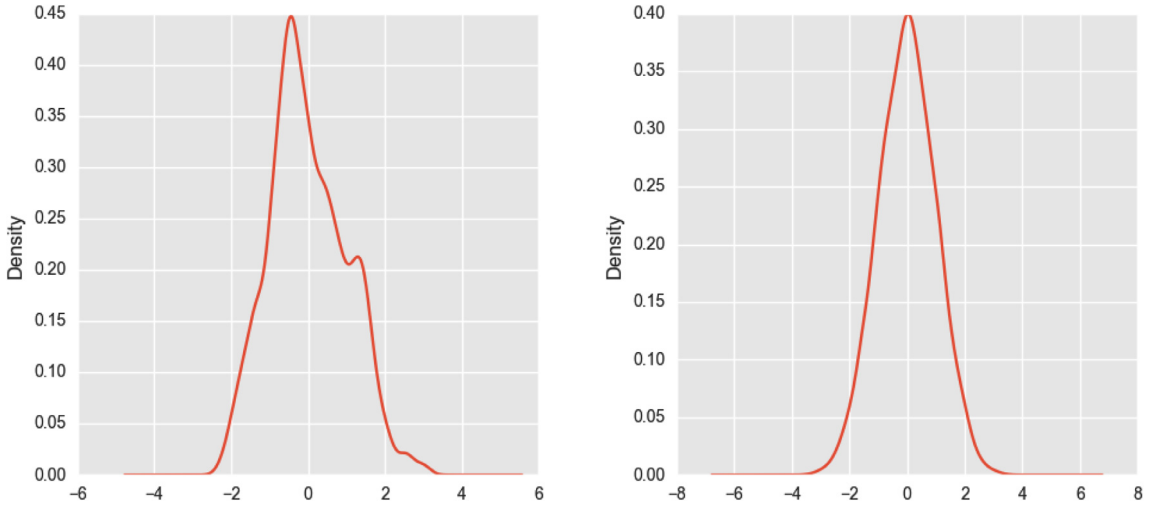
Let us consider the three different cases which can arise:

- α wins: The statistics provided in Table 1 show that Team α is the stronger team coming into the match, as it has a higher *Form* value. Therefore, this outcome is more likely to occur, and as a result, Teams α and β receive updates of $+0.264$ and -0.264 respectively.
- β wins: Since Team β is the weaker team coming into the match, the outcome where the weaker team (β) triumphs over the stronger team (α) is an unlikely one. As a result, Teams α and β receive updates of -0.462 and $+0.462$ (greater than the previous outcome).
- draw: Even though a *Draw* outcome gives equal points to each team, this outcome is much more beneficial to the weaker team (as it had a relatively higher probability of losing), which is captured by the *Form* update, with the form value of the weaker team (β) being increased by a value of 0.198 and that of the stronger team (α) being decreased by the same factor.

3. Feature selection

Feature selection is the technique of selecting only the most relevant and best performing features, in order to reduce the total number of features (Chauhan & Kaur, 2014; Kaur, Chauhan, & Ahmed, 2012; Kaur, Chauhan, & Alam, 2010, 2011). After performing extensive feature engineering, we had a total of 33 features (Table 2). Since having such a large number of features increases our feature space

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.



(a) Home team attack ratings.

(b) Differential attack ratings.

Fig. 1. Univariate distribution comparison for *Class A* and *Class B* features.

dimension greatly, we aimed to select the best performing and most relevant features by performing feature selection. We first divided our features into two broad classes, *Class A* and *Class B*, with *Class A* containing the individual features for both the Home and Away teams and *Class B* containing the *Differential* features. The rationale behind the *Class B* features was twofold. Firstly, the *Differential* features had a much better univariate distribution (Figs. 1(a)–1(b)), which is quite useful for some statistical learning techniques. Secondly, by encoding the information held by two variables in one, we were able to reduce the dimensions of the problem hyperspace, which makes our models less likely to get stuck in a local minimum.

We tested our feature set with the Gaussian naive Bayes, support vector machine, random forest and gradient boosting models by fine tuning each model for optimal performance through feature selection and hyper-parameter tuning.

We tested the Gaussian naive Bayes model with both *Class A* and *Class B* features, and found it to perform much better on the latter set. This was as expected, as the *Class B* features were a better fit to the Gaussian probability density function and there were few features that had high correlations amongst them. We observed some high correlation values among the rating statistics, and thus achieved a performance boost by dropping these predictors.

For the SVM models, we adopted a more formal feature selection technique, namely recursive feature elimination (RFE) (Guyon, Weston, Barnhill, & Vapnik, 2002). RFE builds multiple models by eliminating features sequentially from our entire pool, eventually returning the most relevant features. We found this technique to be of great assistance for determining the best performing features for the models above, thus helping us to fine tune them better.

For the random forest and gradient boosting models, we did not apply any kind of external feature selection, as these algorithms are designed to perform feature selection on their own at fitting time, and do not suffer from having highly correlated features in the feature set.

4. Predictive models

A lot of previously-applied methods have involved the modelling of predictive systems by reducing the ternary classification problem to a binary classification one, wherein a prediction is made for each team as to whether it will win the match or not. The results reported for this approach are quite commendable; however, it does not generalize well for solving the true problem of prediction, and thus has little significance in real world applications. Our ideology in regard to this problem was to create a well-generalized system that could be used for long term effective predictive modeling, which is why we tackled the original multi-class ternary classification problem over a period of 11 years. To do this, we tried out various machine learning models, the details of which are explained below.

4.1. Gaussian naive Bayes

The first predictive model that we employed on our feature set was the multivariate Gaussian naive Bayes model. Naive Bayes is a deceptively simple yet a very efficient algorithm that is used for performing classification. It is a purely probabilistic method which uses the principles of conditional probability to compute the posterior probabilities of a data instance belonging to a particular class.

For some value ν of our feature vector X, the probability distribution of ν given a class c is:

$$P\left(X = \nu|c\right) = \frac{1}{\sqrt{2\pi\sigma_c^2}} exp \frac{-(\upsilon - \mu_c)^2}{2\sigma_c^2},$$

where σ_c is the standard deviation of the distribution and μ_c is the mean of the distribution.

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

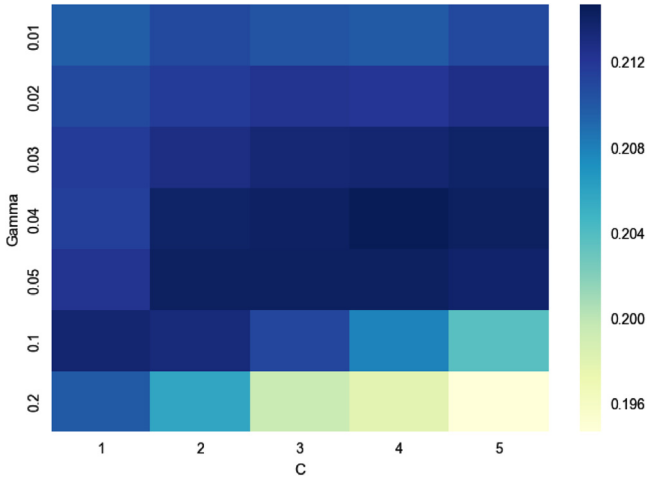


Fig. 2. RBF SVM hyper-parameter optimization for *C* and *Gamma*.

The reason why the naive Bayes is called naive is its strong independence assumption (all features are assumed to be independent of one another). Also, when dealing with continuous data, the naive Bayes assumes that the data associated with each class is distributed according to a Gaussian probability density function. Due to these inherent model assumptions, we did not expect our Gaussian naive Bayes classifier to provide us with great results; however, we were able to obtain quite reasonable results after fine-tuning our classifier to account for the above assumptions.

4.2. Support vector machine

The second predictive model that we used was support vector machines (SVMs). SVMs are a set of supervised learning algorithms that are extremely effective when dealing with high-dimensional feature spaces. They use the kernel trick for efficient performance in non-linear classification cases. Kernels are basically functions that take a low-dimensional input and map it to a corresponding high-dimensional feature space in order to find the most suitable decision boundaries. We made use of two kernels for our predictive analysis, namely the linear kernel and the radial basis kernel (RBF). Though SVMs are originally non-probabilistic classifiers, we computed the probability estimates using five-fold cross-validation.

The main hyper-parameters for our RBF SVM model were *C* and *Gamma*. Since we only had to optimize these two parameters, we chose to use a grid search for the hyper-parameter optimization. The *C* parameter represents the cost of misclassification on the training data. A low *C* makes the decision surface smooth, while a high value aims to classify all training examples correctly by giving the model freedom to select more samples as support vectors. Thus, it is crucial to find the most suitable value for *C* in order to prevent it from either under- or over-fitting. *Gamma* is the free parameter of the Gaussian radial basis function. If *Gamma* is large, the variance is small, implying that the support vector does not have widespread influence. A large *Gamma* leads to models with a high bias and low variance. The results of our hyper-parameter optimization for *C* and *Gamma* are shown in Fig. 2.

4.3. Random forest

The next predictive model was the decision tree ensemble technique, random forest (Breiman, 2001). Random forest is a robust machine learning algorithm which is capable of mapping complex nonlinear decision boundaries. It overcomes the high variance problem caused by decision trees through building a large number of trees by bootstrapping samples, and then taking the majority vote results in order to improve accuracy and limit overfitting.

When building classification trees, the cost criterion for determining a split is generally either the *Gini Index* or *Cross-Entropy*. After tuning the hyper-parameters, we found that the best results were obtained using the *Gini Index* (G). The *Gini Index* measures the total variance across all of the classes, and is therefore a measure of node purity.

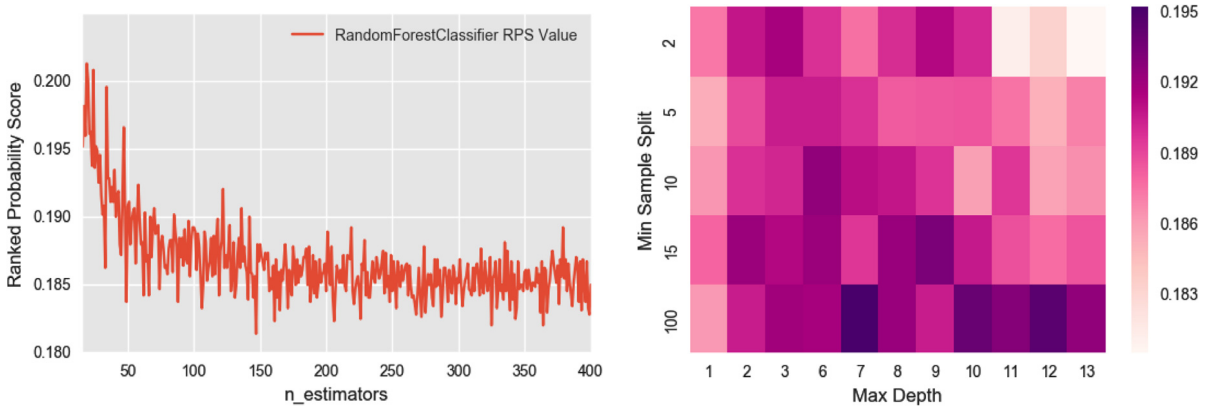
$$G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}),$$

where p_{mk} is the proportion of training observations in the mth region that are from the kth class.

Random forest is a special-case bagging technique which builds uncorrelated trees so as to achieve greater reductions in the variance. It does this by considering only a subset of the available predictors each time it makes a split. Using a grid search, we found that the model performed best when it considered the *logarithm* of the total features available when making the split.

We tried two different approaches for tuning the hyper-parameters of the random forest model, namely a random search and a grid search. Since we had a lot of hyper-parameters to tune, we started out by trying a random search; however, we found that it skipped the global minima. We therefore used a grid search for our hyper-parameter optimization, and found it to yield the best set of

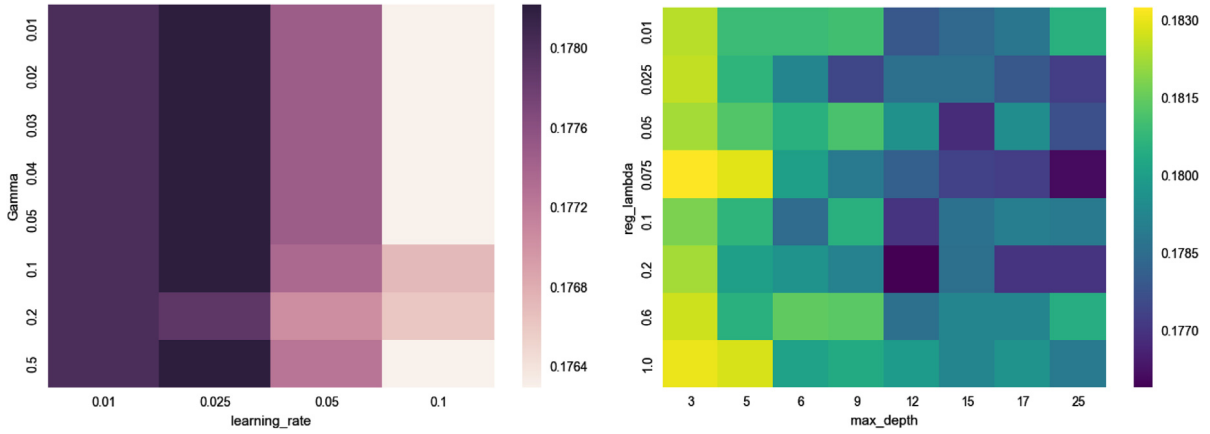
Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.



(a) Tuning for *number of trees*.

(b) Tuning for *max depth* and *min sample split*.

Fig. 3. Random forest hyper-parameter optimization.



(a) Tuning for *learning rate* and *gamma*.

(b) Tuning for *max depth* and *lambda*.

Fig. 4. Gradient boosting hyper-parameter optimization.

values for our hyper-parameters, even though it was computationally more expensive. Some of the most important parameters that needed to be optimized for our random forest model were the *number of trees* to build, the *splitting criterion* to consider, the *maximum depth* of each tree and the *minimum sample split*. The results of the grid search hyper-parameter optimization for the random forest are displayed in Figs. 3(a) and 3(b).

4.4. Gradient boosting

Our final predictive model was (extreme) gradient boosting. Like random forests, gradient boosting is a decision tree ensemble technique; however, gradient boosting works on the principle of *boosting*, whereas random forests work on the principle of *bagging*. Both techniques involve combining a group of classifiers; however, in *bagging* each classifier is trained in parallel with a random subset of the data, whereas in *boosting* the classifiers are trained sequentially on the entire dataset, with the model accuracy being improved after each iteration.

We used the open source framework XgBoost (Chen & Guestrin, 2016) for implementing gradient boosting on our set of features. XgBoost works on the general principle of gradient boosting but uses a more regularized model formalization to control over-fitting, meaning that it performs better.

The number of hyper-parameters that required tuning in order to optimize our gradient boosting model were greater than that for the random forest model, and hence, it took a considerable amount of time to identify the optimal values of the hyper-parameters. We again chose to use a grid search to find the optimal hyper-parameter values. Since we had to tune a large number of parameters, we only display the results of a few of the important parameters, including *gamma*, which specifies the minimum loss reduction required to make a split, the *learning rate*, the *maximum depth* of each tree, and *lambda*, which controls the regularization strength. The results of the grid search hyper-parameter optimization for gradient boosting are displayed in Figs. 4(a) and 4(b).

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

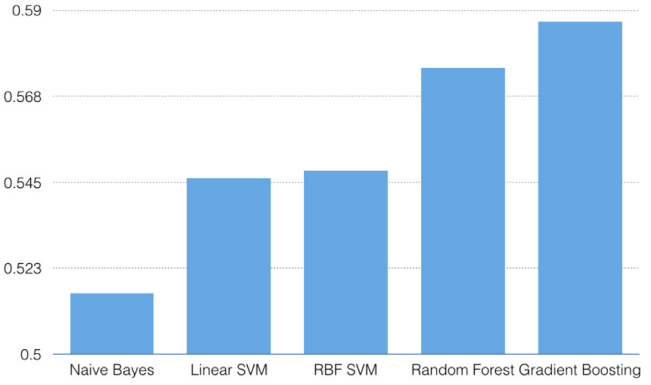


Fig. 5. Mean test accuracy scores of the different machine learning models.

5. Results and discussion

For experimental purposes, we divided our dataset into nine seasons of training data, from 2005 to 2014, and kept the remaining two seasons, from 2014 to 2016, as test data. Unlike some prior research (Odachowski & Grekow, 2013), we did not break our problem down to a binary classification problem, but used the original three-class multi-classification.

We used the programming language Python[1] for our research. We made use of the *Pandas* (McKinney, 2010) package for our data pre-processing and the *Scikit-Learn* (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, et al., 2011) package for the implementation of our machine learning models.

We applied various different machine learning algorithms to our crafted feature set, and the testing score accuracy results that we obtained for each model are shown in Fig. 5.

We found that our best performing model was the gradient boosting model, followed by the random forest, then the two support vector machine models with an RBF and a linear kernel respectively. The model with the lowest accuracy was the Gaussian naive Bayes, though it really did surprisingly well, given the model's inherent mathematical assumptions.

As was stated earlier, one of the factors that makes the prediction of football results tough is the high incidence of draws (25% of games in our test set), which is the least likely outcome of a match. After performing a thorough analysis of our models, we found that each model under-predicted draws, and some drastically, as is explained below.

The first step when employing our models, and one of the most important, was to determine the value of the hyper-parameter k for our features such as *Corners*, *Shots on Target, Goals*, *Streak* and *Weighted Streak*. We chose to use a universal value of k for all of these features, as we thought that this would ensure uniformity across the entire dataset. We employed 10-fold cross-validation for determining the optimal value of k, the results of which are shown in Fig. 6. We observed that the best value of k for our gradient boosting, random forest and naive Bayes models was six. Thus, even though the SVM model performed best for $k = 3$, we chose $k = 6$ across all of our models so as to ensure uniformity.

Table 3
Gaussian naïve Bayes results.

(a) Confusion matrix

	Predicted win	Predicted loss	Predicted draw
Actual win	197	46	47
Actual loss	57	96	36
Actual draw	72	45	44

(b) Precision–recall table

	Precision	Recall	F1-score
Home wins	0.62	0.66	0.63
Away wins	0.51	0.51	0.51
Draws	0.33	0.25	0.29

The confusion matrix, precision and recall results obtained from our Gaussian naive Bayes model are shown in Table 3(a) and (b).

We achieved quite reasonable results from our naive Bayes model, which gave us a best performance score of 0.519 mean accuracy after fine-tuning. As expected, the model performed quite well on home wins, with precision and recall scores of 0.62 and 0.66 respectively. Surprisingly enough, even though the model under-predicted draws, it yielded a recall of 0.25 (second-best among all models), which suggests that our probabilistic algorithm models the occurrence of draws quite nicely, and has a lower yield of false negatives in the case of draws.

We were able to obtain a lot of insights about the distribution of our data points in our feature hyperspace by analyzing the results of our linear SVM in Table 4(a) and (b).

An analysis of the linear SVM results showed that it could not model the occurrence of draws at all. This helped us to conclude that the data could not be separated by a linear hyperplane. Thus, even though our linear SVM performed better than the naive Bayes on the mean accuracy metric, we did not pursue it (or any other linear models) further, due to its poor performance in modelling draws.

[1] Python Software Foundation, see *Python language reference, version 2.7* (available at http://www.python.org).

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

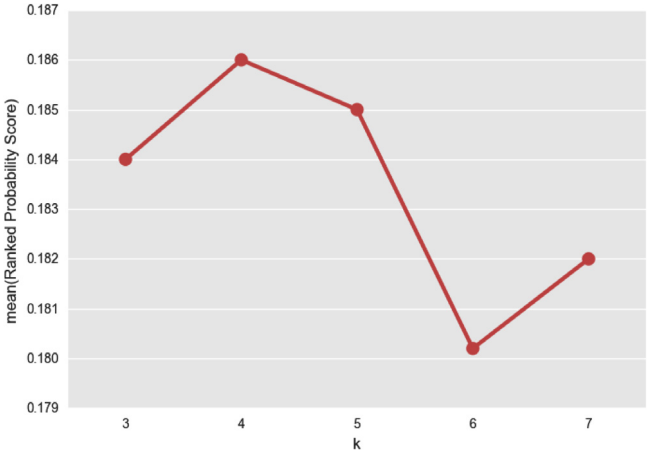


Fig. 6. 10-fold cross-validation results for the random forest for optimizing the hyper-parameter *k*.

Table 4
Linear SVM results.

(a) Confusion matrix			
	Predicted win	Predicted loss	Predicted draw
Actual win	254	36	0
Actual loss	96	93	0
Actual draw	122	39	0
(b) Precision–recall table			
	Precision	Recall	F1-score
Home wins	0.54	0.88	0.67
Away wins	0.55	0.49	0.52
Draws	0.0	0.0	0.0

Table 5
RBF SVM results.

(a) Confusion matrix			
	Predicted win	Predicted loss	Predicted draw
Actual win	238	37	15
Actual loss	84	96	9
Actual draw	110	36	15
(b) Precision–recall table			
	Precision	Recall	F1-score
Home wins	0.55	0.80	0.67
Away wins	0.57	0.51	0.54
Draws	0.39	0.09	0.15

Table 6
Random forest results.

(a) Confusion matrix			
	Predicted win	Predicted loss	Predicted draw
Actual win	225	36	29
Actual loss	64	101	24
Actual draw	86	40	35
(b) Precision–recall table			
	Precision	Recall	F1-score
Home wins	0.60	0.78	0.68
Away wins	0.57	0.53	0.55
Draws	0.40	0.22	0.28

Table 7
Gradient boosting results.

(a) Confusion matrix			
	Predicted win	Predicted loss	Predicted draw
Actual win	222	37	31
Actual loss	58	99	32
Actual draw	88	31	42
(b) Precision–recall table			
	Precision	Recall	F1-score
Home wins	0.60	0.77	0.67
Away wins	0.59	0.52	0.54
Draws	0.40	0.26	0.31

To overcome this issue, we then tested out the radial basis (RBF) kernel for our SVM and obtained better results (see Table 5(a) and (b)).

The RBF SVM performed quite well on both the home and away wins, with a staggering recall value of 0.80 on home wins, suggesting quite a low false negative rate. Regarding draws, even though the model returned a low false positive rate, as is indicated by its precision value of 0.39, it more than doubled the false negative error, giving a recall value of just 0.09. However, despite this drawback, our RBF SVM still outperformed the Gaussian naive Bayes on both the mean accuracy score and ranked probability score metrics.

The next model used was the random forest, which achieved a test score accuracy of 0.57. The random forest produced quite balanced results across all three classes. As expected, it performed extremely well on home wins, with a precision score of 0.60 and a recall score of 0.78, and it was also able to overcome the shortcomings of the SVM models relating to their poor performances on draws. Even though it also under-predicted draws, it performed much better than the SVM model on both the true positives and false negatives, returning an F1 score of 0.28, in comparison to 0.15 for the RBF SVM (Table 6(a) and (b)).

Our best performing model by far was the gradient boosted model. Even though the results (Table 7(a) and (b)) appear to be nearly identical to the random forest results, the gradient boosted classifier was found to be the best for modelling draws, the most unlikely event of a match.

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

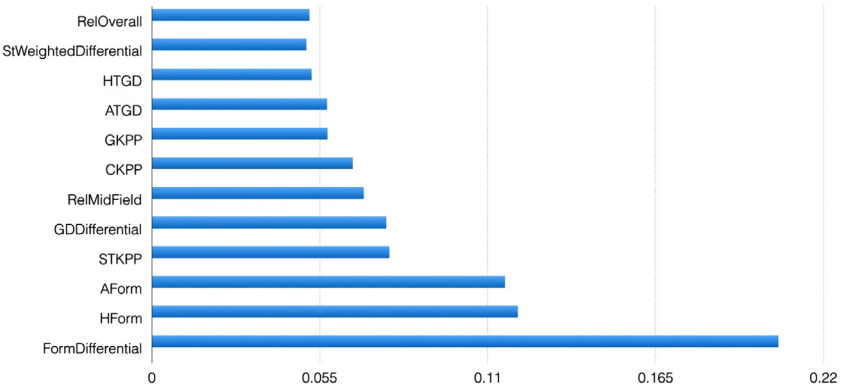


Fig. 7. Feature importance, recorded by the mean decrease in the Gini index.

This can be observed from the fact that gradient boosting has a higher recall (0.26) than the random forest (0.22) in the case of draws, ultimately leading to a higher F1 score (0.31) than that of the random forest (0.28). Even though the incremental gains in numerical values of the results of gradient boosting over those of the random forest may not appear large, the difference becomes quite evident and useful when we evaluate the performances of the two models on a different metric (RPS), as is explained later.

After tuning our random forest model properly, we analyzed our features and produced a feature ranking plot. We performed a feature importance analysis by recording the mean decrease in the *Gini Index* by splitting over a given predictor, with greater reduction indicating a higher importance of the feature. We observed (Fig. 7) that only two of the best performing features were not engineered features (*Rating* statistics), suggesting that feature engineering is crucial in the predictive analysis of football matches. We also discovered the high importance of the *Form* feature, which proved to be much more significant than our other features.

To add to our understanding further, we visualized the performances of our classifiers by implementing a receiver operating characteristic (ROC) curve. Analyzing an ROC curve is more useful and better in some avenues than using an individual error rate, as it is measured only for a single classification threshold, while the former plots for every discriminative threshold. Even though ROC curves are typically used for binary classification, they can be extended to multi-class classification by adopting the *one vs. all* approach and observing the micro-averaged curve for analyzing the overall performance of the classifier. It is quite evident from the ROC curves (Figs. 8(a)–8(c)) that the gradient boosted classifier outperforms the other models in terms of predicting both wins and draws, as well as in overall performance.

We determined the accuracy of our forecasting models using the ranked probability score (RPS). A thorough analysis of the different scoring rules used in assessing football forecasting models was undertaken by Constantinou and Fenton (2012), and it was demonstrated that RPS is a better validation metric than other scoring rules such as the Brier score and maximum log likelihood.

The ranked probability score was introduced in 1969 (Epstein, 1969), and is a measure of how well forecasts that are expressed as probability distributions match the observed outcomes. Both the location and the spread of the forecast distribution are taken into account when judging how close the distribution is to the observed value.

The RPS for a single instance is defined mathematically as:

$$RPS = \frac{1}{r-1}\sum_{i=1}^{r-1}(\sum_{j=1}^{i} p_j - e_j)^2,$$

where r is the number of potential outcomes, p_j is the forecasted outcome at position j, and e_j is the observed outcome at position j.

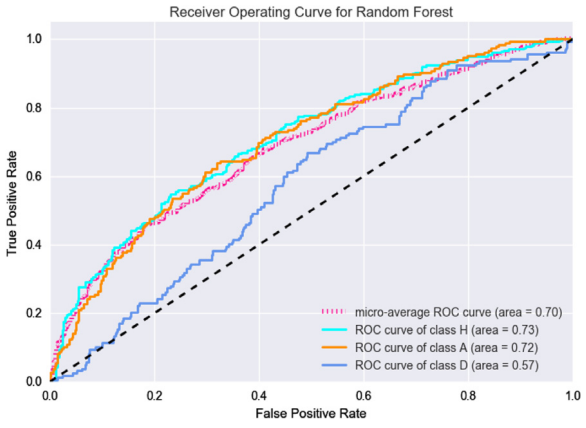
The lower the ranked probability score (RPS), the closer the predicted distribution is to the observed distribution, meaning that a lower RPS value signifies a better performance. The ranked probability scores of our forecasting models are displayed in Fig. 9. We found that the best performing models were the random forest and gradient boosting models, followed by the RBF SVM. It can be observed that the performance of the gradient boosting model is superior to that of the random forest classifier, which can be attributed to the fact that the gradient boosted classifier was better at modelling draws.

We also performed a comparison of the true and predicted results by class for our probabilistic (naïve Bayes) and ensemble (gradient boosted) methods (Figs. 10(a) and 10(b)), and found that the two models performed more or less the same in the case of away wins (with gradient boosting edging out naïve Bayes by a small margin). However, it can be observed that the ensemble algorithm (gradient boosting) is better at predicting draws than the probabilistic algorithm (naïve Bayes), as was mentioned above.

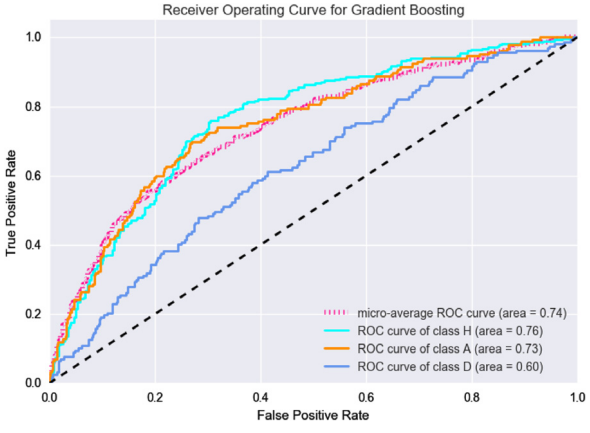
When performing classification, we want to predict not only the class label, but also the probabilities associated with each class, especially for football predictions. This probability gives us some kind of confidence in our predictions. Calibration curves (also known as reliability curves) are a way of assessing visually how confident a model is about its predictions. The predictions of a perfectly calibrated classifier can be interpreted directly as a confidence level. For instance, if a perfectly calibrated classifier

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

(a) ROC curve for RBF SVM.



(b) ROC curve for random forest.



(c) ROC curve for gradient boosting.

Fig. 8. Receiver operating characteristic curves for RBF SVM, random forest and gradient boosting.

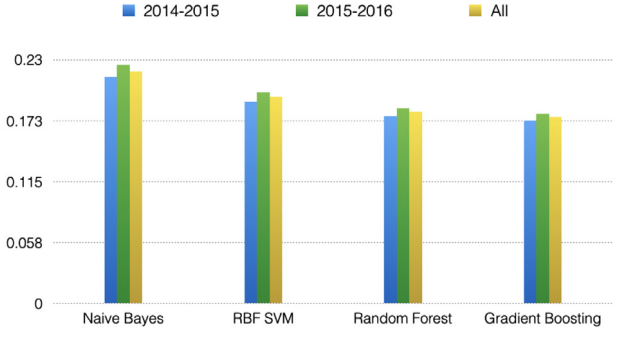
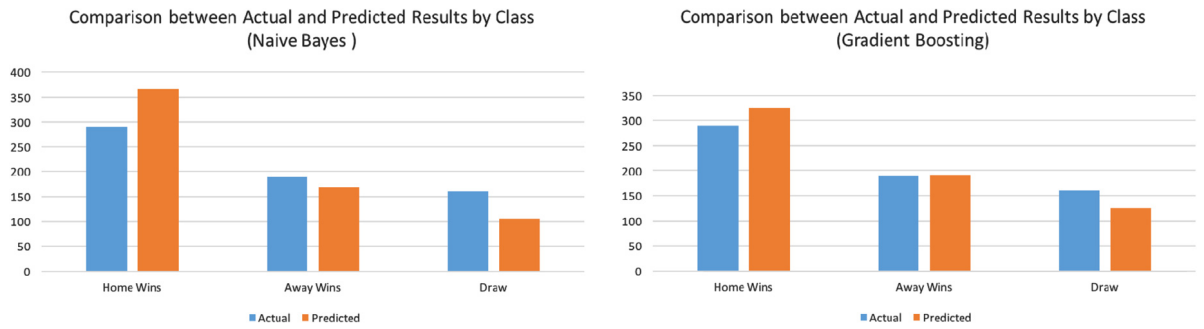


Fig. 9. Ranked probability scores of the different machine learning models (where *all* denotes the RPS across the entire test set (2014–2016)).

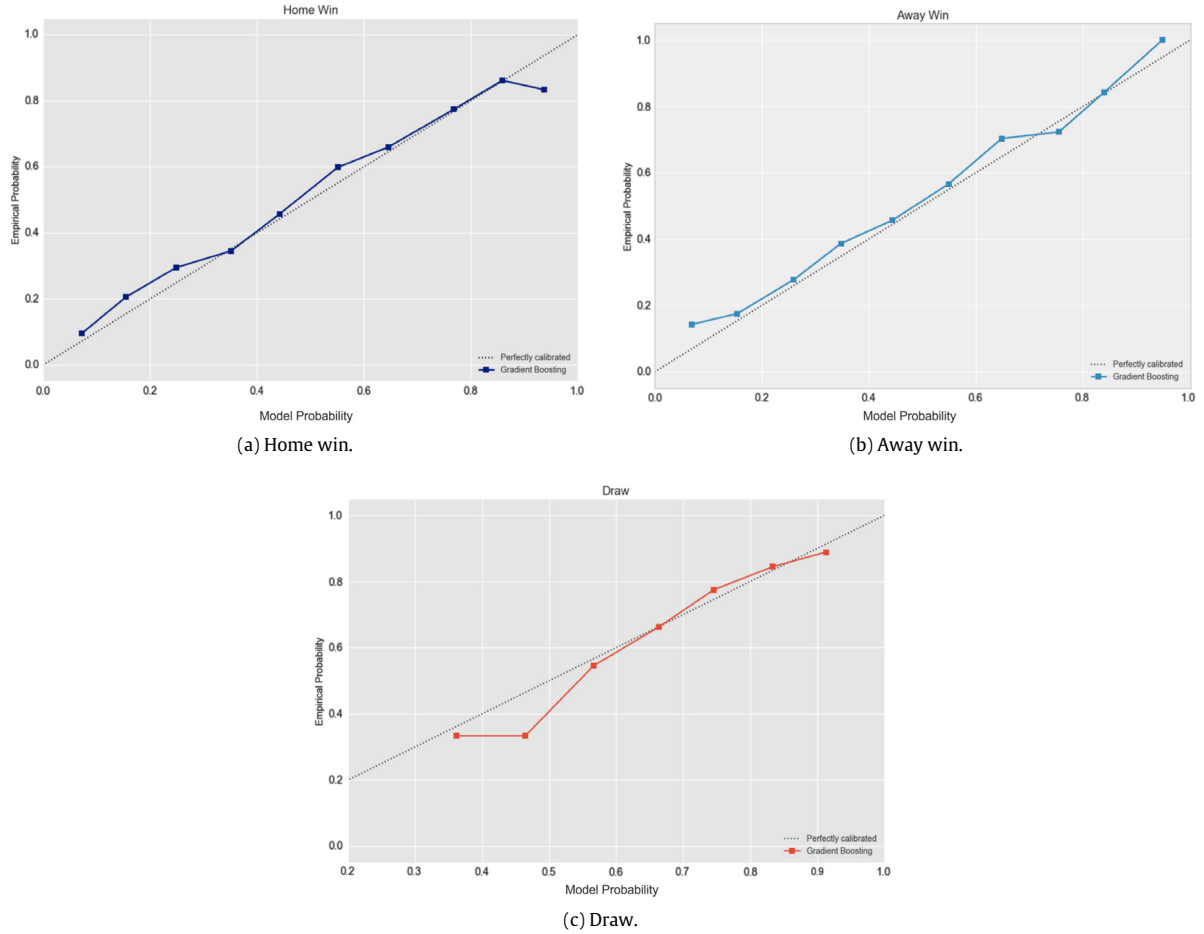
predicts that an event will occur with 0.6 probability, then that event will occur 60% of the time. As was said by Boshnakov et al. (2017), while perfect accuracy is probably unattainable for the problem of football prediction, perfect calibration is a much more achievable target, as an imperfect model can still have perfectly calibrated predictions. Since our best-performing model was the gradient boosted classifier, we visualized its calibration curves for the three outcome classes, *away win*, *draw* and *home win* (Figs. 11(a)–11(c)). A perfectly calibrated model will align exactly with the $y = x$ line. When the curve lies above the diagonal, the model underestimates the probability that the event will occur, while a curve below the diagonal indicates that it overestimates the probability of the event occurring. An inspection of our model calibration curves showed that the curves for all three classes were aligned with the $y = x$ line

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.



(a) Naïve Bayes. (b) Gradient boosting.

Fig. 10. Comparison between the true and predicted results by class for naïve Bayes and gradient boosting.



(a) Home win. (b) Away win.

(c) Draw.

Fig. 11. Probability calibration curves for the home win, away win and draw outcomes.

fairly well except at a few points (such as bin value (0.4–0.5) in Fig. 11(c)), which shows that our model predictions are reasonably well calibrated.

Various bookmakers' odds were used as a benchmark against which to compare our forecasts. The odds were all obtained from the same public data source, *Football UK* (http://www.bbc.com/sport/football/24354124). We used the implied probabilities (inverse odds) which were computed from the odds provided. In order to make greater profits, the bookmakers set unfair odds, where the implied probabilities sum to a value greater than one, meaning that the inverse odds are not representative of the actual probabilities. Thus, basic normalization, as mentioned by Štrumbelj (2014), was used to rectify this problem. The

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.

Table 8
Ranked probability scores of betting organizations and our machine learning models.

	Bet 365	Pinnacle Sports	Naïve Bayes	RBF SVM	Random forest	Gradient boosting
2014–2015	0.1951	0.1950	0.2233	0.2189	0.2137	0.2100
2015–2016	0.2074	0.2072	0.2349	0.2303	0.2253	0.2212
All	0.2012	0.2011	0.2291	0.2246	0.2195	0.2156

mathematical computation for performing such normalization is explained below.

Let $\boldsymbol{O} = \{o_1; o_2; o_3\}$ be the bookmakers' odds for *away win*, *draw* and *home win*, respectively. Then, the set of implied probabilities ($\boldsymbol{\pi} = \{\pi_1; \pi_2; \pi_3\}$) for the odds are computed by simply taking their inverse:

$$\pi_i = \frac{1}{o_i} \forall i = 1, 2, 3,$$

and we compute the normalized probabilities by dividing the inverse odds by the booksum (Π), such that:

$$p_i = \frac{o_i}{\Pi},$$

where

$$\Pi = \sum_{i=1}^{3} \pi_i.$$

We compared our model forecasts with the odds from two betting organizations, namely *Bet365* and *Pinnacle Sports*, and the results are depicted in Table 8. Since we had to drop the first k game weeks of each season, as has been mentioned, we compared our model predictions with each other as well as with those of the betting organizations from game week 6 onwards. Thus, the RPS values presented in Table 8 are from game weeks 6 to 38 for the English Premier League.

Our research has yielded quite encouraging results in regard to the problem of predicting football matches correctly. Table 8 shows that our random forest and gradient boosting models perform reasonably well relative to the bookmaker's predictions. Even though our model forecasts were not able to outperform the bookmakers' predictions, the predictions from our best performing model (gradient boosting) were only outperformed by a margin of around 0.014 to 0.015, which shows the promising nature of the models given the limited statistics that were available to the models relative to the betting organizations, which incorporate factors such as information about injured players and the presence of a key player.

6. Conclusion

While our research has produced promising results and provided many important insights in regard to the factors that come into play when analyzing a football match, it still lags behind a few leading industry methods. The major constraint faced by our models was the limited amount of data that we had. Even though we have been able to engineer many important and key features, one major factor in yielding better results would be the availability of more detailed and sophisticated features, such as injury information, the presence of a key player, psychological effects, and many other such factors. Another interesting avenue that we could explore would be the use of social media data to facilitate this. We also plan to use regression-based models to compute the goals scored by each team, thus performing an even more in-depth study and analysis of football matches. Given the quite reasonable results that we have achieved, we could also extend our system to the generation of betting odds for the analysis of economic profits.

Acknowledgments

This research work was catalyzed and supported by the National Council for Science and Technology Communications (NCSTC), Department of Science and Technology (DST), Ministry of Science and Technology (Govt. of India), New Delhi, India [grant recipient: Dr. Harleen Kaur and grant No. 5753/ IFD/ 2015-16]. The authors gratefully acknowledge financial support from the Ministry of Science and Technology (Govt. of India), New Delhi India.

References

Asif, M., & McHale, I. G. (2016). In-play forecasting of win probability in one-day international cricket: a dynamic logistic regression model. *International Journal of Forecasting*, *32*, 34–43.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: a comparative study. *Decision Support Systems*, *50*, 602–613.

Boshnakov, G., Kharrat, T., & McHale, I. G. (2017). A bivariate Weibull count model for forecasting association football scores. *International Journal of Forecasting*, *33*, 458–466.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Chauhan, R., & Kaur, H. (2014). Predictive analytics and data mining: a framework for optimizing decisions with R tool. In *Advances in secure computing, internet services, and applications* (pp. 73–88). Springer.

Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794).

Constantinou, A. C., & Fenton, N. E. (2012). Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models. *Journal of Quantitative Analysis in Sports*, *8*, 1559-0410.

Constantinou, A. C., & Fenton, N. E. (2013). Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries. *Journal of Quantitative Analysis in Sports*, *9*, 37–50.

Constantinou, A. C., Fenton, N. E., & Neil, M. (2012). pi-football: A Bayesian network model for forecasting Association Football match outcomes. *Knowledge- Based Systems*, *36*, 322–339.

Constantinou, A. C., Fenton, N. E., & Neil, M. (2013). Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks. *Knowledge-Based Systems*, *50*, 60–86.

Crowder, M., Dixon, M., Ledford, A., & Robinson, M. (2002). Dynamic modelling and prediction of English football league matches for betting. *Journal of the Royal Statistical Society. Series D. The Statistician*, *51*, 157–168.

Dixon, M. J., & Coles, S. G. (1997). Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, *46*, 265–280.

Elo, A. (1978). *The rating of chess players, past and present*. New York: Arco.
Epstein, E. (1969). A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, *8*, 985–987.
Goddard, J. (2005). Regression models for forecasting goals and match results in association football. *International Journal of Forecasting*, *21*, 331–340.
Godin, F., Zuallaert, J., Vandersmissen, B., De Neve, W., & Van de Walle, R. (2014). *Beating the bookmakers: leveraging statistics and Twitter microposts for predicting soccer results*. Workshop on Large-Scale Sports Analytics, KDD.
Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*, 389–422.
Hvattum, L. M., & Arntzen, H. (2010). Using ELO ratings for match result prediction in association football. *International Journal of Forecasting*, *26*, 460–470.
Joseph, A., Fenton, N. E., & Neil, M. (2006). Predicting football results using Bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, *19*, 544–553.
Karlis, D., & Ntzoufras, I. (2010). Robust fitting of football prediction models. *IMA Journal of Management Mathematics*, *22*, 171–182.
Kaur, H., Chauhan, R., & Ahmed, Z. (2012). Role of data mining in establishing strategic policies for the efficient management of healthcare system –a case study from Washington DC area using retrospective discharge data. *BMC Health Services Research*, *12*(S1), P12.
Kaur, H., Chauhan, R., & Alam, A. M. (2010). *An optimal categorization of feature selection methods for knowledge discovery, visual analytics and interactive technologies: data, text and web mining applications*. IGI Global Publishers.
Kaur, H., Chauhan, R., & Alam, M. A. (2011). Spatial clustering algorithm using R-tree. *Journal of Computing*, *3*(2), 85–90.
Kaur, H., Lechman, E., & Marszk, A. (2017). *Catalyzing development through ICT adoption: the developing world experience*. Switzerland: Springer Publishers.
Kaur, H., & Tao, X. (2014). *ICT and millennium development goals: a United Nations perspective*. New York, US: Springer Publishers.
Koopman, S. J., & Lit, R. (2015). A dynamic bivariate Poisson model for analysing and forecasting match results in the English Premier League. *Journal of the Royal Statistical Society. Series A. Statistics in Society*, *178*, 167–186.
Lessmanna, S., Sung, M.-. C., & Johnson, J. E. V. (2010). Alternative methods of predicting competitive events: An application in horserace betting markets. *International Journal of Forecasting*, *26*, 518–536.
McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th python in science conference* (pp. 51–56).
Odachowski, K., & Grekow, J. (2013). Using bookmaker odds to predict the final result of football matches. In *Proceedings of the 16th international conference on knowledge engineering, machine learning and lattice computing with applications* (pp. 196–205).
Owramipur, F., Eskandarian, P., & Mozneb, F. S. (2013). Football result prediction with Bayesian network in Spanish league-Barcelona team. *International Journal of Computer Theory and Engineering*, *5*, 812–815.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research (JMLR)*, *12*, 2825–2830.
Rotshtein, A. P., Posner, M., & Rakityanskaya, A. B. (2005). Football predictions based on a fuzzy model with genetic and neural tuning. *Cybernetics and System Analysis*, *41*, 619–630.
Rue, H., & Salvesen, O. (2000). Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society. Series D. The Statistician*, *49*, 399–418.
Schumaker, R. P., Jarmoszko, A. T., & Labedz, C. S. (2016). Predicting wins and spread in the premier league using a sentiment analysis of Twitter. *Decision Support Systems*, *88*, 76–84.
Štrumbelj, E. (2014). On determining probability forecasts from betting odds. *International Journal of Forecasting*, *30*, 934–943.
Štrumbelj, E., & Vračar, P. (2012). Simulating a basketball match with a homogeneous Markov model and forecasting the outcome. *International Journal of Forecasting*, *28*, 532–542.
Vračar, P., Strumbelj, E., & Kononenko, I. (2016). Modeling basketball play-by-play data. *Expert Systems with Applications*, *44*, 58–66.
Zhou, W., & Kapoor, G. (2011). Detecting evolutionary financial statement fraud. *Decision Support Systems*, *50*, 570–575.

Rahul Baboota is a Computer Science undergraduate at Indraprastha University, India. His research interests include data mining, machine learning and deep learning, as well as their applications to sports analysis and computer vision. He is currently working on a project funded by the Department of Science and Technology (Government of India) that focuses on social media healthcare analysis. He is also a part of the Center for Artificial Intelligence at Indraprastha Institute of Information Technology.

Harleen Kaur is a faculty member at the School of Engineering Sciences and Technology at Jamia Hamdard, New Delhi, India. She has recently worked as Research Fellow at United Nations University (UNU) in IIGH-International Centre for Excellence, Malaysia where she has conducted research on funded projects from South-East Asian Nations (SEAN). She is currently working on an Indo-Poland bilateral international project funded by the Ministry of Science and Technology, Govt. of India (GoI), New Delhi, India and the Ministry of Polish, Poland. In addition, she is working on a national project on intelligent big data analytics catalysed and supported by the National Council for Science and Technology Communication (NCSTC), Ministry of Science and Technology, Govt. of India (GoI), New Delhi, India. Her key research areas include data analytics, big data, applied machine learning and predictive modeling. She is the author of many publications in referred journals and also authored/ edited several reputed books. She is a member of various international bodies and also a member of the editorial board of international journals on data analytics and machine learning. She is the recipient of Ambassador for Peace Award (UN Agency) and also active researcher duly funded by external agencies.

Please cite this article in press as: Baboota, R., Kaur, H., Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting (2018), https://doi.org/10.1016/j.ijforecast.2018.01.003.