[redacted]

RANDOM FOREST ALGORITHMS FOR EVALUATING OFFENSIVE EFFICIENCY IN FOOTBALL

Ioannis Constantinides
483435

Supervisor: Velden, M. van de
Second assessor: Alfons, A

Master Thesis [Econometrics with Specialisation in Business Analytics and Quantitative Marketing]

The content of this thesis is the sole responsibility of the author and does not reflect the view of either Erasmus School of Economics or Erasmus University

1 May 2020

Abstract

Football is a team sport which has been changing its view towards data-backed decisions in the past decade. In this study, we focus on offensive efficiency in football and use modified random forest algorithms to construct an expected goals metric. The low-scoring nature of the game created a large gap between the overall attempted shots and those which were converted to a goal. The modified methods employed account for the imbalance in the data. We find that using a random forest-quantile classifier outperforms other algorithms in terms of predictive accuracy. The constructed metric assigns a probability of scoring for each shot given some characteristics, including individual skill, spatial-temporal data and defensive ability of the opposing team. We then provide ideas for applications of such a metric and discuss how it can potentially be a useful tool for football clubs.

Contents

1 Introduction

The use of sports analytics in football decision making has been gaining ground in the past decade and has many times proven its potential for better, more informed in and out of pitch decisions (Kidd (2018),Kumar (2013)). Rasmus Ankersen [1], a strong advocate for the use of analytics in football decisions, argues that replacing traditional decision making with statistical analysis can provide a competitive edge over opponents. The owner of Danish club "FC Midtjylland" has seen unprecedented success in recent years through the use of data analysis. His argument is enforced by his club's rapidly inclining position from the time he took over. Still research on football analytics is limited, when compared to other sports like baseball or tennis. The dynamic and open nature of football is what makes it both interesting but also challenging for statistical analysis (Ruiz, Lisboa, Neilson, and Gregson (2015)). The football industry sees countless decisions that have to be made both considering the in play related ones, like players to scout and acquire and efficient tactics to use, as well as ones related to commercial profit for the club, financial stability and others. Undoubtedly, these decisions can be optimized through the use of historical data and better informed decisions [2].

A major tactical part in football lies with optimizing the offensive efficiency of a team by being able to accurately evaluate attacking players. The most valuable ability of an attacking player is being able to convert opportunities to goals. A complication arises with the low frequency of goals, with the average shot conversion in football being around 10% for competitive games(Bertin (2015b)). Evaluating offensive ability is therefore dependent on a low frequency variable when compared to a series of events which occur in a 90-minute game.

A possible approach is to focus more on total shots taken towards goal. Since goals in football inevitable emerge from shots on target, this is a sensible approach. However, only using the number of shots taken towards goal does not provide any information on the quality of the shot and as a result the likelihood of a shot resulting to a goal. Factors like location of the shot, ability of the shot taker, defending pressure from the opposing site, are all considered constant when only looking at the number of shots taken. The benefit of taking into account these different factors is that different importance weights can be provided to each shot taken, where a higher likelihood of conversion denotes a larger expected offensive impact.

The notion of evaluating the likelihood of a shot being converted, or goal expectancy[3] manages to take into account these kind of factors in order to better evaluate offensive efficiency. By accounting for all factors which may affect a shot, a probability of each shot resulting in a goal can be derived. This provides further insights for offensive ability. The analysis can uncover insightful information of tactics and shot locations which result to higher offensive efficiency. Additionally, the construction of an expected goals metric supplemented by actual goals can more accurately describe the offensive efficiency of player or a whole team. This can indeed prove more helpful than standard descriptive statistics like shots on target or shot conversion rate.

In this study, we focus on estimating the probability outputs for each shot in our data-set with the use of a variety of random forest algorithms. As previously implied, the low prevalence of goals, results in an imbalanced data-set for which standard classification algorithms prove to perform relatively badly. We expect that taking the imbalance into account gives better probability estimates We further explain how a normal random forest algorithm is affected by the imbalance problem and investigate how different random forest methods perform when faced with this problem.

[1] How data, not people, call the shots in Denmark
[2] How computer analysts took over at Britain's top football clubs
[3] Goal Expectation and Efficiency

2 Literature Review

Researchers on football analytics used to regard shots on target as the ultimate measure of offensive performance (Armatas, Yiannakos, Papadopoulou, and Skoufas (2009),Armatas and Yiannakos (2010)). However, this statistic alone, does not provide any information on the quality of each shot taken and consequently, its associated probability resulting to a goal. In recent years, more researchers have decided to shift focus on the notion of expected goals and construct a metric which incorporates useful information describing a shot. The rationale is to more accurately distinguish between high and low probability of shot conversion and acquire a better measure of offensive efficiency.

Numerous factors play a role when trying to assess a given shot. Location of the ball and distance away from goal have been the most prevalent ones, firstly introduced by Pollard and Reep (1997) and subsequently used by the vast majority of the literature on the topic (Bertin (2015b), Armatas and Yiannakos (2010), Eggels, van Elk, and Pechenizkiy (2016), Rathke (2017), Lucey, Bialkowski, Monfort, Carr, and Matthews (2014)). As suggested by Armatas and Yiannakos (2010), the location of the ball at the time of the shot significantly impact the probability of it being converted. More specifically, a large difference in conversion probability was found for shots taken inside the box when compared to those outside. The two factors are both essential as only including distance would then completely dismay the effect of the angle to the goal. Fairchild, Pelechrinis, and Kokkodis (2018) provide the rationale for that, arguing that even if the distance of two shots is the same, a shot taken from a convenient angle is more likely to result in a goal. Defender proximity at the time of a shot has been seen by many studies (Bialkowski et al. (2014), Pratas, Volossovitch, and P Ferreira (2012)) as another crucial factor. Also characteristics such as which part of the body, assist method and speed of the shot have been widely used (Pratas et al. (2012)).

A lot of studies identified the importance of situational variables like quality of opposition, the psychological pressure on the offensive players and the time of the game for offensive efficiency (Pratas et al. (2012), Tiehan (2015)). Along with that, the findings of Eggels et al. (2016) indicate that the quality of the attacking side is also important. These studies used average metrics of team quality both for the attacking and defending teams. Instead of using this approach, for the attacking team and more specifically the individual player who executes the shot, we solely focus on characteristics which directly impact the player's shooting ability. These are described in more detail in Section 3.

Rampinini, Impellizzeri, Castagna, Coutts, and Wisløff (2009) suggest an increase in performance of top quality offensive players when faced against more competitive teams. For that, we create a variable which incorporates the rank difference between the two teams for each game played and evaluate higher quality players based on that. Riley (2014) chooses an approach which focuses only on shots on target, boosting the learning procedure of the algorithm with a higher number of ”good” quality shots. The scope of this study is to easily distinguishing between higher and lower quality shots, therefore more information for lower quality shots is beneficial so we do not follow this approach. Instead all attempted shots are included in the analysis.

The advancements in data gathering and processing of recent years, have made sports analytics a convenient tool which is now widely used by researchers. Many researchers started developing their own predictive goals models which is essentially categorized as a supervised classification problem with only two possible outcomes (Riley (2014), Bertin (2015a), Matthias (2014)). The prevalent and most widely used approach is that of logistic regression. Its simplicity is its main advantage and can arguably work well in some cases as seen by Colin (2013). Lucey et al. (2014) incorporate strategic aspects of the game, such as assist method, and estimate the likelihood of each shot resulting in a goal with logistic regression. In this study, the applications of such a model for both individual and team offensive performance are clearly explained and developed.

A limitation of this approach lies with the low-scoring nature of football. This implies that most data sets which utilize all executed shots available, are imbalanced with the vast majority of observations being labelled as not successful, or no goal. The logistic regression does not in itself correct for that imbalance, inducing bias towards predicting more successful efforts as not successful. The bias in predictions arises from the structure of the objective function needed to be maximized. This function enforces equal weights to the likelihood for each class, consequently penalising an incorrect misclassification in the minority class the same as for the majority. The result is a very low number of observations classified to the minority class. Okabe, Tsuchida, and Yadohisa (2019) provide a weighted logistic regression which accounts for this problem. Also in the study by Lucey et al. (2014) this is again evident when comparing the average error rate of correctly predicting the 'goals' class when compared to 'no goals'.

Since goals are rare in football, Eggels et al. (2016) overcome the problem of imbalance by employing a mixture of oversampling the minority group (Chawla, Bowyer, Hall, and Kegelmeyer (2002)) and cluster based under-sampling (Yen and Lee (2009)) before they train their classification algorithms. In their study they employ and compare three classification algorithms, namely, Logistic Regression, Random Forests and ADA boost. Their findings hint towards less volatile and more accurate predictions for both classes with the use of the Random Forest algorithm. We also use the Random Forest algorithm for the classification problem as described in section 1 but choose a variety of different techniques to deal with the problem of imbalance. These are described in more detail in section 4.

Lastly, Ruiz et al. (2015) build an expected goals model using an artificial neural network. The output of their model are the associated probabilities of scoring for each shot which are then used to evaluate the performance of offensive players in the English Premier League. Their work gives a good idea of how such outputs could be applied to the football industry. In our work we take a similar structure to better understand and apply the predicted probability outputs for our model.

3 Data Description

This section is used to describe the data-sets used for the study. It gives an overview of each data-set used, and the description of important variables. Also, we provide explanations on the relational schema followed for merging these data-sets.

3.1 Data-Sets

For this study, six data sets are used. The Events data is the most crucial one, which describes all events that occurred during a football game. The contextual data gives further information for each game described in the Events data. Next, for individual and team ratings, we use data from the video-game 'FIFA 16'. Lastly the Team Ranking database consists of 3 data tables of rankings for each of the three leagues utilized for this study. More detailed information for all data-sets is given in the sections below.

3.1.1 Events Data

The Events data used for this study, is taken from 'Kaggle'[4]. For this study I only use data for the three biggest leagues in European football, namely the English 'Premier-League', the German 'Bundesliga' and the Spanish 'La-Liga'.

This data-set consists of 11 different events or instances in the game including passes, interceptions, set-pieces and attempts on goal among others. For the three leagues the data consists of $5,112$ matches played and a total of $445,339$ events. Matches are recorded from 2011/2012 season to 2016/2017 season As described in Section 1 this study focuses on assigning the probability of each shot resulting to a goal, thus out of all events in the data-set only completed shots are considered. These, amount to a total of $60,645$ observations. The relevant variables which characterize each attempted shot are defined in detail in table 1.

The dependent variable '*is_goal*' is binary and the aim of this study lies with accurately predicting it for previously unseen data. All variables which describe in game characteristics except '*time*' described in table 3.1.1 are categorical. More information on the important variables like '*location*' is given in section 5.

It should be noted that the source of the data-set mentions that the data is collected manually and is derived by text commentary on football games.

[4] https://www.kaggle.com/secareanualin/football-events

Table 1: Events Data

Variable	Description
id_game	Unique identifier of game
id_event	Unique identifier of event
time	Minute of the game
event_type	Primary Event that occurred. 11 unique events recorded.
event_team	Team that produced the event.
opponent	Team which the event happened against.
player1	Name of player involved in primary event.
shot_place	In the event of a shot, there exist 13 possible placement locations of the shot.
shot_outcome	4 possible outcomes in the event of a shot (Blocked, on taget, off target, crossbar)
is_goal	Binary variable indicating if the shot resulted to a goal.
location	Location of player at the time of the primary event.
bodypart	Bodypart by which primary event occurred.
assist_method	Method of assist in case of a shot.
situation	4 types: 1-Open Play, 2-Set piece, 3-Corner, 4-Free kick
fast_break	Binary Variable indicating whether the primary event occured in counter attack or not.

3.1.2 Contextual Data

The 'Contextual' data set contains important information on the events described above and these include things like the season and the date of the game. Table 11 in the appendix (8) gives a detailed description of these variables.

3.1.3 FIFA Ratings

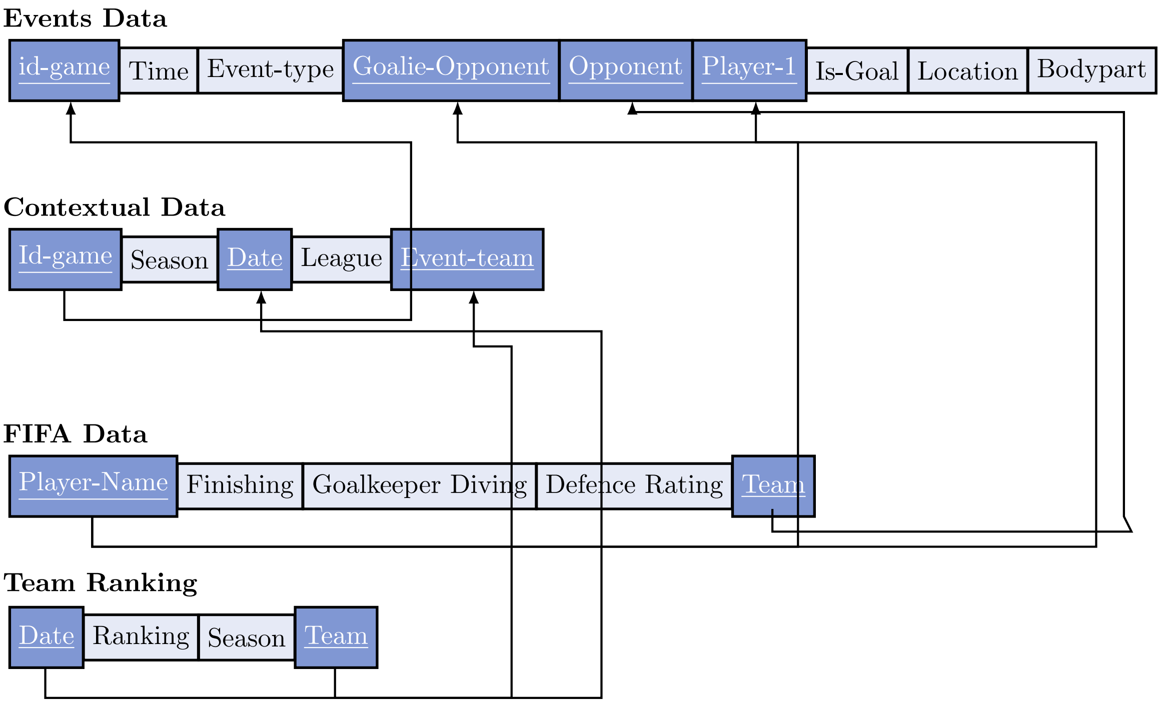
The FIFA football video-game has historically been perceived as a good source of extracting individual player statistics due to its abundance of detailed ratings on a variety of football related attributes. For this study, we use ratings for the 'FIFA 17'[5]. The data-set consisted all available attributes for each player. For this study only some of these attributes are relevant. Firstly, attributes related to shooting ability is deemed important for in-field footballers. Then, to have a measure of resistance the relevant attributes for goalkeeper abilities are extracted. Lastly, to have a measure of defending pressure from the opposing team the overall defending ability of each team is extracted. Detailed explanation of the variables can be found in Tables 12 and 13.

3.1.4 Team Rankings

This data-set is used to extract information on the difference in ranking between the two teams competing in each game. More information on the variable is given in Section 5. Three different data-sets are constructed for each relevant league in the analysis. Using the official websites for each league, we use the web-scraping package '*rvest*' in 'R' to extract the final rankings for all the seasons involved in the events data-set. Each data-set contains a column indicating the team name and 6 columnns, one for each season played with the elements being the final ranking of the team in the given season.

[5] `https://www.kaggle.com/artimous/complete-fifa-2017-player-dataset-global`

Figure 1: **Relational Diagram of Tables**
The highlighted variables represent the 'keys' used to merge into one final data-set



3.2 Merging the Data-Sets

For our analysis to be carried through, the data-sets described in section 3.1 are all merged into one final data-set using the relevant 'keys' from each. The merge procedure is acchieved using the package '*dplyr*' in 'R' as it provides fast and efficient merging of tables. Keys[6] are defined as the variables which can be used as links between different tables. In figure 1 the keys used are highlighted among some but not all of the relevant variables for each data-set.

The 'Team Rankings' table, described in Section 3.1.4, is linked with the 'Contextual' data-set based on both the 'team' but also the date of observation. Note that for each team there are 6 different rankings for each season thus linking with key only set on the respective team would not produce the desired results. Then the $Id - game$ variable, a unique code for each game, is used to link the 'Events Data' with the 'Contextual Data'.

The 'FIFA Data' contains three important aspects needed to be merged with the 'Events Data'. "$Player - 1$' describes the player which executed a shot in the 'Events Data'. This is linked with the '$Player - Name$' from 'FIFA Data' to extract the relevant shot attributes. Secondly, we need a measure of the opponents defending abilities as a measure of resistance to the player executing the shot. Given data on the opponent team we find the goalkeeper for the given season. Then, given the '$Player - Name$', we link the relevant goalkeeping attributes to the "Event Data".

A limitation of the 'Events Data' is that it does not contain Spatiotemporal data on defenders proximity. Lucey et al. (2014) and Fairchild et al. (2018) both suggest that the distance of the closest defender to the ball at the time of a shot, or defender proximity, has a significant role in predicting whether a shot results to goal or not. Specifically

[6] `https://www.techopedia.com/definition/1780/key`

the shorter the distance to the ball of the defender, the lower the probability of conceding a goal. To overcome this drawback, we extract the overall defence rating as given in the ’FIFA Data’ as a measure of general pressure applied to the shot taker. A higher defence rating for the team defending a shot, is generally associated with more ’resistance’ to the opposing team trying to attack. We expect that this metric captures some of the variation of defender proximity. The $'Team'$ key is linked with $'Opponent'$ from the Events table and then the goalkeeper for the team in the current season is derived. At the end, we end up with one table containing all the information needed for the analysis.

4 Methodology

This section gives an overview of the methods used for this study. As the case in the majority of the studies discussed in Section 2, we start by briefly discussing the logistic regression for classification. We then introduce the idea of decision trees for classification and then move on to Random Forests. We then delve deeper in the variations of the original Random Forest algorithm, which correct for the problem of imbalance in the data-set.

As a supervised classification problem, there are numerous different algorithms that could have been used instead of Random Forests (Hastie, Tibshirani, Friedman, and Franklin (2005)). We choose to mainly focus on the random forest algorithm and evaluate the performance of different variations of the algorithm in the presence of imbalanced data sets.

4.1 Logistic Regression

The logistic regression is a linear classification method which manages to model the probability of an observation belonging to a certain class. It follows the assumptions of the ordinary least squares method but at the same time constraining the target variable between 0 and 1 (Hastie et al. (2005)). For the binary setting of this study, we want to model $p(X)$ where $p(X) = Pr(Y = 1|X) = \beta_0 + \beta_1 X$ and X is the matrix of independent variables. In order to constrain the target variable between o and 1 and transforming it to a probability the logistic regression utilizes the logistic function:

$$p(X) = \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)} \tag{1}$$

In order to chose values for the parameters of logistic regression, we use maximum likelihood estimation. After setting up $p(X)$, we set up the log-likelihood function for the product of all observations. We maximize this function by setting its first order derivative with respect to the β's equal to 0. The estimation of the β's is done and therefore we can use those to make probability predictions on the target variable. We will not further discuss their estimation as it is outside of the scope of this study. As James, Witten, Hastie, and Tibshirani (2013) argue, the logistic regression can work well in cases where the true relationship between the independent variables and the target variable is linear. Also, its greatest advantage over more sophisticated machine learning algorithms arises due to the clear interpretations of the explanatory variables. By the use of the log-odds (James et al. (2013)) that can be derived from the estimated coefficients, we can accurately understand the relative effect of a factor on the overall probability of classification. The output for each observation i is a probability ($Pr(Y_i = 1|X_i)$). We then need to set a threshold probability for which any case above is classified as a goal and below as no goal. A natural choice and widely used is 0.5. In the sub-sections which follow, we tune our algorithms such that mis-classification is minimized. Thus for consistency reasons, we set the threshold such that it minimizes the mis classification error.

The main disadvantage of the method is that we can not account for non-linear dependencies in the explanatory variables, as its decision criterion is linear. In problems with a large number of explanatory variables, it is highly likely that non-linear relationships arise. Additionally, in cases with categorical factors, each category is represented by a separate variable, further increasing the number of variables in the regression.

To better account for these limitations, non-linear classification algorithms can be employed. Decision trees and random forests are expected to perform better in our problem where non-linear dependencies are expected to be present in the data due to the highly associated explanatory variables discussed in section 3. We use the logistic regression as a benchmark to compare the performance of more complex classification algorithms.

4.2 Decision Trees for Classification

Prior to talking about the main methods used in this study, it is necessary to have a short introduction on the idea of decision trees for classification problems (James et al. (2013)).

The scope of a decision tree is to find a function $\hat{f}(X)$, such that the predictors X are mapped to the response variable Y. For this study, the response variable Y_i is a binary variable for all observations i. In essence, decision trees segment the predictor space into sub-regions (R) using a suitable splitting rule. These sub-regions are mutually exclusive and each observation in the training set belongs to a sub-region. In the case of classification, the response variable is qualitative thus we predict the "most commonly occurring class" (Lewis (2000)) for the training observations in the region. This is also known as the Majority class vote.

An important aspect which will also be used later when we get in more detail about random forests, is the choice of splitting rule. In regression trees, it is common to minimize the Sum of Squared Residuals (SSR). However, the SSR is not meaningful in the case of classification. An alternative measure to split the predictor space into the m^{th} region is the Gini index:

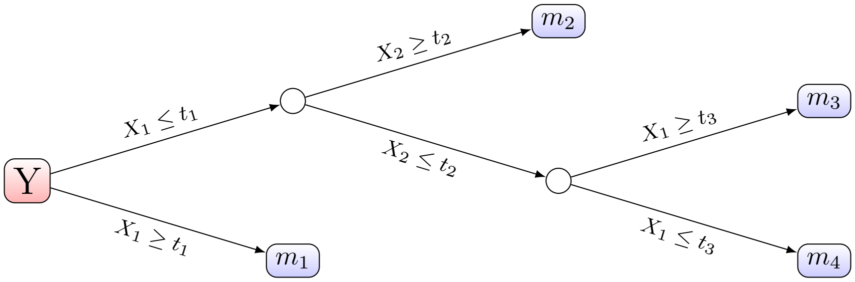
$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2}$$

$\hat{p}_{mk}$ is the proportion of observations in the m^{th} region that actually belong in the k^{th} class. Thus, the Gini index(2) can be interpreted as a measure of variance within the m^{th} region. The tree is grown based on this criterion until a stopping condition is met.

In the case of binary classification there exist only 2 classes. Assuming that p is the proportion of observations in the second group and that the proportion of all classes must add up to one, equation 2 becomes:

$$G = p(1 - p) \times (1 - p)p = 2p(1 - p) \tag{3}$$

The splitting rule described above decides for each node which variable and at which value the optimal split lies. If no further splitting benefits the Gini- index, the observations up to that point make one region. We also provide a simplified example to visualize how tree partitioning works.



In this example, we assume we only have two predictors, X_1 and X_2. We also assume that these predictors are continuous and t_i for $i = 1, 2, 3$ are values of the predictors. Starting from X_1, we split the predictor space in two, grouping observations below and above t_1. We see that no further splitting of observations can be made for $X_1 \geq t_1$ thus those observations make region m_1. We continue partitioning the tree for observations which satisfy the condition $X_1 \leq t_1$.

Decision trees are conceptually easy to build however should be grown large in order to avoid bias. A major limitation of large decision trees is their 'instability' or high variance. This means that changing the training set could result to different estimation and therefore predictions of the target variable. This arises due to random errors at each node propagated to the next ones. In the above example, a low error in the first split of X_1 is propagated until the last split of X_1. A way to deal with this problem is to build an ensemble of trees and average their predictions to get a more reliable estimate. In the next section we explain how the process of 'Bagging' reduces the variance of individual trees.

4.3 Bagging

A key concept leading to the Random Forest Classifier, is the idea of bagging or constructing an ensemble of trees firstly introduced by Breiman (1996). The introduction to the idea of bagging is what inspired Breinman to later introduce the Random Forest algorithm.

As described by Hastie et al. (2005), bagging essentially averages the predictions made over a bootstrapped sample. Bootstrapping (Efron and Tibshirani (1985)) in machine learning, is the process by which many sub-samples (with replacement) of the whole observation set are extracted.

Bagging of decision trees is the process of constructing B decision trees for each bootstrap sub-sample of size N and then using the majority vote, for classification problems, to assess predictions. Hastie et al. (2005) imply that the bias of B such predictions is the same as that of an individual one as each tree is identically but not necessarily independently (due to the bootstrap samples) distributed. So no improvement in bias is achieved through bagging. From the previous section, we defined $\hat{f}(X)$. The limitation of decision trees is that for a number of training sets, $\hat{f}(X)$ could exhibit high variation. To overcome this limitation, bagging constructs B trees out of B random training sets and averages the predictions made for each set:

$$\hat{f}_{bag}(X) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(X) \tag{4}$$

This process reduces the variance of the classifier even when the depth of the tree is large. Bagging an ensemble of trees can be much more reliable than growing an individual tree for predictions. However the random training sets extracted out of the same data-set are prone to high correlation between them. As Hastie et al. (2005) has shown, the higher the correlation between the training sets, the higher the variance of the new averaged estimate.

The next section introduces the Random Forest Algorithm, which manages to significantly reduce the correlation between the B trees thus decreasing the variance of the averaged estimate.

4.4 Random Forests

The Random Forest algorithm was firstly introduced by Breiman (2001). Decision trees are able to capture interactions between explanatory variables by splitting the predictor space and then make predictions for the target variable. However, as discussed in Section 4.2, decision trees can be extremely noisy when it comes to their predictive performance(Hastie et al. (2005)). A method to compensate for the extreme variability of an individual decision tree is Bagging (4.3). Still, the variance of the bagged estimate, $\hat{f}_{bag}(X)$ increases as the correlation between each tree is higher (Hastie et al. (2005)). Lower correlation between the estimates, can result in lower variability of the final prediction. The Random Forest Algorithm (1) manages to provide more precise estimates relative to bagging of decision trees, by reducing the correlation between the B constructed trees, thus "the improvement made is by further reduction of the variance" (Hastie et al. (2005)). The algorithm as described by Breiman (2001):

```
Result: Random Forest for Classification
1. For b=1,..B;
  A) Draw a bootstrap sample $Z$ of size $N$ (with replacement);
  B) Construct a decision tree ($T_b$) using $Z$ by following the below steps for every terminal node, until the
  minimum node size is reached ;
      i) Select at random $m$ out of the $p$ available variables ;
      ii) Use the Gini criterion (2) to find the optimal split and optimal variable ;
      iii) Split the node in two resulting nodes ;
2. Output the ensemble of Trees $T_b$ for b =1 to B. ;
3. For each tree, classify a new observation $x$ using the majority vote:
Let $\hat{C}_b(x)$ be the class prediction of the $b^{th}$ tree for observation x. ;
4. Using the output from (3), classify $x$ using the majority vote rule from all B trees.
$\hat{C}^b_{rf}$ =majority vote $\hat{C}_b(x)^B_1$
```

Algorithm 1: Random Forest - Classification

At each node of a tree, Algorithm 1 randomly chooses a subset of the whole predictor space. Also each of the B trees is grown on a different bootstrapped sample. These two techniques are responsible for eliminating the problem of highly correlated trees. This in turn lead to lower variance in the estimate of the target variable leading to higher prediction accuracy. As discussed by Lewis (2000), bagging is limited by the method of tree-pruning and consequently more important predictors are much more common than less important ones. By choosing a random subset of predictors for each iteration, more predictor variables are included after growing B trees. This allows for less important variables to contribute towards the output of the target variable. Each tree results in a predicted response. Let $(X_1, y_1), ..., (X_N, y_N)$ be the predictor and target variables in the training set respectively. For classification problems, the most commonly occurring class for an observation, m, with predictor values X_m, is the averaged predicted response $\hat{y}_m$.

For the random forest algorithm (1) to perform optimally, there exist some important hyper-parameters that need to be determined. The number of trees to be grown denoted by *ntree* indicates how many trees are to be grown. m denotes the number of predictors randomly chosen at each node of a tree as described at step B)i) in algorithm 1. The minimum node size which describes the depth of each tree is denoted by *nodesize*. As indicated by previous research (Breiman (2001), Cutler, Cutler, and Stevens (2012),Probst, Wright, and Boulesteix (2019) among others) for a random forest classifier to perform optimally it is crucial to find the optimal combination of values for these parameters. Following Hastie et al. (2005) and Probst et al. (2019) in their methodology the out-of-bag error is used for comparing different values. Out-of-Bag error minimization is discussed further in section 5.

Cutler et al. (2012) along with others show how the random forest algorithm performs much better than bagging in terms of prediction accuracy (Prasad, Iverson, and Liaw (2006),Strobl, Malley, and Tutz (2009)). In fact, Cutler et al. (2012) show that random forests do not overfit as opposed to bagging and this property makes the algorithm suitable for a large variety of applications. Over-fitting occurs when the mapping function is closely fit to a set of limited data-points. This implies that changing the training set results to great differences in estimated values. The random forest by construction manages to overcome this problem.

In this particular study, we do expect to have some imbalance as the ratio of no goals to goals is expected to be large. The target variable y is expected to be highly imbalanced due to the low-scoring nature of the game. More details about the imbalance in the particular data-set can be found in Section 3. In such cases, training algorithm 1 would produce results skewed towards the most populated class, in our case, shots that were not converted to a goal. The classifier would essentially classify a new observation with ease to the majority class and predictive accuracy for the minority class would be significantly low(Chen, Liaw, Breiman, et al. (2004)). Section (4.5) gives an overview of the methods used to account for it.

4.5 Random Forests for Imbalanced Data-Sets

Imbalanced data-sets are common in many classification problems and several methods have been proposed to account for imbalanced data sets (Yang and Wu (2006), Sun, Wong, and Kamel (2009)). Imbalanced data-sets can be defined as those in which the qualitative target variable exhibits high imbalance in the frequency of its classes. Imbalanced data has been found to severely impact the accuracy for the vast majority of classification algorithms including Random Forests (Galar, Fernandez, Barrenechea, Bustince, and Herrera (2011)).

The objective of a random forest algorithm (1) is to minimize the overall classification error rate defined by:

$$C.E = \frac{\sum_{i=1}^{N} I[y_i - \hat{y_i}]}{N} \times 100 \tag{5}$$

Where I is an indicator variable, and $\hat{y_i}$ is the predicted response for the i^{th} observation. For this study, $y_i \in (0,1)$ is a binary variable.

In the case of imbalance in the target variable, y_i, the overall classification error rate (5) can be minimized by predicting all observations to belong to the majority class (Sun et al. (2009)). Obviously, this is not desired if we want accurate predictions for the minority class.

The rest of this section, describes modified Random Forests (RF) algorithms which take the imbalance into account and try to improve the prediction accuracy in the minority class.

4.5.1 Random Forests - Quantile Classifier

A relatively new study (O'Brien and Ishwaran (2019)) proposes an efficient and intuitive way to deal with imbalance. To be able to grasp the alteration from the original RF algorithm (1), firstly we need to define the criterion by which the estimated probability for each observation x results to the assignment of the class. Each tree constructed in algorithm (1) outputs a probability for each observation x and in turn that probability decides in which class it should be classified. Random Forests use the Bayes classifier (Berrar (2018)) for the aforementioned decision:

$$\delta_B(x) = \mathbf{1}_{\left[p(x)] \geq \frac{1}{2}\right]} \tag{6}$$

Here, $p(x)$ is defined as the conditional class probability of the data with $p(x) = \mathbb{P}\,(Y = 1|X = x)$. Equation (6) classifies an observation x to class $\mathbf{1}$ if the conditional class probability is greater or equal to $\frac{1}{2}$. This is also called the median classifier.

Considering a highly imbalanced data set, the Bayes rule (6) is inclined to classify most observations in the majority class(Cutler et al. (2012)). This occurs because the estimation of the random forest will be skewed towards the majority class. Of course this favours the correct classification of the majority class but hinders the accuracy of prediction for the minority class. In cases where accurate prediction of the minority class is important, we have to use alternative classifiers which manage to take imbalance into account. O'Brien and Ishwaran (2019) construct a method in which the sum of true positive and true negative rates are both maximized.

This is achieved by using a density based approach with the goal to maximize both the majority and the minority class accuracy. The only difference with the normal random forest algorithm is that we change the Bayes criterion given in (6) and it becomes the following:

$$\delta_D(x) = \mathbf{1}_{\left[f_{(X|Y)}(x|1) \geq f_{(X|Y)}(x|0)\right]} \tag{7}$$

Where $f_{(X|Y)}$ represents the conditional density of the predictors given the value of the target variable. The classifier in (7) assigns an observation x to the class for which the conditional density $f_{(X|Y)}$ is higher. In that sense, basing the decision boundary on the conditional density of the explanatory variables, removes the effect of the prevalence of the minority class labels. Most importantly, as proved in their study (O'Brien and Ishwaran (2019)), using classifier (7) maximizes the accuracy for both the majority and the minority class. Define a risk function applicable to binary problems for a classifier $\hat{\delta}(x)$ by:

$$r(\hat{\delta}, \lambda_0, \lambda_1) = \mathrm{E}[\lambda_0 I_{(\hat{\delta}(x)=1|Y=0)} + \lambda_1 I_{(\hat{\delta}(x)=0|Y=1)}] \tag{8}$$

The parameters λ_0 and λ_1 are always above 0 and denote the importance weight, or penalty, of mis classification; λ_0 is the weight when $Y = 1$ is predicted and the actual response is $Y = 0$, λ_1 is the penalty when $Y = 0$ is predicted when the actual response is $Y = 1$. I denotes an indicator function which is one if its condition holds. Note that if the weights are equal $\lambda_0 = \lambda_1 = 1$ the risk function simplifies to mis-classification error defined in (5). In relatively balanced sets using the Bayes criterion (6) minimizes the mis classification error. In their work O'Brien and Ishwaran (2019) show that for highly imbalanced sets, if the importance weights are chosen given the criterion in 7, then the classifier is able to maximize the joint performance for both classes.

Implementing this modified random forest algorithm, requires to calculate both conditional densities in 7 at each observation x and assign x to the class for which the conditional density is higher. The only difference in algorithm (1) is that in step (3), we substitute the density-based classifier (7) instead of the Bayes classifier.

4.5.2 Balanced Random Forests

Another method to deal with the problem of imbalance is proposed by Chen et al. (2004). It is a simple yet powerful approach which again seeks to maximize both the prediction accuracy in the majority and minority class. It is based on under-sampling the majority class and repeating this procedure. This technique was firstly introduced by Kubat, Matwin, et al. (1997) with selective under-sampling of the majority class.

Chen et al. (2004) argue that with extremely imbalanced data sets, there exists a significant probability that in step 1A of algorithm 1, the bootstrapped sample consists of only a minor number of observations from the minority class. Training of the model on minority class examples is thus minimal, reducing the accuracy in minority class predictions. The Balanced Random Forest algorithm (BRF) as described by Chen et al. (2004) is given in algorithm (2).

Algorithm 2 is a repeated down-sampling procedure. In step 2A, instead of selecting a bootstrap sample from the entire data-set, each tree is constructed by selecting a bootstrap sample from the minority class, or less frequent stratum, along with an equal size sample from the majority class, with replacement. Xie, Li, Ngai, and Ying (2009) and Chen et al. (2004) show that for large enough data-sets, the BRF algorithm can provide sufficient prediction accuracy, in both strata, after a small number of repetitions. This is again dependent on each data-set as highly heterogeneous data-sets are expected to need more repetitions or re-samplings to get reliable estimates.

Galar et al. (2011) and Chen et al. (2004) indicate that BRF can be computationally efficient with a sufficiently large data-set . In that case, the less intensive strata is represented with enough information such that the training algorithm can already provide good class predictions with a low number of repetitions, S. Additionally, the needed number of repetitions becomes lower in cases where the classes of the target variable are well separated by the input characteristics, X.

In cases of imbalanced data sets, the idea of repeatedly re-sampling and building a tree for each random subset of equal class proportions seems ideal for training a more precise algorithm. The Bayes criterion employed in random forests, is shown to work better when imbalance in the data is not high.The equal sized classes along with the

```
Result: Balanced Random Forest for Classification
1. For b=1,..B;
    A) Draw a bootstrap sample from the minority class equal to the number of minority cases. Randomly
    draw the same number of cases for the majority class, with replacement. ;
    B) Induce a decision tree ($T_b$) on the cases from A, by following the below steps for every terminal node,
    until the minimum node size is reached ;
        i) Select at random $m$ out of the $p$ available co-variates ;
        ii) Use the Gini criterion (2) to find the optimal split and optimal variable ;
        iii) Split the node in two resulting nodes ;
2. Output the ensemble of Trees $T_b$ for b =1 to B. ;
3. For each tree, classify a new observation $x$ using the Bayes Criterion ;
4. Using the output from (3), classify $x$ using the majority vote rule from all B trees. ;
5. Aggregate the predictions made from the $S$ constructed random forests for each observation $x$
```

Algorithm 2: Balanced Random Forest - Classification

idea of averaging over a large number of trees assist in better prediction of the target variable. Essentially, this method builds a decision tree on B balanced subsets of the whole data, eliminating the imbalance which is originally expected to create problems due to optimally classifying most observations in the majority class.

By iteratively re-sampling from the majority class, we want to gain as much information as possible while keeping the subsets balanced. Something which is interesting to observe is how much information is actually gained by continuously drawing sub-samples from the majority class. We expect that if the cases in the majority class are heterogeneous and do not exhibit strong patterns, more sub-samples are beneficial and give more information for the final classification. On the other hand, if the characteristics of the majority class are close, it seems that different balanced subsets for each tree might not be necessary. For that, I also construct a balanced random forest with only one subset of equal sized groups and compare results with the rest of the algorithms. In essence, this is a normal random forest algorithm where we randomly draw a balanced sub-sample from the data.

4.6 Re-sampling methods

An indispensable part of machine learning algorithms is the process of re-sampling. This involves splitting the data-set into the training set, where our algorithm uses all of the data to fit the model and a test-set where we can evaluate the fitted model on data never again seen or trained on. This procedure makes sure that the fitted model can be extrapolated and predictions can be accurate enough. In this section, we describe how we use the prediction error on the test set as a method of comparison between models and for the optimal choice of hyper-parameters.

4.6.1 K-Fold Cross-Validation

As described above, by splitting the data-set into a training and test set we can assess how our model performs to previously unseen data. For example, we can use 80% of the cases for training and the remaining 20% for the test set. The problem here is that our estimate for the test error may be volatile and can change a lot when another random sample split for training and test is made.

To overcome this limitation, we use K-Fold Cross Validation. The idea is simple yet powerful when it comes to estimating the test error. We randomly divide our data-set into K equal sized groups. We then repeatedly fit the model by the desired random forest algorithm to the $K-1$ groups and leave one group each time out to be the test set. This process results to K estimates of the test error. We then average the results of the test error for each repetition by:

$$CV_k = \frac{1}{K}\sum_{i=1}^{K} MSE_i \tag{9}$$

The fundamental idea behind this method, is to achieve more reliable estimates of the prediction error after averaging the results for each hold-out set. (Jung and Hu (2015)). In general, this helps in cases where the data has high variability. If that is not the case, we expect that a single random hold-out set would provide similar results to K-Fold.

For the BRF, it is important that the hold-out set for each iteration is taken from the original data-set and not from the sub-sample used in algorithm (2). This would ensure that the prediction error is based on a test-set representative of the true sample. To achieve that, we firstly split the data-set in K-groups. We expect that on average each set includes the same proportion of goals and not goals. For each iteration of the K-fold we train our data set using sub-samples as described in section 4.5.2 out of the $K-1$ folds. The prediction error each time is calculated on the K^{th} hold-out set.

The above procedure can also be used for other performance measures such as the accuracy of the model in terms of correctly predicting the respective classes. The accuracy for each class and for every different algorithm is calculated using K-fold cross validation.

4.6.2 Out-Of-Bag Error & Optimal Hyper-parameters

The random forest algorithm has some hyper parameters or ”tuning parameters” which decide the performance of the classifier. These need to be optimized in order for the classifier to perform at its optimal. Depending on the data set and the method at hand the values of these parameters need to be tuned for the algorithm to perform at its best. For all different versions of the algorithm described in section 4, the tuning parameters are the same. Table 2 presents random forest parameters to be tuned.

Table 2: Random Forests Hyper-Parameters

Hyper Parameter	Description	Standard Value (Classification Problems)
ntree	Number of trees in each iteration	500, 1000
mtry	Number of randomly selected variables at each node	$\sqrt{p}$
nodesize	Minimum size (observations) for the terminal node	1
sample size	sample size to construct the trees	N
replacement	with or without replacement of observations	With replacement
splitting rule	rule by which each node of a tree is split	Gini index

Researchers have developed numerous methods to find optimal parameters. For this study we use the grid search approach as described by Feurer and Hutter (2019). We define a predetermined set of possible values for the hyper-parameters and the grid search essentially evaluates the Cartesian product for these. That is, for each possible combination of values, a model is fitted and the one with the lower test error is chosen as the optimal. A limitation of this method is that it is computationally expensive as the amount of models needed to be fit grows with each added value for each hyper parameter.

In this study, we do not optimize over the splitting rule, replacement and sample size. The splitting rule used is the Gini index for classification. We use a bootstrapped sample for all methods and the sample size is as described in section 4 for each method. The *ntree* parameter is chosen to be 1000. As argued by Probst and Boulesteix (n.d.) for classification problems 'tuning for number of trees is not necessary' and a high enough number of trees is sufficient to give the desired results.

In this case, the two remaining hyper parameters to be optimized is *mtry* and *nodesize*. Using the grid search approach in our case is relatively not computationally expensive and when compared to other methods, such as random search, can provide a better result due to the evaluation of more possible combinations of values for hyper parameters

The evaluation strategy is to find the combination of values for the two hyper parameters which minimize the misclassification error. To estimate the mis classification error, we use the Out-Of-Bag (OOB) error. In random forests, the sample size chosen for growing a tree does not contain all of the observations as a bootstrapped sample, with replacement, is extracted. As explained in algorithm 1 only m out of p available variables are chosen at each node. This procedure leaves a proportion of rows in the data out when training the model. These left-out examples can be used to form accurate estimates of the misclassification error. In this study, we use the OOB error to decide upon the best hyper parameters for each random forest algorithm. We therefore estimate the OOB misclassification error for each combination of hyper-parameters and choose the combination which minimizes it.

4.7 Evaluation Measures

After tuning our models using the OOB error we want to be able to compare their accuracy using the K-Fold cross validation method. There exist several measures to evaluate the performance of classification but we want to focus on the ones most suitable for the data-set. The most commonly used metric to assess a classification model is accuracy however in the case of imbalanced data-sets, this could be misleading. Additionally, our model selection should be made on a combination of metrics rather than an individual one. Akosa (2017) suggests possible alternatives to 'Accuracy' when dealing with an imbalanced data-set. Table 10 provides a list of the chosen metrics along with their derivations.

$$\begin{aligned}
\text{True Positive Rate (TPR)} &= \frac{TP}{TP+FN} \\
\text{True Negative Rate (TNR)} &= \frac{TN}{TN+FP} \\
\text{Accuracy} &= \frac{TP+TN}{TP+FN+TN+FP} \\
\text{G-mean} &= \sqrt{TPR \times TNR} \\
\text{Precision} &= \frac{TP}{TP+FP} \\
\text{F-measure} &= \frac{2 \times \text{Precision} \times TPR}{\text{Precision} + TPR} \\
\text{Adjusted F-measure} &= \sqrt{F_2 \times invF_{0.5}}\ , \\
F_2 &= \frac{\text{Precision} \times TPR}{(TPR \times 4) + \text{Precision}} \\
invF_{0.5} &= 1.25 \times \frac{\text{Precision} \times TPR}{(TPR \times 0.5^2) + \text{Precision}}
\end{aligned} \tag{10}$$

Table 3: Confusion Matrix

	Predicted Negative Class	Predicted Positive Class
Actual Negative Class	**TN**	**FP**
Actual Positive Class	**FN**	**TP**

The true positive (TP) values denote the cases where both the predicted and actual target variable is a goal. In contrast the true negative values (TN) are those cases where the actual and the predicted target variable corresponds to no goal. As is always the case with classification algorithms, there exists a trade-off between those two values (O'Brien and Ishwaran (2019)). For this study, the aim is to firstly strike a balance between correct predictions of majority and minority class. Of course, we want to be able to better understand offensive efficiency which implies that we are able to sacrifice wrong predictions in the majority class for more accurate predictions in the minority class. The metrics in (10), provide us with the available information to understand which methods work better in that direction.

As discussed earlier, overall 'Accuracy' or misclassification error can be a misleading indicator of performance in cases of imbalanced data. The measures in (10), when combined can act as indicators towards the best model choice given the purpose we employ them for.

The first two measures 'TPR' and 'TNR' measure the accuracy of the positive and negative class respectively. These indicate the direct trade-off which exists between minority and majority class accuracy for each algorithm. As we have a slight preference over better predictions for the minority class we also employ 'Precision', a measure of exactness in the minority group. It evaluates the proportion of correctly classified minority cases out of all cases classified as the minority group.

The next measure we want to employ, is the 'F-measure'. It measures the balance between 'Presision' and 'True Positive Rate' and provide a more realistic measure of a test's performance for the positive cases. Its purpose is to enable us to better understand which model is both accurate and precise in predicting the minority class cases. However the 'F-measure' works better in balanced sets. For an imbalanced data-set, Akosa (2017) suggest that the 'Adjusted F-measure', works with the same idea as the 'F-measure' but can adjust for the imbalance present in the data. Both of the above 'F-scores' reach their best value, meaning perfect precision and recall, at a value of 1. The worst F score, which means lowest precision and lowest recall, would be a value of 0.

Lastly, a measure to better understand the balance between the majority and minority class accuracy is the geometric mean ('G-mean'). A low prediction accuracy for the classification of the minority cases drugs down the overall G-mean value even if the predictive accuracy of the majority class is high.

The choice of the 'best' classification method is based on the combination of the measures in (10). The purpose of this model is to strike a balance between accurate predictions of both classes with a slight preference towards the correct prediction of the minority class.

5 Results

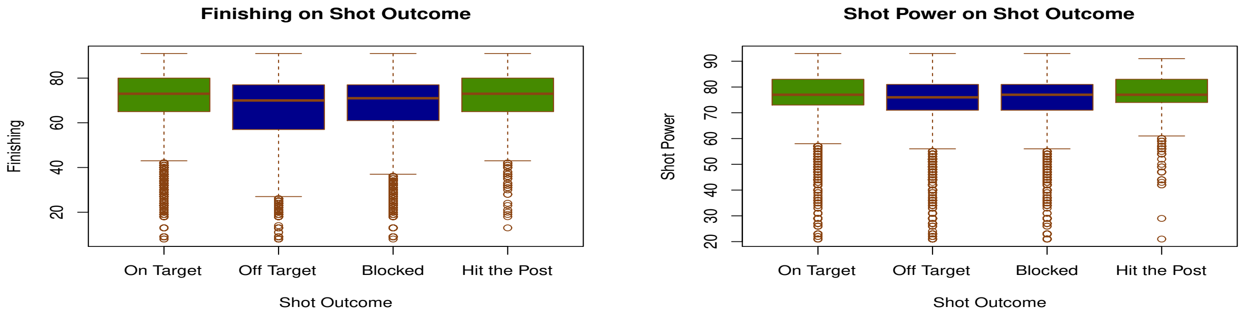
5.1 Exploratory Data Analysis

Prior to employing the methods discussed in section 4, it is important to understand the data at hand and gather some initial insights. In this section, we briefly discuss some preliminary insights which describe the available information. Then we focus on constructed variables and their idea, which are expected to help in the predictive analysis to follow.

The final data-set consists of completed shots from 1, 077 individual players and 71 teams. As noted in section 3, the data consists of 60, 645 attempted shots. Of those, only 6, 037 resulted to a goal. This amounts to a 11% overall conversion rate, implying a highly imbalanced data set as defined in section 4.5. As briefly discussed in section 3.1.1, the location variable is categorical with 19 possible locations of attempted shots. For this study, we want to exclude locations outside the attacking half. The rationale is that given our effort to quantify the quality of a shot, resulting to a more accurate prediction, shots not taken inside the attacking half can contain a fair amount of randomness and could hinder the performance of our classification algorithm. A detailed visualisation of the locations can be found in figure 3. Figure 3 segments the attacking half of the pitch into mutually exclusive location bins as described in the original data-set.

Additionally, 955 shots that come from direct free-kicks or penalties are excluded from this analysis. As argued by sports scientists, studying free kicks and penalties falls in an entirely different category of analysis (Casal, Maneiro, Ardá, Losada, and Rial (2014)). Other attributes related to the quality of a free-kick taker should be taken into account. Additionally a lot of studies focus on the strategic thinking prior to the execution of a free-kick by both the taker and the goalkeeper (Chiappori, Levitt, and Groseclose (2002))

A crucial question worth of investigating is whether and how individual shot characteristics change with respect to shot outcome. Firstly, we want to observe how the different attributes of the shot-taker interact with conversion rate from different locations specified in figure 3. The conversion rates (%) presented in table 15 are interestingly enough, aligned with the expectations one would have. We note that players with high 'Finishing' quality can convert their attempted shots much more frequently rather than players with lower rating on finishing. A 'clinical finisher' is defined as a player which possesses the ability to 'finish' a chance from close range (Bloomfield, Polman, and O'Donoghue (2007)). What is more appealing, is when focusing on different locations, the highest difference in the aforementioned frequency occurs at location 13, agreeing with the theoretical background.



(a) Boxplot with Finishing attribute on y-axis and 4 possible shot outcomes on x-axis

(b) Boxplot with Shot Power attribute on y-axis and 4 possible shot outcomes on x-axis

Figure 2: Boxplots for Shot Outcome

A special category of shots are those which end up hitting the crossbar. Considered as the toughest of luck in the footballing world, we should be careful how we treat those attempts. Many researchers argue that an attempt that ended up hitting the crossbar, might have as well been a goal with some favouring luck (Biswas (2018)). That is, the quality of a shot in terms of precision might be considered as high as one which resulted in a goal. It therefore makes more sense to regard the characteristics of a shot that hit the crossbar similar to one which resulted to a goal.

Figure 2 enforces this opinion as the distribution of shots that ended up as a goal is similar to those that hit the bar when compared to off-target and blocked shots. The data-set has $1,059$ shots which hit the crossbar and were not scored. In the training sets of the classification algorithms used, we change the target variable and regard it as a goal for shots hitting the bar. This is not done in the test set. We expect that such a change would increase the precision of the classification.

5.1.1 Variables Construction

After the initial insights and the suggestions made in the existing literature (Armatas and Yiannakos (2010),Eggels et al. (2016)), we construct a distance measure , traditionally seen as an important factor in shot prediction. For that, we use the centre of each location as presented in figure 3, and measure the Euclidean distance to the centre of the goal. In this way, distance is now a continuous variable that is directly derived from the locations variable.

For any shot taken, the surrounding circumstances can affect its outcome. For example, consider the difference between a friendly game and a crucial league game. It is thought that the 'pressure' felt by the shot taker can affect the outcome of the shot (Anderson (2013),Tiehan (2015)). To account for that, we use data on final rankings of the teams involved in each game and for each shot. The variable created, describes the rank difference between two teams. In a competitive game with two teams being closely matched, the stakes are high and the pressure on the offensive players is greater. We expect to capture that, with higher values indicating less pressure and consequently enhanced probability of scoring.

Tiehan (2015) argue that 'pressure' is also present at the closing stages of a game. last 10 minutes, which is still 'closely matched'. Closely matched can be defined as 1 or 0 goal difference between the two competing teams. For that, we construct a binary variable which equals to 1 when both the timing and the 'closely matched' conditions are met and 0 otherwise.

Lastly, combining the data set from FIFA which indicates preferred foot for each shot taker, and the Events data which contains information on which foot was used to execute the shot, we create a binary variable which is 1 when the shot was taken with the preferred foot and 0 otherwise.

5.2 Performance Evaluation

As described in section 4 prior to training the algorithms and evaluate their performance, we need to estimate the optimal parameters for each one. The number of random variables chosen at each iteration of the algorithm is the most important parameter to tune. As argued by Hastie et al. (2005), the optimality of *mtry* achieves the best possible trade-off between low correlation between trees and low variance for the predictions. The results are given in table 4. The number of trees is constant for all methods and is set to ntree $= 1,000$.

Table 4: Optimal Hyper-Parameters

Method	mtry	nodesize
Random Forest	2	10
QRF	6	7
BRF	5	2
MBRF	6	1
BRF (one subset)	2	5

The hyper-parameters derived with OOB error estimation in table 4, are then employed to train the respective algorithms. For logistic regression there are no hyper-parameters to optimize. Instead we estimate the β parameters using the maximum likelihood method. We do not discuss in more depth the coefficients estimated in logistic regression and their possible interpretations, as the focus of this study is not the causality of each explanatory variable but the overall performance of each algorithm in striking a balance between majority and minority class predictions.

Table 5: Performance Comparison

Method	TNR	TPR	G-mean	Precision	F	Adj - F	Accuracy
Logistic (Full Data)	0.99	0.26	0.51	0.73	0.38	0.11	0.91
Logistic (Balanced Subset)	0.74	0.58	0.6	0.17	0.2	0.12	0.62
Random Forest	0.98	0.18	0.42	0.45	0.26	0.11	0.895
BRF	0.8	0.61	0.7	0.25	0.31	0.13	0.78
Random Forest (Balanced Subset)	0.67	0.6	0.61	0.19	0.23	0.13	0.64
QRF	0.68	0.71	0.69	0.2	0.3	0.14	0.67

Table 5 compares the performance of the methods described in section 4. To estimate the values as described from the confusion matrix (21), we use 10-Fold cross validation as previously described (4.6.1). The choice of $K = 10$ is arbitrary, but in general a higher value for K provides less bias towards estimating the true value of the test error (James et al. (2013)). Interestingly, comparing the values in table 5 derived after 10-Fold cross validation with randomly selecting a single test set consisting 10% of observations gives very small and fairly insignificant differences. This hints towards low variance within the data-set as suggested by Raschka (2018), as more estimations of the prediction error do not contribute to more accuracy. For further confirmation of the above finding, we also run an experiment with randomly choosing one test set of 10%, and run the models for a number of times. The results were again similar to those in table 5 further enforcing the above notion.

The confusion matrices for the methods used can be found in the appendix. In the next section we discuss the findings presented in table 5 and provide possible interpretations for each algorithm's performance.

6 Discussion & Applications

The imbalance algorithms employed all manage a significant improvement over the normal random forest algorithm and the benchmark logistic regression in accurately predicting the minority class. In this section, we discuss the possible interpretations of the performance comparison. Then we use the model of choice and provide some applications in which football clubs could utilize such methods to better understand underlying patterns in their offensive capabilities.

6.1 Discussion

By construction, the random forest algorithm tries to minimize the Gini-coefficient (2). The objective is to minimize the misclassification error which is equivalent maximizing the overall accuracy of the model (Accuracy = 1 - Misclassification error). As we argued in section 4, in cases of imbalanced sets, this does not help in achieving good performance for the minority class as the model classifies as many observations to the majority class as possible. This is evident in table 5, as the random forest algorithm manages high accuracy in predicting the majority class but really low for the minority class. This is also the case with the logistic regression on the whole sample. Due to the large number of cases classified to the majority label, the accuracy of the model is higher compared to all others. Conversely, the low number of cases classified to the minority group give high precision compared to all others.

Something we expected prior to the analysis seems to be true as both the logistic and the random forest regression without any adjustments for imbalance give similar results. 'Precision' value is much better for logistic when compared to the random forest implying less false positive predictions with the logistic function.

Comparing the two algorithms which only use one balanced subset of the data we note that the random forest performs better in accurately and precisely predicting the minority class when compared to the logistic regression. Additionally the values for 'G-mean' and 'Adjusted F' are close, but still slightly better for the random forest method, giving the random forest method the edge when it comes to striking a balance between majority and minority class predictions. This again indicates that the explanatory variables exhibit a non-linear relationship towards the target variable.

As we previously discussed in section 4, we employ modified algorithms which account for the imbalanced data-set at hand. QRF manages to strike a balance between the majority and the minority class prediction without using sub-samples based on the population of the minority class (Meinshausen (2006)). Interestingly, the accuracy when predicting the minority class is slightly higher compared to the majority class.

Comparing BRF with QRF the latter outperforms the former in the majority of the employed metrics. Notice that BRF outperforms the QRF when it comes to predicting the majority class. As suggested in the relevant paper (O'Brien and Ishwaran (2019)), we can see that the accuracy of the majority class and minority class are similar, further confirming the notion of balancing the two accuracy rates as suggested by the authors. The variable of interest, in this case converted shots, is predicted far better by the QRF algorithm.

A possible explanation for better performance of the QRF is that of similar conditional probability distributions for the features space. O'Brien and Ishwaran (2019) define an observation in the minority class to be 'rare' if 4 or 5 of its nearest neighbours, measured by mahalanobis distance, come from the majority class. Otherwise the observations are 'safe'. The QRF is shown to outperform the BRF for data-sets where a large percentage of minority cases are defined as 'rare' or more simply the two groups are not easily distinguishable. This can be seen in the conditional distributions of the feature space for the two classes as $f(x|0) = f(x|1) = f(x)$ almost everywhere in the feature space. If a large part of observations are 'rare', it is then implied that data from both classes is important to better estimate the algorithm's split points. In that sense the QRF would perform better than BRF. After measuring the mahalanobis distance for all minority cases we find that 83% of the observations are 'rare'.

The mechanism behind BRF is repeatedly drawing random balanced samples from the original data. It is interesting to understand whether that is beneficial. For that we use one balanced subset and employ the random forest on it. We also do the same for the logistic regression. As expected the true positive rate is for both much better when compared to their full data counterparts. When comparing the BRF with one balanced sample random forest there is no big improvement over the classification of the minority class. This can be attributed to the fact that repeated sampling in BRF is done with the same samples for the minority class. The improvement of repeated sampling is visible in the true negative rate. As we draw random sub-samples from the majority class, it is expected that more sub-samples contain more information for the majority class, therefore leading to more accurate predictions.

The confusion matrices provided in the appendix (8), indicate that all algorithms, except the normal random forest, over-predict the minority class. On one hand, this improves the accuracy for the minority class, but also would result in an inflated number of expected goals when classifiers are actually applied.

Concluding on what is said above, we suggest that the QRF algorithm works best towards the scope of the analysis. The combination of the employed measures indicates that the QRF manages to better account for the trade-off in predictions of majority and minority class, while having the best prediction accuracy for the minority class. For the following section we use the QRF algorithm to further understand how it could be applied for better decisions in the football industry.

6.2 Applications

After evaluating the methods described in section 4, it is of interest to understand some applications where our chosen model can prove valuable towards football clubs. Our aim is to use the model and understand the offensive effectiveness of both individual players and teams. We also want to compare the probability output from the model to the standard descriptive statistics like shots taken, or shots conversion.

6.2.1 Individual Player Performance

A benefit of the classification algorithm used is that for each shot taken, we can output an associated probability of it resulting to a goal. To assess an individual player we can estimate the expected goals out of all shots taken and have a complete analysis of the effectiveness of that player.

Table 6: Top and Lower Expected Goals Players

Player	Shots	Goals	Shot Conversion	Expected Goals
Lionel Messi	711	127	0.178	139.24
Cristiano Ronaldo	904	109	0.121	123.7
Robert Lewandowski	592	101	0.171	115.6
Karim Benzema	357	68	0.19	82.63
Sergio Aguero	400	63	0.158	79.1
...	...	...	...	...
Clemens Fritz	84	2	0.0238	5.67
Johannes Geis	178	2	0.0112	6.28
Juanfran	78	1	0.0128	7.14
Pascal Gross	94	1	0.0106	8.9
Marcel Schmelzer	130	1	0.0076	3.48

In table 6 the top and lowest five players as indicated by expected goals are shown. For the comparison to be rational, we only keep players which are consistent shooters. We only present the results for players who attempted 70 or more shots. The shot conversion column is calculated by goals by total shots.

The first thing to notice is that the ranking of players arranged by expected goals is not the same as it would be if in the case of shot conversion. This shows that using the Expected Goals metric is different than shot conversion. The attributes that contribute to 'Expected Goals' evaluate how many goals each players should have scored given the aspects described in section 3.

Evidently in table 6 'Expected Goals' value is higher than actual goals scored for all players. At first this seems counter intuitive as it would be rational to expect that high-quality players would score more goals than they are expected. The explanation lies with the modifications made to the normal random forest algorithm and as evident in the confusion matrix for QRF in the appendix (8), the classifier tends to over-predict observations for the 'Goal' class. Thus we do expect that the expected goals are inflated when compared to the actual goals scored.

In that sense, another metric has to be constructed to provide a benchmark in evaluating offensive ability. To overcome this problem we use a 'Shot Efficiency' metric, which is essentially the expected goals per shot for each player. This is calculated by dividing the 'Expected Goals' column with number of shots. The results are presented in table 7.

Table 7: Top and Lower "xG/Shot" Players

Player	Shots	Goals	Shot Conversion	Expected Goals	xG per Shot
Luis Suarez	185	44	0.237	54.12	0.268
Diego Costa	209	46	0.22	58.19	0.256
Mario Gomez	162	35	0.21	41.6	0.244
Claudio Pizarro	187	36	0.19	42.63	0.238
Karim Benzema	357	68	0.19	82.63	0.192
...	...	...	...	...	...
Ricardo Rodriguez	84	2	0.0238	5.67	0.025
Tom Huddlestone	178	2	0.0112	6.28	0.023
Pascal Gross	94	1	0.0106	8.9	0.02
Johannes Geis	178	2	0.0112	6.28	0.012
Marcel Schmelzer	130	1	0.0076	3.48	0.012

Table 7 presents the first and last players ranked by expected goals per shot. The players vary a lot when compared to table 6, especially for the first 5 spots. The 'xG per shot' metric indicates the probability of scoring for each player given the past attempted shots. Differences can occur because of the player's ability to shoot, but also decision making at the time of the shot. The exhaustive list still ranks the top class players high enough as expected but now players in the mid-range in terms of ability make their appearance in the list. A possible reason is that top quality players have the freedom to try shots at a higher rate than others. Table 6 indicates that. As football is a low scoring game and the probability of scoring is on average much lower than not, it is rational that with more shots executed, the expected goals per shot actually drops. This is evident in all top quality strikers found in table 6. It is though interesting to examine possible patterns which constitute the expected goals per shot probability.

Ranked number 8 in expected shots per goal is a not so well known player, especially up to 2017 where we draw our information, "Julian Brandt". The left midfielder played for "Bayern Leverkusen" from 2014 to 2018 and ranks higher than top class players like 'Ronaldo' and 'Messi'. To further understand possible reasons for that we should more closely examine the characteristics of the player's shots.

The shooting qualities associated with the player are at an average level compared to other midfield and offensive players in the data-set. The first thing we observe is that approximately 21% of the player's shots came from position 9 as indicated in figure 3. Position 9 has an overall shot conversion of 7.6%, lower than the total average of shots converted (9.6%). 'Brandt' has an astonishing conversion rate of 31.7% for the same position. In terms of tactics this clearly indicates that given his starting position, he is successful closing in towards goal and shooting from the left angle inside the box. Another aspect which contributes to his high expected goals per shot ratio, is the player's ability to shoot with his 'weaker' foot. The player scored 28% of his goals with his weaker left foot. A characteristic which is not there for the vast majority of the players examined, as of all the goals scored, only 13% were scored with the weaker foot.

It is worth mentioning here, that with the start of the 2019-2020 season Julian Brandt was acquired by German rival club "Borussia Dortmund". After the end of 2017, where the data available for this study ends chronologically, the player's transfer value saw an increase of 150%.

Another such example is that of "Sadio Mane" a right offensive winger who now plays for one of the most successful football clubs, "Liverpool". Most of the shots executed from the player (90%) are recorder from his time in the English club "Southampton". Examining the characteristics of those shots, the shooting ability of the player is above average. The location where the shots are taken align with the average for each location, so nothing significant can be found in that aspect. An aspect that stands out is the placement of the shots. 26 out of a total of 31 goals scored, were placed in the top or bottom corners, considered as the most difficult shots to be defended by the opposing goalkeeper. This is significantly higher than the average rate of successful shots placed in the corners. This characteristic stands out the most for the player, and is most probably what influences the expected goals per shot rate to be at a relatively high value.

The examples in the previous paragraphs, indicate a way in which football clubs could utilize the constructed metric, to evaluate offensive players, understand the underlying patterns of good or bad offensive efficiency and act accordingly.

6.2.2 Football Results Prediction

Another possible application of determining the probability of each opportunity resulting to a goal, is being able to predict the number of goals in a single game. It would be beneficial to examine whether our chosen model can accurately predict the number of goals for games in which the data was not used to train the model. In that aspect, this application is different than individual players evaluation. Additionally, we want to understand if an expected goals metric is useful as a statistic to describe the overall offensive effectiveness of a team in a 90-minute match when compared to traditional descriptive statistics like possession and total attempts on goal. Expected goals are again calculated as the sum of the output probabilities of shots executed by each team.

Table 8 presents the standard statistics given at the end of each game regarding offensive efficiency along with the expected goals metric. For this game the expected goals manage to accurately portray the result of the game and consequently the offensive performance of each team. This is also evident in the total attempts for each team. On one hand the metric manages to accurately predict the result of this game, however when it comes to offensive efficiency, it does not tell us much more than the standard statistics used. It is much more interesting to look at cases where the shots on target can not solely describe the result of the game.

Table 8: Everton vs Manchester United 20/04/2014

	Everton	Manchester United
Possession	39%	61%
Shots	9	17
Shots on Target	2	5
Expected Goals	0.4468	2.181817
Actual Goals Scored	0	2

A more peculiar example is presented in table 9. Possession is higher for the home team and the total attempts on and off target weigh greatly towards the home team again. These not only hint towards a win for the home team but a fair gap in goals scored too. The expected goals metric though covers a lot of the imbalance and one thing which prevails is that the quality of the taken shots is low given that 13 shots in total result to an expected value of approximately 1. Also, by looking at each of the output probabilities for each shot and using the random forest default cutoff point of 0.5, no shot scores a higher probability than that, resulting to a prediction of no goals for both teams.

Just inspecting at the total shots, is misleading in understanding the attacking danger posed by the team. In that sense, the expected goals metric here proves to be a valuable metric to better understand the quality of shots taken. In the case that expected goals still predicted a wide margin win for the home team, then other causes could have been in play like good goalkeeping from the away team or just an unlucky day for the home team.

Table 9: TSG 1899 Hoffenheim vs Eintracht Frankfurt 26/04/2014

	TSG 1899 Hoffenheim	Eintracht Frankfurt
Possession	58%	42%
Shots	13	1
Shots on Target	7	0
Expected Goals	1.343424	0.105
Actual Goals Scored	0	0

The example in table 10 is chosen to be presented to further evaluate the power of the expected goals metric in cases where the final scoreline goes against all descriptive statistics. The benefit in such cases lies with understanding whether the attacking ability of the team having most of the chances is responsible for not delivering or whether other factors are more important. On average it is expected that the team having most opportunities and shots on target gets the win, but this is not a convention that needs to be true in all cases.

Table 10: West Brom vs Watford 16/04/2016

	West Brom	Watford
Possession	57%	43%
Shots	17	11
Shots on Target	10	5
Expected Goals	0.754	0.897
Actual Goals Scored	0	1

In table 10, the numbers directly observed in the game, are all in favor of the home team, and would otherwise indicate a better performance resulting in a win for 'West Brom'. However the actual score does not reflect the opportunities that each team created. A big advantage of the metric comes in play in these kind of situations. It gives us the ability to evaluate whether the total shots taken are indeed a fair representation of performance with other factors, like pure luck, affecting the outcome, or whether the shots taken were not 'dangerous' for the opposing site. For the latter we expect a relatively low probability of conversion. Despite the vast inequality between created chances, we see that the expected goals prediction reflects more accurately the actual score. This indicates that the shots taken by the home team, even greater in number, had a lower expected probability when compared to the away side's shots.

In this case, the usefulness of the metric to accompany the actual goals scored and shots attempted is illustrated. The information which is contained in the metric give a more accurate reflection in terms of efficiency in offense. Of course, other factors like defending efficiency, tactical discipline and others are not taken into account and can also play a significant role for the scoreline. Still we can get a good evaluation towards decision making when it comes to the 'finishing touch' and can avoid misleading summary statistics in those situations.

7 Conclusions

This paper focuses on evaluating different modifications of the random forest algorithm to account for class imbalance. Obviously there are more modifications that could have been used and tested, like the Weighted Random Forests (Chen et al. (2004)), over-sampling of the minority class (Bhagat and Patil (2015)) among others. Other methods could also be tested for the same data-set with a different parameter setting. For the ones observed and for the scope we focus, the Quantile Random Forest approach works best.

The feature space we use is satisfactory for a good expected goals prediction, however other features could have been helpful. As described in section 1 football is described as a dynamic team sport, perhaps the main characteristic which makes it so popular. Spatiotemporal variables like speed of the ball at the time of the shot would undoubtedly be important in better accuracy of predictions. Additionally defender proximity or a measure of how many defenders are around the ball at the time of the shot are also missing. Lastly, location is given in categories (3) thus all shots executed from a category have the same unconditional mean. This is not realistic as the positions within each category differ. This could have been avoided with the use of continuous data for the position of the shot in the (x, y) plane and out of that accurate information about another important factor, the angle to the goal, could have been extracted.

This study, with the use of a variety of random forest algorithms, constructs an expected goals metric and provides some ideas about the possible applications this could have for football clubs. The first step was merging the data-sets so that we can utilize information on offensive and defensive abilities of the teams in a game, since the probability of conversion heavily depends on those abilities. Then exploratory research revealed valuable insights about further factors that could be constructed. Using the re-sampling methods discussed in section 4, we derived the optimal hyper-parameters and evaluated their performance.

The output probabilities for each shot in the data-set were used to construct an expected goals metric which can be used for a variety of applications by sports analytics related firms and football clubs. Individual player analysis is one of those, as we can understand more about the opportunities created for the player at hand, and evaluate his efficiency in front of goal. We then also described how the constructed metric can be used as a supplementary statistic to actual goals scored in a game, to further understand the quality of chances and offensive ability of a team.

In section 6.2.1 we briefly discussed how we could understand more underlying factors about the expected goals value with investigating the placements of the shot; bottom corners, centre, top corners. Something which would potentially be interesting for future research is to further understand the probability of a shot resulting to a goal given the shot placement and the angle at which the shot is executed. Given the characteristics of the shot, study the expected probability for converting for each different placement. This would require the variables that we indicated above and a large enough data-set such that for all classes of shot placement there exist sufficient cases.

8 Appendix

Table 11: Contextual Data

Variable	Description
id_game	Unique identifier of game
Date	Date of game
League	Club League
Season	Year Played
Country	Host Nation of League
fthg	Full-time home goals
ftag	Full-time Away goals

Table 12: FIFA-Attackers Attributes related to Shooting

Variable	Description
Shot Power	Player Attribute that determines the strength of a player's shootings
Long Shot	Player Attribute that determines a player's accuracy for the shots taking from long distances
Finishing	Player Attribute that determines the ability of a player to score
Volley	Player attribute which determines the ability of a player to instantly score from a high-pass
Composure	Player Attribute that determines a player's state of being calm and controlling during important matches

Table 13: FIFA-Opponents Defending Attributes

Variable	Description
Defence Rating	Team Attribute that determines the overall defensive capabilities of the opponent
Goalkeeper Positioning	The goalkeeper's ability to position himself correctly when saving shots.
Goalkeeper Diving	The goalkeeper's ability to make a save whilst diving through the air.
Goalkeeper Handling	The goalkeeper ability to cleanly catch the ball and hold on to it.
Goalkeeper Reflexes	Determines how quickly the goalkeeper reacts to a shot on goal.

Table 14: Descriptive Statistics on Shooting Quality

	Shot Power	Finishing	Long Shots	Composure	Curve	Volleys
No. > 80	150	52	42	158	66	29
No. < 80	705	803	813	747	789	826
Mean	75.5	67.7	68.7	74.58	67.09	65.77
Min	21	8	9	32	7	8
Max	93	91	88	94	90	93

Table 15: Shot Location and Conversion

Percentages of conversion (x%:y%). Each x element in the table is the percentage of conversion for players with an above 80 attribute rating (attribute on horizontal axis). Each y element gives the same percentage for attribute rating below 80.

Shot Locations	Shot Power	Finishing	Long Shots	Composure
18	**9.1% : 0%**	6% : 0%	**15% : 0%**	7% : 0%
15	3.4% : 2.6%	4.4% : 2.6%	4.6% : 2.6%	3.6% : 2.6%
9	7.8% : 6.4%	9.5% : 6.5%	8.5% : 6.6%	8.4% : 6.3%
11	8.2% : 6.6%	9.1% : 6.6%	11% : 6%	8.3% : 6.7%
3	17% : 14%	19.2% : 13.9%	17.8% : 14.7%	17% : 14%
10	20% : 20%	22.8% : 20%	22% : 20.5%	20% : 20%
13	53.6% : 47.8%	**57.9% : 46%**	62.9% : 48%	**55.5% : 48%**
12	19.8% : 18.5%	20.8% : 17.3%	15% : 18%	18.6% : 18.9%

Figure 3: 13 Possible Shooting Locations

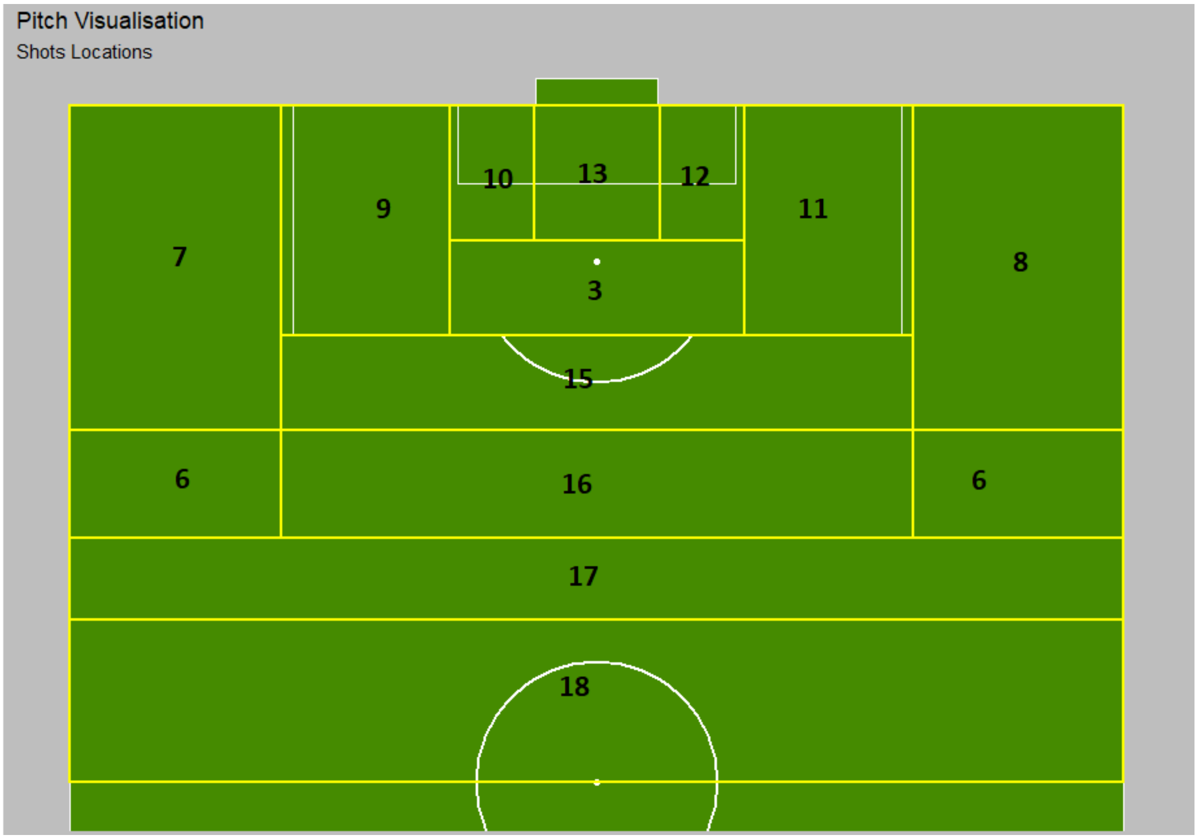


Table 16: Locations

Location	Shots	Goals	Convergence
3	19755	3116	0.16
6	373	8	0.021
7	715	67	0.094
8	654	53	0.081
9	5378	380	0.071
10	960	200	0.21
11	5136	366	0.071
12	920	180	0.19
13	1876	947	0.50
15	23939	703	0.029
16	451	10	0.022
17	435	6	0.014
18	53	1	0.019

Table 17: Confusion Matrix - Random Forest

	Predicted Negative Class	Predicted Positive Class
Actual Negative Class	7605	195
Actual Positive Class	714	160

Table 18: Confusion Matrix - QRF

	Predicted Negative Class	Predicted Positive Class
Actual Negative Class	5603	2588
Actual Positive Class	270	636

Table 19: Confusion Matrix - BRF

	Predicted Negative Class	Predicted Positive Class
Actual Negative Class	6545	1646
Actual Positive Class	362	544

Table 20: Confusion Matrix - MBRF

	Predicted Negative Class	Predicted Positive Class
Actual Negative Class	6532	1659
Actual Positive Class	283	551

Table 21: Confusion Matrix - BRF_1

	Predicted Negative Class	Predicted Positive Class
Actual Negative Class	5717	2678
Actual Positive Class	295	635

References

Akosa, J. (2017). Predictive accuracy: a misleading performance measure for highly imbalanced data. In *Proceedings of the sas global forum* (pp. 2–5).

Anderson, J. (2013, 04). Competitive balance in european football.

Armatas, V., & Yiannakos, A. (2010). Analysis and evaluation of goals scored in 2006 world cup. *Journal of Sport and Health Research*, *2*(2), 119–128.

Armatas, V., Yiannakos, A., Papadopoulou, S., & Skoufas, D. (2009). Evaluation of goals scored in top ranking soccer matches: Greek "super league" 2006-07. *Serbian Journal of Sports Sciences*, *3*(1), 39–43.

Berrar, D. (2018, 01). Bayes' theorem and naive bayes classifier.. doi: 10.1016/B978-0-12-809633-8.20473-1

Bertin, M. (2015a). The third to last thing i'll ever write about expected goals. *Michael Bertin Blog*.

Bertin, M. (2015b). *Why soccer's most popular advanced stat kind of sucks.* Deadspin.

Bhagat, R. C., & Patil, S. S. (2015). Enhanced smote algorithm for classification of imbalanced big-data using random forest. In *2015 ieee international advance computing conference (iacc)* (pp. 403–408).

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., & Matthews, I. (2014). Large-scale analysis of soccer matches using spatiotemporal tracking data. In *2014 ieee international conference on data mining* (pp. 725–730).

Biswas, R. (2018). Continuous fuzzy evaluation methods: A novel tool for the analysis and decision making in football (or soccer) matches. In *Continuous fuzzy evaluation methods: A novel tool for the analysis and decision making in football (or soccer) matches* (pp. 1–63). Springer.

Bloomfield, J., Polman, R., & O'Donoghue, P. (2007). Physical demands of different positions in fa premier league soccer. *Journal of sports science & medicine*, *6*(1), 63.

Breiman, L. (1996). Bagging predictors machine learning 24 (2), 123-140 (1996) 10.1023. *A: 1018054314350*.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Casal, A. C., Maneiro, R., Ardá, T., Losada, J. L., & Rial, A. (2014). Effectiveness of indirect free kicks in elite soccer. *International Journal of Performance Analysis in Sport*, *14*(3), 744–760.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, C., Liaw, A., Breiman, L., et al. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*, *110*(1-12), 24.

Chiappori, P.-A., Levitt, S., & Groseclose, T. (2002). Testing mixed-strategy equilibria when players are heterogeneous: The case of penalty kicks in soccer. *American Economic Review*, *92*(4), 1138–1151.

Colin, T. (2013). Goal expectation and efficiency. *https://statsbomb.com/2013/08/goal-expectation-and-efficiency/*.

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. In *Ensemble machine learning* (pp. 157–175). Springer.

Efron, B., & Tibshirani, R. (1985). The bootstrap method for assessing statistical accuracy. *Behaviormetrika*, *12*(17), 1–35.

Eggels, H., van Elk, R., & Pechenizkiy, M. (2016). Explaining soccer match outcomes with goal scoring opportunities predictive analytics. In *Mlsa@ pkdd/ecml.*

Fairchild, A., Pelechrinis, K., & Kokkodis, M. (2018). Spatial analysis of shots in mls: A model for expected goals and fractal dimensionality. *Journal of Sports Analytics*, *4*(3), 165–174.

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3–33). Springer.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463–484.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Jung, Y., & Hu, J. (2015). Ak-fold averaging cross-validation procedure. *Journal of nonparametric statistics*, *27*(2), 167–179.

Kidd, R. (2018). *Soccer's moneyball moment: How enhanced analytics are changing the game.* SportsMoney.

Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml* (Vol. 97, pp. 179–186).

Kumar, G. (2013). *Machine learning for soccer analytics* (Doctoral dissertation). doi: 10.13140/RG.2.1.4628.3761

Lewis, R. J. (2000). An introduction to classification and regression tree (cart) analysis. In *Annual meeting of the society for academic emergency medicine in san francisco, california* (Vol. 14).

Lucey, P., Bialkowski, A., Monfort, M., Carr, P., & Matthews, I. (2014). quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. In *Proc. 8th annual mit sloan sports analytics conference* (pp. 1–9).

Matthias, K. (2014). What are expected goals? *American Soccer Analysis*.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(Jun), 983–999.

Okabe, M., Tsuchida, J., & Yadohisa, H. (2019, 05). F-measure maximizing logistic regression.

O'Brien, R., & Ishwaran, H. (2019). A random forests quantile classifier for class imbalanced data. *Pattern recognition*, *90*, 232–249.

Pollard, R., & Reep, C. (1997). Measuring the effectiveness of playing strategies at soccer. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *46*(4), 541–550.

Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, *9*(2), 181–199.

Pratas, J., Volossovitch, A., & P Ferreira, A. (2012). The effect of situational variables on teams' performance in offensive sequences ending in a shot on goal. a case study. *The Open Sports Science Journal*, *5*(1).

Probst, P., & Boulesteix, A. (n.d.). To tune or not to tune the number of trees in random forest? arxiv 2017. *arXiv preprint arXiv:1705.05654*.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), e1301.

Rampinini, E., Impellizzeri, F. M., Castagna, C., Coutts, A. J., & Wisløff, U. (2009). Technical performance during soccer matches of the italian serie a league: Effect of fatigue and competitive level. *Journal of science and medicine in sport*, *12*(1), 227–233.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

Rathke, A. (2017). An examination of expected goals and shot efficiency in soccer. *Journal of Human Sport and Exercise*, *12*(2), 514–529.

Riley, M. (2014). A shooting model an explanation and application. *Michael Bertin Blog*.

Ruiz, H., Lisboa, P., Neilson, P., & Gregson, W. (2015). Measuring scoring efficiency through goal expectancy estimation. In *Esann 2015 proceedings of the european symposium on artificial neural networks, computational intelligence and machine learning* (pp. 149–154).

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, *14*(4), 323.

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, *23*(04), 687–719.

Tiehan, L. (2015). Analysis on psychological pressure of college football players. *International Journal of Simulation–Systems, Science & Technology*, *16*.

Xie, Y., Li, X., Ngai, E., & Ying, W. (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, *36*(3), 5445–5449.

Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, *5*(04), 597–604.

Yen, S.-J., & Lee, Y.-S. (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, *36*(3), 5718–5727.