Creation of Soccer Expected Goals (xG) models with distinct non-parametric classification methods

Federico Chung and Alejandro Olaya Gómez

STAT 527, Department of Statistics, University of Washington

Abstract

Modern soccer statistical analysis has developed a lot of metrics and tools to better understand, predict and help coaches make better tactical decisions. New mectrics have been developed, the most famous one being 'Expected Goals'(xG), which models the quality of opportunities created during a match. In this project, we use three different non-parametric classification techniques (logistic regression, random forests and boosting) to create statistical models to estimate xG. We find that given the available data, there is no much difference between the outputs of such methods.

Group Work Description

For the means of this project, the overall structure and main ideas were discussed by both authors. Data cleaning and processing was a joint effort as well as the preparation of the final write-up.

The Random Forest model was created by Federico Chung while the Logistic Regresssion and Boosting models were created by Alejandro Olaya.

The link to the github repository with the used code is here.

1 Introduction

During the last years, the use of sports analytics has been more and more prevalent in professional sports. Although the use of statistical tools has been principally applied to football and baseball, it is known to have grown in other sports as well. The nature of football and baseball as sports where the events take time in more of a "discrete" manner has always made them the perfect topic study for statisticians, as data was widely available. On the other side, sports that are more "continuous" such as soccer and hockey have not attracted much interest until now, as the quality and quantity of data collection process has grown. This phenomenon can be explained by the introduction of tracking systems which have made it possible to obtain a wide array of information for the "beautiful game". During the last few years, one of the most important metrics used in this sport has been the Expected Goals (xG). This statistic parts from the modeling of shots as Bernoulli random variables, where the success is defined as the shot ending in a goal and failure when it doesn't. The xG is the parameter of such random variable, as its expectation (the parameter *per se*) is the probability of scoring a goal. In that sense, the xG's in a match are used to describe the quality of the opportunities players and teams create throughout the game. Plenty of

times, the final score does not reflect the development of the game, so the xG serves as a less biased statistic for what really happened during the game. Our goal is to analyze open shots without blocks in open plays, and predict whether they will result in a goal conversion (1) or not (0) using different non-parametric statistical methods for classification such as logistic regression, random forests and boosting. These methods will come at hand, since the predicted probabilities of scoring a goal can be directly interpreted as the xG.

2 The Data

Data regarding soccer matches is not broadly available at low costs. Big companies such as Wyscouts and Opta (StatsPerform) usually track, generate and sell the information needed to build xG models. Therefore, it is extremely hard to get good and free soccer datasets that track events. Pappalardo *et al.* ([5]) made available a dataset consisting of all the match events for the 2017-18 season for the 5 main leagues in the world: England (Premier League), Spain (La Liga), France (Ligue One), Italy (Serie A) and Germany (Bundesliga). The authors provide the data in json format; the two main datasets used for the project were:

- Events: dataset with the information about every play in every match during the season which includes variables such as type of event, moment in which happened, position in which it happened, etc.

- Players: dataset with the all the relevant information (skilled foot, height, weight, nationality, etc.) regarding all the players that participated in the aforementioned competitions during the 2017-18 season.

The data processing was one of the hardest and most time consuming tasks during the process mainly because of the amount of information worked with. The main data processing duty was to filter every open shot play (shots that were not blocked) and its characteristics

and link it to a previous pass (if existing) and its characteristics. The reasoning behind this is that we believe that shots coming from passes (like headers from a cross) are really different to shots not coming from passes (like taking a chance from outside the area after dribbling an opponent); therefore, knowing if the shot came from a pass was a necessary variable for modelling. Furthermore, the characteristics of the pass can also increase or decrease significantly the probability of scoring a goal in a following shot.

The final dataset used for modelling consisted of 30,862 observations (shots) divided intro three different sets: validation (15%), training (70%) and testing (15%). It is important to notice that the proportion of goals in the data was of 0.1388 and that the split of the data was performed looking that the 3 subsets had similar proportions. The dataset included the following covariates on the open-play shots:

- is goal: binary outcome variable which takes the value of 1 when the shot ended in the net and 0 when it did not.
- matchPeriod: categorical variable indicating if the shot was duringthe first or second period of the match.
- is CA: binary variable that indicates whether or not the shot comes from a counter-attack play.
- league: categorical variable for the 5 considered leagues.
- eventSec2: exact second at which the shot takes place.
- skilled foot: binary variable indicating whether the shot was taken with the player's skilled foot or not.
- x meter: position of the shot relative to the x axis (measured as a perpendicular line to the midfield division line).

Table 1: Distribution of goals by Match Perdiod and Skilled foot

Match Period	Goals	Total Shots	Percentage
1H	1874	14224	13.1749
2H	2411	16638	14.4909

Skilled Foot	Goals	Total Shots	Percentage
head/body	823	6185	13.3064
No	866	5565	15.5615
Yes	2596	19112	13.5831

- y meter: position of the shot relative to the y axis (measured as a parallel line to the midfield division line).
- angle to goal: angle from the shot relative to the goal line.
- distance to goal line: distance from the shot to the goal line.
- subEventName: categorical variable with the type of preceding pass (No pass, simple pass, cross, smart pass, head pass and high pass) as defined by Wyscout.
- passes is key pass: binary variable indicating if the preceding pass (if any) is a key pass as defined by Wyscout.
- passes is through: binary variable indicating if the preceding pass (if any) was a through pass as defined by Wyscout.
- passes skilled foot: binary variable indicating if the preceding pass (if any) was performed with the player's skilled foot.
- passes distance: distance of the preceding pass (if any).

Table 1 shows The distribution of goals among the 2 periods and the skilled foot of the player taking the shot. It seems that the goals are well balanced among both periods, with a slight increase in the second half. What is more surprising is the fact that shooting with

the non-dominant foot has a higher conversion rate than shooting with the skilled foot. This might be explained by the fact that players tend to try less probable shots with their skilled foot and only shoot with their other foot in more secure occasions.

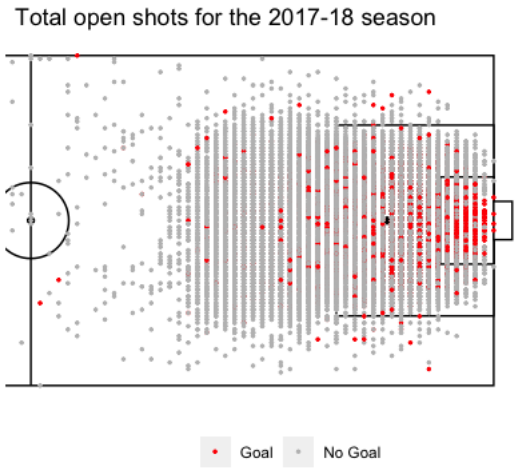

Figure 1: Total shots for the 2017-18 season

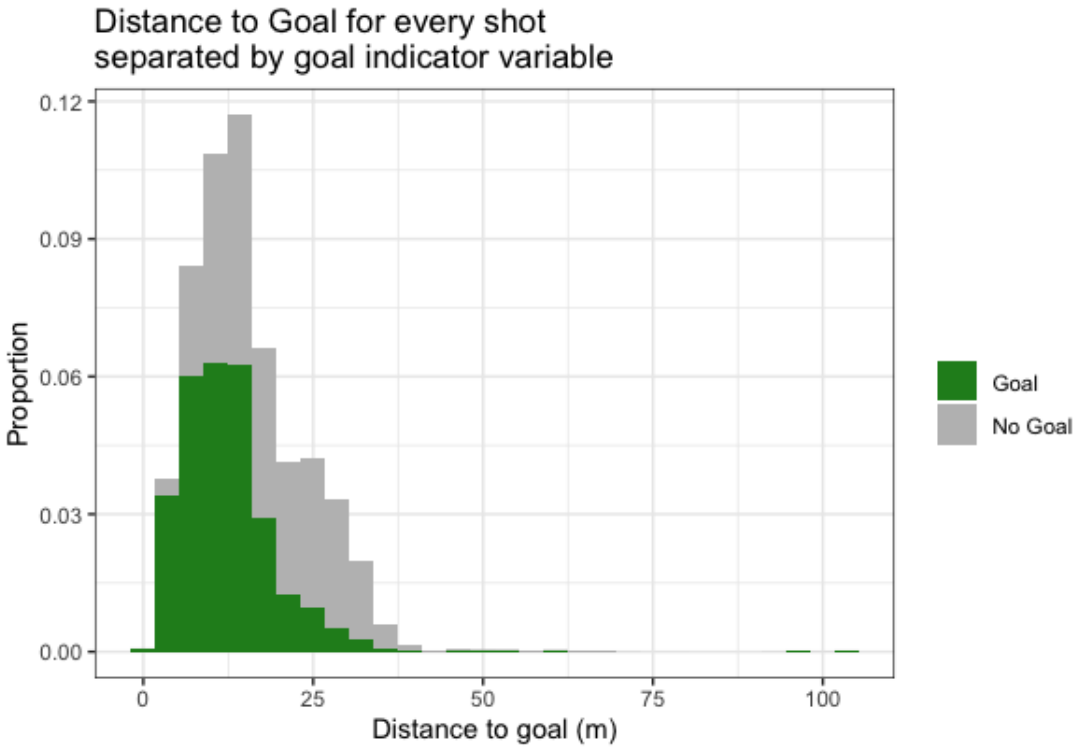

Figure 2: Total shots histogram for the 2017-28 season

Figure 1 shows the location of all the shots taken during the 2017-18 season in the 5 leagues, identifying which ones finished in goal. It is clear that there is an important location factor involved with the probability of scoring a goal: obviously, the closer you are to the goal line, the easier the shot can be. This can also be seen from Figure 2, where the histogram shows that the goal proportion decreases as the distance to goal line increases. It looks like both the histogram for goals and non-goals have a mode at something like 10 meters. This is explained by the fact that most shots are taken from that distance, whether or not they turned into a goal. We must also notice that on the left part of the plot both histograms are similar, indicating that the closer a shot is taken to the goal, it is more likely for it to be converted. On the right side of the plot we can observe the opposite: the further the shot is taken, the more difficult it is to become a goal.

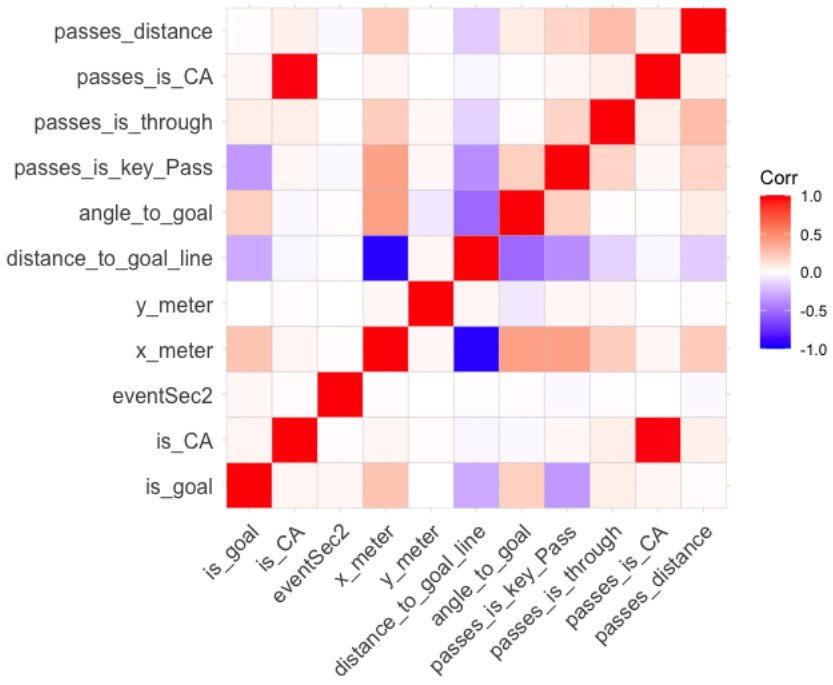

Figure 3: Correlation plot for shots coming from passes

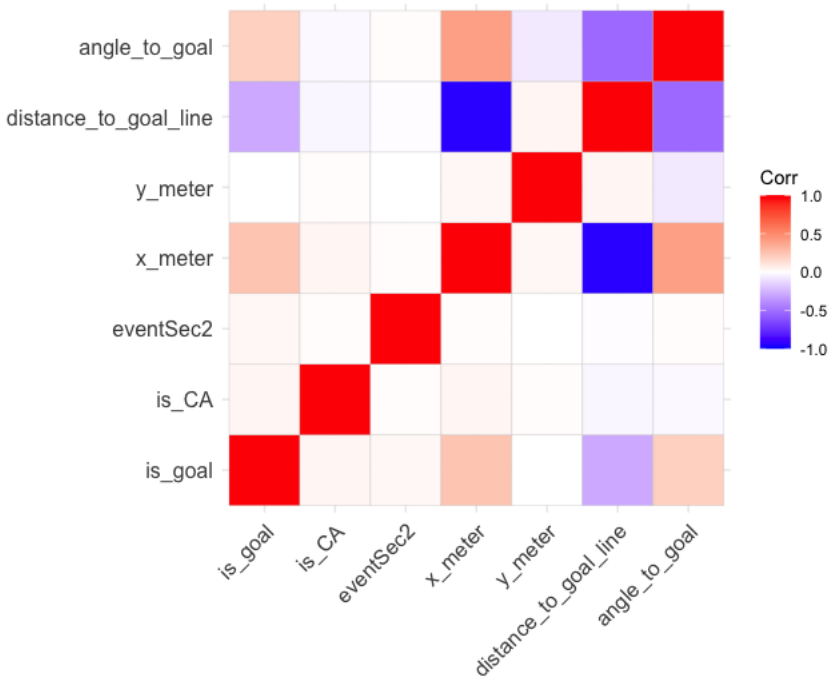

Figure 4: Correlation plot for shots not coming from passes

Figure 3 shows the correlation matrix for all the shots having a previous pass. We must notice that the goals have high positive correlation with the angle of the shot and the x-position on the field. On the other side, they have negative correlation with key passes and with the distance of the shot (as already seen from previous figures). Finally, Figure 4 shows the correlation matrix for shots that do not have a preceding pass. It can be observed that the scoring of a goal keeps basically the same correlations as in Figure 3.

3 Non-parametric Classification

3.1 Logistic Regression

The first method used in this study was logistic regression. It sets a baseline for classification and it can perform really well by modeling the posterior probabilities of the classes via linear

functions of the the explanatory variables ([2]).

The first step taken in order to predict the outcome of our shots dataset via logistic regression was to choose a value of λ in order to perform a penalized regression. Using the package 'glmnet' ([1]), a 10-fold cross validation was applied to the training set in order to select the best parameter for the penalized regression.

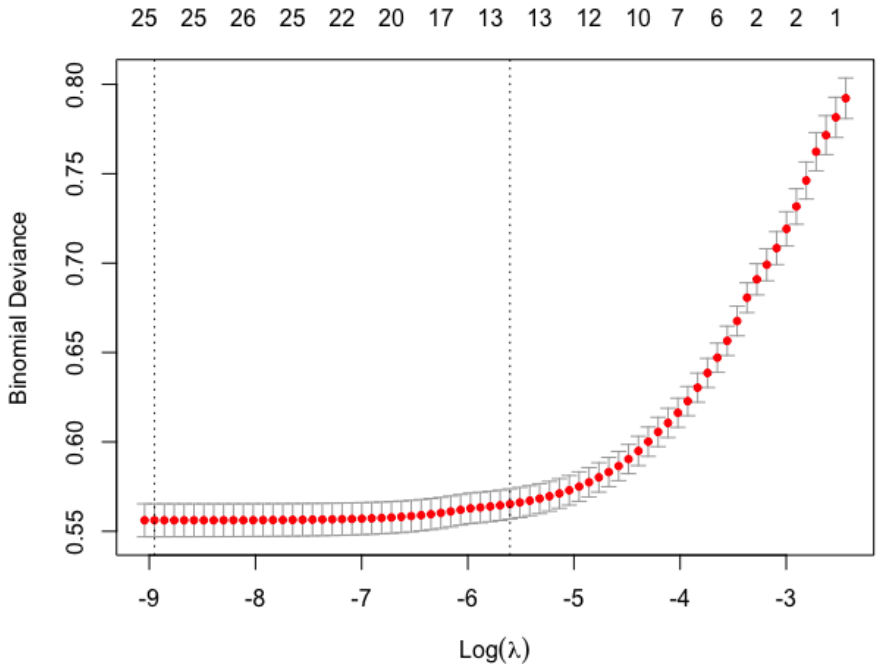

Figure 5: Deviance for the parameter λ selection using 10-fold cross validation

Figure 5 shows that using the selected value for $\lambda = 0.00306$ ($\log \lambda = -5.78$) generates a model which uses 13 variables.

Table 2 shows all the estimated coefficients using the training dataset obtained via the non-parameteric penalized logistic regression. As we can see, we have 13 non-zero coefficients and an intercept. It is surprising to notice that the position of the shot regarding the x-axis was not selected, although the y-axis, the distance to goal and the angle to goal were selected. As we expected, there is a negative coefficient for the distance to goal, meaning that the

Table 2: Summary of non-zero coefficients for the logistic regression

Variable	**Coefficient**
Intercept	0.3172576
matchPeriod2H	0.0351709
is_CA	0.3045063
skilled_footNo	0.8018910
skilled_footYes	0.9413206
y_meter	0.0099444
distance_to_goal_line	-0.1636727
angle_to_goal	0.0024772
subEventNameNo Pass	-0.5608943
subEventNameSmart pass	0.2677652
passes_is_key_Pass1	-4.5696472
passes_is_key_PassNo Pass	-0.3461474
passes_is_through1	0.7784860
passes_skilled_footYes	0.4656465

further the shot is taken, the less probable it is going to end in goal. What results really surprising is the fact that having a preceding key pass has a negative effect on the log-odds of scoring a goal; this is weird since Wyscout defines this passes as 'A pass that immediately creates a clear goal scoring opportunity for a teammate' ([3]); furthermore, it is the coefficient with the largest absolute effect on the log-odds! Finally, the biggest positive effect on the log-odds of scoring is shooting with the skilled foot, which makes a lot of sense, especially when you consider that most players do not have the same ability on both feet (in fact, this is a very rare event, a characteristic that not even all elite players possess).

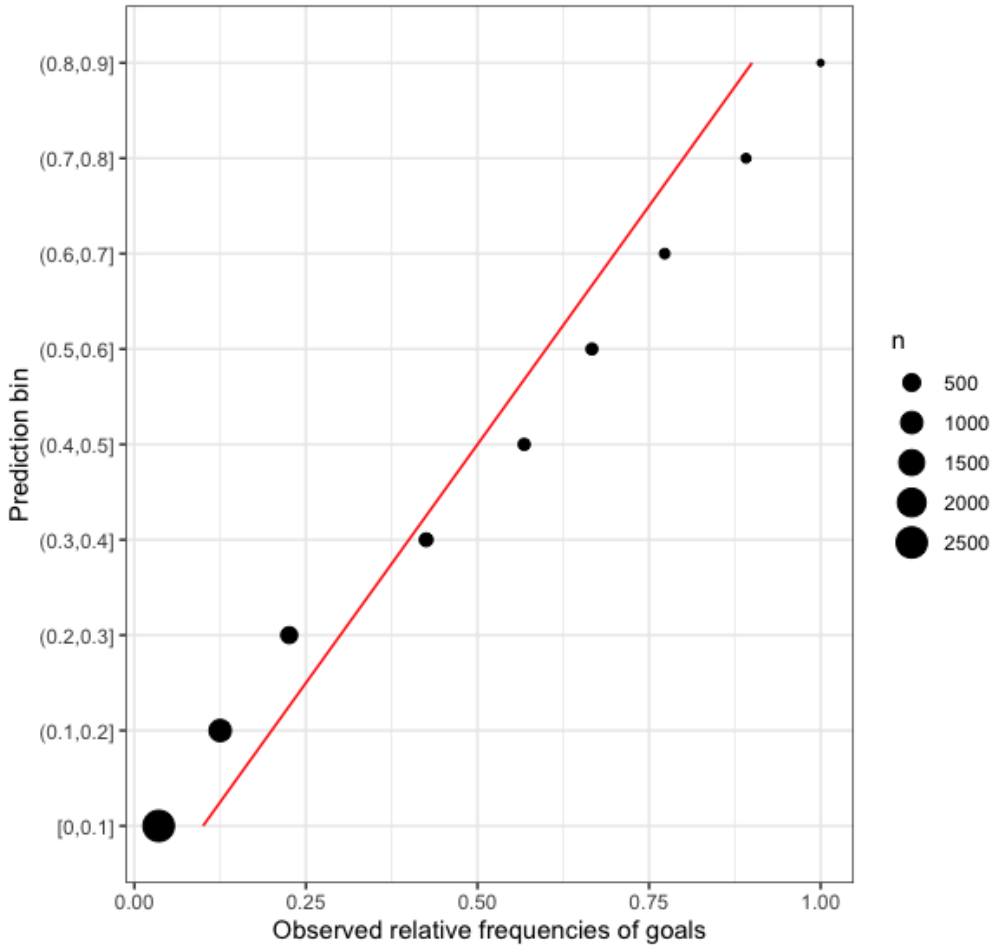

Figure 6: Calibration plot for the logistic regression

Figure 6 shows the prediction bin (made by 10% increments) against the actual frequencies of goals in the validation set. We can see that the model over-predicts a little in shots that do not turn to goals so often and under-predict on the other side (it is a little conservative on 'easy' shots). However, the calibration does not seem to be alarming.

The next step was to consider the threshold by which the predictions are labelled as goal (1) or no goal (0). Using the validation subset, the misclassification rate, sensitiviy, specificity, precision and AUC were computed for distinct values of such threshold.

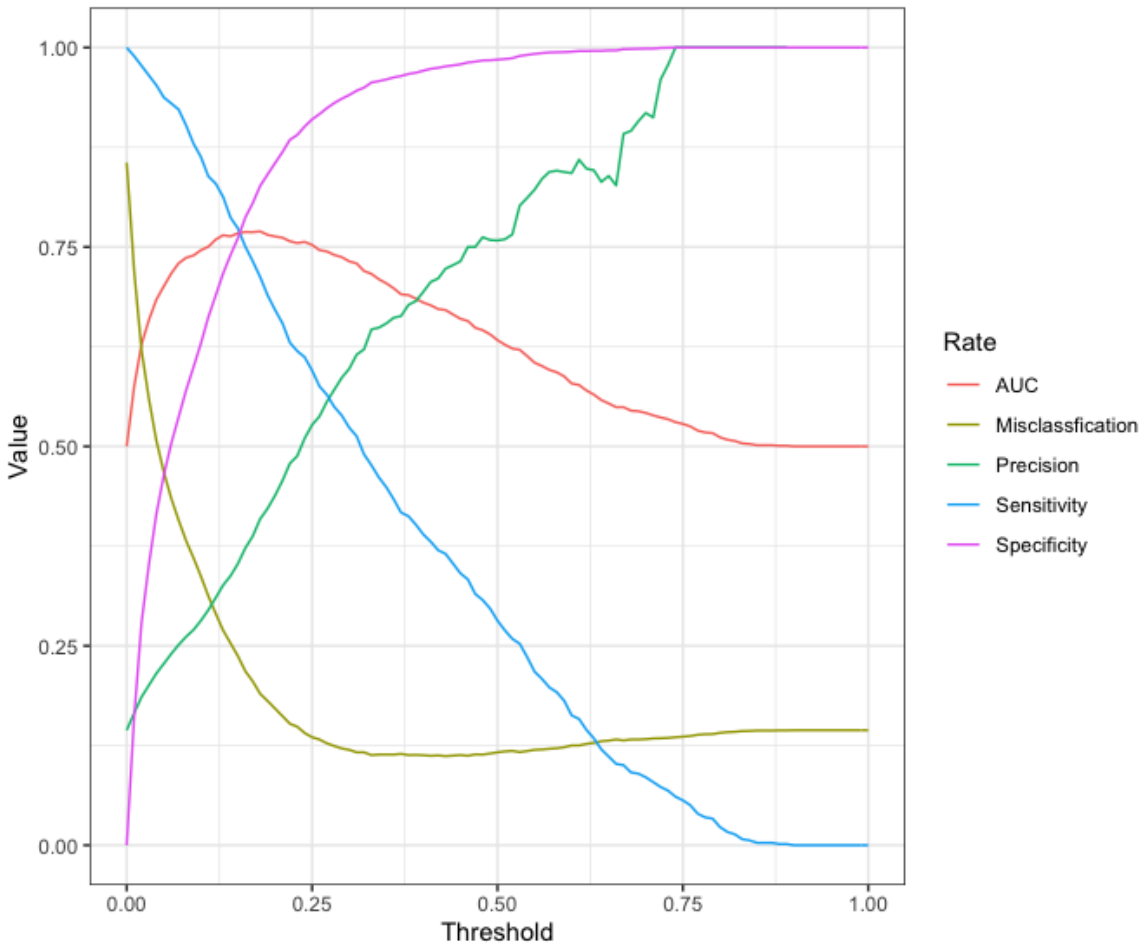

Figure 7: Distinct rates for the value of the threshold

Figure 7 shows how those different rates change when the threshold varies. First of all, there is no point in maximizing the specificity since that would mean setting the threshold = 1 and classifying all the shots as no goals. There is no use in correctly predicting all failed shots because in that case the precision cannot be computed (there are no positive labels), the sensitivity is 0 and the AUC is 0.5, which is the same as flipping a coin. In the case in which we wanted to minimize the misclassification error, our validation set tells us to set the threshold to 0.46, getting a misclassification rate of 0.1121. This would have sense in the setting of wanting a good test since the specificity and sensitivity would be high, but

at the cost of having more false positives (lower precision). The greatest AUC (0.7710) is achieved when the threshold is 0.16, but this comes at the cost of even more false positives than achieving the lowest misclassification error. If we wanted a test in which goal predictions really turned out to be goals (high precision), we would set the threshold to 0.76, but this comes at the cost of more false negatives. At the end, as most of the things in applied statistics, this decision depends on the context at hand. Predicting goals in soccer is not as important as correctly predicting rare diseases (where we would want the highest possible precision), so we have the luxury of trying to maximize other rates. In this analysis, we will present the results on the test sample with threshold of 0.46 and 0.76.

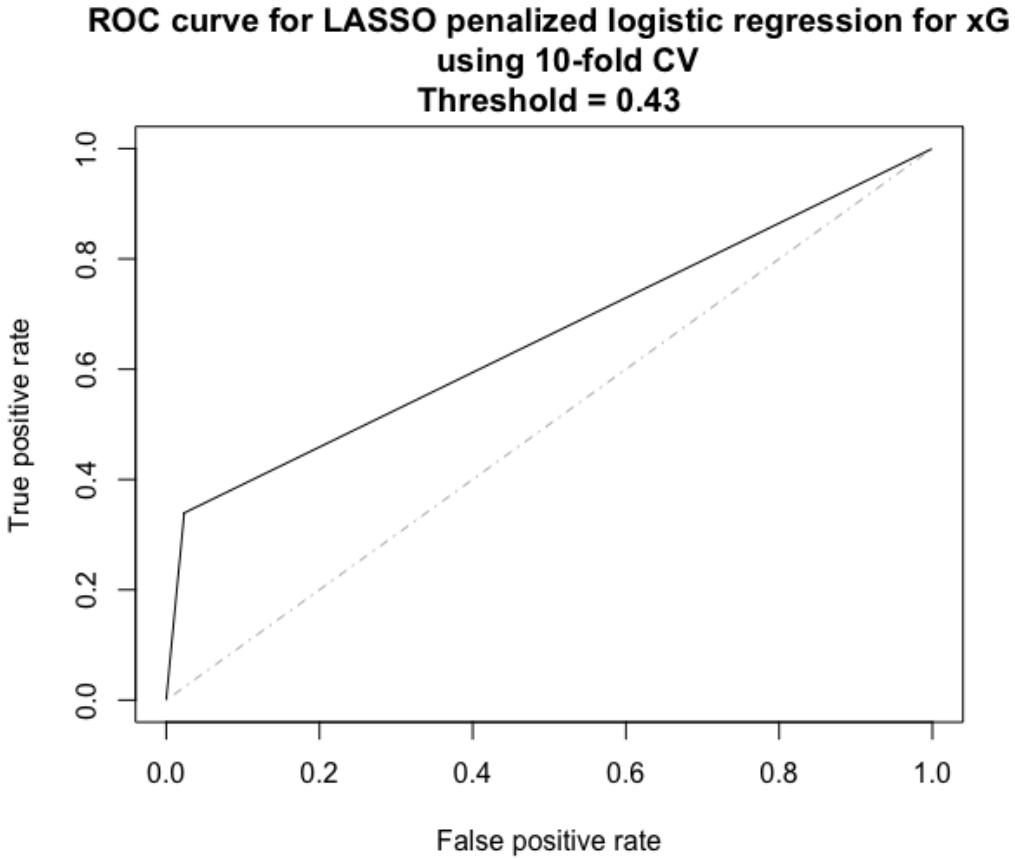

Figure 8: ROC curve for logistic regression

Finally, the fitted model was applied to the test set and estimated probabilities of scoring a goal were obtained for every shot. When the threshold was set to 0.43, the misclassification rate was of 0.1154, the AUC was 0.6458. Figure 8 shows the ROC curve for those predictions. When the discrimination threshold was set to 0.74, the misclassification rate was 0.1362 and the AUC was 0.5293.

3.2 Random Forest

The second method we used was random forests. It is a very good prediction method that builds a large collection of B de-correlated trees. Trees allow for partitions based on recursive binary splitting we create a binary split whether a player will score a goal or not.

The random forest algorithm performs bagging on trees, but it limits the amount of randomly chosen variables it can use for each split. Using the package 'ranger' ([6]), a OOB validation was applied to the training set in order to select the best tuning parameters for the random forest.

We will tune two parameters for the random forest algorithm:

- m = limit of randomly chosen predictors for each split
- n_{min} = minimum node size. Setting this number larger causes smaller trees to be grown (depth parameter).

Recommendations from the inventors of the Random Forest Models suggest we should use $m = \sqrt{p} = \sqrt{29} = 5$ while $n_{min} = 1$, for classification. We decided to tune these parameters using OOB. Unlike CV, OOB summarizes the overall misclassification rates when applying each of the B resample trees to the out of bag cases that were not part of the resample.

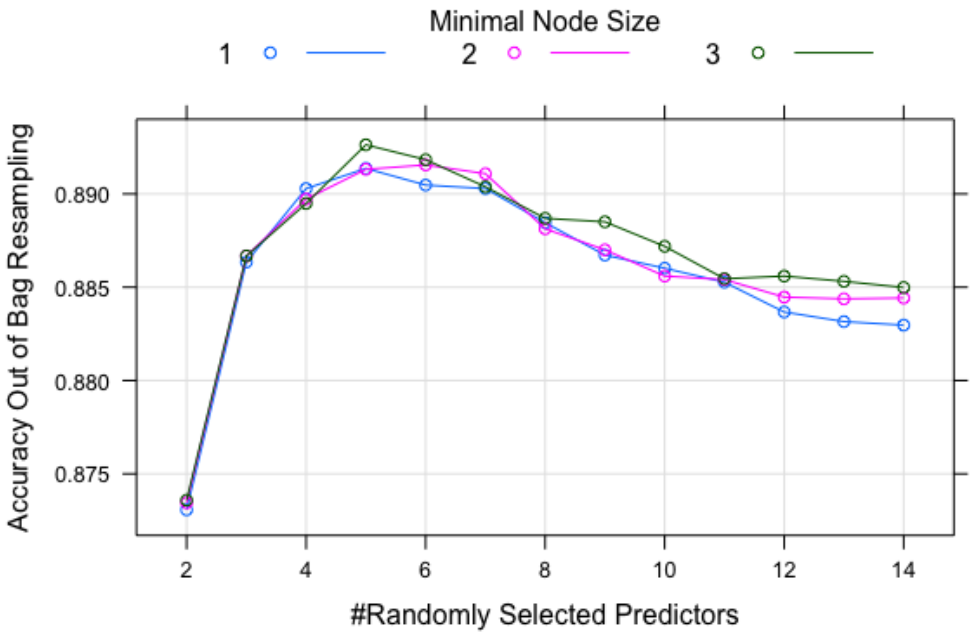

Figure 9: OOB Accuracy plot Random Forest

We can see in Figure 9 the effects of the variance-bias tradeoff when we look at the number of randomly selected predictors m. The bigger m is we tend to get OOB accuracy, but there comes a point where too many predictors is detrimental to the accuracy of the model due to overfitting. That is why we see a concave shape for all 3 different node sizes. Minimum node size seems to not have a big effect on the OOB estimates of our models. However it is important to note that $n_{min} = 1$ is not always going to lead to the best OOB estimates. In fact OOB Accuracy seems to be higher with $n_{min} = 2, 3$ for certain predictor sizes such as 9-11. Our tuning estimates shows that the best random forest model has a $n_{min} = 2$ with $m = 6$.

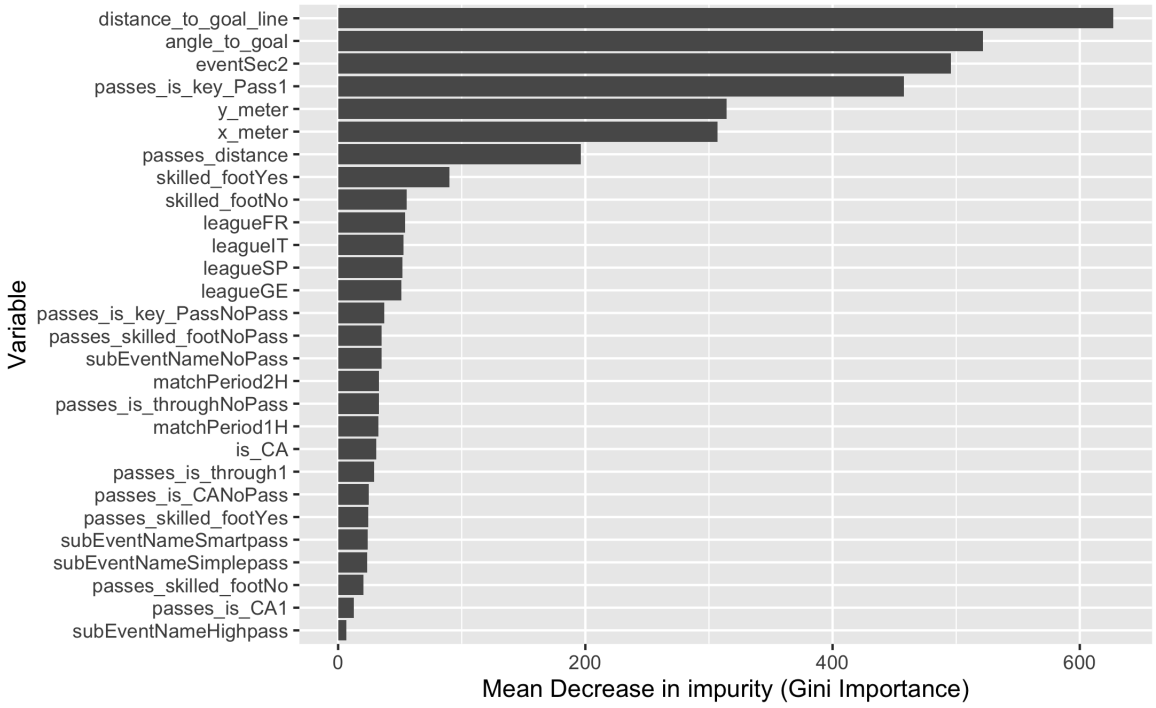

Figure 10: Variable Importance Random Forest

Our model evaluates variable importance using the mean decrease in Gini which is a measure of node purity. Overall from what we can see in Figure 10 is that continuous variables tend to have higher importance than categorical variables, but this is expected as continuous variables are split more often than categorical variables. From what our model suggests, passes leading to the goal do not tend to have high importance in predicting whether a shot will lead to a Goal. One passing variable which is if the pass was a key pass tends to be highly predictive of whether a player will score]. However the type of pass, cross, through pass, head pass, high pass do not seem to be as important. Instead, variables that directly measure the state of the shot such as the distance to the goal line and the angle of the shot and whether the shot was shot with weak/strong foot tend to be far more important in determining whether a shot will end up converting to a Goal.

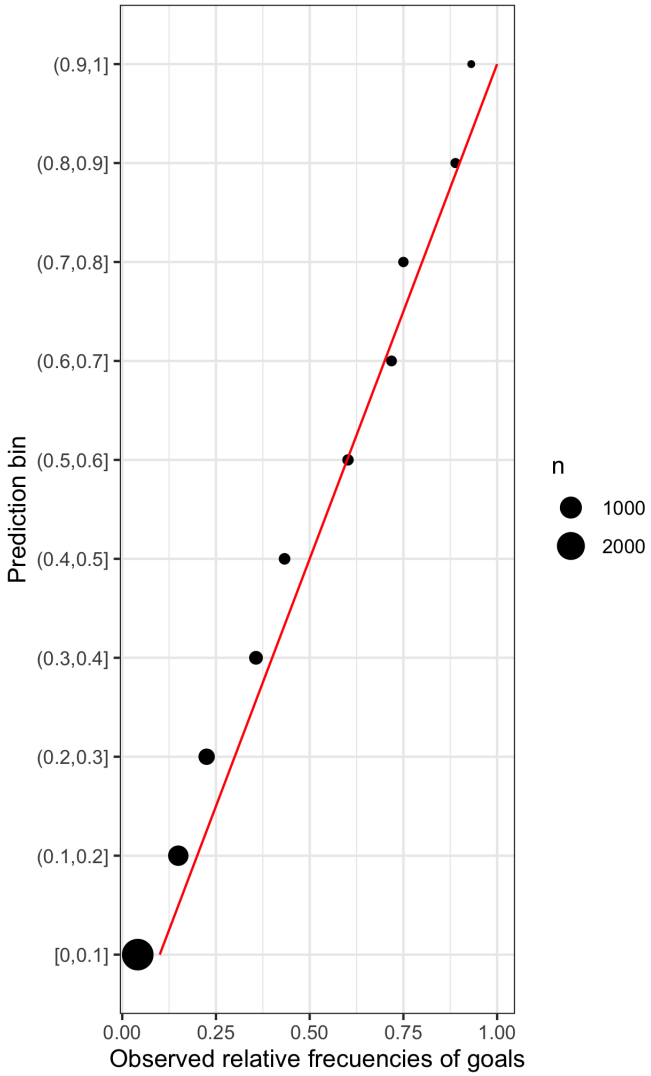

Figure 11: Calibration plot Random Forest

Similar to the logistic regression calibration, Figure 11 the random forest predictions we can see that the model also overpredicts shots that do not turn to goals. This will potentially lead to a smaller cutoff rate to increase accuracy.

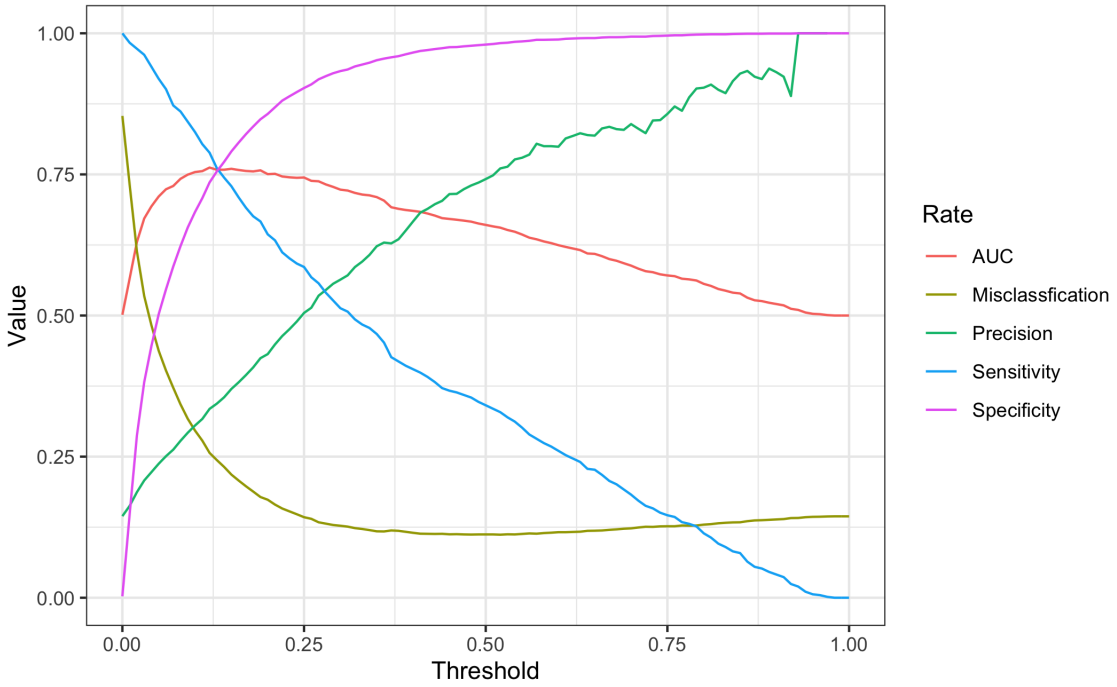

Figure 12: AUC plot Random Forest

From Figure 12 we can see the tradeoff of increasing or decreasing the threshold. Because we predict more goals the lower the threshold, the more likely we are to have more false positives. Becuase there are more no goals than no goals, it makes sense that misclassification decreases fastly at the lower threshold levels and doesnt increase as much as we increase the threshold. In the case in which we wanted to minimize the misclassification error, our validation set tells us to set the threshold to 0.52, getting a misclassification rate of 0.1138, which is higher than the misclassification rate of the logistic regression. The problem with just looking at misclassification or accuracy in our models is that we tend to predict less goals. The greatest AUC = 0.7621 is achieved when the threshold is 0.12, this makes sense as there are more no goals than goals in the model so if we wanted to balance specificity and sensitivity this would be the best threshold, but in this case we wiould over estimate our xG predictions. The results and thresholds for our models depend on what the model is used

for. For example a defensive coach would want an overprediction of xG for the opponents shots so that he can evaluate how to minimize risky changes coming from the other team. While an attacking coach might want an underprediction of xG to better understand what type of shots tend to be more effective in front of goal.

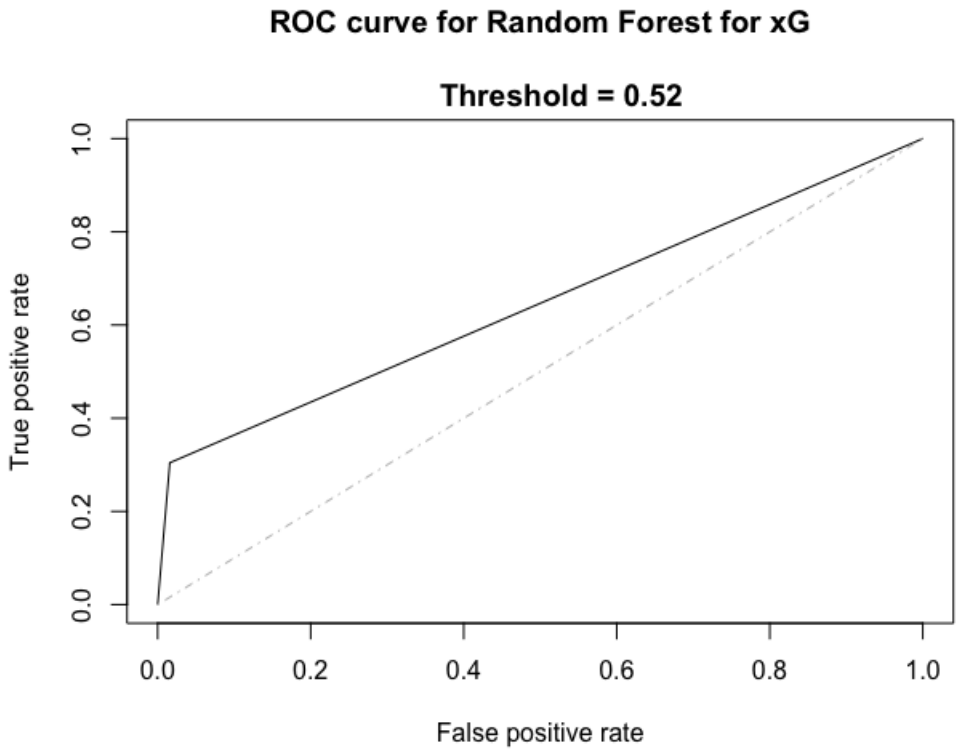

Figure 13: AUC plot Random Forest

The fitted model for the random forest model was applied to the test set and estimated probabilities of scoring a goal were obtained for every shot. When the threshold was 0.52, the AUC was 0.6443. Figure 13 shows the ROC curve for those predictions. Finally, using the thresholds that maximized the precision and minimized the misclassi- fication rates, the estimated probabilities for the test subset were labelled. When using a threshold of 0.93, the test sample misclassification rate was of 0.1399 with an AUC of 0.6443.

Table 3: Selected variables from the stepwise regression

Variable	**Coefficient**
Intercept	11.2535
matchPeriod2H	0.2354
is_CA	0.4921
eventSec2	0.0000
skilled_footNo	1.2079
skilled_footYes	1.3235
x_meter	-0.1015
y_meter	0.0261
distance_to_goal_line	-0.2788
subEventNameHigh pass	-0.5203
subEventNameNo Pass	-1.5584
subEventNameSimple pass	-0.7290
subEventNameSmart pass	-0.1808
passes_is_key_Pass1	-6.7022
passes_is_through1	0.9786

3.3 Boosting

The last method used was boosting. We know that for this kind of algorithm we have three tuning parameters:

- d, the number of splits (depth parameter)
- λ the shrinkage rate
- B, the number of trees

Again, we performed 10-fold CV to select them, but first, we did a backward step wise regression (as recommended) in order to select the variables that we are going to use (choosing the model by AIC).

As we can see from Table 3, the variables selected are too similar to the ones seen for the penalized logistic regression in Table 2. One thing to notice is that in this model, the angle to goal line remains with a zero coefficient (although having the (x, y) information and

the distance to goal is equivalent to having the angle). Again we observe that there is a negative relantionship between the previous pass (if any) being classified as a key pass and the log-odds on scoring a goal.

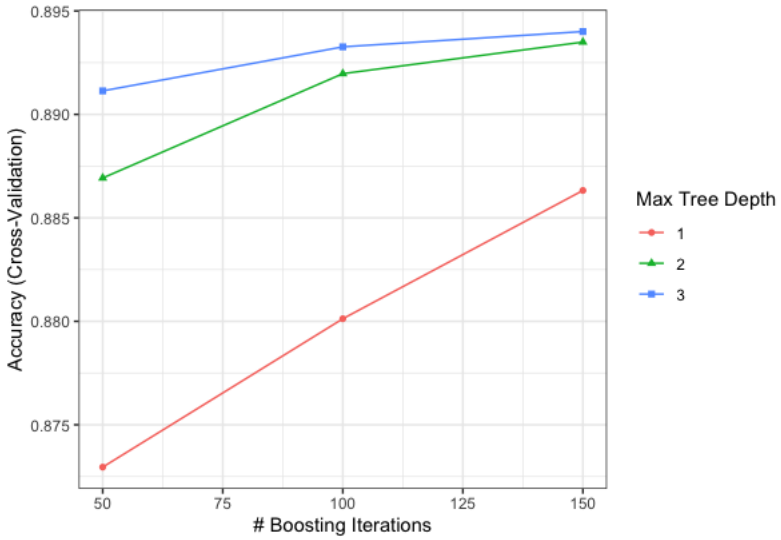

Figure 14: Accuracy plot for the 10-fold Cross Validation

Then, using the 'caret' library ([4]) for the 10-fold cross validation, the parameters were set on $\lambda = 0.1$, $B = 150$ and $d = 3$. Figure 15 shows that the selection of these parameters maximizes the accuracy in the training sample.

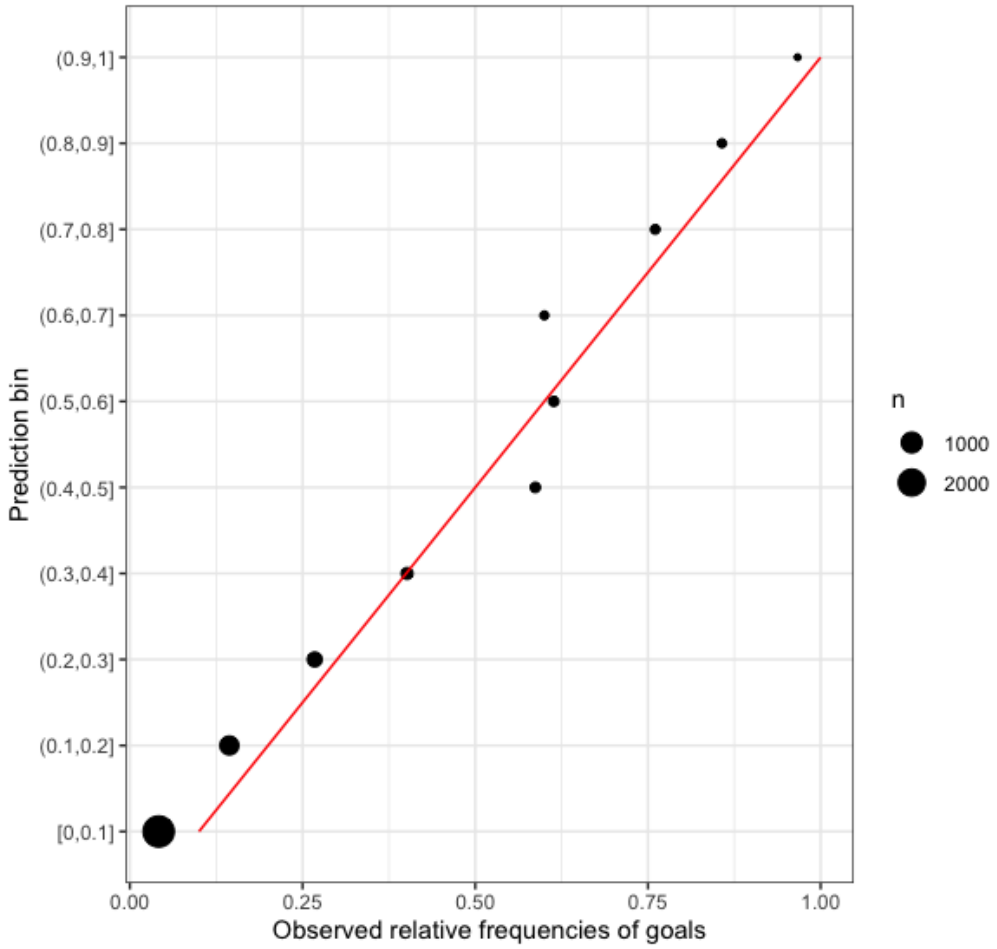

Figure 15: Predicted goal probabilities vs actual frequencies using boosting

Next, we checked the calibration of the predicted probabilities on the validation set using 10% bins. Figure 15 shows that the model is well calibrated although it seems to be a little bit off for the (0.4, 0.5] bin. Outside from that, there is no sign of problem with the predictions.

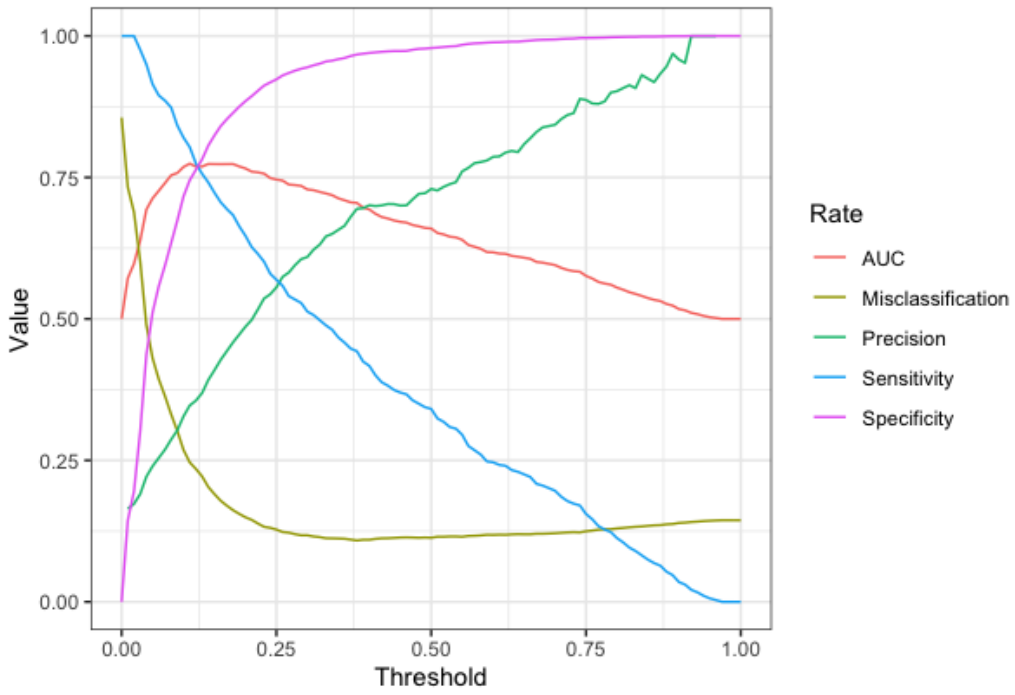

Figure 16: Classification metrics for distinct threshold values

We also used the validation set to help us decide a good cutoff for the labeling as goal (1) or no goal (0). Figure 16 shows that is the goal was to maximize the AUC, then a threshold of 0.14 would help us achieve an AUC of 0.7795. If we wanted to maximize the precision, then we would get a maximum value of 1 with a threshold of 0.90 (but our AUC would be 0.51). Finally, if we wanted to minimize the misclassification error, then a threshold of 0.40 would let us have a rate of 0.1104 with AUC equal to 0.6938.

Finally, using the thresholds that maximized the precision and minimized the misclassification rates, the estimated probabilities for the test subset were labelled. When using a threshold of 0.90, the test sample misclassification rate was of 0.1371 with an AUC of 0.5250. Using the threshold value of 0.40 yielded a misclassification rate of 0.1119 with an AUC of 0.6821. The first results are basically unacceptable, since they are close to worthless (in the sense of being no better than a coin flip).

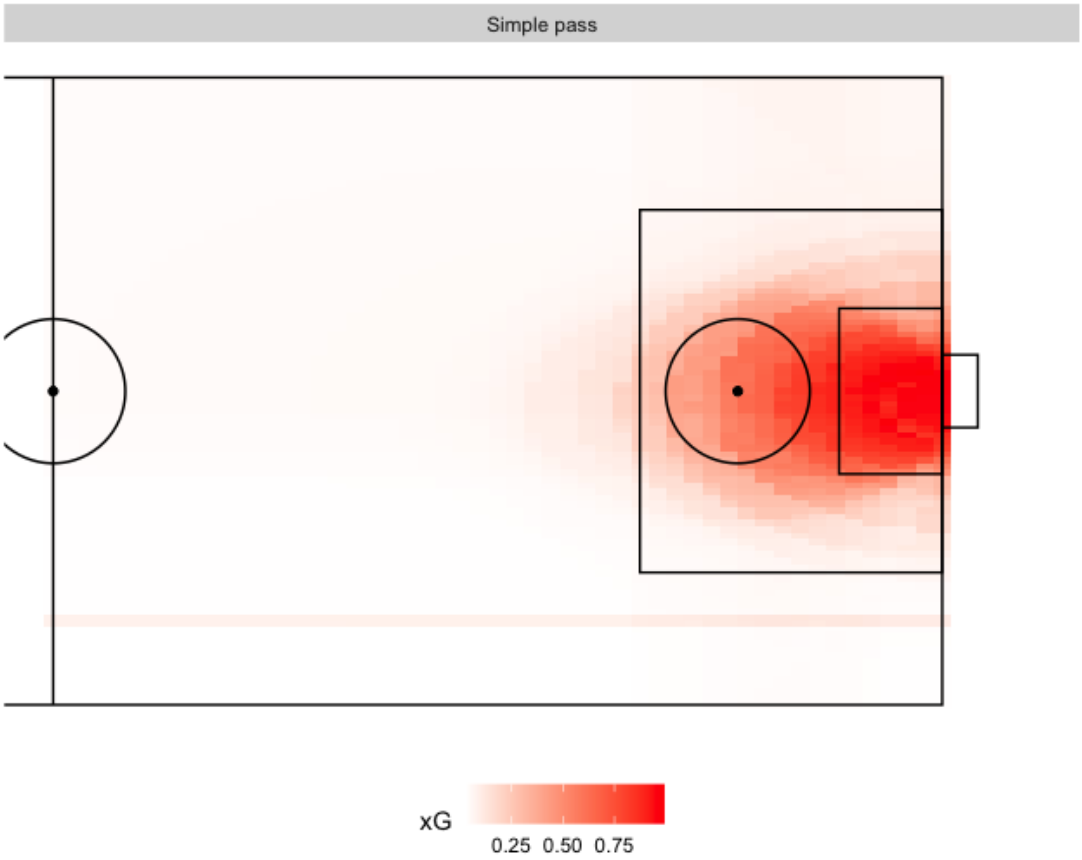

Figure 17: Predicted Probabilities with Boosting

To give an illustration of what the test predictions look like in the pitch, we can assume a scenario where a simple pass traveled 10 meters, the player used his skilled foot, the pass was not through, not key and not counter-attack, playing in the German league and on the second half. Figure 17 shows a heatmap of a soccer pitch made with the predicted probabilities of a shot coming from the described scenario. We can observe that the probability of scoring reaches it highest in the small area and it grows dimmer in an elliptical way.

4 Conclusions

When performing any kind of statistical analysis, one must take into account the quality of its data and the context of the problem. Setting a baseline with logistic regression and observing that both the random forest and boosting modelling method do not outperform it can make us think of two immediate things:

- The modelling with the logistic regression might be enough to explain the outcome.
- Our dataset and models need of more complexity: it is evident that the characteristic of the shots alone cannot be sufficient to have an amazing xG estimation. With more data about the position of the teammates and rivals, pass chains, speed and acceleration of the shot, etc., better models could be created.

When comparing all the methods, Random Forest has the lowest misclassification rate at 0.1138, but that is not a significant decrease compared to the more simpler logistic regression model which has a misclassification rate of 0.1154. Finally, boosting has the best misclassificataion rate at 0.1119, still not that better than any of the two other methods.

References

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. URL: https://www.jstatsoft.org/v33/i01/.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[3] *Key pass*. URL: https://dataglossary.wyscout.com/key_pass/.

[4] Max Kuhn. *caret: Classification and Regression Training*. R package version 6.0-90. 2021. URL: https://CRAN.R-project.org/package=caret.

[5] Luca Pappalardo et al. "A public data set of spatio-temporal match events in soccer competitions". In: *Scientific Data* 6 (Oct. 2019). DOI: 10.1038/s41597-019-0247-7.

[6] Marvin N. Wright and Andreas Ziegler. "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R". In: *Journal of Statistical Software* 77.1 (2017), pp. 1–17. DOI: 10.18637/jss.v077.i01.