Title: Introducing Expected Goals (xG) Premium

Group 7: Hpone Myat Khine (A0125002E), Kwang Hun Lee (A0231958W), Sae Jin Jang (A0231989M)

1. Introduction & Problem Statement

1.1 Introduction

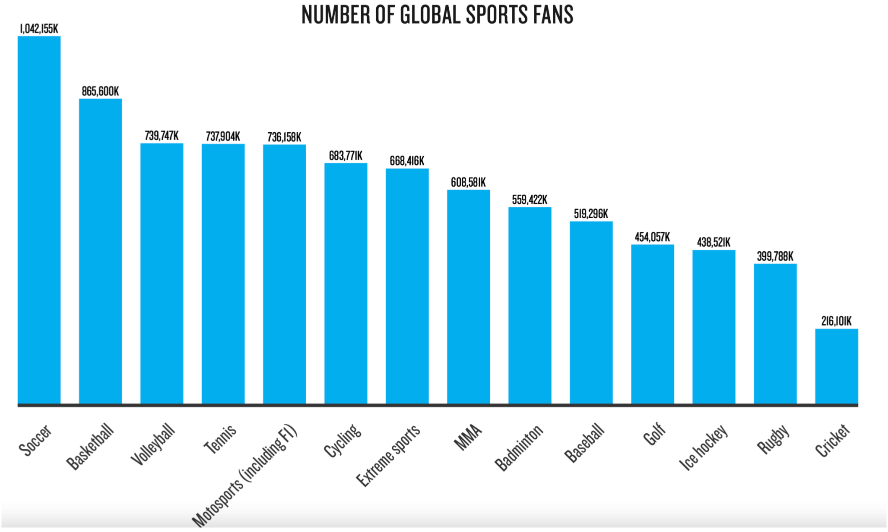


Figure 1: Chart of Global Sports Fan

Football, also known as soccer in the United States, is one of the most popular sports in the world. The popularity is also boosted in part by the surge in dominance particularly from the European football associations - from 1990 to the latest 2018 World Cup, European nations have emerged champions in all but two (1994 and 2002 both of which went to Brazil). It is without a doubt too that these European leagues dominate the football scene both in terms of prestige and skill as well as value - with the top 10 most valuable soccer brands all coming from Europe.

1.2 Problem Statement

Football teams have been trying to find ways to score more goals in each game to win and to achieve better results in each season of their respective leagues. Traditionally, football teams have used past goals, shots, and shots on target to evaluate the goal-scoring ability of a player. However, recent trends show that more teams are using data analytics to find scientific approaches to evaluate a threatening player more objectively. Expected Goals (xG) measures the quality of goal-scoring chances by calculating the probability of scoring a goal from a given shot. This metric is not only beneficial in determining how to score more goals, but it's also important in developing teams' tactics, positioning right players at the right area, and recruiting right players in the following season to achieve better results.

2. Data Identification & Exploratory Data Analysis (EDA)

2.1 Data Set

In our study, open data source from StatsBomb was used to predict expected goals in future games. The data was obtained by extracting all the shots from all the freely available data which consists of 39 unique competitions and year spanning national and league football for men's and women's football. Some examples include men's 2020 Euros to men's La Liga, the top-tier Spanish league, 2016/2017 season data to women's World Cup 2019 data. The only competition that was excluded was the 1999/2000 men's Champions League as the data returned an error. The data for a particular shot include detailed information such as *play_pattern* (From Throw in/Counter/Corner/Goal kick/etc.), *body_part* (Right/left foot, head, others), x and y coordinates of the start and end location, etc. These would later be our candidate features to create our models after data cleaning.

One key caveat for this report is that our Expected Goals (xG), in the football world, is known as non-penalty Expected Goals (npxG). As we will discuss below, penalties were discarded from our analysis. However, we have stuck to using xG instead of npxG for brevity and ease of reporting.

This problem in its basic form is a binary classification problem where the models are tasked to predict 1 if the shot is predicted to result in a goal and 0 otherwise. Interestingly, the expected value of a goal is equivalent to the probability of a goal.

2.2 Data Cleaning

The target binary variable, *goal*, was created by using the outcome variable where if the outcome was a goal, the value would be 1 and 0 otherwise. Features like *deflected*, *open_goal*, *one_on_one*, *redirect*, *follows_dribble*, *first_time*, *under_pressure* were all either NaN or True values. These features were created into binary features where NaN values are 0 and True values are 1. Goals from penalty kicks were removed from our dataset as we are actually predicting the non-penalty expected goals as mentioned previously. In total, 27, 289 shot cases were used in our analysis.

2.3 Feature Engineering

Intuitively, the distance from the goal to the shot of the player would be an indicative feature for a goal-scoring opportunity. This is evident from the shot map of all 27, 289 non-penalty shots.

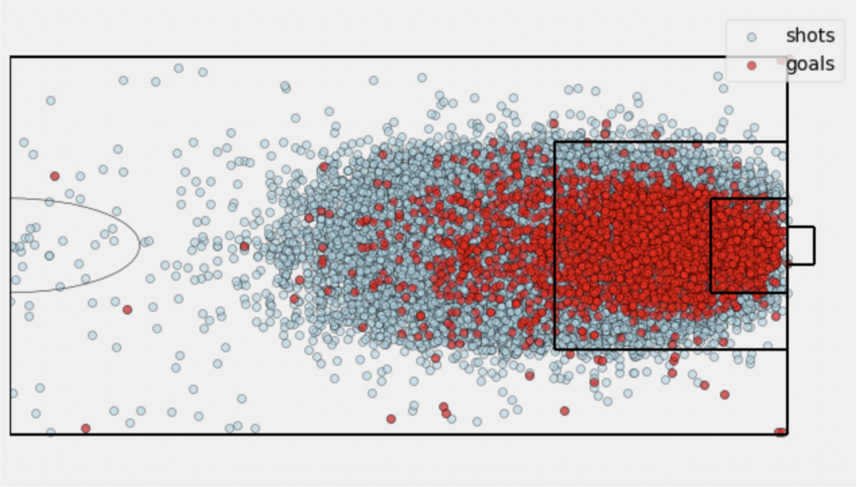
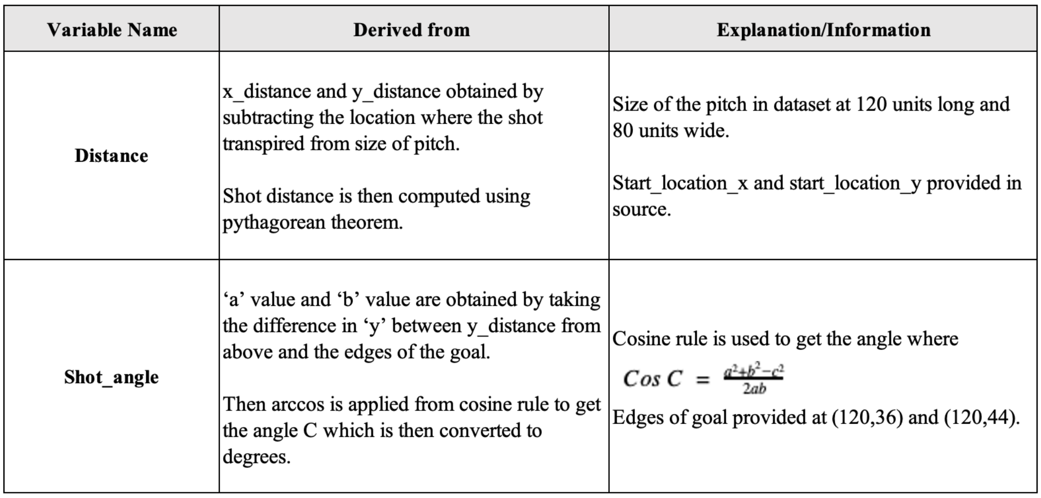


Figure 2: Shot Map

The *Distance* variable was constructed by using the distance formula with the x and y coordinates of the shot and the length and width of the football pitch. Another feature that was engineered was the angle of the shot where it is reasonable to assume that the smaller the angle, the harder it is to score a goal. The table below lists the variables that were feature engineered.

Table 1: Feature engineered data



Variable Name	Derived from	Explanation/Information
Distance	x_distance and y_distance obtained by subtracting the location where the shot transpired from size of pitch. Shot distance is then computed using pythagorean theorem.	Size of the pitch in dataset at 120 units long and 80 units wide. Start_location_x and start_location_y provided in source.
Shot_angle	'a' value and 'b' value are obtained by taking the difference in 'y' between y_distance from above and the edges of the goal. Then arccos is applied from cosine rule to get the angle C which is then converted to degrees.	Cosine rule is used to get the angle where $Cos\ C\ =\ \frac{a^2+b^2-c^2}{2ab}$ Edges of goal provided at (120,36) and (120,44).

2.4 Exploratory Data Analysis (EDA) & Feature Selection

To accurately forecast the likelihood of the goal, relevant features had to be selected. From the available 33 columns, the irrelevant ones - such as *index* and *time* were ignored. The (x, y) start coordinate features served as building blocks to distance and shot_angle features, so they were dropped from the feature selection. The initial possible relevant features were selected intuitively as well as after having done some EDA. These relevant features are listed in Table 2, and the detailed feature exploration is included in the attached code.

We further examined each of the 12 features in Table 2 against the target binary variable, *goal*, and by comparing multiple features against each other to find any further insights or incongruencies with our intuition. For instance, we can see that an open goal will be more likely to result in a goal than a goal that is not open which is hard to disagree with.

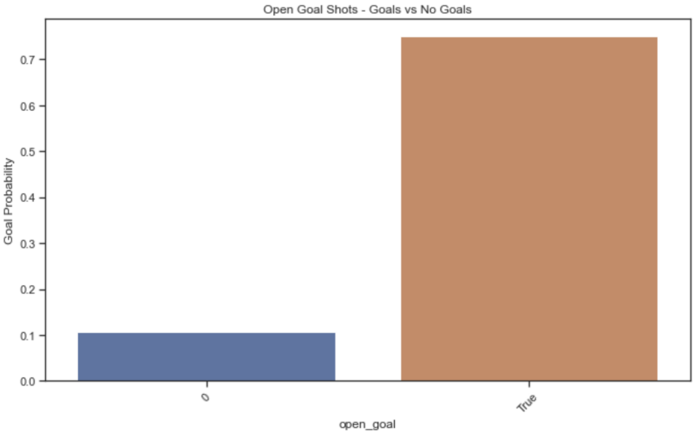


Figure 3: Probability of Goal Given Open Goal

The two features that were engineered can be observed in a scatterplot, and it is clear to see a roughly inverse relationship with the shot angle and distance. As expected, most shots are successful when the distance between the shot and the goal is minimal and the angle of the shot is wide.

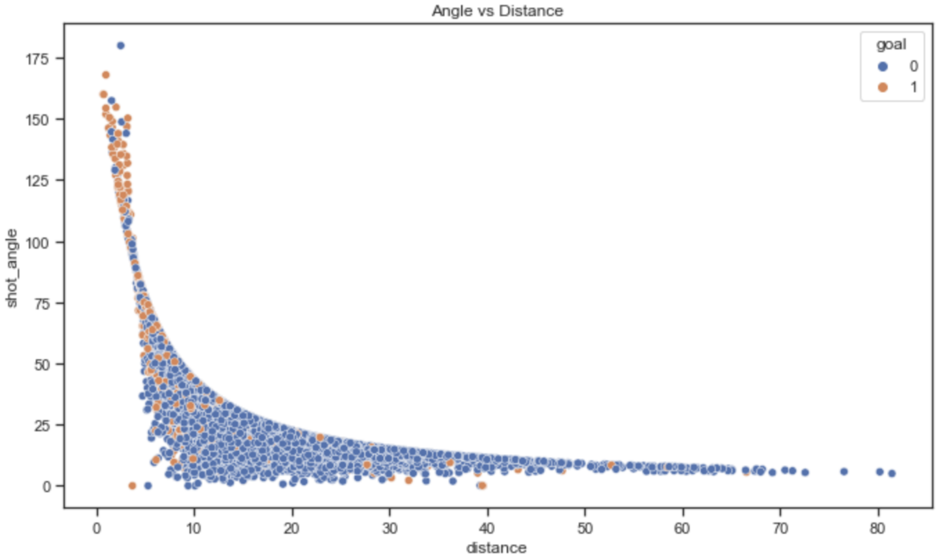


Figure 4. Likelihood of Goal for Shot Angle vs Distance

It was determined that no further features needed to be dropped as each feature provided its own value to potentially improve the model performance based on our EDA.

Table 2: Feature Selection

Variable Name	Source	Explanation
play_pattern	Provided in source	Type of play e.g. regular (open) play, from corner kick, from counter
under_pressure		Whether the player was under pressure at moment of shot
body_part		(Legal) Part of body used
technique		Technique used e.g. normal or volley
first_time		Whether the shot was taken at first instance without prior touch
follows_dribble		If the shot followed dribble (as opposed to first_time)
redirect		If shot was redirected
one_on_one		If it was a one to one matchup
open_goal		If the shot was an open goal
deflected		If the shot was deflected off someone
distance	Feature Engineered	Distance of shot from goal
shot_angle		Angle of shot from goal

2.5 Pre-processing

According to our initial analysis, from Figure 5 only 11.43% (3,119) of the total 27,289 shots led to goals. This is a low conversion rate and indicates most shots taken do not result in a goal; this also results in a class imbalance as there are a lot more instances of 'No Goal' than ‘Goal’, which could adversely impact the models or when measuring accuracy as the algorithm can achieve a high accuracy by simply predicting the majority class - ‘0’.

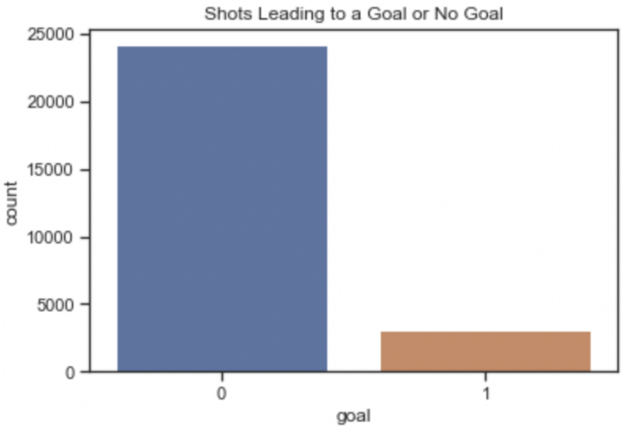


Figure 5: Count of shots that result in goal (or not)

To ensure a more balanced class, we opted to undersample the majority class to ensure a 1:1 split between both a goal and no goal. The figure below shows the balanced instances after undersampling. Although this was ~ 90% decrease in sample, we believe it would give a more reliable modelling. The test set was untouched so that the model could then be used on data as how it appears in the real world - where most of the shots taken would not result in goals.

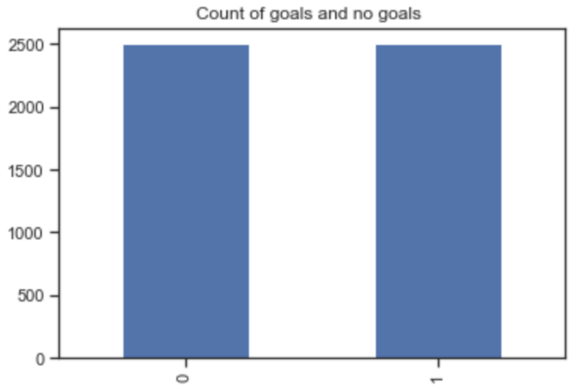


Figure 6: After under-sampling; Balanced split seen

Before training our models with our selected features, we used a label encoder to transform categorical features (*play_pattern*, *body_part*, and *technique*) to discrete values so that the data could undergo training to create our models.

An initial simple analysis was conducted with a Decision Tree to identify the most important features amongst those shortlisted.

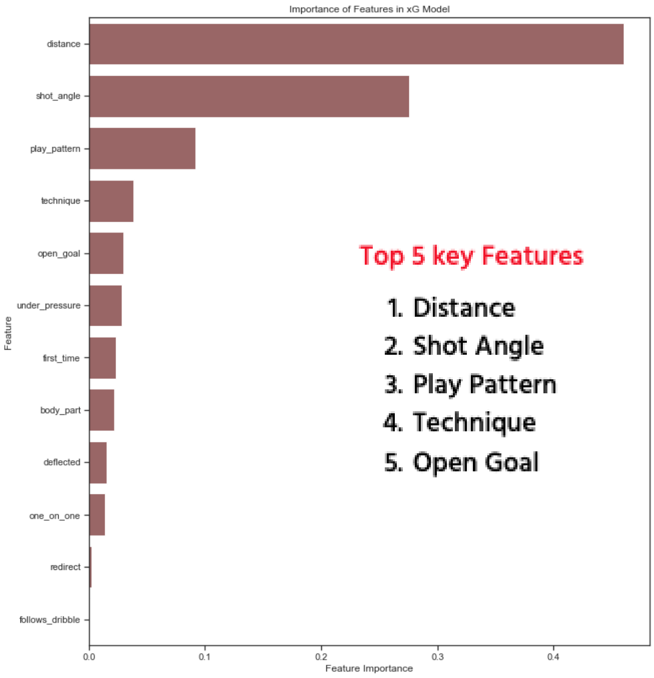


Figure 7: Importance of features from Decision Tree

Distance, *shot_angle* and *play_pattern* were the top 3 most important features with *distance* and *shot_angle* being the most important. This intuitively makes sense as we mentioned that a closer distance would mean an easier opportunity to score. Likewise the angle of the shot could make the difference between a goal or no goal.

Having identified the important features, the entire training set was trained on a variety of models to identify the best performing as well as ease of interpretability. In addition, we sought to also balance out the variance as well as strong predictors that might dominate from Decision Tree classifiers.

3. Model Comparison & Evaluation

The following models were utilised in training and their metrics compared:

- Logistic Regression
- Decision Tree
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors (KNN)
- Bagging (of Decision Tree)
- Random Forest
- Naive Bayes Gaussian (NB)
- Support Vector Machine (SVM)

3.1 Model Comparison

In developing our model, *stratified K-fold* and *cross_val_score* was used to obtain their mean accuracy scores to identify the more important models. Once identified, the hyperparameters were further tuned and their accuracy scores were reproduced. The information is shown herein:

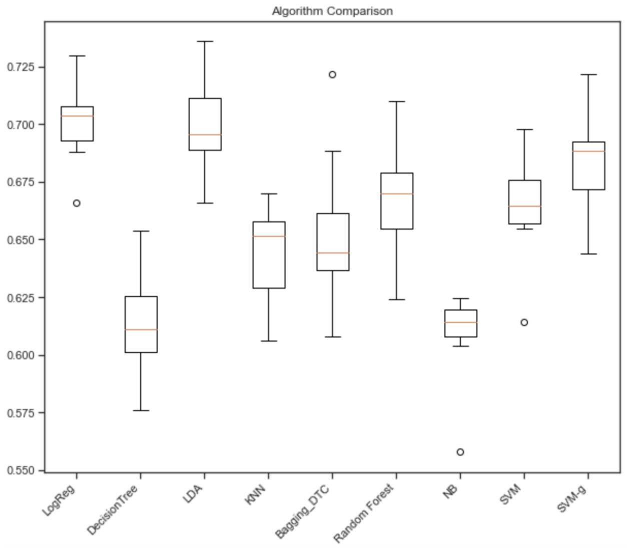


Figure 8: Cross_val_score algorithm comparison

Table 3: Comparison of Accuracy after tuning

Model	Mean Accuracy	Accuracy after tuning
Decision Tree	0.613313	Not Done; Model not important
Naive Bayes GuassianNB	0.609306	Not Done; Model not important
Bagging Decision Tree	0.651475	0.684866
K-NearestNeighbor	0.643881	0.7107
Random Forest	0.668661	0.716196
SVM	0.664472	0.71583
SVM (gamma - auto)	0.683255	
Linear Discriminant Analysis	0.699243	0.716196
Logistic Regression	0.700838	0.72151

All tuned models showed an improved accuracy after tuning. However, as accuracy can be misleading and not generally indicative especially as was noticed with an imbalanced class, a classification matrix was created and the tuning results for each is further expounded on below:

The classification matrix represented henceforth is of the form:

		Predicted	
		No Goal	Goal
Actual	No Goal	True Negative	False Positive
	Goal	False Negative	True Positive

Figure 9: Classification Matrix Template

In addition the values obtained from the confusion matrix are also shown together in table:

Precision: The share of predicted positive cases that are correct

→ Precision = $\frac{TP}{TP+FP}$

Recall: The share of actual positive cases which are predicted correctly

→ Recall = $\frac{TP}{TP+FN}$

F-1 Score: An harmonic mean of recall and positive i.e. it strikes a balance between them

→ F-1 = 2 × $\frac{Precision * Recall}{Precision + Recall}$, where 0≦F-1≦1, with value closer to 1 being the 'best' i.e. where precision and recall are both 1 and value closer to 0 being worse off.

AUC: Area under the ROC (Receiver Operating Characteristics) curve. Value shows degree of separability of classes 0≦AUC≦1; the higher the better the model is at distinguishing classes

Table 4: Overview of classification metrics

Score	LR	LDA	KNN	SVM	BA	RF
Precision	0.24	0.24	0.23	0.24	0.22	0.24
Recall	0.69	0.69	0.65	0.72	0.69	0.72
F-1 Score	0.36	0.36	0.34	0.36	0.33	0.36
AUC	0.774	0.773	0.751	0.78	0.739	0.782

*LR = Logistic Regression; LDA = Linear Discriminant Analysis; kNN = k-Nearest Neighbor; SVM - Support Vector Machine; BA = Bagging; RF = Random Forest

For brevity, we present the confusion matrix of the top 3 best performing models below - Random Forests, SVM, and Logistic Regression

All models produced low precision compared to recall as the prediction, across different models, of a goal was only correct around 25% of the time, and – of all the goals – each model could identify around 70% of the goals. As we see below, this was due to the fact that a minority of shots ended up as a goal, so the challenge of the models would be accurately predicting a goal and not so much predicting a shot that does not lead to a goal.

Confusion Matrix

1) Logistic Regression (LR)

GridSearchCV was run to identify the hyper-parameters.

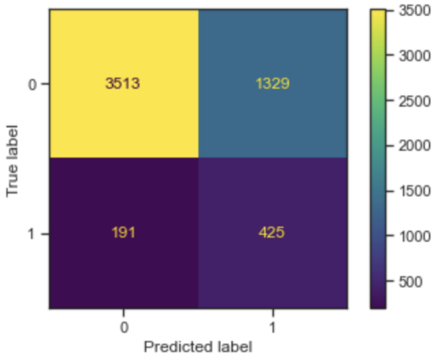


Figure 10: Logistic Regression Confusion Matrix

2) Support Vector Machines (SVM)

Given the sample size and that the time scales minimum quadratically, this would be impractical to run on the full set. As such, to optimise time spent optimizing, it was decided to run RandomizedSearchCV instead on:

- Tuning to be limited to the hyper-parameters Gamma and C.

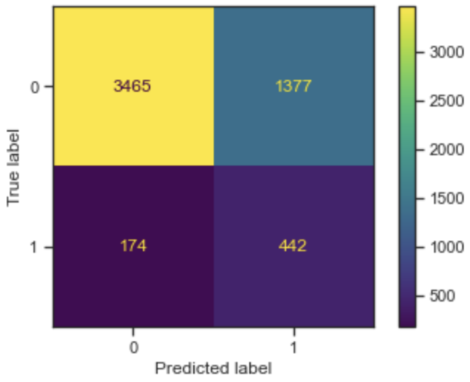


Figure 11: Support Vector Machines Confusion Matrix

3) Random Forest (RF)

For Random Forest, it was built on top of the classic Decision Tree and a follow-up to Bagging as an improvement to reduce variance as well as reduce influence of strong predictors. The hyper-parameters were tuned with RandomizedSearch before the entire training set was fit on the tuned parameters. GridSearch was not utilised here as it was computationally expensive and would not warrant the improvement in performance. Instead, RandomizedSearch was applied.

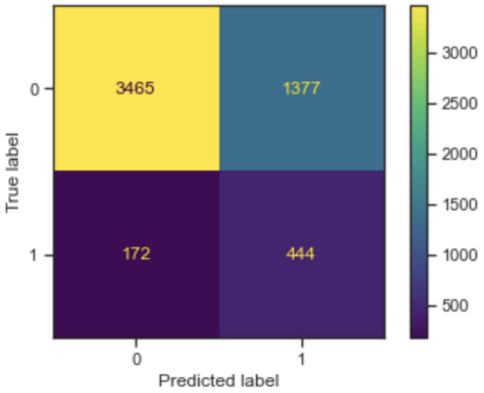


Figure 12: Random Forest Confusion Matrix

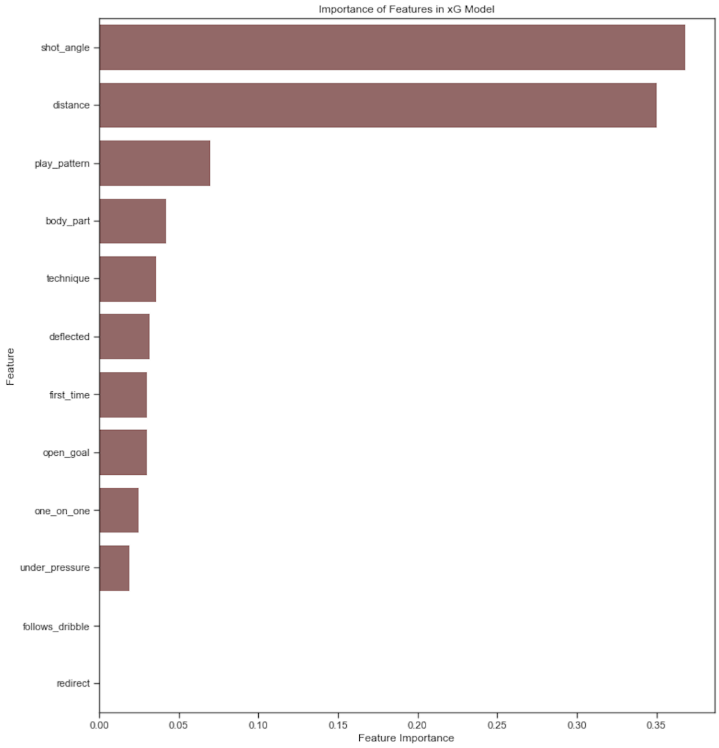


Figure 13: Feature importance with Random Forests

With Random Forests, the importance of the features takes on an interesting turn as *shot_angle* now ranks top followed by *distance* just slightly behind. Both metrics are intuitively important and might also correlate with each other as a closer distance would indicate a wider shot angle being possible which was apparent in the scatterplot during EDA. These features would thus be vital in calculating xG.

3.2 Model Evaluation

The effectiveness of the model was tested by computing their associated ROC curves (and also AUC scores) and benchmarked against StatsBomb's own xG that they used.

The 3 best performing models were narrowed down.

Table 5: Overview of top 3 performing models

Score	Random Forests	SVM	Logistic Regression
Accuracy	0.716	0.716	0.722
Precision	0.24	0.24	0.24
Recall	0.72	0.72	0.69
F-1	0.36	0.36	0.36
AUC	0.782	0.78	0.774

(vs. StatsBomb xG: AUC = 0.810)

ROC Curve & AUC

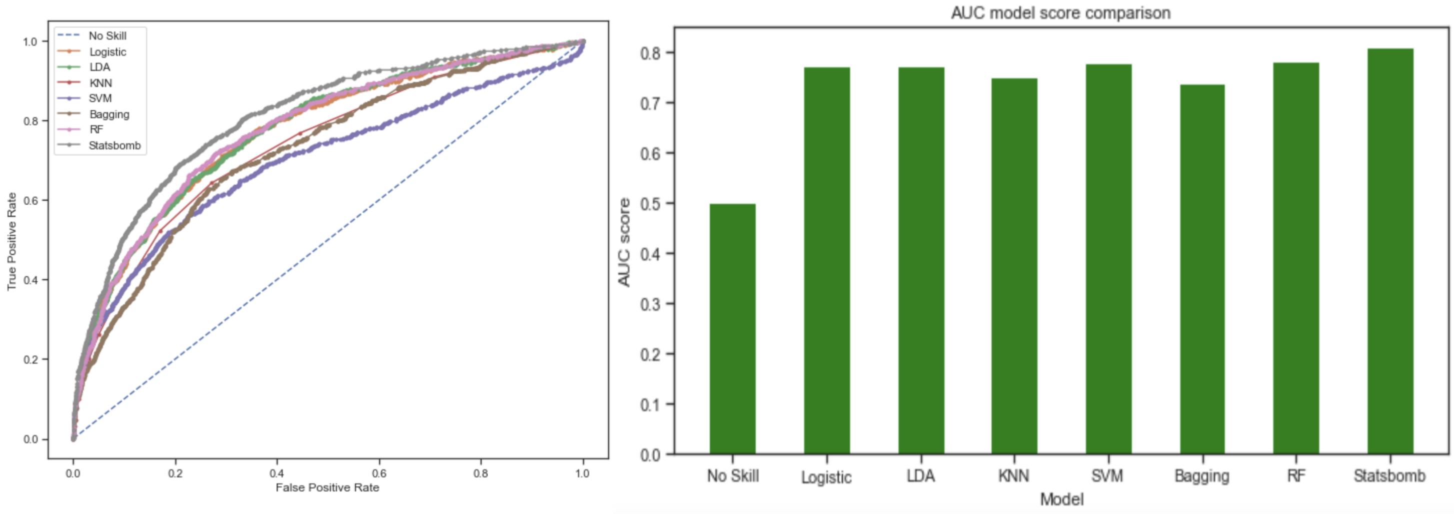


Figure 14: ROC Curve and AUC score

It can be seen from the graph that the best performing model - Random Forests - performs comparably with about a 3.5% difference to that of StatBomb's. The effectiveness of bagging and random forests over the classic Decision Tree is also evident as Decision Tree is easily outperformed. Logistic Regression (4.4%) and SVM (3.7%) hold up well too and show that even a simpler model can perform well. The difference between StatsBomb's model and ours will be further addressed below.

4. Business Application

As mentioned in the introduction, the model can be used to help teams better model the probability of scoring a goal. Whereas in the past, teams and fans would have to be content with a simplistic 'shots attempted' vs 'shots on target metric', the ubiquitousness of data particularly in sports promises huge leaps.

With a myriad of data that would in the past be left unused, it is now optimised for each team in helping to make key decisions by providing insights that might be otherwise missed. In considering the stakeholder's interest and ease of understanding, our team strongly recommends a combination of Random Forests - for predictive power together with either Logistic Regression or SVM for explanatory power. Even though they come at the cost of a 1% less accurate prediction model, it more than makes up for by offering ease of interpretability and usage.

Given the nature of xG, the probability that is obtained from the model is of itself the 'expected value'. This metric would help the teams in their analysis as they would be better able to quantify the shot that their players have made and the possible areas of improvement.

5. Conclusion, Limitation & Further Exploration

One limitation is that the specialised and extensive datasets that could more accurately model the xG is typically owned and kept secret by specialised companies in the field and understandably so given the potential benefits. Whilst we benefited from the dataset from StatsBomb, the data is not as comprehensive given that it is an open dataset (the full dataset is utilised by the company to help provide analytics for real football clubs with StatsBomb being arguably the biggest football analytics company). With that in mind however, our feature engineered data together with the other features combined to be within 3.5% of their model which is a remarkable achievement, and the model could be further improved upon as more data sets become available over time.

Another limitation for the xG model itself here is that it doesn't take into account the individual player (or his/her skill). Since the overall figure is built on a huge aggregated dataset, the end result would be an average figure.

A further exploration could be on fine-tuning further the xG model. This could arise from deep diving into the different aspects such as the type of play - if it's from an open play, a free kick, as well as the body part used to score the goal. For instance, players who are predominantly left-footed (in a right-footed dominant industry) would naturally tend to score with their left-foot. Likewise, headers could appear to have a higher xG as they might be attempted closer to the goal vs taking a shot with the foot which would be further out.

<References>

Lange, D. (2021, September 10). *Brand value ranking soccer clubs worldwide 2021*. Statista. Retrieved November 19, 2021, from https://www.statista.com/statistics/234493/football-clubs-in-europe-by-brand-value/.

Nielsen . (n.d.). (rep.). *How The World's Biggest Sports Properties Engaged Fans in 2020*. Retrieved November 14, 2021, from https://nielsensports.com/wp-content/uploads/2021/05/Nielsen-How-the-Worlds-Biggest-Sports-Properties-Engaged-Fans-in-2020.pdf.

Statsbomb. (n.d.). *Statsbomb/open-data: Free football data from StatsBomb*. GitHub. Retrieved November 12, 2021, from https://github.com/statsbomb/open-data.

"Drawing a Pitchmap - Adding Lines & Circles in Matplotlib." *FC Python*, 18 Dec. 2020, https://fcpython.com/visualisation/drawing-pitchmap-adding-lines-circles-matplotlib.