Expected Goal Analysis

Oriol Garrobé and Dávid Hrabovszki

28/08/2020

Abstract

Introduction

Trying to play the best football possible in order to win all the titles is a quest that all the best teams in the world follow. Practicing methods have changed, and therefore football istelf. Because of the evolution of technology and the rapidly increasing amount of data - as done in other sports such as baseball - football is introducing datascience techniques to its toobox in order to improve the game.

Teams use data science and machine learning approaches to analyse which aspects of the games can be improved to become the best team. Artificial intelligence (AI) in sports is very relevant nowadays, and it changed the way coaches approach their practice sessions.

From this point of view, a large number of researchers have worked on sports analytics and in particular in football trying to build the state of the art models to find the hidden insights from the sport that the human eye cannot see by analysing data.

This project aims to study the probability of scoring of a player when shooting, so professionals can use this approach to look for better positions of shooting - those situations when a player has a big chance of scoring - and avoid those that apparently look good but they have a lower chance of goal.

As mentioned, there is a bast number of papers regarding football, but there is still room fore more. There are so many aspects of the game that can be analysed, and the game can be heavily improved using AI. From this regards, this project is based in a metric from StatsBomb called expected Goal [reference this shit http://statsbomb.com/wp-content/uploads/2019/10/Benjamin-Larrousse-Improving-Decision-Making-For-Shots.pdf]. The aim is to work over this metric and try to improve it by adding some other variables and try different machine learning models that could give a better result.

The data used is the datased provided by StatsBomb itself [reference this shit]. Apart from the StatsBomb dataset, aiming to go one step further, ratings from the players in the game are used, since it is believed that are players that in the same circumstances achieve different results. This means that not only the scenario is considered in this project but also the actors. For this purpose player ratings are included to the dataset coming from the FIFA video game.

A more sophisticated approach to analyse the shot situations in a football game would clearly help practitioners to design more technical interventions and strength and conditioning programmes for players. Accordingly, it is the purpose of this study to develop and validate a machine learning algorithm to identify the best shooting situations for players.

Methods

Dataset

The project is mainly based on StatsBomb [reference this ????] dataset. This is a free dataset provided in order to new research projects in football analytics. The data from statsBomb includes very detailed and interesting features relevant for the project such as: location of the players on the pitch in any shot - including he position and actions of the Goalkeeper-, detailed information on defensive players applying pressure on the player in possession, or which foot the player on possession uses among others.

The data is provided as JSON files exported from the StatsBomb Data API, in the following structure: * Competition and seasons stored in competitions.json. * Matches for each competition and season, stored in matches. Each folder within is named for a competition ID, each file is named for a season ID within that competition. * Events for each match, stored in events. The file is named for a match ID. * Lineups for each match, stored in lineups. The file is named for a match ID.

In particular, the dataset only contains information about F.C. Barcelona games in the spanish national championship La Liga from 2005 up until 2019.

As stated earlier, player skills are also considered in this project. To have an unbiased rate of players, ratings from the most played aroung the world football video game FIFA from EA Sports [reference this] are used. Also, it is considered which is the preferred foot of the player, information sourced from the same database. More in particular, this players information is gotten from *FIFAindex*.

Data Preprocessing

Data from StatsBombs comes in JSON format. One easy way to work with JSON data in R is through the **rjson** package that transforms JSON data - which is a combination of nested lists - to nested dataframes.

The StatsBomb dataset conatining all the relevant information for the project is the **Events** dataset. Each data point in this data set is an event that occurred in a specific game, such as: pass, block, dribble or shots, among others.

The first step working with the dataset is to erase all the incomplete datapoints or those with wrong information that would afterwards make the models fail. From this point, and because this project is only focused on shots, the dataset is filtered by the variable **type** which contains the type of action of the event. This variable is filtered so only the events regarding shots remain. A **shots** dataset, way lighter than the one used before is created.

It is important to emphasize on the fact that this dataset contains an extremely useful information which is enclosed in the variable called **shot.freeze_frame**. This **shot.freeze_frame** states where are positioned on the pitch all the players at the moment of the event. It is considered that relevant since allows to make a perfect picture of the scenario at the moment of the shot.

From this regards, and using the information in **shot.freeze_frame**, a number of new features can be computed and added to the dataset. More in particular, geometric features regarding the position of the striker, the defenders and the goalkeeper are computed. Such features will allow to know, for instance, which is the distance to target of the ball, whether the goalkeeper is properly positioned or if there are defenders close enough to disturb the striker.

Finally, every data point from the *shots* dataset, containing also the geometric features computed, is complemented with information of the players. For instance, the rates of the striker and the goalkeeper in action are added or whether the player shooting is using his preferred foot or not.

Finally, some features from the **shots** dataset which are not relevant to study are removed, creating a dataset for analysis called **anal** that will be the one used to train the models. The final features in the **anal** dataset are:

Variable	Definition
id	Numeric. Number representing a unique player.
strong_foot	Boolean. States whether the player use the strong foot or not.
overall_rating	Numeric. Overall rating from FIFA ratings.
shot.power	Numeric. Shot power from FIFA ratings.
shot.finishing	Numeric. Shot finishing from FIFA ratings.
gk_rating	Numeric. Overall rating for goalkeepers from FIFA ratings.
gk.reflexes	Numeric. Goalkeeper reflexes from FIFA ratings.
gk.rushing	Numeric. Goalkeeper rushing from FIFA ratings.
gk.handling	Numeric. Goalkeeper handling from FIFA ratings.
gk.positioning	Numeric. Goalkeeper positioning from FIFA ratings.
shot.first_time	Boolean. States if the shot is the first of the game for the given player.
under_pressure	Boolean. States if the player shooting has opponents close enough to disturb the shot.
home	Boolean. States whether the player shooting is playing home or away.
dist	Numeric. distance to center of the goal from shot taker.
angle	Numeric. angle of the goal from the from shot taker (in degrees).
obstacles	Numeric. number of players (teammates & opponents NOT including GK) between goal and shot taker (inside the triangle of the goalposts and the shot taker).
pressure_prox	Numeric. Distance from closest opponent to shooter.
pressure_block	Boolean. Can the goalkeeper save the shot by being inside the triangle.
gk_obstacle	Boolean. States whether the goalkeeper is between the ball and the goal.
gk_pos	Numeric. goalkeeper's positioning, best if gk is standing on the line that halves the angle of the shot (value between 0 (angle is the same), to 1 (angle is halved).
gk_pos_adjusted	Numeric. same as gk_pos, but it is less strict with shots with a tight angle.
gk_dist_from_player	Numeric. distance between goalkeeper and shot taker.
gk_dist_from_goal	Numeric. distance between goalkeeper and the center of the goal.
goal	Binary. 1 the shot is goal, 0 the shot is not goal.

Machine Learning Models

Logistic Regression

DAVID

Random Forest and AdaBoost

Both Random Forest and AdaBoost models are very good choices when training a classification model. This models are built on the decision tree model.

The Random Forest model in particular generates a number of trees based on a bootstrapped subset of the sample and only considering a subset of variables at each step. The result is a wide variety of trees. From this point the data is run over all trees and that yield a decision, the decision given by more trees is the one that prevails. This is called Bagging. To run the Random Forest model, the package **randomForest** [somehow reference this??????????????] from CRAN is used.

AdaBoost on the other side also generates a number of trees, but only trees that have a root node and two leaves with no children, this trees are called stumps and are not great at making accurate classification, they are weak learners. From this point, once one stump is made the data is run through it, and a decision is made. The errors made by this stump influence when creating the next one and so on. Therefore, some stumps have a greater impact to the model. To run the AdaBoost model, the package **mboost** from CRAN is used.

In order to get the best model possible and not to make it too computational expensive, in both models it is important to choose the optimal number of trees. To do so, both algorithms are run a number of times with

different amount of trees. Based on the error rates, the best model in is chosen, and afterwords trained with the data to get the best results possible. In figure [whatever whatever] it can be seen the error rates against the number of trees. The smallest error rate for training data in the Random Forest model appears when 40 trees are created, and for the AdaBoost model it appears when 30 trees are used. Therefore, the final models are set.

$$HEREIWANTTOPUTACOOLGRAPHSHOWINGTHEERRORRATES$$

KNN model

The KNN (K-Nearest Neighbours) model is a classification model that identifies the number k of closest labelled points to the one studying and estimates its class. The class chosen is the one with the bigger amount of neighbors with said class. This is a powerful algorithm but it is very important to choose the optimal number of neighbours, in other words, choose the value of k. This value is chosen with Cross Validation (CV), which is a model validation which partitions the data in complementary subsets, performs analysis on one subsets and validates on the others in order to give a more accurate model. The final number of neighbours used in the KNN algorithm is 9, which gives the best result for prediction. To run the KNN algorithm with CV, the package **kknn** from CRAN is used.

Results

- **Accuracy**. Stands for the ratio of correctly predicted observation over the total observations. If it is high it means that there are a lot of good predictions but it must be taken into account that for assymetric datasets or uneven classes - those where one class is larger than the other - it can be biased. In particular, in the dataset used there are way less shots that ended up being a goal than those that not. So, with a naive predictor all the datapoints could have been set to not goal/miss and the accuracy would have been still high. That is other metrics must be used in order to take this into account.
- **Recall**. Otherwise called sensitivity, stands for the correctly predicted positive observations to all observations in an actual class. This metric checks how many properly predicted points are predicted within a class, in this case **goal**. This means that it tries to show how many predicted goals are actually goals. Since in football there are not many goals, it is a very innacurate metric, as many times a situation that should clearly end up in goal in the end is not. This is why this one is not a very indicative metric for the purposes or the project. However, it is good to compute it in order to get some conclusions.
- **F1-score**. This metric is a weighted average of precision and recall, being precision the number of properly predicted observations in a class over all the observations fom that class. From this point, it takes both false positives and false negatives into account too, it is very useful when the goal is to predict uneven classes. In this case, as there are way more observations that are not goal than those that are goal, this metric is the one that mirrors best the quality of the algorithm.

Model	Accuracy	Recall	F1-Score
xG StatsBomb	86.36%	21.79%	34.09%
Logistic Regression	85.89%	23.72%	35.24%
Random Forest	60.00%	22.05%	32.24%
AdaBoost	83.33%	11.18%	19.71%
K-NN	68.50%	27.79%	39.54%

Discusion

In this study, a Machine Learning algorithm to quantify the probability of a football player scoring is develooped, using the positioning of the players in the pitch and their skills. 4 classification approaches using

25 variables which derived from the particular scenario at the moment of the shot were examined. It has been proven that it is possible to classify whether a shot is going to be a goal or not with good accuracy, being the best performing method the logistic regression with an F-1 score of 35.24%, a recall of 23.72% and an accuracy of 85.89%. It also has improved the xG metric from statsBomb by adding player skills to the dataset and testing other Machine Learning approaches. This will provide professional of the sport such as players, coaches or managers some insights from the game that can be very useful to improve their performance.

Models based on decision trees such as Random Forest or AdaBoost provided mixed results that did not improve the previous works on the field. In particular, the optimal Random Forest algorithm with 40 trees yielded an F1-score of 32.24%, a recall of 22.05% and an accuracy of 60%. Looking at this results, it can be seen that eventhough the F1-Score is close to the one from xG, the overall accuracy is poorer, which means that many shots that were classified as miss where actually a goal. On the other hand, the optimal AdaBoost algorithm with 30 trees yielded an F1-score of 19.71%, a recall of 11.18% and an accuracy of 83.33%. All the results are worst than the xG metric, bringing no improvement to the field. This model failed as almost no goals were predicted, only 11.18% (recall) of the goals were predicted properly. From this point of view, these two algorithms do not bring improvements to the field or relevant information that can be used to improve the game.

Another model used to classfy goals is the K-NN algorithm. This algorithm yielded better results than models based on decision trees with an F1-score of 39.54%, a recall of 27.79% and an accuracy of 68.50%. It has the best F1-Score of all models used and also improves the xG metric. As stated before, the F1-score is the metric that mirrors best the quality of this algorithm, and this method could be the one chosen for the project. It also has the best recall among all the models. However, it has a relevant lower overall accuracy compared to xG or Logistic Regression. It yields a better F1-score and a better recall due the fact that predicts many more situations as goals than Random Forest, Adaboost or xG. However, many situations that were not goal were classified as goal, this is why this model has a lot of room to improve.

Finally, ### HERE WRITE ABOUT WHY WE CHOOSE LOGISTIC REGRESSION

- do something with the best model, such as showing that some players score when they shouldn't or choose the best player for the given situation

The limitations of this study must be acknowledged. The data sample consist only of **F.C. Barcelona** games in the national regular season **La Liga**. From this regards, barcelona players, which are know to be very effective and playing against most likely worst teams can lead to some biased results. It is also worth to be mentioned than as the dataset starts in 2005, many of the shots are done by **Leo Messi**, for many the best player in history and definetly on of the best strikers of all times. This can also lead to not very relevant results for the football comunity. This is why this project used player ratings, as an attempt to introduce this information to the models, that can be used therefore at all levels. It would be good therefore, to add more shot situations from many different competitions from different countries and different teams. This would give a more realistic dataset and also it would provide more data points that would definetly help train the algorithm, probably achieving better results.

Another possible improvement would be to create more variables. With the dataset provided the authors created a data set that comprised all those variables that could have an effect on the result of the shot. New variables were computed giving a proper dataset to work on. However, more variables could have been added or created. With more variables, a better picture of the situation of the shot is drawn, and by practicing **Feature Engineering** a better dataset could have been created. With this new dataset other Machine Learning approaches, more modern and complex could have been applied. For instance, would be good to train a Neural Net in order to predict goals. In order to do so high computational resources are needed, that is why Neural Nets are not used in this studio, but could probably improve the results.

Another improvement to the study would be to apply the same methods to other types of events other than **shots**, such as success in **passes** or **tackles**. This would provide information not only to the strikers but to every player in the field, improving - as this is the final aim of Football Analysis - the game.

- future work (feture engineering, neural nets, more variables???, not only shots but also passes or whatever)

- the results obtained can be used not only to improve the strikers success but also to improve the defensive positioning or to evaluate goalie skills (WHERE TO WRITE THIS?? IN DISCUSSION OR IN CONCLUSION???)
- reference some papers that are close and find differenes?? or similarities??

Conclusion

Football is a very demanding sport, and the high impact that it has in society makes teams strive for perfection. From this point, game demands are extensively analysed and new concepts coming from data science are introduced to coaches plans. This projects created a classifier that provides more information regarding the shooting positions of players. It intends to help players choose wiser when shooting, or in the other hand to advise them not to shoot when the chances are very low and a pass could create a better situation. Also it provides information, as in this project not only the situation but the players involved are studied, about which palyers should take which shots. It is commonly known that best players take more shots, but there are situations where a player which apparently is not that good is the best fit for that particular shot in that particular situation.

The algorithm, built on a simple logistic regression and only 25 relevant features classified the Scoring Probability with a combined precision and recall of 35.24% (F1-score), a recall of 23.72% and an accuracy of 85.89% improving the results of the xG study from statsBomb. This study tries to improve the football analysis comunity and football itself providing a tool to assess shooting scenarios in professional football.

References