Calculating Expected Goal Probability from Soccer Shots with Machine Learning

Isaac Wolters
School of Mathematical and Computational Sciences
University of Prince Edward Island
Charlottetown, Canada
iewolters@upei.ca

***Abstract*—Data and statistics play a huge role in many different fields, including sports. In soccer, the expected goal statistic (xG) is used to estimate the probability of a goal being scored from a particular shot or action. This paper presents a machine learning approach to calculating a shot's xG. Various regression models are used to predict the xG value based on a number of features, such as shot location, type, and situation. These models are evaluated using the mean absolute error and the R-squared value, among other techniques. Polynomial regression provided the best model. It had a mean absolute error of 0.0574 and an R-squared value of 0.5471. The success of this model gives us a good base to build off of in future research and analysis.**

***Index Terms*—machine learning, regression, expected goals, xG, soccer**

I. Introduction

Soccer is a huge part of my life. I play university-level soccer, as well as ref, coach and watch the sport daily. I also enjoy fantasy sports, which is all about using stats and trying to predict how certain players will do. I wanted to find a data set that would allow me to explore some aspect of the sport that consumes so much of my life.

In recent years, data and statistics have increased in popularity and usage in sports, including soccer. There are teams in the top professional soccer league in the world that use data as the principal method for analyzing players to recruit. Instead of buying players who are widely known but too expensive, they opt to sign players who might be less known but whose statistics show that they are and would be a good player [1]. Similarly, in the movie, Moneyball, a professional baseball team with a low budget recruits players based on data and performs well above expectations [2]. These cases are not unique, as the majority of professional sports teams use data in some way to influence their decisions regarding player and team performance.

One particular statistic in soccer that has drastically increased in usage is called expected goals, or xG. The expected goals stat indicates the probability of a goal being scored from a particular shot or action. There are many different xG models, but they mainly consider the position of the shot, the shot type, and the situation of the shot. Some models even consider the position of the goalkeeper and defenders on the opposing team [3]. For example, a shot taken from 6 meters directly in front of the goal with a player's stronger foot would have a high xG, say 0.65, while a shot taken off to the side from 20 meters away might only have an xG of 0.1. Expected goals provide an opportunity for teams to analyze player and team performance. From a player's perspective, if a striker is scoring fewer goals than their total xG, they know that they need to improve. Looking at it from a team's point of view, if their total xG is higher than their opponents in a particular game, then they know that they had a good chance of winning that game.

As this is my first machine learning project, the goal of the research is not to revolutionize the sport or improve upon an advanced statistic that has already been created. Rather, the goal of this research is to experiment with machine learning models and understand how they work. This research also hopes to help people understand an advanced statistic such as xG, and what factors influence it in various ways. With positive results, this research could lead to further research and analysis on various aspects within sports.

The next section describes in more detail the data set, as well as the machine learning models and techniques used. Following that, section III displays the results and analysis of the models, while section IV discusses the outcome and concludes the paper.

II. Background

The data set that will be used to predict a soccer shot's expected goal probability contains information on almost 300 000 shots from different games and leagues between 2014 and 2020. Some of the features in the data set include a shot's X and Y coordinate, the shot type, a right footed shot, for example, and the shot situation, a shot from a direct free kick, for example. It also includes features that are not used, such as the minute of the goal, the teams that are playing, which team took the shot, among others. While this data set does not have information on defender and goalkeeper positions which would allow for extremely precise xG predictions, it contains enough information to explore the stat, see what influences it, and examine various machine learning models and techniques while doing so.

The X and Y coordinate features of the data set are values between 0 and 1. The higher the value of the X coordinate, the closer the shot was to goal. The Y coordinate measures across the width of the field. If the Y coordinate is 0.5, it indicates the shot came from a central position. I decided to create a new

feature of Y sections, where the Y coordinates were split into 3 different categories: wide, channel, and central. Y coordinates between 0.4 and 0.6 were classed as central, coordinates between 0.2 and 0.4 and between 0.6 and 0.8 were classed as channel points, and the points outside of 0.2 and 0.8 were classed as wide points. During the linear regression analysis, I found that the Y coordinate coefficient was very low, and this was due to the model considering the Y coordinate as a line, rather than a normal distribution shape. You would expect the coefficient of the Y coordinate to be one of the higher values, as the more central the shot, the more chance it has of being a goal. By splitting the Y coordinates into the 3 channels and using those groups, it removed any linearity from the Y coordinates, which resulted in a much more accurate coefficient to what would be expected. Looking at figures I, II, and III, the effect of the X and Y coordinates on xG are visualized. It is clear to see that the closer to goal and the more central the shot is, the higher the xG value. The shotType feature indicated how the shot was taken. The categorical values were RightFoot, LeftFoot, Head, and OtherBodyPart. Similarly, the situation feature was categorical, with values of DirectFreekick, FromCorner, SetPiece, and OpenPlay. Figures IV and V reflect these features' effects on xG respectively. In the modeling, the categorical features were converted into numerical columns with the pandas get_dummy_variables method. This created separate columns for each of the possible values of the features, with a 1 and 0 indicating if that particular value occurred or if it had not, respectively. Doing this allowed me to perform regression on the data set.

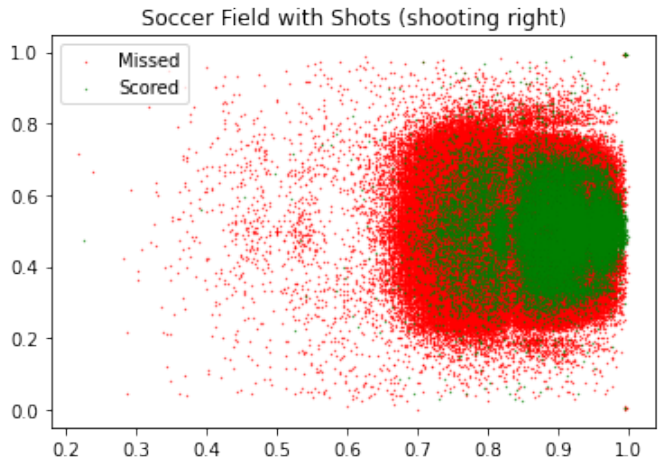

Fig. 1. Shots on soccer field

Throughout this analysis, regression models were needed, as the target feature, xG, is a continuous value from 0 to 1. I began with linear regression to give a simple baseline to compare with other regression models that were used later. Linear regression is a very simple machine learning model that finds the best fitting line through the target variable data based on the input data. This line can then be used to predict future target values when given new input data [4].

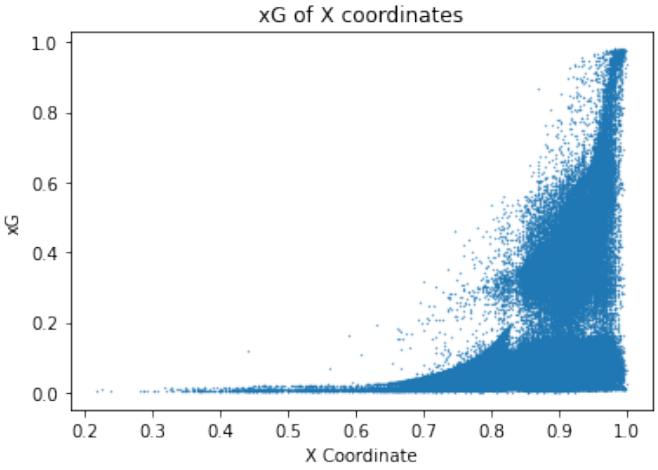

Fig. 2. X coordinate's effect on xG

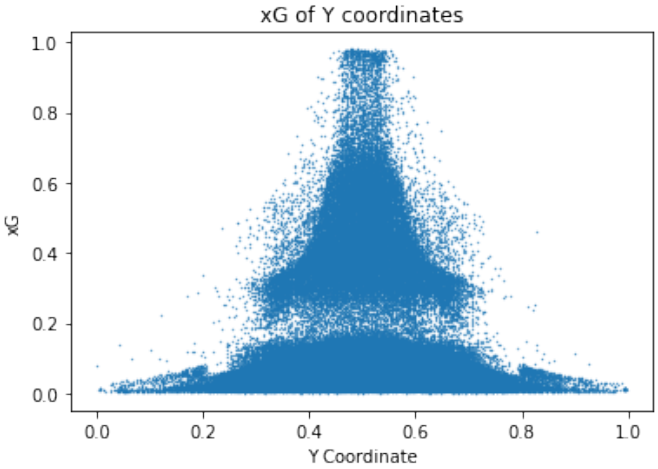

Fig. 3. X coordinate's effect on xG

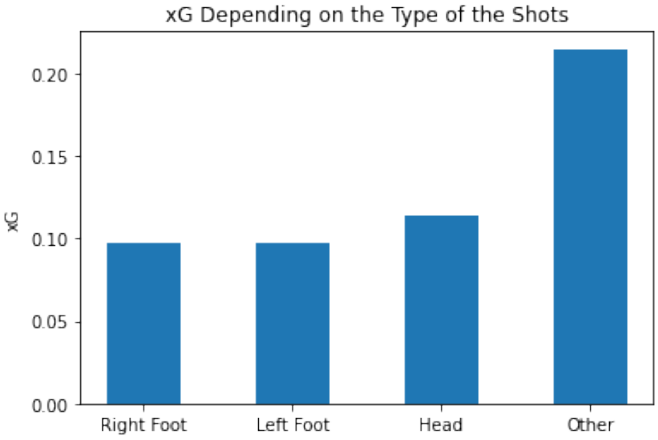

Fig. 4. Shot type's effect on xG

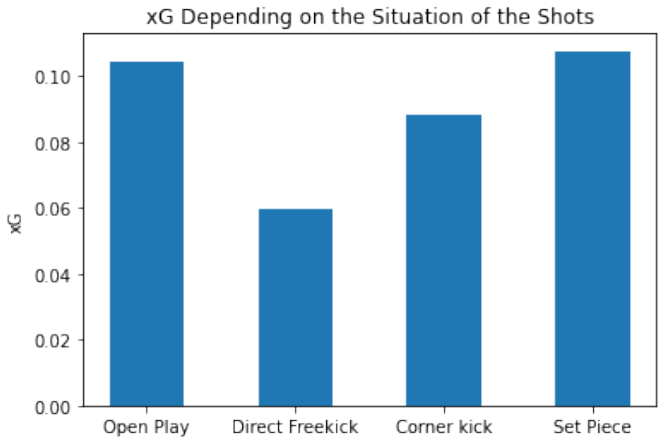

Fig. 5. Shot situation's effect on xG

The linear regression model was used to perform manual feature selection, which indicated that using all four features resulted in the most accurate model. The linear regression model was also used to perform dimensional reduction. This is a process of combining features in order to reduce the number of parameters in the model. This allows the model to be more computationally efficient and occasionally improved performance [5].

After the linear regression model, other regression models were used to compare models and determine the best. A polynomial regression model was used along with parameter tuning to determine the ideal degree of the polynomial to use. Polynomial regression is a similar machine learning model to linear regression, although it allows a polynomial function to be fitted to the data, rather than just a line [6]. This polynomial regression model was then combined with dimensional reduction, using the ideal number of dimensions that was previously determined. Next, simple ridge regression and lasso regression models were explored. Finally, a logistic regression model was created. Logistic regression is different to all of the previously-mentioned models, as it is a regression model that is mainly used as a classifier. Logistic regression uses the sigmoid function as it's form of regression, instead of a line or polynomial function. If the model predicts a value of over 50%, it is classified in one group, and if the model predicts a value of under 50%, it is classified in the other group. This concept can also be extended to multiple classes, however that is not needed for this project. Logistic regression also provides a probability for its classification, which is what will be used to predict the expected goal values [7].

Multiple techniques were used to analyze and compare these models. First, the mean absolute error provided the mean difference from the predicted values to the values being predicted [8]. The goal is to minimize the mean absolute error in the models. The mean absolute error was favoured over the mean squared error, as squaring a value between 0 and 1 would actually result in a smaller error, which was not desired. It is important to note that these models result in very low mean absolute errors. This is due to the target feature, xG, which is a probability, or percentage. As we are predicting a percentage, the mean absolute error can also be thought of as a percentage. For example, if our mean absolute error is 0.08, it indicates that our mean absolute error is 8%. R-squared values were also computed for these models. The R-squared value gives a measurement as a percentage reflecting the fit of the regression model, and how much variation of the dependent variables that can be explained by the independent variable [9]. The goal is to maximize the R-squared value of the models. The third method that was used to evaluate these regression models was a residual plot. The residual is the difference between the predicted target value and the actual target value for a particular case. When plotted, residuals should be dispersed across the horizontal axis and as close to the horizontal axis as possible [10]. These three techniques were used to analyze the machine learning models and select those that performed the best.

III. Results and Analysis

A. Linear Regression

First, a linear regression model was created to predict the xG values. This initial model resulted in a mean absolute error of 0.0799 and an R-squared value of 0.3583. While these numbers are not great, they provide us with a baseline to compare future models with. A residual plot was also created for the linear regression model, as seen in figure 6. The plot is quite skewed to the right, with a clear line extending up and to the left. Ideally, the residual plot is normally distributed across the horizontal axis. Overall, the linear regression model is not the best way to predict expected goals, however, we use this model to apply further machine learning techniques in the form of feature selection, dimensional reduction, and learning curves.

1) Feature Selection: The linear regression model was used to perform manual feature selection. As mentioned in the background, the mean absolute error and the R-squared value are two of the primary statistical techniques of analyzing performance. As you can see in Table I, using all four columns, X, Y (in sections), shotType, and situation, resulted in the linear regression model having both the lowest mean absolute error and the highest R-squared value. This shows that all four columns are useful to the model and that none of them need to be removed.

TABLE I
Feature Selection Results using Linear Regression

Table Features	**Measures**	
	MAE	***R-Squared***
All	0.0799	0.3583
No shotType	0.0805	0.3199
No situation	0.0809	0.3436
No shotType or situation	0.0813	0.2859

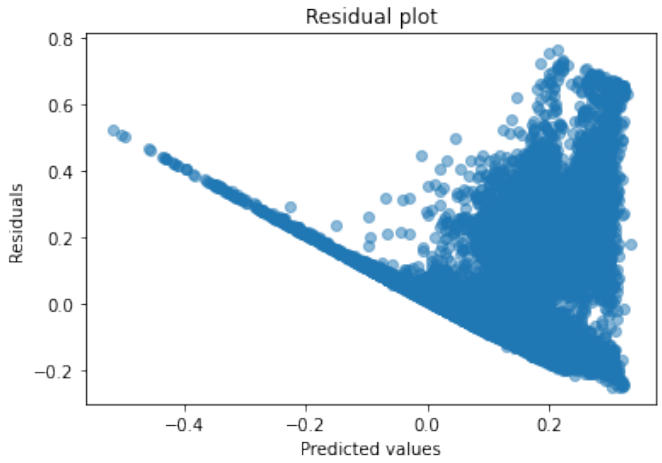

Fig. 6. Residual plot of linear regression model

2) Dimensional Reduction: Next, dimensional reduction was performed on the linear regression model. Again, the mean absolute error and the R-squared value were used to analyze the results and determine the best dimensional reduction.

Figure 7 and figure 8 both represent the results of the dimensional reduction. Figure 7 plots the mean absolute errors for each linear regression model against the number of dimensions, while figure 8 does the same thing with the R-squared value. Looking at the figures, maintaining all nine of the dimensions would be the best option, however an elbow in both of the plots can be seen with a dimension of six. This will become useful later in our results as we attempt to optimize the polynomial regression model.

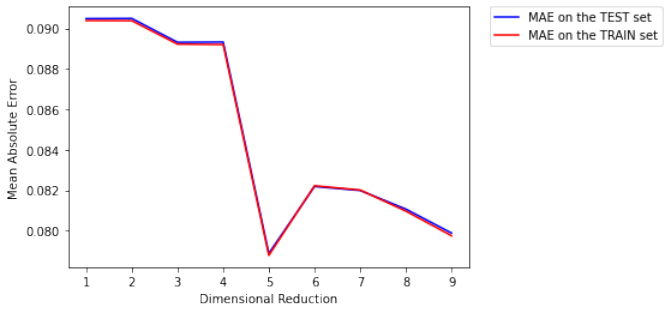

Fig. 7. Mean Absolute Error of dimensional reduction

3) Learning Curve: To summarize so far, the linear regression model using all four columns and all nine dimensions, once split using dummy variables, remains the best predictor. Figure 9 shows the linear regression model's learning curve to examine any potential under-fitting or over-fitting. The figure becomes very narrow as the training size is increased, which indicates that the model does not suffer from over-fitting. However, the mean absolute error always stays relatively high, which indicates a high bias and under-fitting in our model. To work around that, other regression models will be examined

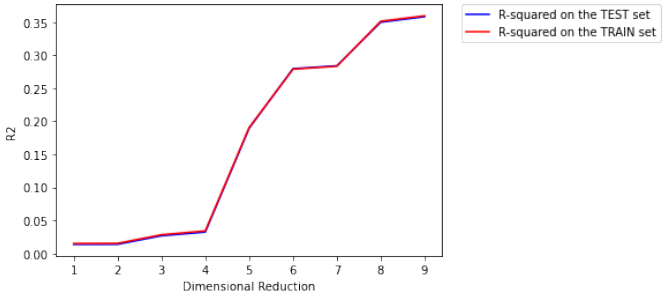

Fig. 8. R-squared of dimensional reduction

to determine the best model for predicting xG.

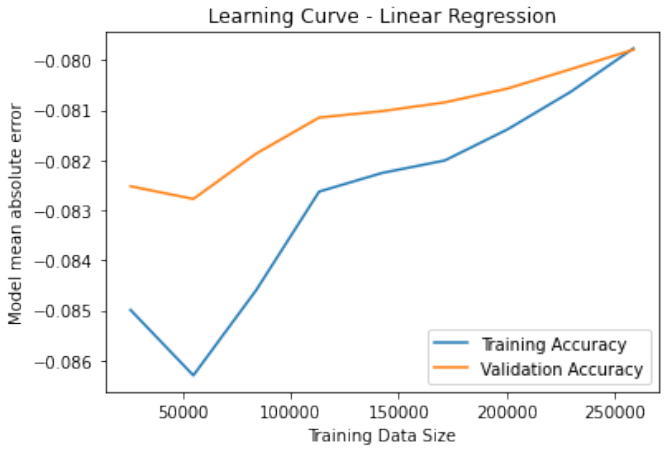

Fig. 9. Learning curve of linear regression model

B. Polynomial Regression

The analysis began with linear regression to provide a simple base model. Looking at the learning curve, the linear regression model suffered from under-fitting, so other models are needed to try and reduce the bias. The next model to try is polynomial regression. The expected goal stat is certainly not a linear value. Instead, xG could be represented by a right skewed graph, where the majority of shots have a lower xG. Polynomial regression will be used to determine if this skew can be captured in a model.

First, parameter tuning was used on the degree of the polynomial regression model to determine the best degree to use. Unfortunately, the calculation time exponentially increased and errors were encountered whenever the degree increased over three. I attempted to combine this with the dimensional reduction that was performed in the linear regression section. This would reduce the complexity of the polynomial model and allow more degrees to be examined. As mentioned in the linear regression section, reducing the dimensions to six would be the best choice to maintain the most variance possible.

Combining polynomial regression with dimensional reduction allowed the parameter tuning to explore degrees up to and including three this time. Table II shows the results of these

polynomial regression models and their corresponding mean absolute errors and R-squared values.

TABLE II
POLYNOMIAL REGRESSION DEGREE TUNING WITH SIX DIMENSIONS

Table	Measures	
Degree	*MAE*	*R-Squared*
1	0.0822	0.2797
2	0.0653	0.4959
3	0.0574	0.5471

Looking at the results from Table II, using a degree of three with the dimensions reduced to six results in a much lower mean absolute error to the linear regression model (decrease of 0.0225) and a much higher R-squared value to the linear regression model (increase of 0.189). This is a significant improvement to our existing model. Looking at this new polynomial regression model's residual plot in figure 10, the residuals are much more normally distributed across the horizontal axis, although it is not perfect.

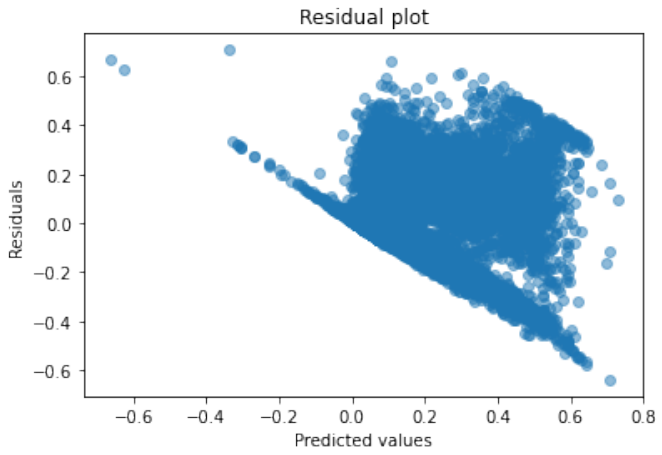

Fig. 10. Residual plot of polynomial regression model

C. Ridge and Lasso Regression

Next I experimented with some lesser known regression models to see how they would perform. In short, they performed very similarly to the linear regression model. They resulted in similar mean absolute errors, R-squared values, and residual plots. While these results may not fully demonstrate the full capabilities of these particular regression models, they seemed similar to the linear regression model, and they showed no signs of reaching the levels of the polynomial regression model with a degree of three and having six dimensions.

D. Logistic Regression

The final regression model that was used to calculate the xG of shots from the data set was a logistic regression model. Although logistic regression is primarily a classifier, it calculates a probability as a result and classifies based on that probability. To use the logistic regression model to calculate an expected goals value, I needed to do things differently than the other models. First, instead of the target value being the xG value of the shot, the target value for this model was a classification of the shot resulting in a goal or not. Next, I needed to split the data set twice, with the same random state. For the first split, I used the xG value as the target feature, and for the second split, the goal classification was used as the target feature. This allowed me to fit the logistic regression model and predict whether the shot would be a goal or not, as well as give me the probability that it would result in a goal. Looking at the definition of expected goals, this probability is exactly what we are looking for, so using the first split that gave the shot's xG, I was able to compare the predicted probability of the shot being a goal with the actual xG value of the shot using different splits with the same random state.

The logistic regression model's results showed that it was the second best predictor, just behind the polynomial regression model. This is likely due to the fact that the logistic regression model only predicts xG values between 0 and 1, as it actually calculates a probability as opposed to some of the other models. This model had a mean absolute error of 0.0592 and an R-squared value of 0.5093. Looking at figure 11, the logistic regression model's residual plot was actually the best out of all of the models, again due to only predicting values between 0 and 1. Despite being primarily a classifier, the logistic regression model was one of the best predictors for a shot's xG.

Feature coefficients show which features have the most influence on the model. The coefficients do not make sense when you reduce the dimensions like what was done with the polynomial regression model, as the features are combined and lose their readability. However, for the logistic regression model, the dimensions were not reduced, so the coefficients provide a good picture of the features that influence a shot's expected goal value the most. These coefficients can be seen in Table III. Note that due to the dummy variables, the columns one of those dummy variables are based off of a primary category for that column. In this case, the situation coefficients vary from a base value of direct free kick situations, the shot type coefficients vary from a base value of headed shots, and the Y coordinate section coefficients vary from a base value of central shots. As seen in Table III, the X coordinate is by far the most influential feature. After that, the features are all fairly similar, but Y_central, indicating shots taken from a central area were also quite influential to the model.

E. Analysis

Finally, we will review our most influential models. Looking at table 3, we can see that our simple linear regression model provided a good base to start with, but was not as accurate as the other models. Next, the polynomial regression model with a degree of three and having six dimensions had the lowest mean absolute error and the highest R-squared value, closely followed by the logistic regression model. However, looking back to their residual plots in figures 10 and 11 respectively, the logistic regression model had a slightly more normally

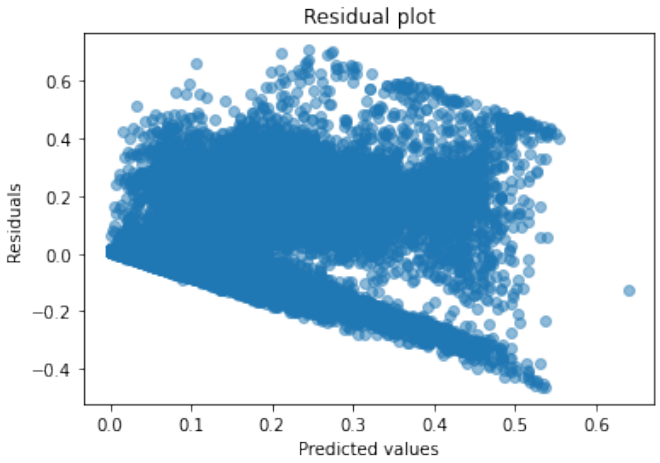

Fig. 11. Residual plot of logistic regression model

TABLE III
LOGISTIC REGRESSION COEFFICIENTS

Feature	Coefficient
X	13.6975
Y_Central	0
Y_Channel	−1.1467
Y_Wide	−1.4746
situation_DirectFreekick	0
situation_FromCorner	−1.3860
situation_OpenPlay	−0.8397
situation_SetPiece	−1.1053
shotType_Head	0
shotType_RightFoot	0.9269
shotType_LeftFoot	0.8505
shotType_OtherBodyPart	0.4562

distributed plot, while the polynomial regression model's plot was marginally influenced by some outliers.

TABLE IV
REVIEW OF MOST INFLUENTIAL MODELS

Table Model	Measures	
	MAE	***R-Squared***
Linear Regression	0.0799	0.3583
Polynomial Regression with deg 3 and dims 6	0.0574	0.5471
Logistic Regression	0.0592	0.5093

IV. CONCLUSION AND DISCUSSION

Multiple different models were used to find the most accurate predictor of a shot's expected goal value. Out of all of the models, there were two models that stood out in terms of performance. First, the polynomial regression model with a degree of three and reduced to six dimensions resulted in both the lowest mean absolute error and highest R-squared value, with 0.0574 and 0.5471 respectively. The logistic regression model also had similar results, although it was not quite as accurate, with a mean absolute error or 0.0592 and an R-squared value of 0.5093, although the logistic regression model had the most normally distributed residual plot.

The X coordinate was undoubtedly the most influential feature, with a coefficient value of 13.6975, which was more than 10 higher than any other feature coefficient. This is clear to see from figures 1 and 2, where we visualized the data set. The closer the shot to the goal, the higher the xG value and the more likely of the shot resulting in a goal.

While these machine learning models will not replace any of the current models that are used to calculate a shot's expected goal value, that was not the expected result. The data set did not include every feature that is necessary to calculate a precise xG. For example, it did not have information on defender or goalkeeper positions. However, these models provided a good base for understanding the expected goals advanced statistic, and I was able to experiment with various regression models and try to optimize their performances. Now that I have more of a base to work with in terms of experience with applied machine learning, I will be able to achieve better results with future projects and research.

REFERENCES

[1] S. O'Brien. "How Brentford flipped the script and staged a data revolution to become England's smartest club." TalkSport. https://talksport.com/football/fa-cup/659667/brentford-data-revolution-england-smartest-club-championship-leicester-fa-cup/ (Accessed March 6, 2023).

[2] D. Surcic. "Summary: Moneyball: The Art of Winning an Unfair Game by Michael Lewis." Wordsrated. https://wordsrated.com/summary-moneyball-the-art-of-winning-an-unfair-game-by-michael-lewis/ (Accessed March 6, 2023).

[3] "What Are Expected Goals (xG)?" StatsBomb. https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/ (Accessed March 6, 2023).

[4] A. Bolufe-Rohler. (2023). Linear Regression. [Plain Document Text]. Available: https://moodle31.upei.ca/pluginfile.php/933317/mod_resource/content/3/Lecture%202%20-%20Linear%20Regression.pdf

[5] A. Bolufe-Rohler. (2023). Dimensionality Reduction. [Plain Document Text]. Available: https://moodle31.upei.ca/pluginfile.php/933358/mod_resource/content/3/Lecture%2011%20-%20Dimensionality%20reduction.pdf

[6] A. Bolufe-Rohler. (2023). Polynomial Regression and Cross-validation. [Plain Document Text]. Available: https://moodle31.upei.ca/pluginfile.php/933326/mod_resource/content/5/Slides%20Lecture%204%20-%20ML%20concepts.pdf

[7] A. Bolufe-Rohler. (2023). Logistic Regression. [Plain Document Text]. Available: https://moodle31.upei.ca/pluginfile.php/933340/mod_resource/content/4/Lecture%207%20-%20Logistic%20Regression.pdf

[8] "Glossary." C3. https://c3.ai/glossary/data-science/mean-absolute-error/ (Accessed March 7, 2023).

[9] "Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?" Minitab. https://blog.minitab.com/en/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit (Accessed March 7, 2023).

[10] U. Gohar. "How to use Residual Plots for regression model validation?" Medium. https://towardsdatascience.com/how-to-use-residual-plots-for-regression-model-validation-c3c70e8ab378 (Accessed March 7, 2023).