# Data Analytics in Football: Defining the Playing Styles of Singapore PL Teams

Jadhav Chaitanya Dhananjay
School of Computer Science and Engineering (SCSE)

Asst Prof Komar John
Physical Education and Sports Science, NIE

***Abstract -*** Modern football has many tactical approaches. The optimal playing style for a team depends on many factors - for example, the quality of their players, opposition, and game state. Coaches have the responsibility of identifying the best playing style for their team to win. This study aims to identify the most effective playing styles in the Singapore Premier League. The primary playing styles that are identified are Maintenance, Build Up, Sustained Threat, Direct Play, Counterattack, High Press, Fast Tempo, and Crossing. The raw data used for this study consists of all game events (passes, shots, tackles), their location on the pitch, and the percentage of time that a team spent in each playing style for each game in the 2019-2021 seasons. Using Python programming, we can create an Expected Goals model using shot data that can objectively evaluate the quality of a shot and probability of it being a goal. By applying this model on our event data, we can get an understanding of the quality of a shot alongside the patterns of play that a team used to arrive at it. After collecting this data, we can use regression trees to identify combinations of playing styles that lead to a high expected goals value. This provides us with a set of rules that coaches in the Singapore Premier League can follow to create effective chances. This data is also useful for teams during match preparation to easily identify the strengths and weaknesses of their opponents.

**Keywords –** Expected Goals; Playing Styles; Data Science; Python Programming; Sports Science

## 1 INTRODUCTION

Data analytics has revolutionized modern football. Today, it is used in almost every aspect of a club's activities, from coaching to scouting and recruitment. A strong understanding and application of data science enable teams to overcome financial inequalities and field competitive line-ups at a fraction of the cost. In leagues with a small number of teams, establishing a competitive advantage in this field can create a stable base that can ensure prolonged periods of success. Our motivation is hence to begin laying these foundations by performing preliminary analyses to find trends behind the performances of teams.

In this paper, we will focus on identifying effective playing styles in the Singapore Premier League using Regression Trees. To do so, we will first create an Expected Goals model that can objectively evaluate the quality of all shots. Expected Goals is a metric that aims to quantify the probability of a team scoring or conceding a goal. It is measured on a scale of 0 (impossible to score) to 1 (certain goal). It has gained popularity over the years as teams started adopting it as a metric for measuring performance and conducting recruitment (Rathke, 2017).

Using Regression Trees on shot data with their computed playing styles, we can identify sets of "rules" that lead to effective outcomes. Further analysis on these trees can reveal the strengths and weaknesses of individual teams.

## 2 METHOD

### 2.1 DATA SOURCE

The data for this study originates from games played in the Singapore Premier League from the 2019-20 to the 2021-22 season (IRB-2021-580). Event data for every game is stored as a separate CSV file, with each row representing an event that has taken place during the game. Events include, but are not limited to: passes, shots, saves, goals, tackles, and fouls. The type of the event is recorded alongside its location in x and y coordinates, the time in seconds it occurred, and the player and the team that performed the event.

### 2.2 ISOLATING SHOT DATA

To create an Expected Goals model, we must first extract all shots that were taken from our data set. This can be done by filtering rows which have event IDs referring to shots and goals. Using Python, files were created for all possessions in each season containing the following event IDs:

- 8010: Goal
- 4020: Wide Shot
- 6010: Shots Blocked
- 4040: Shots Blocked

- 4010: Shot on Target
- 4030: Shot into the Bar / Post
- 4050: Shot blocked by Field Player

This provides us with a list of all shots that were taken in each season.

Further processing was carried out before creating an Expected Goals model. The dimensions of pitches in Singapore are 105x68 metres, as observed from event data. To avoid shots taken in the second half of the pitch (52.5 metres onwards) from negatively affecting the weights of the training model, we normalized all shot data to use the goal in the first half as the frame of reference. This was achieved by subtracting 68 from the y coordinate and subtracting 105 from the x coordinate for all shots taken in the second half of the pitch.

Further filtering was carried out to remove all shots that originated from set pieces (free kicks, penalties, and corners) and headers. These can disproportionately affect the coefficients of the trained model. The final dataset is a list of all open-play shots in the Singapore Premier League from 2019-2021. We can now use this dataset to create an Expected Goals Model.

### 2.3 EXPECTED GOALS MODEL

Logistic regression is the optimal statistical tool for creating an Expected Goals Model. It can maximize performance with small training data sets (Leuven, 2020) and is well suited for modelling the likelihood of binary outcomes (in this case, goals, or misses) based on a set of predictor variables.

The model outputs log odds for scoring a goal – defined as the natural logarithm of the probability of success divided by the probability of failure (Rotella, n.d.). Logistic regression outputs coefficient values that indicate the relationship and importance between each predictor variable on the log odds of scoring a goal. Logistic regression training methods can also use both qualitative and categorical predictor variables for fitting, making it the perfect fit for the wide range of event data that we have.

The Expected Goals model was created using the statmodels Python package (Seabold & Perktold, 2010). The methodology used to create the model is as follows:

1. Round the x and y coordinates of all shots to one decimal place.
2. Remove entries from the training data set that do not have a x or y coordinate.
3. Drop shot values that are too close to the boundaries:
    a. Shots with a x coordinate greater than 40.
    b. Shots with a y coordinate greater than 60.
    c. Shots with a y coordinate lesser than 10.
4. Create a binary variable “goal_label” that marks all shots leading to goals as 1 and all misses as 0. This will be used as the response variable for fitting the model.
5. Calculate the Euclidean distance of all shots from the centre of the goal with coordinates (0,34). This is done using Python’s math module and dist function.
6. Calculate the angle in degrees between the shot location and the centre of the goal with coordinates (0,34). This is done using Python’s math module and atan2 function.
7. Using the statmodels package, fit the dependent variable “goal_label” against the independent variables:
    a. Pos_X: The x coordinate of the shot
    b. Pos_Y: The y coordinate of the shot
    c. Distance: Calculated in Step 5
    d. Angle: Calculated in Step 6

The trained model outputs the following information:

1. The intercept term of the model
2. The coefficients for each predictor variable. These represent the change in log odds of scoring a goal for a unit increase in each variable.

This output can subsequently be used to derive a formula for the probability of scoring a goal with coefficients for each predictor variable.

### 2.4 REGRESSION TREES

Now that we have a means to evaluate the quality of a shot, we can begin analysing possession data to identify effective combinations of playing styles.

The data for this portion of the analysis is sourced from a previous study which classified every possession in the 2019-2021 Singapore Premier League seasons into eight playing styles (Ong, 2022). The playing styles identified were Maintenance, Build-up, Sustained Threat, Fast

Tempo, Direct Play, Counterattack, Crossing and High Press.

For each event, the percentage of time spent in each playing style was recorded. This was accomplished by dividing the pitch into different sections and measuring the amount of time spent in each of them.

By filtering all shots from this dataset and calculating the Expected Goals for each shot, we can obtain information about the styles of play in a possession that leads to the creation of effective chances.

Regression Trees are subsequently used to find combinations of playing styles that lead to high Expected Goals. The playing styles breakdown for each possession are used as the "X" (training) variable, with the Expected Goals value as the "Y" (target) variable.

Following the decision nodes for a trained Regression Tree to a leaf node can reveal:

1. The characteristics of this possession that lead to a favourable outcome.
2. The Expected Goal value for all shots that have this style of play.
3. The sample size for this occurrence.

The Regression Trees were created using the DecisionTreeRegressor function in the scikit-learn Python package (Pedregosa et al., 2011). The methodology used to generate them is as follows:

1. Import playing styles data.
2. Filter all shots and goals from event data.
3. Find the expected goals of all filtered possessions by using the formula derived from the Logistic Regression model.
4. Generate all possible combinations of 3 playing styles using the itertools module in Python.
5. For each combination, fit regression trees of lengths 3-5 on the possession data:
    a. X values – playing styles of combination.
    b. Y value – expected goals of possession.

The resulting set of decision trees provide a comprehensive series of classifications that can be analysed to identify trends in effective playing styles.

# 3 RESULTS

## 3.1 EXPECTED GOALS MODEL

Table 1 – Coefficients for Independent Variables for the Expected Goals Model

| Variable Name | Coefficient |
|---|---|
| Intercept | 0.2204 |
| X Coordinate | -0.0281 |
| Y Coordinate | -0.0062 |
| Distance | -0.0998 |
| Angle | -0.0081 |

The above table shows the intercept and the coefficients for each independent variable from the trained model. Negative coefficients for each of our variables indicate that there is an inverse relationship between each variable and the Expected Goal value.

The value of the coefficients also reflects feature importance. For our model, distance from the goal is the primary determining factor, as it has the largest coefficient value. The other variables, ranked in terms of importance are: X Coordinate, Angle and Y Coordinate.

Equation (1) shows how the coefficients can be used to calculate the probability of a goal being scored.

$$P(goal) = 1 / (1 + exp(-(0.2204 - 0.0281 * pos\_x - 0.0062 * pos\_y - 0.0998 * distance - 0.0081 * angle))) \quad (1)$$

This trained model can be provided to Singapore Premier League teams in the form of a web application for them to track their performance and quality of chances. A proof-of-concept for this is available at https://indicium15.github.io/xG.html, which allows a user to interactively change the position of the ball on the first half of the pitch and view the resulting Expected Goals value.

Figure 1 below shows a scatter plot of all shots from our dataset shaded with their Expected Goals value. The darker the shade, the higher the Expected Goals of the shot. Figure 1 and all subsequent plots were made using the Seaborn and Matplotlib Python packages (Hunter, 2007; Waskom, 2021).

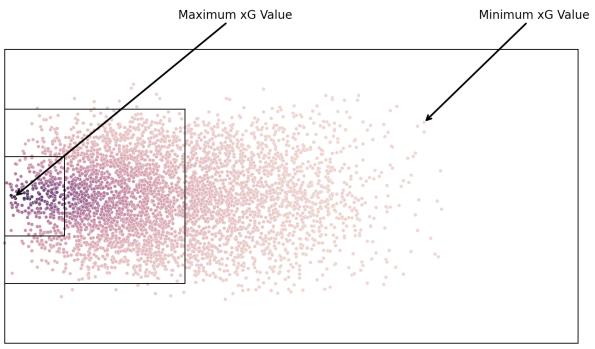


Figure 1 Scatter Plot of all Shots Shaded with Expected Goal Values.

Applying this formula on all shot data reveals that the shot with the highest Expected Goals was at coordinates (0.9, 33.9) - right in front of goal - with a 47.2% chance of scoring. Consequently, the shot with the lowest Expected Goals was at coordinates (38.4, 51.1) – near the halfway line, with a 0.43% chance of scoring. This shows that our model is working as intended.

We can also create a heat map visualization of Expected Goal values for all possible shot coordinates in the first half. Figure 2 below shows the result of this calculation. The heap map has contours which connect areas on the field with the same Expected Goals value.

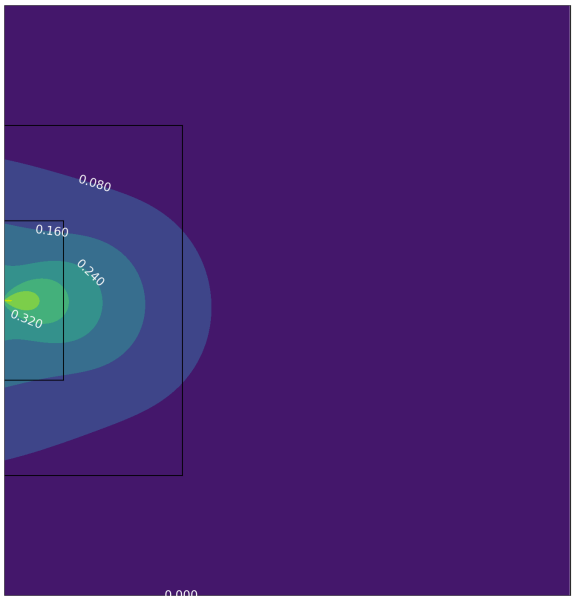


Figure 2 Heat Map of Expected Goals Model.

Our model assigns nearly all shots outside of the penalty box with an almost zero percent chance of scoring. This can be attributed to the relatively small sample size used for training the model and the removal of shots taken from free kicks.

The accuracy of this model can be iteratively improved in the future as more event data is collected for subsequent seasons of the Singapore Premier League. Separate models can also be made for other game states such as set pieces, penalties and headers once enough data is collected.

### 3.2 REGRESSION TREES

Before analysing the output from the regression trees, we must first set a benchmark for the minimum Expected Goals value for a shot to be considered effective.

Figure 3 below shows a histogram of the Expected Goals of all shot data from the regression trees.

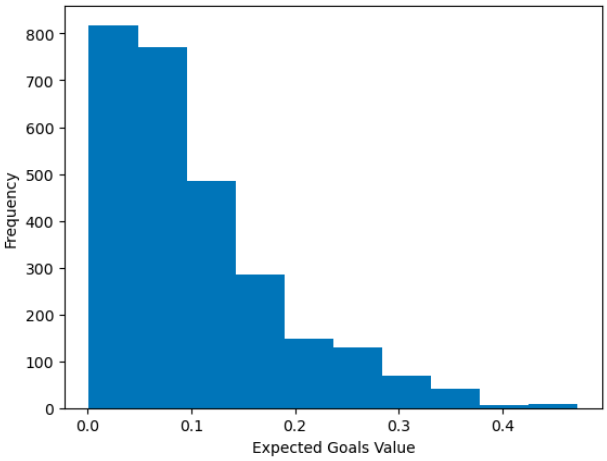


Figure 3 Histogram of Expected Goals of all Training Shots.

Using the pandas describe function, we can determine that the 75$^{th}$ percentile of data is 0.143 (Mckinney, 2010; Reback et al., 2021). We disregard values above 0.350 as the sample size of shots with these outcomes is too small to draw any definitive conclusions between the style of play and outcome of the shot.

Regression Trees of depths 2, 3 and 4 were generated for every combination of variables as described in the method. However, further inspection revealed that trees of depth 2 did not provide sufficient classification of data to reveal any actionable insights. An example of a tree of depth 2 generated can be seen in Figure 4 below.

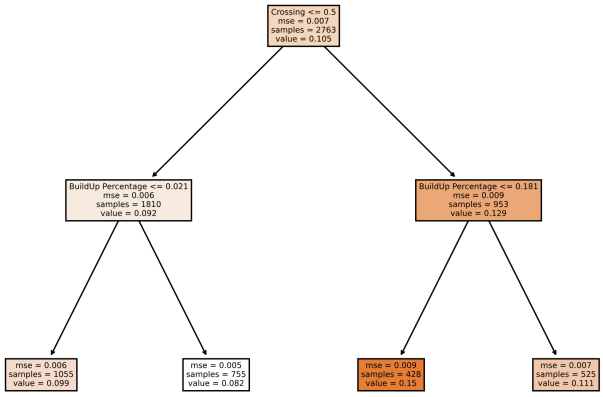

Figure 4 Regression Tree of Build-up Percentage, Direct Play Percentage and Crossing - Depth 2.

On the other hand, regression trees of depth 4 tend to overfit data as there are too many leaf nodes generated. Furthermore, each leaf node has too small of a sample size to draw any meaningful observations. Therefore, for this analysis, only regression trees of depth 3 were considered.

Figure 5 below shows an example of a regression tree of depth 3 generated from the variables Crossing, Threat Percentage and High Press.

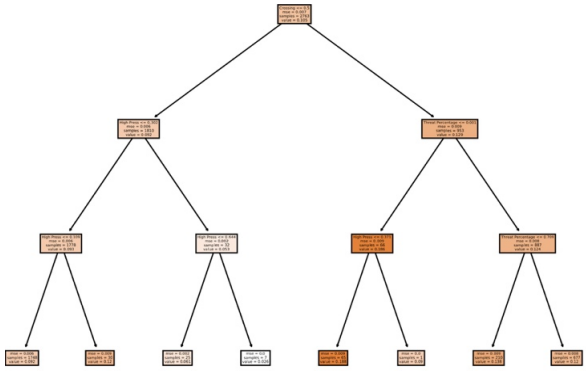

Figure 5 Regression Tree of Crossing, Threat Percentage and High Press - Depth 3.

We can observe that the fifth leaf node from the left contains the largest value of Expected Goals as it has the darkest shade. Further inspection reveals that the leaf node contains 65 samples of shots with 0.188 Expected Goals. By tracing a path from the root node to the leaf node, we can determine a set of constraints which lead to this outcome:

1. Crossing > 0.5, and
2. Threat Percentage <= 0.001, and
3. High Press Percentage <= 0.371

By applying these constraints on our data set, we can get more information about the teams that used this combination of possession styles the most:

1. Lion City Sailors: 18 instances
2. Geylang International: 11 instances
3. Tampines Rovers: 9 instances

This gives us information about the strengths of each individual team and the styles of play they can use effectively. Other notable sets of constraints derived from the regression trees are:

1. Crossing > 0.5 and Threat Percentage <= 0.001 and Fast tempo <= 0.75: Resulting in 63 samples of shots with 0.188 Expected Goals
    a. Lion City Sailors – 18 instances
    b. Geylang International – 11 instances
    c. Tampines Rovers – 9 instances
2. Crossing > 0.5 and Threat Percentage <= 0.001 and Direct Play Percentage <= 0.812: Resulting in 55 samples of shots with 0.193 Expected Goals
    a. Lion City Sailors – 16 instances
    b. Geylang International – 10 instances
    c. Tampines Rovers – 7 instances
3. Crossing > 0.5 and Threat Percentage <= 0.001 and Maintenance Percentage > 0.588: Resulting in 22 samples of shots with 0.216 Expected Goals
    a. Lion City Sailors – 12 instances
    b. Tampines Rovers – 7 instances
    c. Geylang International – 6 instances
4. Crossing > 0.5 and Buildup Percentage <= 0.181 and Threat Percentage <= 0.058: Resulting in 69 samples of shots with 0.191 Expected Goals
    a. Lion City Sailors – 17 instances
    b. Tampines Rovers – 11 instances
    c. Geylang International – 10 instances
5. Crossing > 0.5 and Fast Tempo <= 0.625 and Counterattack Percentage > 0.214: Resulting in 47 samples of shots with 0.16 Expected Goals
    a. Lion City Sailors – 8 instances
    b. Balestier Khalsa – 7 instances
    c. Hougang United – 6 instances
6. Crossing > 0.5 and Buildup Percentage <= 0.181 and Fast Tempo > 0.125: Resulting in 193 samples of shots with 0.163 Expected Goals
    a. Tampines Rovers – 35 instances
    b. Albirex Niigata – 27 instances
    c. Lion City Sailors – 26 instances

## 4 CONCLUSION

The findings from the computed regression trees provide an insight into the styles of play that are effective in the Singapore Premier League. For

example, shots from crosses have a higher Expected Goals value compared to other shots from open play.

Teams can use this data to quantify their performance on the pitch and objectively compare themselves against their peers. Using data to find the strengths of opposition teams is also an effective means of scouting that provides actionable insights without the need to perform extensive analyses. For example, our findings show that Lion City Sailors can consistently create good chances from crosses. Therefore, teams who can prevent them from making crosses stand a better chance of winning.

Playing Style data and Expected Goals values can play an essential role in data-driven recruitment, allowing teams to maximize their returns on investment. Teams can use Expected Goals to find undervalued players in the domestic market. They can also identify overperforming players and avoid recruiting them before they regress to normal performance levels.

The consistent collection and application of event data for the Singapore Premier League can ensure that these models are iteratively refined, and the accuracy of their findings improved over time.

## 5 ACKNOWLEDGEMENTS

I would like to acknowledge the funding support from Nanyang Technological University – URECA Undergraduate Research Programme for this research project.

Finally, I would like to give my most heartfelt graduate to Asst Prof Komar John from the School of Physical Education and Sports Science for giving me the opportunity to explore one of my deepest passions. His guidance and input throughout the project were invaluable.

## 6 REFERENCES

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Leuven, K. (2020). *How data availability affects the ability to learn good xG models*.

Mckinney, W. (2010). *Data Structures for Statistical Computing in Python*.

Ong, E. (2022). *Data Analytics in Sport: Using Data to Predict Football Performance - PESS21002 - YouTube*. https://www.youtube.com/watch?v=wlzIYW8KqLQ

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Rathke, A. (2017). An examination of expected goals and shot efficiency in soccer. *Journal of Human Sport and Exercise*, *12*(Proc2). https://doi.org/10.14198/JHSE.2017.12.PROC2.05

Reback, J., McKinney, W., jbrockmendel, Bossche, J. Van den, Augspurger, T., Cloud, P., Hawkins, S., gfyoung, Sinhrks, Roeschke, M., Klein, A., Petersen, T., Tratner, J., She, C., Ayd, W., Naveh, S., patrick, Garcia, M., Schendel, J., … h-vetinari. (2021). *pandas-dev/pandas: Pandas 1.2.4*. https://doi.org/10.5281/ZENODO.4681666

Rotella, J. (n.d.). *Probability, log-odds, and odds*.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python. *PROC. OF THE 9th PYTHON IN SCIENCE CONF*. http://statsmodels.sourceforge.net/

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/JOSS.03021