# Creating an Expected Goal Model for Singapore football: how much data is necessary to get the optimal model?

Siddhant Pathak
School of Computer Science and Engineering

Asst. Prof John Komar
Physical Education and Sports Sciences

***Abstract -*** This research paper elucidates the development and evaluation of a bespoke Expected Goal (xG) model catered to the nuances of Singaporean football. The core objective entails the construction of a robust and precise predictive model that quantifies the probability of goal attainment from diverse spatial positions on the football pitch. Recognizing the pivotal role of data quality and quantity in the training of xG models, this study endeavours to address the inherent limitations by harnessing an exhaustive dataset encompassing temporal and spatial particulars extracted from two seasons of English Premier League encounters. This dataset serves as a gold standard to facilitate the training and comparative analysis of multiple models. To account for heterogeneity in data availability, the training regimen is partitioned into five stratified levels that epitomize the extent of informational inputs afforded to the model. Variegated machine learning methodologies, including Logistic Regression, K-Nearest Neighbours, Linear and No-Support Vector Classifiers, Decision trees, and a bespoke Deep Neural Network, are judiciously leveraged to calibrate the probabilistic models. A judicious process of fine-tuning culminates in the proposal of an ensemble model, yielding an exemplary accuracy surpassing the 90% threshold on the test set. This seminal research not only propels the frontiers of xG modelling and machine learning in football but also confers pragmatic ramifications for performance analysis and decision-making within the sporting realm. By ascertaining precise xG prediction models tailored explicitly for Singaporean football and cognate domains marred by data paucity, this study imparts a cost-effective methodology to enrich performance analysis. Moreover, it engenders an enriched corpus of literature on machine learning in football, affording a fertile foundation to propel future scholarly inquiries within this domain.

**Keywords -** Expected Goal (xG), machine learning, neural networks

## 1 INTRODUCTION

Expected Goals (xG) has become a prevalent statistical tool in the realm of professional football, providing valuable insights into the performance of teams and individual players. Nevertheless, the accuracy and reliability of an xG model heavily rely on the availability of high-quality and sufficient data for training purposes. In this study, our objective is to develop an xG model tailored specifically for Singaporean football, with the aim of predicting the likelihood of a shot resulting in a goal or not. We approach this task as a classic probability prediction problem, where we seek to quantify the probability of goal attainment based on the characteristics of the shot. [1]

To achieve our goal, we will employ machine learning techniques that leverage historical data on shots and corresponding outcomes in Singaporean football matches. By analyzing a comprehensive dataset encompassing various spatial and temporal details extracted from multiple seasons of matches in the English Premier League, we aim to establish a robust and precise predictive model for xG estimation in the context of Singapore football. By utilizing this model, we can effectively quantify the probability of a shot leading to a goal from different positions on the football pitch, considering the intricacies and nuances of the Singaporean football landscape. [2]

Through this research, we aspire to contribute to the advancement of xG modeling and machine learning in the field of football. Our findings and the developed xG model have practical implications for performance analysis and decision-making within the realm of Singaporean football. By creating a precise and tailored xG prediction model, we offer a cost-effective methodology for enriching performance analysis and providing valuable insights into goal-scoring probabilities, thereby facilitating better strategic planning and decision-making processes for teams and players in Singapore football.

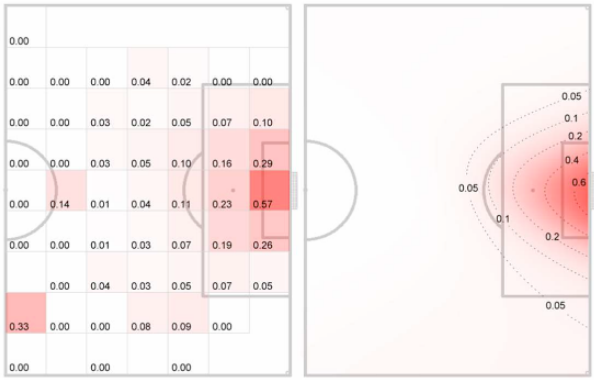


Figure 1: Heatmap of shot-conversion based on distance from goal [3]

As shown in the above shot conversion rates, it is very important to note that the probability tends to zero as we go farther from the field. However, this

statistical result does not consider other factors that have a physical and mental impact on the psyche of the player – namely, **time left, goals scored by both the teams, number of red cards/fouls, type of play (open play/set-piece)**. Regardless of the position, these dynamic natural features tend to influence the conversion rate as well.

# 2 METHODOLOGY

## 2.1 DATASET

The dataset consists of spatial and temporal information collected using sensors placed upon players based in the Premier League (previously known as the Barclays English Premier League). The process of data collection is beyond the scope of this research.

The dataset consists of 244 matches played over the span of two seasons (2015-16 and 2016-17). Each match has the same directory structure. For each match, there existed the following files:

1. Events CSV: This consists of all the major and minor events that occurred during the match.
2. Player Trajectory CSV: For each player that played in the game, including those who were substituted as well, this consists of spatial and temporal data about the player, at regular intervals of 10 seconds.

Given below is a summarized data schema (only the relevant columns are mentioned here) explaining the Events CSV introduced above.

Table 1 Data Schema for Events CSV

| Column Name | Data Type | Remarks |
|---|---|---|
| IdPeriod | int64 | Categorical variable denoting which half of the game (first/second) |
| MatchDate | datetime | Date on which the game was played |
| MatchTime | datetime | Time at which the game commenced |
| Time | float64 | Time (relative to the game start) during the game |
| Match | string | Title of the match |
| Player1name | string | Name of the player involved (similarly, Player2name) |
| Player1team | string | Team of the said player similarly, Player2team) |
| EventName | string | Categorical variable indicating the kind of event |
| LocationX | int64 | Relative horizontal (X) coordinate of the said player at the time of the event |
| LocationY | int64 | Relative vertical (Y) coordinate of the said player at the time of the event |
| PhaseType | string | Categorical variable indicating the type of play (Set Piece/Open-Play) right before the event occurred |
| ScoreHomeTeam | int64 | Number of goals scored by team 1, i.e., home team. (similarly, ScoreAwayTeam) |
| RedCardsHomeTeam | int64 | Number of red cards incurred by team 1, i.e., home team. (similarly, RedCardsAwayTeam) |
| AttackingDirection | int64 | Categorical variable denoting which side of the field team 1 has to score (left/right) |

For this research, the football field was assumed to be a two-dimensional Cartesian Plane as shown in Figure 2:

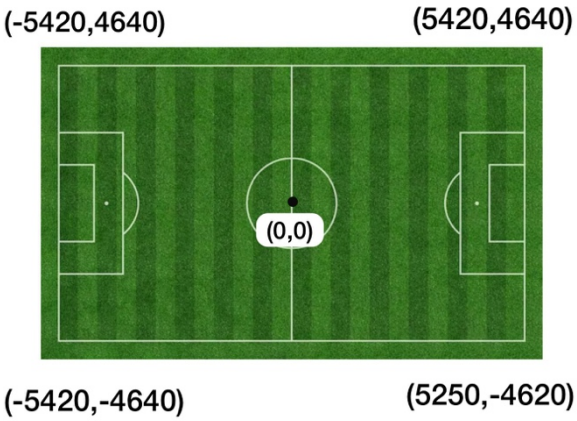


Figure 2: Reference coordinates

## 2.2 ANALYSIS

For the scope of this research and due to lack of computational resources, the second type of files (i.e., player trajectory CSVs) are ignored during the analysis segment.

We propose a two-step pipeline to break down important tasks such as preprocessing, model creation and fine tuning.

1. Process Pipeline: for loading, cleaning, and preparing data into test and train datasets.
2. Model Creation Pipeline: for creating, building, and evaluating ML/DL models and run inferences, visualize their results and store for future reference. Fine-tuning has been performed as well to find the optimal system architecture.

### 2.2.1 PROCESS PIPELINE

The aim of this pipeline is to clean the dataset and prepare it to be used for our next step, Model Creation Pipeline. The aim is to keep and use minimum yet important features to predict probability of goal scoring. After performing extensive exploratory data analysis and domain expertise, Table 2 showcases the features to be used with remarks about the reasoning behind it.

Table 2 Features to be used

| Feature Name | Remarks |
|---|---|
| TimeLeft | Categorical variable denoting which half of the game (first/second) |
| ScoreHomeTeam, ScoreAwayTeam | Date on which the game was played |
| HomeTeam | Time at which the game commenced |
| PhasePlay | Time (relative to the game start) during the game |
| LocationX, LocationY | Title of the match |
| Player1name | Name of the player involved (similarly, Player2name) |
| Player1team | Team of the said player similarly, Player2team) |
| EventName | Categorical variable indicating the kind of event |
| LocationX | Relative horizontal (X) coordinate of the said player at the time of the event |

The preparation of a final dataset by aggregating the necessary events from all files from 244 matches is described below:

The steps shown in the Figure 3 are as follows:

1. Get the list of all the relevant files from all the sub-directories from the two seasons.
2. Create a raw dataset by aggregating and concatenating all the files into one. Save this as a local copy.
3. Encode the categorical variables (using One-Hot Encoding strategy) accordingly and save the changes as a local copy.
4. Split the dataset obtained dataset in above set to X (response) and Y (target).
5. Up-sample the dataset to account for the class imbalance between the ‘goal’ and ‘no-goal’ classes. The SVM-SMOTE algorithm is used here.
6. Save it as local copy.

#### 2.2.1.1 LEVELS OF INFORMATION

In order to address the variations in data availability and quality, this research paper has implemented a systematic approach by dividing the training process into five distinct levels. These levels serve as indicators of the amount of data made available to train the predictive model. By partitioning the training regimen in this manner, the researchers acknowledge that not all datasets possess the same level of informational inputs, which can significantly impact the accuracy and reliability of the models developed.

The purpose of this stratification is to ensure that the models are calibrated and evaluated under different degrees of data availability, thereby capturing the heterogeneity in the quality and quantity of information. This approach enables the researchers to assess the performance of the models across various scenarios and identify the optimal level of data required for accurate predictions. By systematically exploring different levels of information, the study aims to provide insights into the effects of data paucity and determine the minimum threshold of data needed to construct reliable expected goal (xG) models.

In essence, the researchers have recognized the importance of data availability in training accurate xG models and have taken a comprehensive approach by categorizing the training process into multiple levels. This not only allows for a more nuanced analysis of the models but also provides practical implications for performance analysis and decision-making in the field of sports. By understanding the impact of varying data availability on the models' performance, this research contributes to the development of cost-effective methodologies for enriching performance analysis, particularly in contexts where data scarcity is a challenge. Additionally, it adds to the existing body of literature on machine learning in football and paves the way for future scholarly investigations in this domain. Refer to Figure 5 to see detailed bifurcation of the different levels of information described in this sub-section.

### 2.2.2 MODEL CREATION PIPELINE

In the second step, referred to as the Model Creation Pipeline, various statistical and probabilistic models are constructed, built, and rigorously evaluated. We explore various methodologies, including Logistic Regression, K-Nearest Neighbours, Linear and No-Support Vector Classifiers, Decision trees, and a Deep Neural Network. Fine-tuning, an essential component of the pipeline, is conducted to optimize the models' performance by fine-tuning their hyperparameters and system architectures. The models are then subjected to comprehensive evaluation, employing accuracy metrics to assess their predictive capabilities accurately. Refer to Figure 4 for a clearer understanding.

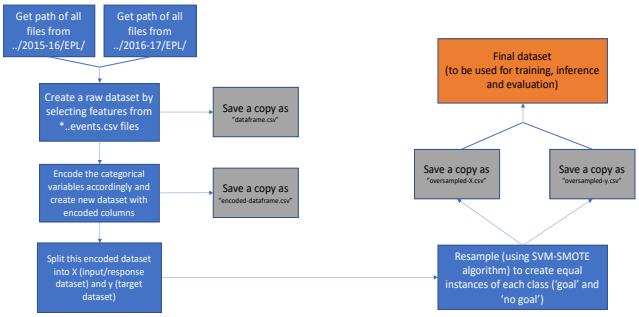


Figure 3: Process Pipeline

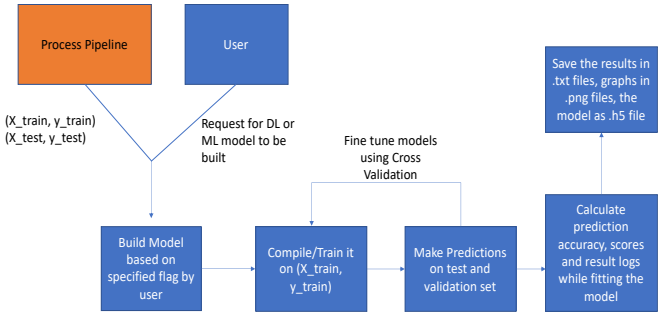


Figure 4: Model Creation Pipeline

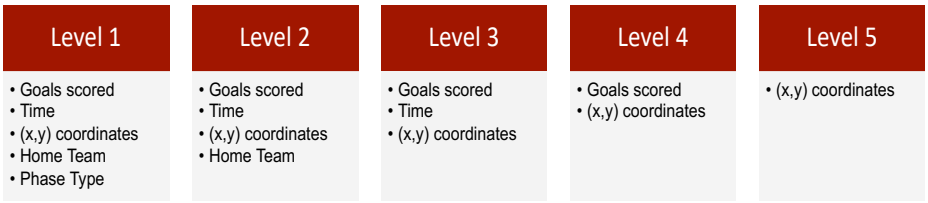


Figure 5: Levels of Information bifurcation

Additionally, the Model Creation Pipeline encompasses real-time inference generation using the trained models, allowing for timely predictions based on new inputs. To facilitate a comprehensive understanding of the models' performance, the researchers employ visualization techniques to present the results in a clear and interpretable manner. These visualizations aid in identifying the models' strengths and weaknesses, facilitating further analysis and decision-making. Furthermore, the results obtained from the pipeline are meticulously stored for future reference, ensuring reproducibility, and enabling comparative analyses over time.

We take two approaches to model creation pipeline:

1. Machine Learning: We use the in-built class of Logistic Regression under the Scikit-Learn package in Python 3.7. We use 70% of the data (randomly selected) to train the model and test it and compare the accuracy with the actual model on the remaining 30% data.
2. Deep Learning: We use TensorFlow 2 and its high-level Deep Learning API - Keras, which allows us to build, train and evaluate and execute all sorts of neural networks. We use 80% of the data (randomly selected) to train the model and test it and compare the accuracy with the actual model on the remaining 20% data.

### 2.2.2.1 MACHINE LEARNING APPROACH

We use some of the traditionally used machine learning algorithms:

1. Logistic Regression: A popular classification algorithm that models the relationship between the input features and the output using a logistic function, enabling the classification of data into distinct categories. Logistic Regression model computes a weighted sum of the input features (plus a bias term) and it outputs the logistic of this result. [5], [7]
2. Decision Tree: It constructs a hierarchical structure of if-else rules based on the input features, enabling effective classification and regression tasks by recursively partitioning the data into subsets.
3. K-Nearest Neighbors: It is a non-parametric algorithm utilized in this study to predict expected goals. It classifies data based on the majority vote of its k nearest neighbors in the feature space, particularly useful for spatial and pattern recognition tasks in Singaporean football. [6]
4. Support Vector Classifier: It constructs decision boundaries in the feature space to separate different classes, with the linear variant focusing on linear separability and the non-linear variant utilizing kernel functions to capture complex relationships in the data. [8]

### 2.2.2.2 DEEP LEARNING APPROACH

After performing up-sampling on the dataset, we obtain approximately 8,000 data points to train our models. This increase in data points through up-sampling provides us with a larger and more balanced dataset, enhancing the training process and potentially improving the predictive performance of our models.

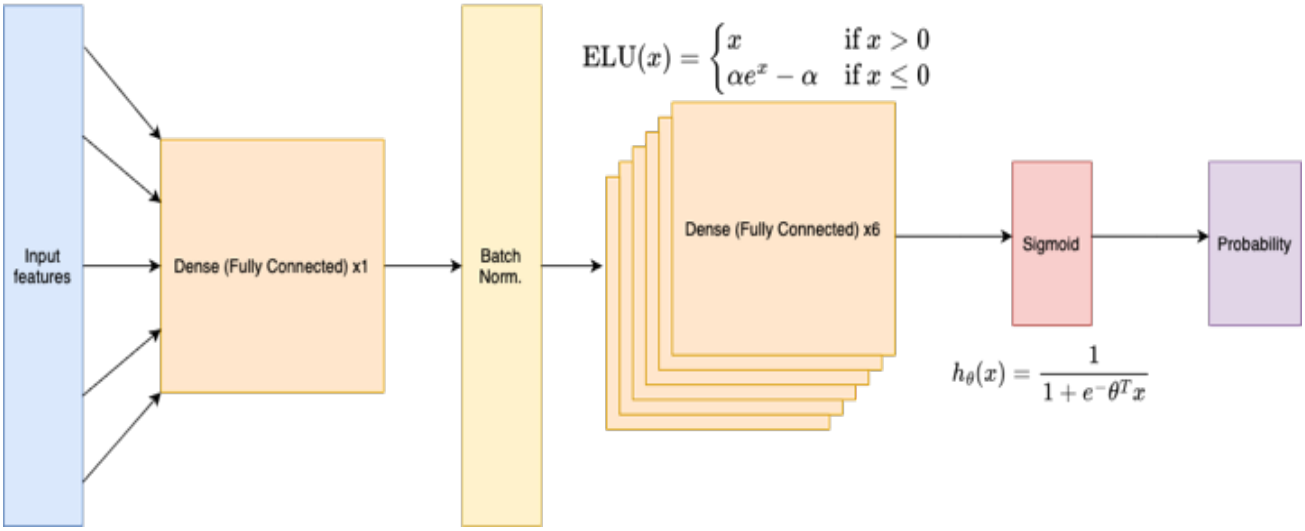


Figure 6: Proposed deep neural network architecture

Deep Learning is a powerful approach that we utilize to create our models. It offers us the freedom to customize various aspects, including mathematical techniques, network architectures, and other design choices. This flexibility allows us to tailor the models specifically to our task, leveraging the capabilities of Deep Learning algorithms to extract intricate patterns and representations from the data. By utilizing Deep Learning, we can overcome limitations and explore different approaches, enabling us to develop models that are highly suitable for our specific problem, potentially leading to more accurate and sophisticated predictions. [4], [9]

Our proposed architecture for the custom Deep Neural Network (Figure 6) is as follows:

1. It has 158 neurons for each layer. There is a batch normalization after the first hidden layer, followed by 6 neuron hidden layers. Finally, the output layer has one neuron containing the probability value. They all use the *Exponential Linear Unit* Activation function.
2. The hyperparameter se are determined via a 5-fold Cross Validation process, which ran simulations for hundreds of combinations and produced the one with highest accuracy.

We use the **StratifiedKFoldCrossValidation** class provided under the Scikit-Learn package in Python. To use our TensorFlow-based model, we wrapped our neural network with a Keras-ScikitLearn wrapper function.

## 3 RESULTS

The ROC (Receiver operating characteristic) curve is used to compare the performance of all the machine learning models. The ROC (Receiver Operating Characteristic) curve is a graphical representation that illustrates the performance of a binary classification model. It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The ROC curve provides a comprehensive evaluation of the model's trade-off between sensitivity and specificity, with a higher area under the curve indicating better overall performance. [15]

For easier understanding, here the results, performance evaluations, graphic visualizations shown are only for specific levels of information - Levels 1, 3 and 5. The trend, nature of the graphs is the similar and the inferences drawn from the same as well.

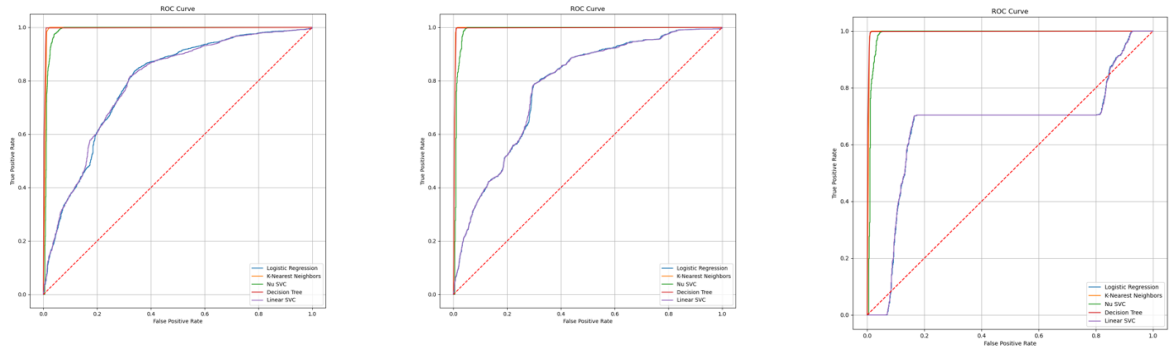


Figure 7: ROC Curve for the different ML models (L to R: Level of Information 1, 3 and 5)

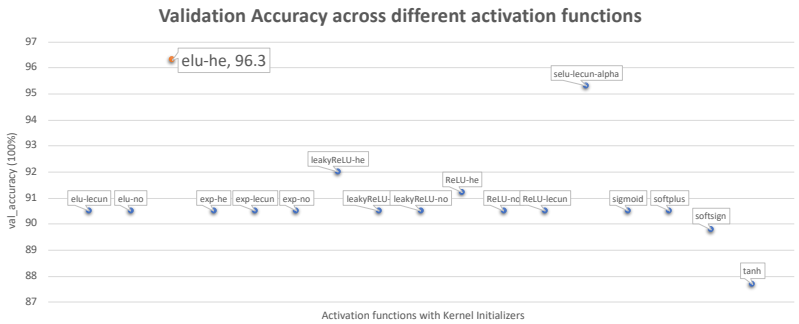


Figure 8: Validation accuracy - activation functions comparison plot

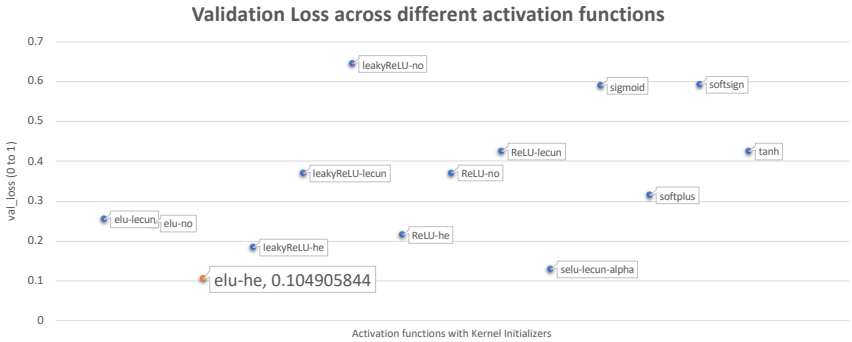


Figure 9: Validation accuracy - activation functions comparison plot

Figure 7 showcases a similar trend in the performance of various machine learning models. Logistic Regression, and Linear SVC (Support Vector Classifier), displayed similar performance, stagnating in the range of 0.68-0.74 accuracy (compared over 10 simulations). On the contrary, K-Nearest Neighbors, No-Support SVC and Decision Trees showcased brilliant accuracy, in the range of 0.90-0.99 (over 10 simulations). [10], [11], [13]

The hyper-parameters for the neural network were determined by a thorough examination of different settings and environments, incorporating theoretical knowledge with domain expertise. Given in Figures 8 and 9 are some comparison plots drawn for Validation Accuracy/Loss across different activation functions.

Table 3 Selected hyperparameters

| Configuration | Value |
|---|---|
| Activation Function | Exponential Linear Unit (ELU) |
| Batch Size | 580 |
| Kernel Initializer | He-Normal |
| Optimizer | Adadelta |
| Callbacks | EarlyStopping, ModelCheckpoint |

| Configuration | Value |
|---|---|
| Epoch | 150 |

After extensive deliberation, Table 3 summarizes the configuration used for training the Deep Neural Network architecture. Using these settings, the neural network was trained on a M2-based MacBook setup - using the Metal Plugin from TensorFlow (to compensate for the lack of CUDA and GPU-enhanced processing). The training results are shown in Figures 10 and 11. [12]

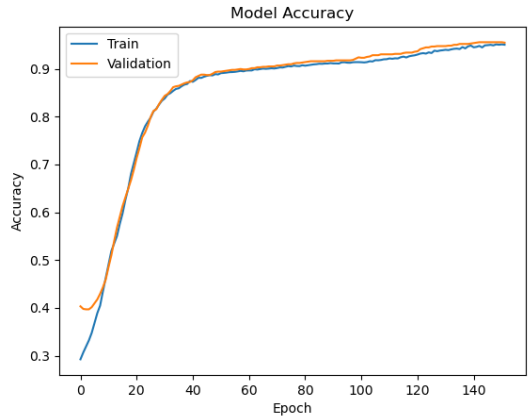


Figure 10: Model Accuracy Plot over training

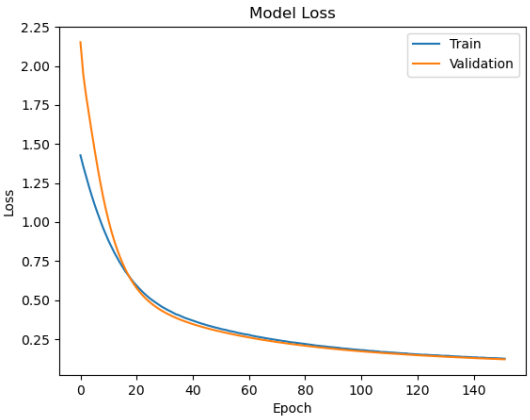


Figure 11: Model Loss Plot over training

Table 4 Evaluation Metrics for Level of Information (1)

| Model Name | Train Accuracy | Test Accuracy |
|---|---|---|
| Logistic Regression | 0.716 | 0.723 |
| **Decision Tree** | **0.995** | **0.992** |
| **K-Nearest Neighbors** | **0.990** | **0.985** |
| Linear SVC | 0.772 | 0.723 |
| **No-Support SVC** | **0.906** | **0.912** |
| **Deep Neural Network** | **0.961** | **0.975** |

Table 5 Evaluation Metrics for Level of Information (3)

| Model Name | Train Accuracy | Test Accuracy |
|---|---|---|
| Logistic Regression | 0.714 | 0.718 |
| **Decision Tree** | **0.997** | **0.994** |
| **K-Nearest Neighbors** | **0.991** | **0.986** |
| Linear SVC | 0.714 | 0.721 |
| No-Support SVC | 0.907 | 0.893 |
| **Deep Neural Network** | **0.945** | **0.947** |

Table 6 Evaluation Metrics for Level of Information (5)

| Model Name | Train Accuracy | Test Accuracy |
|---|---|---|
| Logistic Regression | 0.584 | 0.608 |
| **Decision Tree** | **0.995** | **0.992** |
| **K-Nearest Neighbors** | **0.995** | **0.993** |
| Linear SVC | 0.603 | 0.628 |
| No-Support SVC | 0.908 | 0.897 |
| **Deep Neural Network** | **0.988** | **0.992** |

Tables 4, 5 and 6 summarizes the results about the performance of all the different models discussed within this paper. All the models with more than 90% accuracy in both testing and training steps are highlighted in bold.

## 4 DISCUSSION

K-Nearest Neighbors, No-Support SVC, and Decision Trees outperform Logistic Regression and Linear SVC by capturing non-linear relationships, handling complex decision boundaries, and adapting to high-dimensional data. They excel at modeling intricate patterns and interactions, providing higher accuracy and flexibility. Logistic Regression and Linear SVC are better for linearly separable or low-dimensional data with simpler decision boundaries. [14]

Custom neural networks are often considered superior to k-nearest neighbors (KNN), decision trees, logistic regression, and linear support vector machines (SVM) due to their ability to learn complex non-linear relationships in data. Unlike these traditional models, custom neural networks can automatically extract intricate patterns and features from input data, allowing them to handle intricate tasks and achieve higher predictive accuracy in various domains. Additionally, neural networks have the advantage of adaptability and scalability, enabling them to handle large-scale datasets and accommodate different types of data efficiently.

ELU with He-normal initialization outperforms other variants in validation accuracy and loss. It addresses limitations of other activation functions, providing benefits of ReLU (no vanishing gradient) and Leaky ReLU (no dead neurons). ELU reduces bias shift, enhances deep neural network learning, and handles negative inputs effectively. It captures nuanced information, improving performance in machine learning tasks.

He-normal initialization performs better than LeCun initialization and no normalization in deep neural networks. It scales neuron weights considering the number of inputs, maintaining a stable gradient flow during training. It works well with activation functions like ReLU, addressing vanishing and exploding gradients. LeCun initialization assumes a fixed fan-in and may not be optimal for all activation functions. Absence of normalization leads to unstable weight updates and hinders learning, especially in deeper networks.

## 4.1 DATA SHIFT

Data shift refers to the change or shift that occurs in the characteristics and patterns of data over time. In the case of this research done on players' spatial and temporal data from 2014-2016, it means that the way players think and play the game has evolved since that period.

In recent years, several notable changes have occurred in the football landscape, contributing to shifts in player mentalities compared to the period of 2014-2016. One significant factor is the rapid advancement of sports science and technology, which has led to a greater emphasis on data analysis, player monitoring, and performance optimization. This scientific approach has influenced training methods, nutrition, and injury prevention, resulting in athletes who are fitter, faster, and more resilient.

Moreover, there has been a shift in coaching philosophies, with an increased focus on tactical flexibility, pressing, and positional play. Coaches now encourage players to be more versatile, adaptable, and intelligent in their decision-making on the field. Additionally, changes in the rules and regulations of the game have impacted playing styles, such as stricter enforcement of fouls, alterations in offside interpretations, and the introduction of video assistant referee (VAR) technology.

As a result, the deep learning model trained on data from 2014-2016 may not accurately reflect the current trends and behaviors of football players. To maintain the model's relevance and effectiveness, it would be beneficial to retrain it using more recent and representative data that captures the evolving nature of the game. By incorporating the latest player insights and strategies, the model can better assist coaches, analysts, and decision-makers in understanding and optimizing performance in today's dynamic football landscape.

## 4.2 ENSEMBLE LEARNING

Ensemble learning is a powerful technique in machine learning that combines the predictions of multiple individual models to obtain a more accurate and robust prediction. In a formal manner, ensemble learning involves constructing an ensemble by training a set of diverse base models on the same dataset or different subsets of the dataset. These base models can be of different types or trained using different algorithms. The predictions of these base models are then combined using various methods, such as majority voting, weighted averaging, or stacking, to generate the final ensemble prediction. The aim of ensemble learning is to leverage the diversity and complementary strengths of individual models to improve overall prediction performance and generalization capabilities.

After extensive research and investigation, and deriving insights from performances shown in Tables 4, 5 and 6, we propose a similar approach. We select the models with more than 90% accuracy on the test set, perform hyperparameter-tuning using Cross-Validation and then collate them together as shown in Figure 12.

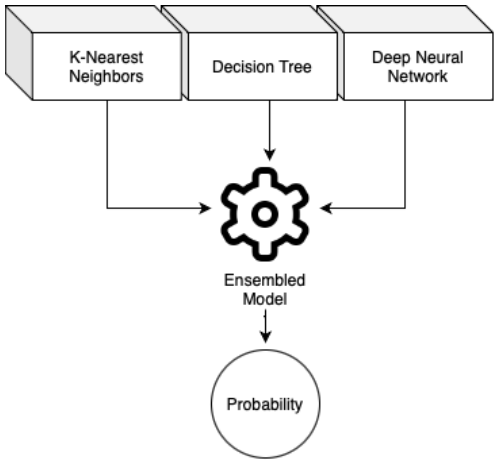


Figure 12: Proposed Ensemble Model

Ensemble learning is generally better than using a single model because it can reduce bias and variance, leading to improved prediction accuracy and robustness. By combining multiple models with different perspectives and approaches, ensemble learning harnesses the collective wisdom of diverse models, mitigating individual model weaknesses and enhancing overall performance.

Unfortunately, due to time constraints, the experimentation and implementation of this approach is out of the scope of this research.

## 5 CONCLUSION

In conclusion, this research paper presents a comprehensive exploration and evaluation of a bespoke Expected Goal (xG) model specifically designed for Singaporean football. The study addresses the challenges associated with data quality and quantity by utilizing an extensive dataset comprising spatial and temporal information from two seasons of English Premier League matches. The research employs a stratified approach, partitioning the training regimen into five levels that represent the varying degrees of data availability. Various machine learning techniques, including Logistic Regression, K-Nearest Neighbours, Linear and No-Support Vector Classifiers, Decision trees, and a custom Deep Neural Network, are employed to calibrate the probabilistic models. Through a meticulous process of fine-tuning, an ensemble model is proposed, which achieves outstanding accuracy exceeding the 90% threshold on the test set. This seminal research expands the frontiers of xG modelling and machine learning in football, offering practical implications for performance analysis and decision-making within the realm of sports. By providing precise xG prediction models tailored specifically to Singaporean football and similar domains characterized by limited data, this study presents a cost-effective methodology to enhance performance analysis. Furthermore, it contributes to a rich body of literature on machine learning in football, laying a solid foundation for future scholarly inquiries in this domain.

The applications of this research in the sports industry are extensive and impactful. The development of a bespoke xG model catering to Singaporean football opens up new avenues for performance analysis and decision-making in the sport. Coaches and analysts can leverage the predictive capabilities of this model to gain valuable insights into the probability of goal attainment from various spatial positions on the football pitch. This information can inform strategic decisions regarding player positioning, attacking patterns, and defensive strategies. Additionally, the research demonstrates the effectiveness of machine learning methodologies in handling data paucity, which is a common challenge in many sporting domains. The methodology proposed in this study can be adapted and applied to other sports with limited data availability, enabling performance analysis and decision-making in resource-constrained contexts. For example, similar predictive models could be developed for sports like cricket, where comprehensive data might be scarce for certain regions or lower-level leagues. Overall, this research has the potential to revolutionize performance analysis and decision-making in the sports industry, offering valuable insights and practical applications for coaches, analysts, and sports organizations.

For testing purposes, a locally-hosted Gradio-based simple website was developed with the front-end consisting of simple input fields/blanks and back-end consisting of the efficient model.

## ACKNOWLEDGMENT

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

## REFERENCES

[1] Rathke, Alex. (2017). *An examination of expected goals and shot efficiency in soccer. Journal of Human Sport and Exercise.* 12. 10.14198/jhse.2017.12.Proc2.05.

[2] Scarf, Phil. (2006). *Modelling the outcomes of association football matches*. 48th Annual Conference of the Operational Research Society 2006, OR48. 59-72.

[3] Ruiz, H., Lisboa, P.J., Neilson, P., & Gregson, W. (2015). Measuring scoring efficiency through goal expectancy estimation. The European Symposium on Artificial Neural Networks.

[4] Ericperko. (n.d.). Ericperko/ann_logreg: Artificial Neural Networks and logistic regression for EECS 440. GitHub. https://github.com/ericperko/ann_logreg

[5] A guide to logistic regression with TensorFlow 2.0. Built In. (n.d.). https://builtin.com/data-science/guide-logistic-regression-tensorflow-20

[6] Harrison, O. (2019, July 14). Machine learning basics with the K-nearest neighbors algorithm. Medium. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=KNN%20works%20by%20finding%20the,in%20the%20case%20of%20regression).

[7] Li, S. (2019, February 27). Building a logistic regression in Python, step by step. Medium. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

[8] Pledel, T. (2021, March 2). Using AI to classify a book. Medium. https://medium.com/mlearning-ai/using-ai-to-classify-a-book-c9dd21146759

[9] Pramoditha, R. (2022, May 19). Replicate a logistic regression model as an artificial neural network in Keras. Medium. https://towardsdatascience.com/replicate-a-logistic-regression-model-as-an-artificial-neural-network-in-keras-cd6f49cf4b2c

[10] Probability calculation using logistic regression. TIBCO Product Documentation. (n.d.). https://docs.tibco.com/pub/sfire-dsc/6.5.0/doc/html/TIB_sfire-dsc_user-guide/GUID-C4D05ED0-3392-4407-B62A-7D29B26DC566.html

[11] Real Python. (2023, June 26). Logistic regression in python. Real Python. https://realpython.com/logistic-regression-python/

[12] Rendyk. (2023, May 18). Tuning the hyperparameters and layers of neural network deep learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/

[13] S, P. (2023, April 27). Building an end-to-end logistic regression model. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/building-an-end-to-end-logistic-regression-model/

[14] Singh, S. (2023, April 28). Top 10 interview questions on evaluation metrics in machine learning. Medium. https://pub.towardsai.net/top-10-interview-questions-on-evaluation-metrics-in-machine-learning-407c547e7b46

[15] What is a ROC curve and its usage in performance modelling. What is a ROC Curve and its usage in Performance Modelling. (n.d.). https://www.tutorialspoint.com/what-is-a-roc-curve-and-its-usage-in-performance-modelling