Verifying Gambling Odds using Deep Learning and Machine Learning

Nikhil Vaidyanathan

Abstract

One of the applications of Machine Learning within sport is the ability to create accurate models that account for several factors that affect performance, utilizing readily available datasets ranging between different sporting leagues and seasons. Machine Learning expedites the analytical process, and allows for reliable and precise predictions. Deep Learning is used in a similar way, with applications ranging from injury recommendations to recruitment. In football, many factors are at play during the ninety-minute game. This paper intends to answer the question of whether a Cross-Entropy Cost Function integrated with an adjusted multivariable, 3-D logistic regression can be used to predict and model the probability of a player taking a given volume of shots on target or scoring a given number of goals in the 2021/2022 English Premier League (EPL) season. The models created will be used to verify if bookmakers such as SkyBet are employing enhanced odds which have a large disparity between marketed and actual probabilities. This paper proposes an unsupervised approach created from a modified dataset including outliers over the course of the whole EPL season.

1. Introduction

Artificial neural networks (ANN) are typically used in classification and regression of a particular dataset and "are computing systems inspired by the biological neural networks that constitute animal brains" (Yang, 2014). ANN are used conventionally in a large majority of sectors such as banking and healthcare. Neural networks are utilized in applications that require the use of the machine learning technique's random function, allowing for optimization through running past data millions of times.

Three dimensional graphs in machine learning are predominantly used to explore and represent a relationship between multiple independent variables. The use of machine learning techniques are additionally able to classify and visualize patterns through the application of mathematical formulas, allowing for the creation of three-dimensional graphs with its trends being elucidated. Machine learning enables speedy processing of data and provides accurate predictions and regression. The paired use of ANN and multivariable 3-D graphs allows for ease of visualization and greater accuracy.

The legal sports gambling industry is currently estimated to be valued at $140 billion, with a large majority of this share coming from the world's leading sport, Associate Football or Soccer. One of the largest players in the market, SkyBet was valued and bought by The Stars Group for £3.4 billion ($4.8 billion) in 2018. SkyBet's marketing strategy for football bets involves a high usage of "enhanced odds", wherein a price boost is given to select odds. The price boost gives increased returns to the gambler who bets on the odd. Most enhanced odds are shots on target bets, which are the prediction of the volume of shots a player may have on goal. The price boosts on Enhanced Odds are generally enticing to a gambler. These enhanced odds are promoted all over SkyBet's social media and may additionally be used as a "Welcome Bonus" for signing up or depositing money in SkyBet.

Using a model that integrates machine learning and deep learning, and by examining evidence from thousands of English Premier League (EPL) shot and goal data, we can create models that predict the percentage chance that a player scores a certain volume of goals or shots and compare the model's predictions to SkyBet's odds to prove whether SkyBet uses enhanced odds for events with a significantly lower probability of the event occurring.

2. Rationale and Hypothesis

In the United Kingdom (UK), gambling is heavily advertised, and one may conclude that gambling advertisement traps customers in a never-ending cycle of addiction. The rate of gambling is extremely prevalent in the most impoverished parts of the UK; places in which losing a great amount of money results in the loss of livelihood. A Gambling Commission survey found that "81% of respondents from Great Britain reported seeing traditional gambling advertisements, 78% reported seeing sponsorships, and 68% reported seeing online advertising" (Killick & Griffiths, 2022). Additional studies have found that "new customer enhanced odds were often promoted" (Killick & Griffiths, 2022) with a small sample size determining enhanced odds to be a large majority of betting corporation's promotions. Newall et al. (2019) reported that "gambling companies used tactics to make the bets appear more 'urgent' than necessary."

Gambling odds work by rewarding users proportionately more for lower probability events. Therefore, enhanced odds essentially boost the payout a gambler will receive as if the probability of the event was lower. Most of these enhanced odds being shots on target odds can easily trick a gambler as the distinction between shots on target and general shots may be easily overlooked. For example, Mohammed Salah of Liverpool FC completes over 4 shots every ninety minutes, however, when the distinction of shots on target is made, Salah completes 1.6 shots on target every ninety minutes. Therefore, SkyBet's May 7th, 2/1(33%) odd of Salah, Harry Kane, and Heung-Min Son to have one or more shots on target each may seem like a certainty in the eyes of a gambler, after watching all three players shooting over one time in most games. However, with further inspection, due to the nature of the game being that of a big game, as well as all three players

having important games without registering a shot on target, it is unsurprising that this event did not occur, and the probability of this event occurring would likely be below the 33% marketed.

This study aims to build upon this example to see if there is a consistent pattern of betting companies like SkyBet overstating odds in their promotional material. From a preliminary scan of the betting data collected, alongside the extremely manipulative tactics used, this study hypothesizes that bookies such as SkyBet are creating enhanced odds that are significantly higher than the probability of the said event occurring.

3. Machine Learning/Deep Learning Information

3.1 Definitions

- **Loss –** Loss refers to the disparity between the machine learning model's prediction output in comparison to the input values. The lower the loss, or **cost,** the lower the deviation between predicted values and actual values, which in most cases signals model accuracy. In the training of a machine learning model, empirical risk minimization occurs where there is an element of curve fitting used which ensures the model optimizes or minimizes loss.
- **Neuron –** "in deep learning models are nodes through which data and computations flow. (McCullum, 2020)". Neurons in a deep learning model, similar to a biological context have synapses with **weights**, which affect the overall output of the model as neurons with greater weights determine the neuron's overall importance to the output variable. Neurons in an ANN perform calculations in addition to sending output signals to other Neurons through synapses.

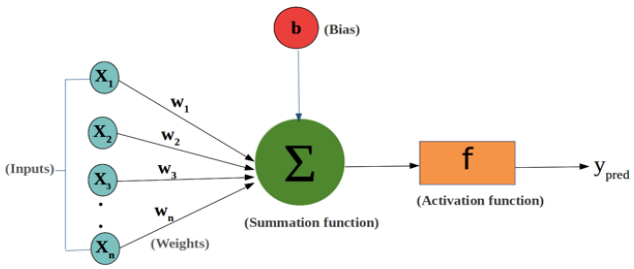

Figure 1: Visual Representation of the Skeleton of an Artificial Neuron (Ganesh, 2020).

- **Epoch –** An epoch is an arbitrary measure of time, signaling that an "entire dataset has been passed forward and backward through the neural network only once." (Sharma, 2017). A great number of epochs enable the deep learning/machine learning model to "learn" by running through the data several times. Through an iterative process, weights in the neural network are changed to fit the dataset with a curve after each epoch ran. After a certain number of epochs, the aim of a model is to converge output values, thus representing a curve fit.
- **Bias –** In a Cross Entropy Cost Function or other models that utilize Artificial Neural networks, Bias is used as another parameter in addition to **weight** to adjust the dichotomous outcome variable. Bias allows a machine learning model to come closest to the optimal curve fitting, by increasing or decreasing the value of the activation function at a given independent variable value. Bias in Machine Learning models is equivalent to adding an intercept to a basic linear graph, where neuron weights are similar to the slope of the function.
- **Deviation –** In this paper, "deviation" is defined as the absolute value of the percentage difference between two probability outputs with the same independent variable inputs. Deviation is additionally used to compare weight and cost values depending on the

specific model training method results. Deviation is calculated by the following function, where “u” represents a value:

$$D = \left| \frac{(u_0 - u_1)}{u_0} \times 100 \right|$$

3.2 Machine Learning Techniques Used

Binary Cross Entropy Cost Function is an objective loss function concerning the classification of binary data. The function produces an average cost at the i-th scalar value of a model, in comparison to an actual value from the input matrix. The deep learning method is also referred to as a regularization method, which are techniques that make slight modifications in a learning algorithm so that neither overfitting nor underfitting of a prediction curve occurs. In a binary cross entropy cost function, the rate at which the neuron changes its weight and bias is "determined by the partial derivatives of the cost function with respect to weight and bias, $\partial J/\partial w$ and $\partial J/\partial b$" (Nielson, 2020). The function is typically used in problems that require optimization or require a minimization of losses.

A binary cross entropy cost function, run over a given number of epochs, can allow the user to create a visual representation of costs as the weights of a multivariable problem are changed. The 3-D graph generated is convex in shape, and as each epoch passes, the cost at a given weight converges towards a minima. The model reduces the loss and improves the output probability until global minima is reached. This can be mathematically represented by the partial derivations with respect to both bias and weights.

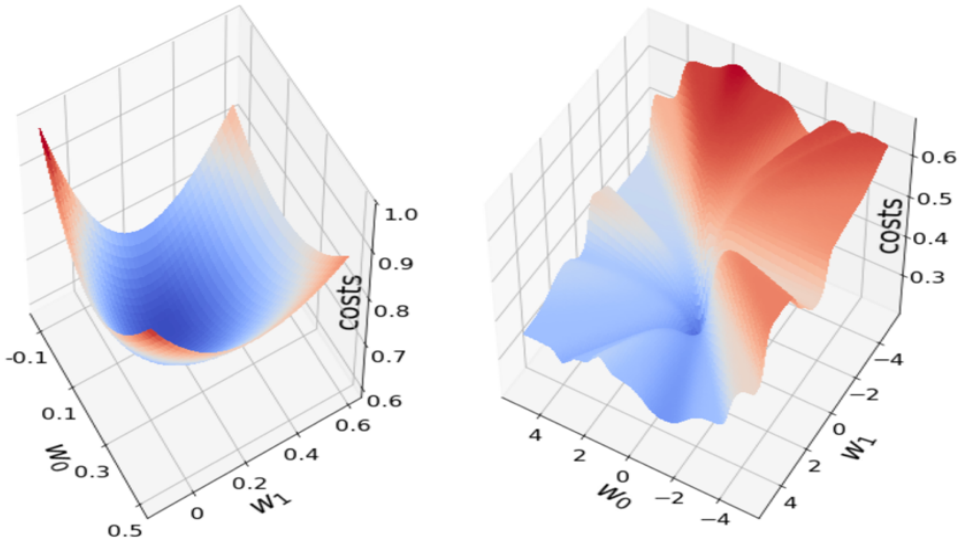

Figure 2: A Visualization of Costs in a **Cross Entropy Cost Function** vs a **Mean Squared Error (MSE) Function** (Roeschl, 2020)

Other cost functions such as **Mean Square Error (MSE)** are used in similar contexts. The MSE cost function is calculated by the square of the difference between observed and predicted values, and is a more general estimate of cost/loss in comparison to a cross entropy cost function. MSE functions involve multiple minima for the cost function, resulting in a lack of convergence in minima, as seen in Figure 2. The multiple local minima result in a far more difficult choice for choosing weight values. Additionally, the global minima of an MSE function lacks mathematical proof that said weight values are the global minima, thus finding the said global minima requires qualitative observance. Due to this characteristic lack of minima convergence, this study chooses to use a Cross-Entropy Cost Function.

Multivariable logistic regression models the logarithmic relationship between multiple independent variables and a binary dependent variable, using a sigmoid function. The benefit of using multivariable logistic regression is that it also provides a percentage probability of an event

occurring at given input parameters. The multivariable logistic regression algorithm classifies a dataset, placing data in between a set of two discrete classes. In this study the two classes used are binary, where "1" refers to the event occurring and "0" referring to an event not occurring. The Multivariable logistic regression creates a 3-D graph depicting the exponential relationship between the various independent variables and the dependent, predictor variable.

Logistic regression produces a sigmoid or logit function, which represents a probability between 0 and 1 inclusive, as represented in the function's range in Figure 3. The function incorporates all input values, and in an integrated cross-entropy cost function, it results in an optimal curve fit. In the model, outlier values are pushed towards the "1"s or the "0"s in y-axis.

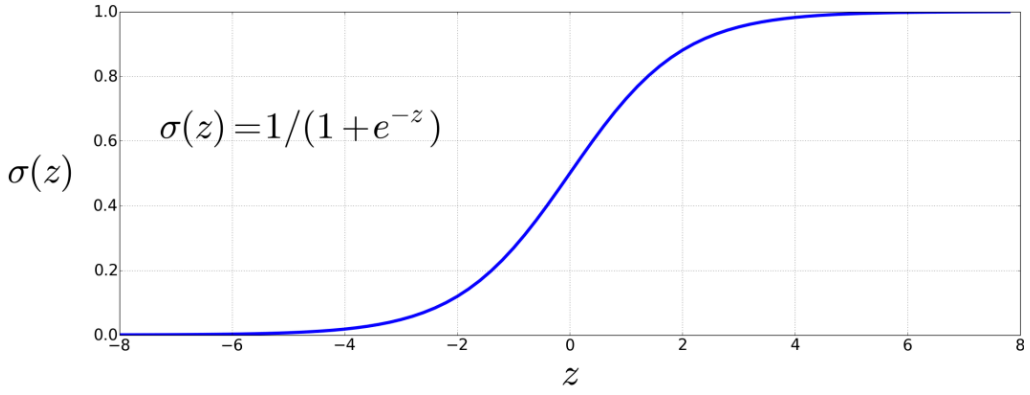

Figure 3: A sigmoid function with the independent variable z with a simple logistic regression equation (Jurafsky & Martin, 2021)

Logistic regression paired with a cross entropy cost function results in a greater minimization of cost/loss global minima. The sigmoid functions produced by multivariable logistic regression acts as an activation function for the deep learning/machine learning integrated model. For this data, logistic regression is used, since probability of an increasing amount of shots on target or goals decreases exponentially.

In this study, the logit function produced is represented in code by scipy's expit module, specifically p = expit(X @ w.T + b) within the code; The function contains bias(b), weights(w) and the input matrix of independent variables(X). The logit or sigmoid function involves a dot product which depicts the sum of the products of each vector value in the linear algebra aspect of this model.

3.3 Binary Cross Entropy Cost Function Partial Derivations:

To find the partial derivatives of costs – represented by the variable J – with respect to a specific weight (w_j), several formulas must be input to the cost function, where the chain rule can be utilized to find the final derivative.

This complete partial derivation method is cross checked through Wolfram Alpha, which produces the same results.

$$J = -\frac{1}{N}\sum_{i=1}^{N} y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)$$

The original function can be determined here with: N representing the amount of samples, the outcome variable is represented with y_i and the predicted value is represented by $p_i \in [0, 1]$. p_i is the sigmoid function derived from a standard logistic regression, with the formula shown below that contains the modified exponent z_i. b represents a scalar bias value, w represents a weight vector in the form $w = [w_0\ w_1]$. X represents the input matrix and is used for iterative purposes for each epoch to change the weight vectors to ensure optimal curve fitting.

$$p_i = \frac{1}{1 + e^{-z_i}} \qquad z_i = X_i w^T + b \qquad X = \begin{bmatrix} (x_0)_0 & (x_1)_0 \\ (x_0)_1 & (x_1)_1 \\ \cdots & \cdots \\ (x_0)_N & (x_1)_N \end{bmatrix}$$

The derivation of a sigmoid function can be represented as $\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x))$ and the derivative of the natural logarithm is x^{-1} therefore the derivation of cost with respect to a specific weight can be shown by:

$$\frac{\partial J}{\partial w_j} = -\frac{1}{N} \sum_{i=1}^{N} \frac{\partial J}{\partial p_i} \frac{\partial p_i}{\partial z_i} \frac{\partial z_i}{\partial w_j}$$

$$= -\frac{1}{N} \sum_{i=1}^{N} [\frac{y_i}{p_i} + \frac{1 - y_i}{1 - p_i}(-1)] \left[p_i(1 - p_i)\right] x_j =$$

$$= -\frac{1}{N} \sum_{i=1}^{N} [y_i(1 - p_i) - (1 - y_i)p_i] \, x_j =$$

$$= -\frac{1}{N} \sum_{i=1}^{N} (y_i - p_i) \, x_j =$$

$$= \frac{1}{N} \sum_{i=1}^{N} (p_i - y_i) \, x_j$$

The partial derivative of the cost function with respect to bias can be represented by:

$$\frac{\partial J}{\partial b} = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)$$

since the partial derivative of the z_i with respect to bias is 1.

The rationale behind choosing a multivariable logistic regression is based off the logic that the probability of said event occurring decreases as the volume of shots on target or goals increases. Additionally, multivariable logistic regression is implemented since players are exponentially less likely to have a high volume of attacking actions (shots on target or goals) against teams who are significantly better than their team on paper. Multivariable logistic regression fundamentally depicts both these observations, hence why it is chosen to prove whether or not gambling corporations are inflating odds for shots on target and goal bets.

3.4 - Data Sets

The study creates a dataset using player shots on target game-by-game data throughout the 2021-2022 EPL Season. The data is gathered from Football Reference(fbref) Premier League individual player stats. Fbref is sourced from Statsbomb and Opta. Since SkyBet uses Opta data as evidenced by some of their enhanced odds advertisements, a source that contains Opta data is necessary.

Table 1: Premier League dataset statistics

Match Count	Player Count	Season
2887	96	2021/2022

To maintain a sufficient sample size, ninety-six players with ten or more shots on target are used for this study.

ClubELO.com provides the game-by-game Elo ratings of each of the twenty Premier League teams. It also includes the Elo changes throughout the 2021-2022 Premier League season, including home data and the percentage probability of a win, draw, or a loss.

SkyBet provides the enhanced odd data which specifies the date, match, player stat, predicted probability, and the price boost for enhanced odds. This was collected during the April to May 2022 period, near the end of the season, where the majority of prior shot/goal data had been previously established. SkyBet UK fractional odds are converted into probabilities using the formula below:

$$A/B = \frac{B}{(A+B)}$$

3.5 Ranking Methods

ELO – *Elo* is a rating system created by Arpad Elo for ranking of chess players. The ranking method quantifies a given player or team in relation to other players or teams. Elo is currently used in Association Football, American Football, Tennis, and a variety of other sports. The rating is based off pairwise comparisons, where the difference in relative Elo can be used to determine the percentage probability of a win, loss, or draw based on the disparity between two team's Elo rating. Since each team in the Premier League plays each other twice, pairwise comparisons are a valid system of ranking teams. Elo can be used to predict the probability of a given match outcome using the formula below:

$$E_a = \frac{1}{1 + 10^{(R_B - R_A)/400}}$$

The mutability of a given team's Elo rating is also represented by ClubELO's modified Elo ranking. ClubELO's weighted goal system ensures that Elo Points are exchanged proportionately

to the square root of the margin. The changes in Team Ratings with this factor accounted for are represented by the formula below:

$$\Delta Elo_{Margin} = \frac{20(Result - E_a)}{\Sigma(\sqrt{Margin} \times \frac{P(Margin)}{P(Win)+P(Loss)})} \times \sqrt{Margin}$$

Elo covers factors such as team form, represented by changes in Elo, player team quality, and opposition team quality each represented by respective Elo ratings. Elo ratings are used in the data due to number of factors it can cover.

AdjustedELO – AdjustedELO adapts the Elo method, creating a differential between a given player's team rating and his opposition team's rating in addition to a home constant if the said player is playing on home field. Similar studies such as Robberechts, et al. 2021 makes use of rating differential - the difference in Elo ratings between two teams – in the prediction of win percentage. This study adds on to the rating differential methods by accounting for the home field advantage through the AdjustedELO Method.

3.6 Other Factors Accounted For

a) Home Field Advantage

Phrases such as "fortresses" have been used in describing the environment in which a visiting team arrives at, especially in stadiums with large fanbase such as Liverpool F.C's Anfield. While determining the probabilistic model, certain factors must be accounted for, to reach an accurate probability. One of the factors which has been looked at largely by peer-reviewed, related work is the effect of a home field advantage. Home advantage mainly occurs due to "crowd involvement, travel fatigue [of the visiting team], familiarity of facilities, and referee bias that benefits the home

teams" (Price, et al., 2022). The same study concludes that "the home field advantage resides more heavily in offensive-based statistics" with home teams taking more shots from open play and "reaching the opponent's box more times". Since attacking data such as opponent half touches, successful key passes, shots from inside the box, shots from the "Danger Zone", and Expected Goals(xG) is proven to be affected by Home/Away games, the factor is accounted for in the AdjustedELO method used.

ClubELO has a system where a home field advantage is given a numerical Elo value, as represented by the formula:

$$HFA = \frac{3}{40} \sum_{i=1}^{N} \Delta Ei$$

This home field advantage has been calculated over the course of over half a century, applying this formula to 646,129 games. The 0.075 constant is a randomly selected constant which eventually allows for convergence between predicted and actual match outcomes. The home field advantage has been calculated to be between 42-44 points for teams in the Premier League by the end of the 21/22 season, when ELO differential is accounted for. 42 points generally comprises of relegation teams while 44 points generally comprises of high Elo teams such as Manchester City F.C. For the sake of simplicity, a constant of 43 points is used throughout the study. This constant is added to the ELO disparity between two teams in the AdjustedELO method adopted by this paper.

b) "Clutch Factor"

A football player who has shown prior empirical evidence of being able to change the outcome of a game is far more likely to take a higher volume of shots on target or goals against tougher opposition. The principle behind the clutch factor is that not all goals are created equal,

and a repeated amount of goals in important scenarios can indicate a high level of decisiveness. A given player's clutch factor will be quantified by adapting and implementing the Added Goal Value(AGVp90i) formula below, determining their clutch factor against Top 6 teams ranked by ELO.

$$\mathrm{AGVp90}_i = \frac{\sum_{k=1}^{K_i} 3 * \Delta P(\mathrm{win}|x_{t_k}) + \Delta P(\mathrm{tie}|x_{t_k})}{M_i} * 90,$$

The AGV formula has been used in similar football predictive studies such as Robberechts. 2021 in addition to National Hockey League (NHL) predictive models since its creation by Pettigrew. 2015.

c) Goalscoring Ability/Ability to take a greater amount of volume of shots

To account for players with different average shot on target and goal volume, an adjusted formula is proposed by this study. To illustrate the need for accounting for shot/goal volume we can take two players as examples: Player A – 0.3 goals p90 – and Player B – 0.45 goals p90 –. Assuming Ceteris Paribus, the likelihood of Player A scoring one goal is lower than the likelihood of the same event occurring with Player B. This study proposes to account for this factor by inputting adjusted shots on target and adjusted goal values which calculates the disparity between a given volume and the average per 90 volume. With this adjustment, one goal will be worth 0.7 Adjusted goals for Player A and 0.55 Adjusted goals for Player B. Zero goals will be worth -0.3 Adjusted goals for Player A and -0.45 Adjusted goals for Player B.

Making this adjustment allows for these two players to be treated as separate input values by the deep learning model and ensures that the two players will have different output values represented on one curve.

4. Related Work:

Artificial neural networks have been used in the prediction of NFL matches for as long as nearly half a century. Additional work has been done in EPL by using player related features and team related features. The deep learning model analyzing the neural networks was found to be the most accurate out of the models used by Rudrapal & Boro, 2020 with an accuracy of 73.57%. The research additionally contains an "approach mostly dependent on recent past matches" (Rudrapal & Boro, 2020) which applies to the work of this paper.

Other related works have utilized the expected goal(xG) models in prediction of goals in a given game. Due to the luck factor that exists in football, xG has been deliberately overlooked in this study. A critique that can be applied to this research paper's focus on expected stats is the uncertainty associated, for example, goals can be scored without xG values due to goalkeeper errors. xG does an extremely accurate job predicting goals based off positional data, however, it is truly impossible to predict the positions of shots prior to a game. Expected goal models vary depending on the source, the model used by Football Reference – cited from Statsbomb – , can have over a 10% difference in values compared to other sources such as Understat. For example, the disparity between Understat and Statsbomb's values are 6.44% for Liverpool's 21/22 EPL season, a margin of error too big to ignore.

5. Technical Approach:

5.1 Experimental Setup

The models are run on a 2018 MacBook Air with specs of: 1.6 GHz Dual-Core Intel i5 processor, Monterey 12.5 OS, and with 8 GB of memory. The code is run on Python 3.10.4 on

Jupyter Notebook. Packages included are: scipy for logit function creation; sklearn for logistic regression cross checks; NumPy for matrix and array multiplication; pandas for data frame conversions from the csv files derived from the excel datasets; celluloid for cost value and component parameter values storage; and matplotlib for 3-D graph creation where Axe3D, plt, cm, and gridspec are used.

5.2 Classification of Different Prediction Models

Five logistic regression graphs are created: The first exploring the probability of a certain amount of shots on target for the players who are amongst those with the highest clutch factor(≥ 0.5 AGVp90i) in addition to being high volume shooters – players in the top 20 for shots on target; The second graph exploring the probability of a certain amount of shots on target for the players who are amongst the high volume shooters but lack the clutch factor(>0.9 shots on target p90); The third graph exploring the probability of a certain amount of shots on target for the remaining, low-volume shooters in the dataset(<0.9 Shots on Target p90); The fourth graph exploring the probability of a certain amount of goals for high volume scorers (>0.5 Goals p90); And the fifth graph exploring the probability of a certain amount of goals for low volume scorers (<0.5 Goals p90). The dataset excludes any players with 0 goals, hence the total amount of players used for shots on target over the three datasets is greater than the total amount of players used for goals over the two datasets.

5.3 Programming Methodology

This study relies on prior empirical evidence to create probabilistic models. The probability of the said event occurring is a discrete, dichotomous variable(y), predicted by the implementation of both Deep Learning and Machine Learning methods. The model runs for 100,000 epochs at a

learning rate(α) of 0.025. An initial value of 250,000 epochs was set. However, due to the convergence of cost, bias, and weights at both values, 100,000 epochs are used for programming efficiency. With regards to curve fitting, overfitting may occur at higher values, and underfitting occurs at lower values; therefore, 100,000 epochs represent an optimal curve fitting. For each epoch run at a low learning rate(lr), the model is trained by storing parameter and cost values in neurons, changing these values based off the partial cost derivatives.

For initial values in the Cross Entropy Cost Function, this study uses the conventional starter weight values at 0 and 0. The code also involves the random selection of initial bias as 0.25. Theoretically, any bias value can be used as the values eventually converge before the 100,000th epoch. The calculated bias value over 100,000 epochs is used to create a fixed-intercept model, where weights and costs are recalculated only through the partial derivative of cost with respect to a specific weight. This fixed intercept model learns at a lower lr of $\alpha = 0.001$ over the course of 1,000,000 epochs for testing purposes.

All shots on target graphs comprise of eight scenarios for each player performance against a given team, with a player having a binary value of 1 for the actual volume of shots on target between 0 and 7 during a given game, and 0 for the other seven scenarios. All goal graphs comprise of six scenarios for each player performance against a given team, with a player having a binary value of 1 for the actual number of goals between 0 and 5, and 0 for the other five scenarios. All scenarios are then subject to the adjusted shots on target formula or adjusted goals which gives the x_0 data value. Having a scenario for every data point allows the program to concatenate low probability scenarios such as 7 shots on target or 5 goals in a game to a value close to 0.

Table 2: Size of each dataset, based off of scenarios previously described. *One Datapoint refers to an x_0 value, an x_1 value, and a y value.*

Dataset	Datapoints
1	2464
2	6568
3	17300
4	1908
5	15128

The code calculates y-values on a meshgrid with AdjustedELO as the x_1 axis, using the binary values of the different scenarios for cost reference. The 99,999th epoch is depicted for each graph, where cost has converged, enabling accurate visualizations. The accuracy of the model is tested by comparing the weights, bias, and cost to a fixed intercept model as well as sklearn's inbuilt logistic regression model.

5.4 Model Precision and Accuracy Cross-Checking with Other Logistic Regression Models

Table 3: Cross-Model testing for the each of the five models created with the fixed intercept model, *FI denotes the fixed intercept model*

Model	Weight 1	Weight 2	Cost	FI Weight 1	FI Weight 2	FI Cost
Model 1	-1.5915	0.0170	1.0473	-1.5214	0.0230	1.0278
Model 2	-1.8503	0.0168	0.9260	-1.8503	0.0168	0.9261
Model 3	-1.9596	0.0101	0.4537	-1.9595	0.0101	0.4537
Model 4	-4.4225	0.0117	0.5223	-4.4246	0.0115	0.5220
Model 5	-3.3851	0.0059	0.3102	-3.3851	0.0059	0.3102

Table 4: Cross-Model testing for the each of the five models created with the sklearn's inbuilt logistic regression model

Model	Weight 1	Weight 2	Cost	Sklearn Weight 1	Sklearn Weight 2	Sklearn Cost
Model 1	-1.5915	0.0170	1.0473	-1.5896	0.0143	1.0384
Model 2	-1.8503	0.0168	0.9260	-1.8512	0.0172	0.9250
Model 3	-1.9596	0.0101	0.4537	-1.9596	0.0101	0.4537
Model 4	-4.4225	0.0117	0.5223	-4.4231	0.0111	0.5221
Model 5	-3.3851	0.0059	0.3102	-3.3854	0.0059	0.3104

Table 5: Cost disparities between the deep learning model, the sklearn model and Fixed-Intercept model. *Scientific notation is used as percentage values decrease*

Model	FI Cost	Sklearn Cost Disparity
Model 1	1.86 %	0.85 %
Model 2	1.07e-2 %	1.08%
Model 3	4.34e-4 %	9.72e-3 %
Model 4	0.58 %	0.38 %
Model 5	6.34e-3 %	0.64 %

Cross testing was run with the aim of having similar, converging cost values over multiple testing methods. Cost values indicate the model's predictive accuracy, while the disparities between each cost value indicate the model's comparative precision. From the tables above, we can validate the original deep learning model's comparative precision due to a cost disparity of less than 2% for every individual test.

We can see that out of the five models, the most inaccurate and imprecise model was Model 1. This is not surprising due to the limited set of datapoints where only nine players being represented in the input dataset. Model 1 has the highest cost and variation between the cross tests. Models 2 and 4 have higher costs and greater cross-testing variation in comparison to Models 3 and 5, the two largest datasets. Overall, the models with larger datasets and lower ranged input domains (Models 3 and 5) have the best accuracy and precision results. A possible explanation is that curve fitting improves over far greater volumes of data. Cost reduces with lower domain range within datasets and Models 3 and 5 are classified as representing low volumes of shots on target

and goals. These two models have lower variety and range in comparison to models with a high-volume classification.

Due to the cross-tests generally having negligible disparities for weights and cost, no changes to the code are made to cause more accurate results. Cost values may seem high; The reason for the high cost value is explained by the fact that the many outliers over a season are accounted for. Additionally, Sklearn may have a marginally lower cost value, however, certain adjustments can be made to the existing model in order to increase accuracy in predicting shots on target and goal events.

5.5 Further Testing/Training

After validating the cost values of the models developed with an inbuilt logistic regression model, further testing is required in order to prove that the integrated deep learning and machine learning models created are fit for predictive capabilities. To further test the models developed, a test/train split of 80/20 is used for all five models/datasets.

Table 6: *Comparison of cost values for a test dataset and a complete dataset for each of the five models*

Model	Testing with 80% of Data	Testing with full Dataset
	Cost	
1	1.0624	1.0473
2	0.9586	0.9260
3	0.4695	0.4537
4	0.5715	0.5523
5	0.3246	0.3102

As seen in *Table 6* there is a slight increase in cost values when the test dataset implemented, however, the divergence between cost values is largely negligible, hence validating these models when a smaller dataset is utilized.

To further vindicate the decision of creating a binary cross-entropy cost function integrated with multivariable logistic regression, cross testing with another machine learning technique which involves classification and regression is necessary. This study chooses to compare the models created to a standard Support Vector Machine (SVM) model.

SVM – The Support Vector Machine Algorithm is a Machine Learning commonly used for regression and classification purposes.

A SVM algorithm works by creating a hyperplane – the best decision boundary that separates a space of a given dimension into classes, which can be used to predict new datapoints by placing said datapoint into a given class. Specifically, data used inputted for the test/train aspect of validating said model is delineated on the same n-dimensional space, with the location of each datapoint relative to the hyperplane being used to make predictions.

SVMs are used in this study as "SVMs also have the ability to project space through a non-linear function, lifting the data to a space with a higher dimension where a linear decision boundary does separate classes." (Stradling, 2016)

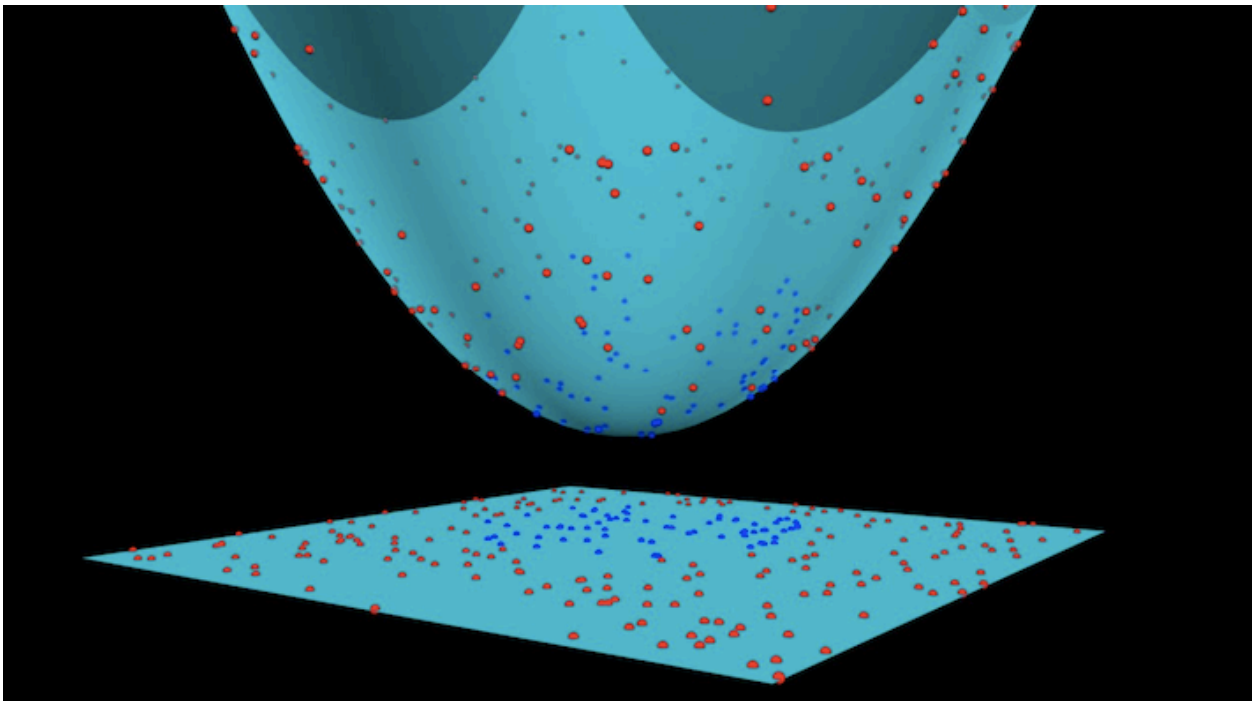

Figure 4: SVM with a Polynomial Kernel Visualization (Aharoni, 2007); *kernel refers to how the data is separated: kernel='linear' would indicate that the data is separated in a linear manner, and so forth*.

The SVM model and the binary cross entropy cost function integrated with fixed-intercept logistic regression will be compared on the basis of predictions made from the remaining 20% of data, as per the 80/20 test split. Predictions will be classified into four categories: True Positive, True Negative, False Positive, and False Negative outcomes. True Positives are positive outcomes predicted correctly, whereas False Positives are Type I errors, wherein positive outcomes are predicted negatively. True Negatives refer to negative outcomes predicted correctly, wherein False Negatives are classified as Type II errors, or negative outcomes predicted as positive outcomes. Recall can be defined as the ratio of true positives to positives identified, whereas precision is the ratio of true positives to the sum of false positives and true positives.

Table 7: Test/Train split results for the fixed intercept logistic regression model developed by this study

Dataset	Tested Datapoints	Fixed Intercept Logistic Regression Model						
		TP	TN	FP	FN	Precision	Recall	Accuracy
1	492	30	362	68	32	0.31	0.48	0.80
2	1313	104	959	190	60	0.35	0.63	0.81
3	3460	320	2516	511	113	0.39	0.74	0.82
4	381	58	269	48	6	0.55	0.91	0.86
5	3025	387	2344	186	117	0.68	0.77	0.90

Table 8: : Test/Train split results for a standard polynomial SVM algorithm

Dataset	Tested Datapoints	SVM						
		TP	TN	FP	FN	Precision	Recall	Accuracy
1	492	14	332	98	48	0.13	0.23	0.70
2	1313	75	921	228	89	0.25	0.46	0.76
3	3460	243	2317	710	190	0.25	0.56	0.74
4	381	18	254	63	46	0.22	0.28	0.71
5	3025	245	1817	704	259	0.26	0.49	0.68

As seen in Tables 7 and 8, the Fixed Intercept logistic regression model outperforms the SVM algorithm consistently in terms of prediction, as evidenced by higher precision, recall, and accuracy scores in all five datasets. This result validates the study's use of the multivariate logistic regression model over other standard models for the reasons detailed earlier in the paper.

However, a common trend that can be seen is the high incidence of low precision and low recall, especially in Datasets 1-3. The consistently low precision values indicate that there is a systematic issue with the models' capability in predicting negative outcomes.

Additionally, it can be noted that accuracy scores may be inflated in this testing as there is a 7:1 ratio of negatives to positives for Datasets 1-3 and a 5:1 ratio of negatives to positives for the remaining datasets. As there are many values close to 0 due to the high amount of negative outcomes, the amount of True Negatives predicted doesn't truly reflect the accuracy of the models

developed. To improve the model's capability to predict events accurately and increase precision/recall, two significant micro adjustments are made.

5.6 "Zero Values"

From the data, there is general trend where increasing shot or goal volume relative to a player's average shot or goal volume has an exponentially lower probability. A logistic regression model accounts for this general trend by enumerating the probability of a given player to have no shots or goals as an absolute certainty. However, this concatenation of the zero shots or goals probability is highly inaccurate, and therefore results in a high cost/loss value. Additionally, the sum of all probabilities exceeds 1, which is impossible. This phenomenon causes the model to predict 0 values as certainties in many cases, hence increasing the amount of False Negatives and resulting in low precision.

This study proposes to correct this inaccuracy by neglecting the logit function's "zero values" altogether, instead creating a probability distribution with the logit function's non-zero values. This would drastically decrease the incidence of False Negatives, thus solving the issue of low precision and increasing accuracy.

The creation of a distribution curve is done by implementing a Bernoulli's Distribution to the zero value and using every non-zero value that the logit function predicts. Using the probability of all non-zero values, the zero value is calculated so that the combined probability of every even is equal to 1. The zero value is calculated by the formula below:

$$P(0) = 1 - \sum_{i=1}^{\infty} P(i)$$

This change can be visualized with a simple logistic regression function for the probability of Mohammed Salah's shot on target volumes against Watford:

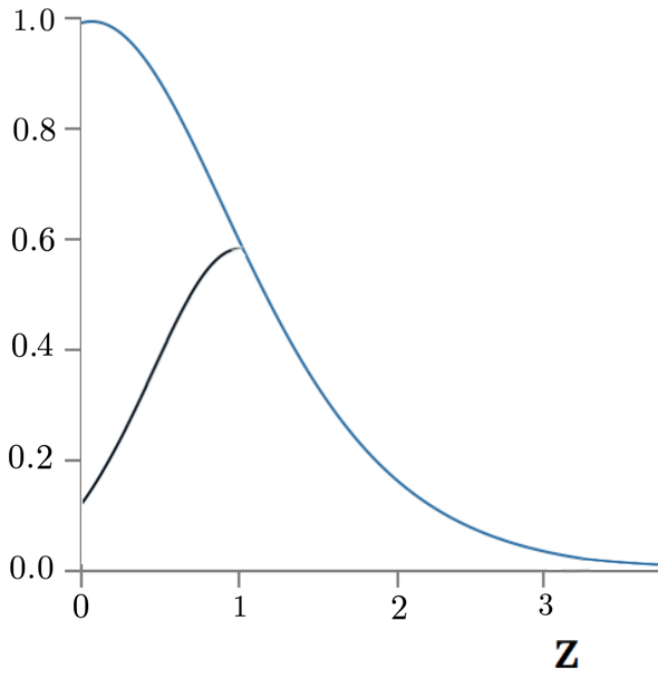

Figure 5: Modification to a Simple Logistic Regression Model with the implementation of an accurate "0 value". *Z represents Shots on Target, while the y-axis represents probability. The black line represents the modification, and the blue line represents the original logistic regression function.*

Making this change reduces the cost value of a given dataset by a significant amount, with Models 1, 2, and 4 having cost values that are concordant with the larger dataset models. The modification allows for an accurate "peak" – or the most probable outcome –, close to the player's average shot or goal volume.

5.7 Adjustment and Scaling of the Models

Another adjustment made to the logistic regression model is the scaling of the x_0 and x_1 values. Without the scaling, x_0 values may exceed a possible minimum, creating impossible probabilistic predictions. The logistic regression function responds to the input values by extending the maximum probability prediction value beyond the minimum adjusted "0 value", and in some cases it extends x_1 values beyond what is theoretically possible. This results in some models having

probabilities for a rating differential of 2000 Elo points in addition to -15 shots on target, which are impossible to occur. The overextending of the logistic regression model causes normal shot or goal probabilities to have heavily inflated probabilities at non scaled Elo values, resulting in the addition of normal ranged non-zero probabilities to be above the probability threshold of 1.

The x_0 and x_1 are divided by a factor "s" which is the ratio of the model's minimum to the possible minimum of the opposite independent variable.

Table 9: Scaling values for both x_0 and x_1 values for each model's graphical representation and predicted probabilities. *1 indicates that no scaling is required, because the model minimum and the theoretical minimum are equal.*

Model	ELO scaling	p90 Scaling
1	9.375	1
2	4.924	1
3	3.889	1
4	2.667	4
5	2.500	1.5

With the implementation of these scaled values, an accurate Bernoulli's distribution curve can be created. The sum of all non-zero values under a given scaled model will always be under 1, allowing for a "0 value" to be added. The calculated "0 values" also make qualitative sense after reviewing an overall dataset. The scaled models do not predict 0.99 for any non-zero individual event, which is accurate to the unpredictable nature of Associate Football.

The trend of lower scaling values for Model's 3-5 is in line with the fact that they are far more accurate than the first two models, hence proving the necessity for this micro adjustment.

Using the proper scaled values would increase the amount of accurate predictions in general, improving the models' scores for precision, recall, and accuracy.

6. Final Model Results

6.1 Graphical Representations of Probabilities

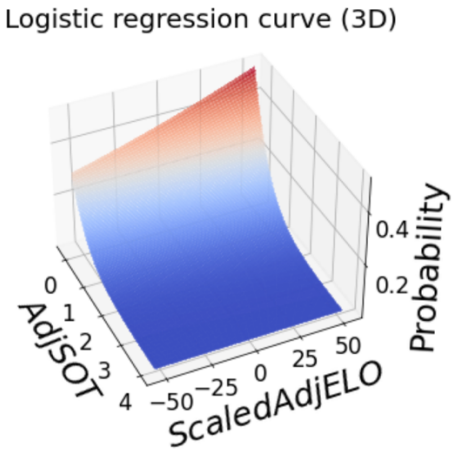

Figure 6: *Scaled Model 1*

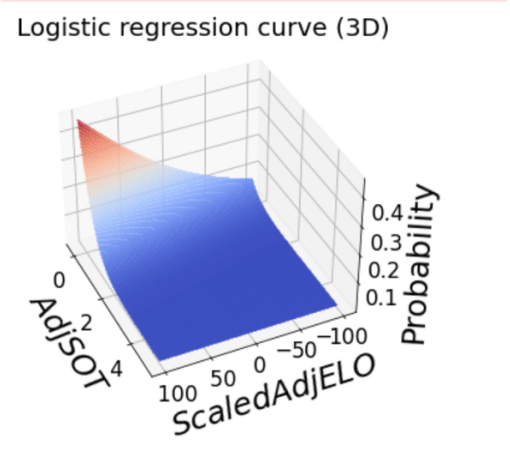

Figure 7: *Scaled Model 2*

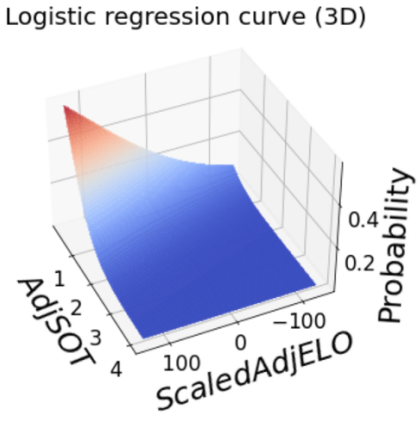

Figure 8: *Scaled Model 3*

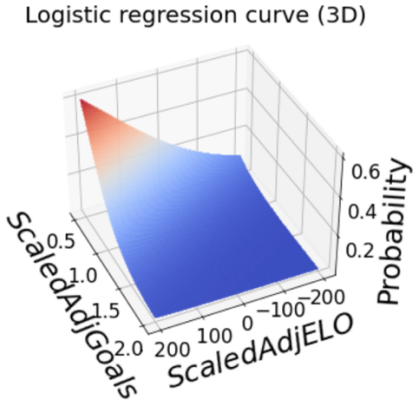

Figure 9: *Scaled Model 4*

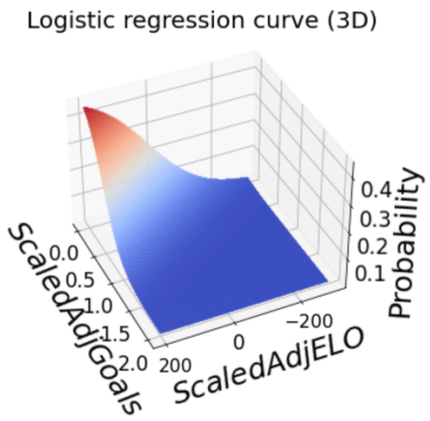

Figure 10: *Scaled Model 5*

Note: *All "0 values" for Figures 6-10 have negative x_0 values due to the Adjustment of Goals or Shots on Target*

For each scaled model created, the calculated cost decreases with the adjustments made, indicating high accuracy. Each graph has a decreasing probability of a non-zero event occurring as the Elo rating disparity approaches negative values (better teams). From Figures 6-8, *Model 1* has the highest probability of taking a greater volume of shots on target against any team.

These findings are congruous to the input data, as *Model 1*'s input players have shown that they are the best performers in decisive EPL games. *Model 3* may seem like it has a higher probability of high-volume shots in comparison to *Model 1* and *Model 2*. However, this can be explained by most players in *Model 3* having an average of 1 shot on target less than most of the classified players in *Models 1* and *2*. Therefore, a player classified to *Model 3* having 0.5 adjusted shots takes the same number of shots as a player classified in the two other models with - 0.5 adjusted shots.

If we look at *Models 4* and *5*, the visualizations suggest that players classified in *Model 4* are more likely to score a greater volume of goals. *Model 4* classified players are also more likely to score against tougher opposition in comparison to *Model 5* classified players, as evidenced by

the lower gradient descent as scaled rating disparity approaches negative values. Both prediction observations are in line with what should be happening, indicating general accuracy of the models. An interesting feature of *Model 5* is that in high negative Elo disparity games against the "Top 6", players with a low average volume of goals are highly unlikely to have any goals, hence the sum of the probabilities of all non-zero events is far below 0.3. This is not surprising as all the players from low Elo, relegation teams such as Norwich F.C have scored one goal in four matches against high Elo teams like Manchester City F.C.

Overall, with the models showing general qualitative accuracy, a verification of gambling odds can be completed. By classifying gambling data into one of these five graphs, the "*.predict"* function is used to calculate the predicted values of SkyBet's enhanced odd events.

6.2 Gambling Comparison

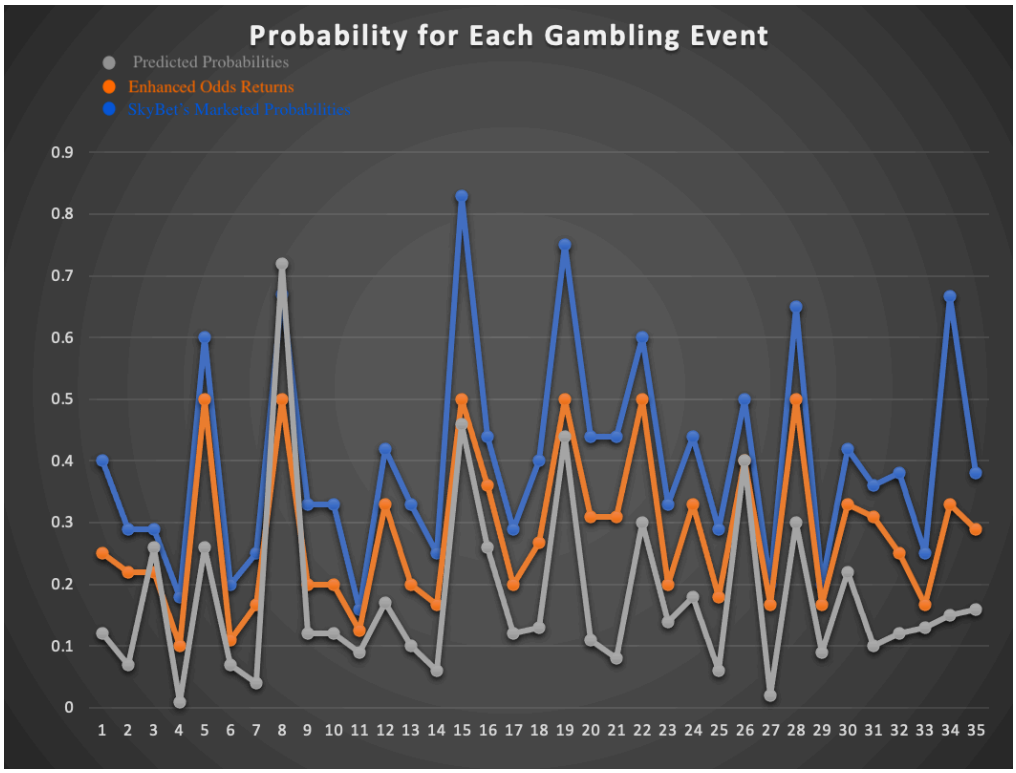

Figure 11: Comparison for each Price Boosted Enhanced Odd event between SkyBet's original probability (*Indicated in Blue)*, Price Boosted probability (*Indicated in Orange)*, and the Models' Predicted probability (*Indicated in Grey)*.

From Figure 11, we can see that almost all enhanced odds are originally marketed significantly higher than the probability of the event occurring. Out of the thirty-five data points, only one has a similar original probability and the model's predicted probability. Even with a price boost, there is still a large disparity between the new marketed odd and the actual probability. With price boosts, only four out of thirty-five events have less than 5% higher probability than the model's predicted probability. Out of the thirty-five events, only eight events occur or 22.9% of the sampled data. The study's models have an average predicted probability of 0.188 for the thirty-five models, which is closer to real life data when the one outlier event is accounted for. The models are extremely accurate for this limited dataset, despite the unpredictable nature of football.

Original odds and the price boosted give average probabilities of 0.43 and 0.31, which are a 128.7 % and 64.9 % higher than calculated probabilities respectively.

Several enhanced odd offers are almost ten times higher the actual probability. These events include 29th April's "Havertz and Ronaldo to have two or more shots on target (9/2 to 9/1)", May 1st's "Richarlison to have two or more shots on target (3/1 to 5/1)", and May 22nd's "Richarlison to have two or more shots on target (4/1 to 5/1)". None of these three events occurred or remotely had a chance of occurring, showing that SkyBet uses enhanced odds maliciously.

Additionally, most of these enhanced odd events that occur and are close to the model's predicted value are capped at a maximum deposit of 10-20 £. This results in having a "safety net" of a deposit cap in the rare occasion where they do not profit from these enhanced odds.

7. Conclusion

The five models clearly indicate that SkyBet is marketing enhanced odds at a probability disproportionately higher than the actual probability of a said event occurring. We can reaffirm the study's hypothesis of "bookies such as SkyBet are creating enhanced odds that are significantly higher than the probability of a said event occurring" as true. The graphs and models created by this study are validated with real football data and the conclusion is consistent with the gambling data. The results of this study are impressive despite the multifaceted problem it aimed to tackle. With the several adjustments made throughout the research process, the results corroborate the patterns within the original datasets.

The fact that the original odds and the price boosted probabilities are 128.7 % and 64.9 % higher than calculated probabilities respectively, show the extent of the manipulation that gamblers

fall for when gambling on enhanced odds. Given such a high disparity, it is clear that the gambling firms' usage of enhanced odds in marketing operations is highly unethical.

One possible way to extend this study for accuracy and greater application is by creating a dataset that ranges over multiple seasons and leagues. This would reduce imprecision and inaccuracy in models that predict probabilities for high shot volume/goal volume players by eliminating the problem of insufficient data. Using Europe's Top 5 Leagues over the course of three years would allow for more comprehensive data, resulting in better curve fitting.

Another way to extend this study is by accounting for a few other important team related factors. For example, teams with a similar Elo rating may be different in defensive and attacking quality, which effects the probability of scoring for or against said teams. ClubELO's *tilt*, which measures offensiveness could be used by further studies to increase accuracy. Adding opposition defensive quality and attacking quality would likely make predictions far more realistic and accurate. Additionally, the model can be applied to other betting companies to assess whether the marketed odds are significantly inflated.

8. Works Cited:

Aharoni, Udi. “SVM with Polynomial Kernel Visualization.” Www.youtube.com, 5 Feb. 2007, www.youtube.com/watch?v=3liCbRZPrZA.

Daniel, Jurafsky, and James Martin. *Speech and Language Processing*. 9 Dec. 2021.

Ganesh, Satya. “What’s the Role of Weights and Bias in a Neural Network?” *Medium*, 30 July 2020, towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f.

“How Much Does Home Field Advantage Matter in Soccer Games? A Causal ...” *ResearchGate*, May 2022, https://www.researchgate.net/publication/360640937_How_Much_Does_Home_Field_Advantage_Matter_in_Soccer_Games_A_Causal_Inference_Approach_for_English_Premier_League_Analysis.

Killick, Elizabeth A., and Mark D. Griffiths. “Sports Betting Advertising: A Systematic Review of Content Analysis Studies.” *International Journal of Mental Health and Addiction*, 24 Feb. 2022, 10.1007/s11469-022-00775-4.

McCullum, Nick. “Deep Learning Neural Networks Explained in Plain English.” *FreeCodeCamp.org*, 28 June 2020, www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/#:~:text=What%20is%20a%20Neuron%20in%20Deep%20Learning%3F.

Newall, P. W. S., Thobhani, A., Walasek, L., & Meyer, C. (2019b). Live-odds gambling advertising and consumer protection. *PloS One,* 14, e0216876. https://doi.org/10.1371/journal.pone.0216876 .

Nielson, Michael. “3.1: The Cross-Entropy Cost Function.” *Engineering LibreTexts*, 11 June 2018, eng.libretexts.org/Bookshelves/Computer_Science/Applied_Programming/Book%3A_Neural_Networks_and_Deep_Learning_(Nielsen)/03%3A_Improving_the_way_neural_networks_learn/3.01%3A_The_cross-

entropy_cost_function#:~:text=Introducing%20the%20cross%2Dentropy%20cost%20function. Accessed 25 Sept. 2022.

Pettigrew, Stephen. *Assessing the Offensive Productivity of NHL Players Using In-Game Win Probabilities*. 2015.

Robberechts, Pieter & Van Haaren, Jan & Davis, Jesse. *A Bayesian Approach to In-Game Win Probability in Soccer.* 3512-3521. 10.1145/3447548.3467194. 2021.

Roeschl, Tobias. "Animations of Logistic Regression with Python." *Medium*, 17 Nov. 2020, towardsdatascience.com/animations-of-logistic-regression-with-python-31f8c9cb420. Accessed 25 Sept. 2022.

Rudrapal, Dwijen & Boro, Sasank & Srivastava, Jatin & Singh, Shyamu. (2020). *A Deep Learning Approach to Predict Football Match Result*.

Sharma, Sagar. "Epoch vs Batch Size vs Iterations." *Towards Data Science*, Towards Data Science, 23 Sept. 2017, towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9.

Stradling, James. "Unsupervised Machine Learning with One-Class Support Vector Machines." Medium, 25 Oct. 2016, medium.com/@jamesstradling/unsupervised-machine-learning-with-one-class-support-vector-machines-129579a49d1d.

Yang, Z.R. "Artificial Neural Network - an Overview | ScienceDirect Topics." *Www.sciencedirect.com*, 2014, www.sciencedirect.com/topics/neuroscience/artificial-neural-network.