HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

PROJECT REPORT

Football Live Win Probability Prediction

TRUONG GIA BACH	TRAN DUONG CHINH	PHAN DUC HUNG
20210087	20210122	20214903

NGUYEN VIET MINH	PHAM QUANG TRUNG
20214917	20214935

Course: Introduction to Data Science

Supervisor: Associate Professor Than Quang Khoat ______________

Signature

Department: Computer Science

School: Information and Communication Technology

HANOI, 12/2023

TABLE OF CONTENTS

1 Introduction

1.1 Problem description

In the dynamic realm of sports, particularly football, uncertainty and unpredictability are driving forces that captivate fans and spectators worldwide. The twists and turns of a football match, with its multitude of possibilities and dramatic twists, keep spectators on edge. Recently, the emergence of advanced analytics and real-time data processing has provided many tools to enhance the spectator experience.

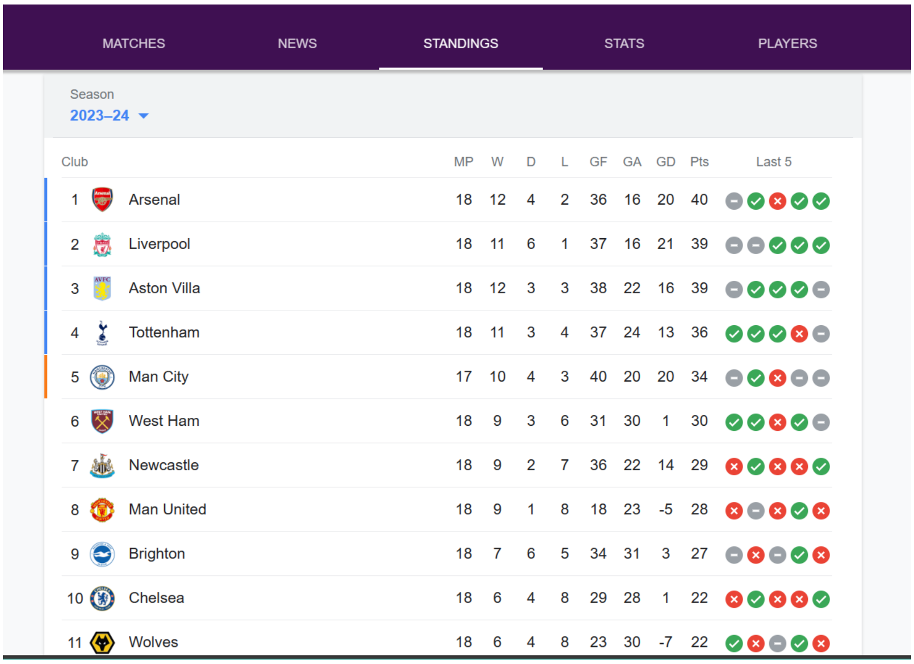
This project harnesses the recent advancement of sports tracking technology, and data analytics to predict the evolving probability of each team's victory in real-time. The integration of sophisticated algorithms and comprehensive data sets allows us to offer an insightful and dynamic visualization of a team's chances of winning throughout the course of a match.

As we worked our way through the project, this report will illuminate the methodology and technological components. By harnessing the power of data analytics, this project aims to provide insights into the factors that contribute to the outcome of a match and help coaches make informed decisions during the game.

Our expected model is designed to take input in the form of a match's game state, whether it is from a live or concluded match. This game state includes the match statistics from the initiation to the point at which we record the game state, covering elements such as the Elo rating of both teams, the specific minute, and the count of shots and passes by the two teams. These particulars are delineated in Sections 2.2: Data Integration and 4.1: Data Preprocessing. The resulting output will be expressed as the probabilities for the home team's Win, Loss, or Draw, presented as a ratio. These probabilities are collectively normalized to ensure that their sum equals 1. The calculated result which have the highest probabilities can be used as a single predict to the result, as our problem falls under the category of a multi-class classification problem in Machine Learning.

1.2 English Premier League

The Premier League is one of the most competitive and most watched football tournaments in the world. It's the biggest competition in the English football league system. The Premier League was founded over 30 years ago and has a long-recorded history. One can easily find detailed statistics for each season online, including details for each game, and even for each player in the league.



Season
2023–24

	Club	MP	W	D	L	GF	GA	GD	Pts	Last 5
1	Arsenal	18	12	4	2	36	16	20	40	− ✓ ✗ ✓ ✓
2	Liverpool	18	11	6	1	37	16	21	39	− − ✓ ✓ ✓
3	Aston Villa	18	12	3	3	38	22	16	39	− ✓ ✓ ✓ −
4	Tottenham	18	11	3	4	37	24	13	36	✓ ✓ ✓ ✗ −
5	Man City	17	10	4	3	40	20	20	34	− ✓ ✗ − −
6	West Ham	18	9	3	6	31	30	1	30	✓ ✓ ✗ ✓ −
7	Newcastle	18	9	2	7	36	22	14	29	✗ ✓ ✗ ✗ ✓
8	Man United	18	9	1	8	18	23	-5	28	✗ − ✗ ✓ ✗
9	Brighton	18	7	6	5	34	31	3	27	− ✗ − ✓ ✗
10	Chelsea	18	6	4	8	29	28	1	22	✗ ✓ ✗ ✗ ✓
11	Wolves	18	6	4	8	23	30	-7	22	✓ ✗ − ✓ ✗

Figure 1.1: Table of the Premier League 2023/24 season[1]

A Premier League season roughly coincides with a school year, starting in August and ending in May of the following year. There are 20 clubs competing in the league, and the number has not changed since the 1995 – 1996 season. A team will play 38 games against all other teams in a season, both home and away, making a total of 380 games a season.

We will collect data from 5 seasons of the EPL, 2017-2018, 2018-2019, 2019-2020, 2020-2021, 2021-2022, 2022-2023.

2 Data collection

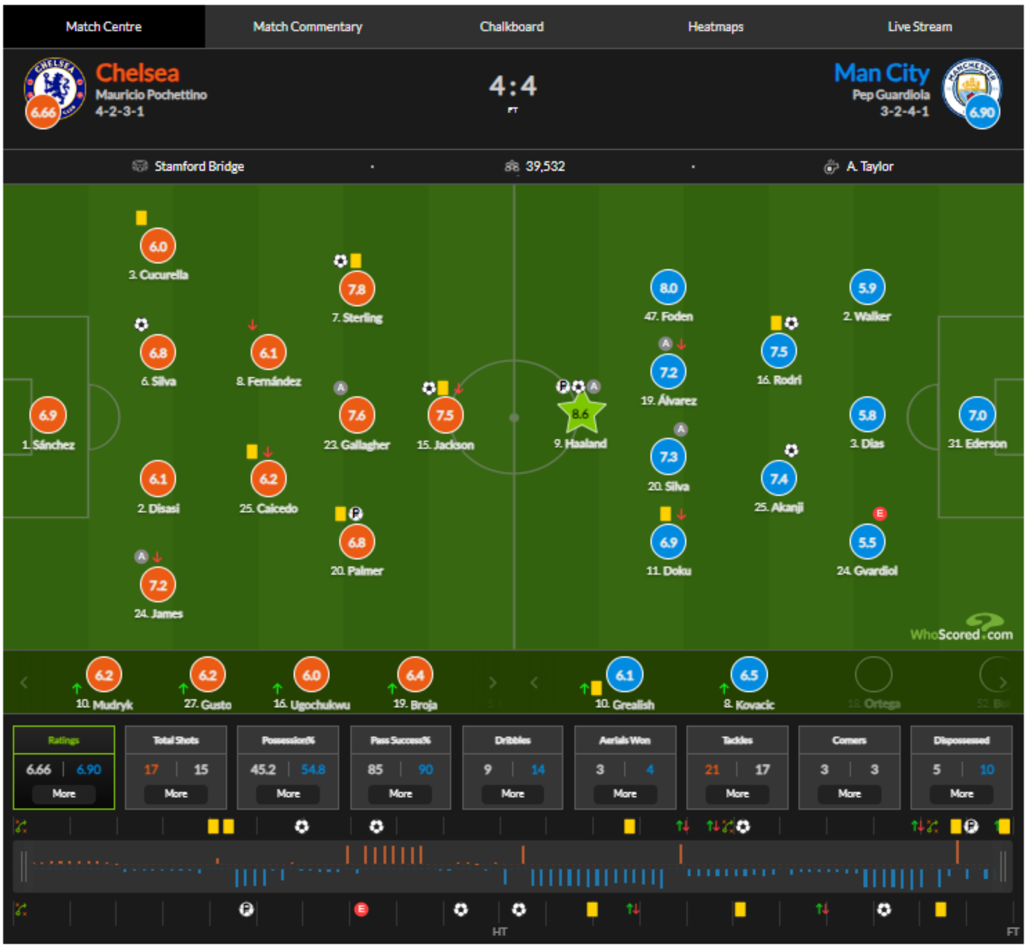
2.1 Data scraping

2.1.1 Event Data

To bolster our in-game predictions, acquiring event stream data is imperative. This dataset encompasses every event occurring throughout matches in the last five Premier League seasons, meticulously recording on-ball actions event by event. Typically sourced from broadcast footage, this data serves as a cornerstone, offering insights vital for clubs, broadcasters, gambling industries, and various other stakeholders.

We acquired this event data by crawling the website `whoscored.com`[2], which implements robust protections against scraping, using Incapsula. To overcome this obstacle, we utilized Selenium with the ChromeDriver extension to emulate authentic user behavior.

Within the Whoscored website, match reports are accessible via URLs in the format: https://whoscored.com/Matches/(match id)/. This link provides access to pre-game information, live data centers, post-match reports, and more. Our project exclusively focuses on the live data



Ratings	Total Shots	Possession%	Pass Success%	Dribbles	Aerials Won	Tackles	Corners	Dispossessed
6.66 | 6.90	17 | 15	45.2 | 54.8	85 | 90	9 | 14	3 | 4	21 | 17	3 | 3	5 | 10
More	More	More	More	More	More	More	More	More

Figure 2.1: An in-game match report from Whoscored

center via the address of format: https://whoscored.com/Matches/(match id)/Live/. Throughout each match, data on match events is tracked and recorded on the website, accessible to us as a JSON file. This file comprehensively encapsulates match information, particularly detailed event data. We scraped this JSON-formatted data and later transformed it into a tabular format for our objectives.

2.1.2 Pre-game Data

We aim to gather data that accurately represents team strength prior to the commencement of a match. There exist various approaches for this purpose, yet our chosen method involves utilizing the ELO rating of the two teams as the sole attributes for configuring the pre-match winning probabilities across all matches.

The concept of ELO ratings was initially introduced in 1978 to formally assess chess players. Subsequently, this rating system found application across numerous sports. In the realm of football, the ELO rating system has been integrated into the renowned FIFA ranking table for national football teams. The ELO rating system's concept is both straightforward and potent, given its continuous updating of club ratings following each match. Points are exchanged between the participating teams in every match: the victorious team receives a justifiable bonus to its ELO rating based on the opponent's rating, while the defeated team experiences deductions in a comparable manner. Moreover, the system factors in the advantage of playing at home and the disadvantage of playing away.

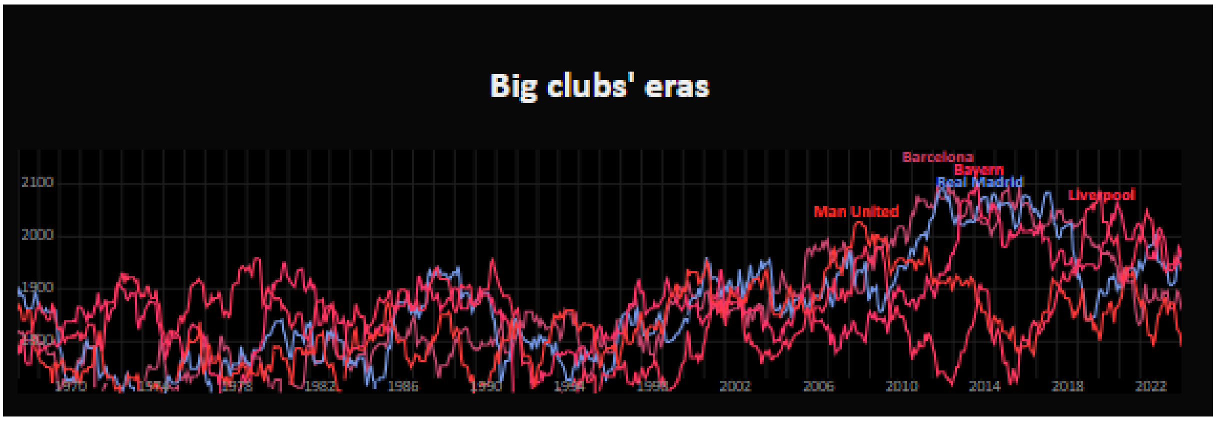


Figure 2.2: ELO ratings have been tracked since 1970

The ELO rating data is gathered from the website clubelo.com[3], renowned for its reliability in meticulously tracking and computing ELO rating data since as far back as the 1970s. The historical data encompassing club ELO ratings on a specific past date can be obtained by scraping through the accessible API, accessible via the following URL: api.clubelo.com/YYYY-MM-DD. Given our model's emphasis on in-game metrics collected during matches, the ELO rating data suffices for estimating the strengths of teams participating in each match.

2.2 Data Integration

To consolidate pre-game team strength attributes with in-game metrics from the scraped data in the previous section, data integration is necessary.

The first step to perform is to transform scraped data into tabular format. Each JSON file contains information for a specific match identified by the game ID. These JSON files with be converted to pandas.DataFrame objects. Each match yields a table containing rows that delineate individual events occurring within the match. These rows encompass diverse attributes concerning each event, such as event type, occurrence time, involved player, position on the pitch, and more. It's important to note that each event type may further divide into sub-event types, enriching our understanding. From this wealth of information, we'll subsequently process attributes for our dataset.

With the date information available for each match in the previously mentioned JSON file, we integrated this data with the API providing comprehensive ELO ratings for any specified date. This integration allowed us to combine pre-game data with in-game data seamlessly.

For every match, we iterated through the events table to track and calculate 27 distinct features. These features, when combined with ELO rating data, are integrated into the complete data file. Consequently, each row within this file encompasses 29 attributes, constituting what we term as a "game state" data instance. Detail description of each attributes can be seen in Table 2.1.

As we want to solve the problem of live win probability, we will predict the outcome from the input of a game state, which includes all of the values of attributes as listed. Many attributes are calculated to be the difference, showing how a team can out-perform the opponent on the field. All values of attributes are total sum of all previous events occur, so for a moment, we only need to know a game state, like how many goals for each team, passes and shots difference to evaluate the live win probability.

Attribute	Description
minute	The time stamp of the event measured in minute
half	The half in which the event occurs (first half and second half)
ht_elo	The Elo rating of the home team
at_elo	The Elo rating of the away team
ht_goal	The number of goals scored by the home team
at_goal	The number of goals scored by the away team
pass	The difference in the number of passes between the home team and the away team
short_pass	The difference in the number of short passes between the home team and the away team
long_pass	The difference in the number of long passes between the home team and the away team
final_3rd_pass	The difference in the number of final third passes between the home team and the away team
key_pass	The difference in the number of key passes between the home team and the away team
cross	The difference in the number of crosses made by the home team and the away team
corner	The difference in the number of corners made by the home team and the away team
big_chance	The difference in the number of big chances made by the home team and the away team
shot	The difference in the number of shots made by the home team and the away team
shot_6_yard_box	The difference in the number of shots made from the 6-yard box between the 2 teams
shot_penalty_box	The difference in the number of shots made from the penalty box between the 2 teams
shot_penalty_box	The difference in the number of shots made from the penalty box between the 2 teams
shot_open_play	The difference in the number of shots from open play between the 2 teams
shot_fast_break	The difference in the number of shots from fast breaks between the 2 teams
dipossessed	The difference in the number of possession losses between the 2 teams
turnover	The difference in the number of turnovers between the 2 teams
duel	The difference in the number of duels between the 2 teams

Table 2.1: Data attributes description

tackle	The difference in the number of tackles between the 2 teams
interception	The difference in the number or interception between the 2 teams
clearances	The difference in the number of clearances between the 2 teams
offside	The difference in the number of offsides between the 2 teams
yellow	The difference in the number of yellow cards between the 2 teams
red	The difference in the number red cards between the 2 teams
result	The result of the match (e.g., win, loose or draw) for the home team

Table 2.1: Data attributes description

2.3 Data cleaning

Data cleaning plays a pivotal role in ensuring the accuracy and reliability of the "Football Live Win Probability" project. In order to extract meaningful insights from the football data, it is imperative to address the inherent challenges associated with the initial data.

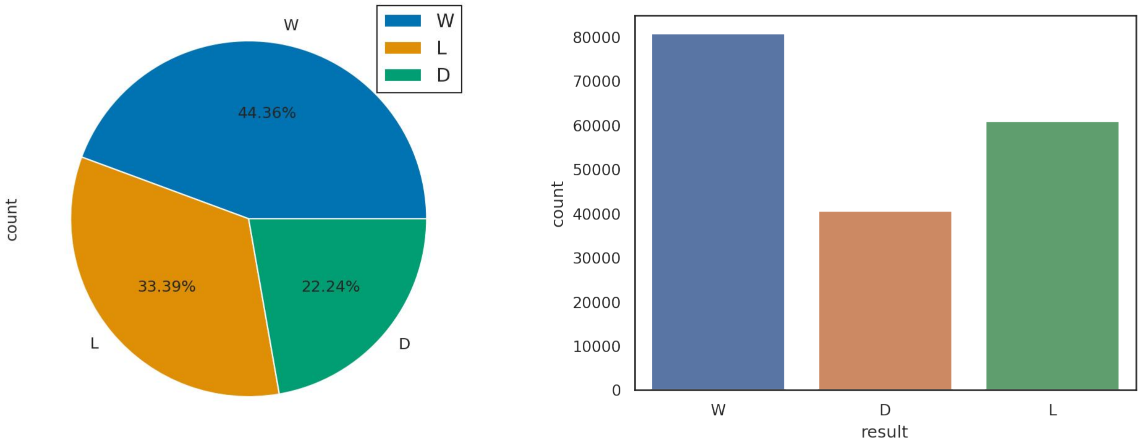
After transforming the event data into game state data, our aim was to condense this information so that each game state represented a minute timestamp. This restructuring involved combining data pertaining to the same minute, ensuring that the number of game state instances for a match equaled the number of minutes within that match. However, we encountered missing timestamps, signifying periods when no events occurred (e.g., interruptions due to injuries consuming minutes during a match). To address this issue, we assigned the game state data for missing minutes to the nearest preceding game state. This approach naturally balanced our dataset in terms of minute attributes.

Additionally, we conducted a cleanup of our dataset by removing erroneous data entries, often stemming from data entry errors on Whoscored during the match.

3 Exploratory Data Analysis (EDA)

3.1 Attributes distribution

All match files from 5 seasons were added up to a file to show the distribution of all attributes for all data points.



(a) Percentage of match by results

(b) Number of match by results

Figure 3.1: Visualization of match results

From the first look, we can observe an apparent imblance within the dataset, with the home team achieve victories for a significant share of the matches (around 44 %) follow by loses and draws. This eminent imbalance can be intepreted as an effect of “home team advantage” phenomenon, where home team benefits from a wide range of advantages, such as the familiarity of the fields, the dominance of the fan base.

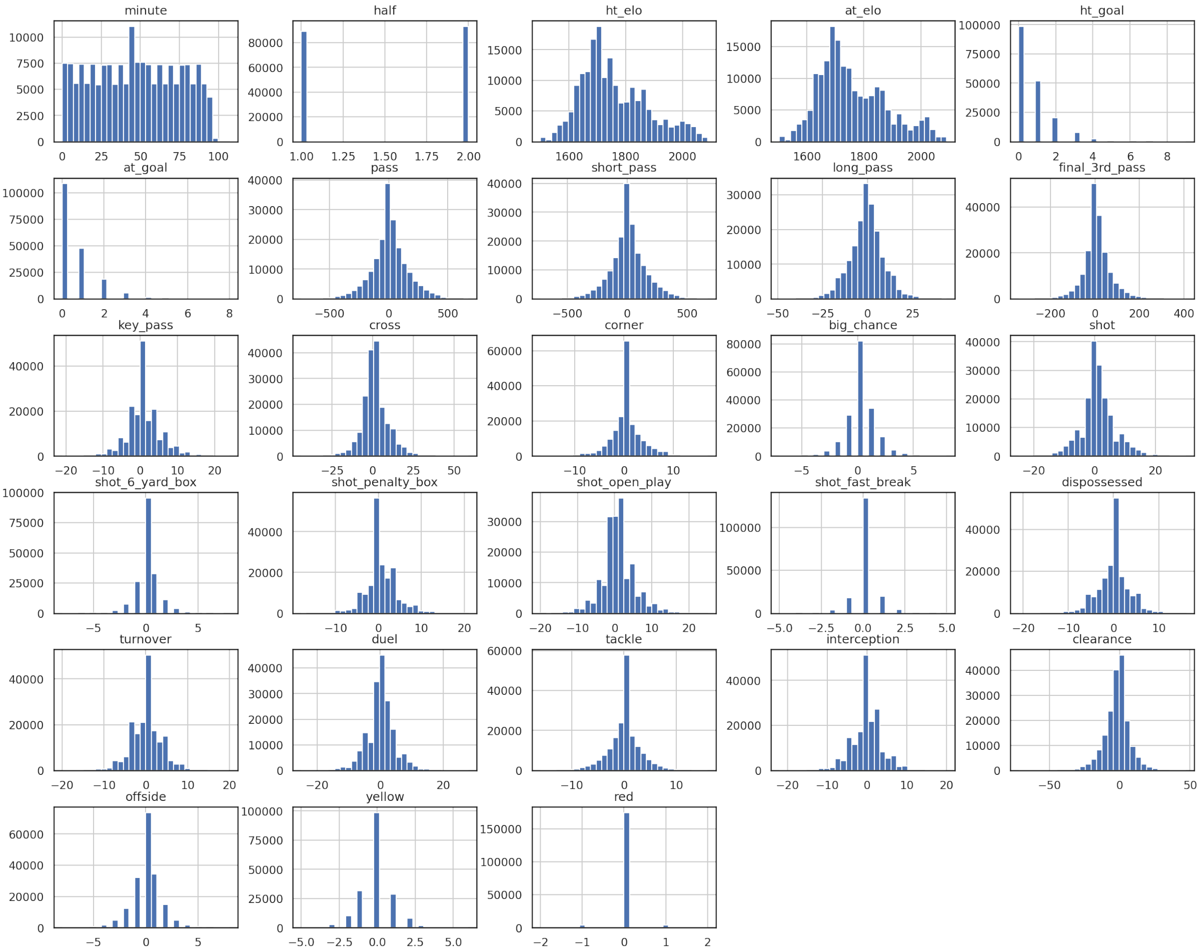


Figure 3.2: Distribution of all attributes in the data

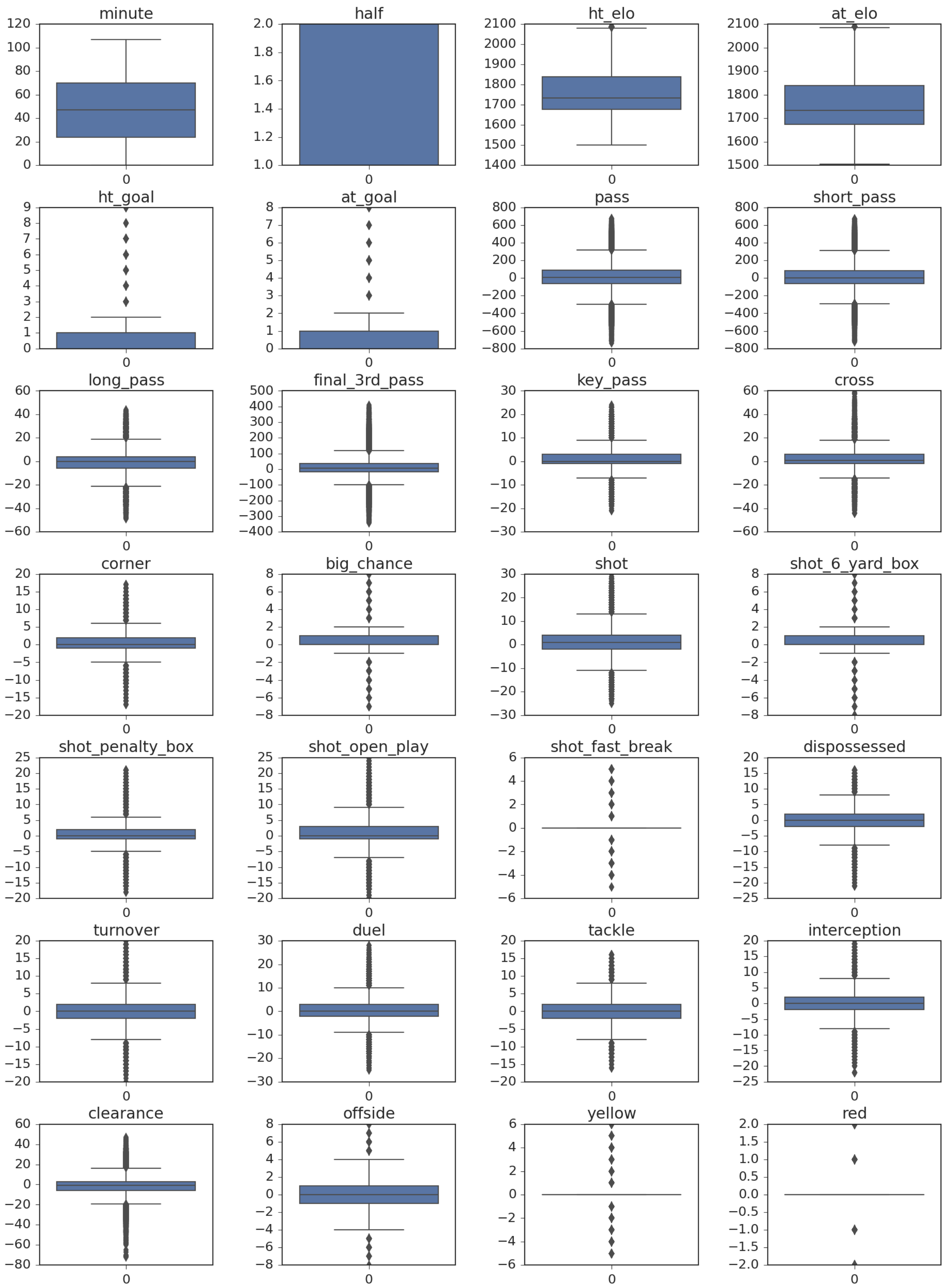


Figure 3.3: Boxplots of all attributes in the data

From observing the distribution of feature values, we make the following conclusions

1. It is noticeable that the majority of attributes show Gaussian-like distribution, with mean zero. This is reasonable due to the large number of instances in the data set and many

attributes indicates the difference between the number of events created by 2 opposing teams,

2. The distribution of elo-score for both home team and away teams look similar, which can be explained by the fact that one football acts as home and away team during the tournament,
3. Several features contain a notable presence of outliers. Something must be done to eliminate these abnormal value before we build our model.

There are differences in the record of numbers of minutes, even though data is collected across a fixed number of matches. This variance arises from our data collection process on Whoscored, where the system fails to recognize any events during specific minutes. These interruptions on the field, such as fouls, injuries, or conflicts, prompt referees to halt the match, resulting in the absence of recorded events and statistics for those minutes. Fortunately, such occurrences are infrequent, and overall, our data remains robust and stable.

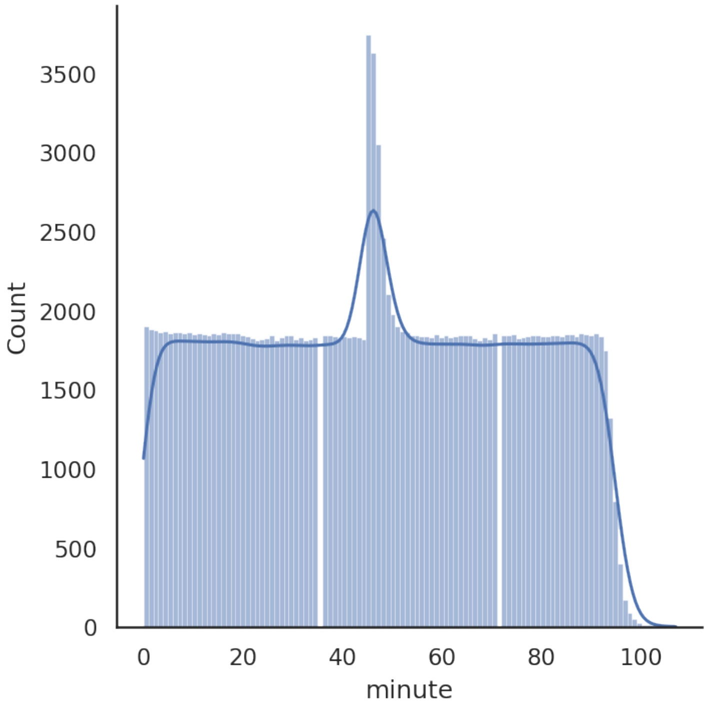


Figure 3.4: Distribution of minutes

Notably, for minutes beyond 45 and before 50, the number of events may be twice the average. This is due to extra time played in the first half, and these minutes are labeled as 46, 47, etc., mirroring the corresponding minutes in the second half. Conversely, for minutes beyond 90, the numbers may decrease, indicating a decline in events as the match extends beyond the regular game time.

The highest number of goals observed in our dataset for a single match is 9. The likelihood of a match featuring 2 goals is the greatest, with the probability decreasing as the number of goals increases. Also, the number of goals shows a Possion-like distribution, with $\mu = 2$.

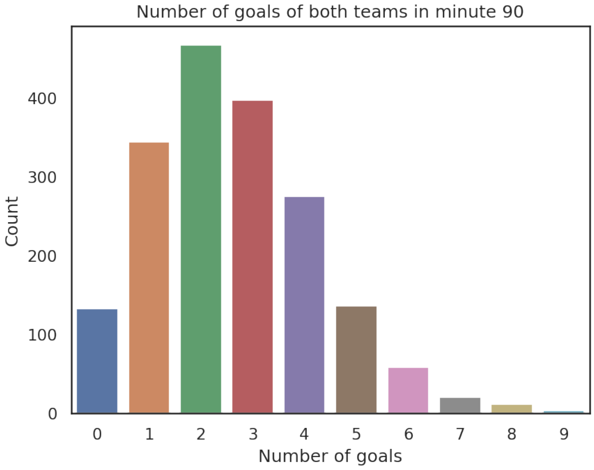


Figure 3.5: Total goals of both teams in minute 90

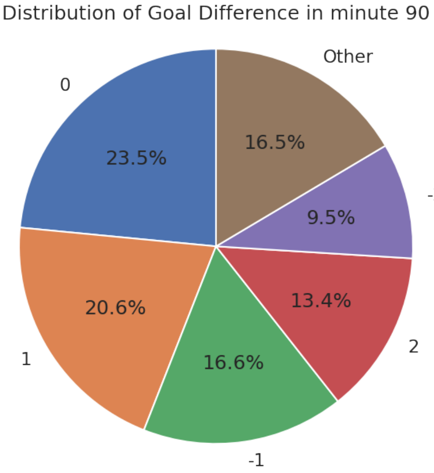
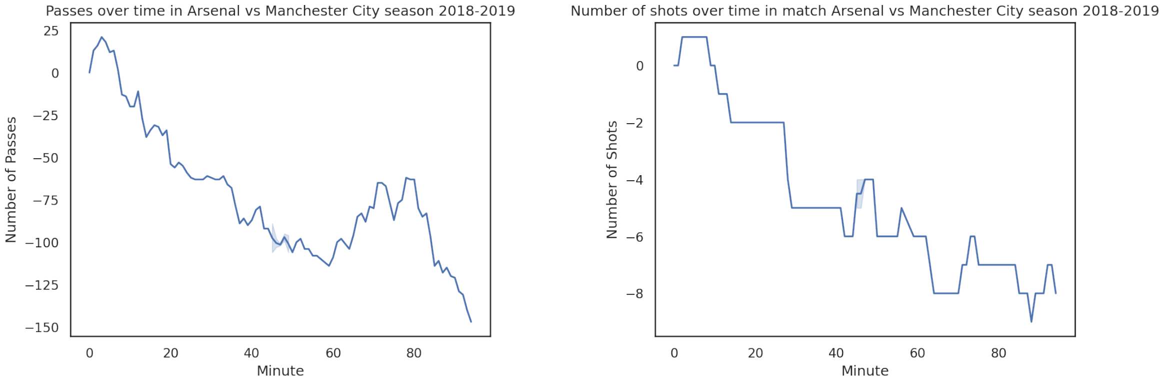


Figure 3.6: Distribution of Goal difference in minute 90

Goal differences in the 90th minute have the highest probability to equal 0, as we are close to witness a draw match. Again we observe that the distribution is right skewed as a consequence of the "home team advantage". Goal difference absolute value observes value equal 1 with highest ratio, at 37.2%.



(a) Total of Passes over time

(b) Total of Shots over time

Figure 3.7: Distributions over time in the match Arsenal vs Manchester City season 2018 - 2019

The total number of passes made shows the possession of each team in the field. Possession is the way how a team control and play with the ball, which typically gives that team the probability to score. The event where possession is lost to another team is called a turn-over. On the field, there are 2 teams competing for only one ball, so at a time only one team can have control over the ball. We can see the possession of 2 teams by the number of passes they make, as passing account for a significant number of events happen during a football match. When a team have high passes over the other team, they have higher possession rate.

3.2 Correlation Analysis

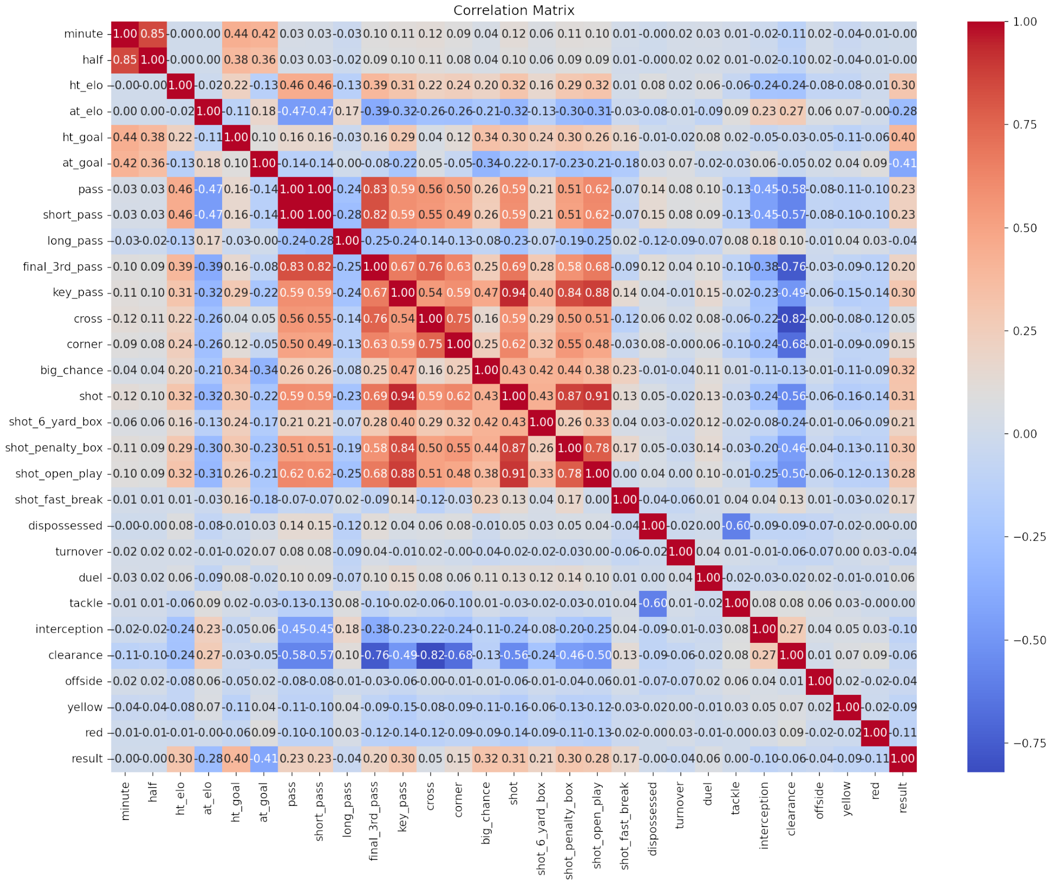


Figure 3.8: Correlation Matrix of all attributes

Range of Correlation Coefficient Values	Level of Correlation	Range of Correlation Coefficient Values	Level of Correlation
0.80 to 1.00	Very Strong Positive	-1.00 to -0.80	Very Strong Negative
0.60 to 0.79	Strong Positive	-0.79 to -0.60	Strong Negative
0.40 to 0.59	Moderate Positive	-0.59 to 0.40	Moderate Negative
0.20 to 0.39	Weak Positive	-0.39 to -0.20	Weak Negative
0.00 to 0.19	Very Weak Positive	-0.19 to -0.01	Very Weak Negative

Table 3.1: Range of Correlation Coefficient Values and the Corresponding Levels of Correlation

In order to optimize the efficacy of our model, we will choose the attributes that affect bear the strongest correlation with the result, both positive and negative. As observed in the heatmap, many pairs of attributes show strong correlation, which can be dropped as they create redundancy in the dataset and may cause the model to overfit.

Approximately 15 percent of feature pairs exhibit a correlation exceeding 0.4. Therefore, we opt to reduce the dimensionality of the data, transforming the variables to extract only the significant information.

Attribute 1	Attribute 2	correlation
pass	short pass	0.998478
key pass	shot	0.919022
shot	shot open play	0.878532
minute	half	0.861287
key pass	shot open play	0.849293
shot	shot penalty box	0.836486
key pass	shot penalty box	0.806242
pass	final 3rd pass	0.777761
short pass	final 3rd pass	0.773254
cross	clearance	0.772210

Table 3.2: Top 10 feature pairs

The Elo ratings of both team are the only pre-game statistic. The Elo difference in a match is often used as an indicator of the relative strengths of the two teams. The probability of a win, draw, or loss can be estimated based on the Elo difference. Typically, a higher positive Elo difference for a team suggests a higher probability of winning, while a higher negative Elo difference indicates a higher probability of losing.

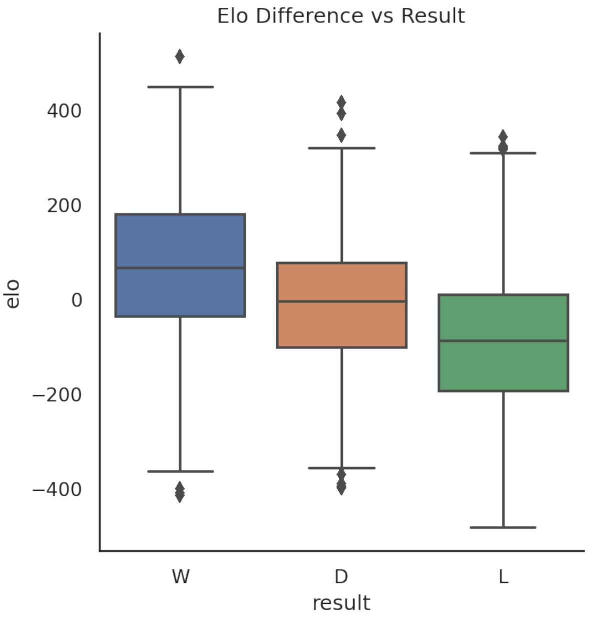
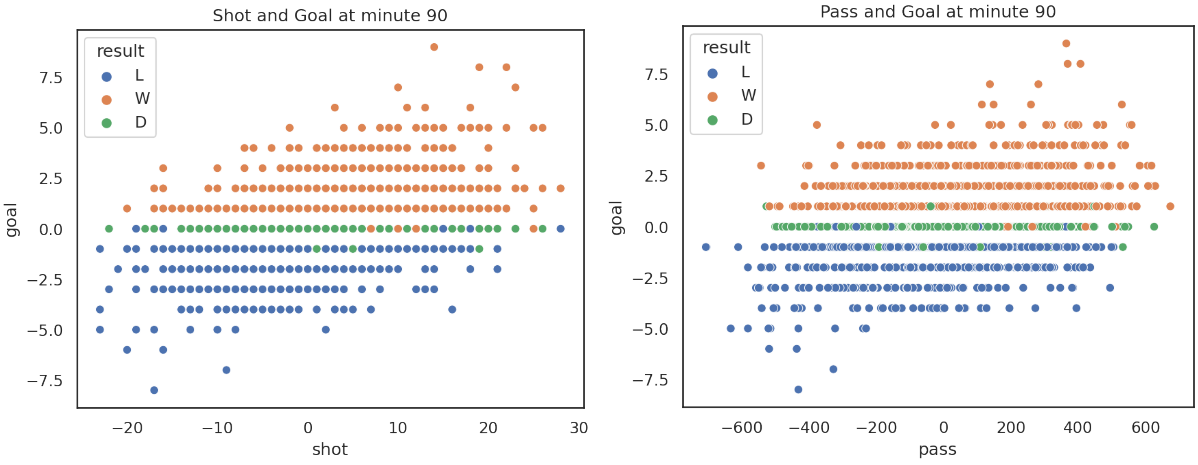


Figure 3.9: Elo difference and Result

In our data, Elo ratings of both teams show a good correlation to the result, as the team with higher elo will be likely to win the match. Though it can have many cases that the team with higher elo encounter a loss, but the higher the difference between the Elo, the more certain the prediction from elo-score becomes.

Given that the outcome of a match is determined primarily by the recorded goal, our focus will be on identifying attributes that exhibit correlation with this important factor. The goal stands as the sole officially documented value that decides the match result.



(a) Pass and Goal

(b) Shots and Goal

Figure 3.10: The Correlation of Pass and Shot to the Goal value in minute 90

Team with higher passes are taking the possession and have high probability to take control of the game. Except from waiting for the opponent's own goal, a team can only score by making shots. Not all the shots result in a goal, but a team who attempt more shots toward the opponent's end have higher chance to actually score a goal. It also makes sense that team with higher possession over the ball have greater number of shots or attempts.

Shot and pass attributes exhibit strong correlation. It makes sense that the team with dominant possession over the ball have better chance of making a shot. Consequently, the team engaging in passing activities more actively will create more shots and thereby more likely to score a goal. From this, we can conclude that pass-related attributes display a strong predictive power over the outcome of a match.

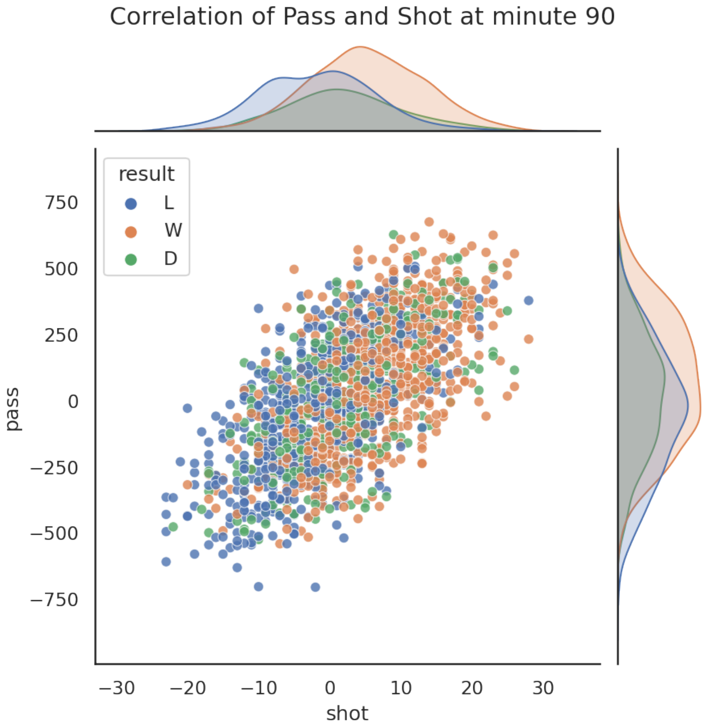


Figure 3.11: Pass and Shot Distribution

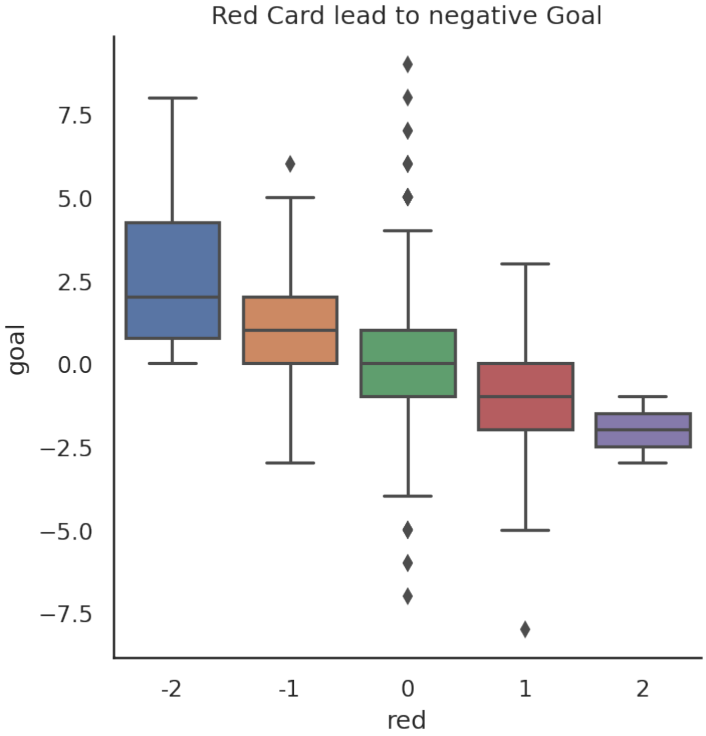


Figure 3.12: Red Card and Goal distribution

In a match, a player will receive a red card for serious offenses, violent conduct, dangerous play or receive 2 yellow cars. He then will be dismissed from the field and also get banned for several next matches, based on the severity of the foul. For his team, playing with fewer players creates an apparent disadvantage, making it more challenging to defend against the opposing team's attacks and limiting offensive opportunities. The team with red cards tend to retreat to preserve the current state of the match, making them less likely to score any more goals.

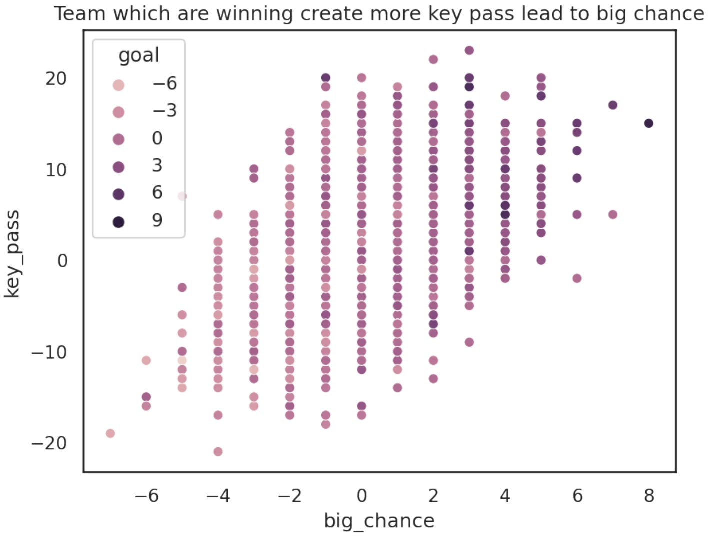


Figure 3.13: Big chance and Key pass Distribution

Big chance refers to a goal-scoring opportunity that is considered significant and has a high probability of resulting in a goal. In a match, big chance can be made by significant key pass, a term refer to the passing action that give the player that receive the ball face a big opportunity to make a shot to score. As the result consider only data that happen in previous minutes, we can not know when a team can make key pass. Figure 3.13 shows that the team that are taking the lead in goals scored make more big chances and key passes.

4 Modelling

Our approach initiates with the implementation of a diverse array of machine learning models, encompassing methodologies like Logistic Regression, K-Nearest Neighbors Classifier, Decision Tree Classifier, Random Forest Classifier, and other ensemble learning models. Subsequently, we conduct an evaluation of the collective performance of these models to discern their suitability and efficacy for our task. This assessment involves utilizing cross-validation techniques on the previously partitioned training dataset.

Figure 4.1 demonstrates that among the evaluated models, specifically the K-Nearest Neighbors Classifier, Decision Tree Classifier, Random Forest Classifier, and XGBoost Classifier, a significant performance superiority is evident. Consequently, we decide to focus our attention on these models.

After fine-tuning these four models, the best estimator from each model is passed into the Calibrated Classifier to create a model with improved calibration attributes.

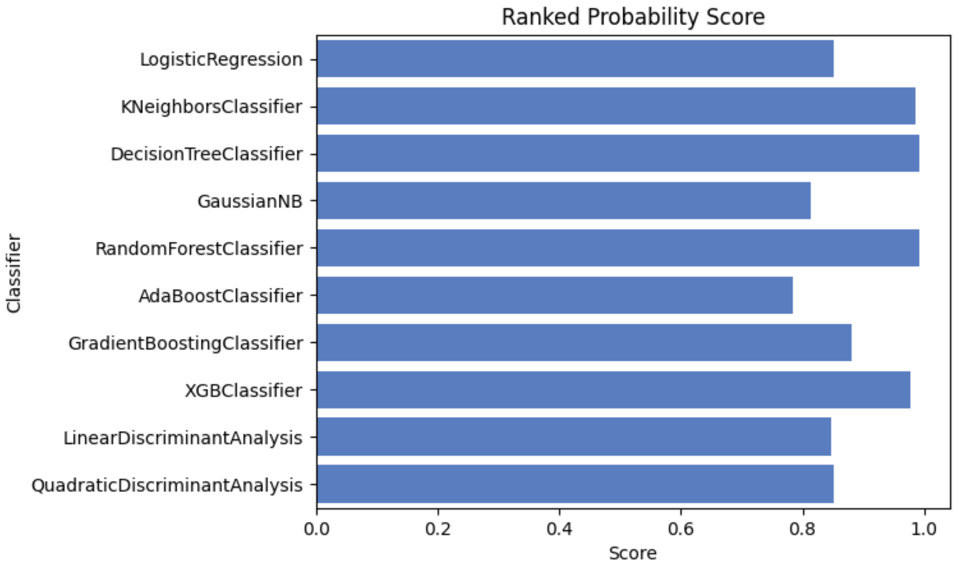


Figure 4.1: Models performances on the dataset

4.1 Data Preprocessing

4.1.1 Dimensional Reduction

From the data collected, we choose the important features to create a smaller data for modelling. We will choose the attributes with high correlation to the feature *result*.

For features that are near to have the same representation as *pass*, *short_pass*, *final_3rd_pass* or *shot*, *shot_penalty_box*, *shot_6_yard_box*, we will choose only a feature to represent, as a way to reduce the dimension of data.

Our data for modelling include 14 features: *minute*, *half*, *ht_elo*, *at_elo*, *ht_goal*, *at_goal*, *pass*, *shot*, *key_pass*, *final_3rd_pass*, *corner*, *big_chance*, *red*, *result*.

4.1.2 Category Encoding

The attribute *result* has the values as W, D, L for categorize Win, Draw and Loss. Model can only work with numerical value, so we will encode the result of attribute *result*. We choose to encode L to 0, D to 1 and W to 2.

4.1.3 Numerical Normalization

We use Standard Scaler from the sklearn.preprocessing[4] library as *StandardScaler* to scale values of numerical attributes to normal distribution. Therefore, the algorithms can learn more effectively.

4.2 K-Nearest Neighbors

KNN is a non-parametric and lazy learning algorithm, meaning it doesn't make assumptions about the underlying data distribution during training and delays decision-making until predictions are required. While KNN is intuitive and easy to implement, its performance can be sensitive to the choice of distance metric and the dimensionality of the data, making it important to carefully tune these parameters for optimal results.

4.3 Decision Tree

Decision tree classification is a machine learning algorithm used for classification task, as it has advantages like interpretability, simplicity. In a decision tree, the data is split into subsets based on certain features, and the process is recursively applied to each subset. The decision tree structure consists of nodes, where each node represents a test on an attribute, branches represent the outcome of the test, and leaves represent the class labels. Each branch pruning is based on outcome calculated from a decision rule. The leaf nodes will be the result for the algorithm, with probability of Loss, Draw and Win.

4.4 Random Forrest

Random Forest is a powerful and versatile machine learning algorithm that belongs to the ensemble learning family. Developed as an extension of decision trees, this algorithm is good in both classification and regression tasks. What makes Random Forest special is its ability to create a multitude of decision trees during training and then aggregate their outputs to enhance predictive accuracy and generalization. Random Forrest is used to handle high dimension data, noisy data and large datasets.

4.5 XGBoost

XGBoost, short for eXtreme Gradient Boosting, is a machine learning algorithm leveraging the gradient boosting technique to create powerful predictive models. XGBoost employs a gradient boosting framework, sequentially training a series of weak learners (usually decision trees). Each tree corrects the errors made by the ensemble of existing trees, leading to a robust and accurate predictive model. It is well-known for its efficiency, scalability and regularization, which makes it suitable for classification problems.

4.6 Fine-tuning

Fine-tuning can help find hyperparameters that can make the model better. We use Bayesian Search Cross Validation method, represented as BayesSearchCV from skopt libabry (scikit-optimize).

We opted for Bayesian search over the two common methods, Grid search and Random search, due to its efficient optimization of train time. The primary distinction lies in how the Bayesian search algorithm refines its parameter selection in each round based on the scores from previous rounds. Instead of randomly selecting the next set of parameters, this algorithm optimizes its choices, often arriving at the best parameter set more swiftly compared to the other two methods. Essentially, it strategically narrows down the search space, discarding ranges that are less likely to yield the optimal solution. Specifically for our task, Bayesian Search notably reduced the training time, a significant advantage given our large dataset, which comprises over 100,000 data instances.

4.7 Calibrated Classification

A calibrated classifier is a model used in machine learning that not only predicts outcomes but also provides a probability estimate for those predictions[5]. In many cases, the initial model doesn't accurately reflect the true likelihood of the prediction being correct. Then we need calibration refers to the alignment of predicted probabilities with the actual likelihood of an event happening. A well-calibrated classifier gives probability estimates that closely reflect the true probability of the predicted outcome.

A calibrated classifier wil,l be used as a second model fit after we already fitted it into the first model in order to give the truest probability possible

In our problem, we use a calibrated classifier provided by sklearn[4] library: CalibratedClassifierCV in sklearn.calibration[6], we test on 2 methods'sigmoid', 'isotonic' choosing the best score one and set cv="prefit". In this case, no cross-validation is used and all provided data is used for calibration. The user has to take care manually that data for model fitting which us have already stratified in code and fitted into the first model above.

How does all this classification work ?

Model Calibration gives insight or understanding of uncertainty in the prediction of the model and in turn, the reliability of the model to be understood by the end-user, especially in critical applications. Model calibration is extremely valuable to us in cases where predicted probability is of interest so our problem with probability for 3 cases W, D, L is very need of this classification.

5 Evaluation

5.1 Methodology

The 1900 matches spanning five seasons are divided into two sets for training and testing purposes, with 20% of the matches (380 matches) allocated for evaluating the models' performance at the study's conclusion. This designated test dataset is reserved until the final stages to impartially assess the models' capabilities on previously unseen data instances. The assessment of model performance relies on two primary scoring metrics: the Ranked Probability Score (RPS) and the Expected Calibration Error (ECE).

5.2 Score Metrics

5.2.1 Ranked Probability Score

Introduced in 1969, the Ranked Probability Score (RPS) stands as a rigorously proper scoring method renowned for its sensitivity to distance. Specifically designed for evaluating probability forecasts of ordered variables, RPS calculation for a singular problem instance involves considering 'r' potential outcomes, utilizing forecasts ('pj') and observed outcomes ('ej' at position 'j'). This score reveals the disparity between cumulative forecast distributions and actual observations. Notably, RPS exhibits a negative bias, notably pronounced with smaller ensemble sizes, as it penalizes differences between forecasted cumulative distributions and realized outcomes.

The formula for RPS is given by:

$$\text{RPS} = \frac{1}{r-1} \sum_{i=1}^{r} \left(\sum_{j=1}^{i} p_j - \sum_{j=1}^{i} e_j \right)^2$$

$$\text{where} \begin{cases} p_j & \text{is the forecasted probability of outcome } j, \\ e_j & \text{is the actual probability of outcome } j, \\ r & \text{is the number of outcomes.} \end{cases}$$

The RPS, computed as the sum of squared differences between cumulative forecast probabilities derived from ensemble members and observations, delineates multi-categorical probabilistic forecasts (0 indicating non-occurrence and 100 denoting occurrence). This score ranges between 0 (indicating a perfect forecast) and 'n-1' (representing the worst possible forecast), with 'n' signifying the categories involved. In instances involving binary forecasts (the minimal categorical scenario for probabilistic forecasts), the RPS mirrors the Brier Score (BS), thereby spanning between 0 and 1.

For our problem, the Ranked Probability Score outperforms other scoring metrics because football match outcomes ('W', 'D', 'L') follow an ordinal (ranked) scale type (i.e., a draw outcome is closer to a win than a loss). This characteristic enables us to distinguish between better and poorer predictions. For instance, a prediction of [40% win, 30% draw, 30% loss] is considered better than [40% win, 20% draw, 40% loss] when the result is a win based on the RPS.

5.2.2 Expected Calibration Error

In classification problems, machine learning models produce estimated probability or confidences. This indicates the certainty associated with the model's prediction. However, for most models, this probabilty does not aligned with the true frequences of the ground truth which the model is trying to predict. Therefore, they need to be calibrated. Model calibration aims at adjusting these confidences to reflect the true probability, which makes the model more accurate and reliable. Expected Calibrated Error calculates weighted average error of the estimated probabilities, thereby results in a single metric which can be used to compare different models. Down below is the formula for Expected Calibration Error (ECE):

$$
\mathrm{ECE} = \sum_{i=1}^{B} \frac{N_i}{N} \left|\mathrm{acc}(B_i) - \mathrm{conf}(B_i)\right|
$$

$$
\text{where} \begin{cases} B & \text{is the number of bins or buckets,} \\ N_i & \text{is the number of examples in bin } i, \\ N & \text{is the total number of examples,} \\ \mathrm{acc}(B_i) & \text{is the accuracy of the model on examples in bin } i, \\ \mathrm{conf}(B_i) & \text{is the average predicted probability for examples in bin } i. \end{cases} \quad (1)
$$

5.3 Model Performance

As mentioned above, we will use the Ranked Probability Score to measure the model accuracy and the Expected Calibration Error to measure the model calibration. A model is called calibrated when the probabilities estimated resemble the real frequency of positive cases. For example, if we take a sample of 100 cases scored by the model whose mean probability of being positive is 60%, then we expect the real rate of positive cases in this sample to be 60%. Calibration is an important characteristic to have in probabilistic models.

To assess the calibration, we use a type of chart called calibration curve. It can be build by:

1. Score the test set and sort the probabilities by increasing order.

2. Divide the test set in a number of bins with equal probability width.

3. For each bin, compute the mean probability outputed by the model and the true rate of positive cases.

4. Plot a graph where x-axis is the mean probability outputed and the y-axis is the rate of positive cases. Then, each bin corresponds to a dot in this graph.

The perfectly calibrated model will have a diagonal curve because in all segments of probability the mean predicted value will be equal to the true fraction of positives.

5.3.1 K-Nearest Neighbor

The K-Nearest Neighbor model has an RPS score of 0.73 and an ECE score of 0.38. The model does not yield satisfactory results due to unsuitable attributes concerning the probabilistic classification problem.

Figure 5.1 illustrates the calibration curve of the K-Nearest Neighbor model without the Calibrated Classifier. As observed, the KNN model exhibits poor calibration properties as the predicted probabilities do not align with the true probabilities.

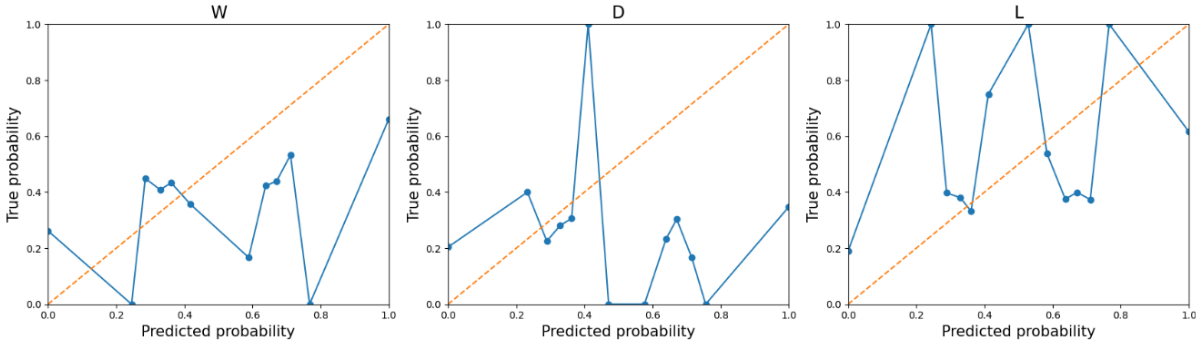


Figure 5.1: KNN Calibration Curve

After passing through the calibrated classifier, a notable enhancement in the calibration curve is evident, even though the RPS score and the ECE score remain unchanged.

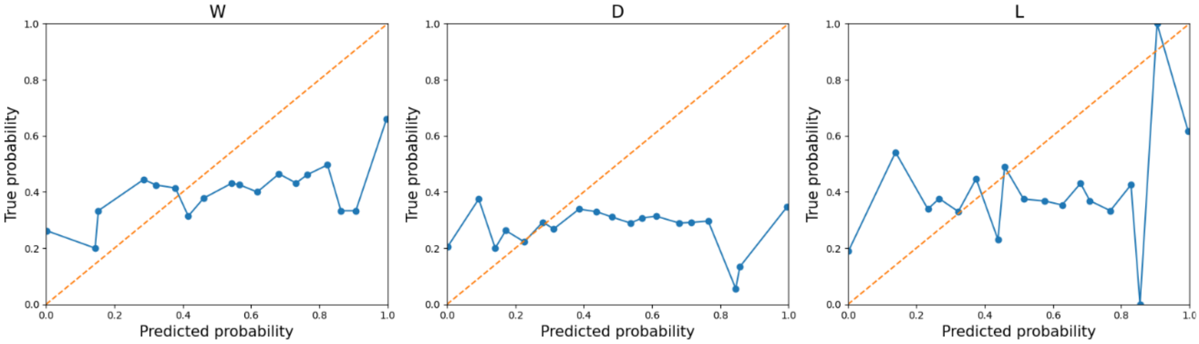


Figure 5.2: Calibrated KNN Calibration Curve

5.3.2 Decision Tree

The decision tree exhibits remarkably high performance in both accuracy and calibration, with an RPS of 0.82 and an ECE of 0.17. The calibration curve in Figure 5.3 clearly demonstrates proximity to a diagonal line, indicating that the predicted probabilities closely align with the true probabilities.

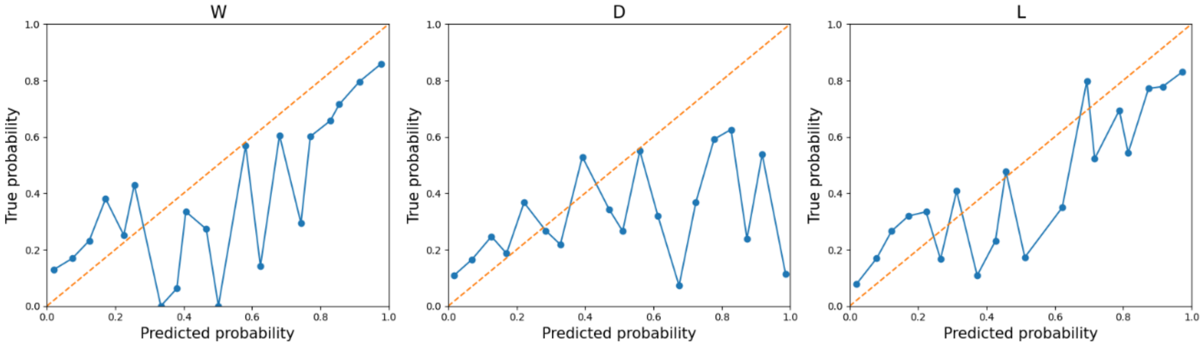


Figure 5.3: Decision Tree Calibration Curve

The Calibrated Classifier further enhances the calibration curve of the Decision Tree predictions, as we can see on Figure 5.4. Additionally, there is a slight improvement in both RPS and

ECE scores on the test dataset. Overall, the Decision Tree performs exceptionally well on the Live Winning Probability problem.

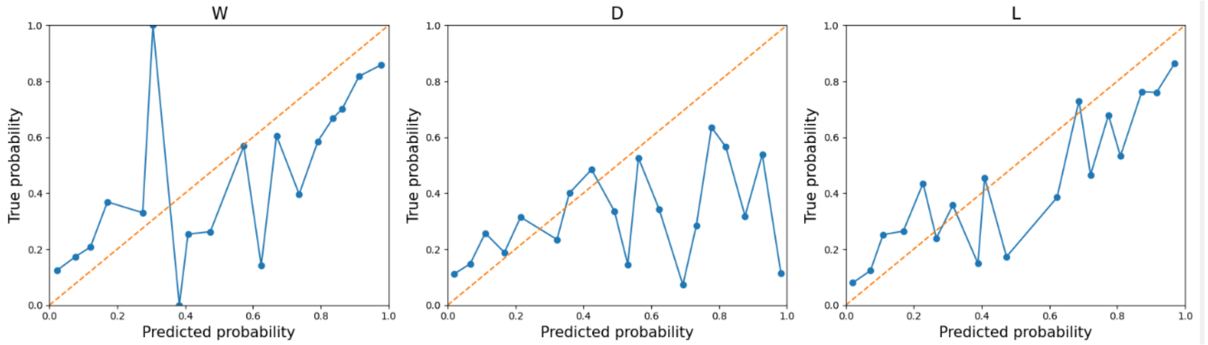


Figure 5.4: Calibrated Decision Tree Calibration Curve

Based on the diagram above, we can assert that Decision Tree algorithm outperforms KNN in this . The predicted probability lines are closer and more stable to that of KNN.

5.3.3 Random Forrest

The Random Forest model yielded excellent results on our scoring metrics, achieving an RPS of 0.80 and an ECE of 0.18.

Additionally, Figure 5.5 illustrates a calibration curve that closely aligns the predicted probabilities with the true probabilities. This alignment creates an almost diagonal calibration curve across all three outcomes: ’Win,’ ’Draw,’ and ’Lose’.

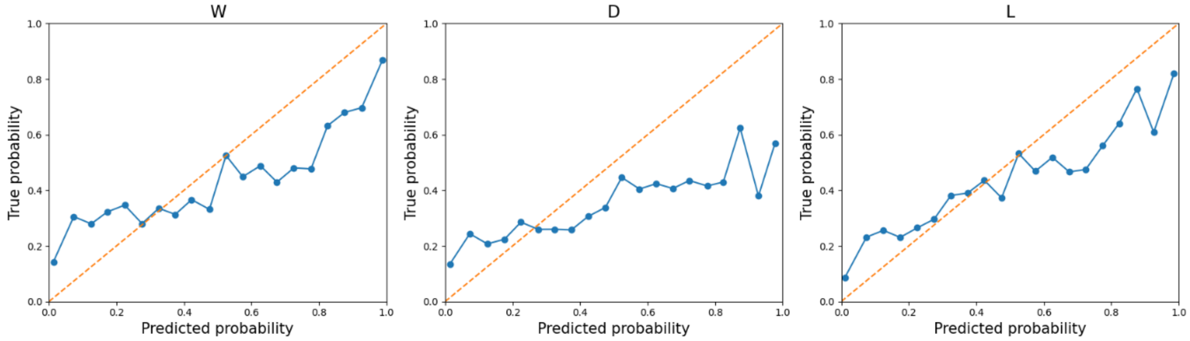


Figure 5.5: Random Forest Calibration Curve

The calibrated classifier actually underperformed compared to the Random Forest model. This is evident in both the calibration curve shown in Figure 5.6 and the lower scores of RPS and ECE. These observations suggest that this strategy may not be suitable for the Random Forest classifier.

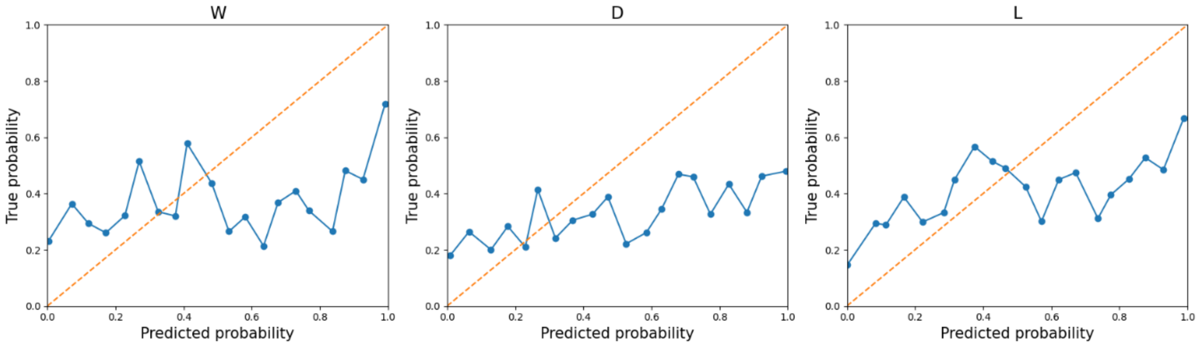


Figure 5.6: Calibrated Random Forest Calibration Curve

5.3.4 XGBoost

The calibration curve depicted in Figure 5.7 presents an almost horizontal line, indicating consistent deviations of predicted probabilities across various intervals or bins from the true probabilities. This situation typically suggests that the model's predicted probabilities do not accurately reflect the actual probabilities for different outcomes or classes. Essentially, the model might consistently overestimate or underestimate the likelihood of specific events occurring, irrespective of the probabilities assigned.

It appears that the XGBoost model might not be well-suited for our task, given its relatively low RPS of 0.69 and high ECE of 0.43.

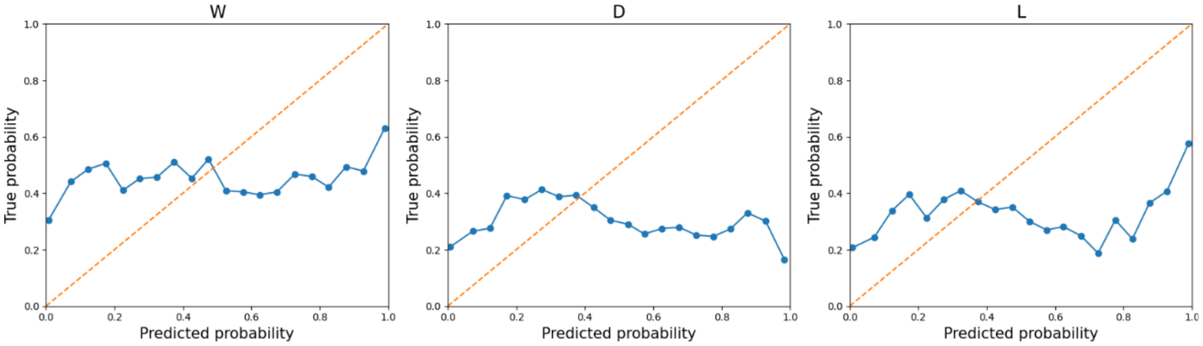


Figure 5.7: XGBoost Calibration Curve

An observation worth noting is that the Calibrated Classifier strategy did not aid our XGBoost model in terms of accuracy and calibration. Overall, among the four models, XGBoost exhibited the poorest performance on the test dataset used for model evaluation.

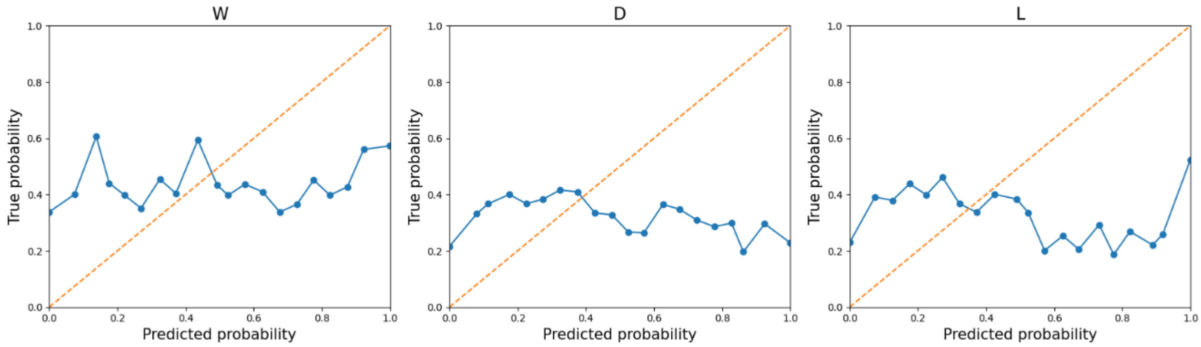


Figure 5.8: Calibrated XGBoost Calibration Curve

5.3.5 Result

Metric Model	RPS	ECE
KNN	0.73	0.38
Calibrated KNN	0.72	0.39
Decision Tree	0.82	0.17
Calibrated Decision Tree	0.82	0.17
Random Forest	0.80	0.18
Calibrated Random Forest	0.75	0.34
XGBoost	0.69	0.43
Calibrated XGBoost	0.65	0.49

Table 5.1: Result Table of all algorithms

In conclusion, both the decision tree and the random forest outperform the other models. By utilizing the calibration curve, we can select the Decision Tree as the best model for our problem.

6 Conclusion

From the data collected from `whoscored.com` and `clubelo.com`, we do the data intergration, data cleaning to create a full data for pre-game and in-game statistics of 5 seasons of EPL (2017-2023). Analysis helps to find data insights, especially good features that highly correlate to the result of the match. It shows that a team with higher possession, higher elo rating in pre-game statitics will have more probability to get the better result. We choose 14 features and preprocess to make a stable data for the models.

To generalize and predict the outcomes of the data, we choose 4 algorithms for modelling, K-Nearest Neighbor, Decision Tree, Random Forrest and XGBoost. We also apply calibration as calibrated classification in the model to comapare the result. The Decision Tree and Random Forrest return the best result, out-performs others algorithm. It also show that the calibration does not improve our algorithms result.

For the future development, we will improve the performance of calibration for all algorithms, re-evaluate the scoring metrics to find the most suitable metrics for this problem. Exploring some deep learning technique as Long Short-Term Memory (LSTM) also holds promise for achieving better results.

REFERENCES

[1] *Premier League Live Scores, Stats & Blog | 2023/24 — premierleague.com*, https://www.premierleague.com/matchweek/12287/blog, [Accessed 26-12-2023].

[2] *Football Statistics | Football Live Scores | WhoScored.com — 1xbet.whoscored.com*, https://1xbet.whoscored.com/, [Accessed 26-12-2023].

[3] L. Schiefler, *Football Club Elo Ratings — clubelo.com*, http://clubelo.com/, [Accessed 26-12-2023].

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] J. Brownlee, *How to Calibrate Probabilities for Imbalanced Classification - MachineLearningMastery.com — machinelearningmastery.com*, https://machinelearningmastery.com/probability-calibration-for-imbalanced-classification/, [Accessed 26-12-2023].

[6] *1.16. Probability calibration — scikit-learn.org*, https://scikit-learn.org/stable/modules/calibration.html, [Accessed 26-12-2023].