Predicting Soccer Result using Dixon Coles Model and its applications

By

WONG ZHEN PING

A project report submitted in partial fulfilment of the requirements for the award of Master of Mathematics

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

January 2021

DECLARATION OF ORIGINALITY

I hereby declare that this project report entitled "**Predicting Soccer Result using Dixon Coles Model and its applications**" is my own work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : Zhen Ping

Name : Wong Zhen Ping

ID No. : 1900585

Date : 5/4/2021

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr Denis Wong Chee Keong, for his guidance throughout the project. His expertise, ideas, feedback, time and encouragement helps a lot.

I am grateful for the opportunity given, to let me do research on this topic.

In addition, I thank my family for supporting me and helping me survive all the stress and not letting me give up.

WONG ZHEN PING

Predicting Soccer Result using Dixon Coles Model and its applications

WONG ZHEN PING

ABSTRACT

The Dixon Coles model uses attack and defend parameters from the team's perspective to predict a soccer result. This approach can be improved by adding the player's rating parameter. This study aims to improve the Dixon Coles model by implementing attack and defend parameters at the player's level. In this context, the player's attack and defending parameter are calculated from the player's rating data, coming from *whoscored.com* (n.d.).

To test the hypothesis that the player's parameter will improve the model, we collect the player's rating data and calculate the player's attack and defending parameter. The parameters were then added to the Dixon Coles model. We predicted season-long matches using the Dixon Coles model (before and after adding player's parameter), and the result from both models was compared. Overall, the results showed an improvement: the player's parameter did improve the original Dixon Coles model prediction. The underlying statistical distribution for the Dixon Coles model is Poisson distribution, its application is extensive, as long as the expecting event is statistically independent and the rate of happening is constant, such as packet loss per hour in networking field, number of customer arrival.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

Soccer is one of the world's most popular sport. It is a competitive sport between two playing teams, and the outcome is decided which team is scoring more goals during the gameplay. There are competitions between countries like FIFA World Cup, Olympics, held once every four years at the international level. At the same time, there are also some international competitions between football clubs held every year, such as the UEFA champions league. Furthermore, at the domestic level, one of the most popular competition is England Premier League, which winner is determined by the team who have the most points collected after season-long matches.

Commonly, the audience will predict the outcome of a match. The home team will win, the away team will win, or it will be a draw. Instead of guessing by sense, M. Dixon (1997) suggested that implementing a statistical model is possible for predicting the probability of an outcome of soccer matches. The Dixon Coles model uses past year data as their parameters, which might lead to inaccuracy. Some of the few reasons are such as the critical player left the team, the team's average age is increasing, and so on. Such factors were not considered in the Dixon Coles model.

In our work, we suggest adding in a player rating from the recent performances to enhance the current Dixon Coles model. The player rating is getting from a source such as *whoscored.com* (n.d.). Their player ratings are based on each event recorded in the game using their algorithms. Every event of importance was taken into account, with either positive or negative effects on ratings weighted to its pitch area and outcome. For example, an attempted dribble (event) in the opposition's final third (area of pitch) that is successful (outcome) will have a positive effect on a player's rating.

1-1 Objective

This project's main objective is to enhance the Dixon Coles model's accuracy by considering additional factors such as player rating from recent performances. The current Dixon Coles model uses historical data, team's past performance to perform prediction. Each player's recent performance can be considered to build the model, as a player's performance can directly affect the team's performance. A more precise estimation shall be achieved by considering the player's recent performance as an additional parameter. The primary source of our player rating data are collected from *whoscored.com* (n.d.). We rely on *whoscored.com* (n.d.) on the player rating data. They computed each player's rating on each match by considering every event recorded in the games.

1-2 Problem Statement

The current Dixon Coles model uses historical score data to predict based on the whole team's past performance. Each player's recent performance can be considered to build the model, as a player's performance can directly affect the team's performance. A more precise estimation should be achieved by considering the player's performance rating.

There will be two transfer windows in the major soccer league for each season, where a football club can transfer their players to the other football club. Some key players may be leaving the football club, which ultimately will impact its performance. Hence, it is crucial to evaluate the soccer match from the team's perspective and the player's perspective.

Another scenario is that the player's performance is not consistent all the time, might be fluctuating due to the team's morale, fitness, fatigue, or emotional reasons. By taking the individual's rating from the past match, we can evaluate the player's recent form and providing more accurate insight into the prediction.

1-3 Methodology and Planning

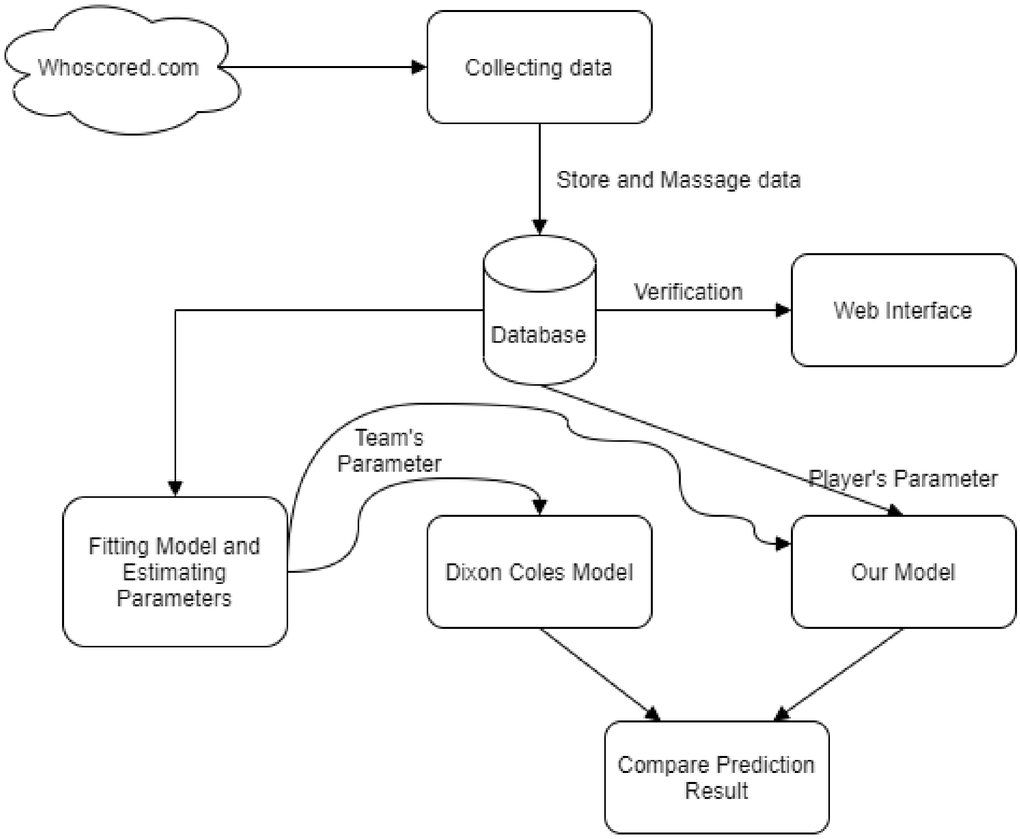
To begin with, we focus on running our model with data from English Premier League. Some data are needed, such as the past result and player's rating for every match. For the past result and player's rating, we are collecting from *whoscored.com* (n.d.).

We are not going to key in the data manually as there are too many matches (380 matches for a single season). To collect past results and player's rating data, we are using the C# environment, as it has a faster performance by nature (compare to language like python) due to its a compiled language. While the data will be store in a relational database rather than storing in a flat file, it will be a more structured way. We prepared a web interface for us to validate our data collected. We randomly picked numerous matches and compared the data we collected against data tabulated in the *whoscored.com* (n.d.) website.

After we got the raw data prepared in the database and verified it, we proceed to estimate the parameters needed for the Dixon Coles model. This parameters estimation was estimated in R, using the past result collected in the database. We used the Generalized Linear Models method to fit the model and estimate parameters.

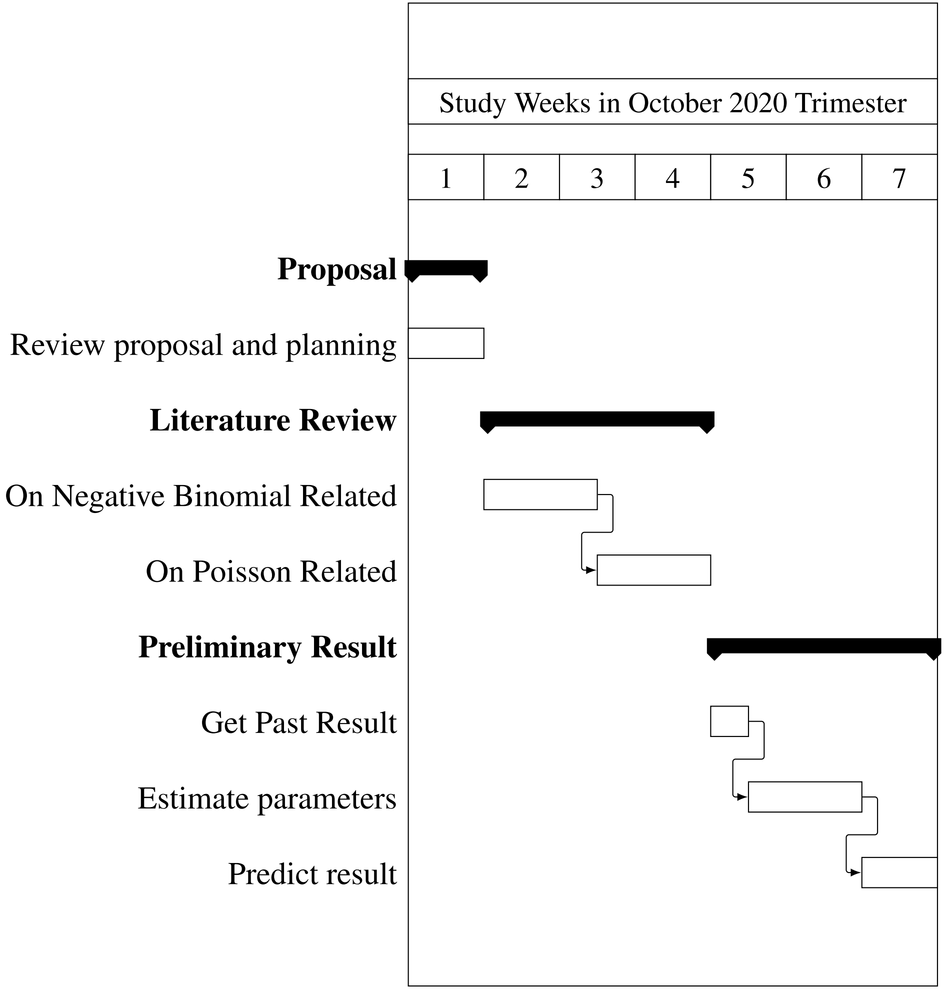
We reconstruct two Dixon Coles models using R, one for the original model and another one is implementing the player's rating. We prefer R over C# here is that R has ready-made mathematical packages available, which will be easier for us to perform mathematical computation. Prediction and simulation are performed in R too. While for data visualization, we are using the ggplot package in R.

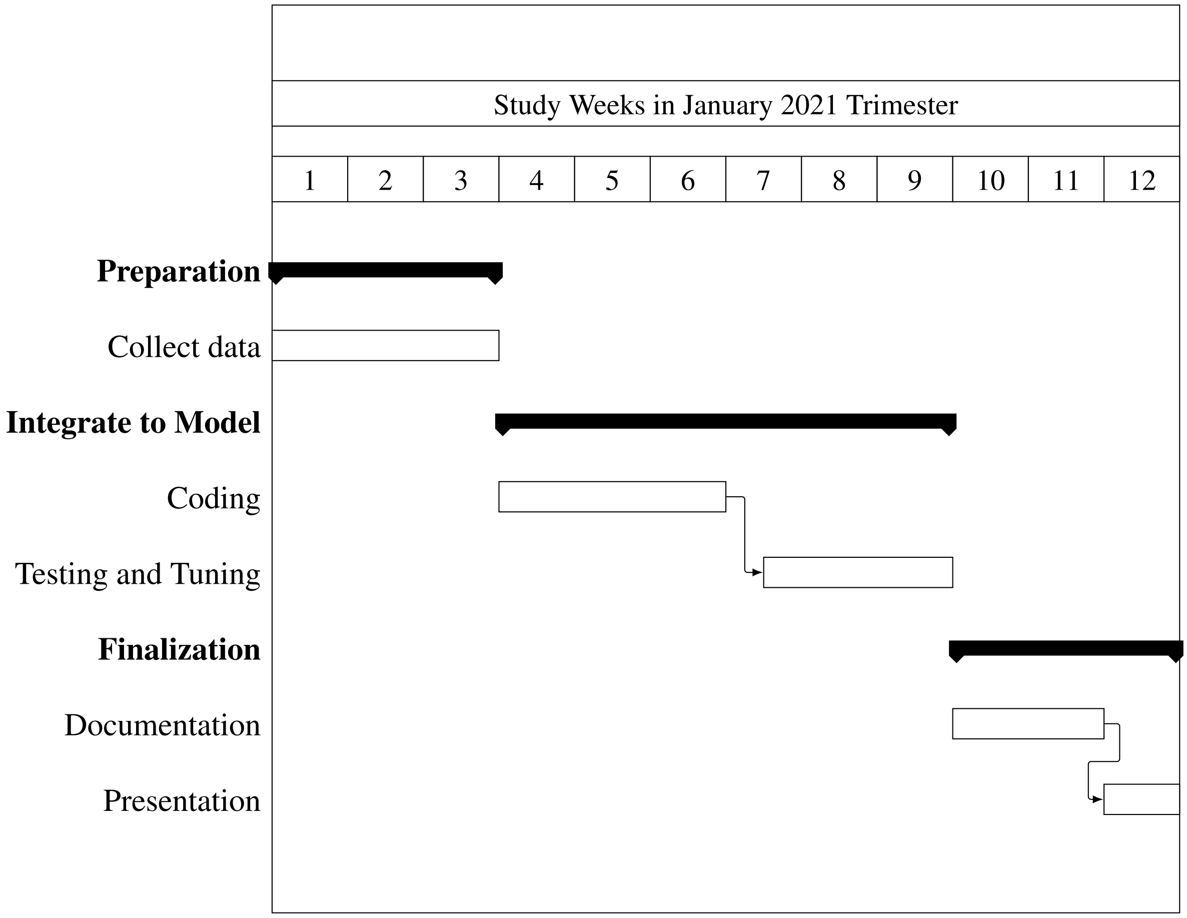
Figure 1.1: Illustration on project plan



1-4 Project Scope

For our scope, we used English Premier League data for 2017/2018 as input data to build and generate the model and will be using 2018/2019 season data for backtesting.




Study Weeks in January 2021 Trimester
1
2
3
4
5
6
7
8
9
10
11
12
Preparation
Collect data
Integrate to Model
Coding
Testing and Tuning
Finalization
Documentation
Presentation

CHAPTER 2: LITERATURE REVIEW

Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval. With a property that these events occur with constant mean and variance, it is also independent of the last occurrence.

The negative binomial distribution is a discrete probability distribution that models the number of successes in a sequence of independent and identically distributed trials before a specified number of failures occurs. The negative binomial distribution is commonly used as an alternative to overdispersed Poisson distribution. Both distributions are commonly used distribution in analyzing and modeling goal-scoring events in soccer.

Early work of Moroney (1956) found that the number of goals scored per team per match at a football match can be describe using Poisson distribution. However, after doing the chi-square test, it indicates that the fit is not as good as expected. Hence he modified the Poisson distribution, which allows the variability. Note that in Poisson, mean equal variance. The mentioned modification is to calculate variance based on the observed distribution. The Chi-square test shows that this modification gives a better overall fit than the ordinary Poisson distribution. This situation of sample variance exceeds the sample mean is an overdispersion of Poisson, and hence adding a new variance parameter to the ordinary Poisson model is equivalent to using a negative binomial distribution.

In 1968, C.Reep (1968) looked into passing moves and found that they fit well with the negative binomial distribution. In football, the ball is passed from player to player until a particular player loses possession of the ball either by interception, tackle by a member of defending team, infringement of rules or the attacking team is shooting at the defending side. The number of passes is defined as "r-pass movement" in their work, and the player's relative skills are essential in having higher r successful passes. Furthermore, they divide the pitch laterally into shooting areas and own half, analyze the goals scored in terms of how the passes are played. For example, the ratio of the goals where pass play is originating from shooting areas to all goals, ratio of shooting area origin attacks to all attacks reaching the shooting area, the average number of shots to score one goal, and so on. They were convinced that chance does dominate the game. More shots will have a higher chance of a scoring goal and hence winning the game. However, this does not mean that with an excess of shots by one team, one team will get more goals and win the match C. Reep (1971) did a follow-up on this, further applying the negative binomial distribution to specific movements or performances in other ball games. They tested the applications to cricket, ice hockey, baseball, and lawn tennis. However, they obtained poor fits result in a situation where individual skills play a more substantial role.

Maher (1982) thinks that a team's number of goals is likely to be a Poisson variable. Each time a team has the ball, it has the opportunity to attack and score. The probability p that an attack will result in a goal is small, but the number of times a team has possession during a match is large. If p is constant and attacks are independent, the number of goals will be Binomial, and in these circumstances, the Poisson approximation will apply very well. The mean of this Poisson will vary according to the team's quality, so if one were to consider the distribution of goals scored by all teams, one would have a Poisson distribution with variable means.

Therefore,Maher (1982) adopted Poisson model with teams' attacking and defensive strengths parameter. Assuming team i is playing at home against team j, with the result is $(X_{i,j}, Y_{j,i})$, we shall assume the $X_{i,j}$ is Poisson with mean $\alpha_i\beta_j$, and $y_{i,j}$ is also Poisson with mean $\gamma_i\delta_j$, and $X_{i,j}$ and $Y_{i,j}$ are independent. We can understand this as α_i representing attacking strength of team i when playing at home and β_j is the defensive strength of team j when playing away, while γ_i is defensive of team i at home and δ_j is the attacking strength when playing away.

Maher then determined the MLEs for the four proposed parameters and further reduced them to 2 parameters, $\alpha_i\beta_j$. These two parameters are significant and sufficient to describe the quality of team i's attack and team j's defense, whether playing at home or away. Although the home ground advantage is a significant factor, it has an equal effect on all the team, and each team's scoring ability is diminished by a constant factor when playing away. However, Maher's model has underestimated the number of goals at one and two goals scored and overestimate the number of $\geq$4 goals and 0 goals. Maher improved the model by using a bivariate Poisson model to model the dependence between scores.

In Keller (1994) work, he stated that suppose two opponents P and Q play a game in which P scores n points with probability p_n, and independently Q scores m points with probability q_m. Suppose that the opponent with the higher score wins and that p_n is Poisson distribution with mean λ, then for any distribution q_n,

$$\frac{\partial}{\partial\lambda}Pr[P \quad beats \quad Q] = Pr[P \quad ties \quad Q]$$

$P(\lambda)$ is a Poisson, and we assume $Q(\mu)$ is Poisson distributed too, so this characterization Keller stated allow us to compute the theoretical probability for all the outcome $(P\ wins, Q\ wins\ or\ ties)$. He has shown that the standard deviation of Expected wins & ties and Actual wins & ties are reasonably slight in his work, indicating that the model is a proper fit to the data. This characterization is essential because it shows that it can predict the game outcome with the Poisson distribution.

The home ground advantage is known as a factor contributing to the soccer match result. S. Clarke (1995) made the point that home advantage is different for each individual club, and the magnitude of advantage is linearly related to the distance between both playing clubs. The magnitude of home ground advantage is computed by the past match's winning margin using least squares. Derivation of the formula is written clearly in their work. There is some discussion why home ground advantage, such as different pitch types (small/large pitch, artificial turf etc.), may cause higher home ground advantage. For example, players usually training in their club's home ground, so they tend to be more familiar with the pitch they're trained with. So, when they are playing in another club's ground, they tend to be unfamiliar with the pitch, and on the other hand, the home team (opponent) player gets an advantage here. However, this is just one of the factors contributing to the home ground advantage. The atmosphere of crowd cheering will be one of the factors also. Do take note that this advantage might not be applicable in 2020 due to the COVID-19 pandemic, as most of the major leagues are playing with an empty stadium. In other words, there will be no fans cheering.

M. Dixon (1997) did further work on Maher (1982) model. Their research found that the model is underestimating the probability for lower score games, such as $0-0, 1-0, 0-1, 1-1$. This is shown when they compare results from Maher's model and empirical estimates for their collected data. To overcome this issue, they add an additional parameter. Do note that this additional parameter, other than inflate probabilities for lower score games, may also correct probabilities of another score outcome, in other words, out of those underestimated outcomes.

D. Karlis (2003) proposed an alternative model, instead of adding a new parameter to tackle the underestimation problem, they modified the bivariate Poisson by inflating only the diagonal probability, in other words, the probability of a draw. Such a model requires the marginal distribution to be a discrete distribution, for example, Poisson, Geometric or Bernoulli distribution.

In M. Dixon (1997) model, they take home ground advantage into consideration. An extension of the bivariate Poisson model by Maher (1982). However, some enhancement has been brought up in regards to their model.

1. The attack and defense parameters are static, but in reality, a team's performance will be dynamic from time to time

2. Should consider time weighting function when calculating the attack and defense parameters. In the sense that the more recent rating will be more influential comparing to the rating from older days.

M. Dixon (1998) did extend the work on M. Dixon (1997), which takes consideration of time remaining to play and current score. They treat the number of goals scored as interacting birth processes. Based on the paper, the scoring rates increase gradually throughout 0 to 90 minutes of gameplay. This could be due to tiredness (and hence prone to mistake in defending side). The current score's influence is more significant when the home team has a narrow lead, the home and away scoring rates decrease and increase significantly, respectively.

Due to the nature of the birth process, this model needs to calculate probabilities of being in each state throughout 90 minutes of gameplay. In other words, all the score possibilities
$(x, y) : x, y = 0, 1, \ldots .)$, integrating over all possible times and for each possible route to arrive at the point (x, y). Authors use Monte Carlo simulation for each match to simulate the goal process. After getting the probability of all score outcomes, we can summarize the probabilities into Home Win, Draw, or Away Win.

With the extra parameters added into this goal-scoring process, the estimation is improved compare to Maher (1982) and M. Dixon (1997).

R. Pollard (1997) by using a notational system, records the events taking place throughout the whole soccer match to assess the effectiveness of playing strategies, and a quantitative variable is developed representing the probability of a goal being scored, minus the probability of one being conceded.

Table 2.1: Comparison of values assigned to different outcome variables

Outcome of	**Outcome Variable**				
team possession	**Goal**	**Shot**	**Weighted shot**	**Preliminary yield**	**Yield**
Shot: goal	1	1	p	p	p
Shot: not a goal	0	1	p	p	p
Possession regained	0	0	0	$p_{i,j}$	$y_{i,j}$
Possession lost	0	0	0	$-p_{i,j}$	$-y_{i,j}$

p is the estimated scoring probability of a shot, $p_{i,j}$ is the estimated probability of scoring from possession orignating as type j in zone i and $y_{i,j}$ is the estimated yield from possession originating as type j in zone i.

From Table 2.2, we can see that for 1000 team possessions, which moves/strategy will be getting a higher yield. Note that the negative in yield indicates that the strategy, on average, would result in more goals being conceded than scored. This, however, should not be taken as since the yield is negative, the team should not execute it. This work by R. Pollard (1997) did not involve any statistical distribution, but this has shown us that the playing strategy can be quantified, and we can further apply this in our research.

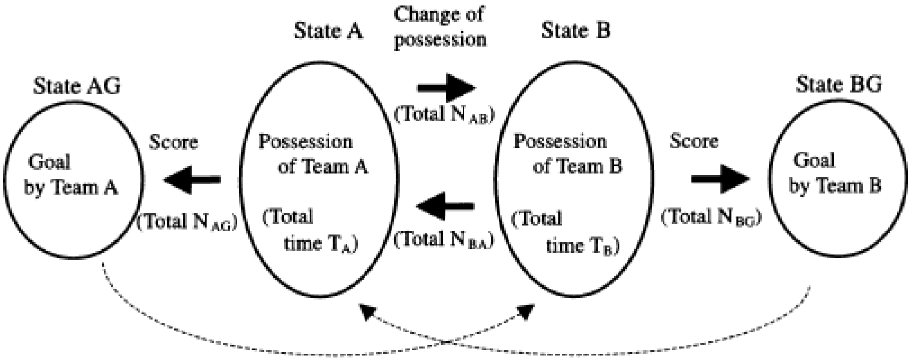
Nobuyoshi Hirotsu (2003) described a Markov process model of a football match. This is a further extension on M. Dixon (1998), which they add in the factors of rates of gaining and losing possession into the model. They pictured a football match as progressing through a set of stochastic transitions in 4 states. as shown in Figure 2.1. And, of course, assumed Markov property (memoryless).

Ian G. McHale (2014) introduce a model that takes a player's goal conversion ability into account. Instead of looking at the team's attacking parameter, Ian G. McHale (2014) focus on modeling goals scored by an individual. Their work uses a goals-per-minute ratio to measure the player's ability to score goals. They separate the process

Table 2.2: Yield per 1000 team possessions from playing strategies in different situations

Situation	**Strategy**	n	**Yield**
Goal kick	Long	99	-2.7
Throw-in in own half	Short	276	-0.2
Possession in zone 4	Short passing only	1372	11.1
	Running with the ball	288	16.3
	Long forward pass	148	23.1
Free kick in zone 5	Direct shot	60	12.5
	Other	143	16.8
Throw-in in zone 6	Short	98	3.5
	Long towards goalmouth	32	21.7
Centres from zone 6	Above waist height	240	33.3
	Below waist height	103	96.6

Figure 2.1: 4 states stochastic transitions



of scoring goals into two parts, creating shots and converting shots into goals. Furthermore, they have two versions of the model to compare, a basic model which covariates carry only team-related information (e.g., team attacking and defense parameter), while an extended model which covariates carries the player position and time spent on the pitch (for shot count model) and (conversion model).

Table 2.3 shows the summary of models and approaches used by researchers. While Figure 2.2 shows how the papers are related and arranged in chronological order—they are mainly divided into three approaches, Negative Binomial, Poisson, and Stochastic Processes.

In the Negative Binomial approach, C.Reep (1968) categorize events happening in the pitch, categorizing the events according to the number of passes that occurred, and the count of occurrences was used as the parameter for Negative Binomial. C. Reep (1971) extend the result to other ball games.

In the Poisson approach, researchers estimate the attack and defend parameters for Poisson to generate the score matrix. Keller (1994) confirms that Poisson is a suitable model for it. M. Dixon (1997) and D. Karlis (2003) did enhancement on Maher (1982) work, to tackle its underestimation for lower score games scenario. Also need to highlight that Ian G. McHale (2014) take player's ability to convert shot to goal into consideration, as an extra set of parameters.

While for the Stochastic Process group, M. Dixon (1998) consider time remaining in play and current score, treat the number of goals as interacting birth processes. Moreover, Nobuyoshi Hirotsu (2003) describe the football match as a 4-states stochastic transition Markov process.

Besides, S. Clarke (1995) found that Home advantage is a factor contributing to the results, R. Pollard (1997) analyzed the football match by quantifying the moves (events) happening on the pitch, and by using the quantified variables, they calculate the yield.

Figure 2.2: Research Flowchart, chronologically

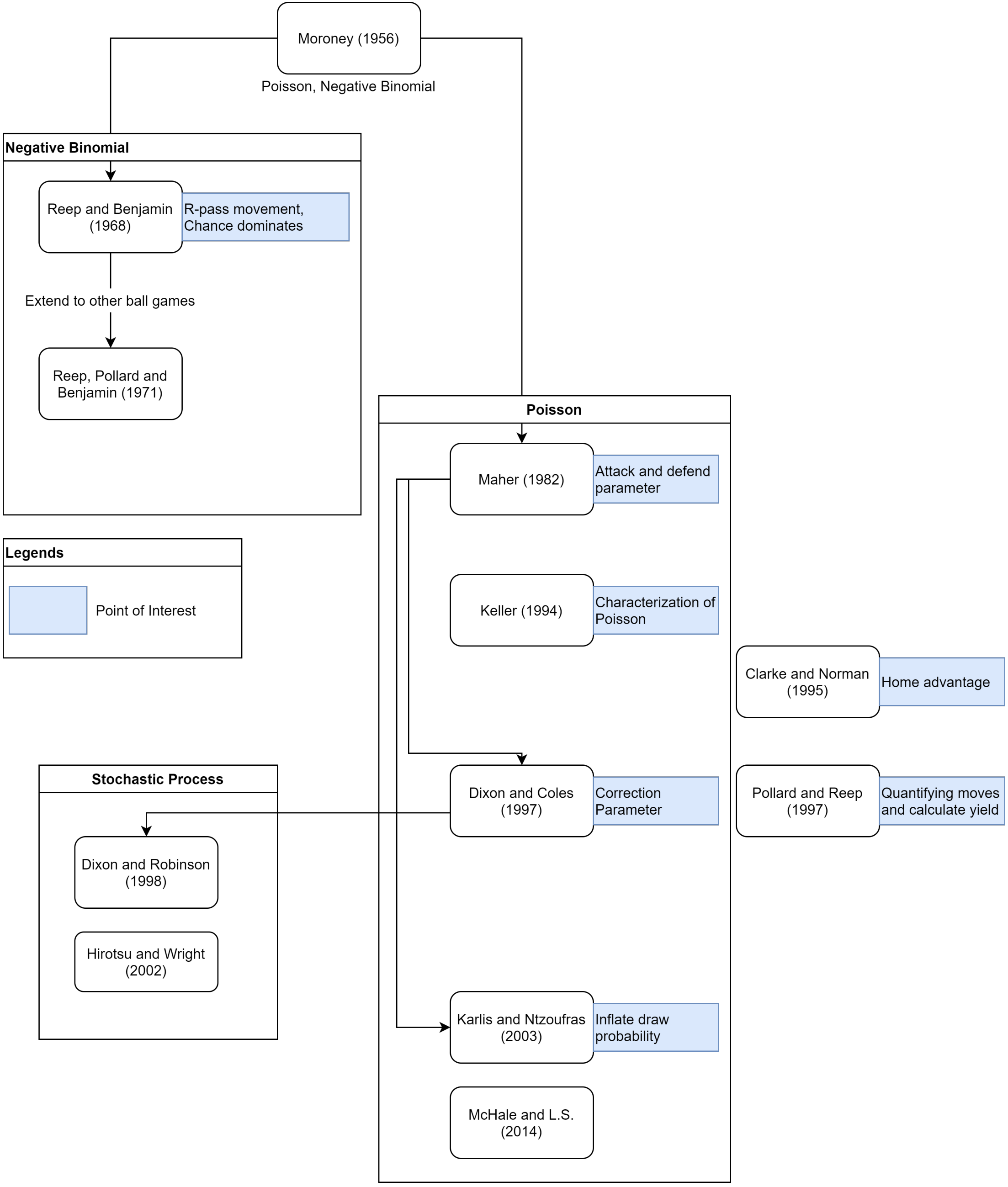

Moroney (1956)
Poisson, Negative Binomial
Negative Binomial
Reep and Benjamin (1968)
R-pass movement, Chance dominates
Extend to other ball games
Reep, Pollard and Benjamin (1971)
Legends
Point of Interest
Poisson
Maher (1982)
Attack and defend parameter
Keller (1994)
Characterization of Poisson
Clarke and Norman (1995)
Home advantage
Stochastic Process
Dixon and Coles (1997)
Correction Parameter
Pollard and Reep (1997)
Quantifying moves and calculate yield
Dixon and Robinson (1998)
Hirotsu and Wright (2002)
Karlis and Ntzoufras (2003)
Inflate draw probability
McHale and L.S. (2014)

Table 2.3: Table of Summary

Author	**Year**	**Model**	**Point of Interest**
Moroney	1956	Poisson, Negative Binomial	No
Reep and Benjamin	1968	Negative Binomial	R-pass movement, Chance dominates
Reep, Pollard and Benjamin	1971	Negative Binomial	Extend C.Reep (1968) to other ball games
Maher	1982	Poisson	Attack and defence parameter
Keller	1994	Poisson	Characterization of Poisson
Clarke and Norman	1985		Home advantage
Dixon and Coles	1997	Poisson	Correction Parameter
Karlis and Ntzoufras	2003	Poisson	Inflate only diagonal probability
Dixon and Robinson	1998	Poisson, Birth Process	Consider time remaining and current score line
Pollard and Reep	1997		Quantify the moves/ strategy and calculate yield
Hirotsu and Wright	2002	Markov Process	4 states stochastic transitions
McHale and L.S.	2014	Poisson	Player’s goal conversion ability

CHAPTER 3: PRELIMINARY RESULT

For this project, we will be focusing on English Premier League 2017/2018 as our training dataset and season 2018/2019 as our testing dataset.

A brief introduction to the model that we will use before we step into the actual code, Maher (1982) proposed a bivariate Poisson model.

$$P\big(X_{i,j}=x, Y_{j,i}=y\big)=\frac{e^{-\lambda}\lambda^{x}}{x!}\frac{e^{-\mu}\mu^{y}}{y!}$$
$$where \quad \lambda=\alpha_i\beta_j\gamma \quad , \quad \mu=\alpha_j\beta_j$$

In this model, i and j refer to home and away teams, respectively, while α denotes the team's attack, β denotes the team's defensive strength, and γ represents the home advantage factor. M. Dixon (1997) determines that the model is underestimating the probability of low scoring games $(0-0, 1-0, 0-1, 1-1)$. This is shown when comparing results from the Maher (1982) model and empirical estimate for data between 1992 and 1995.

The data from 1992 to 1995 provide accurate empirical estimates of various aggregated features. Table 3.1 gives the relative frequency, expressed as a percentage of the scores from $0-0$ to $4-4$

	Away	**0**	**1**	**2**	**3**	**4**
Home		33.4	36.4	19.5	7.9	2.1
0	22.1	8.2	7.4	4.5	1.4	0.4
1	33.0	10.3	12.7	6.4	2.7	0.6
2	24.5	8.2	9.1	4.8	1.9	0.5
3	12.6	4.2	4.5	2.3	1.2	0.4
4	5.3	1.6	1.8	1.1	0.6	0.1

Table 3.1: Empirical estimates of each score probability(%) for joint and marginal probability functions

This can be examined by fitting a Poisson distribution to the aggregated home and away scores in Table 3.1, which reveals that by any criterion, the Poisson model is a near-perfect fit to the aggregated score data. A further assumption of the basic model is that the home and away scores are independent. Table 3.2 will show that this is a valid assumption. It displays

$$\frac{f(i,j)}{f_h(i)f_a(j)}$$

for each home and away score $(i, j), i = 0, ..., 6andj = 0, ..., 5$ where f, f_h, f_a are the joint and marginal empirical probability function for home and away scores respectively.

		Away					
		0	**1**	**2**	**3**	**4**	**5**
	0	111.5	92.0	103.4	82.1	96.4	96.8
	1	93.7	105.7	99.3	103.7	86.9	108.3
	2	99.6	101.7	99.2	97.4	95.9	106.7
Home	**3**	100.3	98.5	91.8	116.6	139.8	75.4
	4	91.0	93.8	108.6	138.0	111.7	90.4
	5	94.1	102.3	114.3	73.3	120.8	130.4
	6	139.1	49.1	146.4	45.3	174.1	-

Table 3.2: Estimates of the ratios of the observed joint probability function and the empirical probability function obtained under the assumption of independence between the home and away scores (figures are multiplied by 100)

This table shows that the assumption of independence between scores is reasonable. The small ratios prove that.

And hence M. Dixon (1997) proposed a modification to Maher (1982) model.

$$P(X_{i,j} = x, Y_{j,i} = y) = \tau_{\lambda,\mu}(x, y)\frac{e^{-\lambda}\lambda^x}{x!}\frac{e^{-\mu}\mu^y}{y!}$$

$$where\lambda = \alpha_i\beta_j\gamma \quad \mu = \alpha_j\beta_i$$

$$and \quad \tau_{\lambda,\mu}(x, y) = \begin{cases} 1 - \lambda\mu\rho, & if \quad x = y = 0 \\ 1 + \lambda\rho, & if \quad x = 0, y = 1 \\ 1 + \mu\rho, & if \quad x = 1, y = 0 \\ 1 - \rho, & if \quad x = y = 1 \\ 1, & otherwise \end{cases}$$

$$\rho, \quad where \quad max(-1/\lambda, -1/\mu) \leq \rho \leq min(1/\lambda\mu, 1)$$

The τ function depends on the ρ parameter, which is considered the strength of correction. This model will be translated to R for further work.

Based on our summary result, estimated attack, defend, home advatage and correction factor for season 2017/2018 is as below:

Home advantage: 0.29

Correction factor (ρ) : -0.13

Team	**Attack**	**Defense**
Arsenal	0.45	-0.06
Bournemouth	-0.04	-0.21
Brighton	-0.32	-0.07
Burnley	-0.30	0.26
Chelsea	0.26	0.25
Crystal Palace	-0.05	-0.12
Everton	-0.06	-0.16
Huddersfield	-0.51	-0.14
Leicester	0.19	-0.21
Liverpool	0.56	0.21
Man City	0.79	0.55
Man United	0.33	0.55
Newcastle	-0.23	0.07
Southampton	-0.24	-0.12
Stoke	-0.28	-0.31
Swansea	-0.53	-0.10
Tottenham	0.43	0.30
Watford	-0.07	-0.26
West Brom	-0.42	-0.10
West Ham	0.04	-0.33

Table 3.3: Estimation result from R

After we estimated the parameters, we predict the score matrix for one match, Arsenal vs. Chelsea. An assumption made is that the max score is capped at 6. Computed probability in are tabulated in Table 3.4.

		Chelsea						
		0	**1**	**2**	**3**	**4**	**5**	**6**
Arsenal	**0**	5.8660866	4.9927	4.5285	2.1451	0.762	0.0217	0.00513
	1	6.1804072	12.1257	7.6343	3.6162	1.2847	0.0365	0.00865
	2	6.3735055	9.0569	6.4351	3.0481	1.0829	0.0308	0.00729
	3	3.5815409	5.0895	3.6161	1.7129	0.6085	0.0173	0.00410
	4	1.5094639	2.145	1.524	0.7219	0.2565	0.00729	0.00173
	5	0.5089388	0.7232	0.5139	0.2434	0.0865	0.00246	0.000582
	6	0.1429971	0.2032	0.1444	0.0684	0.0243	0.00069	0.000164

Table 3.4: Computed Probability (%) for Arsenal vs Chelsea in Score Matrix

We can further see that the winning outcome probability for this match is

Arsenal Win	0.4253858
Chelsea Win	0.3110412
Draw	0.263573

This is computed by summarizing the probability in the score matrix, in Table 3.4. Summation across the lower triangle, except diagonal entry, is Arsenal (Home) win. The upper triangle's summation, except diagonal entry, too, is Chelsea (Away) win. Moreover, finally, summation across the diagonal is the probability of Draw.

Chapter 4: Results and Discussion

As mentioned in the previous chapter, we will be using English Premier League 2017/2018 as our training dataset and season 2018/2019 as our testing dataset. These data will be getting from *whoscored.com* (n.d.). Considering there are 20 teams in the league, each team will be playing in their home ground against the other 19 teams to have $20 * 19 = 380$ matches for a single complete season. Since we are using data across two seasons, there are 760 matches in total.

Recording the scores and players rating manually for 22 players in a match and a total of 760 matches is a very tedious task and error-prone, so we decided to use a little bit of programming to get this data collection done. To have a clearer view on why we need to have this automation, we can refer to Figure 4.1. One single season has 380 matches, and each match has two sides of 11 starting players (effectively 22 starting players in a match), which is essentially $380 * 22 = 8360$, 8360 player's rating data needed to record for a single season, when we are looking at two seasons, it will be 16720 rows of data. A program is a better way to get this done.

We wrote a web application in C# enironment for this purpose. There are three main objectives for this web application to achieve,

1. Able to collect data
2. Store data into database
3. Populate data from database to user interface

Figure 4.2 illustrates the function and overall structure of our web application.

4-1 Web Application

To understand what is going on in our web application, there are three main components we need to obtain from *whoscored.com* (n.d.), the list of matches, the final result for each match, and the player's rating for each match.

Our application firstly browse to *whoscored.com* (n.d.), and download the list of matches, as shown in Figure 4.3. We repeat until we got all the matches we need.

Figure 4.1: Player's rating needed to record

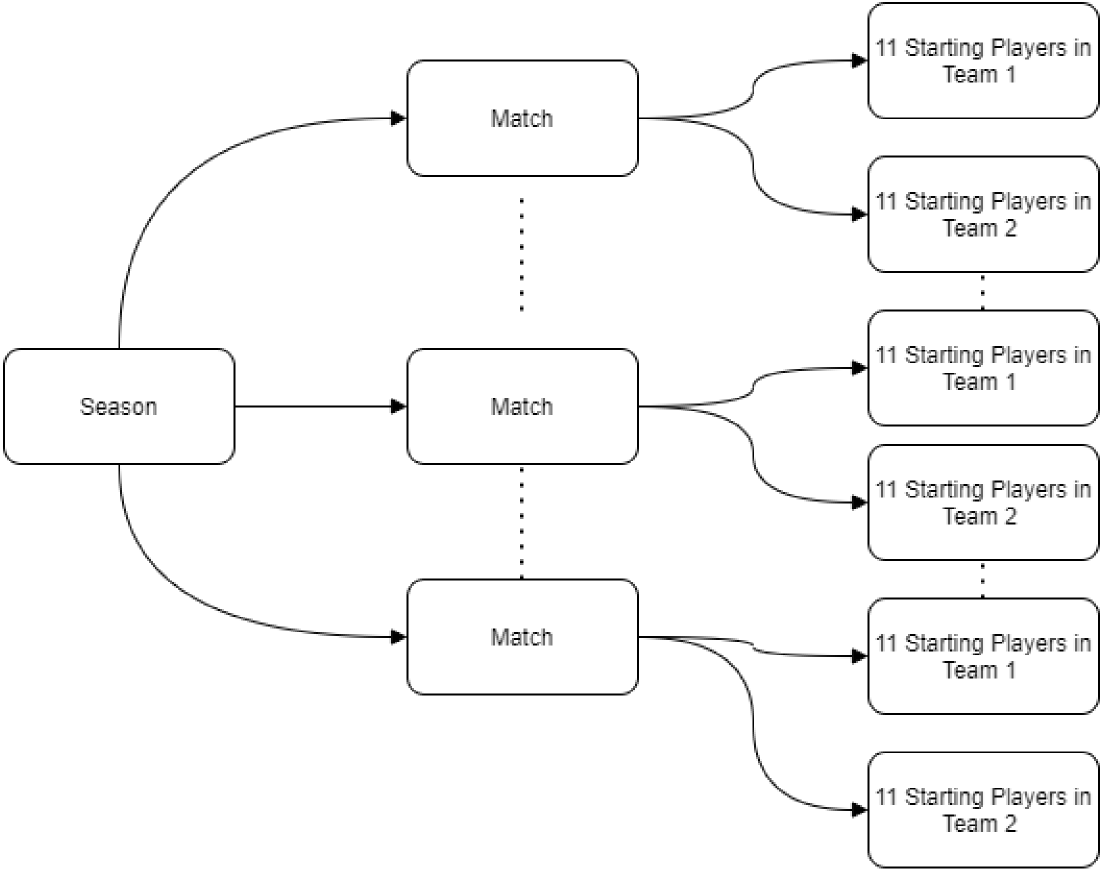
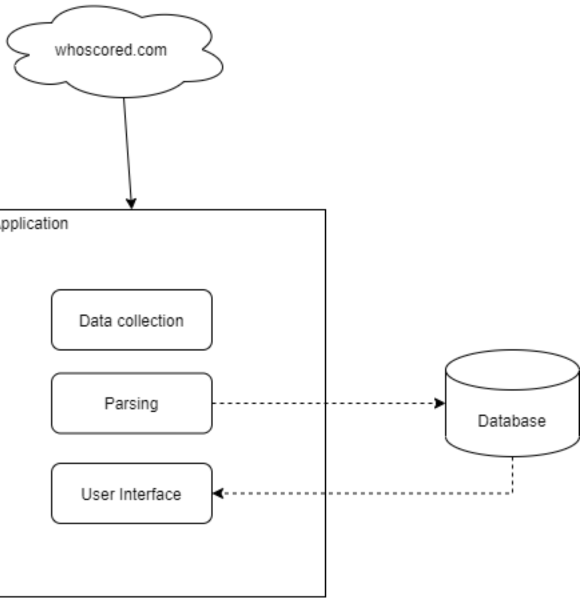
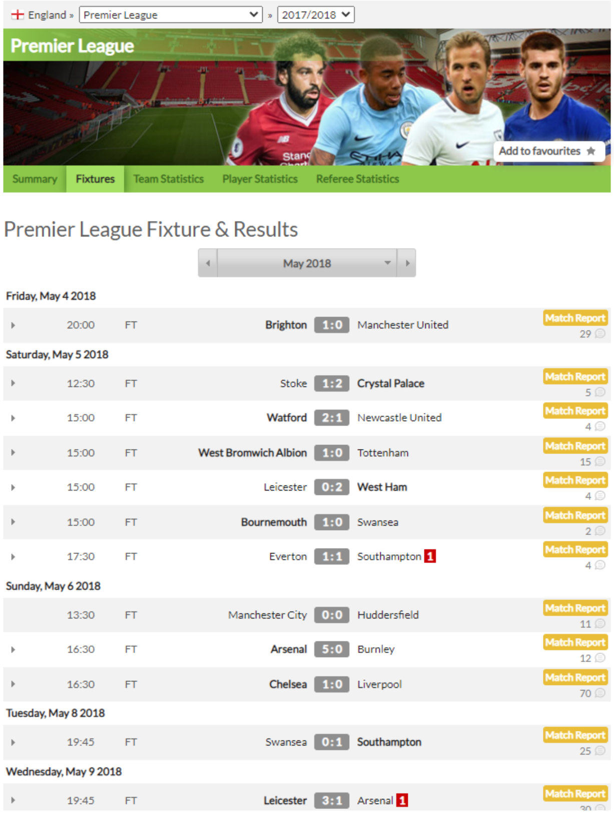

Season
Match
Match
Match
11 Starting Players in Team 1
11 Starting Players in Team 2
11 Starting Players in Team 1
11 Starting Players in Team 2
11 Starting Players in Team 1
11 Starting Players in Team 2

Figure 4.2: Illustrating web application



After that, we loop the match list and visit the Match Centre page, to download the final result and player's rating for each match, as shown in Figure 4.4.

Figure 4.3: List of matches on whoscored.com



Premier League Fixture & Results

May 2018

Friday, May 4 2018

20:00	FT	Brighton	1 : 0	Manchester United	Match Report	29

Saturday, May 5 2018

12:30	FT	Stoke	1 : 2	Crystal Palace	Match Report	5
15:00	FT	Watford	2 : 1	Newcastle United	Match Report	4
15:00	FT	West Bromwich Albion	1 : 0	Tottenham	Match Report	15
15:00	FT	Leicester	0 : 2	West Ham	Match Report	4
15:00	FT	Bournemouth	1 : 0	Swansea	Match Report	2
17:30	FT	Everton	1 : 1	Southampton 1	Match Report	4

Sunday, May 6 2018

13:30	FT	Manchester City	0 : 0	Huddersfield	Match Report	11
16:30	FT	Arsenal	5 : 0	Burnley	Match Report	12
16:30	FT	Chelsea	1 : 0	Liverpool	Match Report	70

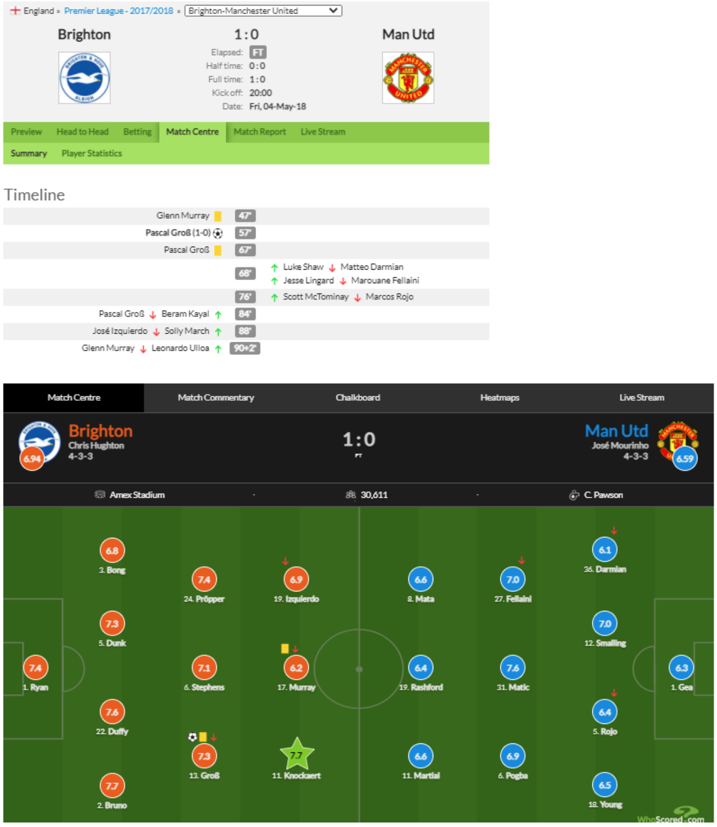
Tuesday, May 8 2018

19:45	FT	Swansea	0 : 1	Southampton	Match Report	25

Wednesday, May 9 2018

19:45	FT	Leicester	3 : 1	Arsenal 1	Match Report	30

Figure 4.4: Match Centre on whoscored.com



Brighton 1 : 0 Man Utd

Elapsed: FT
Half time: 0 : 0
Full time: 1 : 0
Kick off: 20:00
Date: Fri, 04-May-18

Timeline

Glenn Murray 🟨	47'	
Pascal Groß (1-0) ⚽	57'	
Pascal Groß 🟨	67'	
	68'	↑ Luke Shaw ↓ Matteo Darmian ↑ Jesse Lingard ↓ Marouane Fellaini
	76'	↑ Scott McTominay ↓ Marcos Rojo
Pascal Groß ↓ Beram Kayal ↑	84'	
José Izquierdo ↓ Solly March ↑	88'	
Glenn Murray ↓ Leonardo Ulloa ↑	90+2'	

Brighton
Chris Hughton
4-3-3

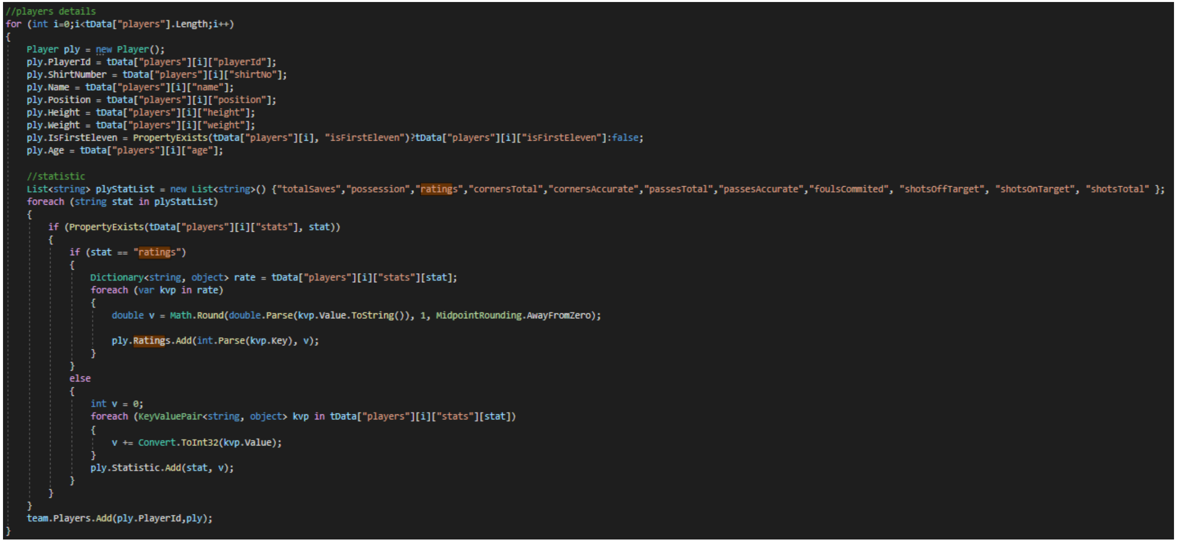
1 : 0
FT

Man Utd
José Mourinho
4-3-3

Amex Stadium · 30,611 · C. Pawson

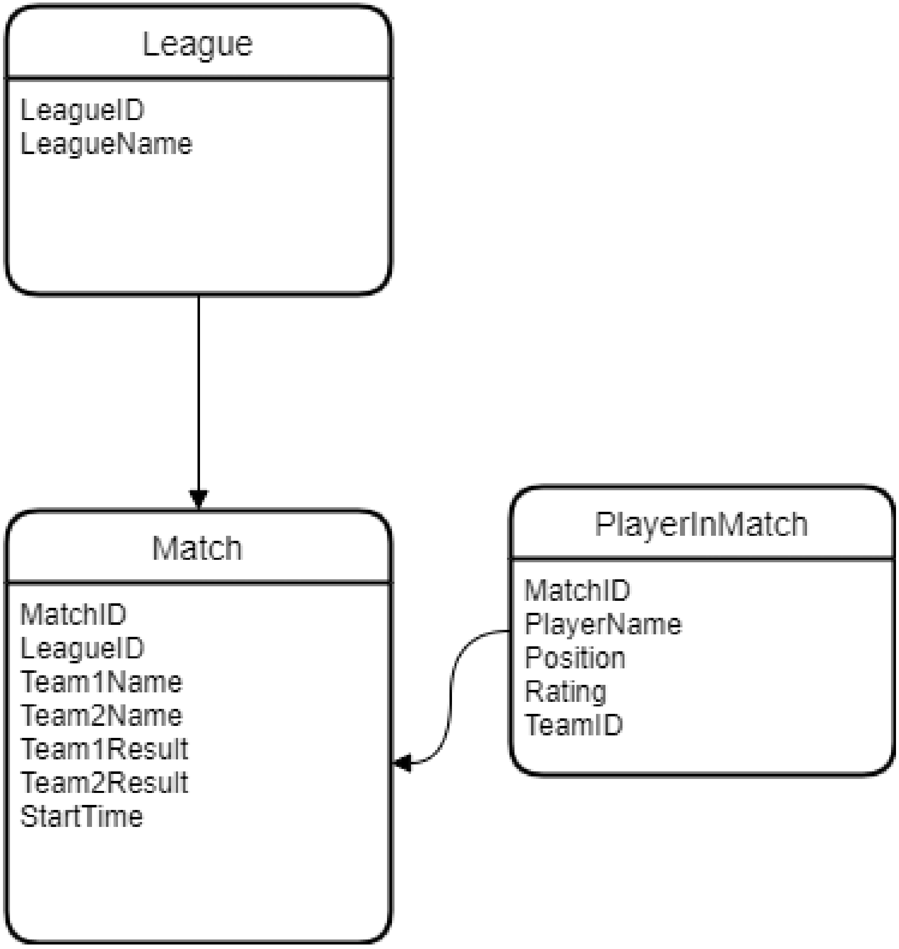
At this point, what we downloaded is raw data, which we need to clean and parse from the raw to get the info we needed. We make use of Regular Expressions and the Json.NET package in C# to do the parsing. A sample of how we perform the data parsing is shown in Figure 4.5. After we obtained the info we need, we store them in the database. The database schema that we are using is tabulated in Figure 4.6

Figure 4.5: Code on data parsing



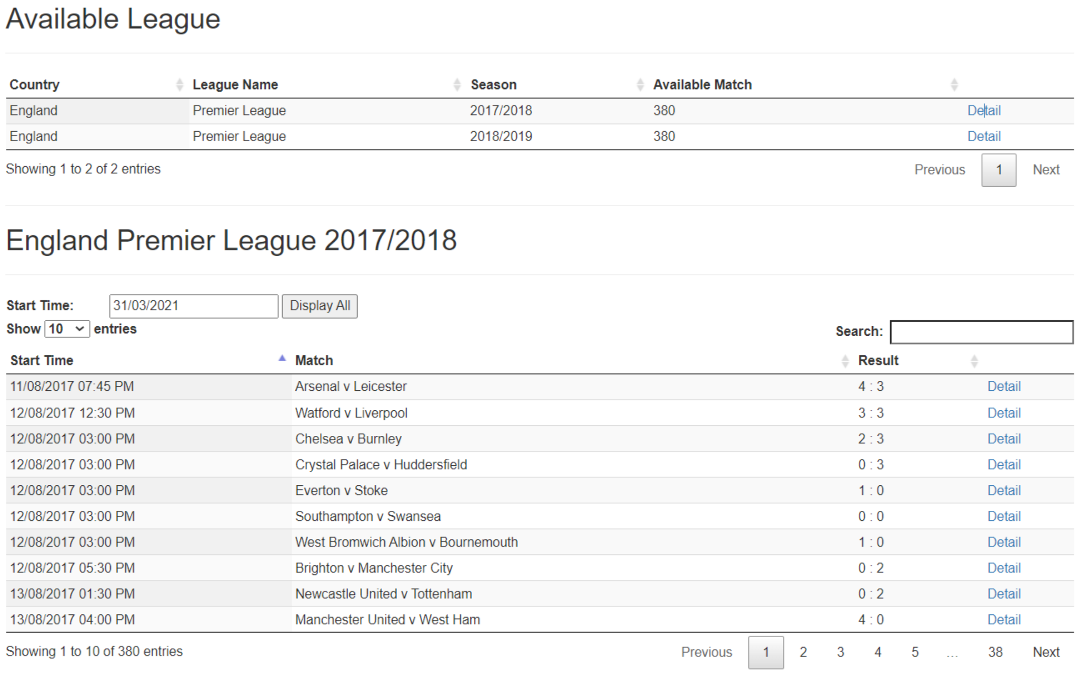
```
//players details
for (int i=0;i<tData["players"].Length;i++)
{
    Player ply = new Player();
    ply.PlayerId = tData["players"][i]["playerId"];
    ply.ShirtNumber = tData["players"][i]["shirtNo"];
    ply.Name = tData["players"][i]["name"];
    ply.Position = tData["players"][i]["position"];
    ply.Height = tData["players"][i]["height"];
    ply.Weight = tData["players"][i]["weight"];
    ply.IsFirstEleven = PropertyExists(tData["players"][i], "isFirstEleven")?tData["players"][i]["isFirstEleven"]:false;
    ply.Age = tData["players"][i]["age"];

    //statistic
    List<string> plyStatList = new List<string>() {"totalSaves","possession","ratings","cornersTotal","cornersAccurate","passesTotal","passesAccurate","foulsCommited", "shotsOffTarget", "shotsOnTarget", "shotsTotal" };
    foreach (string stat in plyStatList)
    {
        if (PropertyExists(tData["players"][i]["stats"], stat))
        {
            if (stat == "ratings")
            {
                Dictionary<string, object> rate = tData["players"][i]["stats"][stat];
                foreach (var kvp in rate)
                {
                    double v = Math.Round(double.Parse(kvp.Value.ToString()), 1, MidpointRounding.AwayFromZero);

                    ply.Ratings.Add(int.Parse(kvp.Key), v);
                }
            }
            else
            {
                int v = 0;
                foreach (KeyValuePair<string, object> kvp in tData["players"][i]["stats"][stat])
                {
                    v += Convert.ToInt32(kvp.Value);
                }
                ply.Statistic.Add(stat, v);
            }
        }
    }
    team.Players.Add(ply.PlayerId,ply);
}
```

Figure 4.6: Database schema to store our data



Next, we need to confirm that our data collected in the database is legitimate and accurate so that our modeling and prediction are meticulous. Hence we build a user interface to verify the data collected, and the user interface will read data from the database. Referring to Figure 4.7, the first table in the figure is showing the league we have. There are two entries, England Premier League 2017/2018 and England Premier League 2018/2019 and with 380 matches available for each entry. This is aligned with what we needed. Clicking on the Detail button on the rightmost column, we will display the matches in the selected league and season. The result is showing in the second table in Figure 4.7. We have the Starting Time of the match, the playing teams in the match, and the final result. When we further navigate, clicking on the Detail button on any row, we can see the details in the match. As shown in Figure 4.8, the data showing in our User Interface is the same as the data displaying in *whoscored.com* (n.d.). This shows that our web application is working as intended, data is stored correctly.

Figure 4.7: Web application display of events we got from whoscored.com



Available League

Country	League Name	Season	Available Match	
England	Premier League	2017/2018	380	Detail
England	Premier League	2018/2019	380	Detail

Showing 1 to 2 of 2 entries

England Premier League 2017/2018

Start Time: 31/03/2021

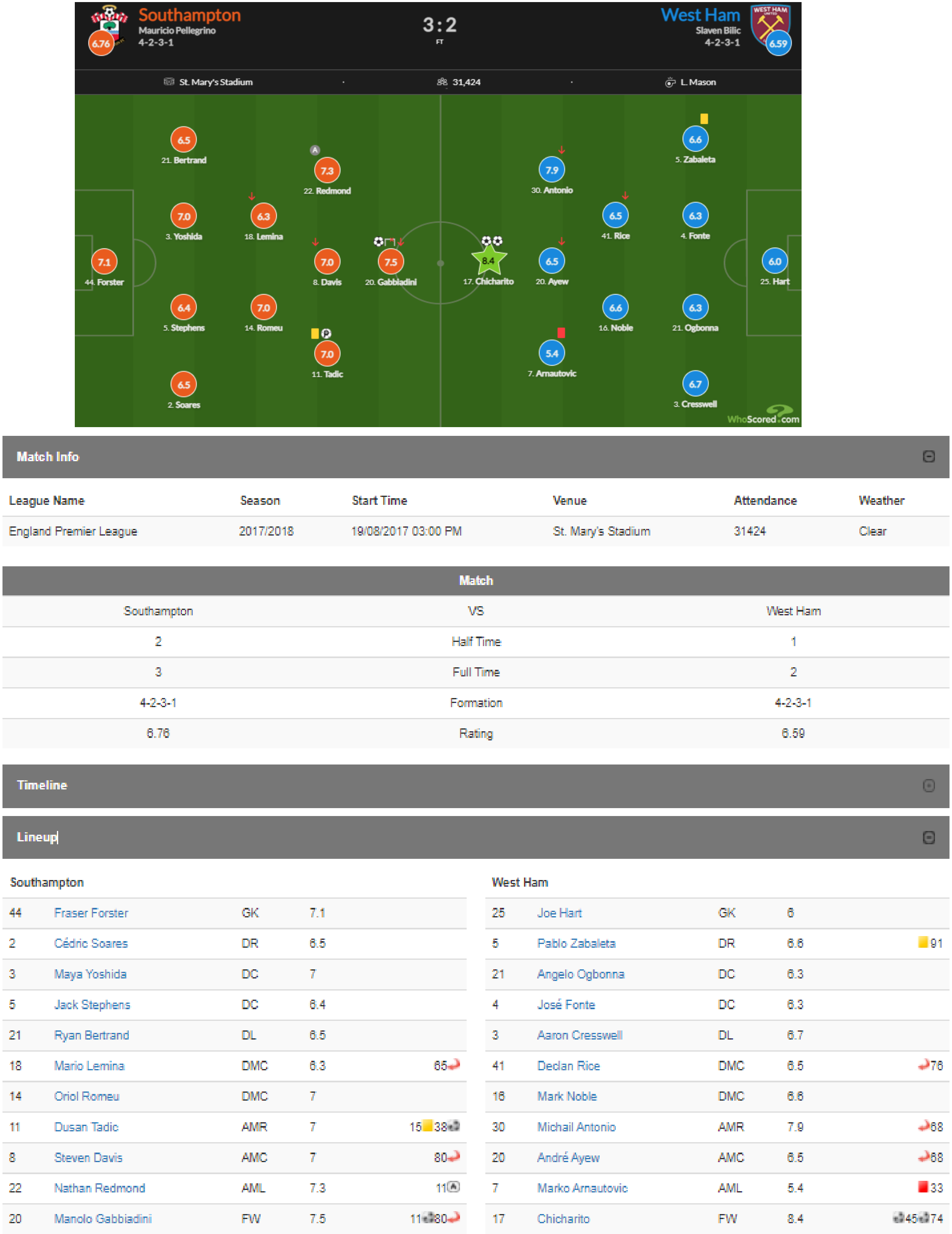
Show 10 entries

Search:

Start Time	Match	Result	
11/08/2017 07:45 PM	Arsenal v Leicester	4 : 3	Detail
12/08/2017 12:30 PM	Watford v Liverpool	3 : 3	Detail
12/08/2017 03:00 PM	Chelsea v Burnley	2 : 3	Detail
12/08/2017 03:00 PM	Crystal Palace v Huddersfield	0 : 3	Detail
12/08/2017 03:00 PM	Everton v Stoke	1 : 0	Detail
12/08/2017 03:00 PM	Southampton v Swansea	0 : 0	Detail
12/08/2017 03:00 PM	West Bromwich Albion v Bournemouth	1 : 0	Detail
12/08/2017 05:30 PM	Brighton v Manchester City	0 : 2	Detail
13/08/2017 01:30 PM	Newcastle United v Tottenham	0 : 2	Detail
13/08/2017 04:00 PM	Manchester United v West Ham	4 : 0	Detail

Showing 1 to 10 of 380 entries

Figure 4.8: Comparing whoscored.com and our collected data



Match Info

League Name	Season	Start Time	Venue	Attendance	Weather
England Premier League	2017/2018	19/08/2017 03:00 PM	St. Mary's Stadium	31424	Clear

Match		
Southampton	VS	West Ham
2	Half Time	1
3	Full Time	2
4-2-3-1	Formation	4-2-3-1
6.76	Rating	6.59

Timeline

Lineup

Southampton				
44	Fraser Forster	GK	7.1	
2	Cédric Soares	DR	6.5	
3	Maya Yoshida	DC	7	
5	Jack Stephens	DC	6.4	
21	Ryan Bertrand	DL	6.5	
18	Mario Lemina	DMC	6.3	65↩
14	Oriol Romeu	DMC	7	
11	Dusan Tadic	AMR	7	15🟨38⚽
8	Steven Davis	AMC	7	80↩
22	Nathan Redmond	AML	7.3	11Ⓐ
20	Manolo Gabbiadini	FW	7.5	11⚽80↩

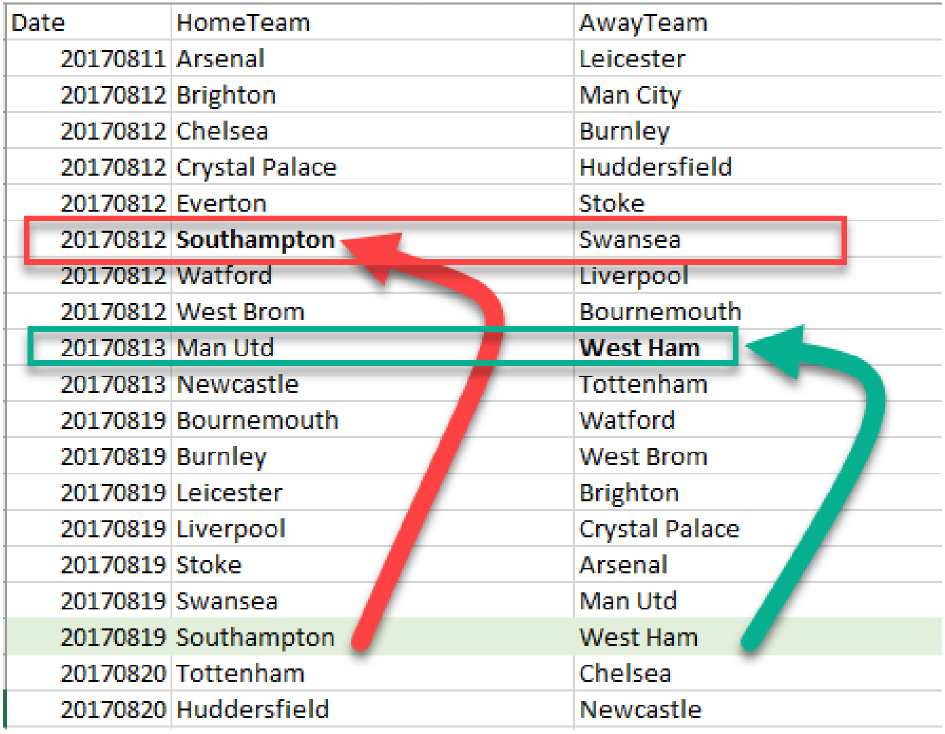
West Ham				
25	Joe Hart	GK	6	
5	Pablo Zabaleta	DR	6.6	🟨91
21	Angelo Ogbonna	DC	6.3	
4	José Fonte	DC	6.3	
3	Aaron Cresswell	DL	6.7	
41	Declan Rice	DMC	6.5	↩76
16	Mark Noble	DMC	6.6	
30	Michail Antonio	AMR	7.9	↩68
20	André Ayew	AMC	6.5	↩68
7	Marko Arnautovic	AML	5.4	🟥33
17	Chicharito	FW	8.4	⚽45⚽74

However, this automation is not as straightforward as it seems to be. While we are doing this automation, *whoscored.com* (n.d.) did add some security challenges on their data, ultimately increasing difficulties in our work. Extra efforts such as decrypting the browser's cookies are needed to get this done to avoid going through the manual way. From the data we collected from *whoscored.com* (n.d.), we will be using match date, playing teams, final score, and the list of player's ratings. Simultaneously, other data like Venue, Attendance, Weather will not be used in our model.

4-2 Calculate Player's Attack and Defend Parameter

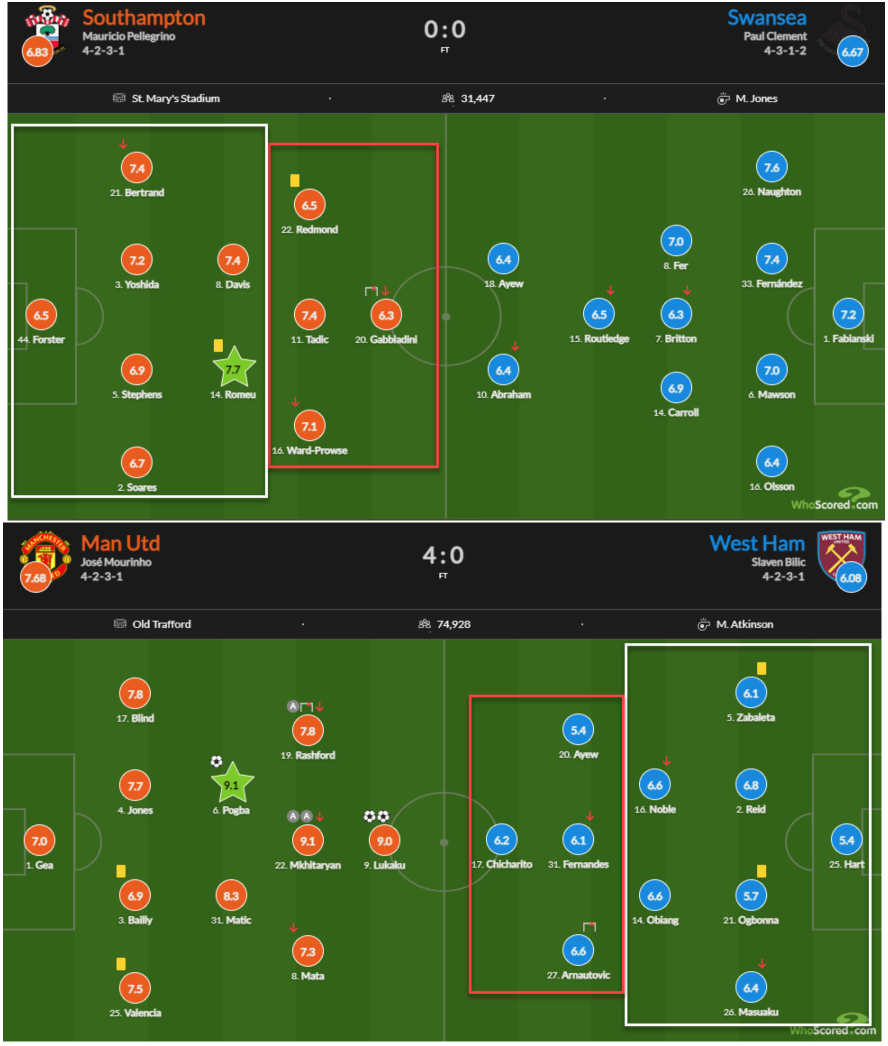
After we got the data ready in the database, we will go through the player's rating and calculate our player's attack and defend ability parameters. In our model, we are interested in the player's rating in the last match. For example, we pick a match in the 17/18 season as our target, Southampton vs. West Ham, playing on 19/08/2017. In this selected match, we have Southampton playing at home and West Ham at away position. So we have to look back at the match schedule, pick the latest previous match, which Southampton was playing at home, and West Ham at away position, and we got Southampton vs. Swansea (12/08/2017) and Manchester United vs. West Ham (13/08/2017), respectively.

Figure 4.9: Selecting past match to get player rating



Date	HomeTeam	AwayTeam
20170811	Arsenal	Leicester
20170812	Brighton	Man City
20170812	Chelsea	Burnley
20170812	Crystal Palace	Huddersfield
20170812	Everton	Stoke
20170812	**Southampton**	Swansea
20170812	Watford	Liverpool
20170812	West Brom	Bournemouth
20170813	Man Utd	**West Ham**
20170813	Newcastle	Tottenham
20170819	Bournemouth	Watford
20170819	Burnley	West Brom
20170819	Leicester	Brighton
20170819	Liverpool	Crystal Palace
20170819	Stoke	Arsenal
20170819	Swansea	Man Utd
20170819	Southampton	West Ham
20170820	Tottenham	Chelsea
20170820	Huddersfield	Newcastle

Figure 4.10: Player ratings for Southampton vs Swansea and Manchester United vs West Ham



Using Southampton vs. Swansea match first, we look into the formation and list of players, and we can see that four players will be contributing to the attack (red squared), and seven players will be contributing to the defending side (white squared). If we observe some players are overlapping, it means that they are playing in midfield, which their roles will be covering both attack and defensive end. So in our model, their rating will be considered in the calculation for both attacks and defend parameters. Going forward from here, we can compute the player's attack and defend parameters by averaging the rating of players on the attacking and defending end. Home Team Player's attack rating (HA) here is 6.825 while defending rating (HD) is 7.114285. Refer to Table 4.1 for calculations.

Shirt No.	**Name**	**Position**	**Role**	**Rating**	**Average Rating**
44	Forster	GK	Defend	6.5	
2	Soares	DR	Defend	6.7	
3	Yoshida	DC	Defend	7.2	
5	Stephens	DC	Defend	6.9	**HD** = 7.114285
21	Bertrand	DL	Defend	7.4	
14	Romeu	DMC	Defend	7.7	
8	Davis	DMC	Defend	7.4	
16	Ward-Prowse	AMR	Attack	7.1	
11	Tadic	AMC	Attack	7.4	
22	Redmond	AML	Attack	6.5	**HA** = 6.825
20	Gabbiadini	FW	Attack	6.3	

Table 4.1: Player list and ratings with position for latest game of Southampton playing at home (prior to our game to be predicted)

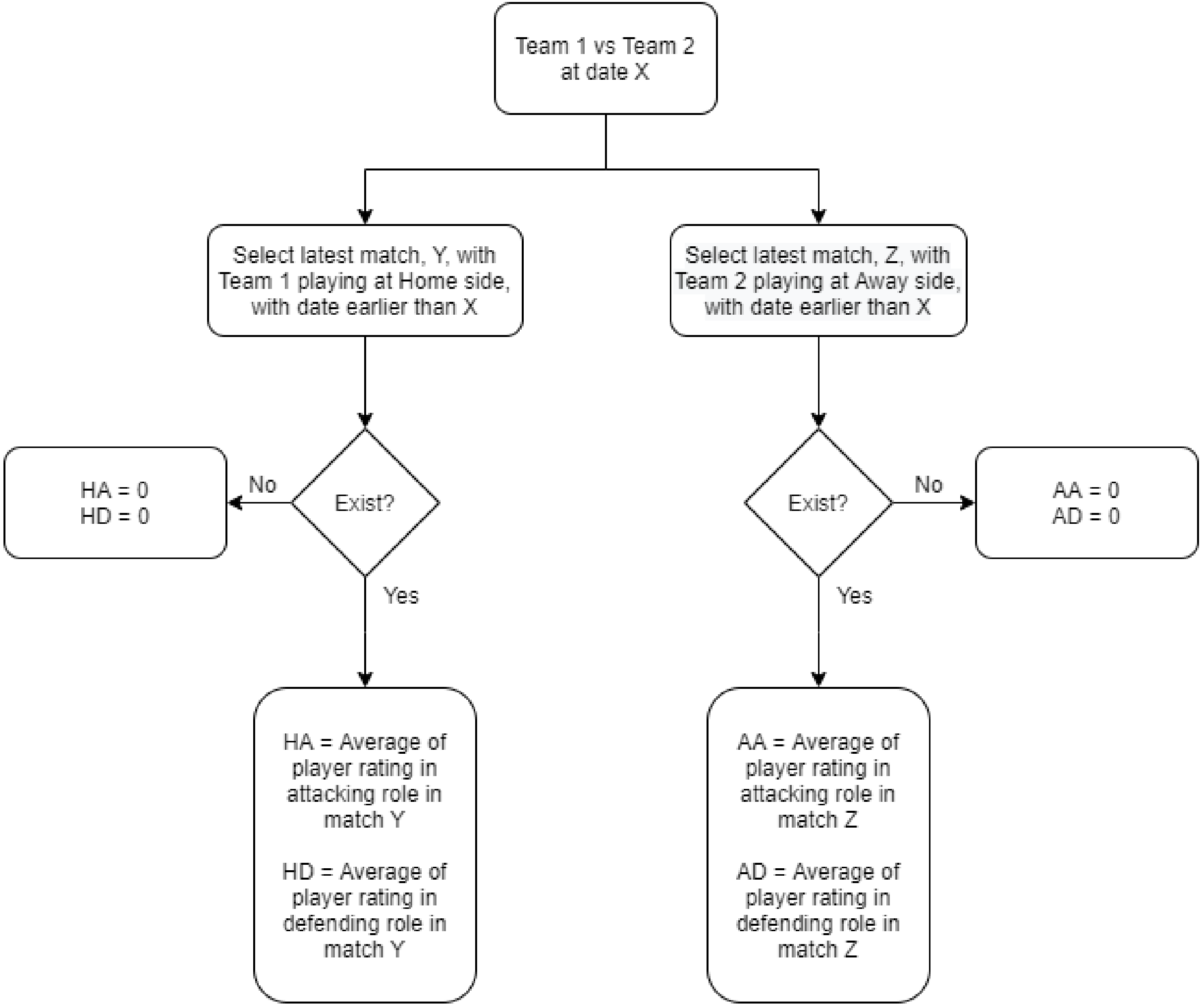
Next, we look at Manchester United vs West Ham on 13/08/2017. We have 4 players on attacking side and 7 players in defending role. Similarly we can get the player's attack rating (AA) is 6.075 and defend rating (AD) is 6.228571. Refer to Table 4.2 for calculations.

Shirt No.	Name	Position	Role	Rating	Average Rating
25	Hart	GK	Defend	5.4	
5	Zabaleta	DR	Defend	6.1	
21	Ogbonna	DC	Defend	5.7	
2	Reid	DC	Defend	6.8	**AD** = 6.228571
26	Masuaku	DL	Defend	6.4	
14	Obiang	DMC	Defend	6.6	
16	Noble	DMC	Defend	6.6	
20	Ayew	AMR	Attack	5.4	
31	Fernandes	AMC	Attack	6.1	
27	Arnautovic	AML	Attack	6.6	**AA** = 6.075
17	Chicharito	FW	Attack	6.2	

Table 4.2: Player list and ratings with position for latest game of West Ham playing at away (prior to our game to be predicted)

Using program, we repeat the calculation for each match in all 760 matches. However there are cases like Cardiff is newly promoted to English Premier League in 2018/2019 season, for such cases, for sure we don't have data on the player's rating in past match. We will set the player's attack and defend rating parameter to 0 in our model.

Figure 4.11: Illustrating steps to obtain HA, HD, AA, AD


Team 1 vs Team 2 at date X
Select latest match, Y, with Team 1 playing at Home side, with date earlier than X
Select latest match, Z, with Team 2 playing at Away side, with date earlier than X
Exist?
No
HA = 0
HD = 0
Yes
Exist?
No
AA = 0
AD = 0
Yes
HA = Average of player rating in attacking role in match Y
HD = Average of player rating in defending role in match Y
AA = Average of player rating in attacking role in match Z
AD = Average of player rating in defending role in match Z

4-3 Implementing Player's Rating into Dixon-Coles Model and Performance Evaluation

After getting the additional parameters we need, we review back Dixon-Coles model. It states that we have

$$P(X_{i,j} = x, Y_{j,i} = y) = \tau_{\lambda,\mu}(x,y)\frac{e^{-\lambda}\lambda^x}{x!}\frac{e^{-\mu}\mu^y}{y!}$$

$$where \quad \lambda = \alpha_i\beta_j\gamma \quad \mu = \alpha_j\beta_i$$

$$and \quad \tau_{\lambda,\mu}(x,y) = \begin{cases} 1-\lambda\mu\rho, & if \quad x=y=0 \\ 1+\lambda\rho, & if \quad x=0, y=1 \\ 1+\mu\rho, & if \quad x=1, y=0 \\ 1-\rho, & if \quad x=y=1 \\ 1, & otherwise \end{cases}$$

$$\rho, \quad where \quad max(-1/\lambda, -1/\mu) \le \rho \le min(1/\lambda\mu, 1)$$

We modify the model above by implementing four new parameters, which we got from player's rating, Home Attack (HA), Home Defend (HD), Away Attack (AA), and Away Defend (AD) to Dixon-Coles model. The proposed model is as below.

$$P(X_{i,j} = x, Y_{j,i} = y) = \tau_{\lambda,\mu}(x,y)\frac{e^{-\lambda}\lambda^x}{x!}\frac{e^{-\mu}\mu^y}{y!}$$

$$where \quad \lambda = \alpha_i\beta_j\gamma(HA - AD) \quad \mu = \alpha_j\beta_i(AA - HD)$$

$$and \quad \tau_{\lambda,\mu}(x,y) = \begin{cases} 1-\lambda\mu\rho, & if \quad x=y=0 \\ 1+\lambda\rho, & if \quad x=0, y=1 \\ 1+\mu\rho, & if \quad x=1, y=0 \\ 1-\rho, & if \quad x=y=1 \\ 1, & otherwise \end{cases}$$

$$\rho, \quad where \quad max(-1/\lambda, -1/\mu) \le \rho \le min(1/\lambda\mu, 1)$$

By using $\alpha, \quad \beta, \quad \gamma, \quad and \quad \rho$ value discussed in chapter 3, and HA, HD, AA, AD value discussed earlier in this chapter, and we obtain the following values, comparison between Dixon-Coles model and our Dixon-Coles with Player Rating. Results are shown in Table 4.3. The final result for this match is 3-2, a victory to Southampton. Dixon-Coles Model predicted Southampton win at 42.6767%, while our enhanced model predicted Southampton's victory at 84.95%.

Outcome	Dixon-Coles Model	Dixon-Coles with Player Rating
Southampton Win	42.6767	84.95191
Swansea Win	28.92918	3.076104
Draw	28.39412	11.97199

Table 4.3: Comparing Computed Probability (%) for Southampton vs West Ham between Dixon-Coles Model and Dixon-Coles Model with Player Rating

The above result looks suitable for this single selected match, and next, we proceed to expand our prediction to more matches. In English Premier League, there are 20 playing teams to play with each other, which sums to 380 matches in a single season. However, every season, there will be three teams promoting into English Premier League while three other teams relegated from English Premier League. Hence our parameters will not be 100% applicable to the 18/19 season. Only 17 out of 20 teams are valid (in other words, still playing in the league). Taking this into consideration, we are only able to predict 272 matches. We cannot predict 108 matches due to lacking team attack and defend parameters from these three teams.

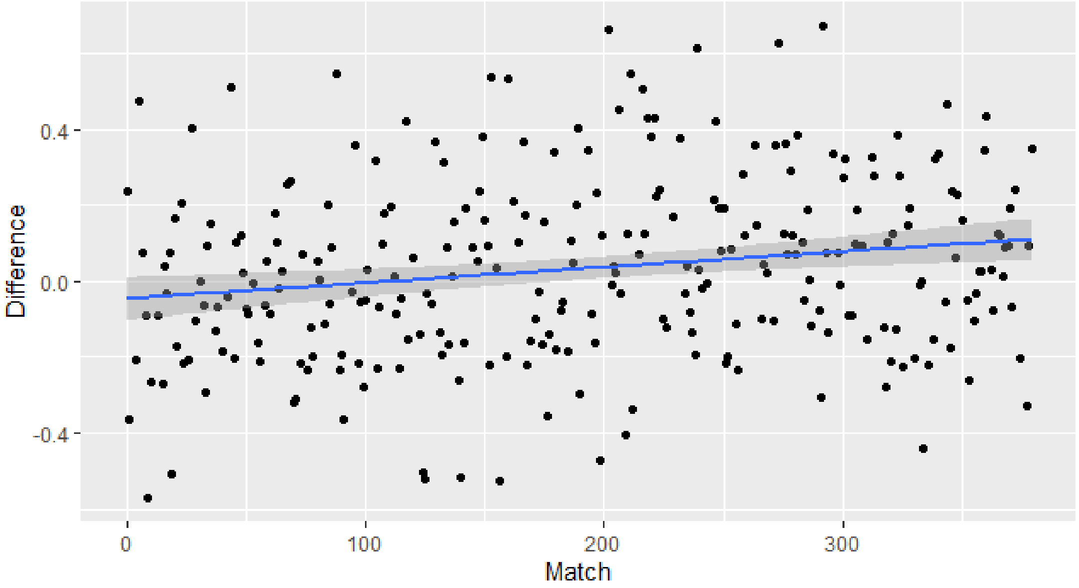
To perform this test, we first read all the 380 matches for the 18/19 season from CSV, along with the participating team's name and player rating parameter (for the steps of obtaining player rating parameter, refer to Figure 4.11). For each match, we perform Dixon-Coles Model and Dixon-Coles Model with Player Rating. After that, we determine the match's outcome and then compare the two models' performance. The differences between the two models are recorded.

The positive value of difference indicating our model predicts better, while the negative value of difference means the original Dixon-Coles model does a better job in prediction. A plot illustrating our result is as shown in Figure 4.12. In summary,

Perform Better	Not Perform Better	Not Comparing
145 matches	127 matches	108 matches

Table 4.4: Did player rating improve the prediction, for season 2018/2019

Figure 4.12: Plot of comparison between both model, for season 18/19



Examining our result, we can see that the most significant improvement is Southampton vs. Tottenham playing on 2019/03/09. The final result is 2-1. Dixon-Coles Model predicted that the probability of Southampton win at 15.24% while our model predicts that victory goes to Southampton at 82.57%, a difference of 67.33%.

While on the other end, our model's biggest flop is Arsenal vs. Manchester City playing on 2018/08/12. The final result is 0-2. Dixon-Coles Model predicted that the Away team win at 61.94% while our Model predicts that the Away team win at only 4.86%. There is a 57.07% difference.

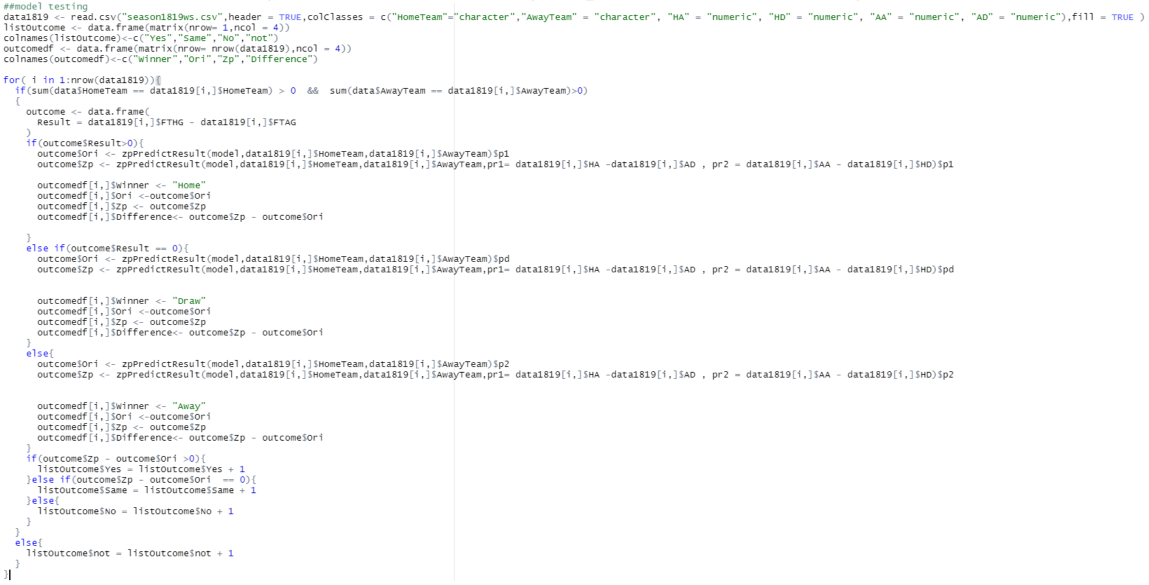
Among the 145 matches which we predict better than Dixon-Coles Model, on average, we predict 21.05% better. Our improvement is ranging from 0.08% to 67.33% Among the 127 matches which we predict worse than Dixon-Coles Model, on average, we predict 17.10% worse, Dixon-Coles Model is better than our Model at the range of 0.03% to 57.07%.

If we compare the prediction among two models by match result,

	Home Win	Away Win	Draw
Our Model	87	41	17
Dixon-Coles Model	40	50	37

Table 4.5: Breakdown of prediction vs match result

Figure 4.13: Illustrating steps to do model testing



```
##model testing
data1819 <- read.csv("season1819ws.csv",header = TRUE,colClasses = c("HomeTeam"="character","AwayTeam" = "character", "HA" = "numeric", "HD" = "numeric", "AA" = "numeric", "AD" = "numeric"),fill = TRUE )
listOutcome <- data.frame(matrix(nrow= 1,ncol = 4))
colnames(listOutcome)<-c("Yes","Same","No","not")
outcomedf <- data.frame(matrix(nrow= nrow(data1819),ncol = 4))
colnames(outcomedf)<-c("Winner","Ori","Zp","Difference")

for( i in 1:nrow(data1819)){
  if(sum(data$HomeTeam == data1819[i,]$HomeTeam) > 0  &&  sum(data$AwayTeam == data1819[i,]$AwayTeam)>0)
  {
    outcome <- data.frame(
      Result = data1819[i,]$FTHG - data1819[i,]$FTAG
    )
    if(outcome$Result>0){
      outcome$Ori <- zpPredictResult(model,data1819[i,]$HomeTeam,data1819[i,]$AwayTeam)$p1
      outcome$Zp <- zpPredictResult(model,data1819[i,]$HomeTeam,data1819[i,]$AwayTeam,pr1= data1819[i,]$HA -data1819[i,]$AD , pr2 = data1819[i,]$AA - data1819[i,]$HD)$p1

      outcomedf[i,]$Winner <- "Home"
      outcomedf[i,]$Ori <-outcome$Ori
      outcomedf[i,]$Zp <- outcome$Zp
      outcomedf[i,]$Difference<- outcome$Zp - outcome$Ori

    }
    else if(outcome$Result == 0){
      outcome$Ori <- zpPredictResult(model,data1819[i,]$HomeTeam,data1819[i,]$AwayTeam)$pd
      outcome$Zp <- zpPredictResult(model,data1819[i,]$HomeTeam,data1819[i,]$AwayTeam,pr1= data1819[i,]$HA -data1819[i,]$AD , pr2 = data1819[i,]$AA - data1819[i,]$HD)$pd


      outcomedf[i,]$Winner <- "Draw"
      outcomedf[i,]$Ori <-outcome$Ori
      outcomedf[i,]$Zp <- outcome$Zp
      outcomedf[i,]$Difference<- outcome$Zp - outcome$Ori
    }
    else{
      outcome$Ori <- zpPredictResult(model,data1819[i,]$HomeTeam,data1819[i,]$AwayTeam)$p2
      outcome$Zp <- zpPredictResult(model,data1819[i,]$HomeTeam,data1819[i,]$AwayTeam,pr1= data1819[i,]$HA -data1819[i,]$AD , pr2 = data1819[i,]$AA - data1819[i,]$HD)$p2


      outcomedf[i,]$Winner <- "Away"
      outcomedf[i,]$Ori <-outcome$Ori
      outcomedf[i,]$Zp <- outcome$Zp
      outcomedf[i,]$Difference<- outcome$Zp - outcome$Ori
    }
    if(outcome$Zp - outcome$Ori >0){
      listOutcome$Yes = listOutcome$Yes + 1
    }else if(outcome$Zp - outcome$Ori  == 0){
      listOutcome$Same = listOutcome$Same + 1
    }else{
      listOutcome$No = listOutcome$No + 1
    }
  }
  else{
    listOutcome$not = listOutcome$not + 1
  }
}
```

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

To summarize, considering the player's rating, as one of the Dixon-Coles Model criteria, did lead to improvement overall. As per what we tested with English Premier League season 2018/2019. We predict 145 matches better from the breakdown from Dixon-Coles Model, while 127 matches are worse than Dixon-Coles Model. Our best improvement is our predicted probability is 0.6733 higher than Dixon-Coles Model, while our worst-case scenario is our predicted probability is 0.5707 lower than Dixon-Coles Model. On average, among the matches that our prediction is better, we are better by probability 0.2105 higher than Dixon-Coles Model. Meanwhile, among the matches that our prediction is worse, we are lower by, on average, a probability of 0.1710.

As we can see, our model is performing better than Dixon-Coles Model overall. But of course, there is still room for improvement. Due to time constraints, we are yet to test some ideas and enhancement.

Among the matches that our prediction is worse than Dixon-Coles Model, it could be due to many reasons, such as the critical player injured, a player being sent out, player fatigues due to the tight schedule, etc. Our model is currently unable to address these scenarios. These are some shortcomings in the current model.

To further improve our model, we propose to work on the following

1. Generate and update the team's attack and defend parameter more frequently

   In our model, the team's parameter is generated by using data from the previous season, 2017/2018. This might be sufficient for the first few matches in the coming season. However, as the tournament continues, the parameters might be outdated. For example, when the tournament is completed 75%, there were many matches played between the beginning of the 18/19 season and the 75% milestone of the 18/19 season. Hence, the parameters generated in the beginning stage of the season are insufficient to describe or insufficient to better grasp what is going on in the pitch.

So we suggest recalculating the parameters more frequently.

2. Expand the test on other leagues/competition

   Due to time constraints, our work is set to fulfilling the most popular league, English Premier League. The result we got is satisfactory in English Premier League. We should expand our model testing to cater to other leagues or competitions, such as the Spain LaLiga, Germany Bundesliga, France Ligue 1, and Italy Serie A. These are the top popular leagues on the European continent. By testing on more leagues, we will evaluate our model's robustness and be able to learn more about the model, whether there are tweaks or other improvements needed.

3. Take player's rating on average of a few matches

   Currently, our model takes the only player rating of the latest historical match. However, undeniable, even the best player could have a dip in performance once in a while. For example, in the last five matches, a top player gets the rating of 8, 8, 8, 8, 5. In our current model, we will consider the player as rating 5. In such a case, we might be underestimating a top player's performance. This can be overcome if we take an average. In this case, if we take the average, the player's rating will be 7.4, which is a better reflection of the player's ability.

4. Take player's rating time decay rate into consideration

   Further, elaborate on the above point. We imagine a scenario where a mediocre player gets a phenomenal performance on one of the past matches. We let the rating for the last five matches be like 9, 5, 6, 5, 5. Our current model will be taking this player's rating is 5, and if we implement the suggestion in point #3, the player's rating will be 6. However, we see that the phenomenal performance only happens once, few matches ago. We should apply a discounting factor so that the rating from older matches doesn't have a similar weightage compare to the more recent matches.

5. Consider player's attendance

   In our current model, we consider the player's rating, but we disregard the fact that the same team might not be playing with the same player lineup in the matches. The key player might be rested/injured, or the manager might decide to do a rotation in the squad, which will make our player's rating parameter not so meaningful. For example, the team rested the critical player with a high rating, let's say a 9 in the past game. So logically, if we take the resting player into our model's consideration, we might be predicting wrongly.

6. Consider substitution instead of only looking at the starting eleven and the time they are playing in pitch

   Currently, our model only considers the starting eleven's rating. However, substitution is also part of the game, and often a good substitution strategy will change the tide of the game. In such cases, considering the substitute player's rating will have a positive influence on our prediction. The substitution player does not play for the full 90 minutes, so we might need to consider discounting the rating, depends on the time they play on the pitch. Further testing is needed to support this.

7. Adjust player's rating according to the opponent

   When a strong team is playing against a weak team, no doubt, the stronger team player will tend to have a higher player rating since it is an easy game for them. We consider a scenario: Arsenal is at ranking #2, played with Leeds United at ranking #20, Arsenal gets overall a higher player's rating. However, the next game for Arsenal is opponent at ranking #1, Chelsea. It is less likely for a player of Arsenal to replicate the same performance when playing against Leeds. We might need to adjust the player's rating accordingly for such cases.

8. Objective of playing

   This commonly happens, especially when the league is in the last few weeks, where some of the teams have secured the places or achieved their objective of the season. Winning or losing the last few matches doesn't affect their result, so the players are not motivated in the game and might be playing casually. On the opposite side, some teams might be fighting to avoid relegations or promotions in the last few games, so their desire to win is higher. This might affect the player's performance as well. We might be able to apply some adjustments for these cases, whether the team is motivated or not.

Besides predicting soccer results, we can also apply this concept to predict red card occurrence or corner kick in a soccer match. We can further apply this model to other sports out of soccer, such as futsal, as they share several similar properties, such as low scoring and time-based. The underlying distribution in the Dixon-Coles model is Poisson distribution, so as long as the event is statistically independent and the rate of happening is constant, we can have a lot of application of this Poisson distribution. Out of sports, Poisson distribution can also be applied to cases in different sectors. For example, in networking, network failures or packet loss per hour. On the website, the number of visitors expecting, in a period of time. In banking services, the number of customer arrival in one hour. In a call centre, the number of phone calls received in a period of time.

REFERENCES

C. Reep, R. Pollard, B. B., 1971. ‘An evaluation of characteristics of teams in association football by using a markov process model’, *Journal of the Royal Statistical Society. Series A (General)* **134**(4), 623–629.

C.Reep, B. B., 1968. ‘Skill and chance in association football’, *Journal of the Royal Statistical Society. Series A (General)* **131**(4), 581–585.

D. Karlis, I. N., 2003. ‘Analysis of sports data by using bivariate poisson models’, *Journal of the Royal Statistical Society. Series D (The Statistician)* **52**(3), 381–393.

Ian G. McHale, L. S., 2014. ‘A mixed effects model for identifying goal scoring ability of footballers’, *Journal of the Royal Statistical Society. Series A (Statistics in Society)* **177**(2), 397–417.

Keller, J., 1994. ‘A characterization of the poisson distribution and the probability of winning a game’, *The American Statistician* **48**(4), 294–298.

M. Dixon, M. R., 1998. ‘A birth process model for association football matches’, *Journal of the Royal Statistical Society. Series D (The Statistician)* **47**(3), 523–538.

M. Dixon, S. C., 1997. ‘Modelling association football scores and inefficiencies in the football betting market’, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **46**(2), 265–280.

Maher, M. J., 1982. ‘Modelling association football scores’, *Statistica Neerlandica* **36**, 109–118.

Moroney, M. J., 1956. *Facts from Figures*, Penguin, London.

Nobuyoshi Hirotsu, M. W., 2003. ‘An evaluation of characteristics of teams in association football by using a markov process model’, *Journal of the Royal Statistical Society. Series D (The Statistician)* **52**(4), 591–602.

R. Pollard, C. R., 1997. ‘Measuring the effectiveness of playing strategies at soccer’, *Journal of the Royal Statistical Society. Series D (The Statistician)* **46**(4), 541–550.

S. Clarke, J. N., 1995. ‘Home ground advantage of individual clubs in english soccer’, *Journal of the Royal Statistical Society. Series D (The Statistician)* **44**(4), 509–521.

whoscored.com, n.d.. `http://www.whoscored.com/`.
URL: *http://www.whoscored.com/*