Modelling football goals

Abstract

This project explores to establish the use of statistical models Namely Naive Poisson Model and Dixon-Coles Model, in analysing football match results. Due to the growing concern of using sports statistics and the massive adoption of sports betting, proper goal prediction models play a critical role. The Naive Poisson Model is a basic model which supposes that goals are ordered independently by means of a regular mean rate. However, the Dixon-Coles Model builds up on this by considering team strengths, dependencies between the teams and temporality, all of which leads to better prediction. From a comparison of these models, the project shows that the Dixon-Coles Model affords superior forecasts especially for low-scoring competencies since the model considers the interdependency of performances by the competing teams. The obtained results, estimated using MSE and other statistical indices, confirm the devised Dixon-Coles Model superior to the Naive Poisson Model. This work enlightens the theoretical domain of Sport Analytics and offers practical implications in the betting market, team assessment, and dynamic prediction system.

Acknowledgment

I would like to express my deepest gratitude to all those who have supported and guided me throughout this project. My sincere thanks go to my advisor, Mr. Ben Parker , whose expertise and insightful feedback were invaluable in shaping the direction and scope of this research. I am also grateful to Brunel university london/Mathematics department for providing the resources and environment conducive to conducting this study. Special thanks are extended to my peers and colleagues who offered their perspectives and shared in the discussions that enriched this work. Finally, I would like to acknowledge the invaluable support of my family and friends, whose encouragement and patience kept me motivated throughout this journey. Thank you all for your unwavering support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Modelling of Football goals:

Many statisticians and data scientists are now focusing on modelling. Modelling in sports, most frequently in football, involves using statistical methods to predict outcomes such as match results and goals scored. Sports activities are always followed by many people. In recent days, many competitions in many countries have collected new information, such as data on each shot taken or information from the competition that contains insights beyond the game.

Many models, including the Dixon-Coles model, enhance predictions by considering these various factors. This project is based on statistical analysis and modelling to determine the optimal model for predicting the final score of a football match. With advancements in technology and machine learning algorithms, accurately predicting football match outcomes has become feasible. The primary aim of this paper is to investigate the effectiveness of the Poisson and Dixon-Coles models in forecasting football results.

These days, modelling is the focus of many statisticians and data scientists. Statistical techniques are used in sports modelling, most commonly in football, to forecast game outcomes and goal totals. Many people are always interested in sports. Recently, a lot of contests across a lot of nations have gathered fresh data, such as statistics on every shot made or information from the competition that offers insights beyond the game.

In other words, it is better to directly model the outcome of the match rather than using the objective model to predict the match scores and then obtain similar results. The results shown for the teams participating in the football match are win, lose or draw. It is very easy to predict the outcome of a race. Traditional forecasting methods only use game results to evaluate team performance and build statistical models to predict the outcomes of future games. In this project there is an argument and interpretation which is the best model to predict the football goals.

1.2 Objective:

The main objective of the project is to examine the use of the Poisson and Dixon-Coles models in predicting football matches. The recent performance of each player can be considered to create a model.

A more accurate prediction can be made by including a player's recent performance as an additional measurement. Demonstrate how the Dixon-Coles model can provide more accurate predictions by using multiple

factors that affect the outcome of a game. Create a framework for instant prediction of match results that can be used in gambling.In chapter 2 given the explanation for literature review by following them with chapter 3 naive model is explained. And in chapter 4 and 5 given detailed explanation of dixon-coles model and brier score.In chapter 6 practical application and difference between dixon and naive model is explained. And in the conclusion there is an argument which explains the best model to predict football goals.

Suggest areas where further research is needed to improve the accuracy and reliability of the football match. Predictions and simulations were also performed in R. We used the ggplot package in R for data visualisation.

1.3 Problem Statement

In recent years, sports models have attracted great interest from researchers and data scientists due to their applications in academic research and commercial sports betting.

An important part of using sports models is to predict the outcome of a football match which is important for bettors. With the emergence of sports betting where bets can be made at any time during the match, the need for accurate and dynamic prediction models is increasing. These models usually use a Poisson distribution with a constant value to represent the average number of goals scored. A naive model of the goals scored during an association football match might be that goals are independent of the teams playing and of any previous goals. However, research shows that this model does not fit particularly well, particularly in the tails of the distribution. A more advanced model is the Dixon-Coles model (Dixon and Coles, 1997), and many modifications of this model have been released subsequently. This project is to perform statistical analysis and modelling to find the best model for predicting the final scores in a football match.

1.4 Goal-based models:

Poisson Model:(Naive Poisson model)

- Assumes goals are scored independently and at a constant average rate (λ).
- Probability of scoring n goals

Dixon-Coles Model:

- Extends the Poisson model by incorporating team strengths and dependencies between the scores of the two teams.
- Adjusts for overdispersion and includes a time-dependent factor for goal scoring.

Chapter - 2

Literature review

- In 1982 Maher supported the assumption that the number of goals scored by each team in a football game is distributed as the Poisson process. In this model, it is assumed that the attacks of everyone are independent Hence in a single opportunity to attack,the probability for anyone scoring will be constant. The Poisson model assumes that the variances in game type are for true mean rates: each team has always played ha-lf their games at home and half away, so there is not a structural difference.
- Maher introduced parameters for team attacking and defensive strengths in their model [14]. But more specifically if team i plays at home against team j, then the goals scored by them (Xi,j) and the number of goals received by that same time in this match modelled as Poisson variable with mean value for Xi,j.Similar to Yi,j. Maher (1982) updated his models to include a home advantage, which improved matters somewhat but still overestimated high-scoring games compared to the number that were actually played and underestimated low- scoring ones. Maher dealt with these problems by introducing a bivariate Poisson model to account for the correlation between scores.
- For example, Harvey (1974) identified home advantage but S. Clarke (1995) reported that differences in levels of crowd support and the physical dimensions of pitches meant such effects were not consistent across clubs.
- Maher's model was extended by M. Dixon (1997) who confirmed that it still under-predicts low scores and over-estimates high scores compared with the combination of totals observed from single sets at all levels, across formats. Among Dixon's adjustments was a new model parameter and unfortunately this often overcorrect, causing the algorithm to give too much weight towards low scoring games. In order to tackle this, D. Karlis (2003) has made the modification in bivariate Poisson model by doing a change of odds parameters and it has certain benefits as well which are already talked about here but below is summary once again:
- To account for the time dynamics, constrained by both the remaining match-time and current score, M. Dixon (1998) proposed a Markov model with predictive accuracy supported by Monte Carlo simulations when compared against historical data set evidence;
- R. Pollard (1997) quantified strategies by measuring the utility of moves but no statistical distributions are used in this work specifically
- Nobuyoshi Hirotsu (2003) further developed the model by incorporating a Markov process with four states to represent possession dynamics. For his purposes in modelling football performance, Ian G. McHale (2014): Individual goal scoring ability They used goals per minute as a predictor of player performance.
- A Brier score is a metric used to measure the accuracy of a prediction. In the context of football goals, it can be used to assess the accuracy of a predicted outcome, such as the probability of a team scoring a

certain goal in a match. Brier scores range from 0 to 1, with 0 being perfect accuracy and 1 being worst accuracy. Glenn W. Brier proposed the Brier score in the 1950s as a way to evaluate the accuracy of predictions.

- In summary, the literature reveals continued progress in football match modelling: Poisson models and Negative Binomial model to capture over-dispersion; home advantage or time dynamics inclusion and individual performance metrics. And given the explanation about the brier score which gives the accuracy of the prediction.

Research Flowchart, chronologically

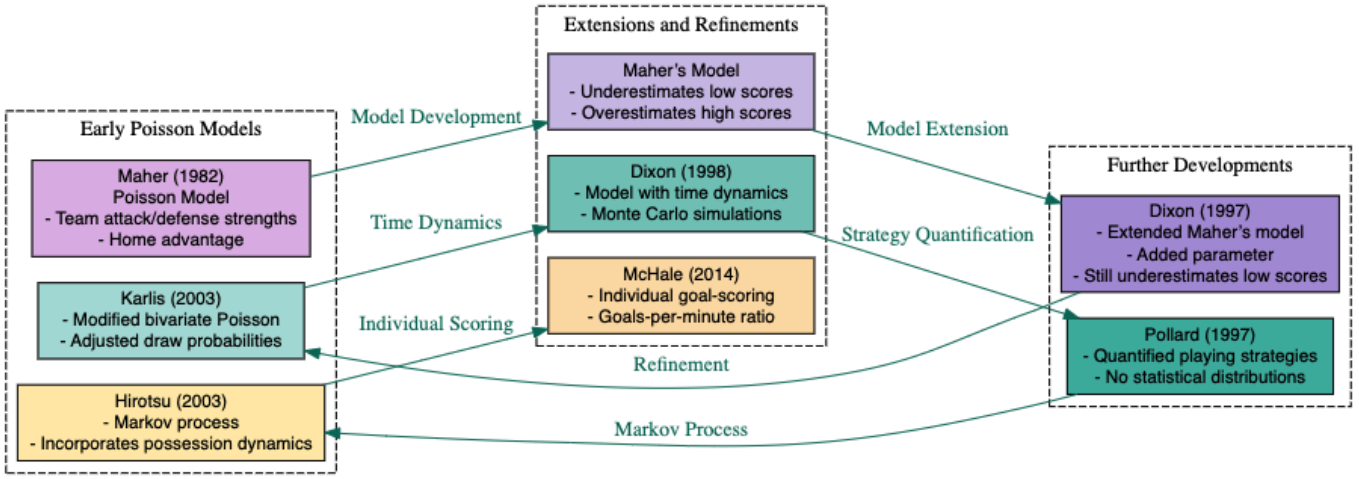

Figure 2.1 :This diagram shows how football goal modelling has grown over time. It started with basic Poisson-based models and moved to more advanced methods. These newer approaches take into account extra elements such as time, strategy, and how individual players perform. Each new step builds on or expands earlier models. The aim is to fix shortcomings and make predictions more accurate in different situations.

Summary of the contributions :

Goddard (2005):

Conducted a comprehensive comparison of various statistical models for football match outcomes, concluding that the Dixon-Coles model consistently outperformed the naive Poisson model.

Hvattum and Arntzen (2010):

Evaluated predictive accuracy in betting markets, demonstrating that the Dixon-Coles model provided better returns than the naive model.

Naive Model

3.1 Introduction:

The naive Poisson model is a base statistical model for event occurrences count in fixed or variable time frames. It is called "naive" because it makes assumptions about data, particularly assumptions regarding equal means and variances which are naturally occurring when data follow a Poisson distribution.

The naive model of football goals usually includes the assumption that the goals scored by each team follow a Poisson distribution. The model is simple and assumes that the goals are independent and the average score over time does not vary. Here is a detailed description and application of the naive Poisson model.

3.2 Poisson Distribution

The Poisson distribution is used to model the number of events (in this case, goals) that occur within a fixed interval of time or space. The probability mass function of a Poisson-distributed random variable N with parameter λ is given by:

$$\Pr(N = n) = \frac{\lambda^n e^{-\lambda}}{n!}$$

- where:
- n is the number of goals.
- λ is the average number of goals scored in a match.

For example:

To explain the above formula, apply this formula randomly in predicting the goals for 0,1,2,3 and 4, whereas λ is the average number of goals scored in a match which is 2 from the above example.

Probability of scoring	Formula	Answer
0	$Pr(N = 0) = \frac{2^0 e^{-2}}{0!}$	0.13533528323
1	$Pr(N = 1) = \frac{2^1 e^{-2}}{1!}$	0.27067056647
2	$Pr(N = 2) = \frac{2^2 e^{-2}}{2!}$	0.27067056647
3	$Pr(N = 3) = \frac{2^3 e^{-2}}{3!}$	0.18044704431
4	$Pr(N = 4) = \frac{2^4 e^{-2}}{4!}$	0.09022352215

Table 3.1

3.3 Key Assumptions of the Naive Poisson Model:

1. Independence of Goals: Goals scored by each team are independent of each other.
2. Constant Rate (λ): Each team scores goals at a constant average rate λ per match.
3. No Home Advantage: The model does not differentiate between home and away teams.

3.4 Implementation of the Naive Poisson Model in R

The coding is available in the appendix section (Coding 1) and below is the plot of the coding available in the appendix section.

The dataset contains detailed match-level data from the 2015-16 football season from the English premiership with 380 entries and 65 columns from kaggle. This dataset provides comprehensive match statistics, outcomes, and a variety of betting odds for each game. It's useful for analysing team performance, predicting results, and studying betting market movements during the 2015-16 season.analyzing team performance, predicting results, and studying betting market movements during the 2015-16 season.

Filter | Cols: « < 1 - 50 > »

Div	Date	HomeTeam	AwayTeam	home_goals	away_goals	FTR	HTHG	HTAG	HTR	Referee
E0	08/08/15	Bournemouth	Aston Villa	0	1	A	0	0	D	M Clatten
E0	08/08/15	Chelsea	Swansea	2	2	D	2	1	H	M Oliver
E0	08/08/15	Everton	Watford	2	2	D	0	1	A	M Jones
E0	08/08/15	Leicester	Sunderland	4	2	H	3	0	H	L Mason
E0	08/08/15	Man United	Tottenham	1	0	H	1	0	H	J Moss
E0	08/08/15	Norwich	Crystal Palace	1	3	A	0	1	A	S Hooper
E0	09/08/15	Arsenal	West Ham	0	2	A	0	1	A	M Atkinsc
E0	09/08/15	Newcastle	Southampton	2	2	D	1	1	D	C Pawson
E0	09/08/15	Stoke	Liverpool	0	1	A	0	0	D	A Taylor
E0	10/08/15	West Brom	Man City	0	3	A	0	2	A	M Dean
E0	14/08/15	Aston Villa	Man United	0	1	A	0	1	A	M Dean

Figure 3.1- An extract of the dataset from english premiership league 2015-2016

The output:

MSE Calculation: The Mean Squared Error is calculated by squaring the difference between actual goals and predicted goals, averaging these squared differences.

Overall MSE: This combines the MSEs for home and away goals to give a single measure of prediction error.

```
MSE for home goals: 1.581517
MSE for away goals: 1.312043
Overall MSE: 1.44678
```

Average goals scored by home and away team

```
Team Avg_Goals
1 Home  1.492105
2 Away  1.207895
```

General Insights:

● **Home vs. Away Performance:** By comparing the two Poisson distribution plots, you can infer whether home teams generally have a higher or lower scoring potential compared to away teams. This is useful for understanding home advantage or predicting match outcomes.

● **Real-World Application:** These plots can be used in sports analytics to predict outcomes, set expectations for matches, or even in betting models to calculate odds based on expected goal distributions.

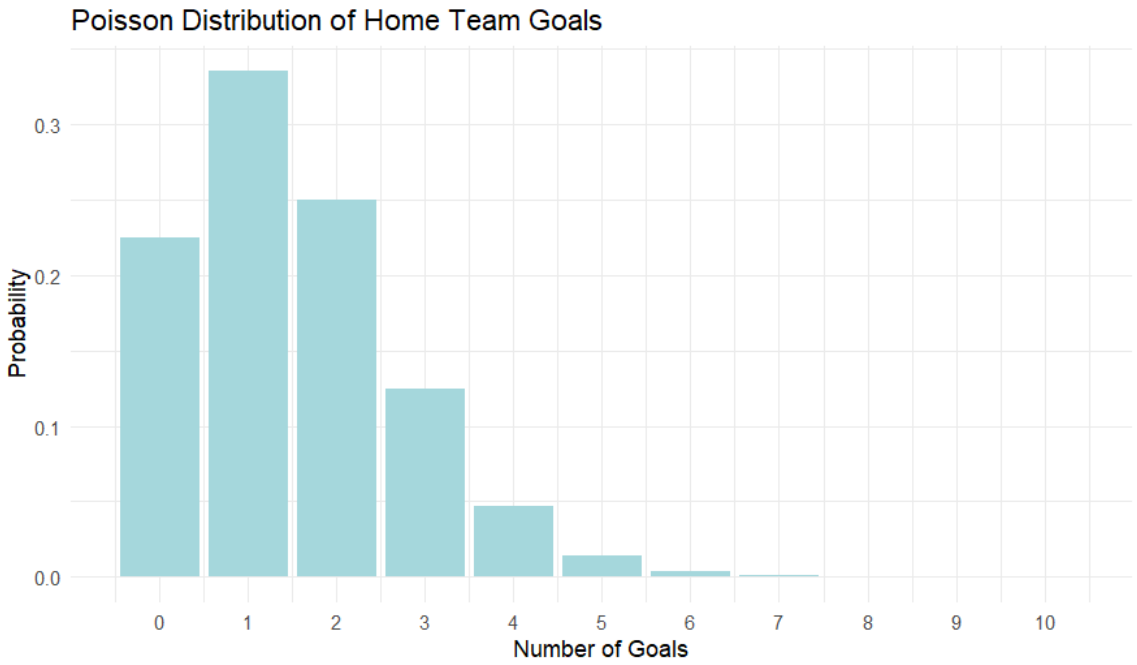

Figure 3.2- Plot between Poisson Distribution of Home Team Goals

Poisson Distribution Plot for Home Team: A bar plot shows the chance of the home team scoring different numbers of goals based on the Poisson model

Interpretation:

The y axis of each bar indicates the Poisson distribution -probability that the home team has scored a specified number of goals. One or two goals will often be represented by the highest bar and larger goal counts would log smaller bars. It explains the expected goals about certain Goal Outcomes. Almost all of the opportunities bunch up here at this average point (λ) that we calculated earlier

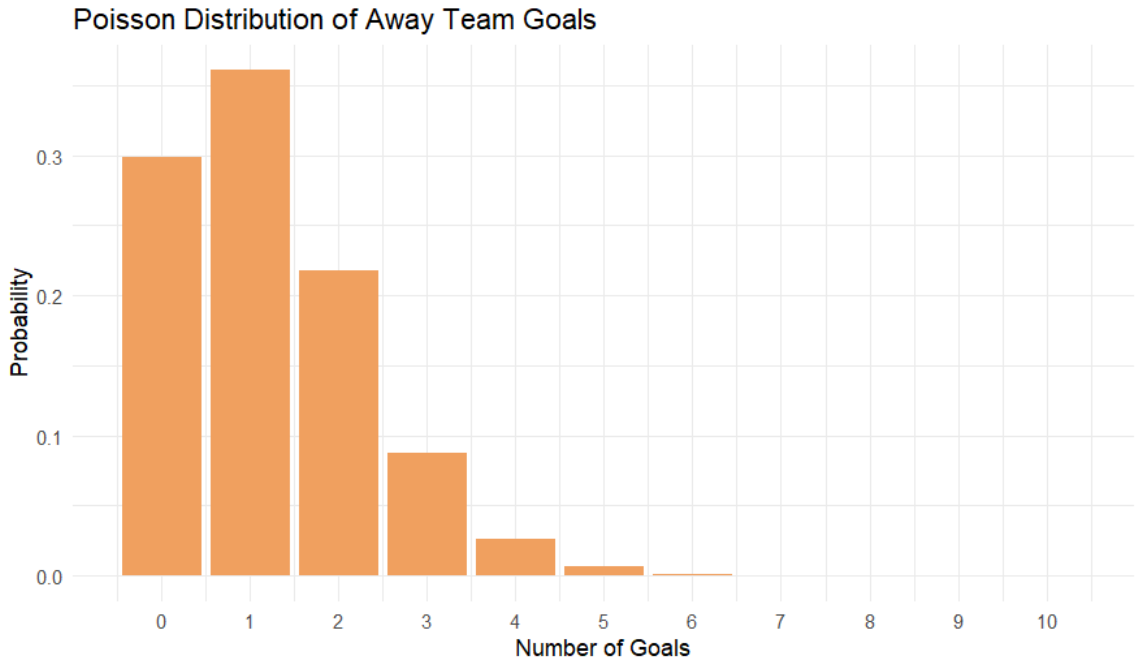

Figure 3.3- Plot between Poisson Distribution of Away Team Goals

Poisson Distribution Plot for Away Team: A similar bar plot is created for the away team's goal distribution.

Interpretation:

- The distribution shows the probability of goals for the away games in all possible values. Most probable number of goals and then to consider how many goals that type of team is likely or less likely to score, given the calculated mean (λ) for the away by simply assuming a Poisson distribution

Naive model Output 2 :

The dataset provided is a comprehensive collection of football match statistics, taken from E0 English Premier League from Kaggle It contains 380 entries (which aligns with the number of matches in a full Premier League season) and 106 columns, each detailing various aspects of the matches. Here's a breakdown of the key columns and their meanings:

	Div	Date	Time	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	R
1	E0	12/09/2020	12:30:00	Fulham	Arsenal	0	3	A	0	1	A	C
2	E0	12/09/2020	15:00:00	Crystal Palace	Southampton	1	0	H	1	0	H	J
3	E0	12/09/2020	17:30:00	Liverpool	Leeds	4	3	H	3	2	H	M
4	E0	12/09/2020	20:00:00	West Ham	Newcastle	0	2	A	0	0	D	S
5	E0	13/09/2020	14:00:00	West Brom	Leicester	0	3	A	0	0	D	A
6	E0	13/09/2020	16:30:00	Tottenham	Everton	0	1	A	0	0	D	M
7	E0	14/09/2020	20:15:00	Brighton	Chelsea	1	3	A	0	1	A	C
8	E0	14/09/2020	18:00:00	Sheffield United	Wolves	0	2	A	0	2	A	M
9	E0	19/09/2020	12:30:00	Everton	West Brom	5	2	H	2	1	H	M

Figure 3.4- An extract of the dataset from E0 English Premier League

This dataset is likely used for detailed match analysis, including predictions based on betting markets and statistical outcomes

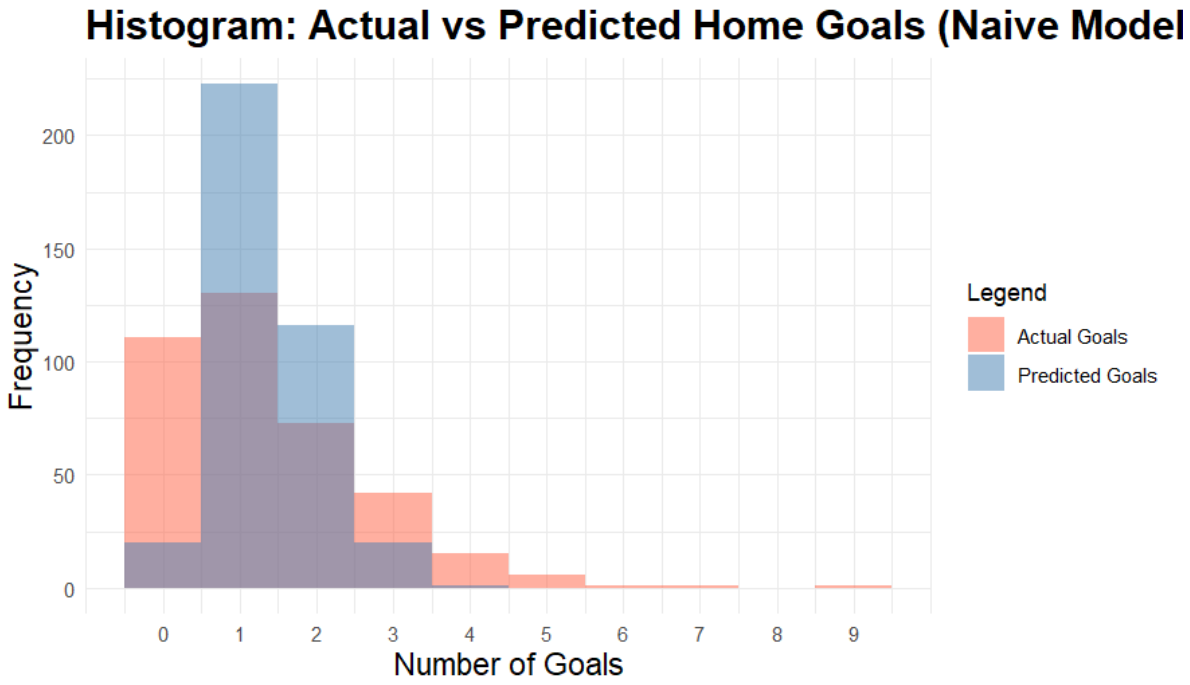

Figure 3.5- Plot between Histogram: Actual vs. Predicted Home Goals (Naive Model)

Histogram: Actual vs. Predicted Home Goals (Naive Model)

- **Purpose**: Fig 3.5 - This above histogram compares the distribution of actual home goals to the predicted home goals of the naive Poisson model. The idea here is to show how well the naive model captures the distribution that real home goals in these matches have taken
- **Explanation**: The histogram splits into two sets of bars that overlap. The red bars reflect the number of goals that the home team has scored in the dataset while the blue bars reflect the number of goals which were predicted for the home team by the naive model.
- In the case of height and spread for red and blue bars standing side by side, we are able to see where the model may perform well and also guess too high or too low on the real number of goals. Where the two patterns line up, that would indicate the model is on target with its prediction of home goals. If they don't match, that's highlighting potential areas to improve upon in this model.

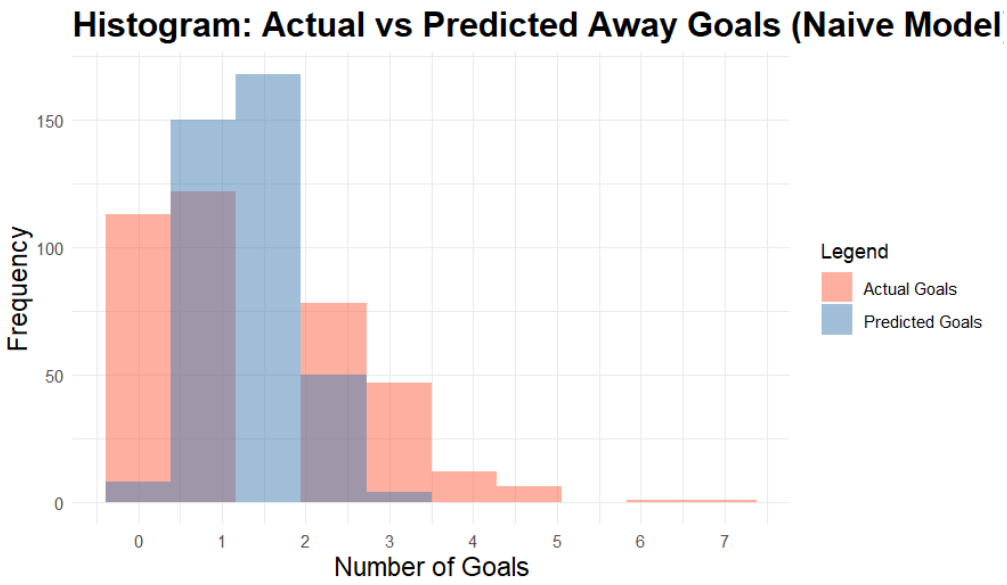

Figure 3.6- Plot between Histogram: Actual vs. Predicted Away Goals (Naive Model)

- **Purpose**: Fig 3.6 - This histogram has the same goal as the first one, but it focuses on away goals. It actually compares how away goals are spread out to how the basic Poisson model thinks they should be. The aim is to check how well the model predicts away goals.
- **Explanation**: The first plot represents the red bars as the real number of scored goals by the away team, while blue is a model prediction number of scored goals.
- This is a clear comparison of how well the model is guessing against what has actually occurred—if the red and blue bars coincide regularly, then the model is good at guessing away goals.

Output Explanation:
MSE for Home Goals: 1.33831908405839

Mean Squared Error (MSE):
The Mean Squared Error (MSE) is calculated for the home goals. It measures the average squared difference between the actual and predicted goals.

MSE for Away Goals: 1.28465149888559

Away Goals MSE:
The Mean Squared Error (MSE) is calculated for the away goals. It measures the average squared difference between the actual and predicted goals.

Chapter 4
Dixon-Coles model

4.1 Introduction:

The Dixon-Coles model is a statistical approach to evaluate team performance and then try to predict football match outcomes. It was introduced by Mark Dixon and Stuart Coles, two statisticians in 1997 by extending the Poisson distribution approach. It believes that the goals scored by teams in whatever soccer game follow the Poisson distribution, and further adds that there are usually dependencies and a factor of time influencing the results of that particular match..

Dixon and Coles showed how such models could be implemented to render the betting market inefficient. To them, inefficiency in the market is achieved when betting odds do not represent real probabilities of the results of the matches.

4.2 Structure of Dixon Coles:

Let $x^k ij$ be the number of goals scored by the home team i against the away team j in match k, $y^k ij$ be the number of goals scored by the away team j on the same occasion and let m be the $y^k ij$ total number of teams considered. In order not to overcomplicate the notation, hereafter x^k (y^k respectively) will be used instead of $x^k ij$ ($y^k ij$ respectively) with the home team (i) and the away (j) team (j) implicitly assigned to every match k First, the two authors define x^k and y^k as follows:

$$\begin{aligned} x_k &\sim \text{Pois}(\lambda_k), \\ y_k &\sim \text{Pois}(\mu_k), \\ \log(\lambda_k) &= \gamma + \alpha_{i(k)} + \beta_{j(k)}, \\ \log(\mu_k) &= \alpha_{j(k)} + \beta_{i(k)}, \end{aligned}$$

with Pois(λ) denoting a Poisson distribution with mean λ, i(k) and j(k)indices which identify the home and away teams playing match k, and where α, β, γ are the attack, defence and home effect parameters, respectively. By considering the matches chronologically, the two authors divide the seasons into a series of half-weekly time points and construct the following function for each time point t:

(1)
$$\mathcal{L}_t(\alpha_i, \beta_i, \rho, \gamma; i = 1, \ldots, m) = \prod_{k \in A_t} \left\{ \tau_{\lambda_k, \mu_k}(x_k, y_k)\, e^{-\lambda_k}\, \lambda_k^{x_k}\, e^{-\mu_k}\, \mu_k^{y_k} \right\}^{e^{-\xi(t - t_k)}},$$
$$A_t = \{k : t_k < t\},$$

with t the time that match k is played, $\tau_{\lambda k,k}(x, y)$ the function depends on the parameter ρ which manages the dependence between x_k and y_k, and ξ ≥ 0 the parameter that regulates the down-weighting of old matches. Consistently with the notation of the original paper, the

(2)
$$\sum_{i=1}^{m} \alpha_i = m$$

included for identifiability.

Hence, the two authors obtain the estimates of the parameters by numerically maximising the function in Equation (1) at each time point t after choosing ξ. The choice of ξ is particularly tough because Equation (1) defines a sequence of non-independent functions that makes it difficult to obtain the value of ξ that maximises the overall predictive capability of the model. To overcome this problem, the two authors focus on the prediction of match outcomes rather than match scores and define the value of ξ as the value

Maximising

(3)
$$S(\xi) = \sum_{k} \left(\delta_k^H \log p_k^H + \delta_k^D \log p_k^D + \delta_k^A \log p_k^A\right)$$

with

(4)
$$p_k^r = \sum_{h,a \in B_r} \mathrm{pr}\left(x_k = h, y_k = a\right),$$

which is implicitly a function of ξ since the score probabilities pr (x_k = h, y_k = a) are estimated from the maximisation of the function in (1) at t_k with the weighting parameter set at ξ and with r = {H,D,A},

- BH = {(h,a):h>a} : representing home wins,
- BD = {(h,a):h=a} : representing draws,
- BA = {(h,a):h<a} : representing away wins.

additionally δ_k^H(δ_k^D,δ_k^A respectively) the delta function equal to 1 if the final result of game k is a home win (draw, away win respectively).

The last, fundamental aspect introduced by Dixon and Coles concerns the description of the dependence structure, which is defined using the function

(5 $T_{\lambda_k,\mu_k}(x.y)$ =

$$\begin{cases} 1-\lambda_k\mu_k\rho & if\ x=0,\ y=0 \\ 1+\lambda_k\rho & if\ x=0,\ y=1 \\ 1+\mu_k\rho & if\ x=1,\ y=0 \\ 1-\rho & if\ x=1,\ y=1 \\ 1 & \text{Otherwise} \end{cases}$$

Subject to

(6) $$\max\left(-\frac{1}{\lambda_k},-\frac{1}{\mu_k}\right) \le \rho \le \min\left(\frac{1}{\lambda_k\mu_k},1\right).$$

By using this function(2), the marginal distributions of x_k and $y_k$are Poisson with means λk and μk respectively and the independence between x_k and y_k is obtained when ρ = 0.

4.3 Mathematical Formulation

The Dixon-Coles model is the calculation of the expected number of goals scored by the home team (λ) and the away team (μ) using this following formulae:

For the home team:

- $\lambda = e^{\alpha H+\beta A+\gamma}$

For the away team:

- $\mu = e^{\alpha A+\beta H}$

Where:

➔ **αH: and αA:** the strengths of attacking for the home team and away team

➔ **βH and αA:** the strengths of defending for the home team and away team

➔ **γ:** represents the home advantage.

The probability of a match ending with a particular scoreline is given by:

(7) $$P(X=x,Y=y)=e^{-\lambda}\frac{\lambda^x}{x!}e^{-\mu}\frac{\mu^y}{y!}\cdot\phi(x,y)$$

Here, φ(x,y) is a factor for the correlation between the goals scored by the home and away teams. The method was first proposed by Dixon and Coles for using φ specifically for purposes of adjustment to the odds on low scores, above all to 0-0, 1-0, 0-1 and 1-1.

4.4 Model Estimation:

The parameters of the Dixon-Coles model are estimated using maximum likelihood estimation (MLE). This involves:

- Defining the likelihood function with respect to the observed match results and the model probability distribution.
- The model's parameters are mostly estimated using historical match data, with a decay factor applied to give more weight to recent matches, ensuring the model reflects the current team form.

Dixon coles coding :

This dataset contains 380 entries and 65 columns, this dataset involves very detailed data regarding matches from the English Premier League for the season 2015-2016. It means complete statistics on the result of each concrete match and various kinds of betting odds. It proves useful to analyse how teams fared, try to predict results, and take a look at what kind of shifts the betting market showed during the 2015–16 season..

Output:

```
Estimated home attack strength: 0.2000922
Estimated away attack strength: 0.09444257
Estimated home defense strength: 0.09444257
Estimated away defense strength: 0.2000922
Estimated rho (correlation): -0.2391318
```

MSE Calculation: The Mean Squared Error is calculated by square of the difference between actual goals and predicted goals, averaging these squared differences.

Overall MSE: This combines the MSEs for home and away goals to give a single measure of prediction error.

```
Mean Squared Error (MSE) of Dixon-Coles model: 0.8730604
```

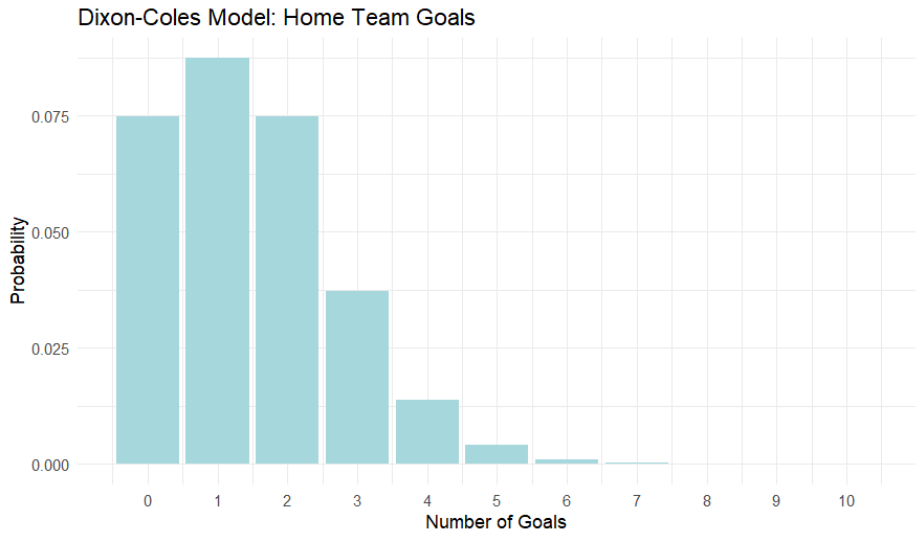

Figure 4.1- Plot of Dixon-Coles Model: Home Team Goals

Dixon-Coles Model: Home Team Goals

- **X-axis (Number of Goals):**This represents the number of games that the home accordingly could win in a game depending on the numbers shown below starting from 0 and leading to 10.
- **Y-axis (Probability):**The probability the home side is to score precisely each of the individual number of goals,according to the Dixon-coles model.
- **Bars:** Each bar represents the number of goals that the home team could score that many goals.

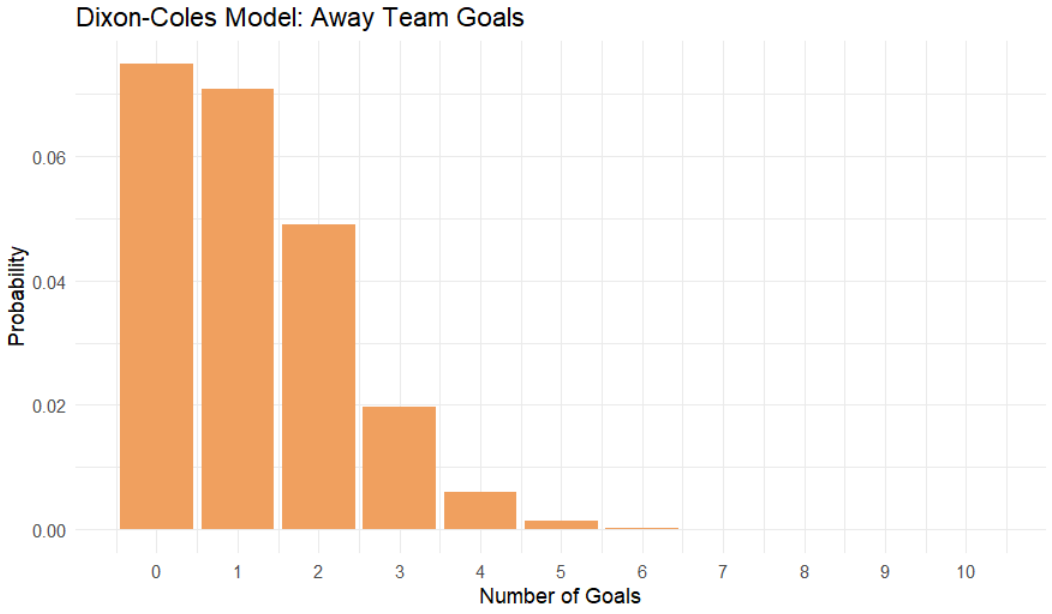

Figure 4.2 - Plot of Dixon-Coles Model: Away Team Goals

Dixon-Coles Model: Away Team Goals

- **X-axis (Number of Goals):** This shows the number of goals that the away team could win in a game going from 0 to 10.
- **Y-axis (Probability):** The probability the away team scores exactly each particular number of goals, based on the Dixon-Coles model.
- **Bars:** Each bar represents the chance that the away team will net a specific number of goals.

Interpretation:

- This graph is similar to the home team's graph. However, it may have a different spread due to how the model has projected the overall performance of the away team.
- Once more, the distribution reflects a Poisson-like pattern that's been adjusted by the Dixon-Coles corrections.

Goal model coding:

The dataset Bundesliga league 2014 contains 24,580 entries and 29 columns related to football statistics, covering multiple leagues and seasons. The dataset covers advanced football metrics like xG, xGA, PPDA, and expected points, providing insights into both offensive and defensive performance. This dataset provides advanced metrics used in modern football analysis to evaluate teams' performance beyond just goals and points.

Expected goals (xG), PPDA, and deep completions help analysts assess how well a team performs in creating chances, defending, and applying pressure during a match. These metrics offer insights into the tactical efficiency and style of play for teams across different leagues and seasons.

	league	year	h_a	xG	xGA	npxG	npxGA	deep	deep_allowed	scored	missed	xp
				column 3: character								
1	Bundesliga	2014	h			2.5701200	1.1984200	5	4	2	1	
2	Bundesliga	2014	a	1.5032800	1.3079500	1.5032800	1.3079500	10	1	1	1	
3	Bundesliga	2014	h	1.2298700	0.3101660	1.2298700	0.3101660	13	3	2	0	
4	Bundesliga	2014	a	1.0351900	0.2031180	1.0351900	0.2031180	6	2	0	0	
5	Bundesliga	2014	h	3.4828600	0.4028440	3.4828600	0.4028440	23	2	4	0	
6	Bundesliga	2014	a	3.4696600	0.8217980	3.4696600	0.8217980	27	0	2	0	
7	Bundesliga	2014	h	2.6987900	0.4431780	2.6987900	0.4431780	14	5	4	0	
8	Bundesliga	2014	h	2.4982600	0.0000000	1.7404900	0.0000000	16	0	6	0	
9	Bundesliga	2014	a	1.2047000	0.6483840	1.2047000	0.6483840	5	7	0	0	
10	Bundesliga	2014	h	2.6436700	0.9428840	1.8860500	0.9428840	11	3	2	1	
11	Bundesliga	2014	a	3.0346700	0.2224470	3.0346700	0.2224470	13	1	4	0	

Figure 4.3 -An extract of the dataset from Bundesliga league 2014

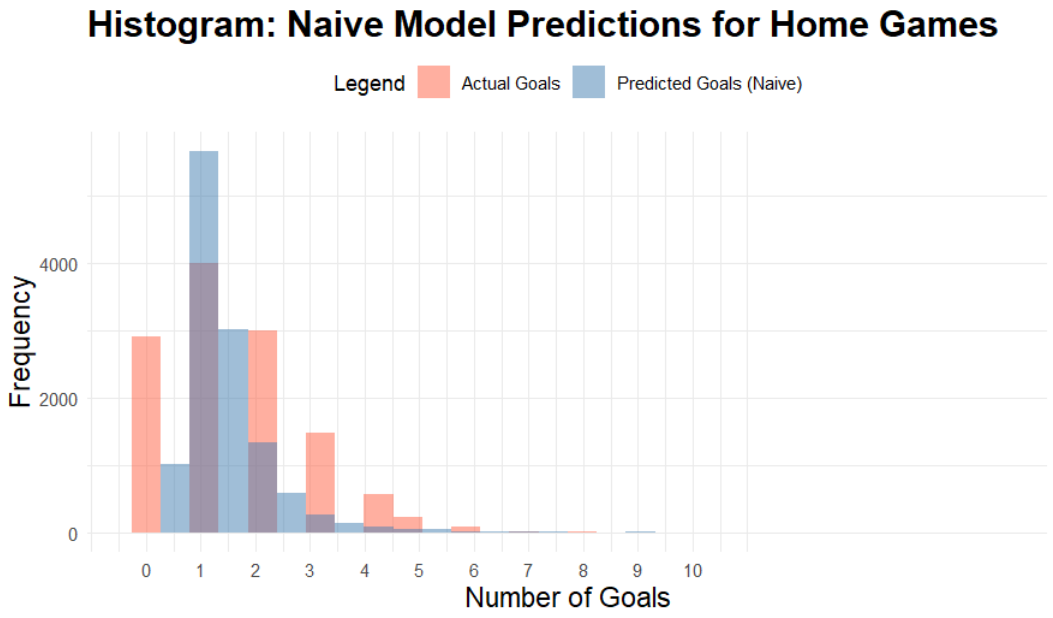

Figure 4.4 Histogram Plot: Naive Model Predictions for Home Games

Plot: Naive Model Predictions for Home Games:

- **Purpose**: Figure 4.4 would show how many times teams scored or were expected to score a certain number of goals in home games, based on a simple prediction method.
- **Blue Area ("Predicted Goals (Naive)")**: This represents the Naive model's goal predictions for the same home games.

Red Area ("Actual Goals"): This shows the goals teams scored during home games.
Actual Goals (Red Bars): Home teams score between 0 and 3 goals most often, with 1 and 2 goals being the most common. Some home teams score 4 or more goals, but this happens less often.

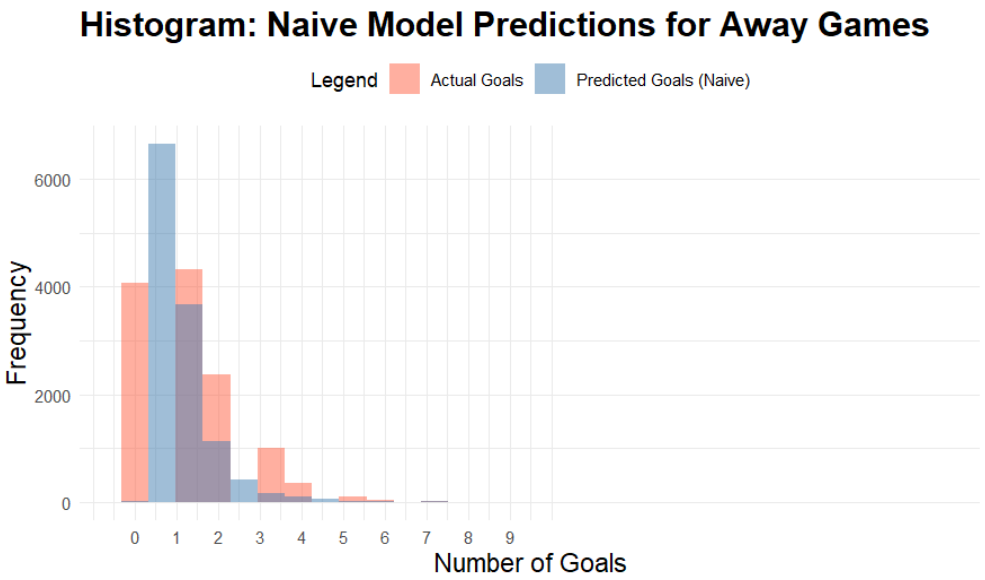

Figure 4.5- Histogram Plot: Naive Model Predictions for Away Games

Purpose: Figure 4.5 would display how often teams scored or were predicted to score a specific number of goals in away games using a basic forecasting approach.

Plot Elements:

Red Area ("Actual Goals"): Shows how likely teams are to score a certain number of goals in away games.

Blue Area ("Predicted Goals (Naive)"): Shows how many goals the Naive model thought teams would score in away games.

Interpretation:

Overlap: To check how good the Naive approach is, look at how the red and blue areas match up for away goals. When both areas grow together, it means the model is closer to what happens.

Difference: When the two areas don't match well, it might mean the Naive model didn't get the goal spread right when it looked at each team on its own.

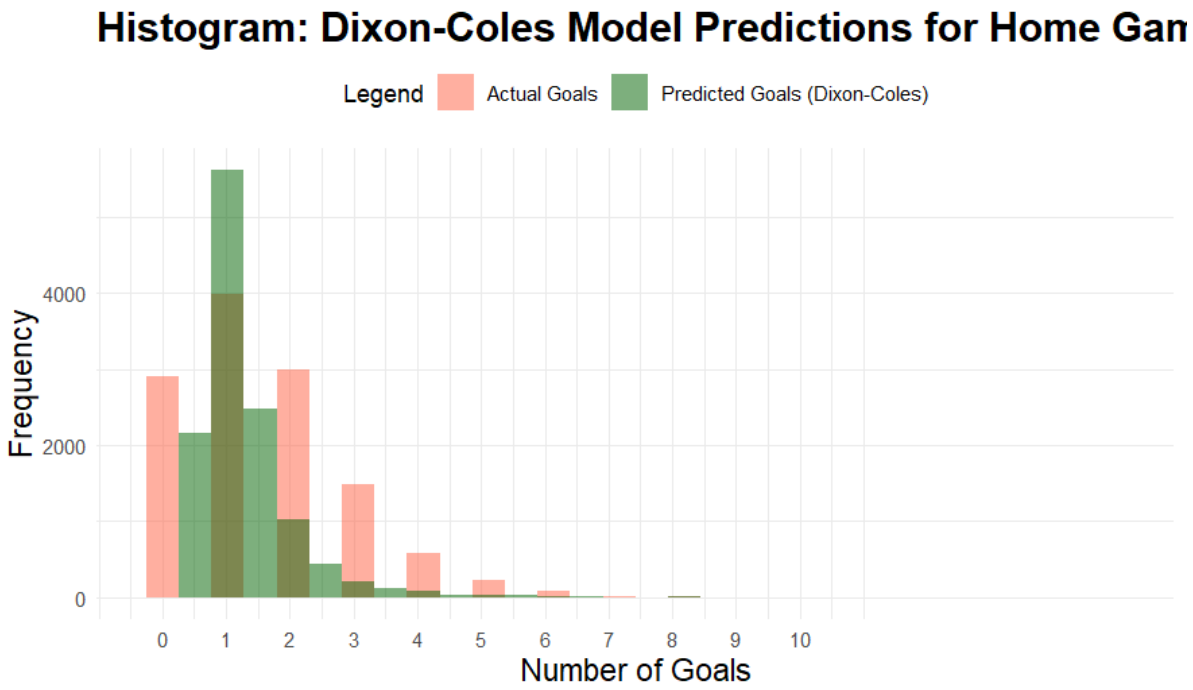

Figure 4.6 - Histogram Plot: Dixon-Coles Model Predictions for Home Games

Purpose: This plot shows the difference between observed Home goal tally and the goal tally as per Dixon-Coles model.

Home Game Predictions:

1. **Actual Goals (Red Bars)**: For home teams, most of the goals scored are between 0 and 3, with a higher concentration around 1 and 2 goals. Some home teams also score 4+ goals, though this happens less frequently.
2. **Predicted Goals (Green Bars)**: The model's predictions align fairly well with the actual distribution of goals from 0 to 3 goals. However, for higher goal counts (4+ goals), the model underpredicts, as it shows a lower frequency of predictions beyond 3 goals.

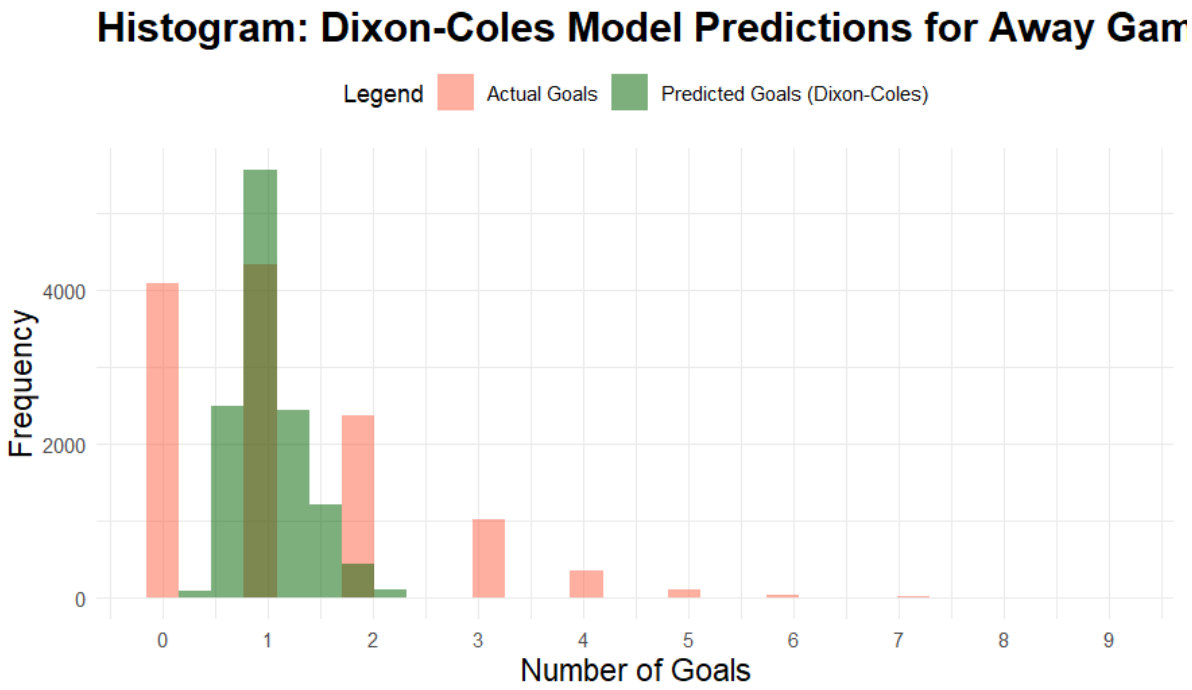

Figure 4.7- Histogram Plot: Dixon-Coles Model Predictions for Away Games

Purpose: This plot shows the difference between observed away goal tally and the goal tally as per Dixon-Coles model.

Away Game Predictions:

1. **Actual Goals (Red Bars)**: The chart shows how many goals away teams score. Most teams get 0 to 2 goals, with peaks at 0 and 1. It's rare for teams to score 3 or more.
2. **Predicted Goals (Green Bars)**: The Dixon-Coles model makes predictions that cluster around 1 goal. These match up well with real scores between 0 and 2. But the model doesn't catch all the times teams score 3 or more, which happens more often than it predicts.

Plot Elements:

Red Area ("Actual Goals"): Shows the real away goals scored. It gives us an idea of how many away goals teams score. The density plot helps us see the pattern of away goals in games.

Green Area ("Predicted Goals (Dixon-Coles)"): Reveals how often different goal counts are predicted. This model comes from Dixon and Coles. It points out how the model's goal predictions lean one way or another.

Interpretation:

Overlap: The darker parts of the map tell us how the Dixon-Coles model guesses away goals. A lot of overlap means the Dixon-Coles model does a good job predicting goals.

Difference: The structure of Area 1 and Area 2 must differ . When Area 1 and Area 2 scores look alike or different, we can guess that the model might need adjustments or doesn't have the right tools to spot conflicts.

The Dixon-Coles model changes how the basic Poisson structure makes predictions. It does this by adding a link between how often home and away teams score.

- **Predicted Goals (Green Bars)**: The model's predictions match up pretty well with the real spread of goals from 0 to 3. But when it comes to higher numbers (4+ goals), the model doesn't predict enough. It shows fewer predictions beyond 3 goals than what happens.

Confusion Matrices:

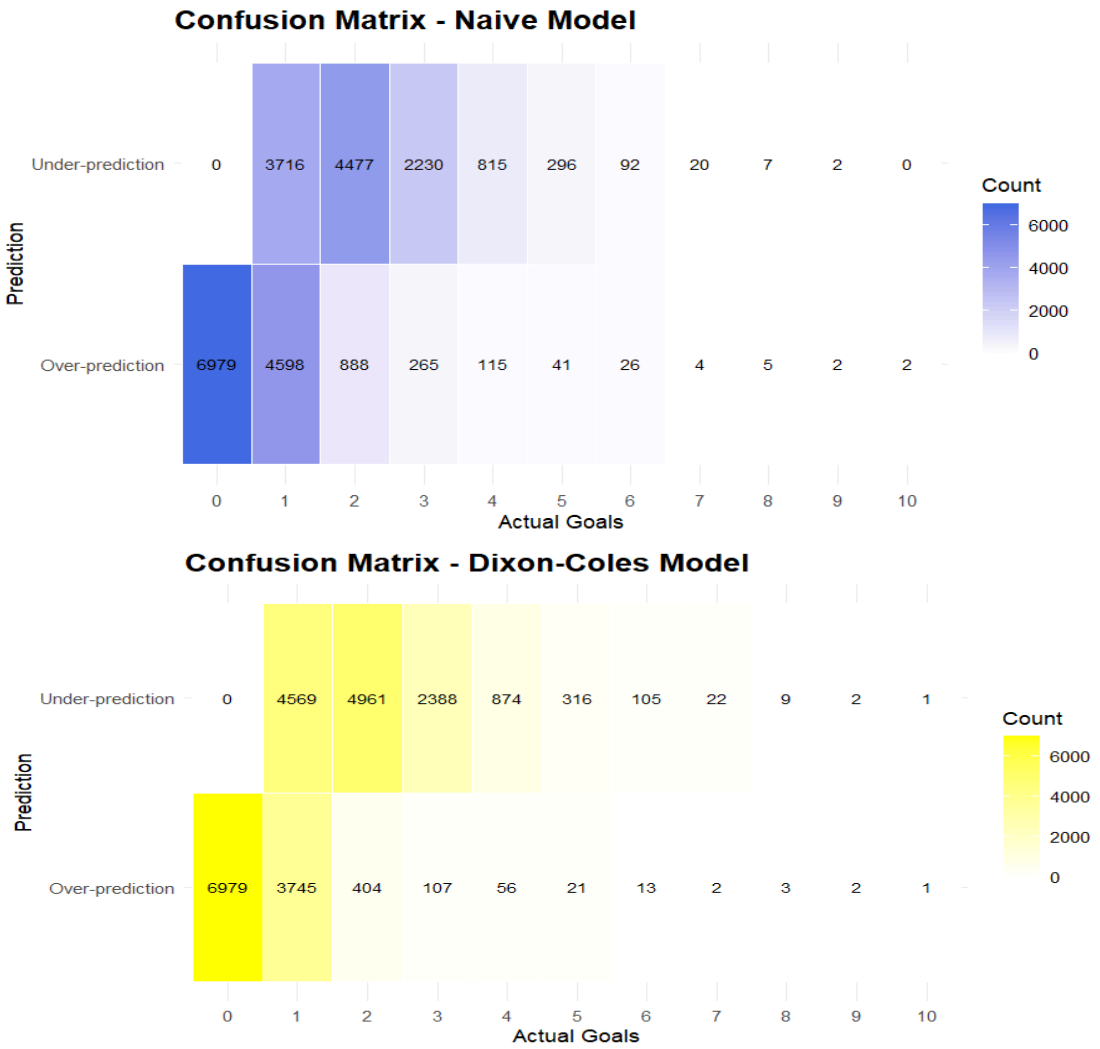

Figure 4.8 - Confusion Matrices for Naive Model and Dixon-Coles Model

Confusion Matrices for Naive :

	Actual										
Prediction	0	1	2	3	4	5	6	7	8	9	10
Over-prediction	6979	4598	888	265	115	41	26	4	5	2	2
Under-prediction	0	3716	4477	2230	815	296	92	20	7	2	0

Confusion Matrices for Dixon :

	Actual										
Prediction	0	1	2	3	4	5	6	7	8	9	10
Over-prediction	6979	3745	404	107	56	21	13	2	3	2	1
Under-prediction	0	4569	4961	2388	874	316	105	22	9	2	1

These tables indicate the extent to which the models have underestimated, given a correct estimation or have even over estimated goals.

Diagonal Tiles (0 Goals for Under-prediction): Look at the Under-prediction row at 0 goals on the X-axis. The value is always 0 here. This makes sense because the model can't predict less than 0 goals.

Over-prediction Counts: The graph displays the frequency with which the model predicted more goals than were scored. For instance, a high count in a tile under "Over-prediction" for 1 actual goal indicates that the model often forecasted more than 1 goal when 1 was netted.

Under-prediction Counts: In the same way, the numbers in the "Under-prediction" section show how many times the model guessed fewer goals than the teams scored.

MSE (Mean Squared Error):

- **Naive Model**: MSE = 0.9989
- **Dixon-Coles Model**: MSE = 1.2078

Counts the mean of the squared difference between the actual and the predicted goals. The higher values of indexes give the possibility to increase the probability of appearance of events, however, the lower values mean better accuracy of distribution.

RMSE (Root Mean Squared Error):

- **Naive Model**: RMSE = 0.9994
- **Dixon-Coles Model**: RMSE = 1.0990

Root Mean Squared Error average of the error values, expressed in the same scale as the observation / Original scale.

MAE (Mean Absolute Error):

- **Naive Model**: MAE = 0.7619
- **Dixon-Coles Model**: MAE = 0.8358

Calculates the magnitude of difference between actual and expected goals by summing the absolute deviations from a mean, thus giving an indication of the extent of prediction inaccuracy.

Interpretation: These are the metrics of models. It is important to note that the lower the values of these metrics, the more accurate is the given model.

Paired t-Test:

Statistical Test Results:

- **t-value: -47.573**
- **Degrees of Freedom (df): 24,579**
- **p-value: < 2.2e-16**
- **Alternative Hypothesis: True mean difference is not equal to 0**
- **95% Confidence Interval: -0.1739501 to -0.1601833**
- **Sample Estimate (Mean Difference): -0.1670667**

T-Test: We used a one sample t-test on the residuals (errors) of the Naive and Dixon-Coles models as the null hypothesis to check for notable differences between the models.

Interpretation:

- The large negative t-value and p-value indicate that chance doesn't explain the difference in how well the models perform.
- The negative mean difference shows that the Dixon-Coles model is more precise on average. Even though the Naive model has better error metrics (MSE, RMSE MAE), the Dixon-Coles model makes predictions that come closer to the actual number of goals scored.
- The paired t-test backs this up revealing that the Dixon-Coles model has smaller residuals on average. This test gives strong proof that the Dixon-Coles model works better to predict soccer match results in this situation.

Chapter 5

Brier Score and Recent player Performance

5.1 Brier Score:

A Brier score is a metric used to measure the accuracy of a prediction. In the context of football goals, it can be used to assess the accuracy of a predicted outcome, such as the probability of a team scoring a certain goal in a match. Brier scores range from 0 to 1, with 0 being perfect accuracy and 1 being worst accuracy.

Predicted outcomes and actual outcomes:
In the case of football goals, this can predict the difference between a team scoring exactly 0, 1, 2 or more goals. The probability is 1 if the event occurs (for example, the team scores the required number of goals).

The formula for the Brier score is:

$$\textbf{Brier score} = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2$$

N is the number of forecasts.

f_i is the forecasted probability for the $i - th$ event.

o_i is the actual outcome for the $i - th$ event, which is 1 if the event occurred and 0 if it did not

5.2 Example Calculation for brier score :

Example Forecasted Probabilities and Actual Outcomes:

If the actual outcome is that the team scores 1 goal, then:

Forecasted Probability (f_i)	Actual Outcome (o_i)
0.80	1
0.60	0

0.70	1
0.50	0
0.90	1

Table 5.1

$(f_1 - o_1)^2$ = $(0.80 - 1)^2$ = 0.04

$(f_2 - o_2)^2$ = $(0.60 - 0)^2$ = 0.36

$(f_3 - o_3)^2 = (0.70 - 1)^2 = 0.09$

$(f_4 - o_4)^2 = (0.50 - 0)^2 = 0.25$

$(f_5 - o_5)^2 = (0.90 - 1)^2 = 0.01$

Sum these squared differences:

Sum = 0.04 + 0.36 + 0.09 + 0.25 + 0.01 = 0.75

B = $\frac{1}{5}$ × 0.75=0.15

Brier Score: 0.15

This Brier score of 0.15 indicates the average squared error between your forecasted probabilities and the actual outcomes. A Brier score close to 0 means your forecasts were generally accurate, while a higher score would suggest less accurate forecasts.

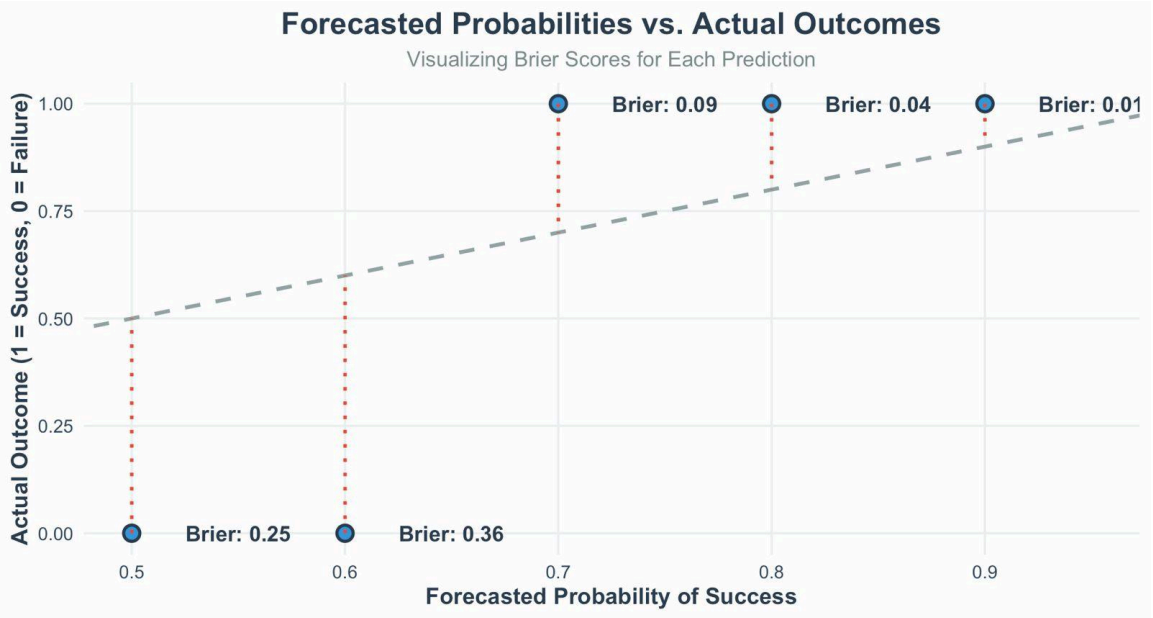

Figure 5.1- Plot between Forecasted Probability of Success vs Actual Outcome

1.Scatter Plot Points:

- X-axis (Forecasted Probability of Success):On the x-axis, each point is the quantified probability that an event (for example, a win or success) will happen.

2. Point Locations:

- Points on the Top of the Plot (Y = 1): These points represent cases where the event actually occurred. The further the point to the right which means closer to 1. 0 on the x-axis.
- Points on the Bottom of the Plot (Y = 0): These points represent cases where the event did not occur. The further the point is to the left (closer to 0.0 on the x-axis), the lower the forecasted probability of the event occurring.

3. Brier Score:

- Brier Score Annotation: Next to each point, the figure illustrates the Brier score of the given prediction. The Brier score is calculated as the squared difference between the predicted probability and the actual outcome.
- Lower Brier Scores (Closer to 0): Indicate better predictions. The predicted probability was close to the actual outcome.
- Higher Brier Scores (Closer to 1): Indicate worse predictions. The predicted probability was far from the actual outcome.

Interpretation Summary:

- Prediction Accuracy: The Brier score provides a measure of the accuracy of prediction. Lower scores mean the predictions were closer to reality. The scatter plot correlates these predictions with actual outcomes.

5.3 Recent performance of player:

- Predicting football goals based on a player's recent performance involves analysing several key factors that can influence their scoring ability
- This approach integrates both statistical modelling techniques and domain knowledge to provide insights into player performance and goal-scoring potential in football matches
- Each player's recent performance can be considered to build the model. A more precise estimation shall be achieved by considering the player's recent performance as an additional parameter.

The dataset appears to be a collection of performance statistics for the **2021-2022** Premier League season. It consists of 537 entries and 11 columns. The dataset provides a comprehensive view of player performance based on actual and expected metrics in the **2021-2022** Premier League season

Filter

	№	Player	Team	Apps	Min	G	A	xG	xA	xG90
1	1	Son Heung-Min	Tottenham	35	3051	23	7	16.99-6.01	7.85+0.85	0.50
2	2	Mohamed Salah	Liverpool	35	2757	23	13	24.36+1.36	9.79-3.21	0.80
3	3	Cristiano Ronaldo	Manchester United	30	2468	18	3	17.21-0.79	4.42+1.42	0.63
4	4	Harry Kane	Tottenham	37	3229	17	9	20.69+3.69	9.82+0.82	0.58
5	5	Sadio Mané	Liverpool	34	2833	16	2	16.83+0.83	5.27+3.27	0.53
6	6	Kevin De-Bruyne	Manchester City	30	2214	15	8	5.95-9.05	11.26+3.26	0.24
7	7	Jamie Vardy	Leicester	25	1807	15	2	9.99-5.01	1.61-0.39	0.50
8	8	Diogo Jota	Liverpool	35	2401	15	4	17.35+2.35	5.87+1.87	0.65
9	9	Wilfried Zaha	Crystal Palace	33	2762	14	1	10.18-3.82	4.89+3.89	0.33
10	10	Raheem Sterling	Manchester City	30	2119	13	5	15.78+2.78	5.02+0.02	0.67
11	11	Ivan Toney	Brentford	33	2905	12	5	12.05+0.05	7.34+2.34	0.37
12	12	Jarrod Bowen	West Ham	36	3003	12	10	12.93+0.93	5.82-4.18	0.39

Figure 5.2 - An extract of dataset from 2021-2022 Premier League season

Output:

Display the cleaned data:

```
A tibble: 6 × 11
    `№` Player           Team              Apps   Min      G     A    xG    xA  xG90  xA90
  <dbl> <chr>            <chr>            <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1     1 Son Heung-Min    Tottenham           35  3051     23     7 17.0   7.85  0.5   0.23
2     2 Mohamed Salah    Liverpool           35  2757     23    13 24.4   9.79  0.8   0.32
3     3 Cristiano Ronaldo Manchester United  30  2468     18     3 17.2   4.42  0.63  0.16
4     4 Harry Kane       Tottenham           37  3229     17     9 20.7   9.82  0.58  0.27
5     5 Sadio Mané       Liverpool           34  2833     16     2 16.8   5.27  0.53  0.17
6     6 Kevin De-Bruyne  Manchester City     30  2214     15     8  5.95 11.3   0.24  0.46
```

Random Forest model to predict goals:

```
randomForest(formula = G ~ xG90 + xA90 + Min + Apps, data = dataTrain,      ntree
= 100)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 1

          Mean of squared residuals: 2.841914
                    % Var explained: 74.09
```

Predictions:

	Player	Team	G	predicted_goals
	<chr>	<chr>	<dbl>	<dbl>
1	Son Heung-Min	Tottenham	23	12.5
2	Jarrod Bowen	West Ham	12	10.1
3	James Maddison	Leicester	12	8.28
4	Neal Maupay	Brighton	8	6.65
5	Martin Odegaard	Arsenal	7	6.23
6	Danny Welbeck	Brighton	6	6.12

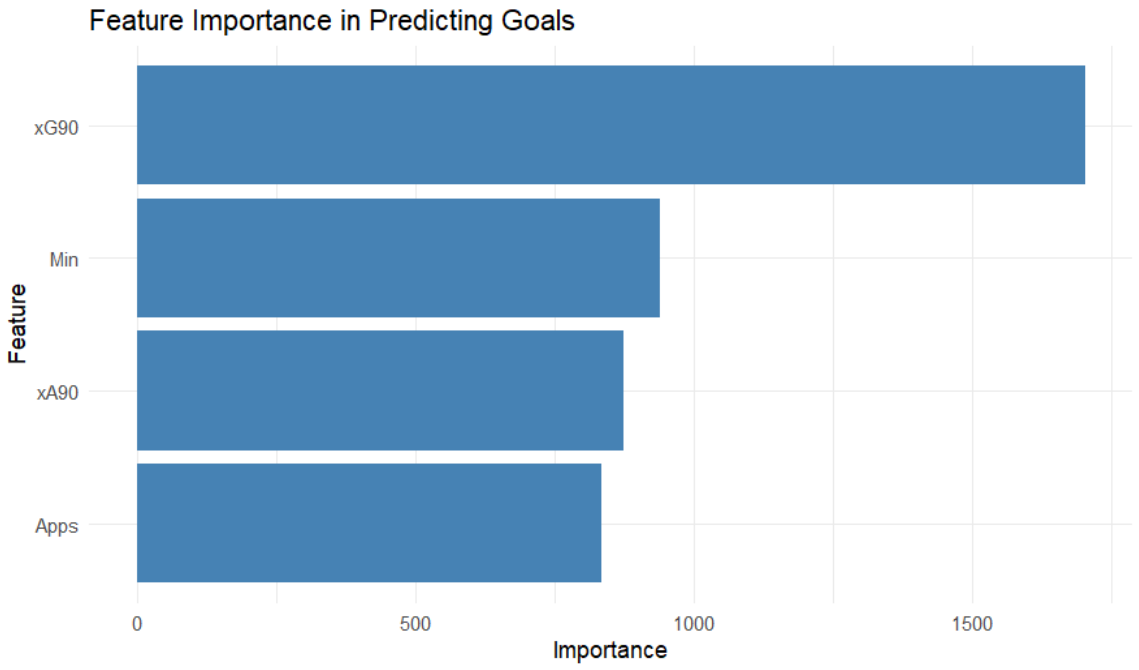

Figure 5.3 - Plot for Feature importance in predicting Goals

To rank the importance of different features (xG90, xA90, Min, and Apps) in predicting the number of goals. The plot is a feature importance chart showing how much each feature contributes to the prediction of goals in the Random Forest model. xG90 is the most important feature for predicting goals, which aligns with expectations as

it directly represents the expected goals Min (minutes played) is the second most important feature, followed by xA90 and Apps.

This ranking indicates that while xG90 is the dominant predictor, other features also play significant roles in determining the number of goals a player is expected to score.

xG90 stands for Expected Goals per 90 minutes. xG90 provides a normalised measure of a player's goal-scoring opportunities, taking into account the time they spent on the field. It helps compare players' effectiveness and efficiency in generating scoring chances, regardless of how many minutes they actually played

Plot Explanation:

X-Axis (Feature): Provides the details on the executed Random Forest features among them being xG90, xA90, Min, and Apps.

Y-Axis (Importance): Displays the 'importance' of each feature whereby the figure represents how many marks each feature receives in helping the model make decisions.

Bars: The length of the bars are proportional to the importance of the feature. When a bar is longer it is usually an indication that the particular feature is more effective in making the prediction.

If xG90 has the highest importance score, it means that the expected goals per 90 minutes is the most critical predictor for goals scored. This insight might suggest that a player with a high xG90 is likely to score more goals, making this a key metric for evaluating a player's offensive capability.

Plot :

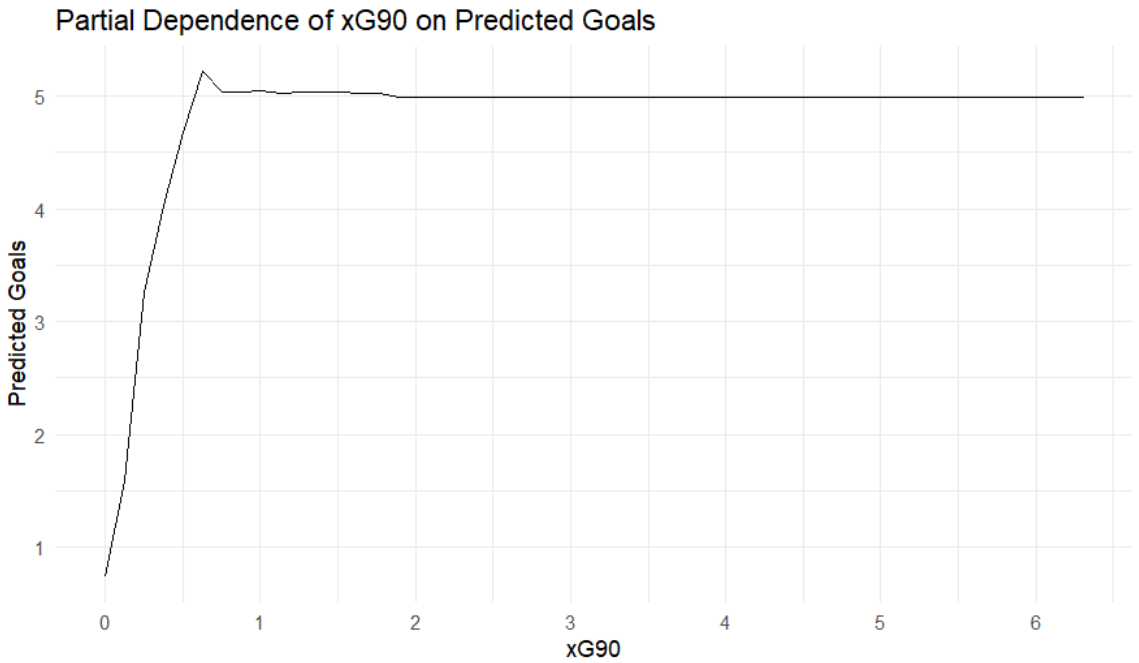

Figure 5.4- Plot for Partial Dependence of xG90 on predicting goals

Explanation:

This plot helps you understand how changes in `xG90` affect the predicted number of goals, holding other features constant. The curve rises steeply at first, indicating that an increase in `xG90` from a low value significantly increases the predicted goals.

We can observe from the plot that, as xG90 goes up, for the model, the predicted goals quickly shoot upward to about 5 and level off. The saturation arises because it is believed by the model either to be rare, or unlikely, to score more than 5 goals and hence does not predict larger numbers for increased xG90. This points to one beyond which the number of goals predicted does not grow substantially with an increase in xG90.

X-Axis (xG90): Symbolises various degrees of measurement for xG90.

Y-Axis (Predicted Goals): Makes use of a graph to represent the predicted number of goals where xG90 is the changing factor.

Curve: The top graph shows the curve of xG90 against the number of predicted goals. For instance, the positive slope means that when xG90 rises, the projected goals also rise.

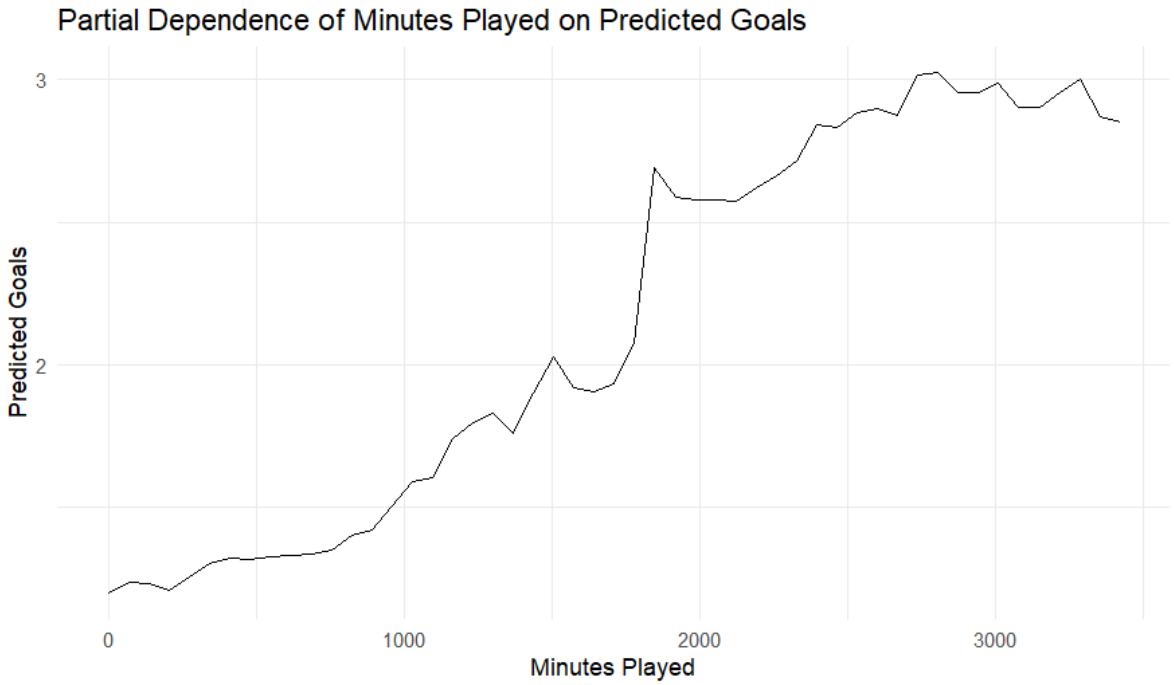

Figure 5.5 - Plot for Partial Dependence of Minutes played on predicting goals

This PDP shows how the number of minutes a player has played (`Min`) affects the predicted number of goals. Understanding this relationship can provide insights into how playing time influences a player's goal-scoring ability.

Plot Explanation:

X-axis (minutes played): It describes the time spent in minutes concerning both teams' or a single player's time used in the game.

Y-Axis (Predicted Goals): If the points record were to tell how many goals are expected while the game is ongoing, it would extrapolate the number of goals that ought to have been scored according to the number of minutes played.

Curve: The curve reflects the relationship between the amounts of playing time and a given number of goals that were to be predicted. Given a curve whose slope has an upward inclination, it will read that in most cases, when more minutes are played, a higher predicted goal is scored.

Positive Correlation: If the graph goes uphill, this means that the more time a player spends on the field, so do his expected goals. This seems quite logical since the more time a player stays on the field, the more scoring opportunities arise.

Feature importance plot: it allows you to find out which one of the features is most important to predict the goals.

Partial Dependence Plots: This view on the rest of the variables will help in predicting the outcome, through the coefficients of xG90 and minimum. It gives a view of how those changes assist in helping degree gainers understand the predictive power of the Random Forest model.

Conclusion :

Predicting Future Success: The trends of player performance measured currently can help predict in what regard they might result in future success. An instance would be a player who constantly records high xG90, and as playing time increases, this player is surely going to start scoring more goals.

By identifying key players, one means understanding which numbers-like xG90 and minutes played-drive the goal predictions. These are what are used by the teams to identify which players are in form, therefore likely to contribute mostly in future games. At some point, this could be useful even when one needs to make strategic decisions, say, whom to field in a crucial game.

Recent statistics are going to be used in making estimates of current form of players: the analysis calculates the players most likely to score. Additionally, it will give room for teams and analysts to make intelligent choices of players and follow ways of play accordingly.

Chapter 6
Practical Applications and Key Differences

6.1 Practical Applications:

● **Betting Markets:** In betting markets, the Dixon-Coles model is used widely as…Since they make their predictions based on the factors such as team abilities, home ground bias, undefined Bookmakers rely on it to give out the odds while the bettors rely on it to set the odds. Intelligence that identifies the available potential wins and make right decisions about how to cooperate sufficiently. In the long run, will therefore help in enhancing profitability and effectiveness of the market.

● **Odds Setting:** The Dixon-Coles model is particularly important to bookmakers in establishing the correct market price of the events, based on the exact probability on the victory, tie or defeat based on the match result.

● **Betting Strategies:** Therefore, using the probabilities that the Dixon-Coles model spits out, bettors focus on value bets. explaining what options can be used in the cases when the offer set by the bookmaker has a higher EV than the model's EV. contrasts, which in its turn assists in increasing the predicted odd in profit-making affairs.

● **Team Performance Analysis:** Football clubs and analysts have used the Dixon-Coles model to evaluate teams in an undefined way This way, they can get information on how for instance, Relative strengths, weaknesses and trends in performance are defined and assist in the matter of strategic cohesion and concretely general team results and performance.

● **Performance Metrics :** The given model is quite helpful in deciding certain aspects that are more effective in the team or need a lot of improvement for the coaches and analysts to take into consideration on the side of their team.

● **Opponent Analysis:** The idea here is that through reconnaissance, teams can gain vital information on the attacks and defence of rivals.

● **Fantasy Sports:** The model is used in fantasy football leagues to predict the players or Their performance during the season. coordinate and schedule bios and transfers that are needed in teams and leagues

● **Incorporating Player-Level Data:** Among these extensions, one that can be regarded as more general is connected with micro-level data increasing of the Dixon-Coles model and reflecting players' performance indicators and their physical conditions.

● **Player Impact:** Other features that can be incorporated in order to improve the readability of the model are the player form, players' injuries, and even afa suspensions.

● **Dynamic Home Advantage:** Thus it has been postulated that home advantage may be fixed or fluctuating to a greater extent between two teams. Outlined below are the dynamic models for home advantage which change the home advantage factor by considering certain factors such as the crowd, the distance covered, and the level of familiarity of the team with the ground undefined. This can enhance the making of the forecast through the integration of machine learning to the model because the model may contain some features that enable the making of decisions based on the pattern and relations in the data.

● **Machine Learning Integration:** Having additional features such as the weather or the referee's inclinations, and the teams.
● **Feature Engineering:** Using advanced features like weather conditions, referee tendencies, and team fatigue.
● **Model Ensembling:** Combining the Dixon-Coles model with other predictive models to improve overall accuracy.

Limitations:
• **Complexity**: Estimation of the parameters and carrying out of the entire model as proposed by Dixon-coles requires a lot understanding in statistical methods and computational power.

• **Data Dependence:** The accuracy of the model strictly depends upon the quality, quantity and detail of data used in the model. This implies that higher precision in parameter estimates and improved prediction accuracy are obtained using accurate and adequate databases.

• **Static Parameters:** Some of the parameters like home advantage are fixed without necessarily changing them when from match to match and/or game to game or from season to season. Static approach may not always capture the changes that may occur in the actual world and thus influence the performance of the model in dynamic environments.

Further Research Directions:

● **Incorporating Player-Level Data:** Including player-level performance metrics could refine the model's predictions by accounting for the impact of key players and injuries on match outcomes. Future research could focus on developing methods to integrate these data seamlessly into the model.

● **Application to Other Sports:** Examining the generalization of Dixon-Coles model to other sports might expand its uses and shed light on its principles. It is therefore possible to extend the model into nuevo, new inventions in the system of sports analytics due to the corresponding distinct patterns of scoring and the rules of the game within various sports.

● **Real-Time Prediction Updates:** Introducing techniques of dynamic appendage of the predictions of various matches that may take place in the future could as matches continue to advance, offer live indications about what possibly could happen in matches. This could be particularly useful in in-play betting and live match statistics, for example, the number of corners, free-kicks, fouls among others.

6.2 Methodology of Dixon Coles model:

● **Data Collection:** Data for the Dixon-Coles model typically includes historical match results, team performance metrics, player statistics, and contextual factors such as home advantage and recent form. Data sources may include sports databases, official league records, and advanced analytics platforms.

● **Model Implementation:** Implementing the Dixon-Coles model comprises several key steps:

i. Data Preparation: Cleaning and preprocessing historical match data, ensuring consistency and reliability for subsequent analysis.

ii. Parameter Estimation: Employ maximum likelihood estimation techniques to determine the model's parameters, such as attack and defence strengths, home advantage, and correlation parameters.

iii. Model Validation: Assessing the model's predictive accuracy through rigorous validation processes, including out-of-sample validation techniques, to ensure robust performance in real-world scenarios.

● **Evaluation Metrics:**

The Dixon-Coles model's performance evaluation encompasses various metrics,
Including:

• **Predictive Accuracy:** Evaluating the alignment between the model's predictions and actual match outcomes.

• **Log-Likelihood:** Assessing the goodness-of-fit of the model to the observed data.

• **Brier Score:** Measuring the accuracy of probability predictions, with lower scores indicating better performance.

• **Hit Rate:** Determining the proportion of matches where the predicted outcome aligns with the actual result, providing a measure of the model's effectiveness in forecasting match outcomes.

6.3 Key Differences Between the Naive Poisson Model and Dixon-Coles Model:

Aspects	Naive model	Dixon coles model
Assumptions	• Goals scored by each team are independent and follow a Poisson distribution with a fixed rate (λ)	• Adjusts Poisson distribution to account for the correlation between the scores of the two teams particularly in low-scoring matches.
Correlation	• No correlation between the scores of the two teams	• Accounts for correlation between scores, particularly in low-scoring games like 0-0,0-1,1-0 and 1-1.
Strength	• Simple and easy to implement • Requires minimal data (just average scored and conceded)	• More accurate in predicting actual match outcomes due to correlation adjustment. • Accounts for real-world dependencies between the team's scores
Weaknesses	• Does Not account for the correlation between scores. • Less accurate, especially for low-scoring outcomes.	• More complex to implement. • Requires estimation of additional parameters (eg., Correlation factor (ρ))
Real-World Application	• Useful for basic modelling and quick predictions. • Suitable when data or computational resources are limited.	• Preferred for more accurate and realistic predictions, especially in sports betting and detailed match analysis.

Table 6.1 : Comparative Analysis: Numerous studies have compared the performance of the naive Poisson model and the Dixon-Coles model

CHAPTER 7
Conclusion

7.1 Interpretation 1:

The comparison of dixon coles and naive model for the coding 1 and coding 5 in appendix

Naive Poisson Model:

MSE for home goals: 1.581517
MSE for away goals: 1.312043
Overall MSE: 1.44678

Dixon-Coles Model:

Mean Squared Error (MSE) of Dixon-Coles model: 0.8730604

2. Comparison and Interpretation:

Naive Poisson Model:

The MSE of Naive Poisson model 1.44678

Similarly, the MSE of 1.44678 gives a slightly larger average squared error for the prediction of goals.

Dixon-Coles Model: The Dixon-Coles model minimised the MSE 0.8730604. This lower MSE value specifies that the model's predictions are closer to the actual observed values compared to the Naive Poisson model.

3. Interpretation:

Improvement in Prediction Accuracy: The Dixon-Coles model exceeds the Naive Poisson model, proved by the lower MSE values for both home and away goals. The reduction in MSE shows that the Dixon-Coles model's predictions are more accurate.

Why Dixon-Coles is Better: The Dixon-Coles model, able to provide the correlation between home and away goals, particularly in low-scoring games, results in better predictive performance. This model better grabs the interaction between the teams' performance, probably leading to more accurate predictions.

Overall Conclusion:

Dixon-Coles Model Superiority: The Dixon-Coles model, with lower MSE values, is important in predicting the number of goals in a football match. This kind of improvement in accuracy is particularly essential for low-scoring games, where the Naive Poisson model fails to provide for the correlation between home and away goals.

7.2 Interpretation 2:

The comparison of dixon coles and naive model for the coding 6 in appendix, which give comparison in full coding using goal model prediction

Model Comparison Using Error Metrics

Mean Squared Error (MSE):

- **Naive Model**: Mean Square Error = 0.9989
- **Dixon-Coles Model**: Mean Square Error = 1.2078
- **Interpretation**: Mean Square Error measures the average squared difference between the actual and predicted goals. A lower Mean Square Error indicates better performance. Here, the Naive model has a lower MSE, suggesting it might be slightly better at minimising large errors.

Root Mean Squared Error (RMSE):

- **Naive Model**: Root Mean Squared Error = 0.9994
- **Dixon-Coles Model**: Root Mean Squared Error = 1.0990
- **Interpretation**: Root Mean Squared Error is the square root of Mean Squared Error and provides an error metric in the same units as the data (goals). The Naive model has a lower Root Mean Squared Error, which aligns with the Mean Squared Error findings.

Mean Absolute Error (MAE):

- **Naive Model**: Mean Absolute Error = 0.7619
- **Dixon-Coles Mod6el**: Mean Absolute Error = 0.8358a
- **Interpretation**: Mean Absolute Error measures the average absolute difference between actual and predicted goals. Lower Mean Absolute Error suggests better prediction accuracy. Again, the Naive model appears to be better according to this metric.

2. Residual Analysis and Paired t-Test

Residuals:

- Residuals are the differences between the actual and predicted goals.
- **Naive Model Residuals**: More negative on average, meaning the Naive model appears to over-predict goals slightly more often.
- **Dixon-Coles Model Residuals**: Close to zero on average, meaning it predicts goals more accurately in terms of proximity to actual values.

Paired t-Test:

- **Objective**: To determine if the average difference of residuals between the two models is statistically significant.

Results:

- t-value = -47.573 (a large negative value indicating a substantial difference between the models)
- p-value < 2.2e-16 (extremely small, indicating the difference is highly statistically significant)
- **Mean Difference**: -0.1671, showing that on average, the Dixon-Coles model's residuals are closer to zero than those of the Naive model.
- **Confidence Interval**: [-0.17395, -0.16018], confirming that the Dixon-Coles model consistently provides smaller residuals.

3. Final Interpretation

Naive Model:

- Despite having slightly better MSE, RMSE, and MAE metrics, this model tends to over-predict goals more often, leading to larger, more consistent errors.

Dixon-Coles Model:

- Although it shows slightly higher error metrics, it tends to make predictions that are generally closer to the actual number of goals scored. This is evidenced by the paired t-test, which shows that the Dixon-Coles model has smaller residuals on average.

Conclusion:

- The Dixon-Coles model is statistically superior because it provides more precise predictions in a compatible manner. While using the Naive model there were large errors that are why there is a better MSE, RMSE, MAE but the Dixon-Coles model is more accurate in the number of predicted goals based on the residuals that are closer to zero.

○ This detailed analysis reveals that apart from MSE, RMSE and MAE, we have to consider the distribution of the residuals and stability of the models. Further, the results of the paired t-test indicate fairly convincingly in support of the proposition that the Dixon-Coles model is more appropriate for Soccer match prediction in this regard.

○ Therefore, conclude that from both the interpretations dixon coles model is better than the naive model in the prediction of football goals because it provides the exact prediction.

Chapter 8
Appendix

Implementation of the Naive Poisson Model in R

Appendix 1: :

```
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(MASS)
library(readr)

# Load the dataset
result <- read_csv("C:/Users/ADMIN/Downloads/result.csv")

home_model_naive <- glm(FTHG ~ HomeTeam + AwayTeam, data = result, family = poisson())

away_model_naive <- glm(FTAG ~ HomeTeam + AwayTeam, data = result, family = poisson())

result$predicted_home_goals_naive <- predict(home_model_naive, type = "response")

result$predicted_away_goals_naive <- predict(away_model_naive, type = "response")

# Calculate Mean Squared Error (MSE) for Home Goals (Naive Model)
home_mse_naive <- mean((result$FTHG - result$predicted_home_goals_naive)^2, na.rm = TRUE)

# Calculate Mean Squared Error (MSE) for Away Goals (Naive Model)
away_mse_naive <- mean((result$FTAG - result$predicted_away_goals_naive)^2, na.rm = TRUE)

# Print the MSE values
print(paste("Naive Model MSE for Home Goals:", home_mse_naive))
print(paste("Naive Model MSE for Away Goals:", away_mse_naive))



ggplot(result) +
  geom_histogram(aes(x = FTHG, fill = "Actual Goals"), bins = 10, alpha = 0.5, position =
"identity") +
  geom_histogram(aes(x = predicted_home_goals_naive, fill = "Predicted Goals"), bins = 10,
alpha = 0.5, position = "identity") +
```

```
  scale_fill_manual(values = c("Actual Goals" = "#FF6347", "Predicted Goals" = "#4682B4")) +
  labs(title = "Histogram: Actual vs Predicted Home Goals (Naive Model)",
       x = "Number of Goals",
       y = "Frequency",
       fill = "Legend") +
  scale_x_continuous(breaks = seq(0, max(result$FTHG, na.rm = TRUE), by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 18, face = "bold"),
        axis.title = element_text(size = 14))

ggplot(result) +
   geom_histogram(aes(x = FTAG, fill = "Actual Goals"), bins = 10, alpha = 0.5, position =
"identity") +
   geom_histogram(aes(x = predicted_away_goals_naive, fill = "Predicted Goals"), bins = 10,
alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("Actual Goals" = "#FF6347", "Predicted Goals" = "#4682B4")) +
  labs(title = "Histogram: Actual vs Predicted Away Goals (Naive Model)",
       x = "Number of Goals",
       y = "Frequency",
       fill = "Legend") +
  scale_x_continuous(breaks = seq(0, max(result$FTAG, na.rm = TRUE), by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 18, face = "bold"),
        axis.title = element_text(size = 14))
```

Appendix 2 :

```
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
X2015_16 <- read_csv("C:/Users/ADMIN/Downloads/2015-16.csv")
View(X2015_16)

head(X2015_16)

X2015_16 <- X2015_16 %>%
  rename(home_goals = FTHG, away_goals = FTAG)
```

```
avg_home_goals <- mean(X2015_16$home_goals, na.rm = TRUE)
avg_away_goals <- mean(X2015_16$away_goals, na.rm = TRUE)

cat("Average home goals (λ):", avg_home_goals, "\n")
cat("Average away goals (λ):", avg_away_goals, "\n")

home_model <- glm(home_goals ~ 1, family = poisson(link = "log"), data =
X2015_16)
away_model <- glm(away_goals ~ 1, family = poisson(link = "log"), data =
X2015_16)

lambda_home <- exp(coef(home_model))
lambda_away <- exp(coef(away_model))

# Print the estimated lambda values
cat("Estimated λ for home goals:", lambda_home, "\n")
cat("Estimated λ for away goals:", lambda_away, "\n")

predict_goals <- function(lambda, goals) {
  dpois(goals, lambda)
}

prob_home_2_goals <- predict_goals(lambda_home, 2)
cat("Probability of home team scoring exactly 2 goals:", prob_home_2_goals, "\n")

prob_away_1_goal <- predict_goals(lambda_away, 1)
cat("Probability of away team scoring exactly 1 goal:", prob_away_1_goal, "\n")

avg_goals_data <- data.frame(
  Team = c("Home", "Away"),
  Avg_Goals = c(avg_home_goals, avg_away_goals)
)

ggplot(avg_goals_data, aes(x = Team, y = Avg_Goals, fill = Team)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
```

```
  labs(title = "Average Goals Scored by Home and Away Teams",
       y = "Average Goals", x = "Team") +
  scale_fill_manual(values = c("Home" = "#A8DADC", "Away" = "#F4A261")) +  # Mild
pastel colors
  scale_x_discrete(limits = avg_goals_data$Team) +
  scale_y_continuous(breaks = seq(0, max(avg_goals_data$Avg_Goals), by = 1))  #
Y-axis scale with increments of 1

home_goal_range <- 0:10
home_goal_prob <- predict_goals(lambda_home, home_goal_range)
home_goal_data <- data.frame(
  Goals = home_goal_range,
  Probability = home_goal_prob
)

ggplot(home_goal_data, aes(x = Goals, y = Probability)) +
  geom_bar(stat = "identity", fill = "#A8DADC") + # Mild pastel color
  theme_minimal() +
  labs(title = "Poisson Distribution of Home Team Goals",
       y = "Probability", x = "Number of Goals") +
  scale_x_continuous(breaks = seq(0, 10, by = 1)) # x-axis scale of 1

away_goal_range <- 0:10
away_goal_prob <- predict_goals(lambda_away, away_goal_range)

away_goal_data <- data.frame(
  Goals = away_goal_range,
  Probability = away_goal_prob
)

X2015_16 <- X2015_16 %>%
  mutate(
    predicted_home_goals = lambda_home,
    predicted_away_goals = lambda_away
  )

# Calculate the Mean Squared Error (MSE) for home goals
```

```
mse_home <- mean((X2015_16$home_goals - X2015_16$predicted_home_goals)^2)
cat("MSE for home goals:", mse_home, "\n")

# Calculate the Mean Squared Error (MSE) for away goals
mse_away <- mean((X2015_16$away_goals - X2015_16$predicted_away_goals)^2)
cat("MSE for away goals:", mse_away, "\n")

mse_overall <- mean(c(mse_home, mse_away))
cat("Overall MSE:", mse_overall, "\n")



ggplot(away_goal_data, aes(x = Goals, y = Probability)) +
  geom_bar(stat = "identity", fill = "#F4A261") + # Mild pastel color
  theme_minimal() +
  labs(title = "Poisson Distribution of Away Team Goals",
       y = "Probability", x = "Number of Goals") +
   scale_x_continuous(breaks = seq(0, 10, by = 1)) +   # x-axis scale with
increments of 1
   scale_y_continuous(breaks = seq(0, 1, by = 1))       # y-axis scale with
increments of 1 (not typical for probability)
```

Appendix 3:

Recent performance of player:

```
install.packages("pdp")
library(tidyverse)
library(randomForest)
library(caret)
library(pdp)

data <- read.csv("Premier_League_players.csv")

# Clean up the xG and xA columns by removing the adjustments (e.g., "+1.36" or
"-0.79")
data$xG <- as.numeric(sub("\\+.*|\\-.*", "", data$xG))
data$xA <- as.numeric(sub("\\+.*|\\-.*", "", data$xA))

head(data)

# Set seed for reproducibility
```

```
set.seed(123)
trainIndex <- createDataPartition(data$G, p = .8,
                                  list = FALSE,
                                  times = 1)
dataTrain <- data[trainIndex,]
dataTest <- data[-trainIndex,]

rf_model <- randomForest(G ~ xG90 + xA90 + Min + Apps, data = dataTrain, ntree =
100)

print(rf_model)
dataTest$predicted_goals <- predict(rf_model, newdata = dataTest)
predictions <- dplyr::select(dataTest, Player, Team, G, predicted_goals)
head(predictions)

# 1. Feature Importance Plot
importance <- importance(rf_model)
var_importance <- data.frame(Feature = rownames(importance), Importance =
importance[, 1])
ggplot(var_importance, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Feature Importance in Predicting Goals",
       x = "Feature",
       y = "Importance") +
  theme_minimal()

# 2. Partial Dependence Plot: Show the effect of xG90 on predicted goals

pdp_xG90 <- partial(rf_model, pred.var = "xG90", train = dataTrain)
autoplot(pdp_xG90) +
  labs(title = "Partial Dependence of xG90 on Predicted Goals",
       x = "xG90",
       y = "Predicted Goals") +
  scale_x_continuous(breaks = seq(min(pdp_xG90$xG90), max(pdp_xG90$xG90), by =
1)) +
  theme_minimal()

# Another Partial Dependence Plot: Show the effect of Min on predicted goals
pdp_Min <- partial(rf_model, pred.var = "Min", train = dataTrain)
autoplot(pdp_Min) +
  labs(title = "Partial Dependence of Minutes Played on Predicted Goals",
       x = "Minutes Played",
       y = "Predicted Goals") +
  scale_x_continuous(breaks = seq(min(pdp_Min$Min), max(pdp_Min$Min), by = 1)) +
  theme_minimal()
```

Appendix 4 :

Coding for brier score :

```
library(ggplot2)

# Input data
forecasted_prob <- c(0.80, 0.60, 0.70, 0.50, 0.90)
actual_outcome <- c(1, 0, 1, 0, 1)

brier_scores <- (forecasted_prob - actual_outcome) ^ 2


results <- data.frame(
  match = 1:length(forecasted_prob),
  forecasted_prob = forecasted_prob,
  actual_outcome = actual_outcome,
  brier_score = brier_scores
)

ggplot(results, aes(x = forecasted_prob, y = actual_outcome)) +
  geom_point(size = 4, color = "#2C3E50", shape = 21, fill = "#3498DB", stroke =
1.5) + # Points with clean colors
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "#95A5A6",
size = 1.2) + # Perfect prediction line
  geom_segment(aes(xend = forecasted_prob, yend = forecasted_prob),
               linetype = "dotted", color = "#E74C3C", size = 1.1) + # Error
lines
  geom_text(aes(label = paste0("Brier: ", round(brier_score, 3))), nudge_x =
0.05, size = 5, color = "#2C3E50", fontface = "bold") + # Brier score annotations
  labs(
    title = "Forecasted Probabilities vs. Actual Outcomes",
    subtitle = "Visualizing Brier Scores for Each Prediction",
    x = "Forecasted Probability of Success",
    y = "Actual Outcome (1 = Success, 0 = Failure)"
  ) +
  theme_minimal(base_size = 15) + # Minimalist theme with larger base size for
clarity
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold", color =
"#2C3E50"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, color = "#7F8C8D"),
    axis.title.x = element_text(size = 15, face = "bold", color = "#2C3E50"),
    axis.title.y = element_text(size = 15, face = "bold", color = "#2C3E50"),
    axis.text = element_text(size = 12, color = "#34495E"),
    panel.grid.major = element_line(color = "#ECF0F1", size = 0.7),
    panel.grid.minor = element_blank(),
```

```
    plot.background = element_rect(fill = "#FDFEFE", color = NA)
  )
```

Appendix 5:
Dixon coles coding :

```
library(dplyr)
library(ggplot2)
library(readr)

X2015_16 <- read_csv("C:/Users/ADMIN/Downloads/2015-16.csv")


X2015_16 <- X2015_16 %>%
  rename(home_goals = FTHG, away_goals = FTAG)

avg_home_goals <- mean(X2015_16$home_goals, na.rm = TRUE)
avg_away_goals <- mean(X2015_16$away_goals, na.rm = TRUE)

# Print the average goals
cat("Average home goals (λ):", avg_home_goals, "\n")
cat("Average away goals (λ):", avg_away_goals, "\n")

# Dixon-Coles log-likelihood function
dixon_coles_log_likelihood <- function(params, data) {
  home_attack <- params[1]
  away_attack <- params[2]
  home_defense <- params[3]
  away_defense <- params[4]
  rho <- params[5]

  lambda_home <- exp(home_attack + away_defense)
  lambda_away <- exp(away_attack + home_defense)

  log_likelihood <- 0
  for (i in 1:nrow(data)) {
    home_goals <- data$home_goals[i]
    away_goals <- data$away_goals[i]

    dc_correction <- ifelse(home_goals == 0 & away_goals == 0, (1 - rho)^(0.5),
                            ifelse(home_goals == 1 & away_goals == 0, (1 +
rho)^(0.5),
                                   ifelse(home_goals == 0 & away_goals == 1, (1 +
rho)^(0.5),
                                          ifelse(home_goals == 1 & away_goals ==
1, (1 - rho), 1))))
```

```
    log_likelihood <- log_likelihood +
      log(dpois(home_goals, lambda_home)) +
      log(dpois(away_goals, lambda_away)) +
      log(dc_correction)
  }

  return(-log_likelihood)
}
# Initial parameter guesses (home_attack, away_attack, home_defense,
away_defense, rho)
initial_params <- c(0, 0, 0, 0, 0)

# Optimize the parameters using optim
optim_results <- optim(
  par = initial_params,
  fn = dixon_coles_log_likelihood,
  data = X2015_16,
  method = "BFGS"
)

# Extract the estimated parameters
estimated_params <- optim_results$par
home_attack <- estimated_params[1]
away_attack <- estimated_params[2]
home_defense <- estimated_params[3]
away_defense <- estimated_params[4]
rho <- estimated_params[5]

# Print the estimated parameters
cat("Estimated home attack strength:", home_attack, "\n")
cat("Estimated away attack strength:", away_attack, "\n")
cat("Estimated home defense strength:", home_defense, "\n")
cat("Estimated away defense strength:", away_defense, "\n")
cat("Estimated rho (correlation):", rho, "\n")

# Predict goal probabilities using the Dixon-Coles model
predict_dc_goals <- function(lambda_home, lambda_away, rho, home_goals,
away_goals) {
  dc_correction <- ifelse(home_goals == 0 & away_goals == 0, (1 - rho)^(0.5),
                          ifelse(home_goals == 1 & away_goals == 0, (1 +
rho)^(0.5),
                                 ifelse(home_goals == 0 & away_goals == 1, (1 +
rho)^(0.5),
                                        ifelse(home_goals == 1 & away_goals == 1,
(1 - rho), 1))))

  prob <- dpois(home_goals, lambda_home) * dpois(away_goals, lambda_away) *
dc_correction
```

```
  return(prob)
}
# Calculate MSE for predictions
calculate_mse <- function(data, lambda_home, lambda_away, rho) {
  mse <- 0
  n <- nrow(data)

  for (i in 1:n) {
    home_goals <- data$home_goals[i]
    away_goals <- data$away_goals[i]
    predicted_prob <- predict_dc_goals(lambda_home, lambda_away, rho, home_goals,
away_goals)

    # Actual outcome (1 if the match ended with home_goals and away_goals,
otherwise 0)
    actual_prob <- ifelse(data$home_goals[i] == home_goals & data$away_goals[i]
== away_goals, 1, 0)

    # Squared error
    mse <- mse + (predicted_prob - actual_prob)^2
  }

  return(mse / n)
}

# Calculate lambda values using the estimated parameters
lambda_home <- exp(home_attack + away_defense)
lambda_away <- exp(away_attack + home_defense)

# Calculate MSE
mse_value <- calculate_mse(X2015_16, lambda_home, lambda_away, rho)
cat("Mean Squared Error (MSE) of Dixon-Coles model:", mse_value, "\n")
print(mse_value)
# Visualizing Dixon-Coles model predictions for home and away goals
home_goal_range <- 0:10
away_goal_range <- 0:10

home_goal_prob <- sapply(home_goal_range, function(x)
predict_dc_goals(lambda_home, lambda_away, rho, x, 0))
away_goal_prob <- sapply(away_goal_range, function(x)
predict_dc_goals(lambda_home, lambda_away, rho, 0, x))

home_goal_data <- data.frame(Goals = home_goal_range, Probability =
home_goal_prob)
away_goal_data <- data.frame(Goals = away_goal_range, Probability =
away_goal_prob)

# Plotting the distribution for home team goals
```

```
ggplot(home_goal_data, aes(x = Goals, y = Probability)) +
  geom_bar(stat = "identity", fill = "#A8DADC") + # Mild pastel color
  theme_minimal() +
  labs(title = "Dixon-Coles Model: Home Team Goals",
       y = "Probability", x = "Number of Goals") +
  scale_x_continuous(breaks = seq(0, 10, by = 1))

# Plotting the distribution for away team goals
ggplot(away_goal_data, aes(x = Goals, y = Probability)) +
  geom_bar(stat = "identity", fill = "#F4A261") + # Mild pastel color
  theme_minimal() +
  labs(title = "Dixon-Coles Model: Away Team Goals",
       y = "Probability", x = "Number of Goals") +
  scale_x_continuous(breaks = seq(0, 10, by = 1))
```

Appendix 6 :

```
library(dplyr)
library(ggplot2)
library(caret)  # For creating the confusion matrix
library(tidyr)
library(readr)
library(corrplot)
library(tidyr)

# Load the dataset
understat_per_game <- read_csv("C:/Users/ADMIN/Downloads/understat_per_game.csv")

# Prepare the data
understat_per_game <- understat_per_game %>%

  select(team, h_a, scored, xG, xGA, deep, deep_allowed) %>%
  rename(

    Team = team,
    HomeAway = h_a,
    GoalsScored = scored,
    ExpectedGoals = xG,
    ExpectedGoalsAgainst = xGA

  )

# Filter out home and away games separately
home_data <- understat_per_game %>% filter(HomeAway == 'h')
away_data <- understat_per_game %>% filter(HomeAway == 'a')
```

```
# Fit Poisson models for home and away goals

home_poisson_model <- glm(GoalsScored ~ Team + ExpectedGoals, data = home_data,
family = poisson(link = "log"))

away_poisson_model <- glm(GoalsScored ~ Team + ExpectedGoals, data = away_data,
family = poisson(link = "log"))

# Predict goals for new matches
home_data$PredictedGoals_Naive <- predict(home_poisson_model, newdata =
home_data, type = "response")

away_data$PredictedGoals_Naive <- predict(away_poisson_model, newdata =
away_data, type = "response")


# Combine home and away data
predicted_data <- rbind(home_data, away_data)

# Dixon-Coles Model Function
dixon_coles_model <- function(lambda, mu, rho = 0.15) {

  ifelse(lambda + mu == 0, 1 - rho, (1 - rho) + (lambda - mu) / (lambda + mu) *
rho)

}

# Apply the Dixon-Coles adjustment
predicted_data <- predicted_data %>%

  group_by(Team) %>%
  mutate(

    Lambda = ifelse(HomeAway == 'h', PredictedGoals_Naive, NA),
    Mu = ifelse(HomeAway == 'a', PredictedGoals_Naive, NA),
    Lambda_Home = mean(PredictedGoals_Naive[HomeAway == 'h'], na.rm = TRUE),
    Mu_Away = mean(PredictedGoals_Naive[HomeAway == 'a'], na.rm = TRUE),
    PredictedGoals_Dixon = ifelse(HomeAway == 'h',

                                  Lambda * dixon_coles_model(Lambda, Mu_Away),

                                  Mu_Away * dixon_coles_model(Lambda_Home, Mu)),

  ) %>%

  ungroup()

home_games_data <- predicted_data %>% filter(HomeAway == 'h')
```

```
ggplot(home_games_data) +
  geom_histogram(aes(x = GoalsScored, y = ..count.., fill = "Actual Goals"), bins
= 30, alpha = 0.5, position = "identity") +
  geom_histogram(aes(x = PredictedGoals_Naive, y = ..count.., fill = "Predicted
Goals (Naive)"), bins = 30, alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("Actual Goals" = "#FF6347", "Predicted Goals
(Naive)" = "#4682B4")) +
  labs(
    title = "Histogram: Naive Model Predictions for Home Games",
    x = "Number of Goals",
    y = "Frequency",
    fill = "Legend"
  ) +
  scale_x_continuous(breaks = seq(0, max(home_games_data$GoalsScored, na.rm =
TRUE), by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 18, face = "bold"),
        axis.title = element_text(size = 14),
        legend.position = "top")

# Filter data for away games
away_games_data <- predicted_data %>% filter(HomeAway == 'a')

ggplot(away_games_data) +
  geom_histogram(aes(x = GoalsScored, y = ..count.., fill = "Actual Goals"), bins
= 30, alpha = 0.5, position = "identity") +
  geom_histogram(aes(x = PredictedGoals_Naive, y = ..count.., fill = "Predicted
Goals (Naive)"), bins = 30, alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("Actual Goals" = "#FF6347", "Predicted Goals
(Naive)" = "#4682B4")) +
  labs(
    title = "Histogram: Naive Model Predictions for Away Games",
    x = "Number of Goals",
    y = "Frequency",
    fill = "Legend"
  ) +
  scale_x_continuous(breaks = seq(0, max(away_games_data$GoalsScored, na.rm =
TRUE), by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 18, face = "bold"),
        axis.title = element_text(size = 14),
        legend.position = "top")

ggplot(home_games_data) +
  geom_histogram(aes(x = GoalsScored, y = ..count.., fill = "Actual Goals"), bins
= 30, alpha = 0.5, position = "identity") +
```

```
  geom_histogram(aes(x = PredictedGoals_Dixon, y = ..count.., fill = "Predicted
Goals (Dixon-Coles)"), bins = 30, alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("Actual Goals" = "#FF6347", "Predicted Goals
(Dixon-Coles)" = "#006400")) +
  labs(
    title = "Histogram: Dixon-Coles Model Predictions for Home Games",
    x = "Number of Goals",
    y = "Frequency",
    fill = "Legend"
  ) +
  scale_x_continuous(breaks = seq(0, max(home_games_data$GoalsScored, na.rm =
TRUE), by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 18, face = "bold"),
        axis.title = element_text(size = 14),
        legend.position = "top")

ggplot(away_games_data) +
  geom_histogram(aes(x = GoalsScored, y = ..count.., fill = "Actual Goals"), bins
= 30, alpha = 0.5, position = "identity") +
  geom_histogram(aes(x = PredictedGoals_Dixon, y = ..count.., fill = "Predicted
Goals (Dixon-Coles)"), bins = 30, alpha = 0.5, position = "identity") +
  scale_fill_manual(values = c("Actual Goals" = "#FF6347", "Predicted Goals
(Dixon-Coles)" = "#006400")) +
  labs(
    title = "Histogram: Dixon-Coles Model Predictions for Away Games",
    x = "Number of Goals",
    y = "Frequency",
    fill = "Legend"
  ) +
  scale_x_continuous(breaks = seq(0, max(away_games_data$GoalsScored, na.rm =
TRUE), by = 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 18, face = "bold"),
        axis.title = element_text(size = 14),
        legend.position = "top")

predicted_data <- predicted_data %>%
  mutate(

    NaiveCategory = case_when(
      PredictedGoals_Naive < GoalsScored ~ "Under-prediction",
      PredictedGoals_Naive == GoalsScored ~ "Correct prediction",
      PredictedGoals_Naive > GoalsScored ~ "Over-prediction"

    ),

    DixonCategory = case_when(
```

```
      PredictedGoals_Dixon < GoalsScored ~ "Under-prediction",
      PredictedGoals_Dixon == GoalsScored ~ "Correct prediction",
      PredictedGoals_Dixon > GoalsScored ~ "Over-prediction"
    )

  )


# Confusion matrix for Naive model
conf_matrix_naive <- table(Prediction = predicted_data$NaiveCategory, Actual =
factor(predicted_data$GoalsScored))

# Confusion matrix for Dixon-Coles model
conf_matrix_dixon <- table(Prediction = predicted_data$DixonCategory, Actual =
factor(predicted_data$GoalsScored))

# Print the confusion matrices
print(conf_matrix_naive)
print(conf_matrix_dixon)

normalize_matrix <- function(mat) {
  row_sums <- rowSums(mat)
  normalized_mat <- sweep(mat, 1, row_sums, "/")
  return(normalized_mat)

}

conf_matrix_naive_normalized <- normalize_matrix(conf_matrix_naive)
conf_matrix_dixon_normalized <- normalize_matrix(conf_matrix_dixon)

naive_data <- data.frame(
  Prediction = rep(c("Over-prediction", "Under-prediction"), each = 11),
  Actual = rep(0:10, 2),
  Count = c(
    6979, 4598,  888,  265, 115,  41,  26,   4,  5,  2,  2,   # Over-prediction
    0, 3716, 4477, 2230, 815, 296,  92,  20,  7,  2,  0    # Under-prediction
  )
)

dixon_data <- data.frame(
  Prediction = rep(c("Over-prediction", "Under-prediction"), each = 11),
  Actual = rep(0:10, 2),
  Count = c(
    6979, 3745,  404,  107,  56,  21,  13,   2,  3,  2,  1,   # Over-prediction
    0, 4569, 4961, 2388, 874, 316, 105,  22,  9,  2,  1    # Under-prediction
  )
)
```

```
ggplot(naive_data, aes(x = factor(Actual), y = Prediction)) +
  geom_tile(aes(fill = Count), color = "white") +
  geom_text(aes(label = Count), color = "black", size = 3) +
  scale_fill_gradient(low = "white", high = "#4169E1") +
  labs(title = "Confusion Matrix - Naive Model", x = "Actual Goals", y =
"Prediction") +
  theme_minimal() +
  theme(plot.title = element_text(size = 15, face = "bold"))

ggplot(dixon_data, aes(x = factor(Actual), y = Prediction)) +
  geom_tile(aes(fill = Count), color = "white") +
  geom_text(aes(label = Count), color = "black", size = 3) +
  scale_fill_gradient(low = "white", high = "yellow") +
  labs(title = "Confusion Matrix - Dixon-Coles Model", x = "Actual Goals", y =
"Prediction") +
  theme_minimal() +
  theme(plot.title = element_text(size = 15, face = "bold"))


# Calculate Mean Squared Error (MSE)

mse_naive <- mean((predicted_data$GoalsScored -
predicted_data$PredictedGoals_Naive)^2)

mse_dixon <- mean((predicted_data$GoalsScored -
predicted_data$PredictedGoals_Dixon)^2)

# Calculate Root Mean Squared Error (RMSE)
rmse_naive <- sqrt(mse_naive)
rmse_dixon <- sqrt(mse_dixon)

# Calculate Mean Absolute Error (MAE)

mae_naive <- mean(abs(predicted_data$GoalsScored -
predicted_data$PredictedGoals_Naive))

mae_dixon <- mean(abs(predicted_data$GoalsScored -
predicted_data$PredictedGoals_Dixon))


cat("Naive Model Metrics:\n")

Naive Model Metrics:

  cat("MSE:", mse_naive, "\n")


cat("RMSE:", rmse_naive, "\n")
```

```
cat("MAE:", mae_naive, "\n")


cat("Dixon-Coles Model Metrics:\n")



 cat("MSE:", mse_dixon, "\n")


 cat("RMSE:", rmse_dixon, "\n")


 cat("MAE:", mae_dixon, "\n")


residuals_naive <- predicted_data$GoalsScored -
predicted_data$PredictedGoals_Naive

residuals_dixon <- predicted_data$GoalsScored -
predicted_data$PredictedGoals_Dixon
# Perform paired t-test

 t_test_result <- t.test(residuals_naive, residuals_dixon, paired = TRUE)

 print(t_test_result)
```

Chapter 9

9.1 Reference:

1. Dixon, Mark J., and Stuart G. Coles. ”Modelling association football scores and inefficiencies in the football betting market.” Journal of the Royal Statistical Society: Series C (Applied Statistics) 46.2 (1997): 265-280.

2. Baboota, R. and Kaur, H. (2019). Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting, 35, 741–755.

3. Crowder, M., Dixon, M., Ledford, A., and Robinson, M. (2002). Dynamic modelling and prediction of English football league matches for betting. The Statistician, 51, 157–168

4. Boshnakov, G., Kharrat, T., and McHale, I. (2017). A bivariate Weibull count model for forecasting association football scores. International Journal of Forecasting, 33, 458–466

5. Constantinou, A. (2019). Dolores: a model that predicts football match outcomes from all over the world. Machine Learning, 108, 49–75.

6. Constantinou, A. and Fenton, N. (2013). Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries. Journal of Quantitative Analysis in Sports, 9, 37–50.

7. Goddard, J. (2005). Regression models for forecasting goals and match results in association football. International Journal of Forecasting, 21, 331–340