Playing the Odds: Accurately Predicting Soccer Games In English Premier League Using Ordered Probit Models and Artificial Neural Networks

Independent Study Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Bachelor of Arts in the Department of Mathematics and Computer Science at The College of Wooster

by
Shivam K C

The College of Wooster
2020

Advised by:

Dr. Colby Long (Mathematics)

Abstract

Several studies have attempted to predict soccer games using various machine learning algorithms. Few of them have succeeded in predicting soccer games with predictive accuracy (PA) as high as 54.6%. This paper aims to predict English Premier League (EPL) soccer games with PA higher than 54.6%. To reach this goal, we build several ordered probit models, and Artificial Neural Network (ANN) using EPL data from 8 seasons (2008-09 season to 2016-17 seasons). The results show that a simple statistical model comes closest to reaching the target set by a complex ANN model: an ordered probit model with 4 predictors has an average PA of 53.5%. However, the model is heavily reliant on betting odds data. Likewise, an ANN model with 4 predictors has an average PA of 50.8%. These results suggest that if we want to build a model with higher PA, which does not rely on betting odds data, then building more complex ANN model may be the key: specifically, building ANN models with more hidden layers and nodes.

Dedicated To

My beloved family, especially Paru. Thank you for always being there for me, our weekend family calls, and my conditioning!

My beloved host family, The Perkins;
Sunita diju and her family;
Sangita diju and her family; and
Millu diju and her family
for your hospitality and kindness. I am grateful to have learned how to live a balanced life, be giving, be systematic, and be open-minded from you. All of you have reminded me that it does take a village to raise a child, and I am still growing.

Pratyush for our weekly mastermind and our commitment to challenge each other to grow and give!

Tony Robbins for introducing me to numerous mental models to navigate through life and guiding me to discover my purpose in life.

Acknowledgements

I would like to take the opportunity to thank my advisor, Professor Colby Long, and my writing coach at the Writing Center, Lynette Mattson, for their constant encouragement and flexibility in working around my student teaching experience. I also would like to thank the entire Mathematics and Computer Science department at the College of Wooster for creating an amazing major experience by maintaining patience and friendly faces throughout my hours of questioning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

“They played a good game, but they lost because they were unlucky.” That is what my dad used to tell me every time we witnessed the best team in a soccer match lose the game that they “should have won.” I did not understand him then because I thought that luck must be foreseeable. Later, I realized that I might have fallen into the trap of “resulting– tying the quality of the outcome too tightly to the quality of decisions,” which is an unfortunate fallacy if one’s ultimate goal for a thesis is to accurately predict soccer games [9]. In soccer, an example of resulting could be when team A plays team B, and a fan decides to bet on team A because he feels that team A is relatively more skillful than team B. Here, he is predicting the outcome of the game only based on the quality of skills of the teams. This is a problem because the skillful teams do not always win games in soccer. A team needs both skill and fortune to win a game [34].

Before being aware of the resulting phenomenon, the main goal for my IS–which was chosen well before we chose the College of Wooster–was to accurately predict English Premier League (EPL) games by predicting both luck and skill of a team. After being aware of the phenomenon, I slightly changed my approach towards achieving the main goal. In this paper, I am asking if I can build a mathematical model to predict soccer games better than the bookmakers and the expert tipsters. Specifically, I am aiming to predict EPL games with at least 55% accuracy in the long run, which would better the best known predictive accuracy of the expert predictors [31].

Personally, the two main reasons why I am aiming to accurately predict EPL games for this research paper are: my love for soccer and soccer predictions, and the challenge of predicting soccer games. Firstly, I am passionate about watching soccer, predicting soccer games, playing soccer and soccer video games (FIFA) with my friends for fun. While I was learning about the College of Wooster and the Independent Study (IS) program the college offers, I instinctively told myself that I will predict soccer games for my IS, which is exactly what I am doing. Secondly, the challenge of predicting soccer games motivates me to accurately predict soccer games as I improve my decision making ability with each incorrect prediction. Make no mistake, soccer is notoriously unpredictable. Consequently, just as chaos breeds opportunities, the unpredictability of soccer offers one a great opportunity to improve his decision making ability, but only if he is willing to listen and learn from his decisions.

Out of soccer, handball, American football, and basketball, soccer seems to be the most unpredictable of them all; this is mainly because of the possibility of a draw as an outcome of a soccer game and the rarity of goals in soccer [34, Chapter 2]. According to Ben-Naim et al., the ”measured frequency of upsets” in the EPL, Major League Baseball (MLB), National Hockey League (NHL), National Basketball Association (NBA), and National Football League (NFL) were 45.2%, 44.1%, 41.4%, 36.5%, and 36.4% respectively [21]. That's why Arsene Wenger, football club Arsenal's manager, once said, "I get nervous before the games because soccer is not mathematics, it's completely unpredictable. That's what makes it so popular" [5]. But, that does not discourage us from using mathematics to predict EPL games.

The main reason why I am choosing EPL games to predict is because EPL is "the most-watched league on the planet with one billion homes watching the action in 188 countries," and it is my favorite soccer league to follow [16]. To begin with, EPL is the highest level of the English soccer league system, which is contested by 20 soccer clubs. It operates on a system of promotion and relegation, where 3 points are awarded to a team for a win, 1 point for a draw, and 0 for a loss. At the end of the season, the team with the most points wins the EPL and the teams that finish in the bottom three of the league table are relegated to the second tier of English soccer. They are then replaced by three clubs promoted from the second tier. Therefore, the stakes for winning and losing in EPL are high and the need to accurately predict soccer games in EPL is great.

This paper will contribute to the scholarship surrounding the sport result prediction field, in particular, the field of application of machine learning algorithms to accurately predict soccer games.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the related work to this research. Also, in this section, we discuss the results and shortcomings of previous modelling approaches. We introduce in Section 3 the datasets used in the research. In Section 4, we elaborate on the methods used for the research. We present the main findings in Section 5. In Section 6, we evaluate and discuss the obtained results. Finally, we conclude the paper in Section 7.

Chapter 2

Related Work

The goal of this research is to build a mathematical model to predict soccer games better than the bookmakers and the expert tipsters. Existing literature regarding the prediction of soccer matches is discussed in Section 2.1. Section 2.2 presents a brief overview of existing literature related to the application of machine learning algorithms to predict soccer matches.

2.1 Models Predicting Soccer Outcomes

According to Tim van der Zaan [37], all the proposed models from the existing literature related to the prediction of soccer matches can be categorized into two types of models: goal-models and toto-models. The goal-models predict the number of goals scored by a team, which indirectly predicts a match outcome. The toto-models directly predict the outcome of a soccer match: either a home team victory, an away team victory, or a draw.

Before we give an example of a toto-model, we will briefly talk about the basics of predictive modelling. Predictive modeling is the process of using data and statistics to predict outcomes with data models [15]. We can use these models to predict results of a variety of situations, ranging from sports outcomes and climate change to technological advances and college students' academic performance. Training datasets are the datasets that are used to build the predictive models and, therefore, are the datasets which the models are trained on. Testing datasets are the datasets used by the trained models to predict certain outcomes. For instance, if one has access to a college's retention dataset for the last decade, then one could train their data models on 8 of the 10 years of datasets and test their predictions on the remaining 2 years. Splitting the data into training data and testing data allows us to detect overfitting, which is a situation when the data models predict with a higher accuracy on the training dataset. However, the data models predict poorly on a (new) test dataset, hinting at the data models' unreliability in predicting the future.

After a brief introduction to predictive modeling, now, let us take an example of a toto-model. Ben Ulmer and Matthew Fernandez's research is one such example, in which their prediction for each EPL game was in one of three classes: win, draw, or loss [36]. For their training dataset, they used EPL 2002-03 season to 2011-12 season data, and for their testing dataset, they used 2012-13 season and 2013-14 season data. The major features they used to predict the games' outcomes were: whether a team is home or away, gameday features for each team, and a team's form measured by its results in recent matches. The machine learning algorithms they used were five different classifiers: Linear from stochastic gradient descent, Naive Bayes, Hidden Markov Model, Support Vector Machine (SVM), and Random Forest. Their best error rates were with their Linear classifier (.48), Random Forest (.50), and SVM (.50), "which is comparable with the error rate of .48 for leading BBC soccer analyst Mark Lawrenson, but still worse than the error rate of .45 for the 'oracle' betting organization, Pinnacle Sports" [36]. From this study, we have learned how to approach building the mathematical models using numerous machine learning algorithms and have accepted their challenge to predict with a higher accuracy rate. Our models (as discussed in Section 3) will be different from the study's models in that we will be using an Ordered Probit model and an Artificial Neural Network model in our research.

Similar to the study discussed in the preceding paragraph, Snyder used a dataset sourced from the Manchester City Analytics program to predict the 380 matches of the 2011-2012 EPL season [35]. Snyder rigorously compared the betting profit obtained from the probabilities assigned by numerous toto-models and goal-models and found that the betting profit of various strategies is significantly higher for toto-models than for goal-models. This finding nudged us to focus on building toto-models instead of goal-models for our project. Therefore, numerous toto-models will be constructed using Ordered Probit Prediction model and ANN, incorporating previous seasons (2008-2016) match results and various other explanatory variables.

Likewise, Tim van der Zaan built four different ordered probit models to predict the outcome of soccer matches of the highest level of the national leagues of England, Spain, the Netherlands, and Sweden. He also researched "various betting strategies that exploit the predicted likelihood of the outcome possibilities assigned by match outcome prediction models" [37]. They extracted their data for betting odds for soccer games of European competitions from *oddsportal.com* and soccer related data from Infostrada (which contains sport databases for the highest level of national leagues of 51 European countries from seasons 1986-87 to 2015-16). Since our IS is focused on EPL, we chose to focus on the van der Zaan's results for EPL (see Table 2.1).

2.2 Applying Machine Learning Algorithms To Predict Soccer

Machine Learning Algorithms (MLAs) are one of the most popular mechanisms used to build predictive models for predicting soccer games. Among MLAs, Artificial Neural Networks (ANN) are widely used. Bunker and Thabtah conducted critical analyses on some studies on sport prediction that have used ANN. The authors proposed a simple sport result prediction 'SRP-CRISP-DM' framework to follow while attempting to solve the complex problem of sport result prediction [25]. They recommend readers include player level data, as "including player level data would have the advantage that we can investigate whether specific players' actions or presence are important for the performance of the team in terms of whether they win or lose" [25]. Also, this article introduced us to numerous high-level papers on sport result prediction. For instance, it is here where we learned about McCabe and Trevathan's expert tipsters beating ANN model.

Likewise, Mccabe and Trevathan ventured to predict results in four different sports: NFL (Rugby League), AFL (Australian Football League), Super Rugby (Rugby Union), and English Premier League Football (EPL) [31].

Table 2.1: Summary of predictive accuracy (PA) for the best models among the related work used for our research. Our aim is to better PA of 54.6% of Mccabe and Trevathan [31].

Scholars	Number of EPL Seasons in Training Set	Model	Predictive Accuracy
Mccabe and Trevathan [31]	5	ANN	54.6%
van der Zaan [37]	3	Basic ECI model (an Ordered Probit model)	52.8%
Ulmer and Fernandez [36]	10	Random Forest and SVM	50%

To accomplish this, they used data from as early as 2002. For EPL, they used team-specific features such as home and away performance, goals scored, goals conceded, overall performance, etc. They fed such features into an ANN, specifically, a model (which is described in Section 3.1.2) in the form of multi-layer perceptron. For EPL, they achieved an average predictive accuracy of 54.6% . They entered their model in a major international tipping competition called TopTipper in the 2006-2007 season, which hosts thousands of human contestants from year to year, of varying skill levels. By the final week of the competition, their model was ranked number one among the contestants. As a result, we are inspired by their model to create our own ANN predictive model.

Chapter 3

Data

3.1 Introduction of Data

The raw data for this research was obtained from a Kaggle dataset titled ”European Soccer Database” [4]. The dataset consisted of data from 11 European leagues such as Belgium Jupiler League, Italy Serie A, Spanish LA Liga, English Premier League (EPL), etc. Since the focus of our research is to predict EPL games, we had to subset the dataset to include information about EPL games. In R Studio, we used “RSQLite” library to convert the SQL database to data frames. Then, we used “dplyr” package to carry out necessary data cleaning.

Below is an overview of what the database as a whole consisted of:

- +25,000 matches
- +10,000 players
- 11 European Countries with their league championship Seasons 2008 to

2016

- Players' and Teams' attributes sourced from EA Sports' FIFA video game series, including the weekly updates
- Team line up with squad formation (X, Y coordinates)
- Betting odds from up to 10 providers
- Detailed match events (goal types, possession, corner, cross, fouls, cards, etc.) for +10,000 matches

There are 380 games played during a season of EPL. So, for each season of EPL, the dataset consisted of 380 observations with 115 variables. Thus, for 8 seasons of EPL, we had 3040 observations with 115 variables. This was before data processing.

We were following in the footsteps of Mccabe and Trevathan in using the variables they used for their ANN model, in our models because their model had the second highest predictive accuracy as we discovered in our research (shown in Table ?) [31]. So, we did not include every variable present in the dataset in our models. Also, not every variable available in the dataset seemed useful. For instance, we did not prefer to include the players and teams' attributes sourced from EA Sports' FIFA video game series because the statistics were based on the video game makers' ratings instead of the players' real-world performances.

To get the variables we wanted for our research, a great deal of time and effort was spent on data processing. The Table 3.1 consists of the variables we used in our models. Specifically, we just used the following variables from the Kaggle dataset to create the variables we wanted:

- season: the season when the EPL game was played.
- stage: the game week when the EPL game was played. There are 38 game weeks in an EPL season. Usually, 10 games are played in each game week.
- id: Match ID.
- home_team_api_id: unique identifier for home team.
- away_team_api_id: unique identifier for away team.
- home_team_goal: the number of goals scored by a home team in a game.
- away_team_goal: the number of goals scored by an away team in a game.
- B365_H: the odds of home team winning the game given by the gambling company Bet365.
- B365_A: the odds of away team winning the game given by the gambling company Bet365.
- B365_D: the odds of draw as an outcome in the game given by the gambling company Bet365.

	Definition

match_outcome (Y1)	3 categories:
	A=Away team win
	D=Draw
	H=Home team win

Goals_against_so_far_d (X2)	(Goals conceded by home team so far in the season – Goals conceded by away team so far in the season) / (Games played by the teams so far).
Goals_for_so_far_d (X3)	(Goals scored by home team so far in the season – Goals scored by away team so far in the season) / (Games played by the teams so far)
overall_performance_so_far_d (X4)	(Home team's performance – Away team's performance) / (Games played by the teams so far). Here, the team's performance refers to their win/loss record. Three points are awarded for a win, one point for a draw and no points for a loss. Performance is then the sum of these values over each round of competition so far.
cum_performance_so_far_d (X5)	(Home team's home performance – Away team's away performance) / (Games played by the teams so far/2)
performance_in_previous_game_d (X6)	(Home team's performance in last game – Away team's performance in last game)
performance_in_previous_n_games_d (X7-X10)	(Home team's performance in last n games – Away team's performance in last n games). Here, n = 2, 3, 4, 5. This is an attempt to gauge the recent form of the teams and consider whether the team is on a winning or losing streak.
goals_for_in_previous_n_game_d (X11-X15)	(Goals scored by home team in previous n games – Goals scored by away team in previous n games). Here, n = 1, 2, 3, 4, 5.
goals_against_in_previous_n_game_d (X16-X20)	(Goals conceded by home team in previous n games – Goals conceded by away team in previous n games). Here, n = 1, 2, 3, 4, 5. Also, goals conceded by a team is expressed as negative number for ANN 1.
B365H_1 (X21)	The odds of home team winning the game given by the gambling company Bet365.
B365A_1 (X22)	The odds of away team winning the game given by the gambling company Bet365.
B365D_1 (X23)	The odds of draw as a outcome in the game given by the gambling company Bet365.

Table 3.1: Summaries of key variables. All variables are numerical except match_outcome, which is categorical. Here, the variables are given aliases like Y1 and so on, because they are used to visualize complex ANN later on in Section 5.

3.2 Pre-processing of Data

We obtained the variables in Table 3.1 by manipulating the above variables from Kaggle dataset. The process is described below:

- For each match (unique id),
    - if home_team_goal > away_team_goal, match_outcome = H
    - if away_team_goal > home_team_goal, match_outcome = A
    - if home_team_goal = away_team_goal, match_outcome = D
- To calculate Goals_against_so_far_d, we had to calculate:
    - Goals conceded by home team so far in the season and goals conceded by away team so far in the season:
        - For a team, this was obtained by summing the goals the team conceded in each game week (stage) up to the game week where the game is being played. For instance, if team A plays team B at home in game week 26, then the goals conceded by home team so far in the season is obtained by summing the goals team A conceded in each stage leading up to game week 26. Similarly, the goals conceded by away team so far in the season is obtained by summing the goals team B conceded in each stage leading up to game week 26.
- To calculate Goals_for_so_far_d, we had to calculate:
    - Goals scored by home team so far in the season and goals scored by away team so far in the season:

    - The process is similar to Goals_against_so_far_d. The only difference is that we are summing the goals a team scored in each game week instead of summing the goals the team conceded in each game week.

- overall_performance_so_far_d and cum_performance_so_far_d are described above in the Table 1.4 The difference between these two is that for cum_performance_so_far_d, we are interested in home team's home performance and away team's away performance so far in the season instead of focusing on the teams' overall performance so far in the season, which we did for overall_performance_so_far_d.

- The remaining variables are extension of the above variables in order to gauge the teams' recent forms.

The major reasons why we used the variables in Table 3.1 in our models are:

1. We modeled the variables Mccabe and Trevathan used in their ANN model because they had the highest predictive accuracy from all the research paper we looked at [31]. Our thought process was let's see what happens when we use the variables, they used in their ANN model and use them in different data models. So, instead of modelling their ANN model, we built two different models in our research: An Ordered Probit Model and an ANN model which is different to theirs as described in Section 4.

2. We used cum_performance_so_far_d so that our model could sense home team advantage if there was any. There were numerous research that mentioned that there is a huge advantage for a home team in soccer [20, 34, 37]. For instance, Vergin and Sosik proposed in their paper that the home team in sport obtains an advantage over the away team [38]. Even in our dataset, we found that the home team won 46.3% of the time and the away team won only 28.1% of the time (see Figure 4.3).

3. We used betting odds in our model because we wanted to see what happens if we added betting odds as some variables in addition to the variables McCabe and Trevathan used in their model.

4. performance_in_previous_game_d, performance_in_previous_n_games_d, goals_for_in_previous_n_game_d, and goals_against_in_previous_n_game_d, were used in our models to include the information of teams' current run of form while predicting the outcomes of a game. Blundell [23], Dixon and Coles [26], and van der Zaan [37] proposed similar process in specifying the recent team's form based on the last couple of matches played.

3.3 Data Scaling for ANN Models

There are two main reasons why it is important to normalize the predictors for ANN models [7].

Firstly, our input variables may have different units. So, the variables will have varying scales of measurement. These differences in the scales across the input variables could cause difficulty in building an accurate predictive neural network. For instance, let us say employees' yearly salary (in dollars) and employees' age are used to predict whether the employees will stay in the job or look for another job. The big scale difference of the yearly salary in thousands of dollars and employees' age can result in a model being confused about what weight values to assign. The model might learn to assign large weight values. This creates instability, meaning that the model will suffer from poor performance during the learning phase due to sensitivity to input values resulting in higher generalization error.

Secondly, if the input variables are in the same scale (0 to 1), then the model will have easier time learning the patterns and assigning weights during the learning phase and the learning process will be much faster. This is because the model is not wasting time being confused about what weights to assign due to the differences in scales of the input variables.

Let us say we want to normalize Goals_for_so_far_d column. Then,

$$Normalized \quad Goals\_for\_so\_far\_d = (Goals\_for\_so\_far\_d – min)/(max – min) \tag{3.1}$$

Here, min = minimum of Goals_for_so_far_d column (not normalized) and

max = maximum of Goals_for_so_far_d column (not normalized).

3.4 Reasons For Certain Variables

If you look at Table 3.1, Goals_against_so_far_d, Goals_for_so_far_d, overall_performance_so_far_d, and cum_performance_so_far_d are what we call "per game" variables because they are divided by the number of games played by the teams so far in the season. The reason why we did this is because we thought just using the non- "per game" version of the variables (in other words, not dividing the variables by games played by the teams so far) would confuse the ANN in the training phase. Specifically, the ANN could struggle calculating accurate weights during backpropagation stage. For instance,

Table 3.2: The table shows a hypothetical example for why a "per game" variable is used in our models.

Teams	Team Performance in GW 6	Teams	Team Performance in GW 38
Manchester City (H)	16	Manchester City (H)	85
West Brom (A)	3	Manchester United (A)	72
raw difference	13	-	13
overall_performance_so_far_d	13/6 = 2.2	-	13/38 = 0.34

From the above table, let us say in game week (GW) 6, Manchester City is playing West Brom at home. Then, the team performance (see Table 3.1) for Manchester City is 16 and that of West Brom is 3. As a result, the raw difference between the home team's performance and the away team's performance is 13 and overall_performance_so_far_d is 2.2. Similarly, let us say in GW 38, Manchester City is playing Manchester United at home. Then, the team performance for Manchester City is 85 and that of Manchester United is 72.

Consequently, the raw difference between the home team's performance and the away team's performance is 13 and overall_performance_so_far_d is 0.34.

Here, if raw difference is used as a predictor in an ANN model to differentiate teams' quality, then it will appear as if we are presenting the ANN with misleading data to learn from. For instance, we expect Manchester City to beat West Brom in GW 6 and most likely a close game between Manchester City and Manchester United because of the difference in teams' qualities. However, we are presenting the ANN with a key predictor for comparing two teams, but the predictor has the same value when comparing Manchester City/West Brom (GW 6) as when comparing Manchester City/Manchester United (GW 38). So, it will be difficult for the ANN to learn from this data that since the the predictor is the same for these two games, but the expected outcomes should be different.

However, if overall_performance_so_far_d ("per game" variable) is used as a predictor in the ANN model, then it will seem like we are presenting the ANN with a useful data to learn from. ANN model will most likely learn that the team quality difference between Manchester City and West Brom (2.2) is higher than the team quality difference between Manchester City and Manchester United (0.34). In other words, the ANN model will learn from this data that since the predictor is different for these two games, the expected outcomes should be different. Thus, we are using "per game" variables.

3.5 Handling of Missing Data

For each season, we do not consider the first 5 game weeks (GWs) because we do not have the data for the variables such as performance_in_previous_n_games_d, goals_for_in_previous_n_game_d, and goals_against_in_previous_n_game_d (see Table 3.1). Once we do have the data after 5 GWs, we use data for each season from GW 5 to GW 38. We could have used data from previous seasons to fill in the data for the first 5 GWs. However, we choose not to do it because we know that at the beginning of a season there are some drastic changes happening at a club with bunch of players leaving and new managers and/or players joining a team. So, a team that played GW 38 of the past season is seldom the same team that plays GW 1 of a new season. Therefore, to take care of this missing data, we remove the rows for the first 5 GWs for each season.

Hence, the final number of variables and observations for EPL data are 22 and 2640, respectively. The initial number of variables and observations for EPL data were 115 and 3040, respectively.

Chapter 4

Methodology

4.1 Proposed Models

4.1.1 Ordered Probit Prediction Model

In this paper, numerous ordered probit models are built to predict the outcome of a soccer match (see Section 5). These models generate probabilities for a home win, draw, and away win for each match.

We could have chosen a multinominal probit model over an ordered probit model as the outcomes of home win, away win, and draw do not seem have a logical order. However, what matters to us is that draw is an outcome between home win and away win. Then, home win and away win can be on the either side (positive or negative) of draw. Similarly, one could argue that there is an order, in that a draw and a home win are ”closer” outcomes than an away win and a home win. Additionally, another reason why we chose ordered probit model is because van der Zaan had successfully used it in his study and had an high predictive accuracy [31]. Also, we compared our multinominal probit model's results to our ordered probit model's results and found that the ordered probit model had a higher average predictive accuracy (53.6%) than the multinominal probit model (43.3%).

A description of the math behind the ordered probit model is given below, adapted from [37]:

An unobserved latent variable (in other words, a dependent variable) is incorporated by the ordered probit model. The model consists of explanatory variables, given by x_i, related to the clubs competing in a match i. As a result,

$$y_i^* = \beta x_i + \epsilon_i, \quad \text{where} \quad \epsilon_i \sim N(0,1) \tag{4.1}$$

Here, under the assumption of fixed predictors, y_i^* is normally distributed as i is standard normally distributed, and y_i^* is linearly related to x_i.

To understand how the value of y_i^* varies per match, let's take an example of a match between two teams that are equally matched: they have the exact same history, qualities, and are playing at a neutral venue. Theoretically, the match should end up in a draw. However, in reality, numerous unexplained factors, such as luck, cause the match to be won by one team or the other. The normal distribution of y_i^* is depicted on Fig. 4.1, which also shows how y_i^* differs in each match depending on team and match variables. Consequently, in every match, the values of latent variable y_i^* changes, while the values of μ_1 and μ_2 remain unchanged for a season. This is because past seasons' data are used to estimate these parameters, and these parameters will be discussed later in this subsection. Hence, for each match, the outcomes of home win, away win, and draw will be given certain probabilities by the ordered probit model.

On Fig. 4.1, the normal distribution curve is divided into three parts by the threshold values μ_1 and μ_2. Here, home advantage is taken into account in this model as the values of μ_1 and μ_2 indirectly include the division of chances assigned to a home win, draw and an away win for a particular season of EPL. If we use performance_in_previous_game_d, difference in the home team's performance and the away team's performance in their last games (see Table 3.1), as the explanatory variable in our model, the latent variable y_i^* will be as follows:

$$y_i^* = \beta[performance\_in\_previous\_game\_d_i] + \epsilon_i \tag{4.2}$$

As a reminder, we are assuming here that the home team and the away team are equally matched. So, $performance\_in\_previous\_game\_d = 0$. Let the ordinal response of the match outcome be $\{1, 0, -1\}$. Then, the following equations show how the latent variable and threshold values relate to each other:

$$y_i = 1 \quad \text{(home win) if} \quad y_i^* + \epsilon_i > \mu_2, \tag{4.3}$$

$$y_i = 0 \quad \text{(draw) if} \quad y_i^* + \epsilon_i < \mu_2 \quad \text{and} \quad y_i^* + \epsilon_i > \mu_1, \tag{4.4}$$

$$y_i = -1 \quad \text{(away win) if} \quad y_i^* + \epsilon_i < \mu_1. \tag{4.5}$$

To calculate the probability of a home team winning, Eq. 4.6 is used. Similarly,

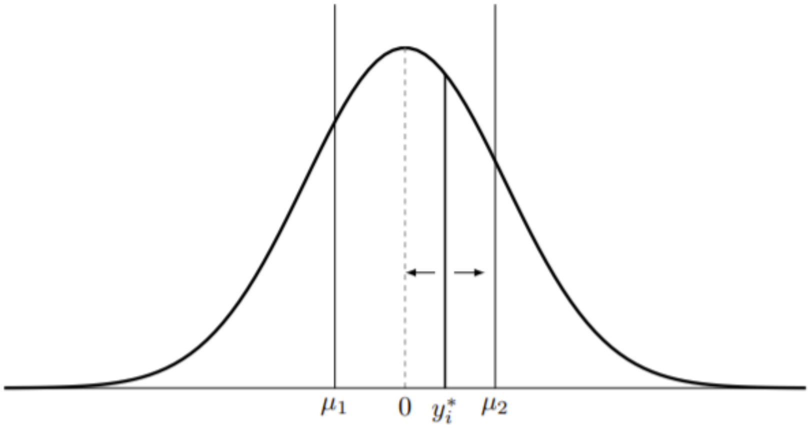


Figure 4.1: This figure shows the relationship between the latent variable y_i^*, which follows a normal distribution, and the threshold values μ_1 and μ_2. Reproduced from [31]

to calculate the probability of an away team winning, Eq. 4.7 is used, and to calculate the probability of a draw, Eq. 4.8 is used. The probability of a draw can also be calculated using the probability of a home team winning and the probability of an away team winning.

$$P[y_i = 1] = P(\epsilon_i > \mu_2 - y_i^*) = 1 - \Phi_i(\mu_2 - y_i^*), \tag{4.6}$$

$$P[y_i = -1] = P(\epsilon_i < \mu_1 - y_i^*) = 1 - \Phi_i(\mu_1 - y_i^*), \tag{4.7}$$

$$\begin{aligned} P[y_i = 0] &= P(\mu_1 - y_i^* < \epsilon_i < \mu_2 - y_i^*) \\ &= \Phi_i(\mu_2 - y_i^*) - \Phi_i(\mu_1 - y_i^*) \\ &= 1 - P[y_i = -1] - P[y_i = 1], \end{aligned} \tag{4.8}$$

where Φ_i represents the cumulative distribution function (CDF) of a standard normal distribution for observation i.

The method of maximum likelihood estimation (MLE) is used to estimate the parameters, μ_1, μ_2 and the variable coefficient(s), β, of the latent variable. Let M be dummy variables for each possible outcome ({-1,0,1}). Then, as that dummy $result_{i,m} = 1$ in case match i ends in outcome $m \in M$ and $result_{i,m} = 0$ if not. Per match, the product over all M outcome probabilities raised by $result_{i,m}$ provides the individual likelihood. The likelihood of the sample is given by the product over the individual likelihood of all matches, resulting in

$$\begin{aligned}\mathcal{L} &= \prod_{i=n}^{N} \prod_{M=-1}^{1} P\left[y_i = m\right]^{result_{i,m}} \\ &= \prod_{i=n}^{N} P\left[y_i = -1\right]^{result_{i,-1}} \prod_{i=n}^{N} P\left[y_i = 0\right]^{result_{i,0}} \prod_{i=n}^{N} P\left[y_i = 1\right]^{result_{i,1}}\end{aligned} \tag{4.9}$$

The calculations of the outcome probabilities from Eq. 4.6, Eq. 4.7, and Eq. 4.8 are incorporated into the sample likelihood of Eq. 4.9. As a result, the sample likelihood can be rewritten to

$$\begin{aligned}\mathcal{L} = \textstyle\prod_{i=n}^{N} \left[1 - \Phi_i\left(\mu_1 - y_i^*\right)\right]^{\text{result}\ i,-1} \\ \textstyle\prod_{i=n}^{N} \left[\Phi_i\left(\mu_2 - y_i^*\right) - \Phi_i\left(\mu_1 - y_i^*\right)\right]^{\text{result}\ i,0} \\ \textstyle\prod_{i=n}^{N} \left[1 - \Phi_i\left(\mu_2 - y_i^*\right)\right]^{\text{result}\ i,1}\end{aligned} \tag{4.10}$$

It is convenient to use the log-likelihood function to estimate the parameters in case MLE will be performed. Transformation of the likelihood function provides the following log-likelihood function:

$$
\begin{aligned}
l = \log \mathcal{L} = \sum_{i=n}^{N} \text{result}_{\,i,-1} \log \left[1 - \Phi_i \left(\mu_1 - y_i^*\right)\right] \\
+ \sum_{i=n}^{N} \text{result}_{\,i,0} \log \left[\Phi_i \left(\mu_2 - y_i^*\right) - \Phi_i \left(\mu_1 - y_i^*\right)\right] \\
+ \sum_{i=n}^{N} \text{result}_{\,i,1} \log \left[1 - \Phi_i \left(\mu_2 - y_i^*\right)\right]
\end{aligned}
\tag{4.11}
$$

Now, maximizing the log-likelihood with respect to the parameters, μ_1, μ_2 and the variable coefficient(s), β, of the latent variable using the Quadratic Hill Climbing method [29], gives the optimal parameters.

This is a simple example of how ordered probit model is applied in our research. However, we have fitted numerous multivariate ordered probit models for our research as shown in Section 5.

4.1.2 Artificial Neural Networks (ANN)

In the field of the sport result prediction, Artificial Neural Networks (ANNs) ”are perhaps the most commonly applied approach among ML (Machine Learning) mechanisms...”[25]. Upon introduction of input variables, the ANN model quickly recognizes and learns the patterns present in the relationship between the input variables and the output variables [31]. This, along with Mccabe and Trevathan’s success with using ANN models to predict soccer games, is the major reason why we are building ANN models for our research.

Within ANN, a set of inputs are converted into a desired output by neurons, which are mathematical functions modeled on the functioning of a

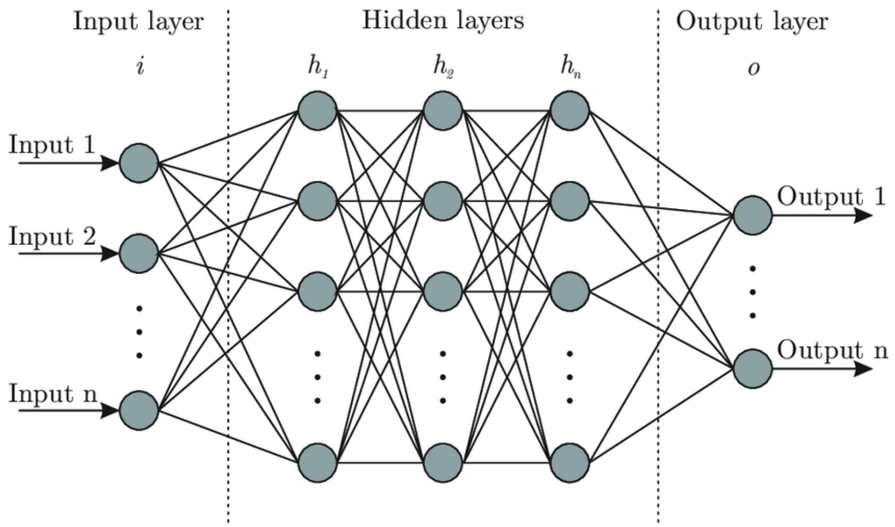


Figure 4.2: This figure shows an example of an ANN with *n* input nodes in the input layer, *n* hidden layers and *n* output nodes in the output layer. Reproduced from [1]

biological neuron [25]. For an example of an ANN structure, see Figure 4.2.

The hidden neurons adjust the weights that contribute to the final decision. Weights are the co-efficients of the equation which we are trying to resolve [11]. Positive weights increase the value of an output, whereas negative weights reduce the value of an output. To achieve high levels of predictive accuracy, the weights associated with neurons are constantly changing and adjusting. The ANN algorithm executes those changes. There is non-linearity in how the hidden neurons adjust the weights. In other words, if we look at Figure 4.2 and observe the lines connecting the nodes, we could see how complex is the process of calculating the weights. It is not a simple linear process. As a result, ”the non-linearity of the hidden neurons in adjusting the weights that contribute to the final decision makes ANN robust” [25].

To optimize the weights, there exists numerous ANN algorithms such as back-propagation, conjugate gradient descent and Levenberg-Marquardt [22]. In our case, we used nnet function from nnet package in R Studio to fit our ANN models. Here, the ”optimization is done via the BFGS method of optim” [13]. Method ”BFGS” is a quasi-Newton method (also known as a variable metric algorithm), that was published in 1970 simultaneously by Broyden, Fletcher, Goldfarb and Shanno [14]. This method uses function values and gradients to build up a picture of the surface to be optimized.

In order to update the weights, a cost function is determined, and its derivative (or gradient) with respect to each weight is estimated. The cost function is used to estimate how badly the models are performing. In other words, a cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between dependent variable and independent variables [8]. Weights are updated following the direction of steepest descent of the cost function. From the documentation of the ”nnet” package we use in R [13], it is quite unclear what cost function ”nnet” is using. So, we assume that the cost function for our ANN model is the default cost function of ”nnet”, whatever that may be. We do guess that the cost function may be a negative log-likelihood cost function, but we are unsure. However, from the documentation we learn that our ANN model is using softmax activation function, which is often placed at the output layer of a neural network. This function turns a vector of K real values into a vector of K real values that sum to 1, so that they can be interpreted as probabilities [17].

Just like the learning algorithms, there exists numerous ANNs. For our research, we are using Multi-Layer Perceptron (MLP), which is one of the most commonly used ANNs. MLP assigns a weight to each input variables based on that variable's significance in the model. This is exactly what we need because we are working with numerous variables (see Table 3.1) which must be weighted according to its contribution to the solution. Further discussions of MLPs and BFGS can be found in most neural network texts, for example [22, 27, 32].

4.2 Comparison of Match Predictions

4.2.1 Ways To Compare the Models

Predictive Accuracy

For this method, the outcome with the highest assigned probability is seen as the predicted outcome. If the predicted outcome matches the actual outcome, then that instance is called accurately predicted. This method counts the accurately predicted matches and divides this by the total number of matches predicted.

k-Fold Cross-Validation

Training a model on 80% of a dataset and testing the remaining 20% to get a predictive accuracy for the model once is not enough. This is because during that one test, we might get lucky and get a really high predictive accuracy. And when the model is fitted to a new dataset to predict the outcomes, the model could perform poorly. Similarly, during that one test, we might get unlucky and get a really low predictive accuracy. And when the model is then fitted to a new dataset to predict the outcomes, the model could perform exceptionally well. To avoid this scenario of training and testing a model only once, we use k-fold cross-validation.

Cross-validation is used in applied machine learning to estimate the skill of a machine learning model on unseen data [6]. It is called k-fold cross-validation because k refers to the number of groups that a given data is split into. We are using 10-fold cross-validation in our research because k = 10 has been found through experimentation to generally result in models ”that suffer neither from excessively high bias nor from very high variance” [30].

The general procedure is as follows; adapted from [6]:

1. Shuffle the dataset randomly.

2. Split the dataset into k groups.

3. For each unique group:

    (a) Take the group as a hold out or test data set.

    (b) Take the remaining groups as a training data set.

    (c) Fit a model on the training set and evaluate it on the test set.

    (d) Retain the evaluation score and discard the model.

4. Summarize the skill of the model using the sample of model evaluation scores. As a result, the predictive accuracy shown in Section 5 is the average predictive accuracy from 10-fold cross-validation of the models.

4.2.2 Baseline Models

Baseline models are the models to which we compare our fitted models to see how well our fitted models perform.

Simple Baseline Model: Home Teams Always Win

To begin, we let our simple baseline model to be the model that always predicts the home team winning a match. We do this because there seems to be consensus among researchers that the home team experiences an advantage over the away team [33, 38]. This phenomenon is also well observed in our dataset (see Figure 4.3), where we see that home teams won 46.3% of the time and away teams won only 28.1% of the time. So, for this baseline model, the predictive accuracy will be 46.3%.

Bookmaker Baseline Model: Based On Bookmaker's Odds

Now, we create a slightly more complex baseline model. We convert the bookmaker's (Bet365's) odds into probabilities as shown in Eq. 4.12 and Eq. 4.13. Table 4.1 gives the bookmaker's odds assigned for the match Arsenal v. Tottenham Hotspur in the season 2013-14. Eq. 4.12 and Eq. 4.13 are applied to these odds, resulting in the bookmaker's probabilities, which given in the same table.

Then, the prediction of this baseline model will be the outcome with highest bookmaker's probability. In other words, the outcome with lowest bookmaker's odds will be predicted by this baseline model. In Table 4.1, this

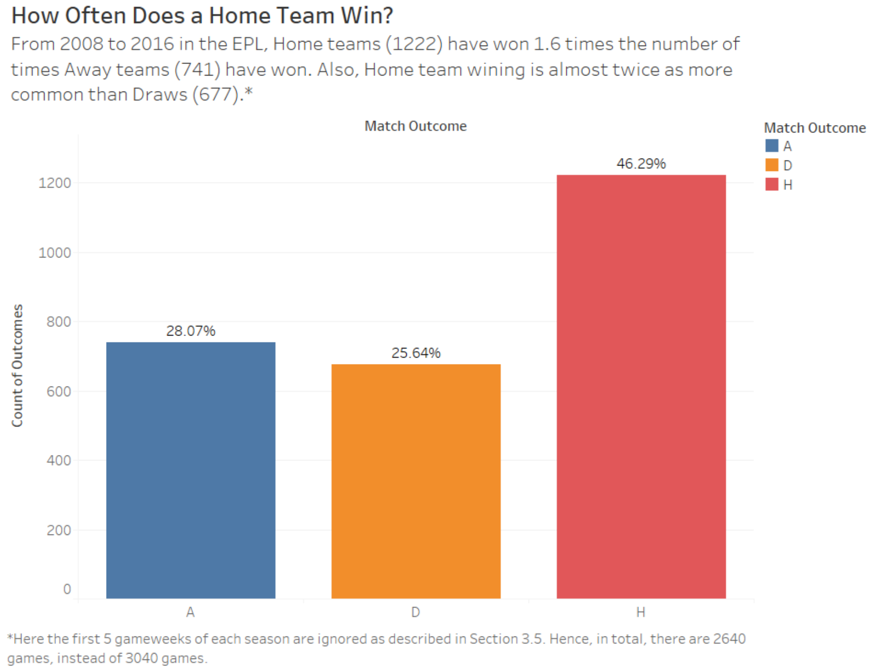


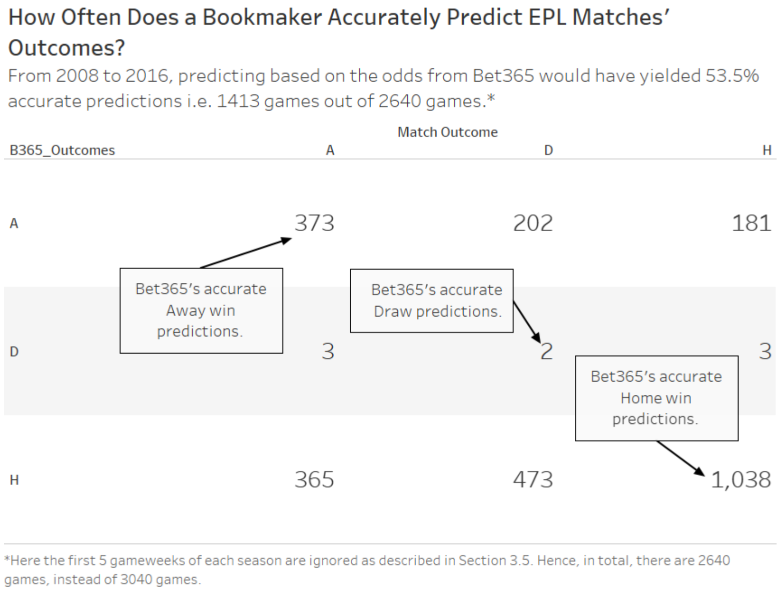
Figure 4.3: This figure shows how often a home team won in our dataset.

baseline model would have predicted home team (Arsenal) win.

$$R = \frac{1}{\Theta_{Home}} + \frac{1}{\Theta_{Draw}} + \frac{1}{\Theta_{Away}}, \tag{4.12}$$

$$\begin{aligned} P[y_i = 1] = \Omega_{Home} = \frac{(\frac{1}{\Theta_{Home}})}{R}, \\ P[y_i = 0] = \Omega_{Draw} = \frac{(\frac{1}{\Theta_{Draw}})}{R}, \\ P[y_i = -1] = \Omega_{Away} = \frac{(\frac{1}{\Theta_{Away}})}{R}, \end{aligned} \tag{4.13}$$

Here, Θ_{Home}, Θ_{Draw} and Θ_{Away} represent the bookmaker's odds for a home win, a draw and an away win, respectively. Similarly, Ω_{Home}, Ω_{Draw} and Ω_{Away}



How Often Does a Bookmaker Accurately Predict EPL Matches' Outcomes?

From 2008 to 2016, predicting based on the odds from Bet365 would have yielded 53.5% accurate predictions i.e. 1413 games out of 2640 games.*

	Match Outcome		
B365_Outcomes	A	D	H
A	373	202	181
D	3	2	3
H	365	473	1,038

Bet365's accurate Away win predictions.

Bet365's accurate Draw predictions.

Bet365's accurate Home win predictions.

*Here the first 5 gameweeks of each season are ignored as described in Section 3.5. Hence, in total, there are 2640 games, instead of 3040 games.

Figure 4.4: This figure shows how often did the bookmaker baseline model get its predictions accurate.

represent the assigned probabilities for a home win, a draw and an away win, respectively.

Arsenal	vs.	Tottenham
$\Theta_{Home} = 2.12$	$\Theta_{Draw} = 3.41$	$\Theta_{Away} = 3.62$
$\Omega_{Home} = 0.453$	$\Omega_{Draw} = 0.282$	$\Omega_{Away} = 0.356$

Table 4.1: The bookmaker's odds and bookmaker's probabilities assigned for the match Arsenal - Tottenham Hotspur in the season 2013-14.
Reproduced from [37]

Consequently, the bookmaker's baseline model will have a predictive accuracy of 53.5% as shown in Figure 4.4.

Chapter 5

Results

5.1 Model Predictions

To begin, we fitted 10-fold cross validation for both ordered probit models and ANN models. Here, we are giving examples of what an instance of fitted ordered probit model and ANN model looks like.

5.1.1 Ordered Probit Model

We fitted an ordered probit model with 22 variables. The reasons for using these specific variables are given in the Data section.

Model Summary

Table 5.1 shows the statistically significant variables in our model. Only 6 out of 22 variables are statistically significant in the model. Each of these 6 variables' p-values are less than 0.05. Thus, we reject the null hypothesis that

Table 5.1: Statistically significant variables for an instance of an ordered probit model. '+' means a positive coefficient, whereas '-' means a negative coefficient in the model.

Predictors	Sign	t-value	p-value
performance_in_previous_3_games_d	+	2.48	1.30x10^{-02} *
performance_in_previous_4_games_d	-	-2.58	9.88x10^{-03} *
goals_for_in_previous_3_games_d	-	-2.11	3.50x10^{-02} *
goals_against_in_previous_3_games_d	+	2.02	4.37x10^{-02} *
goals_against_in_previous_4_games_d	-	-2.24	2.51x10^{-02} *
B365H_1	-	-5.72	1.09x10^{-08} *

each of the variables' coefficients are zero and state that the variables are statistically significant. Table 5.1 also consists of signs of the variables' coefficients. Before we interpret them, it helps to know that in our model, home team winning (H) is in the '+' spectrum and away team winning (A) is in the '-' spectrum. We interpret them in the following way:

1. performance_in_previous_3_games_d: If the difference in home team's performance and away team's performance in their last 3 games is high, then the home team is more likely to win the game.

2. performance_in_previous_4_games_d: If the difference in home team's performance and away team's performance in their last 4 games is high,

then the away team is more likely to win the game. This seems counter-intuitive and is inconsistent to the interpretation from (1). The possible reasons for this and the following counter-intuitive interpretations are explained in Section 6.1.1.

3. goals_for_in_previous_3_games_d: If the difference in goals scored by home team and goals scored by away team in their last 3 games is high, then the away team is more likely to win the game. Again, this seems counter-intuitive because the variable measures home team's attack against away team's attack. If the difference is high then, that means home team's attack is way better than the away team's attack. So, it makes sense to say home team is more likely to win the game. However, that is not the case here.

4. goals_against_in_previous_3_games_d: If the difference in goals conceded by home team and goals conceded by away team in their last 3 games is high, then the home team is more likely to win the game. Once again, this seems counter-intuitive because the variable measures home team's defence against away team's defence. If the difference is high then, that means away team's defense is way better than the home team's defense. So, it makes sense to say away team is more likely to win the game. However, that is not the case here.

5. goals_against_in_previous_4_games_d: If the difference in goals conceded by home team and goals conceded by away team in their last 4 games is high, then the away team is more likely to win the game. Here, the interpretation meets what we normally expect.

Table 5.2: 95% Confidence Interval (CI) for the statistically significant variables in Table 5.1.

Predictors	2.5 %	97.5 %
performance_in_previous_3_games_d	0.060	0.506
performance_in_previous_4_games_d	-0.702	-0.096
goals_for_in_previous_3_games_d	-0.398	-0.015
goals_against_in_previous_3_games_d	0.005	0.363
goals_against_in_previous_4_games_d	-0.515	-0.035
B365H_1	-0.271	-0.133

6. B365H_1: If the betting odds of home team winning the game is high (i.e., the probability of home team winning is low), then the away team is more likely to win the game.

Table 5.2 shows the 95% confidence intervals for the statistically significant variables in Table 5.1. It is more evidence for why the variables are considered statistically significant. If the 95% CI does not cross 0, the parameter estimate is statistically significant. Thus, Table 5.2 result is consistent with Table 5.1.

Model Predictions

Table 5.3: The table shows a confusion matrix to compare predicted outcomes vs. actual outcomes. This is for an instance of an ordered probit model. In other words, this is a confusion matrix for a fold of 10-fold cross validation for ordered probit model.

	Predicted Outcome		
Actual Outcome	A	D	H
A	70	0	86
D	36	0	94
H	27	0	215

Here, we would like to show an instance of how the predicted outcomes from the ordered probit model compare to the actual outcomes on a test dataset. The model accurately predicts 215 home wins and 70 away victories. However, a surprising result is that the model never predicts draw as an outcome. Overall, the predictive accuracy (PA) of this instance is 54.0%.

5.1.2 ANN Model

Fitted ANN Model

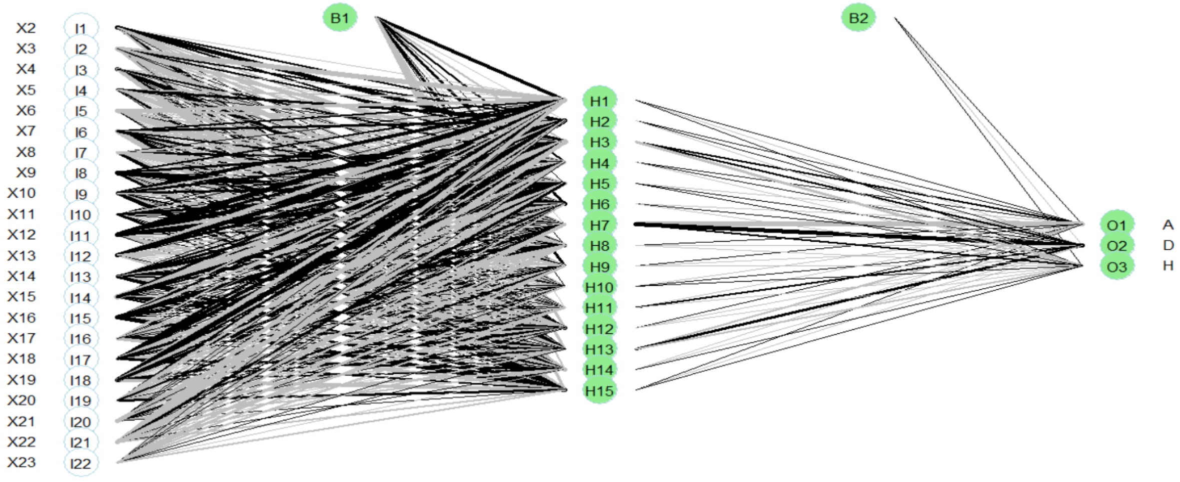


Figure 5.1: This figure shows the structure of our ANN model. There are 22 input variables (as described in Table 3.1), one hidden layer with 15 nodes, and 3 outputs. B1 and B2 are pointing to the weights being used in the model.

We fit an ANN model with 22 input variables, 1 hidden layer with 15 nodes, and 3 outputs (as depicted in Figure 5.1). We chose the hidden layer with 15 nodes because it resulted in higher predictive accuracy (PA) compared to the models with other nodes. We chose 1 hidden layer because experimenting with multiple hidden layers resulted in outputs that did not make sense at all. For instance, the outputs were numbers such as 1.3×10^{-5}, 2.3×10^{-7}, and so on, that did not sum up to 1. This was instead of the outputs that we expected: probabilities ranging from 0 to 1 that sum up to 1.

Important Variables

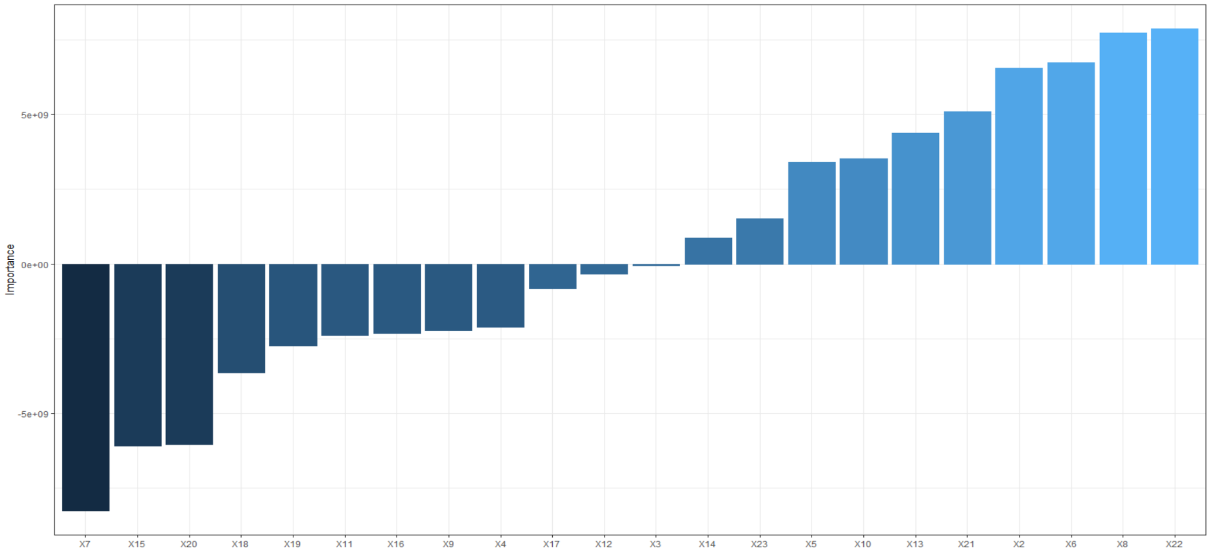


Figure 5.2: This figure shows the relative importance of the 22 input variables for the ANN model in predicting the outcomes.

Here, we talk about the important variables in our ANN model. The variable importance is evaluated using the olden function as described in [12]. The following is the interpretation of the graph:

1. X22, the betting odds of away team winning the game, has the highest positive effect on the output. This means that higher the odds of away team winning the game (i.e. lower the probability of away team winning the game), the more likely home team will win.

2. X7, the difference in home team's performance and away team's performance in their last 2 games, has the highest negative effect on the output. This means that higher the difference in home team's performance and away team's performance in their last 2 games, the

more likely away team will win. Once again, this seems counter-intuitive because the variable measures home team's performance against away team's performance in their last 2 games. If the difference is high then, that means home team's form is way better than the away team's form. So, it makes sense to say home team is more likely to win the game. However, that is not the case here. The possible reasons for this and the following counter-intuitive interpretations are explained in Section 6.1.1.

3. X8, the difference in home team's performance and away team's performance in their last 3 games, has the second highest positive effect on the output. This means that higher the difference in home team's performance and away team's performance in their last 3 games, the more likely home team will win. This interpretation meets our expectation.

4. X15 (goals_for_in_previous_5_games_d), the difference in the number of goals scored by a home team and the number of goals scored by an away team in their last 5 games, has the second highest negative effect on the output. This means that higher the difference in the number of goals scored by a home team and the number of goals scored by an away team in their last 5 games, the more likely away team will win. Once again, this seems counter-intuitive because the variable measures home team's attack against away team's attack in their last 5 games. If the difference is high then, that means home team's attack is way better than the away team's attack. So, it makes sense to say home team is more likely to win the game. However, that is not the case here.

Model Predictions

Table 5.4: The table shows a confusion matrix to compare predicted outcomes vs. actual outcomes. This is for an instance of an ANN model. In other words, this is a confusion matrix for a fold of 10-fold cross validation for ANN model.

	Predicted Outcome		
Actual Outcome	A	D	H
A	61	46	46
D	47	31	58
H	41	61	137

Here, we would like to show an instance of how the predicted outcomes from the ANN model compare to the actual outcomes on a test dataset. The model accurately predicts 137 home wins and 61 away victories. Overall, the predictive accuracy (PA) of this instance is 43.4%. The PA of this instance of ANN model is lower than the PA (54.0%) of the instance of ordered probit model (see Section 5.1.1). However, this model predicts draws, whereas the ordered probit model does not.

5.2 Model Comparisons

Here, we will compare our models' PA's to the baseline models' PA's and the best PA's we have found in our related work.

5.2.1 Ordered Probit Model Cross Validated Results

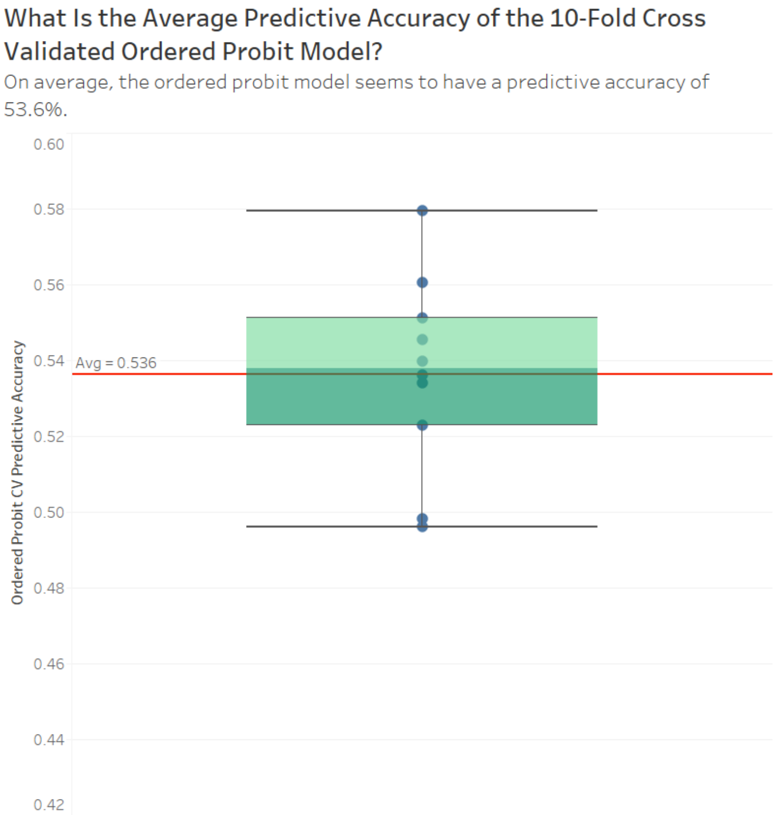
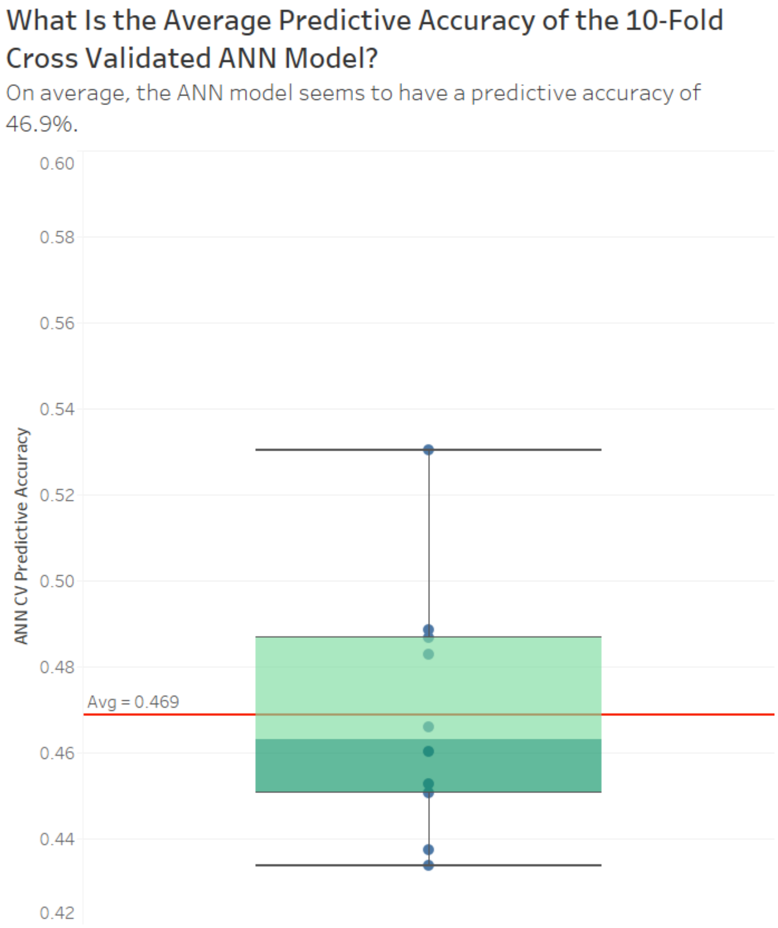


Figure 5.3: This figure shows the average predictive accuracy (PA) of the 10-fold cross validated ordered probit model.

Figure 5.3 shows the boxplot of the PA of the 10-fold cross validated ordered probit model. Here, we can observe that the average PA of the ordered probit model seems to be 53.6%. The PA ranges from 49.6% to 58.0%.

5.2.2 ANN Model Cross Validated Results



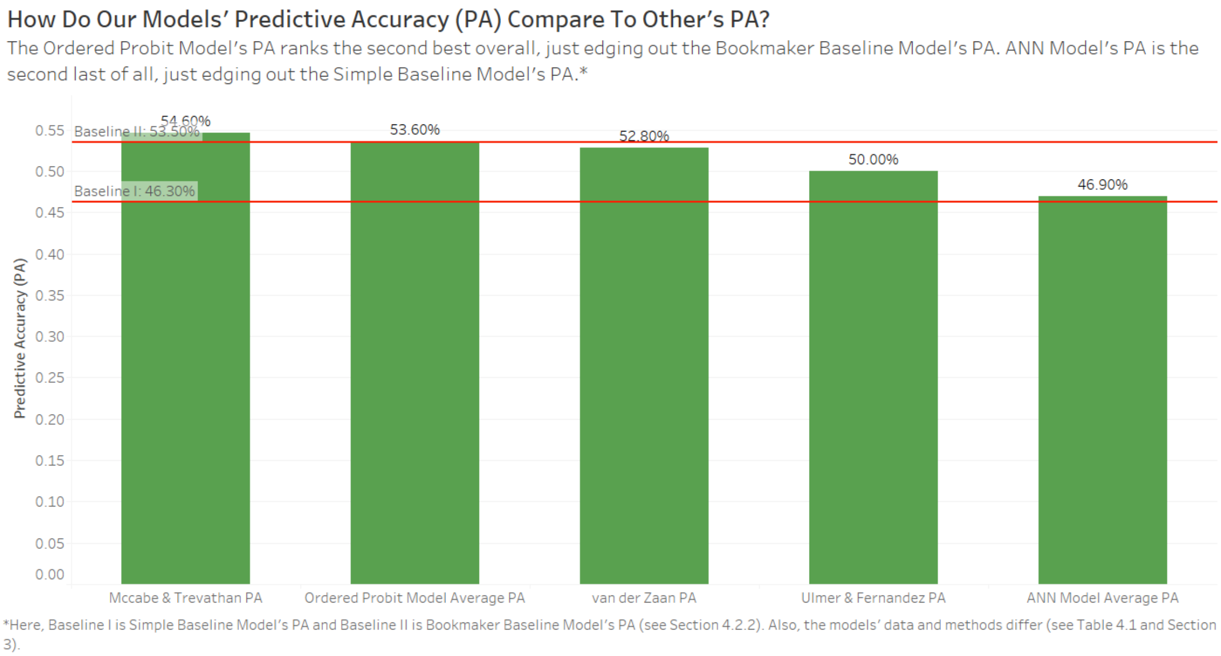
What Is the Average Predictive Accuracy of the 10-Fold Cross Validated ANN Model?

On average, the ANN model seems to have a predictive accuracy of 46.9%.

Figure 5.4: This figure shows the average predictive accuracy (PA) of the 10-fold cross validated ANN model.

Figure 5.4 shows the boxplot of the PA of the 10-fold cross validated ANN model. Here, we can observe that the average PA of the ANN model seems to be 46.9%. The PA ranges from 43.4% to 53.0%.

5.2.3 Comparing All the Models Together


How Do Our Models' Predictive Accuracy (PA) Compare To Other's PA?

The Ordered Probit Model's PA ranks the second best overall, just edging out the Bookmaker Baseline Model's PA. ANN Model's PA is the second last of all, just edging out the Simple Baseline Model's PA.*

*Here, Baseline I is Simple Baseline Model's PA and Baseline II is Bookmaker Baseline Model's PA (see Section 4.2.2). Also, the models' data and methods differ (see Table 4.1 and Section 3).

Figure 5.5: This figure shows the comparison of our models' PA to other's PA.

Figure 5.5 shows the comparison of our models' PA to the baseline models' PA and the best PA we found in our related work. We can see that the highest PA in our research is Mccabe and Trevathan's 54.6% PA [31]. The second highest PA is our ordered probit model's PA of 53.6%, which beats both simple baseline model's PA and bookmaker baseline model's PA. The last PA in our research is our ANN model's PA of 46.9%, which just beats the simple baseline model's PA.

Chapter 6

Discussion

The goal of our research is to build a model that predicts EPL soccer games with at least 55% predictive accuracy (PA) because that just betters the best PA among our related work. Overall, based on PA (see Figure 5.5), our models have done well. In particular, the ordered probit model, performed better than the baseline models and ranks second among the best models in our research. However, if we take a closer look at our fitted models, it appears that we might be better off using the bookmaker baseline model because it is simpler and the interpretation of the variables involved makes sense. That is not the case with our fitted models presented above. Let us take a closer look at our fitted models through their limitations.

6.1 Limitations

6.1.1 Interpretations of Coefficients Are Counter-intuitive

When we interpreted the signs of the predictors present in both the models, there were numerous instances (see Section 5.1.1 interpretation number 2, 3, 4, and Section 5.1.2 interpretation number 2, 4) in Section 5, when the interpretations were not what we would normally expect. We attribute this occurrence to the phenomena outlined below:

Multicollinearity

We hypothesize that the multicollinearity in our models resulted in our interpretations of coefficients not making sense. Multicollinearity is a condition when there is a significant correlation between the predictor variables [19]. Multicollinearity is a problem because it undermines the statistical significance of a predictor in a model. Ideally, the independent variables need to be independent of one another. For instance, let us say we are predicting a dependent variable y using predictors a and b, where a and b are so highly correlated that $a = b$. Let us say for every unit increase in a, y increases by 1 unit, keeping other variables constant. Similarly, for every unit increase in b, y decreases by 0.26 unit, keeping other variables constant. Then, the model will look like the following:

$$y = a - 0.26b \tag{6.1}$$

However, since $a = b$, the following model is the exact same model as the model shown in Equation 6.1:

$$y = -a + 1.84b \tag{6.2}$$

If that is the case, then we have two exact models, but the interpretations for the coefficients are different. This does not make sense.

One way to overcome this challenge is to only include predictors in a model that are not highly correlated. Table 6.1 shows VIF (Variance Inflation Factor) values for each of the predictors in our models. The VIF values are calculated and interpreted using methods described in [2, 18]. Normally, predictors with VIF values greater than 5 tend to cause multicollinearity. In other words, a predictor with a VIF value greater than 5 is likely to be highly correlated with other predictors in the model. Hence, it might be better to exclude that predictor from the model. For instance, in our model, performance_in_previous_4_games_d could highly correlate with goals_for_in_previous_4_games_d because if a home team has been winning more games than an away team in their last 4 games, then the home team is likely to have scored more goals than the away team in their last 4 games.

Table 6.1: The table shows VIF (Variance Inflation Factor) values for each of the predictors used in our models. 17 out of 22 variables seem to have VIF values greater than 5. So, there is a big problem of multicollinearity in our fitted models.

Predictors	VIF Test Values
Goals_against_so_far_d	4.46
Goals_for_so_far_d	6.05
overall_performance_so_far_d	13.06
cum_performance_so_far_d	3.51
performance_in_previous_game_d	7.28
performance_in_previous_2_games_d	15.07
performance_in_previous_3_games_d	23.79
performance_in_previous_4_games_d	34.66
performance_in_previous_5_games_d	24.56
goals_for_in_previous_1_game_d	4.99
goals_for_in_previous_2_games_d	10.12
goals_for_in_previous_3_games_d	16.31
goals_for_in_previous_4_games_d	23.97
goals_for_in_previous_5_games_d	16.95
goals_against_in_previous_1_game_d	4.15
goals_against_in_previous_2_games_d	8.39
goals_against_in_previous_3_games_d	12.93
goals_against_in_previous_4_games_d	17.47
goals_against_in_previous_5_games_d	12.51
B365H_1	4.30
B365D_1	11.38
B365A_1	16.10

Therefore, we experimented with using only 4 predictors shown in Table 6.2 in both of our ordered probit model and ANN model. We chose these 4 because they were statistically significant variables in the last fitted ordered probit model (see Table 5.1). However, this is not an ideal choice because it is strange to use statistical significance from the model in which we are trying to correct the misleading results. But, we did this because we were short on time and thought this was a good starting place to experiment. When we saw a great improvement in the ANN model's predictive accuracy (PA) and only 0.1% decrease in the ordered probit model's PA (see Figure 6.2), we decided to stick with these predictors. Additionally, we have suggested further studies in Section 7 (recommendation number 4) to experiment with different combinations of predictors to see if the model's PA improves or not.

From Table 6.2, we see that all the VIF test values for the predictors are below 5. So, the multicollinearity has been reduced.

Table 6.2: All the VIF test values are below 5. Thus, there is no problem of multicollinearity while fitting our models with these predictors only.

Predictors	VIF Test Values
performance_in_previous_4_games_d	4.08
goals_for_in_previous_4_games_d	2.48
goals_against_in_previous_4_games_d	2.20
B365H_1	1.01

Randomness In Soccer

Now, let us observe what happens to the signs, statistical significance, and variable importance of these predictors when they are present in an ordered probit model and an ANN model.

Table 6.3: The table shows signs and p-values for the predictors used while fitting a new ordered probit model. The betting odds variable is the only variable that is statistically significant.

Predictors	Sign	p-value
performance_in_previous_4_games_d	+	2.75x10^{-01}
goals_for_in_previous_4_games_d	-	6.08x10^{-01}
goals_against_in_previous_4_games_d	-	7.03x10^{-01}
B365H_1	-	3.11x10^{-59} *

From Table 6.3, we see that the p-value of B365H_1 is < 0.05, so we reject the null hypothesis that its coefficient is 0. Hence, it is statistically significant. Also, we observe that the sign interpretation for the variable make sense: If the betting odds of home team winning the game is high (i.e., the probability of home team winning is low), then the away team is more likely to win the game.

However, the sign interpretations for performance_in_previous_4_games_d, goals_for_in_previous_4_games_d, and goals_against_in_previous_4_games_d do not matter. This is because their corresponding p-values are > 0.05: so, we cannot reject that their corresponding coefficients are 0. Hence, the 3 predictors are not statistically significant. Even though we have taken care of the multicollinearity, not all predictors are statistically significant. We attribute this occurrence to the randomness in soccer. For instance, in soccer, we expect a team that has won their last 5 games to beat a team that has lost all its last 5 games. However, in soccer (especially in EPL), there are numerous instances when the underdogs do the unexpected and beat the favorites. Even the model seems to be fooled by the randomness of soccer.

Likewise, if we look at the variables' importance of ANN model (see Figure 6.1), we observe a similar occurrence.

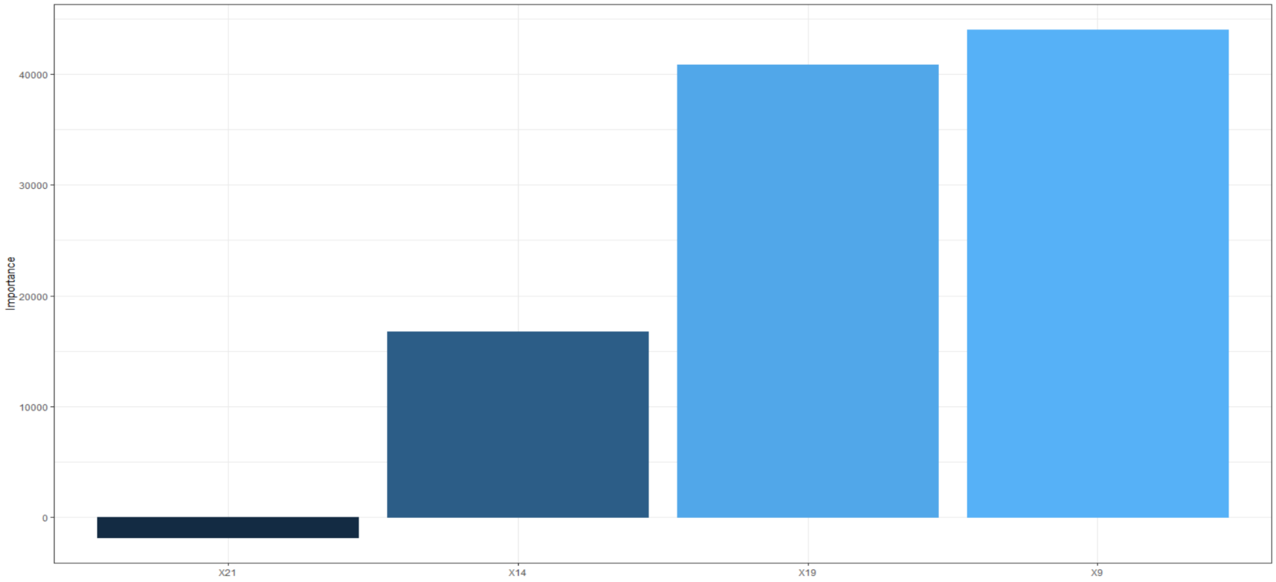
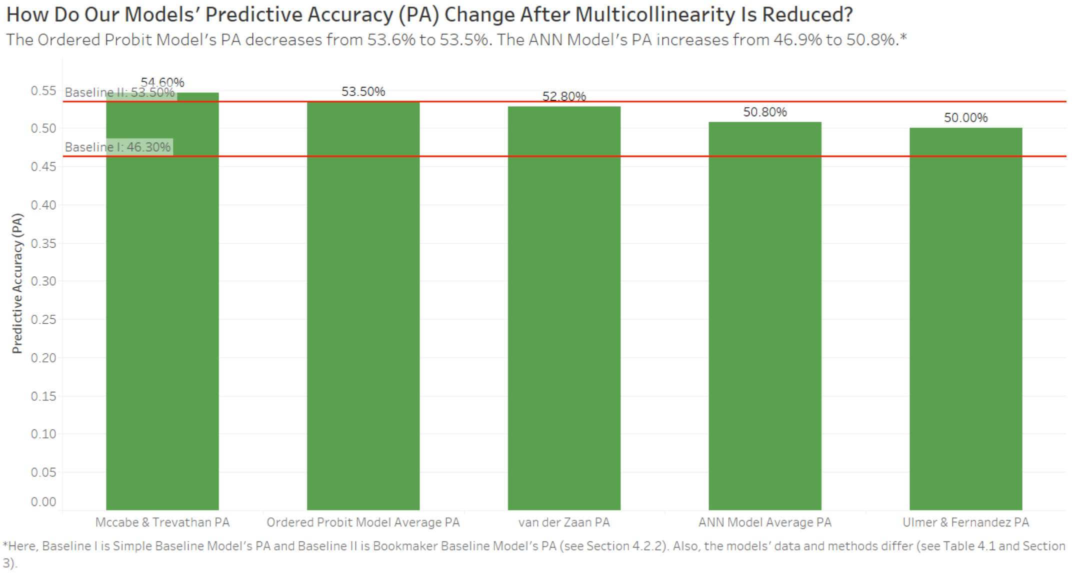


Figure 6.1: This figure shows updated variables' importance for an ANN model with 4 predictors.

Here, the signs of X9 (performance_in_previous_4_games_d, +), X14 (goals_for_in_previous_4_games_d, +), and X19 (goals_against_in_previous_4_games_d, +) do not matter because none of them are statistically significant. However, the sign for X21 (B365H_1, -) matters and makes sense.

This rarity of finding just a few statistically significant predictor, even after multicollinearity being significantly reduced, can be attributed to the randomness in soccer.


How Do Our Models' Predictive Accuracy (PA) Change After Multicollinearity Is Reduced?

The Ordered Probit Model's PA decreases from 53.6% to 53.5%. The ANN Model's PA increases from 46.9% to 50.8%.*

*Here, Baseline I is Simple Baseline Model's PA and Baseline II is Bookmaker Baseline Model's PA (see Section 4.2.2). Also, the models' data and methods differ (see Table 4.1 and Section 3).

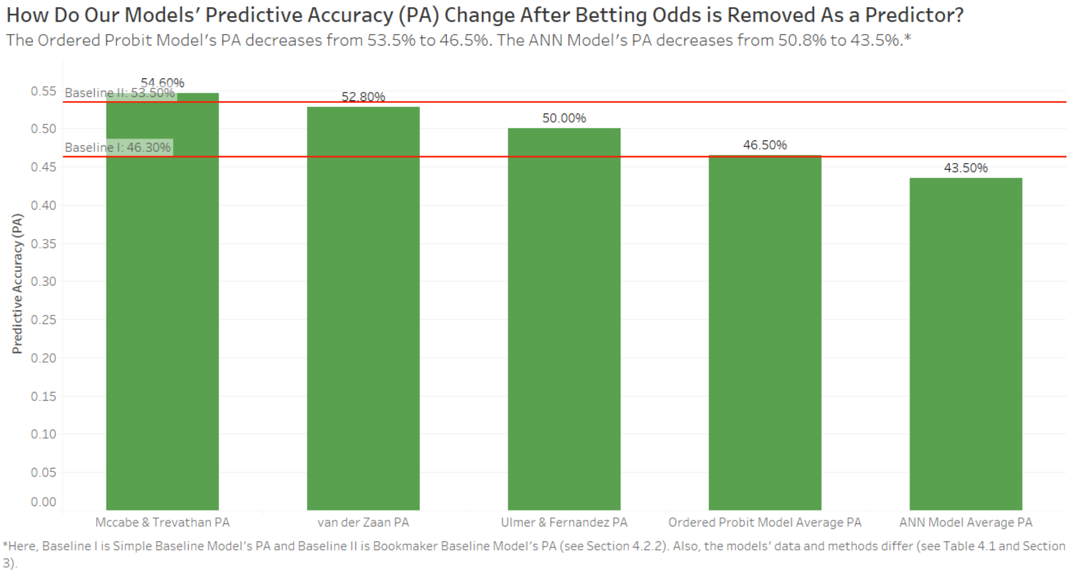
Figure 6.2: This figure shows how our models' PA change after multicollinearity is reduced by using only 4 predictors to predict the outcomes of a game.

On the other hand, the significant reduction in multicollinearity has two major benefits. Firstly, the models are much simpler in terms of number of predictors. Even though the average PA for probit model decreases from 53.6% to 53.5% (see Figure 6.2), we prefer the latter one because the latter one has only 4 predictors compared to the former one, which has 22 predictors. Secondly, the average PA for ANN model significantly rises from 46.9% to 50.8%. We assume that reducing the multicollinearity in our ANN model significantly improved its average PA.

6.1.2 Our Models Are Heavily Reliant on Betting Odds

From Figure 6.2, we see that our ordered probit model has the same predictive accuracy as the bookmaker baseline model. From Table 6.3, we see that the only variable that is statistically significant in our model is B365H_1, a bookmaker's odds for home team winning. So, we ask ourselves if our models are heavily reliant on betting odds. In other words, what happens to our models' average PA when B365H_1 is removed as a predictor?

The answer is clearly depicted in Figure 6.3. We see that both of our models' average PAs decrease significantly. In fact, the ANN model's average PA is ranked last among the PA in our research and the ordered probit model's average PA barely passes the simple baseline model. This shows that our models are heavily reliant on betting odds information from a bookmaker while predicting match outcomes. People can consider this a limitation because they would question our use of a bookmaker's model's outcomes as one of our predictors in our models. But we believe that in order to predict



How Do Our Models' Predictive Accuracy (PA) Change After Betting Odds is Removed As a Predictor?

The Ordered Probit Model's PA decreases from 53.5% to 46.5%. The ANN Model's PA decreases from 50.8% to 43.5%.*

*Here, Baseline I is Simple Baseline Model's PA and Baseline II is Bookmaker Baseline Model's PA (see Section 4.2.2). Also, the models' data and methods differ (see Table 4.1 and Section 3).

Figure 6.3: This figure shows how our models' PA change after betting odds for home team winning is removed as one of our 4 predictors to predict the outcomes of a game.

soccer games' outcomes even more accurately, we need to experiment with every variable available.

6.1.3 Unable To Generalize Our Results For Other Soccer Competitions

The focus of our research is to predict soccer games in the English Premier League (EPL). Thus, it makes sense that our results can only be generalized for EPL. We cannot generalize our results for other soccer leagues and competitions, although it would be interesting to observe what kind of results can be obtained conducting similar research for other soccer leagues.

6.2 Strengths Of Our Analysis

Now that we have presented our limitations, let us talk about the strengths of our analysis:

1. Honest PA: We use cross-validation in our analysis so that we can use our whole cleaned dataset to train our models and test our models. Also, we want to make sure that the PA we are getting is not happening by chance (good luck or bad luck), but that is an average of numerous instances of PA we obtain using 10-fold cross validation. In other words, if we are training a model on 80% of the data and testing it once on the 20% to obtain PA, then this is only one instance of getting a PA. The PA can be very high or very low due to chance. However, if we use 10-fold cross validation, we will obtain 10 different PA's on 10 different instances of how train and test data are split. Then, we will average the 10 different PA's to get an average PA that is less likely to overestimate or underestimate the PA of a model. Hence, one of the strengths of our analysis is that our PAs are slightly more "honest".

2. Preferred Models Have Higher PA Than the simple baseline model: The models we preferred are the ordered probit model and the ANN model with 4 predictors (see Figure 6.2 for the model's PA). An achievement of our analysis is that both models show PA higher than the simple baseline model's PA. Additionally, the ordered probit model's PA does tie with the bookmaker baseline model's PA at 53.50%.

3. Multicollinearity Is Considered: Another strength of our analysis is that

multicollinearity is considered in our models using VIF test values. Then, it is reduced by removing the predictors that are highly correlated to each other or using only one of such predictors. This reduction in multicollinearity clearly improves the simplicity and the average PA of our models (as shown in Figure 6.2).

4. Preferred Models Are Simple: Consequently, our preferred models are simpler and easier to interpret. They only have 4 predictors, but their PA are among the best in our related work.

Chapter 7

Conclusion

In conclusion, we will summarize our findings in terms of George E.P. Box's famous quote: "All models are wrong, but some are useful" [24]. This means that all models are wrong in that they cannot model all the complexities of reality. However, some models can still be of use to help us simplify the complexities of the reality and understand reality a little bit better. All the models we have fitted in our research are "wrong" because they do not consider all the complexities of the reality. However, some of them are "useful" to help us better understand predicting soccer games. For instance, Mccabe and Trevathan's ANN model and our ordered probit model with 22 inputs (see Figure 5.5) are less "wrong" than other models in that they have higher predictive accuracy (PA) of 54.6% and 53.6% respectively. However, both models are complex to interpret. On the other hand, our ordered probit model with 4 inputs (see Figure 6.2) is slightly more "wrong" than the previous two models in that its PA is 53.5%; however, it is much simpler and easier to interpret.

To make our models even more useful and/or less wrong, we recommend the following future work, which we would have done if we had more time:

1. Tweak Our Models' Parameters: Future research could examine the effects of changing numerous models' parameters. Specifically, future research could look at building an ANN model with 3 hidden layers and 10 nodes for each hidden layer, just like Mccabe and Trevathan did. The reason why is that Mccabe and Trevathan's model does a really good job predicting soccer games without using betting odds as one of their predictors. Mccabe and Trevathan's model and our ANN model (see Section 5.1.2) have similar input variables. However, our models differ in the number of hidden layers and nodes used. This could be a reason why Mccabe and Trevathan's model predicts with 54.6% PA and our ANN model predicts with only 46.9% PA. We attempt to build a similar ANN model as they did. However, the output from the model is not what we expected. Hence, we build a simpler ANN model that gives sensible outputs. So, future research could investigate this issue.

2. Use a Different Metric To Compare the Models: Future research might apply a different metric to compare the usefulness of the models. For instance, instead of using PA, one could use likelihood to compare the models [3]. Usually, likelihood will either reward or punish a model for assigning higher probabilities for an accurate outcome or an inaccurate outcome, respectively. So, when two models have similar PA, likelihood can be used to differentiate them.

3. Do Data Transformation and Use Different Combinations of Predictors:

Future studies could investigate transforming input variables based on their relationship with the output variable. Also, different combinations of predictors can be used to see how well a model predicts.

4. Build Models With Other Confounding Variables: Future studies could examine including other confounding variables in a model. Other confounding variables include Euro Club Index (ECI) [10], whether a match is a regional derby or not [39], effect of national cup matches [28], etc. Additionally, we search for the required data to predict games for the current season of EPL (2020/21) so that we can see how our model will fare in the current climate. Unfortunately, we are unable to find the data that we are looking for. Also, cleaning the available data to the data we desire will take more time than the nature of this project allows.

Overall, future studies could use this framework and branch out to build predictive models for other soccer leagues and competitions.

Bibliography

[1] Artificial neural network architecture (ann i-h 1-h 2-h n-o). — download scientific diagram. https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051. (Accessed on 03/09/2021).

[2] Calculating vif for ordinal logistic regression & multicollinearity in r - stack overflow. https://stackoverflow.com/questions/63800411/calculating-vif-for-ordinal-logistic-regression-multicollinearity-in-r. (Accessed on 03/16/2021).

[3] Diagnostic accuracy – part 2¡br /¿predictive value and likelihood ratio. https://acutecaretesting.org/en/articles/diagnostic-accuracy--part-2brpredictive-value-and-likelihood-ratio#:~:text=The%20advantage%20of%20likelihood%20ratios,to%20your%20own%20patient%20population. (Accessed on 03/23/2021).

[4] European soccer database — kaggle.

https://www.kaggle.com/hugomathien/soccer. (Accessed on 02/25/2021).

[5] Football's beauty is its unpredictability, says arsene wenger. https://www.irishexaminer.com/sport/soccer/arid-20321805.html. (Accessed on 12/02/2020).

[6] A gentle introduction to k-fold cross-validation. https://machinelearningmastery.com/k-fold-cross-validation/#:~:text=Cross%2Dvalidation%20is%20primarily%20used,the%20training%20of%20the%20model. (Accessed on 03/10/2021).

[7] How to use data scaling improve deep learning model stability and performance. https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-sca (Accessed on 02/27/2021).

[8] Machine learning fundamentals (i): Cost functions and gradient descent — by conor mcdonald — towards data science. https://towardsdatascience.com/machine-learning-fundamentals-via-linear-regression-41a5d11f5220. (Accessed on 03/09/2021).

[9] Making smart decisions when you don't have all the facts with annie duke — the science of success podcast. https://www.successpodcast.com/show-notes/2018/7/18/making-smart-decisions-when-you-dont-have-all-the-facts-with-annie-duke.

[10] Methodology - euro club index. https://www.euroclubindex.com/methodology/#:~:text=The%20Euro%20Club%20Index%20(ECI,of%20playing%20strengths%20in%20time. (Accessed on 03/23/2021).

[11] Neural networks bias and weights. understanding the two most important... — by farhad malik — fintechexplained — medium. https://medium.com/fintechexplained/neural-networks-bias-and-weights-10b53e6285da. (Accessed on 03/09/2021).

[12] Neuralnettools: Visualization and analysis tools for neural networks. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6262849/. (Accessed on 03/13/2021).

[13] nnet function — r documentation. https://www.rdocumentation.org/packages/nnet/versions/7.3-15/topics/nnet. (Accessed on 03/09/2021).

[14] optim function — r documentation. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/optim. (Accessed on 03/09/2021).

[15] Predictive modeling: The only guide you need — microstrategy. https://www.microstrategy.cn/us/resources/introductory-guides/predictive-modeling-the-only-guide-you-need. (Accessed on 02/24/2021).

[16] Premier league competition format & history — premier league. https://www.premierleague.com/premier-league-explained. (Accessed on 12/02/2020).

[17] Softmax function definition — deepai. https://deepai.org/machine-learning-glossary-and-terms/softmax-layer#:~:text=The%20softmax%20function%20is%20a,can%20be%20interpreted%20as%20probabilities. (Accessed on 03/25/2021).

[18] Vif function — r documentation. https://www.rdocumentation.org/packages/regclass/versions/1.6/topics/VIF. (Accessed on 03/16/2021).

[19] What is multicollinearity and how to remove it? — by sharoon saxena — analytics vidhya — medium. https://medium.com/analytics-vidhya/what-is-multicollinearity-and-how-to-remove-it-413c419de2f. (Accessed on 03/19/2021).

[20] Zacharias Andreou. Who Will Be Crowned King of Europe? A Predictive Model for the Uefa Champions League. (Accessed on 09/10/2020).

[21] E. Ben-Naim, F. Vazquez, and S. Redner. What is the most competitive sport. *arXiv: Data Analysis, Statistics and Probability*, 2005.

[22] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., USA, 1995.

[23] Jack Blundell. Numerical algorithms for predicting sports results. 2009.

[24] George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[25] Rory P. Bunker and Fadi Thabtah. A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1):27 – 33, 2019.

[26] Mark J. Dixon 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):265–280, 1997.

[27] Roger Fletcher. Practical methods of optimization. `https://archive.org/details/practicalmethods0000flet/page/n9/mode/2up`, 1987. (Accessed on 03/09/2021).

[28] John Goddard and Ioannis Asimakopoulos. Modelling football match results and the efficiency of fixed-odds betting. *Journal of Forecasting*, 23(1):51–66, 2004.

[29] Stephen M. Goldfeld, Richard E. Quandt, and Hale F. Trotter. Maximization by quadratic hill-climbing. *Econometrica*, 34(3):541–551, 1966.

[30] Witten D. Hastie T. Tibshirani R. James, G. *An Introduction to Statistical Learning*. 2013.

[31] Alan Mccabe and Jarrod Trevathan. Artificial intelligence in sports prediction. pages 1194–1197, 04 2008.

[32] Lúcio F. C. Pessoa. Multilayer perceptrons versus hidden markov models: Comparisons and applications to image analysis and visual pattern recognition. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.1843&rep=rep1&type=pdf. (Accessed on 03/09/2021).

[33] Richard Pollard and Gregory Pollard. Home advantage in soccer: a review of its existence and causes. *International Journal of Soccer and Science Journal*, 3, 01 2005.

[34] David Sally and Chris Anderson. *The Numbers Game: Why Everything You Know About Soccer Is Wrong*. 06 2013.

[35] J. Snyder. What actually wins soccer matches: Prediction of the2011-2012 premier league for fun and profit. 2013.

[36] B. Ulmer and M. Fernandez. Predicting soccer match results in the english premier league. 2014.

[37] Tim van der Zaan. Predicting the outcome of soccer matches in order to make money with betting. 2017.

[38] Roger C. Vergin and John J. Sosik. No place like home: an examination of the home field advantage in gambling strategies in nfl football. *Journal of Economics and Business*, 51(1):21–31, 1999.

[39] Albina Yezus. Predicting outcome of soccer matches using machine learning. *Saint-Petersburg University*, 2014.