FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Machine Learning-Based Prediction of Football Match Statistics
Student:	Ondřej Herman
Supervisor:	Rodrigo Augusto da Silva Alves, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Instructions

Football is the most popular sport in the world, with 1.5 billion people having watched the 2022 World Cup final, according to the International Federation of Football (FIFA). With the growing interest in sports betting on football, various predictive models have emerged in related literature for match outcome prediction. However, little exploration has been done on predicting statistics of a football match, such as the number of yellow cards, corners, or shots. These statistics are not only essential for predicting match outcomes but also for better understanding teams from a technical standpoint. This bachelor thesis aims to develop a machine learning-based method for predicting football match statistics. Specifically, the thesis will cover the following objectives:

1) Collect open datasets of football statistics from at least four different leagues.
2) Conduct a literature review of existing methods for predicting match statistics.
3) Propose a machine learning-based method for predicting football match statistics.
4) Compare the proposed method's results with at least four baseline models.
5)Present the results in both textual and graphical formats to improve data visualization methods for football match statistics.

Overall, this thesis aims to contribute to the development of machine learning-based methods for predicting football match statistics, which can aid in improving match analysis and decision-making for teams, analysts, and betting enthusiasts.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 27 February 2023 in Prague.

Bachelor's thesis

MACHINE LEARNING-BASED PREDICTION OF FOOTBALL MATCH STATISTICS

Ondřej Herman

Faculty of Information Technology
Katedra aplikované matematiky
Supervisor: Rodrigo Augusto da Silva Alves, Ph.D.
May 11, 2023

Contents

List of Figures

List of Tables

List of code listings

In the first place I would like to sincerely thank my thesis supervisor Rodrigo Augusto da Silva Alves, Ph.D., for very sincere guidance, lot of important advices and for accepting to supervise my thesis after I came back from Erasmus and almost was not able to find a supervisor in time. Also, I must thank to my girlfriend for always supporting me and being there for me. Last but not least I would like to thank my family for all the backing they gave me throughout my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stip- ulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to con- clude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Praze on May 11, 2023 .....

Abstract

Football, the most widely played and followed sport globally, captivates billions of fans worldwide. The significance of predicting match outcomes has garnered attention from statisticians, machine learning researchers, and avid bettors alike. However, while substantial progress has been made in machine learning for outcome prediction, relatively little focus has been placed on forecasting the statistical aspects of the sport. This study aims to address this gap by exploring machine learning methods to analyze and estimate various match statistics as regression problems. Specifically, I investigate six statistics: corners, shots, shots on target, fouls, yellow cards, and red cards. By conducting experiments on four datasets from different football leagues, I evaluate the performance of eight models. My findings reveal that different methods adapt better to certain statistics, and also that some statistics exhibit different behaviors across leagues. Additionally, I observe that certain features, such as the number of corners or shots, are more predictable due to their higher occurrence rates during matches compared to the number of cards.

Keywords Football Statistics Prediction, Machine learning, Sport Analytics

Abstrakt

Fotbal, nejrozšířenější a nejsledovanější sport na světě, poutá pozornost miliard fanoušků po celém světě. Předpovídání výsledků zápasů se dostalo pozornosti statistiků, výzkumníků strojového učení a nadšených sázkařů. Nicméně, zatímco byl učiněn podstatný pokrok ve strojovém učení pro předpověď výsledků zápasů, relativně malý důraz byl kladen na předpovídání statistických aspektů daných zápasů. Tato studie si klade za cíl řešit tento nedostatek prozkoumáním metod strojového učení k analýze a odhadu různých statistik jako regresních problémů. Konkrétně zkoumám šest statistik: rohy, střely, střely na branku, fauly, žluté karty a červené karty. Prováděním experimentů na čtyřech datových souborech z různých fotbalových lig postupně porovnám a vyhodnotím výsledky osmi různých modelů. Má zjištění ukazují, že různé metody se více hodí na určité statistiky a také, že různé statistiky vykazují různé chování v různých ligách. Kromě toho jsem si všiml, že určité vlastnosti, jako je počet rohů nebo střel, jsou předvídatelnější díky jejich vyšší míře výskytu během zápasů ve srovnání například s počtem karet.

Klíčová slova Predikce Fotbalových Statistik, Machine Learning, Sportovní Analýza

List of abbreviations

UEFA	*Union Européenne de Football Association*
FIFA	*Fédération Internationale de Football Association*
EPL	English Premier League (English 1st football league)
SA	Serie A (Italian 1st football league)
GBL	German Bundesliga (German 1st football league)
SLL	Spanish La Liga (Spanish 1st football league)
PR	PoissonRegressor
Ridge	RidgeRegression
RF	Random Forest
XGB	XGboost
MLP	Multi-Layer Perceptron
ANN	Artificial Neural Network
SVD	Singular Value Decomposition
SVDpp	extended Singular Value Decomposition
BO	BaselineOnly model
OMIC	Orthogonal Inductive Matrix Completion
HAD	Home, Away, Draw
VAR	Video assistant Referee
O/U	Over/Under
ON	Odds Number

Chapter 1
Introduction

1.1 Introduction

Football is the most played and followed sport on this planet. For instance, FIFA (the French abbreviation for International Association Football Federation) reportedly claimed that 1.5 billion people watched the 2022 World Cup final, leaving other big sports events at least 1 billion viewers behind [1]. Therefore, it is not a surprise that football is also the biggest in terms of collected data because of the higher attention it receives and the higher demand for knowledge about past and upcoming games.

Nowadays, a lot of teams invest a great deal of money into various modern data analytics about their team's performance. Also, a huge amount of money is in the football betting market, which is reportedly valued at 84.66 billion USD in 2022 and is likely to keep on growing [2].

Predicting the outcome of a match is a popular topic among both researchers in the fields of statistics and machine learning as well as betting enthusiasts. While the outcome is certainly the most sought-after information before a match, there are other interesting statistics that can provide insights into how the game will unfold. The organization teams of football clubs can use estimations of these statistics, such as the number of goals or corners, to change their tactics prior to a game. This information can also be very helpful in predicting which team is likely to win or whether the match is likely to end in a draw. This information can also be used by the the organization teams of the football clubs to adjust their tactics before a match. Therefore, having good estimations of these other statistics can be valuable in making more accurate predictions for the match's outcome.

This work focuses on analyzing and estimating football match statistics, namely (1) corners, (2) shots, (3) shots on target, (4) fouls, (5) yellow cards, and (6) red cards. Unlike predicting whether the match will end in a home team victory, an away team victory, or a draw (HAD), estimating these match statistics seems like a much less explored area. From a betting point of view, one can win the same amount of money by betting on these statistics as one would by betting on the outcomes if the bets were correct. It can be done by making estimations of some statistics and evaluating the risks of the odds given by the bookmakers.

Various machine learning techniques have been developed and used in the field of football prediction in recent years. From the simpler Bayesian models and Decision Trees, to more complicated models like Artificial Neural Networks (ANNs), all models showed their strengths and weaknesses. This will all be analyzed in the Chapter 2 - Literature Review. In this thesis, I propose the use of standard machine learning techniques for regression as well as matrix completion. Both of these approaches are subfields of artificial intelligence, but they have different objectives and approaches. Standard machine learning is the process of training a model to make predictions or decisions based on input data. The goal of machine learning is to learn patterns

and relationships in the data that can be used to make accurate predictions on new, unseen data. In standard machine learning, the data usually consists of input features and a target variable that the model is trying to predict. Matrix completion (or factorization) is a specific approach to machine learning that is used to predict missing values in a matrix. It is often used in recommender systems, where the matrix represents the ratings or preferences of users for items. The goal is to predict the missing ratings so that personalized recommendations can be made to users. A similar approach will be used in this thesis, but it will have to be adjusted for the prediction of football statistics. Home teams will be thought of as users, and away teams will be thought of as items in the input matrix. This is described later in Table 3.1.

1.2 Football

Football is a very well-known sport and is played all around the world. The English Premier League (EPL), Italian Serie A (SA), German Bundesliga (GBL) and Spanish La Liga (SLL) are four of the top ten most watched leagues in the world [3]. These four leagues are the four best leagues according to UEFA Country coefficients [4]. All these leagues follow the same unified rules of football.

Table 1.1 UEFA Country Coefficients for Top European Leagues (22/23)

League	UEFA Country Coefficients
English Premier League	107.712
Spanish La Liga	92.141
Italian Serie A	82.356
German Bundesliga	79.926

1.2.1 Rules

Football is a team sport and owes its tremendous popularity to its simplicity. The game is played on a football pitch with a size of 105 by 68 meters. Standard playing time is 90 minutes, and it is divided into two halves, each having a duration of 45 minutes. There are two goals on the pitch, and the team that puts the ball into the opposing team's goal more times wins. Each team has ten players on the field and one goalkeeper. The goalkeeper can play with his hands in a small box in front of his goal called the *box*. No other player can touch the ball with his hand; otherwise, it is considered foul play, resulting in the opposing team getting the ball.

- The game begins with one team's kickoff from the center of the pitch.
- Whenever the ball crosses the longer border of the pitch, the opposing team of the team that had the last touch of the ball throws the ball in from the line.
- If the ball goes out of play with the last touch being from an attacking player on the base line, then it is a goal kick. If the last touch was the defending player's, it is a corner kick for the attacking team.
- When a player without a ball tackles another player unfairly, or any player touches the ball with his hand illegally, or a player acts unsportsmanlikely, the referee can judge it as a foul, and the opposing team gets the ball. The referee can show a yellow or even a red card for these acts if he considers them serious enough.
- The offside rule states that an attacking player must have at least two opposing players, including the goalkeeper, between him and the opposing goal during a move in order for a pass to be delivered to him.

In every professional football game, there is one referee and two assistant referees. The assistant referees are positioned along the sidelines, judging the game from there and controlling offsides. The main referee keeps a close distance from the ball and has the final say in all situations. In recent years, the *Video Assistant Referee* (VAR) was introduced to judge offsides, fouls in the box because they can lead to penalties and to review goals. Despite the VAR, the referee's style of officiating the game is still a big factor in how many fouls and cards will be given. Some referees tend to judge some acts as fouls, while others do not.

1.2.2 League

A standard professional league consists of X number of teams, with each team facing every other team twice. One time at home and one time at the other team's home stadium. This means that every team plays *X-1* matches at their home stadium and the same amount at other teams' stadiums, with one at each of them. The home advantage is a non-negligible factor in football. The home team usually has much more fans supporting them at the home stadium and also plays on its pitch, where the players should feel more comfortable. The team that wins the match after the standard playing time earns three points in the league table, with the other one earning zero. In case of a draw both teams get 1 point. Two or more teams having the same number of points earned in the league table is a very common phenomenon. Usually the *goal difference* comes into play at that point, with the team that has a better one ending up ranked higher in the league table. Goal difference is the aggregate of goals scored and goals conceded in one league campaign. It can be positive when the team scores more goals than concedes and negative if it is the other way around.

The first team in the league table at the end of a season is crowned domestic champion. A higher ranking in the league table results in more prize money earned by the teams from the league organization. Other top-ranked teams can earn spots in the European leagues or a spot in the qualifications for those leagues for the next season. The number of teams getting these spots in one particular league is determined by the UEFA country coefficients shown in Table 1.1. The last X number of teams will be relegated to a lower league in the next season, and the best teams from the lower league will replace them and get a chance in the higher competition.

1.3 Objectives and Contributions

Despite the remarkable advancements in the field of machine learning for predicting the outcomes of football matches, relatively less attention has been directed towards predicting the statistical aspects of the sport. This discrepancy becomes particularly evident when considering the problem from a regression perspective, such as predicting the number of corners in a match. To the best of my knowledge, this work represents a first effort in systematically investigating machine learning methods specifically tailored for predicting *diverse* football statistics as a *regression problem*.

Scientific Contributions
This thesis produces the following scientific contributions:

1. A thorough examination of past works related to football outcome prediction and "over/under" prediction of certain statistics will be done in Chapter Literature Review (2). First, in Section 2.1, I summarize the works that are concerned with the "over/under" of some statistics (e.g., goals, corners). Then in Section 2.2 I go through the works that focus predicting the outcomes. The last Section 2.2.1 reviews a paper that examines the prediction of the match outcome by first estimating the match statistics.

2. A comprehensive description of all the models using the standard machine learning approach as well as the ones doing matrix factorization that will be used for the predictions will be done in Chapter Methodology (3).

3. Detailed description of the data use in this thesis will be done in the Chapter Data (4). All data sources will be introduced as well as the initial data that was collected from them. Then, all the data preprocessing and feature engineering (see Section 4.3) will be described for both different approaches that will be used for prediction.

4. Explanation of the experiment process from the training and validation procedures, to the final prediction will be done in Chapter Experiment (5). Also, three different metrics will be introduced for later examination of the results.

5. Presentation of the final results is going to be done in Section 5.4. The final predictions will be evaluated on three different metrics to get more insight on their strengths and weaknesses. The focus will then be on the best and worst performing leagues and the best and worst performing models. Also some patterns describing which models are better or worse at predicting certain statistics will be revealed.

The thesis is organized as follows: the next chapter (Chapter 2) contains a literature review that will explore different related works in the field of football prediction in general. Machine learning techniques used in this work are going to be described and discussed in Chapter 3. The gathering of open data from four top football leagues and its preparation will be reviewed in Chapter 4.

The experiment will start with training and optimizing the models on the training datasets; each league will be done separately and the whole process is explained in Chapter 5. The evaluation of the statistical prediction (see Section 5.4) will be done on the latest x number of football matches in the datasets. Various machine learning and matrix completion methods will be compared and their results presented in textual and graphical form. Finally, differences between predictions in the four leagues and various statistics will be reviewed. Also, possible improvements and future work will be discussed.

Chapter 2
Literature Review

In the following literature review, I'll discuss the current state of the art for football match prediction. As was previously noted in the Chapter 1, the majority of the efforts that have been made in the field of football predictions have mostly been focused on the outcomes of matches (i.e., Win/Home, Lose/Away, Draw (HAD)). Although I have made a comprehensive search, I have found just a few works on the matter of predicting football statistics. Note that such works do not focus on predicting the exact number (regression problem) of corners, shots, etc. They were mainly concerned with predicting the over/under of these statistics. The over/under prediction (O/U) means that one is trying to predict whether the number of, for example, corners will be bigger or smaller than a certain threshold (a classification problem). As the number of shots, corners, etc. is an integer, the thresholds in the betting markets are written like O/U 2.5 or O/U 3.5. These works will be discussed in Section 2.1

Although my focus is on predicting the statistical outcomes of matches rather than the results themselves, it is worth noting that there is some degree of inter-connectivity between these two types of prediction. For instance, certain statistical patterns and trends may provide clues as to the eventual outcome of a given match. However, it's important to recognize that my primary goal is to gain insights into the statistical performance of teams and players, and to use this information to inform my predictions and analysis. By focusing on statistical analysis, I hope to provide a more nuanced and comprehensive understanding of the factors that influence the outcomes of sporting events. In the Section 2.2 I will discuss works that interchange both prediction strands.

2.1 Over/Under prediction

Here I will discuss the works that I found concerned with the prediction of some football match statistics in the form of a prediction of whether the number of certain statistics will exceed or not exceed a given threshold.

In [5], researchers used models based on a broad family of distributions called the "discrete compound Poisson distribution". The researchers applied these models to predicting O/U corners, HAD, and O/U shots. One characteristic of this work is that they keep the bookmakers' pre-match odds in the dataset. Both the O/U odds concerning the statistics that they are going to predict and the HAD odds are being used. They use margin removal methods to extract probabilities implied by the bookmakers in the odds. The odds are typically in the form of a number greater than one, which we'll refer to as the "odds number" or ON. For example, if one bets that Team A will have more than 5.5 corners and the ON for that bet is 2, then you stand to win twice your original wager if your bet is correct. The margin removal methods in this work

are used to extract probability numbers from the ON.

The models are evaluated by creating a betting strategy using the respective model's predictions and finding out which model has the biggest profit over time. The author uses the betting on the under bet on the number of corners as a baseline. They justify this choice by empirical analysis performed earlier in the work where they find that the O/U betting market is biased towards the under bet. That means, if one were to choose between (1) blind bet on over, (2) blind bet on under, and (3) a complete random selection (e.g., by considering (1) and (2)), then the best strategy is to choose the under bet. In the end all their models heavily outperform the blind bet on under in the long run.

Another work that is concerned with the O/U predictions is [6]. The aim of this paper is to predict whether there will be more or less than 2.5 goals in a match. The data are from the same source that I am using in this work [7]. They compute an indicator called *Generalised Attacking Performance* (GAP) of each team during a season. GAP is an indicator that condenses the attack statistics of a team, such as shots, shots on goal, or corners. Because the data is time-dependent, the GAP ratings are updated for each team after each match. As in many other works concerned with football prediction of any kind, home and away instances of each statistic are recorded separately, meaning that a team is considered a different team when playing at home or away.

Simple logistic regression approach is taken to estimate the probabilities for O/U 2.5 goals scored in match. In the experiment itself the author tries to find out which recorded statistics or some of their combinations should be used for calculation of GAP. For that, two different betting strategies are used. The author defines a "Level Stakes" strategy, which puts a bet on some O/U odd if the predicted probability is higher than the implied probability from the bookmakers' odds. And "Kelly Strategy" using Kelly Criterion Kelly [8], both of which take into account the probability and the amount of return based on the odds on that particular bet. It turns out that the most relevant GAP can be computed from the combination of previous matches' number of shots and corners. At the end the author shows that the combination of his model and betting strategy is much more profitable than placing random bets on O/U 2.5 goals.

The following work [9] is focused on predicting whether both teams will score in a match (BTTS) and whether the amount of goals will be O/U certain threshold. The thesis was focused on getting a hold of more detailed data. As an example of that, apart from standard statistics from previous matches, ratings from FIFA games were retrieved and used in the dataset. They used models that estimate a probability distribution over all possible scores of a match by using the Poisson distribution based models and also classification models that predict O/U certain threshold or whether both teams score or not. All models were also based on neural network architecture.

The evaluation was done with the final versions of models, that means with the models' best configurations of hyperparameters. They simulated an environment where the model would place bets on real matches and were evaluated by the summed profitability of their respective bets. The result was that in the BTTS question the distribution models and classification models performed on the similiar level, but on the test data the classification models were better. In the case of O/U 2.5 goals the classification outperformed the models based on Poisson distributions.

Different from previous approaches, this work has more detailed data for making predictions and they make a comprehensive comparison of predicting O/U by classification models and models based on Poisson distribution.

2.2 Predicting the Outcome

The following works are about predicting the outcome of a match. From now on, I will call those methods HAD, because most of the time it is a classification problem in which the algorithm tries to predict whether the winner will be the home team (H), the away team (A), or none of them, therefore a draw (D).

In the paper [10] the authors collect public data from 13 seasons of the Dutch Eredivisie. This work has a comprehensive feature engineering description. After feature engineering, they apply dimensionality reduction techniques to their large dataset. They reached an accuracy of 54.7% with MLP (with three or seven principle components from PCA) and Naive Bayes (with three principle components from PCA) on the public data. They also tried predicting only with the bookmakers' odds, which resulted in 55.3%. In the end, they combined the bookmakers' odds and public data datasets, which resulted in 56.05% accuracy.

One recent paper in the field of football prediction is [11]. The dataset consists of five English Premier League seasons (13/14–18/19). Besides basic statistical data such as corners, shots, shots on target, fouls committed, and goals scored, they also use the referee feature, which holds information about the referee of the match. The authors also use teams' and players' statistics from FIFA games, which contain information about attacking, defending, and other abilities rated on a scale from 0 to 100 [12] and they are constant for a team for each season used. The last thing that was used were the betting odds for each match. Various algorithms were used for predictions, and the best performing was Support Vector Machines (SVM) with an accuracy of 61.32%. In the second phase of prediction, they decided to apply feature selection and test 2048 variable combinations. It turned out that almost half of the "most important" variables were the betting odds' numbers. And with them, Random Forests, XGBoost and SVM managed to score accuracy of 63.95% and even one model with specific combination of features scored even 65.26% accuracy. It is pretty clear that the use of the betting odds can help significantly, but it is a reasonable assumption that it could be problematic in the understanding of the underlying phenomena, since the betting odds' features frequently become the most relevant features and the model becomes an ensemble that is hard to interpret.

In the article [13] the researchers used data from 11 seasons (2005–16) of the English Premier League, as these seasons had all the needed statistics recorded, unlike the seasons before them. As for the features, they incorporated numerous statistics such as goals, shots, and shots on target corners and derived more features from them. They created another time-dependent feature like H_{form}, which represents the form of a team when playing at home by summing the points they got from their last five home matches and dividing it by 15 as the maximum that can be obtained in those matches. In a similar way, a feature named *HTGD* was created by iteratively summing the goals that a given team scored and the goals that that team conceded. All of these features were split into home and away instances. Also, data from `fifaindex.com` about teams' attack, midfield, defense, and overall strength were incorporated into the dataset [12]. They used Gaussian Naive Bayes, Random Forest, Support Vector Machines, and XGBoost machine learning algorithms, with XGBoost performing the bets. To evaluate their models and compare them with the bookmaker's prediction metric called *Ranked probability score* (RPS) metric was used [14]. The XGBoost model scored 0.2156 compared to a score of 0.2012 computed from the odds of a betting organization called *Bet365*. Since a lower RPS means better accuracy, they did not outperform the bookmakers' predictions but showed very promising results.

In the work [15] the authors propose a Bayesian approach instead of a standard statistical or machine learning approach for predicting the outcome of football matches. This paper achieved 75.09% accuracy over three seasons of the Premier League. Here, it is clear that when one knows the number of shots, corners, fouls and other statistics, one has a big advantage in predicting the outcome. But it is not applicable for predicting future matches in real life because none of these are known. This high accuracy was also achieved by running 10-fold cross-validation on each season separately, with each fold set divided into 90% training and 10% testing. The result reported for each season is the average of accuracy of the 10 testing sets. This work achieved very high predictive accuracy but would not be applicable in reality, because of its use of data that can only be known after the match and also because of not treating the data as time dependable.

In [16] develops an efficient framework based on deep neural networks (DNNs) and artificial neural networks (ANNs). This work aims to predict matches in the 2018 FIFA World Cup. Results from international football matches from the years 1872 to 2018, extended with "FIFA

soccer rankings," which contain the rankings of national teams, are used as training and validation. The evaluation was done on the 2018 World Cup set of matches. They used *Long-Short Term Memory neural network* (LSTM) for making the prediction. Their model had an accuracy of 63.3% on the test dataset. The presented results were done only on the group stages, because the subsequent play-off matches were highly unpredictable.

An unorthodox way of predicting football match outcomes was introduced in [17]. A set of predictive models for predicting the English Premier League for a 3 month period was developed. The model predicts the outcomes based on tweets related to the teams and matches based on hashtags that are used on Twitter to specify what a given tweet refers to. They used three datasets: (1) the Twitter dataset, (2) the historical statistics data set, and (3) a combined dataset of the two. The Twitter dataset contained roughly 2 million tweets from fans expressing their thoughts on their respective teams. The tweets were obtained via Twitter's open streaming API. For each team, tweets about 8–10 matches were obtained. Tweets with more than one team's hashtag were discarded. The number of tweets related to each team differed a lot among teams, with Manchester United and Liverpool dominating. To process the data, TwitterNLP and Part-of-Speech Tagger, created by the ARK social media research group at Carnegie Mellon, were employed. Only words with the tags Adjective, Verb, Noun, Adverb, Interjection, Emoticons or Possessive were included. The words in the dataset were then stemmed using the Porter stemmer. Subsequently, the dataset was divided into two versions, one with unigrams and another with bigrams. The historical dataset consisted of various time-dependent features and some features that were stable, e.g., the market value of a team. They used various machine learning models for the prediction with *Random Forest* [18] performing the best on the Twitter dataset (65.6% accuracy) and combined dataset (69.6% accuracy) and *Naive Bayes* on the historical dataset (58.9% accuracy). The results of the Twitter data also demonstrated that bigrams outperformed unigrams, suggesting that the information conveyed in tweets is more intricate than what can be captured by simple unigrams. This approach using tweets shows results that are comparable with standard statistical and machine learning techniques. The results also imply that this technique can be combined with standard machine learning techniques to improve results.

2.2.1 Predicting outcome based on predicting statistics

The use of observed and predicted match statistics as inputs to forecast the results of football matches is described in the paper [19], which is possibly the most related to mine thesis. It is shown that if match statistics could be known before each game, highly accurate forecasts of a match's outcome might be produced. The author shows that the *generalised attacking performance* (GAP) [6] is good for predicting the match statistics, which can then be used for predicting the outcome. The betting odds are used in this paper as potential inputs to models as well as a tool for demonstrating how much profit can be made.

Data from [7] are used here. For each match, statistics are recorded, including the number of shots, shots on target, corners, fouls, and yellow cards, as well as odds from multiple bookmakers concerning, for example, the match outcome, O/U 2.5 goals. All of the 22 available leagues are used, and from them, 49884 matches are marked as usable, meaning that they have a relatively large dataset in this field of prediction. Unusable matches are also the first six and last six matches of each season, because in the first case, there is not enough information about the teams' quality in the season, and in the second case, the results at the end of the season are highly unpredictable because of different motivations among teams (relegation, promotion, etc.).

After that, the work focuses on finding the best combination of features to predict the outcome of a match. These features are the predicted statistics about the upcoming game, features containing information about teams' past performance, and before-match bookmakers' odds, so all of those can be available when making the predictions in real life. The variable selection is done using *Akaike's Information Criterion* (AIC). The predicted features are predicted with the use of GAP 2.1 and are evaluated by *mean absolute error* (MAE). Through this, the best

combination of parameters for calculating the GAP is found. The MAE of the original values of the statistics and their sample means are also shown. Both home and away instances of goals, corners, and shots on target show very similar MAE to the sample mean MAE. Only shots off target seem to be more predictable by the methods used.

Finally, two betting strategies, which will be used for the evaluation of the HAD predictions, are introduced. Both betting strategies show that with the optimal selection of parameters, there is very little difference in profit if you incorporate the bookmakers' odds into the dataset. It is also shown how the strategies would profit in different leagues, and it is clear that some leagues are much better for betting based on predicted statistics and features about the team's past performance than others. The key findings in the results reported are:

- The number of shots on target is a robust indicator of the match's outcome. Moreover, considering the observed counts of shots off target and corners alongside the number of shots on target and/or match odds can provide some predictive value.
- Predictions of match statistics are can be informative about the match's outcome if they are accurate enough.
- When predicting match results, the most informative observed statistics do not necessarily match the most informative predicted statistics. For example, the number of shots on target was found to be the most informative observed statistic, whereas the number of shots off target was identified as the most informative predicted statistic. This discrepancy can be attributed to the fact that predicted statistics capture both the significance of the statistic itself with regard to the match outcome and the precision of the prediction.
- The profit of betting strategies has decreased over the last few seasons. The reason behind that might be the incorporation of more information into current betting odds compared to the earlier stages of the data set.

Sharply differentiating from this work, in this thesis I will focus on predicting the statistics with more detail. In Wheatcroft's thesis, the match statistics are predicted from the GAP, which he introduced in his first work [6]. The GAP is computed from different combinations of a few statistics, like shots and corners. Also, more focus is put on using the final predictions of the statistics to predict the outcome. In this thesis, I am going to use weighted averages of all statistics when predicting a single one. Also, some different features will be created from the past matches, and predictions of more different statistics will be done and analyzed.

Chapter 3
Methodology

As can be observed in the literature review (see Chapter 2), there is a lack of studies that concentrate the efforts in the prediction of football statistics. This work aims to bridge this gap and perform an empirical analysis of machine learning methods to predict football statistics. In this chapter I am going to present the models and methods which I am going to be using for predictions. First, I divide them into two general approaches that are used in machine learning and then I talk about their general strengths and weaknesses. Further, I will briefly describe the properties of the methods used in this thesis.

3.1 Machine Learning Approaches

This chapter will outline the models and methods utilized for predicting the statistics of football matches. At first, I outline machine learning and some of its approaches in general and subsequently describe and discuss all the models that I will be using. There are two main categories of techniques in machine learning [18]: supervised and unsupervised learning. In this work, I am not using any unsupervised techniques; from now on, I am only going to focus on the second approach to machine learning called Supervised learning or "learning with a teacher".

Supervised learning is defined by the use of labeled datasets. The labeled input and output data allow the models from the "supervised" family to iteratively make predictions on the data. In each iteration, the model calculates its error and makes adjustments to enhance the prediction in the next iteration. Over time, the model gets better at predicting data from the training dataset.

The supervised models frequently require human intervention in the process of creating or labeling the datasets, so they also require some expertise in the field they are concerned with. Also, these models generally take more time to get trained on the training data and to find the best hyperparameters for the given problem. On the other hand, they make more informed predictions and are more used in some areas like pricing predictions, weather forecasts, or spam detection.

The supervised machine learning problems can be divided into two groups [18], although they are tied together and most of the models can be adjusted and used for both:

Classification : is when machine learning algorithms are used to assign each data point a class based on the values of its features. The classification problem can be divided into a boolean type of classification, where you predict whether something is in a category or not. Or multiclass classification, where the algorithm chooses one class out of more classes. Support Vector Machines, Logistic Regression, Decision Trees and Random Forests are typical examples of models used for this type of problem.

Regression is the second type of supervised learning where the models try to find and understand the relationships between features and their values. After that, the models try to estimate the exact value of the feature that is being predicted based on the values of all the other labeled features. The typical algorithms are Linear and Ridge Regression. Or Polynomial and Lasso Regression.

3.2 Models

In the following section, I will describe the models which are going to be used for my predictions. All of these algorithms are used as a regression predictors since all the statistics which I am going to be predicting are represented as numbers.

For this work, I used only one programming language - Python. I selected it because of Python's popularity for data science and machine learning and its generally growing popularity in recent times. Also in Python, you can make use of so-called *Jupyter notebooks* [20], which can be very useful for data exploration and preparation and are really easy and straightforward to use. At the end of the description of each model, I list the hyperparameters that were tuned in the experimental section.

3.2.1 Poisson Regression

Poisson regression is a machine learning algorithm that can be used for regression problems where the target is a numerical variable that follows a Poisson distribution [21].

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{3.1}$$

It is a type of generalized linear model (GLM) that models the relationship between the predictors and the Poisson distribution's mean. To estimate the model parameters, the Poisson regression algorithm uses maximum likelihood estimation. Given the predictor variables and the presumed Poisson distribution, it seeks to find the parameter values that maximize the likelihood of observing the specified target values.

I selected to use Poisson regression-based models because they showed great results in [5],[22],[23]. I used the implementation of Poisson Regressor from the scikit-learn library for Python. In this library, the algorithm has these hyperparameters that can be adjusted:

Alpha is a parameter that defines the L2 regularization strength of the model. It is a value greater than or equal to zero, where a value of zero means no regularization penalty in the loss function is considered. Setting the parameter to something else than zero can help with overfitting.

Intercept is a boolean value that determines whether to include an intercept term in the model. If set to True, the model will include an intercept term. If set to False, it will not.

Maximum number of iterations for the optimization algorithm to converge. If the optimization algorithm does not converge after this number of iterations, it returns the current best estimate.

Solver holds the information about the algorithm which will be used for optimization.

3.2.2 Ridge Regression

Is a simple model very similar to standard linear regression with the addition of L2 regularization to the loss function. The basics are the same as for linear regression [18] which is the use of the ordinary least squares method and the use of the equation:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1^\top x \tag{3.2}$$

where ŷ represents the predicted response variable value. For this method, it is important to standardize the features before running the learning process, so as to not confuse the loss function's parameters. If one does not do that, problems with different features having different magnitudes just because of their scale can arise. I used a ridge regression-based model because of its universality and simplicity. Again I used the implementation of it called Ridge from the scikit-learn library. Here are the hyperparameters which I optimized, they are very similar to the ones in Poisson Regressor:

Alpha is the regularization strength hyperparameter that controls the amount of L2 regularization penalty. A higher value of alpha results in more regularization, which can help prevent overfitting.

Intercept is a Boolean hyperparameter that controls whether to include an intercept term in the model.

Solver parameter determines the solver to use in the optimization problem. The are seven different options and the 'auto' option chooses the solver based on the data and other parameters.

3.2.3 Random Forests

Random Forests are an ensemble machine learning algorithm. Ensemble learning techniques are composed of a collection of classifiers, such as decision trees, whose outcomes are integrated to determine the most prevalent prediction. Bagging and boosting are among the most widely recognized ensemble methods. The bagging [24] technique involves randomly selecting a subset of data from the training set with replacement, allowing for multiple selections of individual data points. These subsets are then utilized to independently train models. The resulting predictions are aggregated using an average or majority rule, depending on whether the task is a regression or classification problem, respectively. This process results in improved estimation accuracy.

The random forest algorithm is based on a decision tree machine learning algorithm and the previously described bagging method. It consists of numerous decision trees which have a much smaller depth than the usual standalone decision tree and consider only a subset of all features. The random subset of features is generated so that the correlation among decision trees is low. During prediction, the ensemble of decision trees is used to generate a set of predictions, which are then aggregated to produce the final output. This technique allows Random Forest to handle large and complex datasets while avoiding overfitting, and it can be used for both classification and regression problems. I used the RandomForestRegressor from the scikit-learn library, and optimized these three parameters:

Number of estimators specifies the number of decision trees to be used in the random forest. For different problems, different sizes of forests are optimal.

Maximum depth parameter sets the maximum depth of each decision tree. Limiting the depth can prevent overfitting, but it can also result in an underfitting model.

Minimum samples to split specifies the minimum number of samples required to split an internal node of a decision tree. Increasing this value can prevent overfitting, but it can also result in an underfitting model.

3.2.4 XGBoost

XGBoost[25], short for "Extreme Gradient Boosting", is a machine learning algorithm for supervised tasks. It is an ensemble method like the RF algorithm, but unlike the RF which uses the bagging ensemble technique, XGBoost uses boosting. Boosting is a technique that involves iteratively adding weak learners, typically decision trees, to the model to improve its overall predictive power. In each iteration, the algorithm attempts to fit a new decision tree to the residuals, or errors, of the previous iteration. By doing so, the algorithm can improve its predictions by taking into account the errors that were made in previous rounds.

XGBoost uses decision trees as base models, and the algorithm optimizes a specific loss function to improve the model's performance. The decision trees are grown using a depth-first approach, where the algorithm recursively splits the data into smaller subsets based on the most informative features. XGBoost employs several advanced techniques to optimize the performance of its decision trees, such as parallel processing, tree pruning, and regularization, which helps prevent overfitting. I used the xgboost package for Python, here are the parameters which were optimized:

Number of estimators defines the number of trees in the ensemble model.

Maximum depth of each decision tree. A larger value can lead to overfitting, while a smaller value may result in underfitting.

Learning rate parameter determines the step size at which the algorithm will adjust the weights of the features in each round. A smaller learning rate will result in slower learning but may lead to better performance in the long run.

3.2.5 Multilayer-Perceptron Neural Network

A Multilayer Perceptron (MLP) [26] neural network is a type of artificial neural network (ANN) that is widely used for both classification and regression problems. I chose to use this model due to its robustness and ability to perform well on almost any task. MLP consists of multiple layers of artificial neurons, with each neuron receiving input from all the neurons in the previous layer and producing output that is fed to the neurons in the next layer. This type of ANN is called a fully connected network. The MLP is a feedforward neural network, which means that the information flows through the network in one direction, from input to output, without any feedback loops. MLP neural networks are trained using a process called backpropagation, which involves adjusting the weights of the connections between the neurons in each layer based on the difference between the predicted output and the actual output. The weights are updated using an optimization algorithm, which seeks to minimize the error between the predicted output and the actual output.

MLP neural networks are known for their ability to capture complex relationships between variables and are particularly useful for problems that involve nonlinear relationships between the input and output variables. However, they can be sensitive to overfitting, especially when the number of neurons and layers is large, so regularization techniques are often used to prevent overfitting. In this work, a scikit-learn implementation called *MLPRegressor* was used. As for the optimizer I used the *Adam* optimizer instead of *Stochastic Gradient Descent* (SGD). The Adam optimizer updates the learning rate adaptively based on the gradient history, whereas SGD has a fixed learning rate. Here are the hyperparameters which I optimized:

Initial learning rate is a number that determines the step size at each iteration while moving towards a minimum of a loss function.

Maximum number of iterations is an integer value that determines the maximum number of iterations the learning algorithm performs.

Batch size attribute determines the number of samples used in each iteration for adjusting the weights in the network.

Sizes of hidden layers defines the number of neurons in each hidden layer of the MLP. It can be set as a tuple or list of integers, where each integer specifies the number of neurons in that layer.

3.2.6 Matrix factorization

Matrix Factorization [27] is a technique used in machine learning to decompose a large matrix into smaller, simpler matrices that represent the original data in a more compact and meaningful way. The idea behind Matrix Factorization is to identify the underlying patterns and structures in a matrix by breaking it down into multiple factors that can be used to reconstruct the original matrix.

Matrix Factorization has various applications, including recommender systems, image and audio processing, and natural language processing. In recommender systems, Matrix Factorization is used to predict the user's ratings on a set of items, based on their historical ratings and the characteristics of the items. The matrix representing the users' ratings is decomposed into two smaller matrices, one representing the users and the other representing the items, which can then be used to predict the unknown ratings. In this thesis, matrix factorization is used in a non-traditional way. The matrix on which the methods are applied is relatively small compared to the usual cases. If e.g., *Home Team Corners* is the variable of interest, the matrix will look something like this:

Table 3.1 Matrix for factorization example

HT/AT	Team1	Team2	Team3	...	Team20
Team1	nan	6	?	...	7
Team2	3	nan	4	...	?
Team3	2	?	nan	...	5
...	...	...	...	...	...
Team20	1	4	?	...	nan

Where "nan" stands for values that don't need to be computed and are of no interest and "?" is a value that will be computed in the process based on other values.

So for example the number 6 where Team1 and Team2 meets means that Team1 had 6 corners when they played at home against Team2.

Also when I will be dealing with any variable's home instance (e.g. *Home Corners*). The home team will be considered as the *user* and the away team as the *item* from the recommender systems point of view. And it will be the other way around when predicting the away instance of some variable.

Matrix Factorization can be performed using various algorithms, including Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), and Alternating Least Squares (ALS). These algorithms differ in their assumptions about the underlying structure of the matrix, and in their computational complexity and scalability. In the following three sections, I am going to briefly describe the three different algorithms which will be used for prediction in this thesis.

3.2.6.1 Singular Value Decomposition

The first method is SVDpp (Singular Value Decomposition with implicit feedback and user biases) from the Surprise library [28] which is a Python scikit (SciPy Toolkits) for building and analyzing recommender systems. It is a matrix factorization technique for collaborative filtering

in recommender systems, which is an extension of the SVD (Singular value decomposition) algorithm.

In SVDpp, each user and item is represented as a vector in a low-dimensional latent space. The model learns these representations by factorizing the user-item interaction matrix (e.g., user ratings) into two lower-dimensional matrices representing user and item latent factors, and a diagonal matrix representing the strengths of the interactions. The model also includes user biases, which account for differences in how users rate items, and item biases, which account for differences in the overall quality of the items.

To estimate the model parameters, SVDpp uses a stochastic gradient descent algorithm to minimize the regularized squared error between the predicted ratings and the observed ratings. The regularization term helps prevent overfitting to the training data. The model parameters are learned by minimizing a loss function that takes into account the squared error between predicted and actual ratings, the regularization term, and the implicit feedback.

3.2.6.2 BaselineOnly

BaselineOnly is a matrix completion method also from the Surprise library [28]. this method predicts the rating of an item based on a baseline estimate and an item-specific bias. The baseline estimate is the average rating of all the items in the dataset, and the item-specific bias is the difference between the average rating of an item and the overall average rating.

The method works by first computing the baseline estimates and item-specific biases for all the items in the dataset. It then predicts the rating of an item for a user by adding the baseline estimate and the item-specific bias. This method is based on the intuition that a user's rating for an item is influenced by the average rating of all the items in the dataset, as well as the specific characteristics of the item itself.

3.2.6.3 Orthogonal Inductive Matrix Completion

Matrix factorization with bias is a technique widely used in collaborative filtering and recommender systems to predict user-item preferences or ratings. Additionally, the user and items' latent features and bias terms are incorporated to account for inherent biases in the ratings, such as users having a tendency to rate items higher or lower than others. The predictors are in the form of

$$f_{i,j} = \gamma + \alpha_i + \beta_j + u_i^\top v_j,$$

where γ is a general bias, α_i and β_j are, respectively, user and item biases, while u_i and v_j are, respectively, user and item feature vectors.

This model also can be used to predict sports statistics. The bias here, for instance, would represent a tendency for the number of corners of a team to be higher or lower than others. To implement this baseline I used an orthogonal inductive matrix completion algorithm introduced in [29].

Chapter 4
Data

The data is a key element in most professional works and even more so in the field of machine learning. Thus it is very important to make a thorough examination of the acquired data. The task is to collect data from publicly available data sources. The data must contain certain information about the football matches and at the same time go far enough into the past. Thanks to football being arguably the most popular sport in the world finding open data from the top leagues is not the hardest job.

4.1 Data Sources

Publicly available data source [7] was mentioned several times before in the related works (see Chapter 2). This website holds data about matches from more than 20 popular leagues. All data can be easily downloaded in *.csv* format. The four top leagues mentioned in Chapter 1.2 EPL, SLL, GBL, and SA had almost all the data I needed for the experiment on this website. The only problem I encountered was that only the EPL had the column called *referee*, which specifies which referee was assigned to a specific game. The referee can be important when predicting football statistics, especially the corners, fouls, and yellow and red cards. Each referee has different measures and some may incline towards judging something as a foul when some others would not.

The data about who was the referee in matches in SLL, GBL, and SA had to be found somewhere else. Such information can be collected in [30]. This website also holds a lot of interesting statistics about teams' possession of the ball, passing, and numerous other team or player-related statistics. Unfortunately, I did not find a use for all of those, so I only used it to get the referees for the matches I had from the previous data source. The only problem I encountered while merging the data files was that the names of teams differed across data sources (e.g. Düsseldorf-Fortuna Dusseldorf, Atlético Madrid-Ath Madrid). I had to unify the names of teams in these three leagues and then merge the datasets. After that, the joining went smoothly, because the combination of columns *Date*, *HomeTeam* and *AwayTeam* is a unique combination. All data manipulations were done using Python package *pandas* [31].

4.2 Datasets

In the beginning, there are 40 data files, which means ten seasons for each of the four leagues. EPL, SLL, and SA have 20 teams in one season and each team plays every other team two times, resulting in 380 matches in each season. The GBL has only 18 teams in one season so that results in 306 matches. I used matches from seasons 2012/13 - 2021/22 even though the most

recent matches are recorded. For a comprehensive and stable prediction the seasons have to be fully completed, so I only considered the closed seasons. In Table 4.1, all the variables that are recorded in the data can be seen. However, the statistics are post-match statistics, meaning that the values in all of the columns except for *Date*, *HomeTeam*, *AwayTeam* and *Referee* can only be known after the match has been played out. This means that the datasets in this format can not be used for prediction as they would not be usable in real-life practice. Some new features that can be known before the match have to be created from them.

Table 4.1 Initial datasets' variables

Initial dataset's variables	
Variable name	Description
Date	The date of the match
HomeTeam	Name of the home team
FTHG	Full time goals scored by the home team
AwayTeam	Name of the away team
FTAG	Full time goals scored by the away team
FTR	Full time result (H/A/D)
Referee	Name of the referee judging the particular game
H_{shots}	Number of shots of the home team
A_{shots}	Number of shots of the away team
H_{target}	Number of shots on target of the home team
A_{target}	Number of shots on target of the away team
H_{corners}	Number of corners played by the home team
A_{corners}	Number of corners played by the away team
H_{fouls}	Number of fouls committed by the home team
A_{fouls}	Number of fouls committed by the away team
H_{yellow}	Number of yellow cards given to the home team
A_{yellow}	Number of yellow cards given to the away team
H_{red}	Number of red cards given to the home team
A_{red}	Number of red cards given to the away team

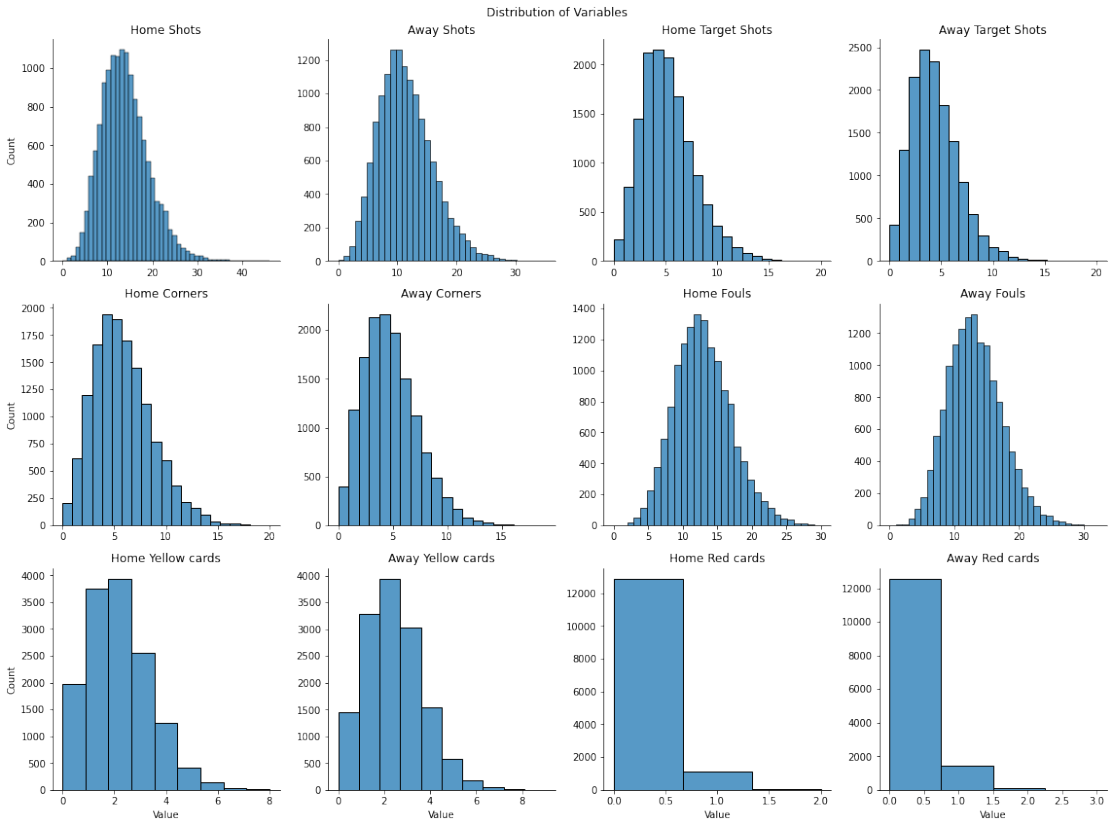
Now features that can be used for training and following prediction have to be engineered from these variables. The last twelve columns are the match statistics that will be the subject of interest in the upcoming prediction. Each of those variables will be predicted separately by all the models introduced.

In Figure 4.1(a), distributions of these match statistics are shown. The distributions regarding *Shots*, *Shots on Target* and *Corners* follow a Poisson distribution. The fouls' distributions look more symmetric. The *Yellow Cards* and *Red Cards* variables have a much smaller number of values, with mostly being equal to zero.

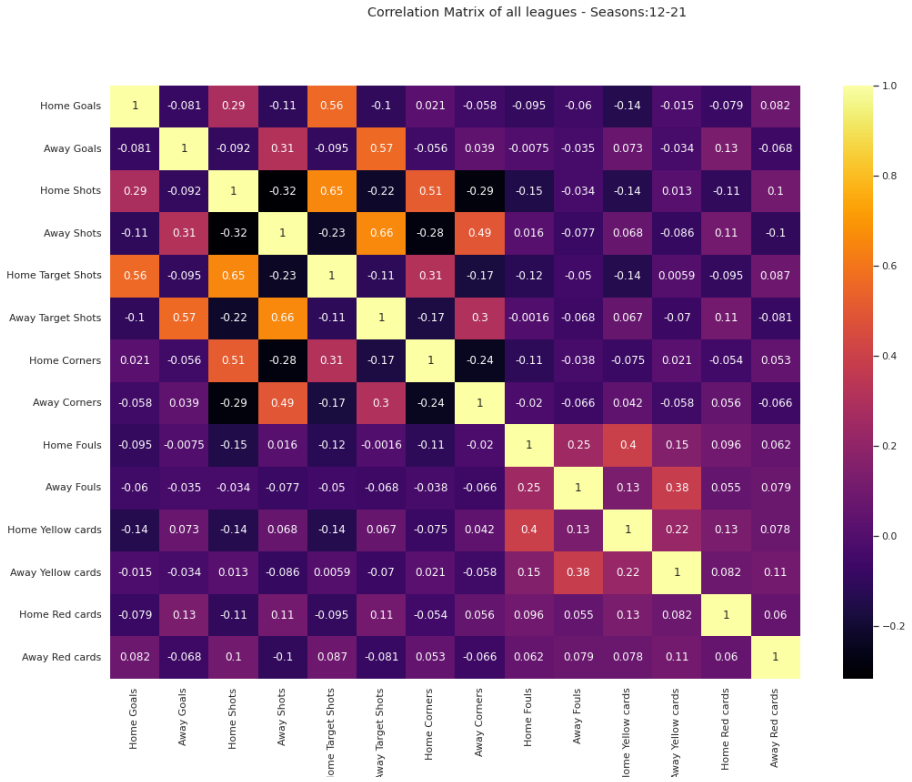
In the second Figure 4.1(b) correlation matrix of these variables combined from all four leagues can be seen. Some key correlation explanations:

- Positive correlation between two variables indicates that when the value of one of those variables falls or rises the second behaves in a similar way.

- Negative correlation between two variables exists if the value of the first variable rises when the value of the second one falls, or the other way around.



(a) Distributions of all the known match statistics



(b) Correlation matrix of all four leagues combined

Figure 4.1 Initial datasets' feature study

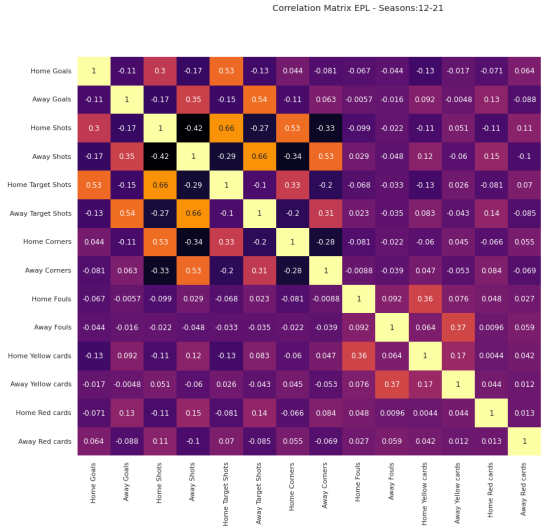


Figure 4.2 EPL correlation matrix

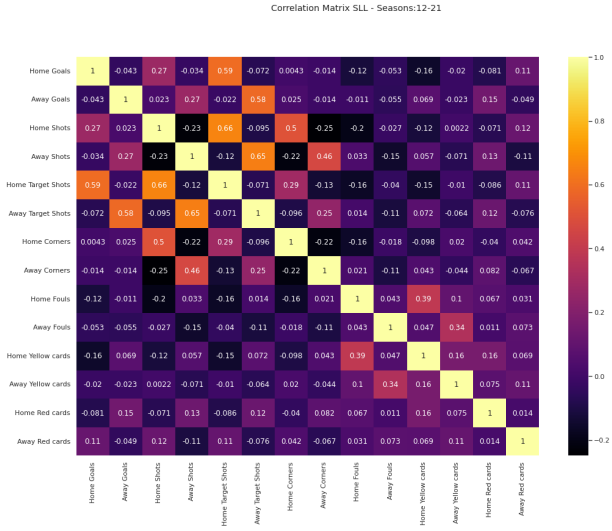


Figure 4.3 SLL correlation matrix

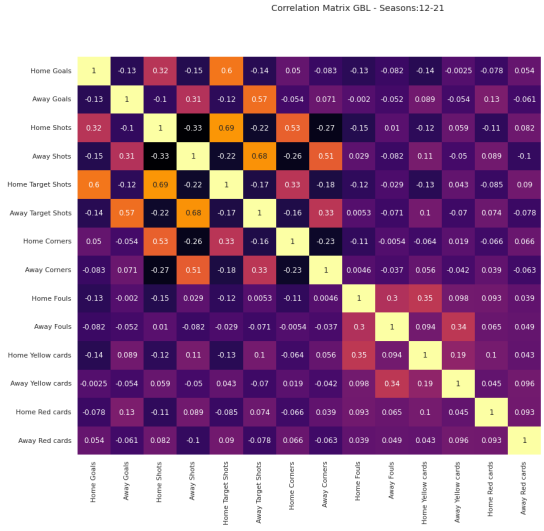


Figure 4.4 GBL correlation matrix

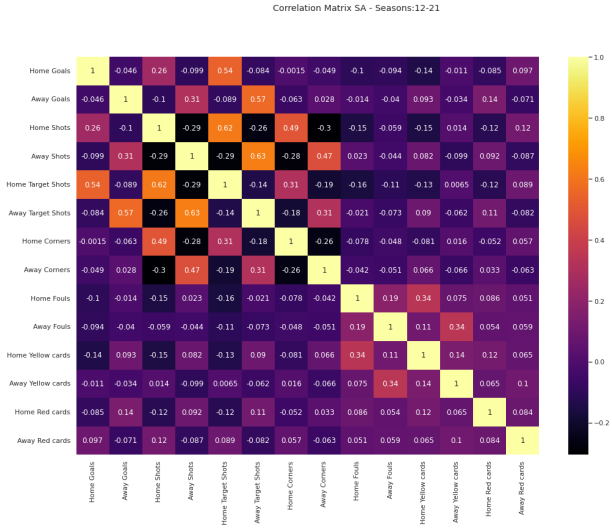


Figure 4.5 SA correlation matrix

Here figures 4.2, 4.3, 4.4 and 4.5 present a correlation matrices for every single one of the four leagues. It is interesting to see that some leagues' have some variables more positively or negatively correlated than others. The biggest difference is between EPL and SLL with EPL having more positive correlations in cases where SLL has even negative correlations. Shots, shots on target, and goals of either home, or away instance, but all related to the same one, are highly positively correlated. Although there is a number of games when one team scores a goal early then defends and still wins it, it is clear that predominantly the teams who score more, also shoot more and pressure more which is also implied by the positive correlation of corners to shots and shots on target. A strong positive correlation can also be seen between fouls, yellow cards, and red cards. This also is because these events are heavily linked to each other with cards being given when one team fouls more frequently or some more serious fouls happen. Negative correlations are the strongest between shots, corners, and shots on target in different instances.

Meaning that usually, one team is the more attacking one and the other one the defending one. A slight negative correlation across all leagues can also be observed between fouls and goals, shots, shots on target, and corners of the same team, meaning that when one team pressures more, they are less likely to commit fouls.

In tables 4.2, 4.3, 4.4 and 4.5, mean, median, standard deviation, minimum, maximum, and values at 25% and 75% of each sorted column can be seen for each league's data.

It is worth noting that the GBL has the most shots and goals but does not have the most shots on target. Another fact that deserves attention is the differences in home and away instances of some statistics. For example, the biggest difference in goals, shots, and shots on target between home teams and away teams can be seen in SLL. The biggest difference across leagues can be observed in fouls and because of fouls' strong positive correlation to yellow and red cards, a relatively major difference between leagues is also observable. EPL has very small numbers of all those statistics compared to all other leagues, but SLL and SA still have higher numbers than GBL.

Table 4.2 EPL features statistics

Feature	mean	std	median	min	25%	75%	max
FTHG	1.52	1.31	1.0	0.0	1.0	2.0	9.0
FTAG	1.22	1.19	1.0	0.0	0.0	2.0	9.0
H_{shots}	13.98	5.64	13.0	0.0	10.0	17.0	43.0
A_{shots}	11.36	4.88	11.0	0.0	8.0	14.0	31.0
H_{target}	5.03	2.93	5.0	0.0	3.0	7.0	20.0
A_{target}	4.15	2.52	4.0	0.0	2.0	6.0	20.0
H_{corners}	5.85	3.1	5.0	0.0	4.0	8.0	19.0
A_{corners}	4.73	2.7	4.0	0.0	3.0	6.0	17.0
H_{fouls}	10.52	3.39	10.0	0.0	8.0	13.0	24.0
A_{fouls}	10.88	3.56	11.0	1.0	8.0	13.0	26.0
H_{yellow}	1.53	1.21	1.0	0.0	1.0	2.0	7.0
A_{yellow}	1.72	1.26	2.0	0.0	1.0	3.0	9.0
H_{red}	0.06	0.24	0.0	0.0	0.0	0.0	2.0
A_{red}	0.08	0.27	0.0	0.0	0.0	0.0	2.0

Table 4.3 SLL features statistics

Feature	mean	std	median	min	25%	75%	max
FTHG	1.53	1.32	1.0	0.0	1.0	2.0	10.0
FTAG	1.14	1.13	1.0	0.0	0.0	2.0	8.0
H_{shots}	13.28	4.92	13.0	2.0	10.0	16.0	36.0
A_{shots}	10.58	4.35	10.0	0.0	7.75	13.0	35.0
H_{target}	4.71	2.56	4.0	0.0	3.0	6.0	18.0
A_{target}	3.71	2.16	3.0	0.0	2.0	5.0	15.0
H_{corners}	5.55	2.89	5.0	0.0	3.0	7.0	20.0
A_{corners}	4.33	2.54	4.0	0.0	2.0	6.0	17.0
H_{fouls}	13.79	4.19	14.0	1.0	11.0	16.0	30.0
A_{fouls}	13.63	4.16	13.0	0.0	11.0	16.0	30.0
H_{yellow}	2.42	1.5	2.0	0.0	1.0	3.0	8.0
A_{yellow}	2.64	1.49	3.0	0.0	2.0	4.0	9.0
H_{red}	0.12	0.34	0.0	0.0	0.0	0.0	2.0
A_{red}	0.13	0.36	0.0	0.0	0.0	0.0	3.0

Table 4.4 GBL features statistics

Feature	mean	std	median	min	25%	75%	max
FTHG	1.66	1.37	1.0	0.0	1.0	2.0	9.0
FTAG	1.32	1.23	1.0	0.0	0.0	2.0	7.0
H_{shots}	14.07	5.2	14.0	1.0	10.0	17.0	36.0
A_{shots}	11.72	4.74	11.0	0.0	8.0	15.0	32.0
H_{target}	5.14	2.7	5.0	0.0	3.0	7.0	16.0
A_{target}	4.29	2.44	4.0	0.0	3.0	6.0	20.0
H_{corners}	5.26	2.9	5.0	0.0	3.0	7.0	19.0
A_{corners}	4.4	2.55	4.0	0.0	3.0	6.0	15.0
H_{fouls}	13.25	4.33	13.0	2.0	10.0	16.0	29.0
A_{fouls}	13.89	4.45	14.0	1.0	11.0	17.0	30.0
H_{yellow}	1.68	1.25	2.0	0.0	1.0	2.0	8.0
A_{yellow}	1.95	1.26	2.0	0.0	1.0	3.0	7.0
H_{red}	0.07	0.26	0.0	0.0	0.0	0.0	2.0
A_{red}	0.09	0.29	0.0	0.0	0.0	0.0	3.0

Table 4.5 SA features statistics

Feature	mean	std	median	min	25%	75%	max
FTHG	1.54	1.27	1.0	0.0	1.0	2.0	7.0
FTAG	1.25	1.17	1.0	0.0	0.0	2.0	7.0
H_{shots}	13.47	5.44	13.0	1.0	10.0	17.0	46.0
A_{shots}	11.19	4.69	11.0	0.0	8.0	14.0	31.0
H_{target}	5.16	2.73	5.0	0.0	3.0	7.0	18.0
A_{target}	4.26	2.47	4.0	0.0	2.0	6.0	16.0
H_{corners}	5.69	3.07	5.0	0.0	3.0	7.0	20.0
A_{corners}	4.65	2.71	4.0	0.0	3.0	6.0	19.0
H_{fouls}	13.98	4.22	14.0	3.0	11.0	17.0	29.0
A_{fouls}	14.22	4.33	14.0	1.0	11.0	17.0	32.0
H_{yellow}	2.19	1.31	2.0	0.0	1.0	3.0	7.0
A_{yellow}	2.44	1.36	2.0	0.0	1.0	3.0	8.0
H_{red}	0.11	0.34	0.0	0.0	0.0	0.0	2.0
A_{red}	0.15	0.39	0.0	0.0	0.0	0.0	3.0

4.3 Feature Engineering

In this section, I will give an explanation of all features that I computed from the collected data. The final datasets are going to be used for training all the machine learning algorithms except the three matrix factorization methods. All features corresponding to a match are generated from matches that have been played before that match, because only this way the process can be applied in reality for predicting match statistics in a future match. First, it is important to take a look at the HAD statistics in the four leagues being analyzed. The following table shows the percentage representation of home team victories, away team victories, and draws in the matches' outcomes.

Table 4.6 HAD outcomes statistics

League	Home	Away	Draw
EPL	44.58	31.66	23.76
SLL	46.04	28.35	25.61
GBL	45.0	30.59	24.41
SA	43.72	31.24	25.04

Table 4.6 shows that the leagues differ in the numbers of wins and draws, with EPL being the most decisive one, meaning that it has the least amount of draws as opposed to SLL which has the biggest amount of draws. But most importantly the relatively large difference between the percentages of home and away victories in all four leagues is clear. The victory of the home occurs in 44.85% matches played as opposed to the victory of the away team which happens in 30.46% cases. This is the reason why every team will be considered a different team when playing at home and when playing away from home. Most of the generated features will be composed of X number of last matches. For example, when the feature named H_{form}, which represents the number of points a home team has collected in the past five games, is going to be created, it will take the points from the last five games of the home team when playing at home. The last away matches of the home will be ignored in this and many other cases.

All features computed in one season are derived from only that one season in one particular league, not considering anything from other seasons in the same league. Because of all features being generated from previously played matches, the first matches (matches from the first round of the league, after that round, each team has one game played) do not hold much information. For example, the matches from the first round in a season cannot have any of those generated features and the matches from the second round will have all those features created from only the matches from only the first round. This is why the first 80 matches in EPL, SLL, and SA and the first 72 in GBL are discarded. Those numbers correspond to the first 8 rounds in each of the leagues. I adopted this number as a compromise between not losing too many matches and getting rid of the ones with the least information.

The final considered features can be seen in Table 4.8 and they were derived from the original dataset (Table 4.1). The features H_{form} and A_{form} reflect the number of points collected by the home team in the last five games played at home and the number of points collected by the away team in the last five games played away respectively. As mentioned in Section 1.2.2 team gets three points for a victory, one point for a draw, and zero points for a loss. Thereby the maximum number of this feature is 15 for five consecutive wins and the minimum is 0 for five consecutive losses.

The features H_{points} and A_{points} hold the number of points collected by the home team in all previous home and away matches and by the away team respectively. While the H/A_{form} features reflect the recent performance which is very important for predicting the following match, these two features recording all points collected compensate for teams that can have a very good season, but not the best results in the last matches. These features are also independent of whether the points were collected at home or away, so they describe the teams' overall strength in the actual season.

The features $HT\text{total}_{\text{GD}}$ and $AT\text{total}_{\text{GD}}$ record the overall goal difference of the home team and away team respectively. Again this feature does not separate matches to home and away, so it reflects the team's performance in general. Goal difference is important because some very strong teams can have a few close losses or draws and some dominating wins, which cannot be captured by the points, but can be the goal difference. $HT\text{home}_{\text{GD}}$ and $AT\text{home}_{\text{GD}}$ capture the goal difference of the home and away team respectively but only considering the home matches of both teams. The last goal difference features $AT\text{home}_{\text{GD}}$ and $AT\text{away}_{\text{GD}}$ work the same as the last two but they consider only the away games of the teams.

The last 12 features are time-weighted averages of all the basic statistics recorded and de-

scribed in the initial dataset. The time-weighted average is similar to the normal average of previous values, with the addition of weights. The more recent the record is the higher the value of weight. The time-weighted averages were calculated by the following equation:

$$twa\_value = \frac{\sum_{i=0}^{n-1} value_i \cdot weight_i}{\sum_{i=0}^{n-1} weight_i}, \qquad \text{where } weight_i = (1-\alpha)^i \tag{4.1}$$

The list of weights that are used to gradually multiply the values is computed by this equation.

$$\text{weights} = \left[(1-\alpha)^t\right]_{t=0}^{n-1} \tag{4.2}$$

where $0 \leq t \leq n-1$ and α is a constant value, in this case, set to 0.2.

After having all these features computed I need to concatenate all the ten seasons in each league. The predictions are going to be done on every league separately, so that will produce four different datasets. As mentioned before the number of discarded matches was set to 80 for EPL, SLL, and SA and 72 for GBL. Before concatenating all the seasons in a single league, these matches will be deleted from the datasets.

Table 4.7 Number of matches in the final dataset (In SA, there was one match missing in one season, that is why it has 1 match less than it should have.)

	Deleted Matches	**Number of Matches**
EPL	800	3000
SLL	800	3000
GBL	720	2340
SA	800	2999

The values in the *Ref* (from Table 4.8) column are categorical, so they cannot be used for training the machine learning models. I used the One-Hot Encoding (OHE) technique to deal with this. The OHE principle is used for the names of the referees. For every name, a new column is created, 1 is written to the corresponding rows where the particular name was originally written, and 0 to every other column on that same row.

The columns *Date*, *HomeTeam*, *AwayTeam* (from Table 4.1) are simply discarded in the final dataset for non-matrix factorization machine learning techniques (see Table 4.8) because they were only used for creating other features. The teams' statistics are captured in the newly created columns, and if the *HomeTeam*, *AwayTeam* would have been kept, they would need to be preprocessed using OHE the same as the *Ref* column, and it would result in more than 50 new columns with information whose positive impact on the prediction is unsure.

In the end, all the numerical variables have to be standardized by scaling them such that the data is normally distributed. For this, I used *StandardScaler* from scikit-learn [32]. The StandardScaler is a data preprocessing technique used in machine learning to scale features to have a zero mean and unit variance. It works by first calculating the mean and standard deviation of each feature in the dataset and then transforming each feature to subtract its mean and divide by its standard deviation. This process ensures that all features are on the same scale and have a similar range of values.

The final dataset's variables are described in Table 4.8 which are all created from the initial data described in Table 4.1.

Table 4.8 One season dataset for prediction after preprocessing

One season dataset for prediction	
Variable name	Description
Ref (OHE)	Name of the referee judging the particular game, mlutiple OHE columns
H_{form}	Number of points earned by the HomeTeam in the last five games
A_{form}	Number of points earned by the AwayTeam in the last five games
H_{points}	Total points earned the season by the HomeTeam
A_{points}	Total points earned the season by the AwayTeam
$HT\text{total}_{\text{GD}}$	Goal difference from all previous matches of the HomeTeam
$HT\text{home}_{\text{GD}}$	Goal difference from all previous home matches of the HomeTeam
$HT\text{away}_{\text{GD}}$	Goal difference from all previous away matches of the HomeTeam
$AT\text{total}_{\text{GD}}$	Goal difference from all previous matches of the AwayTeam
$AT\text{home}_{\text{GD}}$	Goal difference from all previous home matches of the AwayTeam
$AT\text{away}_{\text{GD}}$	Goal difference from all previous away matches of the AwayTeam
H_{shots}TWA	Time weighted average of shots from previous home matches of the HomeTeam
A_{shots}TWA	Time weighted average of shots from previous away matches of the AwayTeam
H_{target}TWA	Time weighted average of shots on target from previous home matches of the HomeTeam
A_{target}TWA	Time weighted average of shots on target from previous away matches of the AwayTeam
H_{corners}TWA	Time weighted average of corners from previous home matches of the HomeTeam
A_{corners}TWA	Time weighted average of corners from previous away matches of the AwayTeam
H_{fouls}TWA	Time weighted average of fouls from previous home matches by the HomeTeam
A_{fouls}TWA	Time weighted average of fouls from previous away matches by the AwayTeam
H_{yellow}TWA	Time weighted average of yellow cards from previous home matches of the HomeTeam
A_{yellow}TWA	Time weighted average of yellow cards from previous away matches of the AwayTeam
H_{red}TWA	Time weighted average of red cards from previous home matches of the HomeTeam
A_{red}TWA	Time weighted average of red cards from previous away matches of the AwayTeam

4.4 Data for Matrix Factorization

Matrix factorization methods compute the predictions using only the season in which they are predicting. All models are going to be evaluated only based on the last X number of matches and all of them will be from the last season. Thereby, only data from the last season (21/22) from each league will be needed. I already showed how a matrix for predicting H_{shots} will look like 3.1. Basically, no complex data preprocessing is needed. The only thing that has to be done is to create the matrix for every single one of the twelve statistics that are going to be predicted. When these matrices with every row representing one home team, every column representing one away team, and values at coordinates corresponding to the meeting of those two teams are created, the last (from a time's perspective) X number of records need to be deleted.

Chapter 5

Experiments

In this chapter, I will describe how the experiment was conducted. The objectives were to find out the differences between predicting various statistics, some having more than forty different values (e.g., shots) and some having only two values (red cards), and also the differences in predicting these statistics in the four leagues, with each one of them being specific in distinctive ways.

5.1 Metrics

In this section, I will describe all the metrics that are going to be used for training, finding the best hyperparameter configuration, and evaluating the results.

5.1.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a common metric used to measure the difference between two continuous variables, typically the predicted and actual values in regression tasks.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.1}$$

where n is the number of samples, y_i is the original value of the i-th sample, and $\hat{y}_i$ is the predicted value for the i-th sample. The absolute value $|\cdot|$ is used to ensure that the error is always positive.

It offers a straightforward and comprehensible way of evaluating model performance because it is expressed on the same scale as the data. That is the reason why I use it in the final prediction evaluation, because when the MAE is, for example, 3.2 predicting, e.g., H_{shots}, the model averages to miss the exact prediction by 3.2. The model performs better at making precise predictions when the MAE is lower.

5.1.2 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a frequently used evaluation metric in machine learning. It measures the average distance between the predicted values and the actual values of the target variable, taking into account the square of the difference between the predicted and actual values. RMSE is particularly useful when large errors are undesirable.

The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.2}$$

where n is the number of observations, y_i is the actual value of the target variable for the i-th observation, and $\hat{y}_i$ is the predicted value of the target variable for the i-th observation.

RMSE penalizes larger errors more than smaller errors due to the squaring operation. It is always non-negative. I opted for using RMSE instead of Mean Squared Error (MSE) because the square root of the final is more comprehensible in the match statistics in the same way as MAE.

5.1.3 R^2 Score

The R^2 score, also known as the coefficient of determination, is a different kind of metric for evaluating regression tasks than the first two. It provides a measure of how well the model fits the data, with higher values indicating a better fit.

The R^2 score is calculated as the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It generally ranges from 0 to 1, with higher values indicating a better fit. An R^2 score of 1 indicates a perfect fit, while an R^2 score of 0 indicates that the model is no better than predicting the mean of the dependent variable. In some cases, the score can be lower than 0, which means that using the mean would explain the variance of the target variable better. The equation for calculating the R^2 score is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.3}$$

where y_i is the actual value of the dependent variable for the ith observation, $\hat{y}_i$ is the predicted value of the dependent variable for the ith observation, $\bar{y}$ is the mean of the dependent variable, and n is the number of observations.

5.2 Validation Procedure

The training process involves feeding the model with a set of labeled data, also known as training data, which is used to adjust the model's inner parameters (e.g., weights and biases in a neural network). The goal of training is to optimize the model's parameters in such a way that it can accurately predict the outcome of new, unseen data. At the start of the experiment, 140 last records from EPL, SLL, and SA and 126 last records from GBL were separated from the final preprocessed dataset. This data is labeled as the testing data and will be used in the end for evaluation.

The training dataset, which consists of all records but the ones that were selected as the testing dataset, is subsequently split into a training set and a validation set. The validation set is used to find the best hyperparameter configuration for a particular problem. The hyperparameters of all models were described in Chapter 3. For the optimization of hyperparameters, I used Mean Absolute Error (MAE) which is a common metric used to measure the difference between two continuous variables, in this case, the predicted and actual values of the desired match statistic. The model is initialized with hyperparameters set to some predefined values and subsequently trained on the training dataset. The model produces its prediction for the validation set, and MAE is computed. If the MAE is the lowest up to that point, the values of the hyperparameters are marked as best and saved for later. This process is repeated until all predefined combinations of hyperparameters are tried out. After that, the best values of hyperparameters for the model predicting a given match statistic are found.

A model with the optimal hyperparameters is then trained on all but the testing data, meaning the training and validation sets combined. That results in the final model, which is going to be used for prediction on the test set. In the following table, the exact sizes of all the sets are shown:

Table 5.1 Sizes of Training, Validation and Testing sets

	Training	**Validation**	**Testing**
EPL	2145	750	140
SLL	2145	750	140
GBL	1660	554	126
SA	2249	750	140

5.3 Experimental Design

The training process and evaluation metrics have already been described. The whole process is now going to be described more generally.

All data preprocessing, model training, validation, and predictions can be done within just one league independently. Therefore, the program that produces the outputs (predictions of match statistics) for this thesis uses a single thread for each league. To simplify the explanation, I am going to describe the process for one league, as it is the same for every league. Twelve target variables are defined in the following order (for details, see Table 4.1):

- H_{shots}, A_{shots}
- H_{target}, A_{target}
- H_{corners}, A_{corners}
- H_{fouls}, A_{fouls}
- H_{yellow}, A_{yellow}
- H_{red}, A_{red}

The process starts with the data preprocessing and feature generation, which were described in the Section 4.3. The dataset and all the target variables are then saved into a *.csv* file. The first target variable's (H_{shots}) values are taken from the file that contains the target variables. Then the data is split into training, validation, and testing datasets. After that, all the non-matrix factorization methods are trained and fine-tuned one by one. Here are the all algorithms described in Chapter 3, with their corresponding abbreviations, which will be used in the final result tables.

PR stands for PoissonRegressor

Ridge stands for RidgeRegression

RF means Random Forest algorithm

XGB stands for XGboost algorithm

MLP is the Multi Layer Perceptron neural network

SVDpp is extended Singular Value Decomposition algorithm from the surprise library

BO is the BaselineOnly model from surprise library

OMIC is Orthogonal Inductive Matrix Completion

Mean is the mean of all training examples applied on the test dataset

After that, data for the matrix factorization methods is prepared by creating the desired matrix (model matrix shown in Table 3.1) described in the Section 4.4.

Now the trained machine learning models create their predictions on the test set, and the matrix completion is done by the three different algorithms. The last X number of matches (the predicted ones) is then extracted from the matrices. The team names and dates are then added to the predictions of the different models along with the original values, and those are saved one by one to files so that one can take a look at which model predicted what values in certain matches. Here is a sample result of one of the model's predictions for H_{shots}.

Table 5.2 Sample of the stored prediction result

Date	HomeTeam	AwayTeam	H_{shots}	Predicted
2022-02-26	Leeds	Tottenham	19.0	14.794
2022-02-27	West Ham	Wolves	13.0	13.913
2022-03-01	Burnley	Leicester	9.0	14.258
2022-03-05	Liverpool	West Ham	22.0	16.505
2022-03-05	Norwich	Brentford	15.0	13.108

The test set MAE for every model is computed and saved for later to create a final result table tables. The process then continues by repeating the same procedure but with different statistics until it completes all of them. A table of resulting MAEs showing all the algorithms' performances on all the match statistics is produced for each league.

5.4 Results

In this section, I am going to present and discuss the results of the experiments. The most trivial baseline for a regression problem is to output the mean of the training set values of the target variables and evaluate them on the test set. Thus, I used the mean as sanity check of the quality of the baselines. To comprehensively analyze all predictions, the results will be evaluated by the three different metrics that were described in Section 5.1.

In tables 5.3, 5.4, 5.5 and 5.6, MAEs (see Section 5.1.1) of all models on all statistics are shown. The models are presented in the order as they were introduced. The green cells show which model had the smallest error on a given statistic, and the red cells show the opposite, indicating which model had the biggest (worst) MAE in predicting a particular statistic.

Another type of metric that I used and evaluated models on was the RMSE introduced in Section 5.1.2. This metric works on a similar scale as the MAE, which can be easily interpreted by humans. But because the RMSE penalizes the bigger errors more, its results are a little different. The results are shown in tables 5.7, 5.8, 5.9 and 5.10, with the colors meaning the same thing as before. The green cells mark which model performed the best on a given statistic, and the red cells mark the worst model on a given statistic.

The last metric which I used was the R^2 score, described in Section 5.1.3. I opted for using this metric because of the fact that different match statistics follow different distributions and they operate on various scales (e.g., in the case of H_{shots} and H_{red}). So simply ranking the models by the aggregate of MAE or RMSE can have some downsides. The advantage of R^2 is that this metric is not affected by the target variables' scales. R^2 also measures how much of the variance in the target variable is explained by the model. So this is a meaningful metric of performance because it presents the results from a different point of view than the MAE and RMSE. In tables 5.11, 5.12, 5.13 and 5.14 these results can be seen. The green cells again show the best-performing model on a single statistic, and the red cells the worst one. Instead of the smallest values meaning the smallest error and therefore better results, like when MAE and

RMSE were used, here the larger values mark a better performance and smaller value a worse performance.

Table 5.3 EPL results **MAE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	3.572	3.583	3.728	3.801	4.331	4.105	4.069	4.23	4.836
A_{shots}	3.337	3.399	3.563	3.613	3.743	4.262	3.771	3.826	4.218
H_{target}	1.825	1.842	1.847	1.823	2.0	1.95	1.881	1.849	2.119
A_{target}	1.737	1.769	1.809	1.857	1.976	1.901	1.877	1.85	1.975
H_{corners}	2.304	2.302	2.342	2.275	2.46	2.415	2.456	2.417	2.634
A_{corners}	1.897	1.869	1.899	1.925	2.09	2.012	1.95	1.969	2.051
H_{fouls}	2.642	2.632	2.621	2.603	2.68	2.842	2.597	2.583	2.699
A_{fouls}	2.993	2.898	2.94	2.826	3.065	2.824	2.697	2.779	3.0
H_{yellow}	1.004	1.001	1.022	0.999	1.031	1.019	1.021	1.101	1.03
A_{yellow}	0.998	0.997	1.013	0.996	1.031	0.99	0.978	0.997	1.03
H_{red}	0.102	0.108	0.115	0.132	0.062	0.11	0.101	0.098	0.104
A_{red}	0.112	0.109	0.119	0.117	0.141	0.121	0.113	0.114	0.117

Table 5.4 SLL results **MAE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	3.58	3.606	3.638	4.051	4.04	3.908	3.548	3.62	3.654
A_{shots}	3.257	3.259	3.186	3.187	3.345	3.48	3.219	3.327	3.347
H_{target}	1.923	1.959	1.896	2.011	1.909	2.066	1.964	1.936	1.982
A_{target}	1.744	1.745	1.659	1.608	1.885	1.757	1.735	1.776	1.86
H_{corners}	2.265	2.266	2.2	2.363	2.205	2.258	2.201	2.216	2.238
A_{corners}	1.99	1.985	1.975	1.916	1.978	2.001	1.991	1.994	2.003
H_{fouls}	2.996	3.005	3.048	3.026	3.143	3.307	3.126	3.302	3.136
A_{fouls}	3.322	3.32	3.334	3.169	3.566	3.846	3.385	3.478	3.519
H_{yellow}	1.347	1.376	1.305	1.343	1.316	1.267	1.256	1.257	1.313
A_{yellow}	1.391	1.39	1.372	1.426	1.404	1.409	1.39	1.399	1.398
H_{red}	0.166	0.166	0.173	0.164	0.176	0.214	0.21	0.213	0.177
A_{red}	0.224	0.221	0.219	0.21	0.236	0.211	0.194	0.191	0.234

Table 5.5 GBL results **MAE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	3.225	3.25	3.243	3.275	3.51	3.746	3.438	3.424	3.697
A_{shots}	3.577	3.61	3.635	3.493	3.878	3.674	3.652	3.668	3.917
H_{target}	1.978	1.986	1.919	1.95	2.06	2.18	2.048	2.024	2.156
A_{target}	1.696	1.716	1.768	1.713	1.888	1.912	1.8	1.937	1.862
H_{corners}	2.11	2.101	2.164	2.028	2.105	2.151	2.079	2.116	2.091
A_{corners}	2.22	2.218	2.228	2.212	2.32	2.281	2.304	2.36	2.402
H_{fouls}	2.865	2.883	2.899	2.867	3.007	2.721	2.669	2.682	3.27
A_{fouls}	3.039	3.057	3.037	3.118	3.238	3.557	3.044	3.055	3.532
H_{yellow}	0.954	0.956	0.959	0.965	0.944	1.031	0.967	0.955	0.943
A_{yellow}	1.017	1.02	1.006	1.007	0.991	1.095	1.032	0.986	0.985
H_{red}	0.072	0.07	0.083	0.077	0.119	0.064	0.046	0.045	0.092
A_{red}	0.092	0.089	0.107	0.121	0.106	0.102	0.097	0.097	0.113

Table 5.6 SA results **MAE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	4.294	4.306	4.244	4.346	4.484	4.425	4.169	4.493	4.569
A_{shots}	4.028	4.044	4.048	4.082	3.955	4.515	4.0	4.052	4.348
H_{target}	1.951	1.963	1.962	1.902	2.103	1.888	1.91	1.929	2.187
A_{target}	1.84	1.836	1.879	1.897	1.877	1.878	1.811	1.902	1.913
H_{corners}	2.271	2.301	2.301	2.32	2.43	2.41	2.246	2.392	2.376
A_{corners}	2.139	2.135	2.133	2.089	2.204	2.186	2.135	2.171	2.275
H_{fouls}	2.969	2.946	2.877	2.937	3.145	3.45	3.108	3.258	3.176
A_{fouls}	3.021	2.996	3.074	3.011	3.296	3.323	3.149	3.093	3.395
H_{yellow}	1.099	1.091	1.102	1.072	1.111	1.146	1.105	1.08	1.099
A_{yellow}	1.12	1.121	1.14	1.091	1.194	1.191	1.179	1.198	1.149
H_{red}	0.195	0.192	0.191	0.21	0.224	0.234	0.22	0.215	0.184
A_{red}	0.209	0.196	0.214	0.225	0.303	0.201	0.193	0.206	0.22

Table 5.7 EPL results **RMSE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	4.442	4.466	4.691	4.8	5.438	5.278	4.921	5.223	5.91
A_{shots}	4.289	4.328	4.539	4.626	4.777	5.498	4.678	4.908	5.294
H_{target}	2.262	2.276	2.29	2.286	2.455	2.376	2.324	2.262	2.633
A_{target}	2.291	2.321	2.403	2.473	2.595	2.462	2.444	2.421	2.576
H_{corners}	2.88	2.89	2.938	2.951	3.083	3.034	3.006	3.007	3.235
A_{corners}	2.373	2.324	2.34	2.375	2.549	2.529	2.403	2.436	2.451
H_{fouls}	3.318	3.309	3.301	3.301	3.315	3.56	3.241	3.225	3.355
A_{fouls}	3.713	3.606	3.628	3.585	3.797	3.56	3.417	3.475	3.74
H_{yellow}	1.225	1.231	1.234	1.299	1.239	1.254	1.23	1.329	1.239
A_{yellow}	1.219	1.222	1.234	1.249	1.24	1.233	1.201	1.212	1.239
H_{red}	0.265	0.268	0.271	0.295	0.268	0.271	0.267	0.261	0.261
A_{red}	0.222	0.224	0.222	0.223	0.224	0.226	0.222	0.22	0.219

Table 5.8 SLL results **RMSE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	4.716	4.758	4.723	5.308	5.236	5.19	4.721	4.86	4.687
A_{shots}	4.057	4.078	4.008	3.975	4.123	4.267	3.967	4.063	4.125
H_{target}	2.418	2.467	2.393	2.658	2.387	2.588	2.492	2.469	2.51
A_{target}	2.097	2.097	2.013	2.012	2.237	2.155	2.092	2.123	2.202
H_{corners}	2.917	2.915	2.834	3.109	2.845	2.965	2.899	2.895	2.85
A_{corners}	2.412	2.42	2.398	2.377	2.414	2.509	2.43	2.411	2.426
H_{fouls}	3.736	3.739	3.738	3.729	3.885	4.042	3.837	4.076	3.872
A_{fouls}	4.163	4.154	4.271	4.04	4.515	4.708	4.224	4.346	4.4
H_{yellow}	1.663	1.688	1.613	1.675	1.599	1.611	1.579	1.586	1.602
A_{yellow}	1.681	1.679	1.658	1.805	1.699	1.703	1.684	1.691	1.689
H_{red}	0.271	0.27	0.272	0.282	0.272	0.305	0.294	0.305	0.272
A_{red}	0.36	0.361	0.358	0.357	0.363	0.382	0.372	0.369	0.363

Table 5.9 GBL results **RMSE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	4.244	4.304	4.22	4.311	4.641	4.619	4.242	4.237	4.604
A_{shots}	4.457	4.463	4.426	4.305	4.827	4.553	4.474	4.541	4.835
H_{target}	2.49	2.494	2.406	2.453	2.546	2.685	2.512	2.5	2.594
A_{target}	2.239	2.24	2.397	2.287	2.429	2.487	2.299	2.55	2.434
H_{corners}	2.551	2.523	2.615	2.471	2.544	2.545	2.47	2.565	2.537
A_{corners}	2.74	2.729	2.785	2.768	2.825	2.916	2.893	2.926	2.955
H_{fouls}	3.667	3.703	3.676	3.671	3.837	3.549	3.44	3.482	4.045
A_{fouls}	3.796	3.822	3.786	3.958	4.084	4.225	3.71	3.778	4.199
H_{yellow}	1.142	1.145	1.14	1.146	1.114	1.259	1.175	1.156	1.116
A_{yellow}	1.279	1.283	1.283	1.288	1.275	1.349	1.293	1.275	1.282
H_{red}	0.157	0.158	0.152	0.154	0.17	0.168	0.158	0.157	0.16
A_{red}	0.182	0.184	0.187	0.206	0.184	0.191	0.18	0.181	0.184

Table 5.10 SA results **RMSE**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	5.943	5.966	5.905	6.113	6.165	6.04	5.994	6.179	6.551
A_{shots}	5.353	5.376	5.388	5.453	5.368	5.794	5.287	5.277	5.766
H_{target}	2.464	2.485	2.447	2.435	2.68	2.464	2.498	2.503	2.77
A_{target}	2.394	2.379	2.36	2.389	2.372	2.386	2.333	2.41	2.426
H_{corners}	2.912	2.95	2.951	3.102	3.285	3.169	3.021	3.228	3.152
A_{corners}	2.696	2.687	2.668	2.683	2.8	2.798	2.714	2.75	2.814
H_{fouls}	3.697	3.682	3.67	3.742	3.863	4.325	3.949	4.163	3.895
A_{fouls}	3.778	3.757	3.798	3.817	4.035	4.004	3.92	3.861	4.095
H_{yellow}	1.395	1.396	1.39	1.401	1.405	1.468	1.423	1.41	1.403
A_{yellow}	1.439	1.439	1.447	1.512	1.485	1.496	1.484	1.503	1.462
H_{red}	0.332	0.325	0.323	0.336	0.33	0.38	0.357	0.353	0.321
A_{red}	0.317	0.326	0.319	0.326	0.352	0.339	0.327	0.342	0.32

Table 5.11 EPL results **R2**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	0.431	0.424	0.365	0.335	0.147	0.208	0.311	0.225	-0.008
A_{shots}	0.332	0.32	0.252	0.223	0.172	-0.087	0.213	0.138	-0.017
H_{target}	0.23	0.221	0.211	0.214	0.093	0.173	0.21	0.224	-0.043
A_{target}	0.208	0.187	0.128	0.077	-0.016	0.101	0.114	0.121	-0.002
H_{corners}	0.188	0.182	0.155	0.147	0.069	0.108	0.124	0.129	-0.025
A_{corners}	0.063	0.101	0.088	0.061	-0.082	-0.044	0.058	0.038	-0.0
H_{fouls}	-0.016	-0.01	-0.005	-0.006	-0.014	-0.18	0.022	0.029	-0.039
A_{fouls}	-0.046	0.014	0.001	0.025	-0.094	0.037	0.113	0.088	-0.061
H_{yellow}	0.022	0.012	0.007	-0.1	-0.0	-0.015	0.024	-0.144	-0.0
A_{yellow}	0.031	0.027	0.009	-0.017	-0.002	-0.016	0.036	0.025	-0.0
H_{red}	-0.031	-0.054	-0.074	-0.274	-0.057	-0.076	-0.044	0.004	-0.0
A_{red}	-0.038	-0.061	-0.039	-0.051	-0.054	-0.072	-0.039	-0.009	-0.013

Table 5.12 SLL results **R2**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	-0.02	-0.038	-0.023	-0.292	-0.257	-0.235	-0.022	-0.08	-0.007
A_{shots}	0.023	0.013	0.047	0.062	-0.009	-0.08	0.066	0.026	-0.01
H_{target}	0.071	0.033	0.091	-0.122	0.095	-0.064	0.014	0.024	-0.001
A_{target}	0.073	0.072	0.145	0.146	-0.056	0.02	0.076	0.052	-0.023
H_{corners}	-0.048	-0.047	0.011	-0.19	0.004	-0.083	-0.035	-0.024	-0.0
A_{corners}	-0.012	-0.019	-0.0	0.017	-0.014	-0.095	-0.027	-0.006	-0.024
H_{fouls}	0.047	0.045	0.046	0.05	-0.031	-0.116	-0.005	-0.137	-0.024
A_{fouls}	0.01	0.014	-0.042	0.068	-0.165	-0.266	-0.019	-0.092	-0.106
H_{yellow}	-0.086	-0.118	-0.021	-0.101	-0.004	-0.019	0.021	0.019	-0.007
A_{yellow}	0.003	0.005	0.03	-0.15	-0.018	-0.024	-0.0	-0.006	-0.007
H_{red}	-0.014	-0.009	-0.025	-0.097	-0.019	-0.289	-0.193	-0.279	-0.02
A_{red}	0.016	0.011	0.026	0.031	-0.0	-0.108	-0.053	-0.031	-0.0

Table 5.13 GBL results **R2**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	0.147	0.123	0.157	0.12	-0.02	-0.011	0.148	0.156	-0.004
A_{shots}	0.147	0.145	0.159	0.204	-0.0	0.11	0.141	0.12	-0.003
H_{target}	0.076	0.074	0.138	0.104	0.035	-0.074	0.06	0.066	-0.002
A_{target}	0.153	0.153	0.029	0.116	0.003	-0.045	0.107	-0.09	-0.0
H_{corners}	-0.025	-0.003	-0.077	0.038	-0.019	-0.02	0.039	-0.03	-0.014
A_{corners}	0.138	0.145	0.109	0.12	0.084	0.024	0.039	0.019	-0.003
H_{fouls}	-0.006	-0.026	-0.011	-0.008	-0.101	0.058	0.114	0.097	-0.224
A_{fouls}	0.05	0.037	0.055	-0.033	-0.099	-0.177	0.093	0.018	-0.162
H_{yellow}	-0.06	-0.066	-0.058	-0.068	-0.009	-0.29	-0.123	-0.094	-0.013
A_{yellow}	-0.006	-0.013	-0.013	-0.02	-0.001	-0.12	-0.029	0.005	-0.011
H_{red}	-0.056	-0.077	0.006	-0.016	-0.25	-0.219	-0.08	-0.056	-0.097
A_{red}	-0.078	-0.096	-0.142	-0.386	-0.107	-0.184	-0.051	-0.059	-0.098

Table 5.14 SA results **R2**

Variable	PR	Ridge	RF	XGB	MLP	SVDpp	BO	OMIC	Mean
H_{shots}	0.167	0.16	0.177	0.118	0.103	0.146	0.159	0.112	-0.013
A_{shots}	0.122	0.115	0.111	0.089	0.118	-0.046	0.129	0.138	-0.018
H_{target}	0.191	0.177	0.202	0.21	0.043	0.191	0.169	0.172	-0.022
A_{target}	0.025	0.037	0.052	0.029	0.043	0.032	0.074	0.019	-0.002
H_{corners}	0.144	0.122	0.121	0.029	-0.09	-0.01	0.083	-0.04	-0.003
A_{corners}	0.072	0.079	0.092	0.082	-0.001	-0.002	0.057	0.038	-0.011
H_{fouls}	0.082	0.089	0.095	0.059	-0.003	-0.257	-0.048	-0.163	-0.02
A_{fouls}	0.104	0.114	0.095	0.085	-0.022	-0.0	0.041	0.072	-0.053
H_{yellow}	0.01	0.009	0.018	0.002	-0.004	-0.092	-0.026	-0.002	-0.001
A_{yellow}	0.029	0.029	0.018	-0.072	-0.034	-0.054	-0.037	-0.058	-0.002
H_{red}	-0.092	-0.046	-0.035	-0.118	-0.076	-0.347	-0.19	-0.157	-0.018
A_{red}	-0.02	-0.078	-0.034	-0.082	-0.261	-0.163	-0.083	-0.181	-0.037

5.4.1 Discussion

The matrix factorization methods showed that in predicting football match statistics, they can compete very well with the standard machine learning techniques. Overall, the simplest matrix factorization model, called BaselineOnly, had some good results in predicting fouls and yellow cards. This may be because the data is relatively simple and not in large numbers. Because of this fact, the models from the standard machine learning techniques like the PoissonRegressor, XGboost and Random Forest models did very well in general, mostly in the first eight match statistics, which operate on a higher scale than the last four. Surprisingly, the Multi Layer Perceptron artificial neural network was one of the worst models. That can be caused by not having enough training data. If the dataset used to train the model is smaller (relatively), it can lead to overfitting, where the model learns the noise in the data rather than the underlying patterns.

The qualities of the final matches chosen for evaluation also had an impact on the results. For example, in the SLL, the last matches in the dataset that were selected for the evaluation must show some different trends than the training data. In all three other leagues, the models performed significantly better, which can be seen very well in the end when comparing tables of R^2 scores. Simply put, the last matches in the SLL were probably more unpredictable than the last matches in the other leagues, which is something that can happen in a football season, especially at the end. However, if the results from figures 4.2, 4.3, 4.4 and 4.5 are considered, it is clear that in some leagues the match statistics have higher correlations between themselves than in other leagues, with SLL having the least positive correlations. On the other hand, the EPL's correlation matrix, which has the most positive correlations, also has the highest R^2 scores in numerous match statistics compared to the other leagues.

The MAE of the prediction by the Mean is sometimes close to other models' errors. The variables have their own different scales and distribution, which can be seen in Figure 4.1 and also in tables 4.2, 4.3, 4.4 and 4.5. So although the MAE in shots in general is somewhere around 3.6, it is caused by the means being around 13. The same applies for all other target variables. The MAE and later RMSE vary across the different match statistics and are directly proportional to the variable's mean, while they also depend on the distribution and standard deviation. In [19] they tried to predict corners, shots on target, and shots off target. The results of shots on target and corners differ by less than 0.1. This being the only work I found that tries to predict the exact values of match statistics, the MAE results achieved in this thesis seem decent. The ensemble algorithm XGBoost [25] performed well in terms of MAE, although it had some worst performances; each time on a different one, it was the algorithm with the smallest error thirteen times. It was better than most of the other algorithms when predicting corners, shots on target, and yellow cards. PoissonRegressor showed promising results in terms of MAE in predicting shots and was very stable, never being marked as being the worst in predicting any statistic. Overall, the BaselineOnly matrix completion algorithm did well and was the best performing algorithm nine times and never the worst. This model was particularly useful for predicting yellow cards and fouls, which are two strongly correlated match statistics. Unfortunately, it is clear that predicting variables like red or yellow cards is hard because of the scale and distribution of these statistics. The results achieved by the models do not differ that much from the MAEs of the Mean prediction, which was even ranked as the best model three times out of the sixteen possible, when predicting one of the statistics concerning cards.

In the RMSE evaluation, the matrix completion algorithm BaselineOnly confirmed its usefulness in estimating fouls and yellow cards, having multiple good results and again not being the model with the highest error in any case. The PoissonRegressor and Random Forest algorithms ranked a bit better compared to the MAE results. PR scored well, especially in the EPL, and the RidgeRegression model was very close to it on all the first six statistics. RF proved its value mostly in predicting in the GBL and SA; it did well on shots. XGBoost algorithm's performance had a lot of highs but also a lot of lows across all four leagues. These results confirm which mod-

els might be better at handling larger errors, while the models that performed better when MAE was used instead of RMSE might be better at dealing with smaller errors and are more "careful" (closer to the mean) with their predictions. Here it is partly confirmed that the SLL or its last matches selected for testing are different compared to the other leagues. In SLL, the prediction by mean was the best in H_{shots}, which shouldn't be the case with the match statistic operating on the highest scale, and was not the worst in prediction of any of the statistics. Whereas in all other leagues, the prediction done by the calculation of the previous mean ended up being good only in case of statistics with smaller scales and smaller numbers of possible values.

In the results evaluated by the R^2 score, the PoissonRegressor and Random Forest models performed very similarly to the RMSE results. PR again dominated the first five variables in the EPL with RidgeRegression being the best in the sixth one, the A_{corners}. RF was the best model for the same target variables. It is interesting that the BaselineOnly algorithm did not lose anything of its past performances when evaluating the R^2 score, showing that it managed to make its good predictions in a robust way. Again, the case of XGBoost is an interesting algorithm's performance in the SLL. It was the best model five times and the worst one four times. Generally, the results were quite similar to the RMSE results, probably because both metrics focus more on how well the models do in predicting values that differ more from the mean. From briefly looking at how the values differentiate across the four leagues, it is clear that the EPL (or its last 140 matches) are more predictable than the other leagues and that the SLL (or again, its last 140 matches) is much harder to predict because all values of the R^2 are very close to zero from both sides.

In conclusion, the predictions of shots, shots on target, corners, and fouls have shown more promising results than the ones concerned with cards. Cards are harder to predict, especially the red ones, because of the small number of different values they have; thus, they can be considered more random and extreme. Classifying the match as one where there will be a red card from one team or one where there won't may produce better predictions in the case of red cards. Better and more usable results in predicting yellow cards may be achieved by the "O/U" prediction of some chosen value, although some algorithms have shown their usefulness in predicting yellow cards by being the best ones on multiple occasions. The other statistics showed more promising results and can certainly be enhanced by using more data, particularly some features describing the teams' attacks and defense strengths.

Chapter 6
Conclusion

This research is dedicated to the prediction of statistical aspects in the sport of football. To accomplish this, at first, an extensive literature review was conducted, as outlined in Chapter 2, to explore previous works in the field. It was discovered that forecasting the precise values of match statistics in football remains a relatively underexplored area. However, this task shares several similarities with other prediction tasks, such as predicting match outcomes, "over/under" goal predictions, and various other statistics.

In Chapter 3, all the proposed methods for prediction are described. I used standard machine learning techniques, which were trained on matches from almost ten seasons prior to the matches the prediction was done on, as well as matrix factorization methods that predicted the desired statistic based just on the values of the given statistic in different matches of that season. The matrix completion methods showed promising results considering that this is not a typical use case for these methods. Their results were comparable with those of the other machine learning techniques.

Another goal was to gather open data from the world's four top football leagues, analyze it, and prepare it for the training process of the models. This was all explained in Chapter 4, which begins by describing the data sources. It continues with a brief analysis of the data and then an explanation of all the features that have been created from the original datasets. When creating features, the emphasis is placed on making the resulting dataset usable in practice. This means that all features are made up only of columns from previous matches and can be known before the match.

In the end, the experiment was conducted and described in Chapter 5. First I introduced the three different evaluation metrics, which were then used for assessing the models' performances from different perspectives. After that, training and validation procedures were described. Here the best configurations of different models are found and optimized, to be used for the final prediction. Subsequently, the prediction results were presented. The three different metrics offered different insights into how different models performed. In each statistic, the mean of that statistic in the training dataset was used as a prediction, and the errors produced from that were used as baselines to evaluate whether the models performed well or not. The Poisson regression model performed the best overall, with Random Forest, Ridge regression, and the matrix completion algorithm from the *surprise* library called BaselineOnly being not that far behind. The results cannot be judged as excellent or bad because I have not found any work that would try to predict the same statistics in a similar way. In [19], computing some statistics is not the main task of the work, but there is an effort to compute some of them. The differences between the error of the mean (which was used as a baseline "model" in that research as well as in mine) and the errors of predictions done by the machine learning models achieved in my thesis are bigger in a positive way (predictive models outperforming the mean). The results can

surely be improved, but considering how little has been done in this field, they look promising and set a certain baseline for further analysis and experiments.

Although this study has provided valuable insights into the prediction of match statistics in football, there is still a lot of work to do in this relatively unexplored field of predicting. In this section, I highlight some potential directions for future work. As mentioned before, when describing [19], knowing the match's statistics can help a lot when predicting the outcome. I achieved results that are better than the mean in a lot of the match statistics, and that is the first step. Looking back, there is a lot of room for improvement. I would consider the use of some more data, for example, the estimated attacking and defending strengths of teams, which can be extracted from the FIFA [12] games and are also used in some other works concerned with predicting football. Also, many more features can be engineered, especially if more data can be collected.

Bibliography

1. *Qatar 2022: World Cup final scores 1.5 bn global viewers*. 2023. Available also from: https://www.sportspromedia.com/news/qatar-2022-fifa-world-cup-final-argentina-france-viewers-engagement/. [Online; accessed 2023-04-18].
2. MESSINGER, Trisha. *Sports Betting Market Worth US$ 225.65 Billion By 2022 - 2031*. 2022. Available also from: https://www.linkedin.com/pulse/sports-betting-market-worth-us-22565-billion-2022-2031-messinger. [Online; accessed 2023-05-5].
3. HC, Chethan. *Most Watched football League In The World Now 2023*. 2023. Available also from: https://urfootball.com/top-5-most-watched-football-league-in-the-world/. [Online; accessed 2023-04-22].
4. . *Country coefficients — UEFA Coefficients*. 2023. Available also from: https://www.uefa.com/nationalassociations/uefarankings/country/. [Online; accessed 2023-05-06].
5. YIP, Stan; ZOU, Yinghong; HUNG, Ronald Tsz Hin; YIU, Ka Fai Cedric. Forecasting number of corner kicks taken in association football using overdispersed distribution. *arXiv preprint arXiv:2112.13001*. 2021. Available also from: https://arxiv.org/abs/2112.13001.
6. WHEATCROFT, Edward. A profitable model for predicting the over/under market in football. *International Journal of Forecasting*. 2020, vol. 36, no. 3, pp. 916–932. Available from DOI: 10.1016/j.ijforecast.2019.11.001.
7. FOOTBALL-DATA.CO.UK. *Football Data*. 2023. Available also from: https://www.football-data.co.uk/. [Online; accessed 2023-03-20].
8. KELLY, James J. A New Interpretation of Information Rate. *Bell System Technical Journal*. 1956, vol. 35, no. 4, pp. 917–926. Available from DOI: 10.1002/j.1538-7305.1956.tb03809.x.
9. VAKNIN, Erik. *Predicting Score-related Events in Soccer*. Prague, CZ, 2021. MA thesis. Czech Technical University in Prague, Faculty of Electrical Engineering. Available at http://hdl.handle.net/10467/95294.
10. TAX, Niek; JOUSTRA, Yme. *Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approach*. 2015. Available from DOI: 10.13140/RG.2.1.1383.4729.
11. RODRIGUES, Fátima; PINTO, Ângelo. Prediction of football match results with Machine Learning. *Procedia Computer Science*. 2022, vol. 204, pp. 463–470. ISSN 1877-0509. Available from DOI: https://doi.org/10.1016/j.procs.2022.08.057. International Conference on Industry Sciences and Computer Science Innovation.

12. *FIFA player ratings explained: How are the card number stats decided? — Goal.com*. 2019. Available also from: https://www.goal.com/en/news/fifa-player-ratings-explained-how-are-the-card-number--stats-decided/1hszd2fgr7wgf1n2b2yjdpgynu. [Online; accessed 2023-04-19].
13. BABOOTA, Rahul; KAUR, Harleen. Predictive analysis and modelling football results using machine learning approach for English Premier League. *International Journal of Forecasting*. 2019, vol. 35, no. 2, pp. 741–755. Available from DOI: 10.1016/j.ijforecast.2018.01.003.
14. EPSTEIN, Edward S. A Scoring System for Probability Forecasts of Ranked Categories. *Journal of applied meteorology*. 1969, vol. 8, no. 6, pp. 985–987. Available from DOI: 10.1175/1520-0450(1969)008.
15. RAZALI, Nazim; MUSTAPHA, Aida; YATIM, Faiz Ahmad; AZIZ, Ruhaya Ab. Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL). *IOP Conference Series: Materials Science and Engineering*. 2017, vol. 226, p. 012099. Available from DOI: 10.1088/1757-899x/226/1/012099.
16. RAHMAN, Md. Ashiqur. A deep learning framework for football match prediction. *SN applied sciences*. 2020, vol. 2, no. 2. Available from DOI: 10.1007/s42452-019-1821-5.
17. KAMPAKIS, Stylianos; ADAMIDES, Andreas. Using Twitter to predict football outcomes. *ArXiv*. 2014, vol. abs/1411.1243.
18. RUSSELL, Stuart J.; NORVIG, Peter. *Artificial Intelligence: a modern approach*. 3rd ed. Pearson, 2009.
19. WHEATCROFT, Edward. Forecasting football matches by predicting match statistics. *Journal of sports analytics*. 2021, vol. 7, no. 2, pp. 77–97. Available from DOI: 10.3233/jsa-200462.
20. *Jupyter Notebook: An Introduction*. 2023. Available also from: https://realpython.com/jupyter-notebook-introduction/. [Online; accessed 2023-04-19].
21. MAHER, M. J. Modelling association football scores. *Statistica Neerlandica*. 1982, vol. 36, no. 3, pp. 109–118. Available from DOI: 10.1111/j.1467-9574.1982.tb00782.x.
22. KOOPMAN, Siem Jan; LIT, Rutger. Forecasting football match results in national league competitions using score-driven time series models. *International Journal of Forecasting*. 2019, vol. 35, no. 2, pp. 797–809. Available from DOI: 10.1016/j.ijforecast.2018.10.011.
23. HUBÁČEK, Ondřej; ŠOUREK, Gustav; ŽELEZNÝ, Filip. Learning to predict soccer results from relational data with gradient boosted trees. *Machine Learning*. 2019, vol. 108, no. 1, pp. 29–47. Available from DOI: 10.1007/s10994-018-5704-6.
24. BREIMAN, Leo. Bagging predictors. *Machine Learning*. 1996, vol. 24, no. 2, pp. 123–140. Available from DOI: 10.1007/bf00058655.
25. CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, USA: ACM, 2016, pp. 785–794. KDD '16. ISBN 978-1-4503-4232-2. Available from DOI: 10.1145/2939672.2939785.
26. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.
27. GOLUB, Gene H.; VAN LOAN, Charles F. *Matrix Computations*. Third. The Johns Hopkins University Press, 1996.
28. HUG, Nicolas. Surprise: A Python library for recommender systems. *Journal of Open Source Software*. 2020, vol. 5, no. 52, p. 2174. Available from DOI: 10.21105/joss.02174.

29. SILVA ALVES, Rodrigo Augusto da. *Towards Comprehensive Cluster-induced Methods for Recommender Systems*. Kaiserslautern, Germany, 2022. PhD thesis. Technische Universitat Kaiserslautern. Available at `https://kluedo.ub.rptu.de/frontdoor/deliver/index/docId/6734/file/Alves_Rodrigo_PhD_Thesis.pdf`.

30. FBREF.COM. *FBref*. 2023. Available also from: `https://fbref.com/`. [Online; accessed 2023-03-20].

31. TEAM, The pandas development. *pandas-dev/pandas: Pandas*. Zenodo, 2020. Version 1.4.2. Available from DOI: `10.5281/zenodo.3509134`.

32. HAO, Jiangang; HO, Tin Kam. Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*. 2019, vol. 44, no. 3, pp. 348–361. Available from DOI: `10.3102/1076998619832248`.

Contents of the attached media

- app
  - prediction_dfs ..... csv files with all predictions of all models
  - app.py ..... the main .py file which needs to be executed for the run
  - data_preparation.py ..... contains code for data preparation
  - league_prediction.py ..... contains code for the whole process for one league
  - OMIC.py ..... contains code of the implementation of the OMIC algorithm
  - prediction.py ..... .py file containing code for model training and prediction
- data ..... directory with all the csv files with data and preprocessed data
- thesis
  - FITthesis_LaTeX_Ondrej_Herman.pdf ..... text of the thesis in PDF format
  - FITthesis_Ondrej_Herman.zip ..... source code of the thesis in LaTeX
- data_preprocess.ipynb ..... jupyter notebook for trying out data preprocessing
- Recommender.ipynb ... jupyter notebook for trying out matrix factorization techniques and data prep