LEOPOLD-FRANZENS-UNIVERSITÄT INNSBRUCK

MASTERTHESIS

DEPARTMENT OF INFORMATION SYSTEMS, PRODUCTION AND LOGISTICS MANAGEMENT

Prediction Of English Premier League Soccer Matches, Based On Player Data Using Supervised Learning

Author:	Giuliano Gaub
Supervisor:	Assoz.-Prof. Mag. Mag. Stefan Haeussler, PhD
Submission Date:	22.12.2022

Abstract

Football is one of the most popular sports watched by millions of people worldwide. The question of how the game will end is not only of interest to fans but also to players of online football games or betting providers. With the increasing amount of data collected on football matches, there are more opportunities to build machine learning models to predict matches. Most approaches are based on team data, such as goals scored, shots taken or fouls committed. This paper, however, investigates whether a prediction model based on team statistics can be improved by adding information about individual players. Self-calculated values based on match statistics and values from the Fantasy League are used to evaluate the football players. 350 football results (win, draw or defeat) of the English Premier League for the season 2021/2022 are predicted. To find out whether the performance of the model can be improved, a model based on match statistics is created in the first step, which is supplemented with values of individual players in the second step to create another model. The performance of the two models is subsequently compared. As a machine learning method, the Random Forest Classifier from the field of supervised learning is used. The result of the research is that the accuracy of the predictive model can be increased by over 2% to 61% by adding the player data. Compared to the benchmarks, the model performs better than the community votes but worse than the bookmakers.

Keywords: *Machine Learning, Supervised Learning, Random Forest, Players' data, Result prediction, English Premier League*

Contents

List of Figures

List of Tables

Acronyms

AWS Amazon Web Services

csv comma-separated values

CV Cross Validation

EPL English Premier League

GCA Goal-Creating Actions

HTML Hypertext Markup Language

MDI Mean Decrease in Impurity

OR Operational research

RPS Ranked Probability Score

RQ Research Question

SCA Shot-Creating Actions

xAG Expected Assisted Goals

xG Expected Goals

1 Introduction

In 2021, before the European Championship started, the University of Innsbruck published a forecast on the outcome of the tournament using machine learning methods. Groll, Hvattum, Ley, et al. (2021) calculated the highest probability for France to win the tournament, followed by England and Spain using hybrid machine learning methods. The eventual winner Italy, followed only in seventh place, with a win probability of 7.9% (Groll, Hvattum, Ley, et al. 2021).

However, looking back to the 2014 World Cup, Microsoft's prediction for the knockout stage reached 100% accuracy, meaning it predicted the correct winner for each match and was thus able to predict Germany as the world champion after the group stage (Sullivan 2014).

These are two examples of predicting the result of football matches, but the field of operational research in sports, also called sports analytics, encompasses much more. Nowadays, the methods of sports analytics are applied not only in football but in almost all relevant team and individual sports (Fried, Mumcu, et al. 2016). In recent years, teams have been investing increasing amounts of money in collecting and analyzing data to improve their scouting, training and tactics. Furthermore, media companies are presenting live statistics during the matches provided by data companies like Amazon Web Services (AWS) for the Bundesliga or Oracle for the English Premier League (EPL). Also, the sports betting market has grown significantly in recent years, with the total turnover in Germany more than doubling between 2014 and 2019 from 4.5 to 9.3 billion euros (Graefe 2022). All these examples show how interesting the topic of sports analytics currently is for various parties, and it is assumed that the market will grow by another 30% until 2027 (Mordorintelligence 2021).

From the large field of sports analytics, this work focuses on evaluating players based on statistics and the result prediction of football matches of the EPL, as this league is currently the strongest and most balanced league in the world. There is a lot of literature on the topic of result prediction in football, but very few address individual player performance. Therefore, the focus of this thesis is, in the first step, to calculate the current form of a player based on game statistics and then, in the second step, find out whether including the players' strength can improve the prediction result.

For the evaluation of the players, values of the Fantasy League and a self-created rating are used. The Fantasy League is a game for the community where participants are team managers who buy players and form a team with a certain virtual budget and then they receive points depending on how well their team members perform on the pitch. The player's performance is calculated from various data, such as goals, passes and successful defensive actions. Since it is not comprehensible how all values of the Fantasy League are calculated, an own rating of the players has generated additionally. This leads to the first Research Question (RQ). RQ1: To what extent can the current form of each football player be calculated based on data from recent matches, distinguishing between goalkeeper, defense, midfield, and attack?

To find out whether player ratings can improve the prediction model, two machine learning models are created. The first model contains classic match statistics, while the second model additionally includes the player ratings. Then the models are compared to find out what influence the player ratings have, which leads to the second RQ. RQ2: To what degree does the calculated form of the players improve the prediction result?

To answer the two questions, this thesis is structured as follows:

Chapter 2 gives an overview of the existing literature on operational research in sports over time and the prediction of football outcomes, especially including individual player data. Subsequently, an overview of the technical setup and the data used is shown in chapter 3. Chapter 4 describes the methods used to analyse the research questions and chapter 5 presents the results of applying the methods to the data. Finally, in chapter 6, the main findings of the thesis are summarised and a further outlook is given.

2 Literature Review

2.1 Operational Research in Sports Over Time

Operational research (OR) is an analytical method for problem-solving and decision-making that is useful for managing organizations. In OR, problems are decomposed into basic elements and then solved at certain steps through mathematical analysis (S. Lewis 2019).

OR can be applied in all facets of sports, whether creating the game schedule for a volleyball tournament, improving the tactics of a handball team, choosing the right golf club, or placing a bet for a soccer match. That is why OR has been used in sports for more than 60 years. In recent years, the term "*Sports Analytics*" has been established in literature, which describes the same as OR in sports.

M. Wright (2009) provides an overview of research on the topic over the last 50 years. The first relevant study on OR in sports was conducted by Mottley (1954), who used OR practices from military operations for the sports of American Football and basketball. Mottley aimed to improve the team performance by optimizing the training, scouting and tactics of the teams (Mottley 1954).

From then on, the popularity of the topic has increased over time. The book *"Moneyball"* by M. Lewis, which appeared in 2004, brought the topic to the attention of the general public, which should boost research significantly in the coming years. The book, which was turned into a film in 2011, is about a protagonist who builds a successful team for Major League Baseball (MLB) in the USA with few resources. The key to success was the clever combination of players whose market value was undervalued on the one hand and who possessed complementary qualities on the other. The financial and sporting decisions were made based on publicly available baseball data and a mathematical valuation model (Baumer and Zimbalist 2014). The story is based on a true incident, which illustrates that the application of OR in sports can indeed lead to success (M. Lewis 2004) (Link 2018).

In the meantime, it is not only the athletes or teams who are interested in the data but also the economy, fans and science. Table 2.1 provides an overview of the fields in which sports analytics is used for certain applications.

The first area is professional sports, where coaching teams use action data from the competition to perform tactical analysis (Carling, Williams, and Reilly 2007). Tracking systems can be used to measure the amount of load during training or competition and the coaches can control the load for the individual players based on the measured values. This data is also of interest for the medical departments, as they try to reduce the risk of injury through overload (Dvorak, Junge, Chomiak, et al. 2016).

Another important area of application of match data in professional sports is the informational support of management decisions in player transfers and player contracts. Given the increasing sums for transfer payments, salaries and performance bonuses, the challenge is to identify talented players at an early stage and employ them under favourable conditions (Buraimo, Frick, Hickfang, and Simmons 2015).

A second area is the economy, including television and other media which have significantly developed their broadcasts in recent years due to the available data. Data service providers such as Opta Sports are providing live data during the game, showing the spectator, for example in football, how high the probability is that a player will score a goal from a certain position. Big data companies such as Microsoft or Oracle use sport as a demonstration domain for the performance of their data analytics components (Kim and Park 2015).

Other stakeholders in sports analytics are betting companies. They use sports data to design betting products or to detect betting manipulation (Deutscher, Dimant, and Humphreys 2017). If the concept of sports data is defined more broadly, activities such as the optimisation of ticket prices, analyses of fan satisfaction or the marketing of fan merchandise can be added to this area as well (Panchanathan, Chakraborty, McDaniel, et al. 2016).

The last area is science, which is divided into sports science and computer science. Sports science can benefit from the huge data resources of leagues, federations, clubs, and sports data companies. The data can be used in many ways, e.g. to assess the effectiveness of existing training concepts, to examine doctrines of sports practice, to define performance standards or to develop new paradigms of theory building (Hanley 2016). The contribution potential of computer science lies in both technological and analytical fields (Perl 2002). An example of this is Big Data. This includes data that is characterised by its large volume, the high speed of data production and the heterogeneity of the data sources (Ohlhorst 2012). This characteristic can increasingly be observed in the inventories of professional clubs. Here, data is available from position recording systems, scouting measures, performance physiology and sport motor tests as well as medical examinations.

Interest group	Primary objective
(Professional) Sports	
Coaching team, Athlete	Performance diagnostics, Load control, Strategy development
Medical department	Injury prevention, Rehabilitation
Scouting Department	Player ratings
Management	Contract design, Marketing
Economy	
TV, Other media	Entertainment, Media enhancement, Differentiation from competitors
Data service provider	Data collection, Preliminary products for professional sports and media
Sponsors	Transport of advertising messages, Contact numbers
IT companies	Demonstration domain for data analytics products
Stadium manager	Improving the stadium experience
Betting provider	Betting products, Odds calculation, Betting scam detection
Game maker	Improve the gaming experience, Differentiate from competitors
Leagues, Associations	Regulation, Marketing, Evaluation of rule changes
Fans	Entertainment
Science	
Sports science	Performance structure analyses, Development of sports science models
Computer science	Development and validation of computer science methods

Table 2.1: Primary objectives of non-academic and academic stakeholders in sports analytics (Link 2018, p. 16)

With the collection of more data in recent years, sports data analytics is also increasing rapidly, and it is expected that the market value will grow up to four billion USD by the end of 2022 (Ricky 2021). From the broad field of sports analytics, this thesis deals with result prediction in football using player data. The following chapters first provide an overview of the literature on sports result prediction in general and then take a look at the literature that explicitly deals with the prediction of sports events using player data.

2.2 Sports Result Prediction Using Machine Learning

In sports, people have always debated who will win the next competition or the next game. Today, large amounts of data are collected about sporting events that can be used to predict future events. The predictions can be made based on experience or data from the past, which leads to the concept of machine learning. Machine learning is defined as a branch of artificial intelligence that allows systems to automatically learn and improve from experience or data without the need for explicit programming (Rebala, Ravi, and Churiwala 2019). Within machine learning, a distinction is usually made between the three categories - supervised learning, unsupervised learning, and reinforcement learning (Alpaydin 2020).

Supervised learning is a method that makes predictions based on an analysis of input data with a given target variable. A distinction is made between classification, where discrete class labels are predicted (ordinal), and regression, which predicts a continuous variable (metric) (Kotsiantis, Zaharakis, Pintelas, et al. 2007).

Unsupervised learning, on the other hand, involves methods that search for previously unknown patterns and correlations in uncategorised data. A typical application is clustering, where unlabelled examples are grouped (Hastie, Tibshirani, and Friedman 2009).

Reinforcement learning stands for a machine learning method where an agent independently learns a strategy to maximise the reward it receives based on a reward function. The agent has independently learned which action is best in which situation. Of course, the reward can also be negative if the agent chooses an action that does not correspond to the reward function (Sutton, Barto, et al. 1998). The reward function describes the value of a certain state or action.

In summary, supervised learning involves training a model on labelled data to make predictions, unsupervised learning involves training a model on unlabeled data to discover hidden patterns, and reinforcement learning involves training an agent to make good decisions in an environment to maximize a reward (Michelle 2021).

In Figure 2.1, the different types of machine learning algorithms are shown. The prediction of football games is mostly handled as a classification problem from the field of supervised learning, where the outcome is categorized as a win, draw or loss.

The first relevant publication on the prediction of results using machine learning methods was done by Purucker (1996). Purucker used unsupervised learning techniques to predict football games in the National Football League (NFL). He used data from game weeks 12 to 15 to predict the 16th game week. Out of 14 games, his prediction was correct eleven times, giving an accuracy of 78.6%. A weakness of the study was the relatively small data set used for training and evaluation. Compared to football, there is no draw in American Football, which means there are only two possible outcomes of a match, making it easier to predict (Purucker 1996).

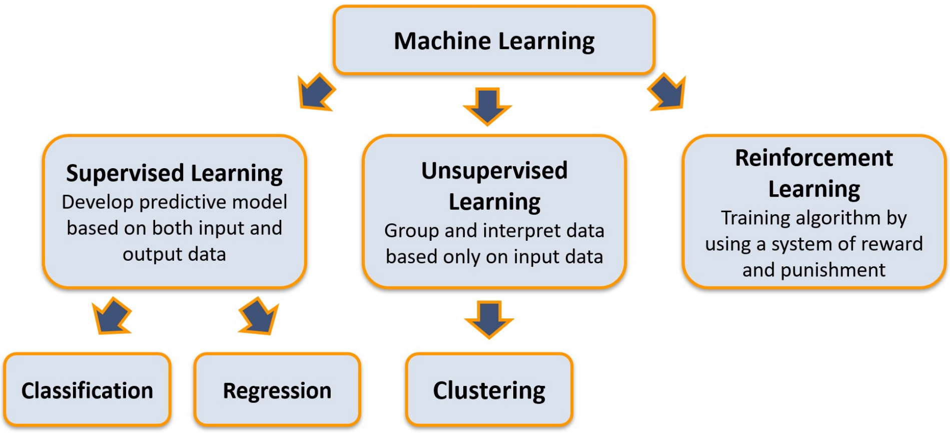

Machine Learning
Supervised Learning
Develop predictive model based on both input and output data
Unsupervised Learning
Group and interpret data based only on input data
Reinforcement Learning
Training algorithm by using a system of reward and punishment
Classification
Regression
Clustering

Figure 2.1: Classification of machine learning techniques (Horvat and Job 2020, p. 2)

Reed and O'Donoghue (2005) were concerned with predicting the results of three Premier League clubs and two rugby clubs for the 2005/2006 season. The method used was an artificial neural network (ANN), which consists of nodes and connections between them. The neural network must be trained with data to be used as a model for prediction and is therefore classified as supervised learning. The following variables were used for prediction: match venue, rest, the team's position on the league table, the opposing team's position on the league table, distance travelled to the match (for both teams), and form. With the application of the ANN, an accuracy of 57.9% was achieved, which beat the benchmark of the expert tips. This was the first time that artificial intelligence surpassed humans in predicting football matches (Reed and O'Donoghue 2005).

A different approach was taken by Shin and Gasparyan, who wanted to take advantage of the work done by the video game industry by using the player ratings of the game FIFA by *EA Sports* for prediction. The idea of this study is to compare whether better prediction accuracy can be achieved with classical match statistics or the data from FIFA. For the prediction, different models of supervised learning are used to create a *real predictor* from the match statistics and a *virtual predictor* from the FIFA data. The *virtual predictor* was created with 33 features that describe the strength of the players. Each feature describes the player's physical or technical skill and has a value in the range between 0 and 100. The astonishing result of the work is that the *virtual predictor* delivers better results than the *real predictor*, which clearly shows that the data from the video game are indeed relevant for the prediction of football results (Shin and Gasparyan 2014).

Prasetio et al. (2016) used logistic regression to predict EPL football results in the 2015/2016 season based on training data from the previous six seasons. Logistic regression is a classification method that estimates the value of a target variable as a function of one or more independent variables. The difference to the linear model is that the target variable is binary, which means it can only take on two possible values. In the case of football result predictions, this means that one of the three possible outcomes can be disregarded, in this case, the draw. As variables home offense, away offense, home defense and away defense were used, whereby the variables are calculated from the values of the video game FIFA and historical match statistics of the teams. However, it is not described how exactly the variables are calculated. The best model achieved an accuracy of 69.5% over one season, with no draws predicted as a limitation (Prasetio et al. 2016).

Baboota and Kaur(2019) created a generalized model to predict the results of two EPL seasons by using feature engineering to engineer and extract meaningful features and apply machine learning algorithms for the prediction. As the home advantage can be important in football, the home/away factor is treated as a global characteristic and for each developed feature a value for the home and away team is calculated.

The first feature to be used is the teams' strength of the computer game FIFA. The values for attack, midfield, defense, and overall strength of a team are used. The next feature is the goal difference at the current time during the season. To reflect the performance of the teams, the number of corners, shots on goal and goals are considered.

In order to determine the current strength of a team, streak and form are introduced as additional features. The streak describes how well the team has performed in the last games. Depending on the outcome of the match, 0, 1 or 3 points are distributed, based on whether the match ended in a loss, a draw or a win. The games are weighted less the longer they are in the past, so the most recent game has the greatest impact on the form.

This does not take into account how strong the opponents of a team have been recently. To include this information as well, the form is introduced. All teams are assigned the same starting value and points are added for a win while points are deducted for a loss. How many points the value changes depends on the strength of the opponent. For example, if a strong team wins against a weak team, only a few points are added, if a weak team draws against a strong team, points are added to the weak team and points are deducted from the strong team. This logic is also known as Elo score and is used among other things in chess to determine the strength of a player and build the world ranking.

Then, the features are divided into two classes, where Class A contains the individual features for the home and away teams and Class B represents the difference values of the competing teams.

After all features were engineered, four models were tested for prediction: Gaussian naive Bayes, Support vector machine, Random Forest and Gradient Boosting. For the first two algorithms, a feature selection is performed to narrow down the features to the relevant ones. It is noticeable that the features of Class B have a higher significance. The other two algorithms do not require any further feature selection, since they filter out the most relevant features during runtime. The two best performing algorithms are the Gradient Boost and the Random Forest algorithm. For the Random Forest algorithm, a graph of feature importance was created, showing that form is the most important feature. Among the ten most important features, there are only two that were not developed by statistics and results (Baboota and Kaur 2019).

To measure the accuracy of the models, the Ranked Probability Score (RPS) is used, which measures how well the prediction, expressed as a probability distribution, matches the actual outcomes (Constantinou and Fenton 2012). As a benchmark, the odds of betting providers are used, where the profit margin of the providers is calculated to obtain fair odds. The result shows that the betting providers achieve a better RPS value, but the difference to the best performing model is minor (Baboota and Kaur 2019).

2.3 Result Prediction Based on the Lineup

Since the focus of this paper is to engineer features from individual player statistics and then make predictions based on the lineup, literature that pays attention to the lineup is particularly interesting.

Ahmadalinezhad and Makrehchi (2020) analyzed and predicted the performance of the lineup in basketball for the National Basketball Association (NBA), using machine learning and graph theory to develop a metric called edge-centric multi-view networks (Ahmadalinezhad and Makrehchi 2020). A network is created where lineups and scores are stored. The lineups are represented by nodes and connected by edges, which represent the result between two lineups. Based on this, the prediction is performed using a network analysis method. The analysis was performed for the years 2014 to 2019, always using the data from the previous season to predict the current season. Overall, an accuracy of 58% was achieved using this method (Ahmadalinezhad and Makrehchi 2020).

Chun, Son, and Choo (2021) aimed to improve the accuracy of baseball game prediction by considering the starting line-up and the substitutes. Since there are many substitutions during a baseball game, the substitutes are particularly relevant but have not been considered in previous literature. A Long Short-Term Memory (LSTM) model is used to find hidden patterns in time series. Therefore, historical data of baseball games were used. On the one hand, the pre-game records noting the starting line-up and the substitutes and on the other hand the post-game records noting all substitutions. Using a deep neural network, 720 Korea Baseball Organization (KBO) games in 2019 were analyzed. The model was able to deliver 12% better accuracy than previous existing models with an accuracy of 77% (Chun, Son, and Choo 2021).

Yang (2019) used Fantasy League data to predict EPL results based on the starting lineup with supervised learning. The first twelve game weeks of the 2018/19 season were used as training data to predict the games of the next three match days. Four machine learning algorithms were executed for prediction and later the results were compared: Random Forest, Support vector machine, Naïve bayes, and K-Nearest neighbour.

First, the players are divided into the classes, goalkeeper, defender, midfield and attacker, and based on this, the lineup for each game is created. To evaluate the players, the first feature is the so-called ICT index. This index is composed of influence which evaluates a player's impact on a match, creativity which assesses player performance in terms of producing goal-scoring opportunities for other players and threat which gauges players who are most likely to score goals (PremierLeague 2022).

The ICT Index is supposed to indicate how well a player has performed in a match, but it is not specified how the values are calculated and how they are weighted.

The next feature is the Bonus Point System (BPS), which represents the points that players receive for each match day. The points are calculated from several parameters such as goals, ICT or clean sheets for goalkeepers.

The two other features that will be considered are the market value and the number of times a player is selected (PremierLeague 2022).

In the first experiment, the ICT index is used as a feature and all four algorithms are applied. The Random Forest algorithm delivers the best results and is therefore used for feature selection. The feature selection comes as a first step to the result that the features influence, creativity and threat individually considered, provide better results than the ICT index, where the values are summarized. Furthermore, it turns out that the best result is achieved with the features influence, creativity, threat, BPS, and selection. Subsequently, these features are applied to the Random Forest algorithm and an accuracy of 81.8% is achieved for the described dataset. Home and away wins are predicted correctly with an accuracy of over 97%, while draws are only predicted with an accuracy of 22.2% (Yang 2019).

3 Programming Language, Packages and Data

3.1 Programming and Packages

The entire programming part of this work was carried out using the Python programming language, which is popular for performing data science and machine learning tasks (Python 2022). The advantages of Python are a large number of libraries available and the possibility of using jupyter notebooks. These allow the user to write scripts in individual cells and execute them independently (Frochte 2019).
The most important libraries used for this project are briefly described in the following:

beautifoulsoup:
Beautifoulsoup is used for web scraping. Web scraping is the collection of data from the internet, mostly from websites. The method makes use of the fact that behind every website there is a Hypertext Markup Language (HTML) code that can be accessed. The HTML code of the respective website is accessed automatically, and the required information is extracted. Beautifoulsoup allows to read the HTML code of the website via the link and to navigate through the HTML tree structure (Richardson 2007).

numpy:
Numpy means *"Numerical Python"* and is a library written mostly in C for performing mathematical operations. The advantage of this library is the effective calculation of large arrays and matrices, as the implementation is aimed at extremely large data sets (Harris, Millman, Walt, et al. 2020).

pandas:
Pandas stands for *"Python and data analysis"* and is a software library used for data manipulation and analysis. The library is built on numpy making it very efficient for large amounts of data. Pandas makes it possible to display data in Python similar to a spreadsheet to simplify data processing. For this purpose, the data is stored as a pandas DataFrame. A DataFrame

can be read from different sources, such as comma-separated values (csv) or excel files. After data transformation in pandas, the DataFrame can be exported again to different data formats (Pandas 2022).

scikit-learn:
Scikit-learn is a machine learning library. It contains all kinds of algorithms that can be used to implement both supervised and unsupervised machine learning methods. The library offers a rich set of pre-built code that can be used to easily create machine learning models without having to write the code for all of them from scratch. For this study, the Random Forest Classifier from scikit-learn is used (Pedregosa, Varoquaux, Gramfort, et al. 2011).

3.2 Data Collection

Since the data needed for the work is not available in one data source, data from different sources is used and combined. All data is stored in csv file format and then further processed with the pandas library.
All player statistics and lineups are scraped from the web page fbref.com where detailed statistics are provided on an individual player basis for each match.
Table 3.1 shows available player statistics from the web page. Some statistics are self-explanatory, whereas others need to be examined more closely in order to understand them.
For touches, each ball action counts as one touch. Receiving a pass, dribbling and playing a pass, therefore, counts as one touch.
Shot-creating and goal-creating actions are actions that lead directly to a shot or a goal. These can be passes, dribblings or drawn fouls.
Progressive passes are either passes completed in the opponent's penalty area or passes that bring the ball at least ten meters closer to the opponent's goal and are not played in the defensive 40% of the field.
Progressive carries are carries that take place in the opponent's half and move the ball at least five meters towards the opponent's goal.
Pressure is a statistic that indicates how often a defensive player gets close enough to the opponent who is holding the ball to put pressure on him (fbref.com 2022).

The most complex statistic is Expected Goals (xG), which have become increasingly popular in recent years. xG indicates the probability that a shot will result in a goal based on the characteristics of the shot. There are several models that calculate expected goals in different ways. fbref.com uses the model provided by Opta which takes into account not only the location of the shooter but also the clarity of the shooter's path to the goal, the pressure on the shooter from defensive players and the position of the goalkeeper (statsperform.com 2022). By calculating the probability of a goal, further statistics can be calculated. Every pass that leads to a certain xG value is counted as an Expected Assisted Goals (xAG) for the passer. Of interest to goalkeepers, the post-shot expected goals statistic indicates how likely the goalkeeper is to save a shot (fbref.com 2022).

Name	**Interpretation**	**Name**	**Interpretation**
Player	Player Name	SCA	Shot-Creating Actions
Pos	Position	GCA	Goal-Creating Actions
Min	Minutes Played	Cmp	Passes Completed
Gls	Goals Scored	PAtt	Passes Attempted
Ast	Assists	Cmp%	Pass Completion Percentage
PK	Penalty Kicks Made	PProg	Progressive Passes
PKatt	Penalty Kicks Attempted	DSucc	Successfull Dribbles
Sh	Shots Total	DAtt	Dribbles Attempted
SoT	Shots On Target	CProg	Progressive Passes
CrdY	Yellow Cards	SoTA	Shots On Target
CrdR	Red Cards	Prs	Pressures
Touches	Number of times a player touched the ball	GA	Goals Against
Tkl	Tacklings	Saves	Saves
Int	Interceptions	Save%	Save Percentage
Blocks	Blocks	PSxG	Post-Shot Expected Goals
xG	Expected Goals	#OPA	Defensive Actions Outside of Penalty Area
xAG	Expected Assisted Goals	SoTA	Shots On Target Against

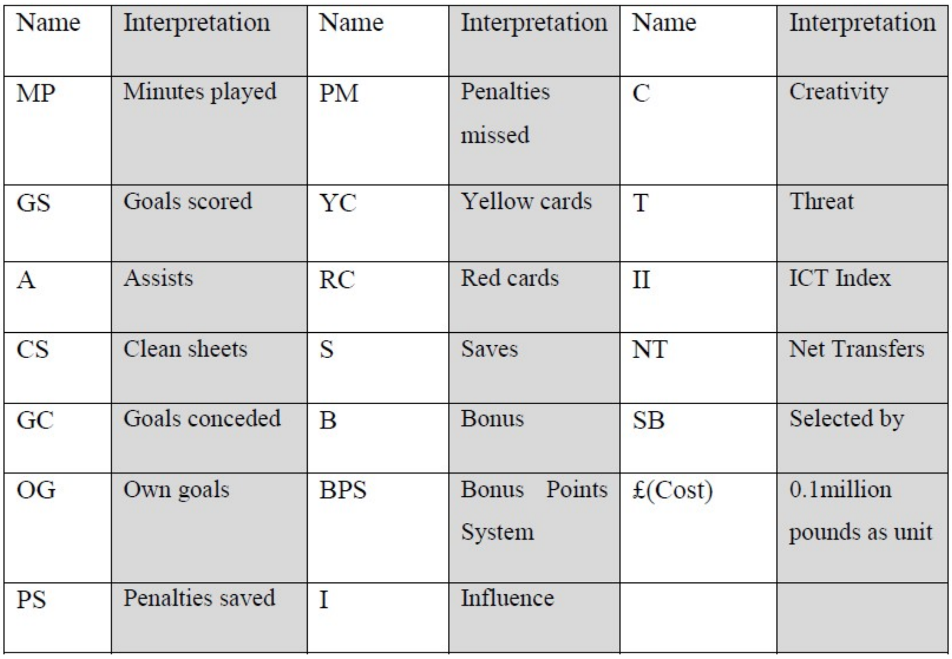
Table 3.1: Player statistics available on fbref.com

The next data source is the Fantasy League values, which are available as a csv file for each season due to Anand's GitHub repository (Anand 2022). For each game, 20 features per player are calculated, which are listed in Figure 3.1.

The match statistics are downloaded from the page football-data.co. There are detailed statistics for each match and also various betting odds, which are later used as a benchmark. Table 3.2 gives an overview of the match statistics.

From the website sofifa.com, the data of the FIFA teams are scraped and saved in a csv file. The website provides many more statistics from the computer game, but for this work only the overall strength of the teams is used.

For the comparison of the prediction models with the community votes, the tips from the users of the website transfermarkt.com are scraped. There, it is indicated in percent how many people have voted for which result.



Name	Interpretation	Name	Interpretation	Name	Interpretation
MP	Minutes played	PM	Penalties missed	C	Creativity
GS	Goals scored	YC	Yellow cards	T	Threat
A	Assists	RC	Red cards	II	ICT Index
CS	Clean sheets	S	Saves	NT	Net Transfers
GC	Goals conceded	B	Bonus	SB	Selected by
OG	Own goals	BPS	Bonus Points System	£(Cost)	0.1million pounds as unit
PS	Penalties saved	I	Influence		

Figure 3.1: All values of the Fantasy League (Yang 2019, p. 31)

Name	**Interpretation**	**Name**	**Interpretation**
Div	League Division	AHW	Away Team Hit Woodwork
Date	Match Date (dd/mm/yy)	HC	Home Team Corners
Time	Time of match kick off	AC	Away Team Corners
HomeTeam	Home Team	HF	Home Team Fouls Committed
AwayTeam	Away Team	AF	Away Team Fouls Committed
FTHG	Full Time Home Team Goals	HFKC	Home Team Free Kicks Conceded
FTAG	Full Time Away Team Goals	AFKC	Away Team Free Kicks Conceded
FTR	Full Time Result	HO	Home Team Offsides
HTHG	Half Time Home Team Goals	AO	Away Team Offsides
HTAG	Half Time Away Team Goals	HY	Home Team Yellow Cards
HTR	Half Time Result	AY	Away Team Yellow Cards
Attendance	Crowd Attendance	HR	Home Team Red Cards
Referee	Match Referee	AR	Away Team Red Cards
HS	Home Team Shots	HBP	Home Team Bookings Points
AS	Away Team Shots	ABP	Away Team Bookings Points
HST	Home Team Shots on Target	B365H	Bet365 home win odds
AST	Away Team Shots on Target	B365D	Bet365 draw odds
HHW	Home Team Hit Woodwork	B365A	Bet365 away win odds

Table 3.2: Match statistics available on football-data.co

3.3 Data Pre-Processing

In general, the quality of the data is very high, as there is no missing data for any of the games considered. Data cleaning in this work is about removing unnecessary information to avoid the tables becoming too large. Therefore, all irrelevant statistics are removed from the pandas DataFrames in the first step.

The larger part is the step of data pre-processing in which the data is put into the right form to be used later for the machine learning algorithms.

First, data that is not already available in csv format is scraped from the internet. This concerns player statistics for each match. With 20 teams in the league, there are 380 games per season involving eleven to 16 players per game. The collected data is stored in a csv file for each season.

Next, the games must be sorted by date, because although a gameweek is given for each matchday, the order of the gameweeks is not always kept due to postponements. Since calculations for the features are based on how teams performed in recent games, the games must be sorted correctly.

Finally, all team and player names from the various sources are adapted to a standard notation to be able to merge the data. This is easy to implement for the teams, as only 24 different teams played in the EPL over three seasons in the considered period. In the case of the players, there are more than 650 players used for each season, some of them with widely differing spellings. For this reason, the player names are automatically matched using a fuzzy matching.

Fuzzy matching is a method that provides an improved ability to process word-based search queries to find matching words that are spelled differently. It attempts to find a match that is above a specified percentage threshold. For the player names, the threshold is set at 90%. This allows about 93% of the player names to be correctly assigned. The remaining player names are added manually (Chaudhuri, Ganjam, Ganti, and Motwani 2003).

In the following, an overview of the most important pandas DataFrames needed to build the prediction models is given:

features:
In this DataFrame all games, their results, and the features that the classic model uses for the prediction are stored.

players:
Holds information on each player for each match. This includes match statistics as well as Fantasy League values with an average of 24130 player ratings per season.

lineups:
All starting line-ups and substitutes are stored in this DataFrame with the information of whether a player is in the starting line-up or not.

features_players:
Includes features about player performance in addition to match statistics. This contains the features from the self-calculated ratings and the values of the Fantasy League.

The DataFrame *features_players* thus contains all data necessary to train the Random Forest model.
Figure 3.2 shows the complete data flow for this study.

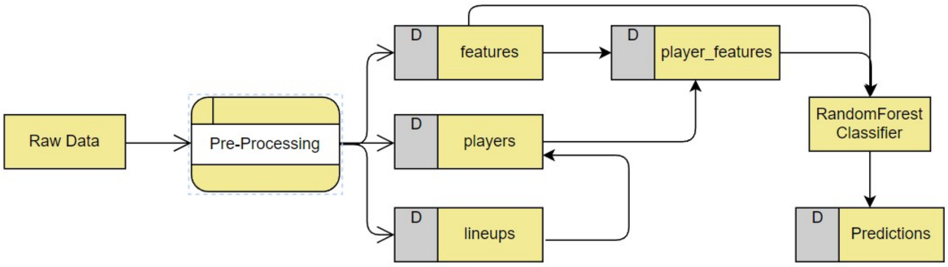


Figure 3.2: Data flow for the project.

4 Methodology

4.1 Player Rating

This chapter describes the procedure for calculating player ratings. The aim is to develop a ranking based on match statistics that reflects the performance of each position group as well as possible. While many providers rate players in the EPL for each match, it is mostly unclear how these ratings are generated. Therefore, in this thesis, a player rating is created, where it is comprehensible how the values are calculated.

4.1.1 Choose Statistics

In the first step, the players are divided into four different groups according to their positions: goalkeeper, defense, midfield and attack, as different statistics are relevant. Table 4.1 lists the statistics for the respective player groups.
As there is no existing literature on this topic, the allocation was made based on a logical understanding of the game of football. For example, for a goalkeeper, goals against and saves are relevant, whereas for a striker, goals and assists are important. Since a good striker is not only characterised by goals or assists, further statistics are added to capture the complete performance. The same applies to the other positions, so each group is evaluated by at least seven statistics.
For the group attack, the xG are subtracted from the actual goals and the xAG are subtracted from the actual assists. Since the xG indicate the probability of a goal being scored from the situation, the difference to the actual goals can be used to determine whether a player is overperforming or underperforming. The same logic is used for the goalkeepers, where the actual goals conceded are subtracted from the post-shot expected goals.

4.1.2 Weight Statistics

Since it is assumed that not all statistics are equally important for evaluating the player groups, the statistics are weighted. Again, there is no existing literature that can be used as a guide. Therefore, a self-developed method is used, which aims to evaluate the players as

Goalkeeper	**Defense**	**Midfield**	**Attack**
Saves	Tackles	Assists	Goals
Goals against	Progressive carries	Goals	Assists
Post-shot expected goals	Progressive passes	Progressive passes	Goals − xG
Save %	Passes %	Progressive carries	Assists − xAG
Passes	Passes	GCA	GCA
Passes %	Assists	SCA	SCA
Actions outside the penalty area	Goals	Dribbles	Shots on target
Post-shot expected goals - Goals	Interceptions	Pressures	
	Blocks		
	Pressures		

Table 4.1: Selected statistics for the different position groups

objectively as possible.

For this purpose, websites that rate players are used as a reference. On the websites, kickset.it and WhoScored.com, each player is rated for all matches and a top ranking per position is published at the end of the season. For each of the four position groups, the ten best-rated players are used as a reference. The average of the two rankings is calculated and used to create the reference ranking. The aim is to adjust the weighting of the statistics to match the reference ranking with the own ranking as closely as possible.

To do so, an attempt is made to minimise the deviation of the top ten players per group with regard to their rank in the ranking over a season between the reference ranking and the own ranking. Since this method is quite complex, only the 2020/21 season is used to find the optimal weights of the statistics, but the weights are applied to all seasons.

For the given problem, a mathematical description is formulated, called a mathematical model, to represent the situation. The model consists of the following components: decision variables, objective function and constraints.

The decision variables are the weights with which the statistics are multiplied. These can be modified in the model to optimise the objective function, in this case, the minimum difference between the ratings.

As a constraint, each statistic is given a weighting value between one and ten. The values are chosen in such a way that each statistic is guaranteed to have an impact on the rating.

The following formula calculates an overall rating for the players:

$$R = s_1x_1 + s_2x_2 + ... + s_nx_n \tag{4.1}$$

R : overall rating

s : statistic

x : weight with $1 \geq x \geq 10$

Based on the overall rating, the players are ranked in ascending order, meaning the player with the highest value is ranked first and the player with the lowest value is ranked lowest. This ranking is compared to the reference ranking with the aim of minimising the deviation. The rank is compared player by player and the amount of the difference is counted as a deviation:

$$Min :\mid oR - rR \mid \tag{4.2}$$

oR : own ranking

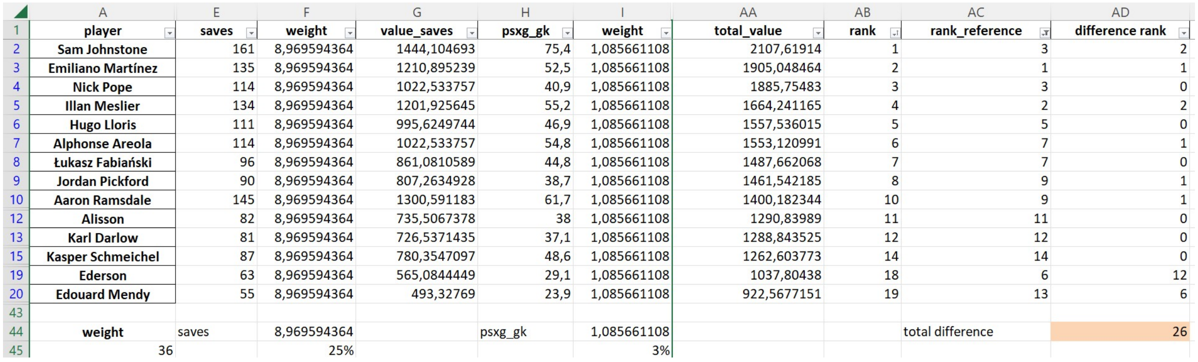
rR : reference ranking

Implementing the optimisation problem in Excel yields the following spreadsheet: Figure 4.1.

Two statistics are shown in columns E and H in this extract, the saves and the post-shot expected goals. To their right there are the weights in columns F and I, followed by the weighted evaluation in columns G and J. Column AA shows the sum of all weighted statistics with the corresponding rank in column AB. The rank for the player from the reference ranking is retrieved in column AC by S-reference from another sheet and the difference of the ranks is calculated in column AD.

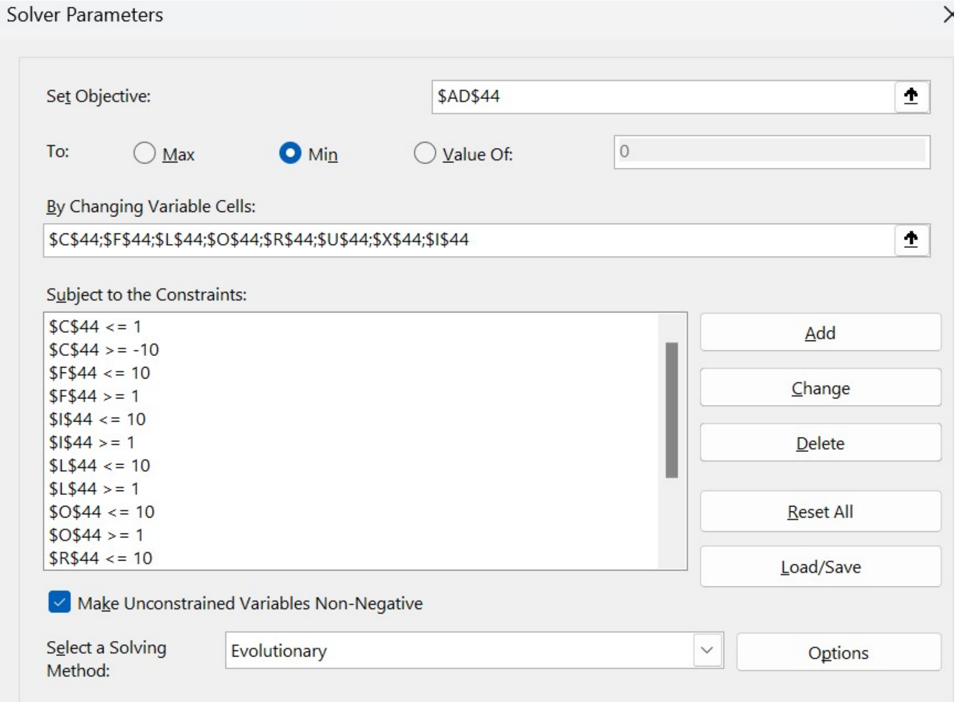
The orange marked cell $AD44$ indicates the total difference of the rankings and is the value to be minimised.

Figure 4.2 shows the solver parameters with the constraints for the weights. Since the formulated problem is not continuous and therefore no algorithm is known that finds the global optimum in an acceptable time, the Evolutionary Algorithm (EA) is used as the solution method. This involves artificially evolving solution candidates for a specific problem based on nature, i.e. EAs are nature-analogue optimisation methods.



	A	E	F	G	H	I	AA	AB	AC	AD
1	player	saves	weight	value_saves	psxg_gk	weight	total_value	rank	rank_reference	difference rank
2	Sam Johnstone	161	8,969594364	1444,104693	75,4	1,085661108	2107,61914	1	3	2
3	Emiliano Martínez	135	8,969594364	1210,895239	52,5	1,085661108	1905,048464	2	1	1
4	Nick Pope	114	8,969594364	1022,533757	40,9	1,085661108	1885,75483	3	3	0
5	Illan Meslier	134	8,969594364	1201,925645	55,2	1,085661108	1664,241165	4	2	2
6	Hugo Lloris	111	8,969594364	995,6249744	46,9	1,085661108	1557,536015	5	5	0
7	Alphonse Areola	114	8,969594364	1022,533757	54,8	1,085661108	1553,120991	6	7	1
8	Łukasz Fabiański	96	8,969594364	861,0810589	44,8	1,085661108	1487,662068	7	7	0
9	Jordan Pickford	90	8,969594364	807,2634928	38,7	1,085661108	1461,542185	8	9	1
10	Aaron Ramsdale	145	8,969594364	1300,591183	61,7	1,085661108	1400,182344	10	9	1
12	Alisson	82	8,969594364	735,5067378	38	1,085661108	1290,83989	11	11	0
13	Karl Darlow	81	8,969594364	726,5371435	37,1	1,085661108	1288,843525	12	12	0
15	Kasper Schmeichel	87	8,969594364	780,3547097	48,6	1,085661108	1262,603773	14	14	0
19	Ederson	63	8,969594364	565,0844449	29,1	1,085661108	1037,80438	18	6	12
20	Edouard Mendy	55	8,969594364	493,32769	23,9	1,085661108	922,5677151	19	13	6
43										
44	weight	saves	8,969594364		psxg_gk	1,085661108			total difference	26
45	36		25%			3%				

Figure 4.1: Extract from Excel Solver to weight the statistics for the player rating



Solver Parameters

Set Objective: AD44

To: ○ Max ● Min ○ Value Of: 0

By Changing Variable Cells:

C44;F44;L44;O44;R44;U44;X44;I44

Subject to the Constraints:

C44 <= 1
C44 >= -10
F44 <= 10
F44 >= 1
I44 <= 10
I44 >= 1
L44 <= 10
L44 >= 1
O44 <= 10
O44 >= 1
R44 <= 10

☑ Make Unconstrained Variables Non-Negative

Select a Solving Method: Evolutionary

Figure 4.2: Excel Solver parameters

After the algorithm has been applied to all player groups in separate Excel sheets, the weights are converted to percent values to illustrate how heavily the statistics are weighted.

4.1.3 Scale Statistics

To make the statistics comparable between the different positions, all ratings are scaled on values between 0 and 100. Where 0 is the worst value and 100 is the best value a player can achieve for a game. The scaling is done with the following normalisation formula, where the old range is from the minimum value to the maximum value per position and the new range is from 0 to 100:

$$\text{NewValue} = \frac{(\text{OldValue} - \text{OldMin}) \cdot (\text{NewMax} - \text{NewMin})}{\text{OldMax} - \text{OldMin}} + \text{NewMin} \tag{4.3}$$

All players are rated with this rating, depending on which position they play. This rating is later used as a feature for the prediction model.

4.2 Prediction Model

This section explains the structure of the prediction models. First, a model with promising features from the existing literature is built using only team statistics. In the following, this model will be referred to as the *"classic model"*. In the next step, the features for the individual players are added to the *classic model*. This model is called the *"improved model"*. After both models have been initialised, the final comparison is which model makes better predictions.

4.2.1 Random Forest Algorithm

This project aims to predict football matches using supervised learning methods. Since various algorithms can be used in this area, it must first be decided which algorithm should be applied. The focus of this work is not on testing different algorithms to figure out which one performs best for this use case, but on the question of whether adding the player data has a positive effect on the prediction accuracy. Therefore, based on existing literature as described in chapter 2, the Random Forest algorithm is used, as it achieved good results in other football score prediction projects and offers good opportunities for evaluation through the feature importances.

The concept of the Random Forest model was developed by Breiman (2001) and is built on decision tree models. In the trees, the predictor space is usually partitioned by binary splits that can be used for metric (regression) or ordinal (classification) responses (Breiman 2001). The composition of the trees is created by randomly selecting a set of features for each split. The maximum size of a tree is specified by the user. This process of tree creation is repeated for a certain number of trees to generate the Random Forest. The optimal number of trees can be determined by tuning the hyperparameters, as described later in chapter 4.2.3. Prediction is then carried out by averaging all response values (regression) or counting them (classification), with the majority of votes deciding.

Figure 4.3 illustrates how the Random Forest algorithm works. Since in this paper the match result is to be classified as a home win, draw or away win, this is an ordinal problem and the Random Forest Classifier is used to solve it.

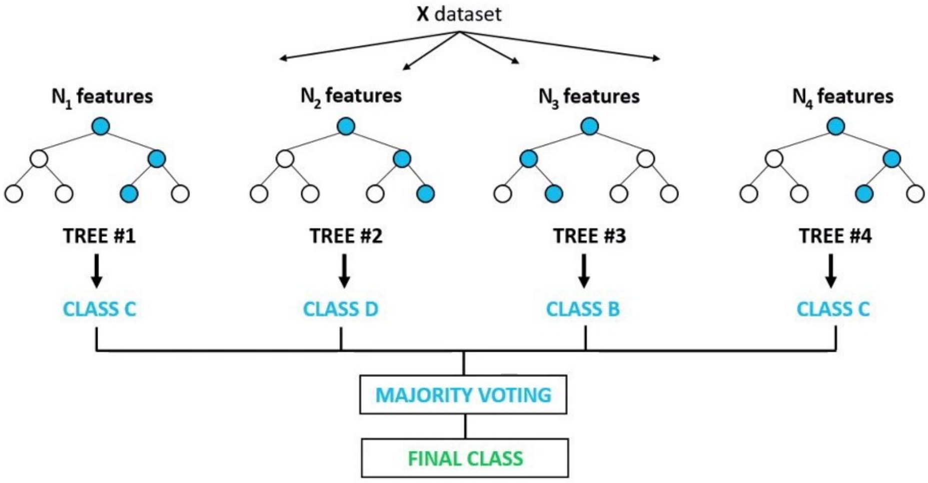


Figure 4.3: Random Forest structure (Chauhan 2021)

In recent years, the Random Forest algorithm has become increasingly popular for machine learning applications because of its good classification abilities, scalability, and simplicity of use (Raschka and Mirjalili 2019).

The advantage that the algorithm offers over others is that it can deal effectively with noise in the data and performs well on weak data. Weak data is data in which the features have only a low correlation in classifications (Ratner, De Sa, Wu, et al. 2016).

This is particularly useful for sports event prediction, as the correlation of the features to the outcome is sometimes low. Another advantage is that the generalisation error converges to a limit value as the number of trees increases, so that the random forest, unlike individual decision trees, does not tend to overfit (Breiman 2001).

4.2.2 Feature Engineering

To train a Random Forest Classifier, features are needed that are well-suited to predict a football result. For the *classic model*, features that have performed well in previous studies are used. Features from the player ratings and the Fantasy League are constructed for the *improved model*.

Classic Model

The first feature is the Elo score, which is a ranking of all teams based on results. Baboota and Kaur (2019) used a very similar feature in their work with the form and found that it works well as described in chapter 2.2.

The implemented Elo rating is based on the rating system developed by Dr. Apad Elo. Each team is assigned an Elo number. The stronger the team, the higher the number. If two teams play against each other, the expected score of the respective teams can be determined from the Elo numbers of the teams. After the match, the Elo ratings of the teams are adjusted to their results.

Depending on the difference between the expected score and the result, a team gains or loses Elo rating points. The system is constructed in such a way that Elo rating points are redistributed among the participating teams. There are different approaches for the calculation of the Elo value. In this work the model of eloratings.net is used, which was developed especially for football games. The special characteristic of this model is that there is also a reallocation of points from the favoured team to the outsider in games that end in a draw and there is an advantage for the home team (eloratings.net 2022).

The Elo score is calculated with the following formula:

$$R_\mathrm{n} = R_\mathrm{o} + K \cdot (W - W_\mathrm{e}) \tag{4.4}$$

R_n : new rating

R_o : old rating

K : weight constant for the tournament, which also contains information about the goal difference by adjusting the value based on it
W : result (1 for a win, 0.5 for a draw, and 0 for a loss)
W_e : win expectancy, which is calculated with the formula:

$$W_\mathrm{e} = \frac{1}{10^{(\frac{-dr}{400})} + 1} \tag{4.5}$$

dr : difference in ratings (adjusted by adding 100 points for a team playing at home)

There is a degree of freedom in this equation, the weighting index K, which must be chosen. With a higher K, the scores converge faster to their true values, but also show more fluctuations. A smaller K gives more stable values, but they take longer to converge. Since in this work the Elo values are reset before each season and thus the period in which the Elo value per team can develop is limited to 38 games per season, the value is chosen relatively high with $K = 30$. Only matches of one league are considered, therefore K is constant for all matches.

The advantage of the Elo value is the simplicity that there is exactly one value for each team at any given time that describes the strength of the team, taking into account past results (Schiefler 2021).

The next feature is the overall team strengths from the computer game FIFA. Shin and Gasparyan (2014) have proven that good results in prediction can be achieved with the values from this computer game. Since the company invests a lot of money every year to represent the players and their strengths as accurately as possible, it makes sense to use these values. For each player and each team, values for defense, midfield and attack are available and updated every week. This work is limited to the teams' overall strength before the start of the season. Especially at the beginning of the season, this characteristic is important, as there are few games available for calculating other features.

The current points scored by a team in the season reflect the team's performance over one year and are used as the next feature. The points are awarded according to the usual calculation in football: win 3 points, draw 1 point and defeat 0 points.

The last feature is the calculation of the team's form. According to Baboota and Kaur (2019), it is suitable to consider a period of six match days. Since luck can play an important role in football, not only the points that a team has scored in the last six games are considered, but also other statistics that express how well a team has played.

Corners are suitable for this because they show that a team has many offensive actions and thus forces the opponent to play the ball out of bounds. Furthermore, shots on the opponent's goal are used, as they also show how many offensive actions a team has in a match. The reason for using the goals scored as the last statistic is logical, as this is what finally determines the result (Baboota and Kaur 2019).

For each feature, there are three values per match, for the home team, away team and the difference between the home and away team. Thus, the *classic model* is created with the features shown in Table 4.2.

Feature
Elo rating
FIFA rating
Points
Form
Corners
Shots
Goals

Table 4.2: Features of the *classic model*

Improved Model

In the next step, the existing model will be extended to include features that are constructed based on the individual player's performance.

For this purpose, the self-calculated player ratings from chapter 4.1 are used. For each match, the starting line-up and the players on the substitute bench are considered. Each position group in the starting line-up represents a separate feature. In addition, all players on the bench are combined into one feature and an overall rating is calculated for the team.

Furthermore, values from the Fantasy League are used, as these values also aim to evaluate players as accurately as possible. As described in chapter 2.3, Yang (2019) came to the conclusion that of all the available Fantasy League statistics, the following give the best results

in predicting the outcome: influence, creativity, threat, BPS and selection. Consequently, these values are also chosen as features for the Random Forest model in this paper (Yang 2019).
The question of how past games are weighted also arises in player evaluation. Exponential smoothing is used to evaluate the form of the players. The exponentially weighted moving average (EWMA) is a statistical measure used to model a time series. More importance is given to more recent data and less to old data. This method produces *"smoothed data"*, which is data that has had the noise removed.
Exponential smoothing is mainly used when the time series does not show a systematic pattern and is volatile. Therefore, the method is used in the financial sector for technical analysis and volatile modelling or in logistics for inventory planning. Since the form of the players also varies and is not linear, the method is suitable for evaluating player performance (Holt 2004).
The EWMA is calculated by the following formula:

$$EWMA_{\mathrm{t}} = \alpha x_{\mathrm{t}} + (1 - \alpha) EWMA_{\mathrm{t\text{-}1}} \tag{4.6}$$

α : smoothing factor with a value between 0 and 1
x_{t} : value of signal x at time t

α is the smoothing parameter, which has to be between 0 and 1. The higher the smoothing factor, the less the old values are taken into account and the more the current values are weighted. The lower α is selected, the more fluctuations are smoothed.
Figure 4.4 shows the influence that alpha has. As an example, the overall rating of all players from the team FC Fulham over one season is taken. For the orange curve, α is set to 0.9, and for the blue curve to 0.1. This clearly shows that the lower α flattens the curve significantly. With a high α, the last games are more heavily weighted, and the form fluctuations are thus stronger.
To find the optimal value for α a feature selection is performed. A total of nine features are created using player statistics only. Since the search for the optimal α with a feature selection for all features becomes too complex, the overall rating is used as a proxy. For this, the overall rating is calculated with $\alpha = 0.1$, $\alpha = 0.2$, ..., $\alpha = 0.9$. Then, the value for each α is used as a separate feature and added to the *classic model*. Now, of the nine added features, the one that

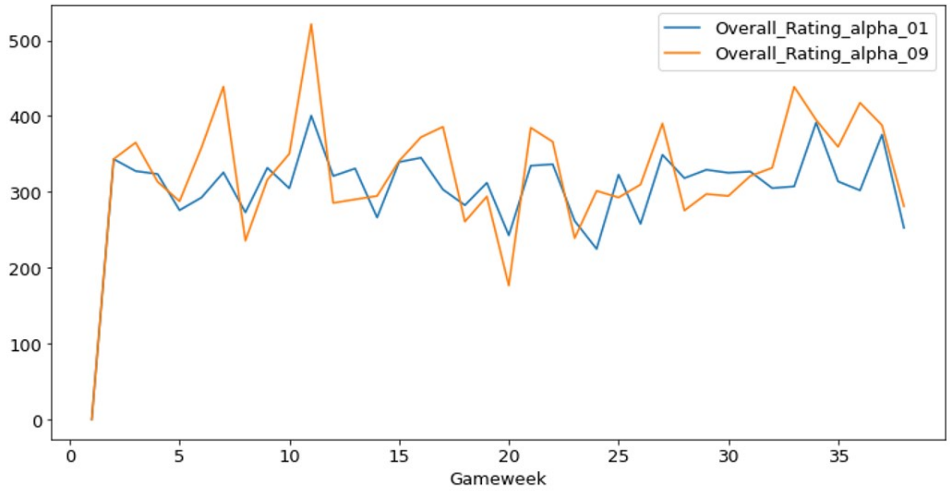


Figure 4.4: Total rating of all FC Fulham players for the 2020/21 season with different values for α

has the best influence on the model is searched for.

For this purpose, the Random Forest model offers the possibility to calculate the feature importance, which indicates how important a feature is for the model. Feature importance is indicated by the Mean Decrease in Impurity (MDI), which counts how often a feature is used to split a node.

Feature selection methods calculate this value and use it to eliminate features. As the model for the feature selection still contains all the features of the *classic model*, but these do not affect the feature selection, the feature with the α that achieves the highest feature importance is searched for.

Figure 4.5 shows all features by descending MDI. The most important feature is at the top. Of the features added for the choice of alpha, marked in the red boxes, the one with $\alpha = 0.9$ is the most important. The result shows that a high α is a better decision-making aid for classifying the result. Due to the high α, games that have been played recently are heavily weighted, suggesting that the form of players and teams can change significantly within a short period of time.

The result of the feature selection is that the smoothing factor α for exponential smoothing is set to 0.9 for player evaluation. This value is applied to all eleven statistics for players listed in table 4.3. Three values are also calculated for each feature, as in the *classic model* for the home team, the away team and the difference.

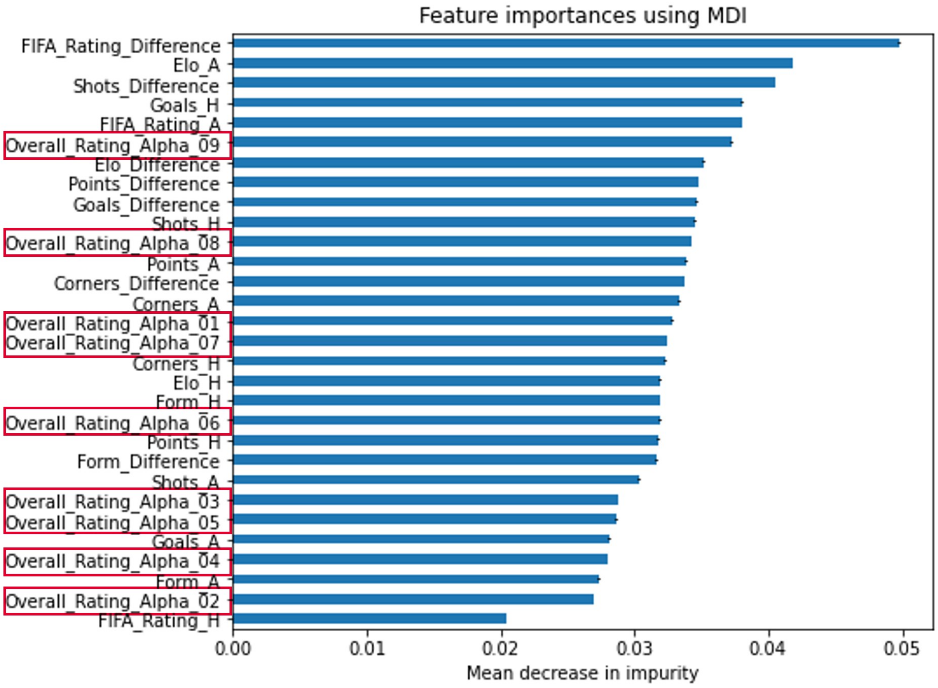


Figure 4.5: Feature selection determining an optimal value for α

Feature Own Rating	**Feature Fantasy League**
Rating Goalkeeper	Influence
Rating Defense	Creativity
Rating Midfield	Threat
Rating Attack	Points
Rating Bench	Number Selected
Rating Overall	

Table 4.3: Added features based on player statistics

4.2.3 Hyperparameter Tuning

The last step before finalising the Random Forest Classifier is hyperparameter tuning, which is an important aspect of machine learning. The structure of all machine learning models is specified by hyperparameters, which might have a significant impact on the accuracy and generalisation of the model. In the Random Forest model, many hyperparameters are set to a default value but can be manually adjusted, for example, the number of trees or the maximum depth of a single tree (Probst, M. N. Wright, and Boulesteix 2019).

In their work, Mantovani, Horváth, Cerri, et al. (2016) highlighted three aspects that explain why hyperparameter tuning is such a complex process. The first is that hyperparameter settings, that work well for one data set, are not guaranteed to give good results for another. The second point is that hyperparameters are often interdependent and therefore cannot be tuned individually. The last point is that tuning hyperparameters is very time-consuming and computationally intensive, especially for high-dimensional datasets (Mantovani, Horváth, Cerri, et al. 2016).

To make the model generalisable and not just tailored to a specific training data set, overfitting must be prevented. Overfitting means that the training data is modelled too accurately and thus negatively affects the performance of the model for new data. To avoid overfitting the Random Forest Classifier through hyperparameter tuning, the concept of Cross Validation (CV) is applied. The most common method, which is also used by sklearn, is the K-fold CV. The training set is split into K number of subsets, so-called folds. Then, K experiments are created for the data set, changing the validation fold in each experiment.

Figure 4.6 illustrates the concept using an example with $K = 5$. Finally, all experiments are iterated over, and the average performance of all folds is returned as the final validation metrics for the model. Searching for the optimal hyperparameters is time-consuming and computationally expensive, therefore some parameters are left at the default value and do not get adjusted. Based on the article by Koehrsen (2018), the following parameters are selected for improvement (Pedregosa, Varoquaux, Gramfort, et al. 2011):

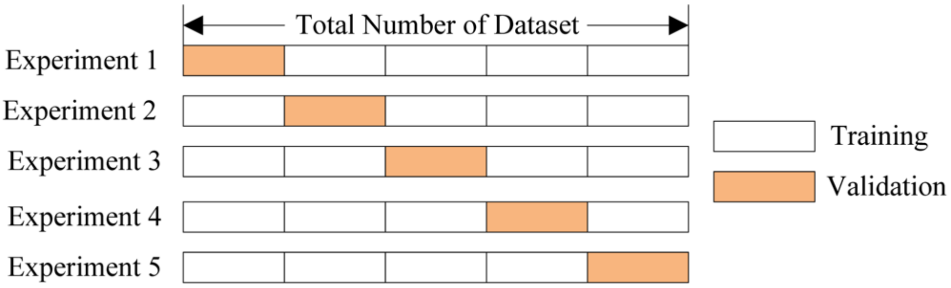


Figure 4.6: K-Fold Cross Validation (Koehrsen 2018)

n_estimators:
The number of trees in the forest.
min_sample_split:
The minimum number of samples required to split an internal node.
min_sample_leaf:
The minimum number of samples required to be at a leaf node.
max_features:
The number of features to consider when looking for the best split.
max_depth:
The maximum depth of the tree.
bootstrap:
Whether bootstrap samples are used when building trees. If false, the whole dataset is used to build each tree (Pedregosa, Varoquaux, Gramfort, et al. 2011).

Once the hyperparameters have been selected, the next step is to decide, which method should be used to optimise them. For this work, the methods of Random Search and Grid Search are combined to efficiently achieve a good result.

First, a Random Search with Cross Validation is performed to narrow down the values of the hyperparameters. For this purpose, threshold values are specified for each parameter, and random combinations are tested within these threshold values. The advantage of this method is that not all combinations are tested and thus a wider range of values can be tested (Bergstra and Bengio 2012).

In the second step, the Grid Search with Cross Validation, a fine grid is created to find the final values for the hyperparameters. In Grid Search, all possible combinations in the specified space for the hyperparameters are tested. Since this is very time-consuming, the range is selected significantly smaller than before with the Random Search. The optimal result of the Random Search serves as the starting value for the grid (Syarif, Prugel-Bennett, and Wills 2016).

Hyperparameter tuning is applied to the *classic model* and the *improved model*. The result after Random Search and Grid Search is shown in Table 4.4. It is noticeable that the optimal values for the hyperparameters are very similar for both models. For both models, the accuracy was improved by more than 2% through tuning the hyperparameters, as shown later in Table 5.3. Subsequently, both models are initialised with the optimal hyperparameters.

Hyperparameter	**Classic Model**	**Improved Model**
n_estimators	300	100
min_sample_split	5	10
min_sample_leaf	4	4
max_features	auto	auto
max_depth	60	60
bootstrap	True	True

Table 4.4: Optimal hyperparameters for Random Forest Classifier

4.2.4 Final Model Definition

After constructing all features and defining all hyperparameters, the Random Forest Classifier can be created using the sklearn library. Since both models consist of 100 and 300 decision trees, respectively, with a maximum depth of 60 levels, Figure 4.7 shows an excerpt from one tree, as the graphic would otherwise become too large.

The nodes indicate, which criterion is used for the decision, how high the gini index is, how many samples are examined and how the distribution of the target classes is. The gini index is a measure that provides information about the distribution of observations in a data set. The smaller the value, the more distinct the distribution. The aim of the decision tree is therefore to reduce the gini value with each split (Tangirala 2020).

The root node splits the data based on the difference in the FIFA ratings. The next level splits the matches based on the difference in goals. The results show that two nodes have been created on the far left and right that have a gini value of zero or close to zero and thus classify the matches clearly. The two remaining nodes must be split further to classify the result better.

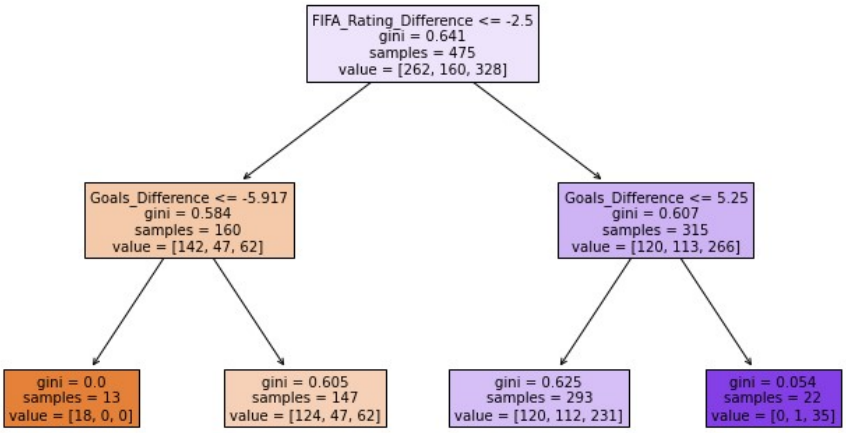


Figure 4.7: Section of a decision tree of the Random Forest Model

5 Results and Interpretation

After describing how the models were created in chapter 4, this chapter shows the results of this work. First, the results of the player ratings are shown and then both models are compared and ranked against the benchmarks.

5.1 Results of the Player Rating

The first result is the weighting of the statistics for the player ratings. Table 5.1 shows in descending order how the values are weighted in percent.

Unsurprisingly, for the goalkeeper position, the number of balls saved is the most important statistic. This is followed by the pass rate and the goals scored minus the post-shot expected goals. As goalkeepers become increasingly involved in the game, it is more and more important for them to play well with their feet, which is expressed by the pass rate. The difference between the post-shot expected goals and the goals scored indicates whether the goalkeeper has made a better or worse save, as assumed by the probability model for the xG, and is therefore meaningful for the performance of the goalkeeper. In fourth place is the number of goals conceded. This is the most important statistic for a goalkeeper in terms of results, but the value depends strongly on the strength of the entire team. All other statistics are weighted significantly lower.

Among the defenders, it is noticeable that the three most important statistics, with progressive carries, pass rate and assists, are related to the offensive game. After that comes the first statistic, tackles, which evaluates defensive actions. All other defensive statistics are weighted only slightly. The reason for this may be that particularly good teams put the opponent under pressure early on and thus win a lot of balls in the attack and midfield. Therefore, the defenders also play an important role in the offensive game, as they are responsible for the build-up of the game and crosses, for example.

Four offensive statistics also dominate the rating for the midfield: assists, Goal-Creating Actions (GCA), Shot-Creating Actions (SCA) and goals. All these statistics are directly related to creating a goal-scoring opportunity or scoring a goal and thus have a direct influence on the result of a match. The group of midfielders includes positions responsible for the offensive, such as wingers, and positions concerned with defending, such as defensive midfielders, and is therefore difficult to evaluate overall. However, it is also evident here that offensive values have a greater influence on the ranking.

For the forwards, the highest weighted statistics concern goals and assists. This is hardly surprising, as the main focus of a striker is to score goals. The most important statistic is the assists minus the xAG followed by the goals and the goals subtracted from the xG. Surprisingly, the actual assists are weighted the least for this group. There is no logical explanation for this, as the assists were also expected to be higher in the rankings. The reason for this is perhaps that the weighting was only carried out with values over one season.

5.2 Evaluation of the Models

For the *classic* and the *improved model* the two EPL seasons, 2019/20 and 2020/21 were used as training data and the 2021/22 season was used for evaluation. For all seasons, the first three match days are not considered due to a lack of available data because many features are calculated from the previous games, which are not available at the beginning of the season. After that, 350 matches remain for each season, resulting in a training set that contains 700

Goalkeeper	**Defense**	**Midfield**	**Attack**
25% - Saves	23% - Progressive carries	23% - Assists	21% - (Assists − xAG)
22% - Passes %	19% - Passes %	23% - GCA	19% - Goals
22% (Post-shot expected goals - Goals)	19% - Assists	23% - SCA	18% - (Goals − xG)
16% - Goals against	15% - Tackles	19% - Goals	17% - Shots on target
7% - Save %	11% - Goals	4% - Dribbles	13% - GCA
4% - Actions outside the penalty area	4% - Blocks	2% - Progressive passes	8% - SCA
3% - Post-shot expected goals	2% - Progressive passes	2% - Progressive carries	4 % - Assists
1% - Passes	2% - Passes	2% - Pressures	
	2% - Interceptions		
	2% - Pressures		

Table 5.1: Weighted statistics for the position groups

matches and a test set with 350 matches.

The Random Forest model calculates probabilities for the outcome of each game. The result is classified into home win, draw and away win, and the probability for the three target classes are calculated. The model votes for the class with the highest calculated probability.

Moreover, there is the possibility to view the importance of the features of the Random Forest, as described in chapter 4.2.2.

Figure 5.2 shows the ten most important features of the *classic model*. The most important feature is the difference in FIFA ratings, followed by the difference in points and the difference in shots. It is surprising that the FIFA values are so important for the model, as they are constant over a season and do not adapt to the form of the teams. The advantage compared to the other features is especially at the beginning of the season, because at that time there is still little data available from previous games. The form and the Elo value, for example, are calculated from previous games and thus become more meaningful later in the season.

The difference in Elo values follows in fourth place; based on previous literature, this value was expected to be more important. Since the Elo values are reset after each season, the time period for the Elo value to develop is too short. To make this feature even more meaningful, Elo values from the previous season would have to be transferred for the start of the current season. However, this would also require values of the teams promoted from the second league, which would further complicate the model.

Considering all features, it is noticeable that the best four features and six of the best ten features are difference values. This is logical, as the difference value takes into account the strengths of both teams.

Figure 5.2 shows the top 10 features of the *improved model*. Again, the difference in FIFA ratings is the most important feature. The second most important feature is the difference in the number of times a player is selected in the Fantasy League, which is the first feature developed on a player basis. Overall, two more features added to the model based on the player rating are among the top ten. The overall rating difference and the number of times a player was selected for the away team. Since a total of eleven features, each with three values for the home team, away team and the difference were added to the model, the proportion of three features from the top ten is relatively small. The other top features are very similar to those of the *classic model*.

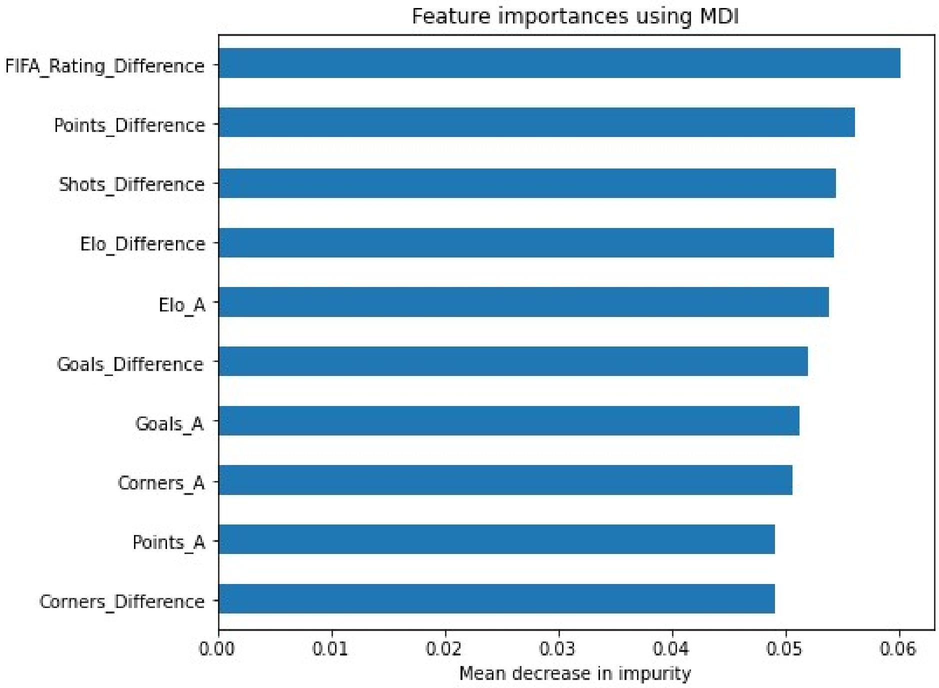


Figure 5.1: Top 10 Features of the *Classic Model*

Two indicators are used to evaluate the models, firstly the percentage of correctly predicted matches and secondly the RPS. The percentage of correctly predicted matches is calculated

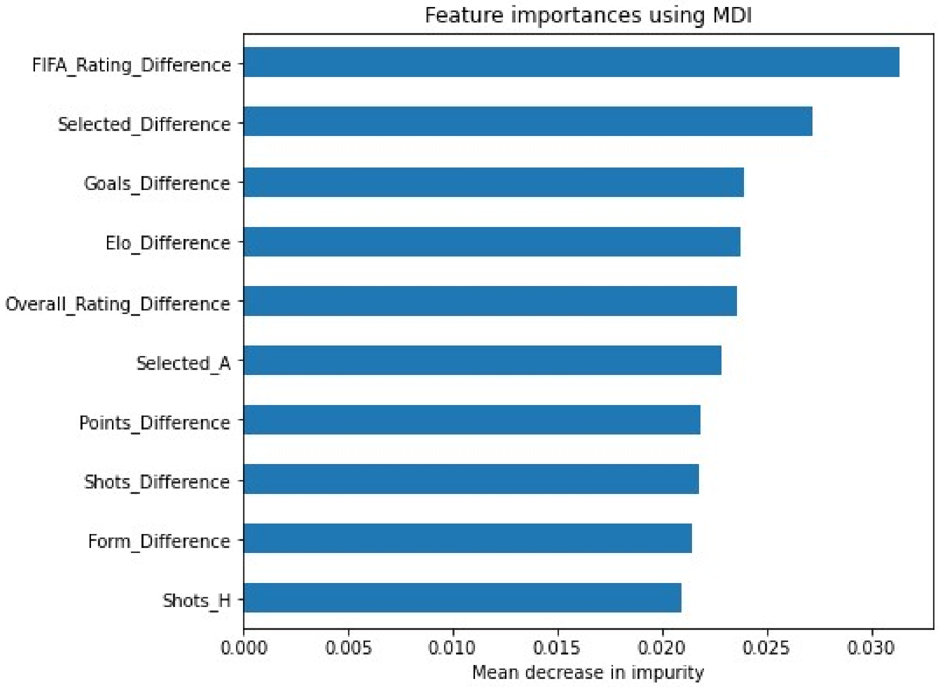


Figure 5.2: Top 10 Features of the *Improved Model*

by dividing the number of correct predictions by the number of all matches:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{5.1}$$

Since this value is easy to understand and classify, but only takes into account the votes at the end, the RPS is also calculated, which takes into account the probabilities for the outcome. The RPS was introduced by Epstein in 1969. The score was then used by Constantinou and Fenton (2012) to evaluate football score prediction models and proved to be the best metric for this purpose. The RPS is based on a so-called scoring rule that assigns a score to each prediction depending on how close the probabilities are to the actual outcome. Therefore, the set of outputs home win, draw and away win is considered as an ordinal scale. A draw is closer to a home win than an away win is to a home win. If the home team leads by one goal, it needs only one goal for a draw but two for an away win. It follows that for a home win the scoring rule should penalise the probability of an away win higher than the probability of a draw. The metric is calculated by the following formula (Constantinou and Fenton 2012):

$$RPS = \frac{1}{r-1} \sum_{i=1}^{r-1} (\sum_{j=1}^{i} (p_{\mathrm{j}} - e_{\mathrm{j}})^2 \tag{5.2}$$

r : number of potential outcomes

p_{j} : forecast at position j

e_{j} : observed outcome at position j

The RPS represents the difference between the cumulative distributions of predictions and observations (Constantinou and Fenton 2012). Since the scoring rule is sensitive to the distance, the penalty score increases the more the predicted cumulative distribution deviates from the actual outcome (Wilks 2011).

The example in Table 5.2 demonstrates how the RPS works. The first three columns give the model's probability of the match outcome, the result column gives the actual outcome, and the last column gives the calculated RPS.

In the first row, the model's probability of a home win is 100% and the result is a home win, so the RPS is at the optimal value of 0.

In row two, the model calculates a 100% probability of an away win, but the result is a home win. The RPS is 1, which is the worst achievable value.

In the third row, a draw is predicted, but the result is again a home win. Since the distance from draw to home win is smaller, the RPS is only 0.5.

Line four shows another probability distributions and the corresponding RPS. It is important to note that the smaller the RPS, the better the prediction of the model.

To obtain the RPS for a season, the mean value of all predictions is calculated.

Home win	**Draw**	**Away win**	**Result**	**RPS**
1.00	0.00	0.00	H	0.00000
0.00	0.00	1.00	H	1.00000
0.00	1.00	0.00	H	0.50000
0.80	0.10	0.10	H	0.02500

Table 5.2: Ranked Probability Score example

5.3 Comparison between the Models

Before both models were finalised, hyperparameter tuning was performed as described in chapter 4.2.3. This increased the accuracy for both models by over 2%. The results of hyperparameter tuning are shown in Table 5.3.

Hyperparameters	**Classic Model**	**Improved Model**
Random Hyperparameters	56.86%	58.85%
Hyperparameters from random search	58.86%	59.43%
Hyperparameters from grid search	59.14%	61.42%

Table 5.3: Accuracies of hyperparameter tuning

The *classic model* correctly classified 207 of the 350 matches to be predicted. This gives an accuracy of 59.14%. The *improved model* was able to correctly classify 215 games, achieving an accuracy of 61.43%.

This means that the accuracy of the model could be improved by 2.29% by adding player statistics. This sounds like a small percentage. However, looking at existing literature as described in chapter 2, it is very difficult to achieve high accuracy in sport event predictions as many factors are unpredictable. Therefore, the improvement in accuracy by more than 2% is a notable result.

Next, the distribution of predictions is examined in more detail. For this purpose, a confusion matrix is generated for each model, which shows the distribution of the predictions and combines it with the actual results.

Figures 5.3 and 5.4 show the confusion matrix of the *classic* and *improved model*. The x-axis is the predicted labels and the y-axis is the actual labels, meaning the rows are the predictions of the model and the columns are the actual results. The main diagonal from top left to bottom right thus gives the correctly predicted matches. This is the case when the actual values and the model predictions match.

First, the results of the *classic model* in Figure 5.3 are examined in more detail. The first line shows that 113 home wins were correctly classified, two home wins were predicted as draws and 34 home wins were predicted as away wins. This gives an accuracy of 76% for home wins. The second row indicates that only 15 out of a total of 80 draws are correctly classified, giving an accuracy of 19%. In the bottom row, 79 out of 121 away games are correctly predicted, which gives an accuracy of 65%.

Next, the results of the *improved model* are considered in Figure 5.4. Here 112 home wins are correctly classified with an accuracy of 74%. The accuracy for the draws is 24% with 19 correct predictions and 84 away wins were correctly predicted with an accuracy of 69%.

In comparison, both models are very similar with regard to the distribution of predictions. There are only marginal deviations whereby the *classic model* correctly predicts more home wins, but the *improved model* delivers better results for draws and away wins.

For both models, it is obvious that the accuracy of the prediction is very low for the draws. In football, a draw can occur when both teams score an equal number of goals, but it can also happen that the game ends goalless. It is difficult to predict draws in football with machine learning algorithms, as a draw can depend on a variety of factors such as the strength of the teams, the strategies of the coaches and the conditions on the pitch. Because there are so many different factors that can influence the outcome of a football match and lead to a draw, it can be hard for a machine learning algorithm to predict exactly when a draw will occur. Furthermore, the result of a draw is the rarest in football and thus it is difficult for the algorithm to find patterns in the data that indicate a draw. In the data set considered for training the classifier, 23% of the matches ended in a draw.
The fact that draws in football are the most difficult to predict and have a negative impact on the accuracy of the models is also described in other papers by Baboota and Kaur (2019) and Yang (2019).

Finally, the RPS values of the two models are used for comparison. Averaged over the season, the *classic model* achieves a value of 0.2156 and the *improved model* a value of 0.2113.

Since a smaller value means a better distribution of the probabilities, this indicator also shows that the model has been improved by adding player statistics.

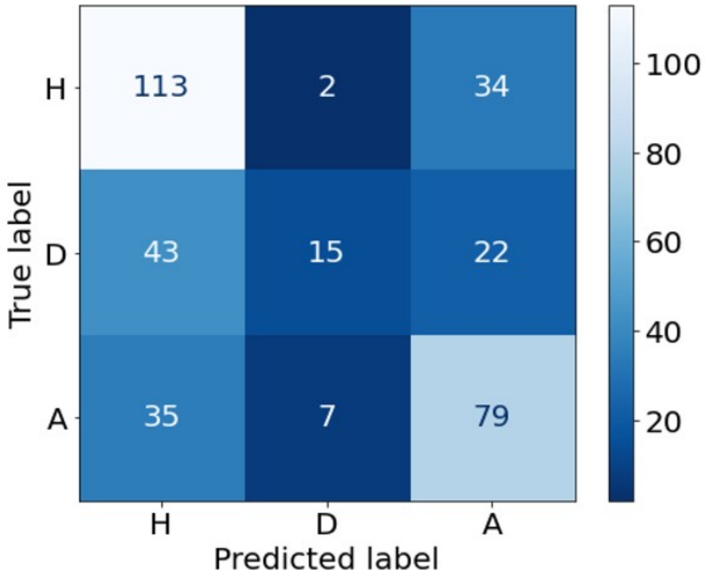


Figure 5.3: Confusion Matrix *Classic Model*

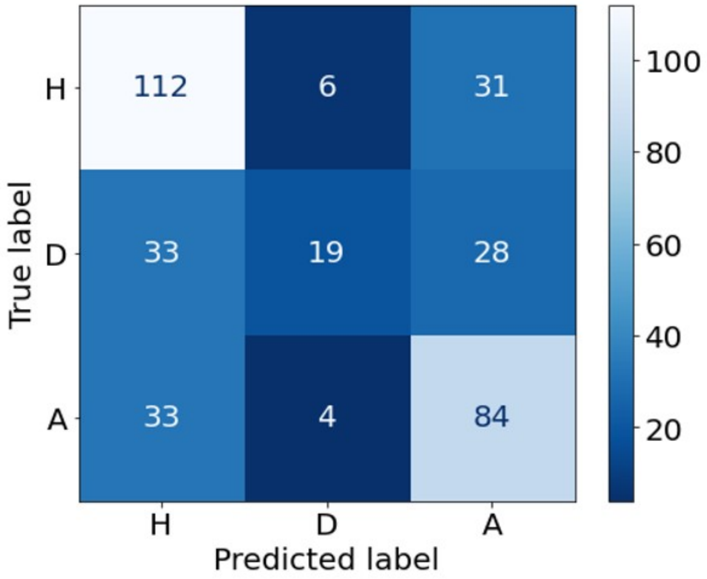


Figure 5.4: Confusion Matrix *Improved Model*

5.4 Comparison to the Benchmarks

In order to be able to classify the performance of the prediction models, they are compared with two benchmarks. Firstly, the predictions of the community from the website transfermarkt.com and secondly, the odds of the bookmakers from Bet365.
The website transfermarkt.com offers users the opportunity to vote before each EPL match on how the match will end. The options are home win, draw and away win. The site shows in percent how many users have voted on which result. This information is read out and saved via web scraping for each match of the 2021/22 season. The data is then compared with that of the models, as these also specify a probability value for each classification.

As can be seen in Table 3.2, the data set of page football-data.co contains not only match statistics but also odds from the betting provider Bet365. For each match, Bet365 offers odds for a home win, a draw and an away win. The betting odds indicate the factor by which the stake is multiplied in the event of a win. By inverting the odds, the bookmakers' probabilities for the results can be calculated:

$$\pi_{\mathrm{i}} = \frac{1}{o_{\mathrm{i}}} \tag{5.3}$$

i : set for home, draw, away
o : bookmaker odds
π : implied probabilities

Since bookmakers want to earn money through their business, they do not offer fair odds but calculate about 10% as a margin for themselves by offering unfair odds. Calculating the probabilities from the unfair bets, the sum of all probabilities is greater than 100%. Therefore, basic normalisation as mentioned by Štrumbelj (2014) is used to solve this problem:

$$p_{\mathrm{i}} = \frac{o_{\mathrm{i}}}{\sum_{i=0}^{3} \pi_{\mathrm{i}}} \tag{5.4}$$

p_{i} : normalized probabilities

The normalised probabilities add up to 100% across all three classes and are thus suitable for comparison with the models (Baboota and Kaur 2019).

In the next step, the RPS value is used as a metric to compare the probabilities of the models with those of the benchmarks. The community votes achieve an RPS value of 0.2204, and the betting providers a value of 0.1913. Thus, the *classic model* (RPS: 0.2156) and the *improved model* (RPS: 0.2113) performed better than the community but worse than the betting providers.

The result shows that the difference between the *improved model* and the betting odds is only 0.02. Considering that the success of the bookmakers depends on calculating the most accurate probabilities for the match outcome and they have a lot of money and resources available for it, it shows that the approach for the model is promising.

6 Summary and Outlook

In this study, it was shown that the inclusion of player data can improve the performance of football result prediction models using machine learning algorithms. Thus, the accuracy of predictions for a season could be increased from 59% to over 61% by adding player data in the EPL.

To use player data, a method was developed to create a rating from various player statistics as described in chapter 4.1. A distinction was made between the position groups goalkeeper, defense, midfield and attack. Different statistics were assigned to each group based on the relevance of the respective position. After the statistics were weighted using reference lists that also evaluate players (see chapter 4.1.2), all ratings were scaled equally to make them comparable between the position groups.

To find out whether the player ratings reflect the performance of the players, two prediction models were set up. One with classic features calculated from team statistics and a second one to which the player ratings are added. The algorithm used was the Random Forest Classifier, described in more detail in chapter 4.2.1.

Promising features from previous literature were selected for the *classic model*. The most important features were the team strengths of the game FIFA, the number of points and shots and the Elo value (see chapter 5.2). As further features, the self-calculated ratings and values of the Fantasy League were added for the *improved model*. Thus, a total of eleven additional features have been added to the model, listed in Figure 4.3.

From the *improved model*, three of the top ten features are among those added based on player statistics (Figure 5.2). This shows that the added features influence the model in a positive way, which is also reflected in the results. The model with the player data has both a higher accuracy and a better RPS value, which is explained in chapter 5.2. This work shows on the one hand that meaningful ratings can be calculated from various statistics for each position group and on the other hand that the consideration of the players improves the prediction model.

Compared to the community votes, both models provide better predictions. However, the models lose the comparison to the bookmakers, who achieve a minimally lower RPS value. Yet, this shows the potential of the model, as the gap to the bookmakers is small and they have significantly more resources available.

One limitation of this work is the features selected for the *classic model*. Features have been selected that have worked well in other papers, but there are many more features that are not used. For example, more match statistics could be used, information about the coaches could be added, or attention could be paid to how long the players have had to recover.

There would also be further opportunities in the engineering of the features for the *improved model*. In this thesis, the differences between home and away teams were calculated for the same position groups. Another possibility would be to calculate the differences between the team parts that meet in the game, e.g. forwards and defense. This would lead to significantly more features, which would make a feature selection necessary.

In the field of machine learning, many algorithms achieve good accuracies. In this work, the Random Forest algorithm was used, but it would be possible to use different algorithms with the data to find out which one gives the best results.

Baboota and Kaur (2019) and Yang (2019) used four different algorithms and then compared which model gave the best results.

Finally, the data set only included three seasons of the EPL, to obtain more meaningful results, more matches would have to be considered. These can be games that are further in the past or games from other leagues.

For further research, additional algorithms can be investigated. Baboota and Kaur (2019) have shown that the XgBoost algorithm gave promising results for a similar use case and besides that, artificial neural networks (Reed and O'Donoghue 2005) are another possibility for the selection of the model.
Regardless of the choice of algorithm, the choice of features is the most important criterion. For the rating of the players, it might be interesting to consider the fitness level, meaning how many games a player has played in the past weeks. In addition to player statistics, it would be useful to add information about the coach and the tactics of the teams to further improve the accuracy of the model.

Bibliography

Ahmadalinezhad, M. and M. Makrehchi (2020). “Basketball lineup performance prediction using edge-centric multi-view network analysis”. In: *Social Network Analysis and Mining* 10.1, pp. 1–11.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Anand, V. (2022). *Fantasy-Premier-League*. `https://github.com/vaastav/Fantasy-Premier-League`. (Visited on 05/05/2022).

Baboota, R. and H. Kaur (2019). “Predictive analysis and modelling football results using machine learning approach for English Premier League”. In: *International Journal of Forecasting* 35.2, pp. 741–755.

Baumer, B. and A. Zimbalist (2014). *The sabermetric revolution: Assessing the growth of analytics in baseball*. University of Pennsylvania Press.

Bergstra, J. and Y. Bengio (2012). “Random search for hyper-parameter optimization.” In: *Journal of machine learning research* 13.2.

Breiman, L. (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.

Buraimo, B., B. Frick, M. Hickfang, and R. Simmons (2015). “The economics of long-term contracts in the footballers’ labour market”. In: *Scottish Journal of Political Economy* 62.1, pp. 8–24.

Carling, C., A. M. Williams, and T. Reilly (2007). *Handbook of soccer match analysis: A systematic approach to improving performance*. Routledge.

Chaudhuri, S., K. Ganjam, V. Ganti, and R. Motwani (2003). “Robust and efficient fuzzy match for online data cleaning”. In: *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 313–324.

Chauhan, A. (2021). *Random Forest Classifier and its Hyperparameters*. https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6. (Visited on 11/14/2022).

Chun, S., C.-H. Son, and H. Choo (2021). "Inter-dependent LSTM: Baseball Game Prediction with Starting and Finishing Lineups". In: *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE, pp. 1–4.

Constantinou, A. C. and N. E. Fenton (2012). "Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models". In: *Journal of Quantitative Analysis in Sports* 8.1.

Deutscher, C., E. Dimant, and B. R. Humphreys (2017). "Match fixing and sports betting in football: Empirical evidence from the German Bundesliga". In: *Available at SSRN 2910662*.

Dvorak, J., A. Junge, J. Chomiak, T. Graf-Baumann, L. Peterson, D. Rosch, and R. Hodgson (2016). "Risk factor analysis for injuries in football players". In: *The American journal of sports medicine*.

eloratings.net (2022). http://eloratings.net/. (Visited on 11/08/2022).

Epstein, E. S. (1969). "A scoring system for probability forecasts of ranked categories". In: *Journal of Applied Meteorology (1962-1982)* 8.6, pp. 985–987.

fbref.com (2022). *2021-2022 Premier League Scores Fixtures*. https://fbref.com/en/comps/9/schedule/Premier-League-Scores-and-Fixtures. (Visited on 05/15/2022).

football-data.co (2022). *Historical Football Results and Betting Odds Data*. https://www.football-data.co.uk/data.php. (Visited on 05/20/2022).

Fried, G., C. Mumcu, et al. (2016). "Sport analytics". In: *a data driven approach to sport business and management*, p. 2017.

Frochte, J. (2019). *Maschinelles Lernen: Grundlagen und Algorithmen in Python*. Carl Hanser Verlag GmbH Co KG.

Graefe, L. (Feb. 2022). *Sportwettenmarkt_Bis_2020*. URL: https://de.statista.com/statistik/daten/studie/1211874/umfrage/umsatzentwicklung-auf-dem-markt-fuer-sportwetten-in-deutschland/#:~:text=In%5C%20den%5C%20Jahren%5C%202014%5C%20bis,Corona%5C%2DPandemie%5C%20jedoch%5C%20einen%5C%20Einbruch (visited on 05/01/2022).

Groll, A., L. M. Hvattum, C. Ley, F. Popp, G. Schauberger, H. Van Eetvelde, and A. Zeileis (2021). "Hybrid Machine Learning Forecasts for the UEFA EURO 2020". In: *arXiv preprint arXiv:2106.05799*.

Hanley, B. (2016). "Pacing, packing and sex-based differences in Olympic and IAAF World Championship marathons". In: *Journal of sports sciences* 34.17, pp. 1675–1681.

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: `10.1038/s41586-020-2649-2`. URL: `https://doi.org/10.1038/s41586-020-2649-2`.

Hastie, T., R. Tibshirani, and J. Friedman (2009). "Unsupervised learning". In: *The elements of statistical learning*. Springer, pp. 485–585.

Holt, C. C. (2004). "Forecasting seasonals and trends by exponentially weighted moving averages". In: *International journal of forecasting* 20.1, pp. 5–10.

Horvat, T. and J. Job (2020). "The use of machine learning in sport outcome prediction: A review". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10.5, e1380.

kickset.it (2022). `https://www.kickest.it/en/premier-league/stats/players/table?positions=1`. (Visited on 11/11/2022).

Kim, N. J. and J. K. Park (2015). "Sports analytics & risk monitoring based on hana platform: sports related big data & IoT trends by using HANA in-memory platform". In: *2015 International SoC Design Conference (ISOCC)*. IEEE, pp. 221–222.

Koehrsen, W. (2018). *Random Forest Classifier and its Hyperparameters*. `https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74`. (Visited on 11/14/2022).

Kotsiantis, S. B., I. Zaharakis, P. Pintelas, et al. (2007). "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160.1, pp. 3–24.

Lewis, M. (2004). *Moneyball: The art of winning an unfair game*. WW Norton & Company.

Lewis, S. (Sept. 2019). *What is operations research and why is it important?* URL: `https://www.techtarget.com/whatis/definition/operations-research-OR`.

Link, D. (2018). "Sports analytics". In: *German Journal of Exercise and Sport Research* 48.1, pp. 13–25.

Mantovani, R. G., T. Horváth, R. Cerri, J. Vanschoren, and A. C. de Carvalho (2016). "Hyperparameter tuning of a decision tree induction algorithm". In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, pp. 37–42.

Michelle, N. (2021). *Machine Learning 101:Introduction*. `https://nicolemichelledsouza.medium.com/machine-learning-101-97960e68f4c7`. (Visited on 11/18/2022).

Mordorintelligence (2021). *Sports analytics market: 2022 - 27: Industry share, size, growth - mordor intelligence*. URL: `https://www.mordorintelligence.com/industry-reports/sports-analytics-market` (visited on 04/18/2022).

Mottley, C. M. (1954). "The application of operations-research methods to athletic games". In: *Journal of the Operations Research Society of America* 2.3, pp. 335–338.

Ohlhorst, F. J. (2012). *Big data analytics: turning big data into big money*. Vol. 65. John Wiley & Sons.

Panchanathan, S., S. Chakraborty, T. McDaniel, M. Bunch, N. O'Connor, S. Little, K. McGuinness, and M. Marsden (2016). "Smart stadium for smarter living: Enriching the fan experience". In: *2016 IEEE international symposium on multimedia (ISM)*. IEEE, pp. 152–157.

Pandas (Oct. 2022). *pandas-dev/pandas: Pandas*. Version v1.5.1. If you use this software, please cite it as below. DOI: `10.5281/zenodo.7223478`. URL: `https://doi.org/10.5281/zenodo.7223478`.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Perl, J. (2002). "Game analysis and control by means of continuously learning networks". In: *International Journal of Performance Analysis in Sport* 2.1, pp. 21–35.

Prasetio, D. et al. (2016). "Predicting football match results with logistic regression". In: *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*. IEEE, pp. 1–5.

PremierLeague (2022). *Fantasy League Statistics*. `https://fantasy.premierleague.com/statistics`. (Visited on 11/18/2022).

Probst, P., M. N. Wright, and A.-L. Boulesteix (2019). “Hyperparameters and tuning strategies for random forest”. In: *Wiley Interdisciplinary Reviews: data mining and knowledge discovery* 9.3, e1301.

Purucker, M. C. (1996). “Neural network quarterbacking”. In: *IEEE Potentials* 15.3, pp. 9–15.

Python (2022). *Python*. `https://www.python.org/`. (Visited on 11/01/2022).

Raschka, S. and V. Mirjalili (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd.

Ratner, A. J., C. M. De Sa, S. Wu, D. Selsam, and C. Ré (2016). “Data programming: Creating large training sets, quickly”. In: *Advances in neural information processing systems* 29.

Rebala, G., A. Ravi, and S. Churiwala (2019). “Machine learning definition and basics”. In: *An Introduction to Machine Learning*. Springer, pp. 1–17.

Reed, D. and P. O’Donoghue (2005). “Development and application of computer-based prediction methods”. In: *International Journal of Performance Analysis in Sport* 5.3, pp. 12–28.

Richardson, L. (2007). “Beautiful soup documentation”. In: *April*.

Ricky, A. (Dec. 2021). *Council post: How data analysis in sports is changing the game*. URL: `https://www.forbes.com/sites/forbestechcouncil/2019/01/31/how-data-analysis-in-sports-is-changing-the-game/?sh=36ebb2b33f7b`.

Schiefler, L. (2021). *Football Club Elo Ratings*. `http://clubelo.com/`. (Visited on 11/15/2022).

Shin, J. and R. Gasparyan (2014). “A novel way to soccer match prediction”. In: *Stanford University: Department of Computer Science*.

sofifa.com (2022). *sofifa*. `https://sofifa.com/teams?r=210064&set=true`.

statsperform.com (2022). `https://www.statsperform.com/`. (Visited on 05/15/2022).

Štrumbelj, E. (2014). “On determining probability forecasts from betting odds”. In: *International journal of forecasting* 30.4, pp. 934–943.

Sullivan, B. (July 2014). *Microsoft’s Bing wins the World Cup with predictive analytics*. URL: `https://techmonitor.ai/technology/microsofts-bing-wins-the-world-cup-with-predictive-analytics-4317114` (visited on 04/10/2022).

Sutton, R. S., A. G. Barto, et al. (1998). “Introduction to reinforcement learning”. In.

Syarif, I., A. Prugel-Bennett, and G. Wills (2016). “SVM parameter optimization using grid search and genetic algorithm to improve classification performance”. In: *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 14.4, pp. 1502–1509.

Tangirala, S. (2020). “Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm”. In: *International Journal of Advanced Computer Science and Applications* 11.2, pp. 612–619.

transfermarkt.com (2022). *Premier League Matchday*. `https://www.transfermarkt.com/premier-league/spieltag/wettbewerb/GB1/saison_id/2021`. (Visited on 05/22/2022).

WhoScored.com (2022). `https://www.whoscored.com/Regions/252/Tournaments/2/Seasons/8228/Stages/18685/PlayerStatistics/England-Premier-League-2020-2021`. (Visited on 11/11/2022).

Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences*. Vol. 100. Academic press.

Wright, M. (2009). “50 years of OR in sport”. In: *Journal of the Operational Research Society* 60.1, S161–S168.

Yang, R. (2019). “Using Supervised Learning to Predict English Premier League Match Results From Starting Line-up Player Data”. In.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.
Ich erkläre mich mit der Archivierung der vorliegenden einverstanden. Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Magister-/Master-/Diplomarbeit/Dissertation eingereicht.

Datum

Unterschrift