Wrocław University
of Science and Technology

Faculty of Fundamental Problems of Technology

MASTER THESIS

Calculating bookmaker odds using machine learning methods

Michał Orfin
Supervisor: prof. dr hab. Michał Woźniak

keywords: machine learning, betting odds, bookmaking, football

Wrocław 2024

Contents

1. Introduction

Ever since the conception of professional football in 19th-century England [1], the sport has been inseparably connected to, at that time illegal, betting. What first started as a (not so)innocent addition to bring more emotion and stakes for viewers, quickly became the road to the glory of some and the downfall of others. With full legalisation and standardization of gambling provided by the British Act of Parliament "The Betting and Gaming Act" established on 1st of September 1960 [2] which directly coincided with enthusiastically developing field of data gathering in football started by Thorold Charles Reep in 1951 [3], came wide-spread popularity and rapid growth. Both domains helped each other out considerably. Legalisation of betting gave more incentive to develop methods of gathering data about football matches, which was used to support bookmakers in their endeavors of setting up the odds.

Throughout the next 60 years, both fields changed beyond recognition. What started as small betting shops making their way out of hiding became multi-million-pound companies employing thousands of people all around the world. What started as gentlemen similar to aforementioned Mr. Reep writing the number of passes made between players on a piece of paper became a lucrative business used not only by betting companies but by football clubs themselves [4]. Nowadays, the landscape of data gathering in football has expanded to an astounding degree, evolving far beyond the early days of scribbling notes on paper. Developing technologies have led to the creation of highly sophisticated and precise tools for data collection turning every second of a football match into a potential piece of insightful information. This includes, among others, multiple high-definition cameras tracking the movement of players on the pitch, sensors in players' boots tracking their movement, speed, and positioning, or even chips inside the ball that measure its trajectory and speed. Coupled with many more methods of gathering data it is possible to generate a remarkably detailed picture of every match.

Moreover, the implementation of Artificial Intelligence (AI) techniques such as machine learning further elevates and widens the possibilities of these data sources. It is documented [5] [6] [7] that machine learning can be used to evaluate various aspects of a football match. These insights can lead to adjustments of players' positioning, team tactics, training regimes, scouting, and most importantly, predictions of outcomes where the motivation for this thesis lies.

In the continually evolving landscape of sports analytics, the growth and complexity of football data present an untapped resource for making informed predictions. The primary objective of this thesis is not only to leverage the potential of machine learning algorithms to derive meaningful insights from this raw data but also to develop a model that accurately predicts Premier League football match outcomes.

The approach of this research centers around utilizing historical data to feed various machine learning models and to investigate the significance of different features within this data. The goal here is to identify the model that best predicts match outcomes and discern the features that have the highest contribution to these predictions.

Moreover, this research seeks to use the predictions of these models to recreate a process of establishing bookmaker odds, thereby offering a novel approach to calculating 1x2 bet odds. The bookmaker odds serve a dual purpose in this study. Firstly, they provide a form of direct comparison for the predictions made by the machine learning models. Secondly, they offer valuable insight into the influence of various factors on the outcome of a match.

In essence, this thesis serves to contribute to the broader field of sports analytics by showcasing the application of data science in football. It highlights how machine learning can transform seemingly meaningless data into accurate predictions, thus paving the way for new possibilities in the realm of sports predictions and betting.

2. Ins and outs of bookmaking

In the context of sports betting, bookmaking involves setting or adjusting odds for various outcomes of a sporting event. The individuals or organizations that perform this task are known as bookmakers or "bookies". The goal of a bookmaker is to set odds in such a way that they attract bets on all possible outcomes of an event, ensuring that they make a profit regardless of the result. Odds are set based on the perceived likelihood of a certain outcome. This is most often based on a combination of statistical analysis involving complicated algorithms and the subjective judgment of experts in that field. The lower the odds, the higher the probability of an outcome. If Team A is perceived to be stronger than Team B then the odds for Team A's win will be lower. The easiest example and the type of bet calculating which will be one of the aims of this thesis, is "1x2", also known as "win-draw-win" bet. It describes the possibility of 3 outcomes:

— 1: Home Team wins
— X: The match ends in a draw
— 2: Away Team wins

It is important to stress that in the majority of football leagues, each team plays all other teams twice, once at Home and once Away. The venue at which the match is played has a big influence on the prediction since it is documented [8] that the majority of teams score more goals in front of their home crowd, concede more on away trips, thus, logically, leading to bigger accumulation of points at their grounds. The odds are decimal numbers bigger than 1 calculated based on the percentage chance of an outcome happening with the caveat that the sum of all outcomes might raise above 100%. This is called overround or vigorish ("vig") and is the way for a bookmaker to ensure that they make a profit. The formulae to calculate the odds looks like this:

$$Odds = \frac{1}{P} \tag{2.1}$$

Where 'Odds' represent the odds for the event and 'P' represents the probability of the event in decimal. For example, if the probability of Team A winning is 0.5 (50%), the odds would be calculated as 1/0.5 = 2.00. This means that for every X amount of money you bet, the payout would be 2X if Team A wins. It is important to mention that the odds depend on the bookmaker and can differ between different companies. As stated above, they are calculated using algorithms and the subjective opinions of experts. This aspect can be significant when it comes to matches played between equal teams where Betting Company A might favour Team A and Betting Company B might favour Team B.

Furthermore, depending on many factors such as player injuries, changes in the team line-up, and even the behaviour of punters themselves, odds change dynamically leading up to the event. However, upon placing the bet the odds for your bet are locked in that moment. This ensures both the bettor and the bookmaker are protected from subsequent changes. This presents the opportunity to strategize and try to "beat the closing line", meaning predicting when the odds for a desirable outcome will be the highest and placing the bet at that moment. With that being said, similarly to the popular saying "Casino always wins" it is safe to say that bookmaking companies always assemble things in a way to make a profit. Setting the odds in their favour, limiting the maximum bet size, shifting the odds based on the amount of money placed on each outcome, and sometimes even refusing bets from successful bettors are some of the controversial and criticized methods of ensuring profits.

Although this thesis focuses on the 1x2 bet it is worth mentioning that it is only one of the hundreds of possible bets offered by betting companies, i.e. over/under bets, correct score bets, first goal scorer bets, yellow card bets, and many more.

3. Database

In this chapter, a detailed exploration of the dataset that forms the cornerstone of this research will be presented. This dataset is a comprehensive collection of match statistics from the English Premier League's 2021-2022 and 2022-2023 seasons. Its richness in variables provides a robust foundation for investigating the predictive potential of various match features. However, before exploring predictive modeling, it is vital to understand the structure and characteristics of the data at hand. Accordingly, this chapter will elucidate the key components of our dataset, discussing its origin, size, and structure as well as some of the more important variables that had a direct influence on the research. Additionally, the process of data preprocessing will be explained and discussed which was essential to ensure the data's readiness for analysis. By the end of this chapter, the reader should have a solid understanding of the nature of the data driving this research, the steps taken to prepare it, and the modeling process.

3.1. Origin

Throughout the analysis of accessible online databases resulted in the site fbref.com [9] being chosen as the source of statistics. Necessary data was gathered through the process of web scraping, also known as web harvesting or data extraction. This is a technique used to collect large amounts of data from websites where data is unstructured or not easily accessible through direct downloads. The process of web scraping involves writing an automated script using one of many programming languages like Python, which was the case for this thesis. This script is used to access the web pages of a specified website and extract the desired information. This script, or 'bot', sends a 'GET' request to the target website's server to access its data. It then parses the HTML or XML response sent back by the server to find and extract the required information.

In the case of this thesis, web scrapping was used to collect match data for every single team that participated in the 2021-22 and 2022-2023 seasons. The automated script was designed to access the page for each of the 20 teams, extract the relevant, subjectively chosen statistical data, and compile it into a structured database. This newly formed dataset served as the base of research.

3.2. Size and Structure

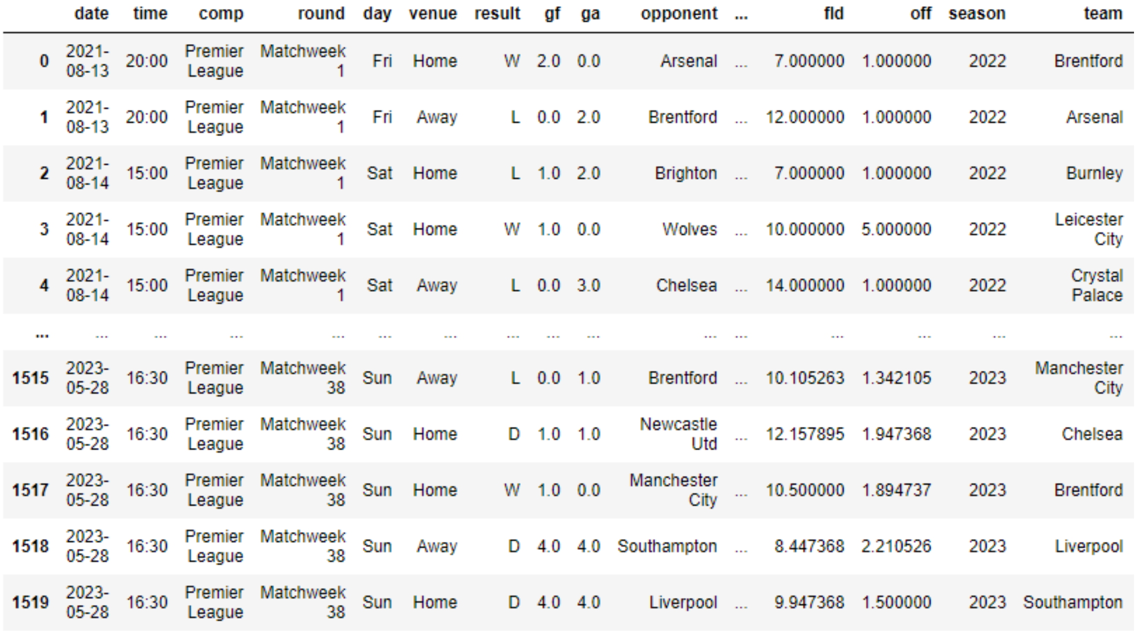
The focal point of the search for usable data is the Premier League table containing ref links to statistics of each team that played during the season. In a single Premier League season, 20 teams play each other 2 times resulting in 380 matches played over 10 months, from August to May. Upon entering any of the participants' detailed pages we are presented with 9 separate tables, each describing one of the aspects of the game. Those sections are:

1. Scores & Fixtures - Basic information about the match: Date, Competition, Result, Goals for and against, Attendance, Captain, Formation, Referee,

2. Shooting - Shooting statistics of the team like Shots Taken, Shots on Target,
3. Goalkeeping - Statistics of a goalkeeper like saves made, save %, attempted passes,
4. Passing - Passing stats of the outfield players like the total amount of passes attempted, completed, distance, and numbers of each pass,
5. Pass Types - Further division of passing into types of passes like deadball, liveball, free kicks, corners,
6. Goal and Shot Creation - Shot Creating Actions (SCA) and Goal Creating Actions (GCA) and sources of each
7. Defensive Actions - Number of Tackles, Interceptions, Challenges, and Blocks
8. Possession - Percentage possession of the ball, number of touches in each part of the pitch, number of Carries of the ball
9. Miscellaneous Stats - Number of fouls, cards, offsides

Each match is represented separately for each of the two teams. For example, if a match played between Team A and Team B is considered, gathering the full scope of that event requires scrapping all 9 tables for both teams and either combining them into a singular row in a newly created dataset or linking them with a set of common columns like date and matchweek. The second approach is the one chosen in this thesis.

Due to the excessive amount of data in the aforementioned tables (around 120 sometimes overlapping columns between each table) a subjective selection of features needed to be performed. Based on personal experience and science articles [10] [11] 34 statistics have been chosen from each section of the fbref.com database alongside other descriptive columns to formulate a database for this project. The process of scrapping resulted in the creation of a 1520x52 table with all the necessary information to perform a machine-learning process.



	date	time	comp	round	day	venue	result	gf	ga	opponent	...	fld	off	season	team
0	2021-08-13	20:00	Premier League	Matchweek 1	Fri	Home	W	2.0	0.0	Arsenal	...	7.000000	1.000000	2022	Brentford
1	2021-08-13	20:00	Premier League	Matchweek 1	Fri	Away	L	0.0	2.0	Brentford	...	12.000000	1.000000	2022	Arsenal
2	2021-08-14	15:00	Premier League	Matchweek 1	Sat	Home	L	1.0	2.0	Brighton	...	7.000000	1.000000	2022	Burnley
3	2021-08-14	15:00	Premier League	Matchweek 1	Sat	Home	W	1.0	0.0	Wolves	...	10.000000	5.000000	2022	Leicester City
4	2021-08-14	15:00	Premier League	Matchweek 1	Sat	Away	L	0.0	3.0	Chelsea	...	14.000000	1.000000	2022	Crystal Palace
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1515	2023-05-28	16:30	Premier League	Matchweek 38	Sun	Away	L	0.0	1.0	Brentford	...	10.105263	1.342105	2023	Manchester City
1516	2023-05-28	16:30	Premier League	Matchweek 38	Sun	Home	D	1.0	1.0	Newcastle Utd	...	12.157895	1.947368	2023	Chelsea
1517	2023-05-28	16:30	Premier League	Matchweek 38	Sun	Home	W	1.0	0.0	Manchester City	...	10.500000	1.894737	2023	Brentford
1518	2023-05-28	16:30	Premier League	Matchweek 38	Sun	Away	D	4.0	4.0	Southampton	...	8.447368	2.210526	2023	Liverpool
1519	2023-05-28	16:30	Premier League	Matchweek 38	Sun	Home	D	4.0	4.0	Liverpool	...	9.947368	1.500000	2023	Southampton

Figure 3.1. Example showcase of the database table

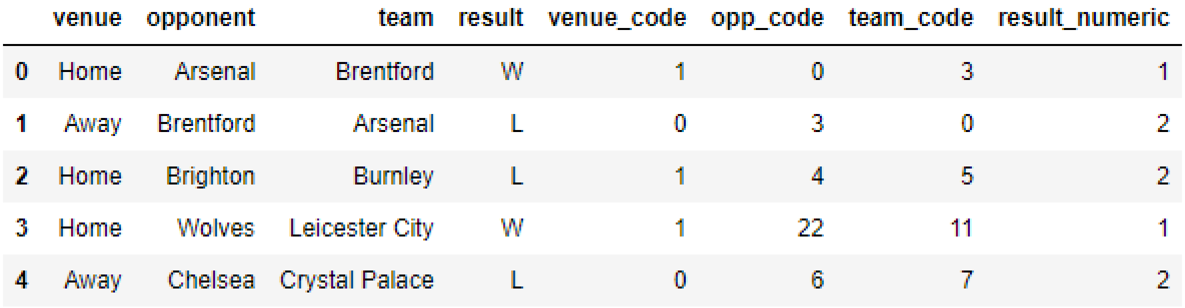
3.3. Preprocessing

Before machine learning can be initiated preprocessing the data needs to be performed. It refers to the transformation and cleaning of raw data into a structured format suitable for analysis or machine learning algorithms. The primary steps involved are:

1. Data Cleaning - helps to handle missing and noisy data. Missing data can be handled by using techniques such as default filling (filling with a specific value), mean, median, and mode filling, or using machine learning algorithms like KNN or regression to predict missing values. Noisy data, which refers to Random variance or errors in a measured variable, can be handled by binning, regression, or clustering.
2. Data Integration - involves combining data from different sources and providing users with a unified view of the data. This process can result in data redundancy and needs proper handling.
3. Data Transformation - It involves transforming the data into an appropriate form suitable for the mining process. This could involve normalization, aggregation, or generalization.
4. Data Reduction - Large datasets can often be reduced in volume yet produce the same, or almost the same, analytical results. Strategies for data reduction include dimensionality reduction, numerosity reduction, and data compression.
5. Data Discretization - This step is required for converting continuous attributes into categorical attributes. Discretization can make the mining process faster and less resource-intensive, but at the same time, important information might be lost during this process.

Fortunately, databases on fbref.com are well organised and are not missing any data so the main goal of preprocessing was preparing parts of the database to be eligible for machine learning. Machine learning algorithms only accept numeric values as predictors and targets thus necessary encoding needed to take place:

1. "venue" column - numbers set for Home - 1 and Away - 0
2. "opponent" column - name of the away/opposition team, each team was assigned an integer number
3. "team" column - name of the home team, similarly to the "opponent" column each team was assigned an integer number
4. "result" column - the most important column that includes the result of each match, Wins "W" mapped as 1, Draws "D" as 0, and Loss "L" as 2 to reflect the 1x2 bet



	venue	opponent	team	result	venue_code	opp_code	team_code	result_numeric
0	Home	Arsenal	Brentford	W	1	0	3	1
1	Away	Brentford	Arsenal	L	0	3	0	2
2	Home	Brighton	Burnley	L	1	4	5	2
3	Home	Wolves	Leicester City	W	1	22	11	1
4	Away	Chelsea	Crystal Palace	L	0	6	7	2

Figure 3.2. Showcase table for preprocessing of the data

Additionally, unwanted columns "match report", "notes", "referee", "captain", and "formation" with insignificant data were removed. Following that necessary preparations for specific machine learning algorithms were performed. Firstly, a copy of the database was created where all values in columns were normalised for one of the soon-to-be-introduced algorithms to work properly. The final step of pre-processing required the last matchweek of season 2022-2023 data to be substituted with averages for each team throughout the season. At the time of writing this thesis season, 2022-2023 of the Premier League was already concluded and fixtures for season 2023-2024 were not released yet, thus the need to artificially create matches that weren't played that could be predicted by the models and those predictions evaluated. The reason for this step will be explained in further chapters of this thesis.

4. Experimental evaluation

This chapter delves into the technical aspects and methodology behind the work performed for this thesis. It discusses the software and hardware environment used for the development, experimentation, and analysis throughout the research. Specifically, the usage of Python programming language and Jupyter Notebook as the primary development environment is covered. The chapter also elucidates on the key Python libraries employed in the course of this research, such as BeautifulSoup for web scraping, Pandas for data manipulation, Scikit-learn for machine learning, Matplotlib and Seaborn for data visualization, and XGBoost for utilizing gradient boosting frameworks. Following that machine learning algorithms will be described in detail focusing on the principles of their work. In addition, the metrics used to evaluate the performance of the models are discussed, providing a detailed rationale for their selection. Finally, the chapter outlines a detailed plan of the experimental protocol, discussing each step of the development and testing process in a chronological manner. This chapter serves as a foundation for understanding the technical implementation of the work and as a guide for replicating the methodology. The link to the GitHub repository containing the entire project will be placed at the end of this chapter. The experimental evaluation aims to answer the following research questions:

4.1. Which machine learning model predicts outcomes most accurately?

Different machine learning models present distinct approaches to predicting the outcomes. With their unique strengths and weaknesses, it is important to establish proper feature and hyperparameter selection. Complexity and vastness of characteristics of the data those models are dealing with means it is essential to prepare a thorough analysis of the outcomes and discover which model captures the intricacies of the game more accurately than others.

This research aims to compare the performance of various machine learning models, like Random Forest, K-Nearest Neighbors, and others on the same dataset based on various metrics which will be discussed in detail in later chapters. Hopefully, at the end of this thesis, a conclusion will be formed clearly pointing out the strengths and weaknesses of each of the studied algorithms.

4.2. Which features are the most important?

Features in machine learning models are the individual, measurable properties or characteristics of the observations being analyzed. Some features might turn out to be more informative than others for making accurate predictions. Understanding the real-life importance of those characteristics and how each model evaluates those features is crucial for this research. The importance of each feature will be determined based on feature importance scores provided by the machine learning model's built-in metrics. Based on experience with the field of football it is safe to assume that features directly connected to scoring or conceding goals will have a higher correlation with the result than statistics like the amount of yellow cards a team received during the match. Of course, during a game, a player with a yellow card might have to play with more caution which might directly lead to said player backing off from an important challenge thus resulting in his team conceding a goal but it is much harder to quantify that type of influence in the statistics and show its importance.

4.3. What impact does the size of the time window for training data have on prediction accuracy

The proper selection of training data is a crucial aspect of machine learning, especially when it comes to time-related data. When predicting matches in football, one of the most important aspects is looking at the team's form, which is how well they've done in X amount of previous matches. For example, if a team has lost the majority of their last 10 matches and statistics reflect that it is safe to assume they won't be the favourites in their next match unless playing against a much weaker opposition. Similarly, a team can suffer a few defeats in spite of playing well and a bold prediction can be made that they will do well in their upcoming game. The further away they match is from the present time the less importance it has in making such predictions.

The database for this thesis consists of matches in chronological order from 2 seasons of the Premier League. Usually in machine learning an 80/20 ratio is applied when it comes to splitting the data into training and test sets. As mentioned in previous chapters, the aim is to predict and compare the outcomes of the last matchweek of the 2022-2023 season but that doesn't mean that only those 10 matches will be the "test" part of the algorithms. Three ways of splitting the database will be carried out:

1. Both seasons will be taken into consideration, training set will consist of over 1.5 seasons starting from August 2021 and ending in February 2023
2. Only season 2023 will be taken into consideration, training set will consist of 80% of the matches played in that season from August 2022 to the beginning of April 2023
3. Only matchweeks 28-38 will be taken into consideration

This approach should correctly reflect the difference between predicting based on historical data and recent form. Between each season teams can often change dramatically, players are sold and bought, coaching staff changes, and with it tactics, training, and many other aspects. The team's performances might differ drastically. For example, Chelsea F.C. finished the 2021-2022 season in 3rd place but only managed 12th place finish in the following season, similarly Leicester City who finished 8th in the previous season got relegated from the league in 18th this season. It leads to the conclusion that data from previous seasons might pollute the predictions with biases.

4.4. How accurate would odds calculated based on machine learning predictions be compared to real-life odds?

Here the aim is to translate predictions of models into the aforementioned "1x2" type of bet and check which model serves this purpose the best way possible. The accuracy of predictions will be highly dependent on the previous steps mentioned. Proper feature and hyperparameter selection, training, and test split will check if a model is capable of successfully forecasting a football match. As mentioned above, the model's resistance to biases and competency in allowing outlier outcomes will be tested. It is important to stress that odds in sports betting are set by bookmakers, who consider a wide range of factors such as team performance, player injuries, and historical match outcomes alongside statistical analysis. This thesis approach will forego the application of expert knowledge and focus only on checking the models themselves. A suitable comparison of outcomes between the models and with real-life betting companies will be presented.

4.5. Language and environment

Python programming language was chosen for this thesis as it provides easy access to libraries with predefined methods for scrapping, machine learning, and visualization of data. The simplicity of the structure of code in Python and its intuitive syntax allowed to develop an easy-to-follow scheme that can be replicated and modified if need be. This language is also one of the most commonly known programming languages and it is often used in the data science community which means there is a wealth of tutorials, guides, and other resources available, which can be particularly useful for troubleshooting and learning new techniques.

Additionally, Jupyter Notebook was chosen as the development environment for the project due to its interactive nature. This environment enables code, visualizations and formatted text to be combined in a single document proving to be an excellent tool for data exploration, prototyping, documentation, and result analysis and showcase.

4.6. Libraries

A variety of libraries were used as support for this thesis:

1. BeautifulSoup is a library used for web scraping purposes. It allows to pull data out of HTML and XML files. It creates a parse tree from page source code that be used to extract data in a hierarchical and readable manner.
2. Pandas is a software library for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
3. Scikit-learn (sklearn) is a free software machine learning library, it features various classification, regression, and clustering algorithms. It is designed to interoperate with the numerical and scientific libraries NumPy and SciPy
4. Matplotlib and Seaborn based on Matplotlib are plotting libraries. They provide an object-oriented API for embedding plots into applications.
5. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.

4.7. Machine Learning Algorithms

For the purpose of this work, it was decided that programming will be limited to simple Machine Learning algorithms. Based on available information about the handling of data in those algorithms and what type of data they are best used with, 4 algorithms were chosen to be studied:

1. Random Forest Classifier is a versatile machine learning method capable of performing both regression and classification tasks through means of an ensemble learning method where a few weak modes combine to form a powerful one. In Random Forest (RF), multiple trees are grown as opposed to a single tree-like in the Decision Tree method. To classify a new

object based on attributes, each pre-specified number of trees give a classification. The forest chooses the classification having the most votes from all the trees in the forest.

2. Support Vector Machine - is a supervised machine learning algorithm that can be used for both classification or regression challenges. In SVM, each item from the dataset is plotted in an N-dimensional space (Where N is the number of features in the dataset), with the value of each feature being the value of a particular coordinate. Then, classification is performed by finding the hyperplane that differentiates the two classes very well.
3. K-Nearest Neighbours is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. However, it is more widely used in classification problems in the industry. KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. KNN algorithm stores all the available data and classifies a new data point based on the similarity,
4. Extreme Gradient Booster is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It produces an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. The name XGBoost, though, actually refers to the engineering goal to push the limit of computational resources for boosted tree algorithms which is the reason why many use it.

4.8. Metrics, Feature selection, and Hyperparameters search

Methods contained in previously mentioned libraries were the key to proper analysis of the data. To properly understand the research of this thesis it is essential to grasp the essence of these methods.

4.8.1. Feature selection

Initially, the database consisted of 34 columns with characteristics describing each match. For the majority of Machine Learning algorithms used in this thesis, it was essential to trim down the number of arguments passed to those algorithms. This task was achieved with the help of feature selectors. It is important to stress that there are multiple different selectors that can be used to perform this task and some of them are unique to some algorithms. It was decided that for the purpose of this thesis selector that is common for all employed algorithms will be used to provide an easy comparison between models - RFE, with the addition of different algorithm-specific selectors as a comparison for RFE for each of the models.

RFE - Recursive Feature Elimination is a feature selection method that was used for all algorithms. The goal of this selector is to perform a search for features recursively considering smaller and smaller sets of features. It does this by fitting the model, ranking the features based on importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains.

Correlation Matrix is a table that shows the correlation coefficients between many variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data and find what variables are most related to each other.

4.8.2. Hyperparameters search

This process involves choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. Different from the parameters of the model which are learned during the training phase, hyperparameters are set before the learning process begins. Techniques such as grid search or randomized search were used to methodically build and evaluate a model for each combination of algorithm parameters specified in a grid or range. This allows for fine-tuning of machine learning models in order to improve their performance.

4.8.3. Metrics

Metrics are used to describe the accuracy of the model's predictions. They are vital to the proper analysis of the outcomes of machine learning. Methods used for this thesis are all derived from sklearn library.

1. Accuracy Score is one of the simplest evaluation metrics for classification. It is the ratio of correct predictions to the total number of predictions. In the multiclass case, it calculates the total number of correctly classified instances across all classes.
2. Confusion Matrix in the context of multiclass classification, is an N x N matrix, where N is the number of classes. Each row of the matrix represents the instances in an actual class, and each column represents the instances in a predicted class. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier.
3. Classification Report: A classification report gives a detailed breakdown of precision, recall, and f1-score for each class in the multiclass problem, as well as some overall metrics.
   a) Precision for a class is the number of true positives (i.e., the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e., the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).
   b) Recall for a class is the number of true positives divided by the total number of elements that actually belong to the positive class (i.e., the sum of true positives and false negatives, which are items that were not labeled as belonging to the positive class but should have been).
   c) f1-Score for a class is the harmonic mean of precision and recall. An f1-score reaches its best value at 1 and its worst score at 0. It is a good way to show that a classifier has a good value for both recall and precision.
   d) Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing.
   e) Micro-average: Aggregating the contributions of all classes to compute the average metric. In a multi-class classification setup, micro-average is preferable if you suspect there might be a class imbalance.
   f) Macro-average: Compute the metric independently for each class and then take the average (hence treating all classes equally).
   g) Weighted-average: Compute the metric for each class independently but when it adds them together uses a weight that depends on the number of true instances for each class.

4.9. Experiment Protocol

In this section, a detailed structure of the code will be presented with descriptions and targets each part of the code was set to achieve. Additionally, in the end, the link to the GitHub repository will be provided.

4.9.1. Web scraping

1. Scraping of initial "https://fbref.com/en/comps/9/Premier-League-Stats" page using requests
2. Parsing HTML links using BeautifulSoup library
3. Extraction of match statistics using requests and pandas
4. Creation of subsets for each of the sections mentioned in 3.2 of this thesis
5. Merging of subsets into a dataset based on common columns
6. Saving to .csv file

4.9.2. Calculating predictions using machine learning

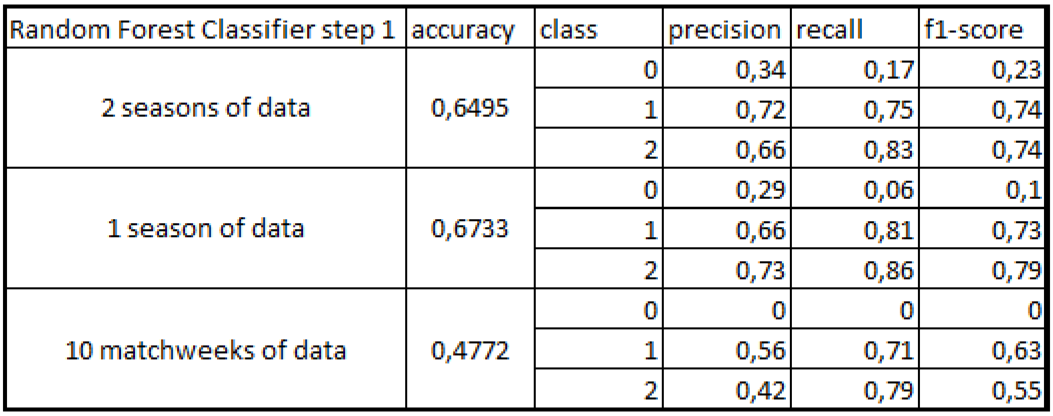
1. Data Preprocessing
2. Establishing parameter grids, mappings, and other variables
3. Random Forest Classifier
   a) Splitting database into training and test sets
   b) Initializing model
   c) Calculating accuracy score, confusion matrix, and classification report using initial conditions
   d) Feature selection and hyperparameters search
   e) Retraining of the model using new sets of features and hyperparameters
   f) Calculating new accuracy score, confusion matrix, and classification report
   g) Finding out what the best subset of features is
   h) Using a final subset of features and chosen hyperparameters to run the training of the model in a for loop using different seeds
   i) Creating a new dataset containing results of multiple predictions and calculation of "1x2" bet
4. Support Vector Machine - follows the same scheme as Random Forest Classifier
5. K-Nearest Neighbours
   a) Creating a copy of the original dataset and normalizing the values
   b) Next points are the same with the exception of the last 2 - KNN has no option to change seeds, it always yields the same results for a set of features and hyperparameters
6. Extreme Gradient Boost - follows the same scheme as RFC and SVM

Link to the repository containing source code for this thesis:
https://github.com/Siemik1997/Football-Match-Predictions-with-Machine-learning—Master-Thesis

5. Results

In this chapter, the results of this thesis' aim to explore and evaluate the effectiveness of different machine learning models will be presented. The investigation into this problem was systematically organized into a three-step process. First, models were built for data splits described in 4.3. This approach leads to a better understanding of the performance of each model across different time frames and varying data compositions. The second step involved testing the models using various feature subsets. This exploration was critical to understanding the impact of specific variables on the model's predictive powers. The final step involved using trained models to forecast upcoming match outcomes. In the following sections details of each step will be presented, providing a comprehensive view of the results obtained and comparisons between the machine learning models employed.

5.1. Random Forest results



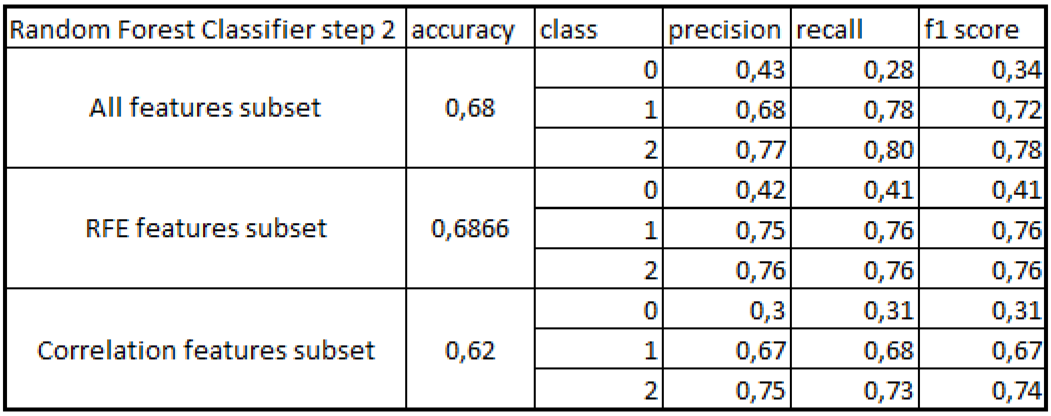
Random Forest Classifier step 1	accuracy	class	precision	recall	f1-score
2 seasons of data	0,6495	0	0,34	0,17	0,23
		1	0,72	0,75	0,74
		2	0,66	0,83	0,74
1 season of data	0,6733	0	0,29	0,06	0,1
		1	0,66	0,81	0,73
		2	0,73	0,86	0,79
10 matchweeks of data	0,4772	0	0	0	0
		1	0,56	0,71	0,63
		2	0,42	0,79	0,55

Figure 5.1. Results of Random Forest Classifier predictions for different time splits

For the two-season split, the Random Forest Classifier showed an accuracy of 64.95%. Precision was highest for class 1 and class 2, with values of 72% and 66%, respectively, while class 0 had a precision of 34%. The recall for class 1 and class 2 was also high (75% and 83%, respectively), but significantly lower for class 0 (17%). The confusion matrix shows a higher level of misclassification for class 0, with more instances being classified as class 1 or class 2. This indicates a possible bias in the model towards class 1 and class 2 for this particular split.

For the one-season split, there was a slight increase in overall accuracy, achieving 67.33%. The precision and recall rates for class 0 remained relatively low but increased for class 1 and class 2. In this split, the model still shows bias towards class 1 and class 2 but less misclassification for class 0 was observed, compared to the two-season split.

For the recent matches split, the accuracy fell to 47.73%. Precision and recall rates for class 0 dropped drastically to 0%, indicating that the model was unable to correctly classify any instance of class 0 in this split. This suggests that the model may not be well suited to making predictions based on data from recent matches only.



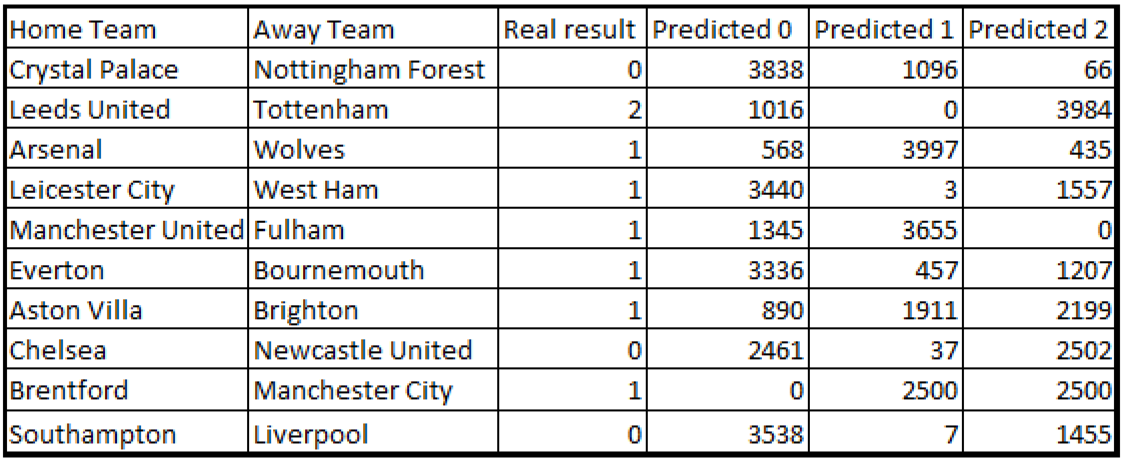
Random Forest Classifier step 2	accuracy	class	precision	recall	f1 score
All features subset	0,68	0	0,43	0,28	0,34
		1	0,68	0,78	0,72
		2	0,77	0,80	0,78
RFE features subset	0,6866	0	0,42	0,41	0,41
		1	0,75	0,76	0,76
		2	0,76	0,76	0,76
Correlation features subset	0,62	0	0,3	0,31	0,31
		1	0,67	0,68	0,67
		2	0,75	0,73	0,74

Figure 5.2. Results for RFC for different sets of features

The first subset of features, which consisted of all available features, yielded an accuracy of 0.68. The precision and recall scores were relatively balanced across the classes, indicating that the model was not particularly biased toward any specific class. Despite the numerous features, this set did not provide the highest accuracy, suggesting the presence of irrelevant or redundant information that may have added noise to the model.

The second subset, named is based on feature selection using RFE. With this subset, the model showed a slight improvement in accuracy, achieving 0.69. Precision and recall rates remained well balanced, showing the model's continued ability to avoid bias towards any specific class. The performance improvement suggests that this subset contained more relevant features for prediction, and removing less important features helped the model focus on key information.

The third subset was selected based on the highest correlations with the target variable. Surprisingly, this subset resulted in a lower accuracy of 0.62, which was the lowest among the three tested subsets. This could be due to correlation not necessarily implying causation – a feature could be strongly correlated with the target variable but not actually be useful in predicting it. This outcome underlines the importance of considering other feature selection methods beyond simple correlation.



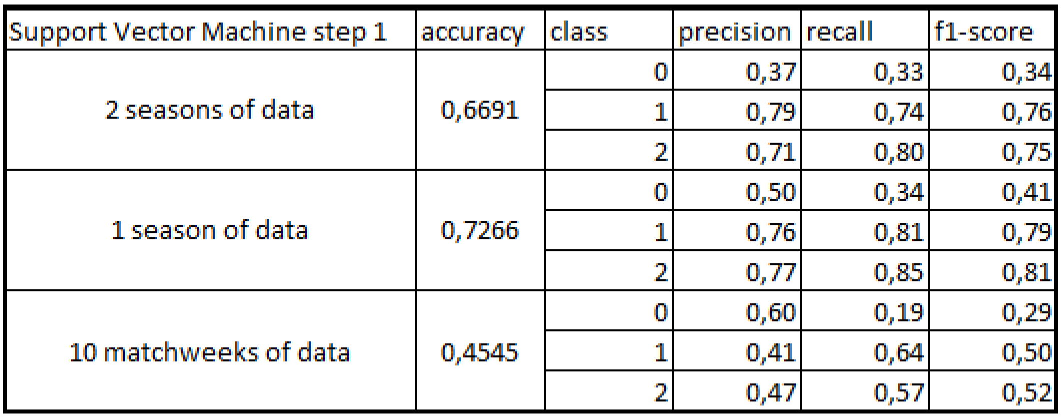
Home Team	Away Team	Real result	Predicted 0	Predicted 1	Predicted 2
Crystal Palace	Nottingham Forest	0	3838	1096	66
Leeds United	Tottenham	2	1016	0	3984
Arsenal	Wolves	1	568	3997	435
Leicester City	West Ham	1	3440	3	1557
Manchester United	Fulham	1	1345	3655	0
Everton	Bournemouth	1	3336	457	1207
Aston Villa	Brighton	1	890	1911	2199
Chelsea	Newcastle United	0	2461	37	2502
Brentford	Manchester City	1	0	2500	2500
Southampton	Liverpool	0	3538	7	1455

Figure 5.3. Predicted results for Matchweek 38 games for RFC

1. The model is most certain about the match between Leeds United and Tottenham, with a strong inclination towards Tottenham winning (3984 out of 5000 votes), and Leeds United not predicted to win at all.
2. Similarly, the match between Brentford and Manchester City is heavily skewed towards a draw or Manchester City winning, with Brentford not predicted to win at all.
3. For matches like Arsenal vs Wolves and Manchester United vs Fulham, the model has a strong inclination towards a specific team winning (Arsenal and Manchester United, respectively).
4. Some matches like Crystal Palace vs Nottingham Forest and Southampton vs Liverpool show a strong inclination towards a draw, with the second most likely result being a win for Crystal Palace and Southampton, respectively.
5. Aston Villa vs Brighton and Chelsea vs Newcastle United's matches are more evenly split, indicating that these matches are the hardest to predict.
6. Interestingly, in the Everton vs Bournemouth match, despite Everton winning, the model mostly predicted a draw. This might be a sign of a close match where a minor factor could tilt the match in either direction.

This approach adds an additional layer of information to the predictions by showing the "confidence" of the model in its predictions. It also provides a good tool for understanding the uncertainty and variability in the predictions. This is especially useful in football where outcomes can be uncertain and even a strong team can have an off day. The model's predictions seem to capture this inherent uncertainty quite well.

5.2. Support Vector results



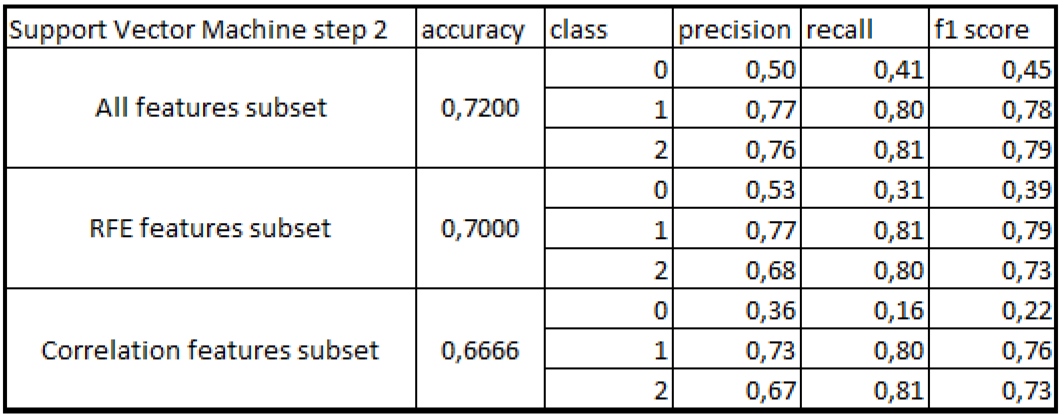
Support Vector Machine step 1	accuracy	class	precision	recall	f1-score
2 seasons of data	0,6691	0	0,37	0,33	0,34
		1	0,79	0,74	0,76
		2	0,71	0,80	0,75
1 season of data	0,7266	0	0,50	0,34	0,41
		1	0,76	0,81	0,79
		2	0,77	0,85	0,81
10 matchweeks of data	0,4545	0	0,60	0,19	0,29
		1	0,41	0,64	0,50
		2	0,47	0,57	0,52

Figure 5.4. Results for Support Vector Machine for different data splits

2-season data: The SVM model gives an accuracy of 67%, which is decent but could be improved. Looking at the confusion matrix, the model is best at predicting the "2" class (which stands for an "away win"), where it correctly identified 126 out of 158 instances. It seems to struggle more with the "0" class (draws), where it only correctly predicted 30 out of 92 instances. The precision, recall, and f1-score for each class reflect this as well. The model performs best for predicting "1" and "2" (home and away wins) and struggles with "0" (draws). It appears that the model might be biased toward predicting victories rather than draws.

1-season data: The SVM model performance improved here with an accuracy of 72.67%. Again, the model performs better on predicting wins (classes "1" and "2") compared to draws (class "0"). The confusion matrix and the classification report show that the model has a high recall for "1" and "2" and a lower recall for "0". This suggests that the model is still struggling to predict draws.

Recent matches: The performance of the SVM model decreases significantly on this data set, with an accuracy of 45%. The model shows a lower performance in predicting all classes. The precision and recall values in the classification report also indicate that the model has difficulty in accurately predicting the match results. This might be due to the smaller size of the recent matches data set and possibly due to the higher volatility in recent matches as they can be influenced by factors like injuries, player form, and more.



Support Vector Machine step 2	accuracy	class	precision	recall	f1 score
All features subset	0,7200	0	0,50	0,41	0,45
		1	0,77	0,80	0,78
		2	0,76	0,81	0,79
RFE features subset	0,7000	0	0,53	0,31	0,39
		1	0,77	0,81	0,79
		2	0,68	0,80	0,73
Correlation features subset	0,6666	0	0,36	0,16	0,22
		1	0,73	0,80	0,76
		2	0,67	0,81	0,73

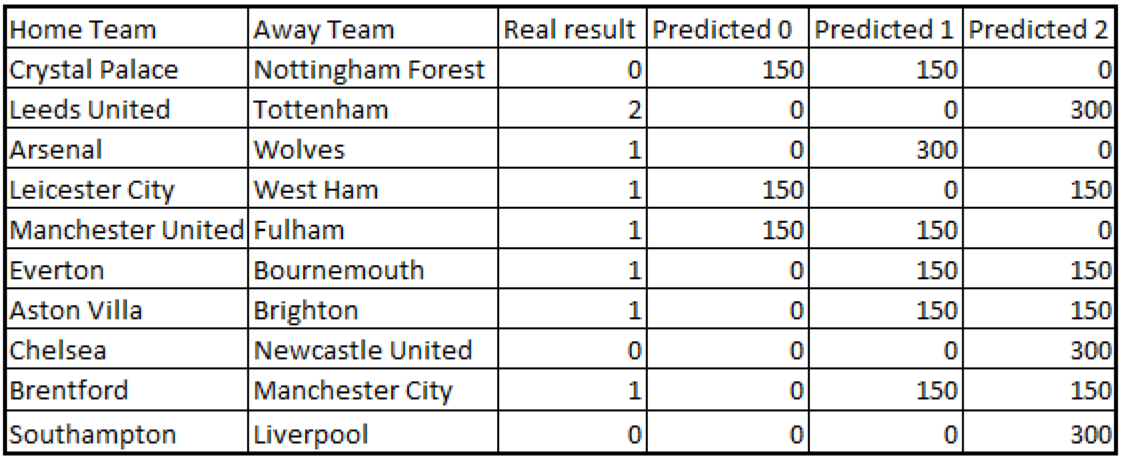
Figure 5.5. Result for SVM for different sets of features

All features: Using all the features, the SVM model gives an accuracy of 72%. The model has good precision and recall for "1" and "2", but these values are considerably lower for "0".

RFE Features: Using the features selected by Recursive Feature Elimination, the accuracy drops slightly to 70%. The precision and recall values follow the same pattern as before, with lower values for "0" and higher ones for "1" and "2". This indicates that while RFE was effective in identifying relevant features, it still wasn't enough to significantly improve the prediction of draws.

Best correlated features: Using only the best-correlated features, the accuracy drops further to 66.67%. The model's performance on predicting "0" worsens here, while its performance on predicting "1" and "2" remains relatively stable. This suggests that while correlation is a good starting point for feature selection, it may not be enough to capture all the relevant relationships between the features and the target variable.

Overall, the SVM model provides decent results, especially when all features are included. However, it consistently struggles to predict draws. This suggests that there might be some complex relationships or features specifically relevant to predicting draws that the model isn't capturing. A potential approach to improve the model could be to conduct a more detailed feature engineering and selection process, focusing on features that might be particularly relevant to predicting draws.



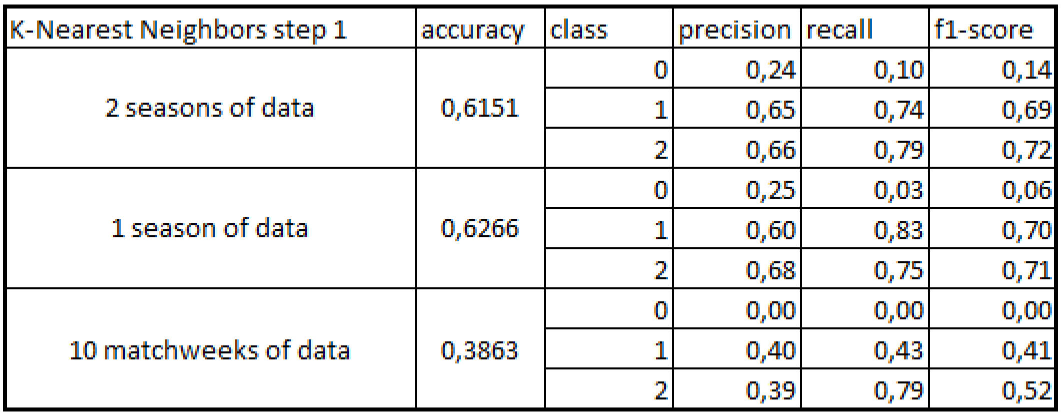
Home Team	Away Team	Real result	Predicted 0	Predicted 1	Predicted 2
Crystal Palace	Nottingham Forest	0	150	150	0
Leeds United	Tottenham	2	0	0	300
Arsenal	Wolves	1	0	300	0
Leicester City	West Ham	1	150	0	150
Manchester United	Fulham	1	150	150	0
Everton	Bournemouth	1	0	150	150
Aston Villa	Brighton	1	0	150	150
Chelsea	Newcastle United	0	0	0	300
Brentford	Manchester City	1	0	150	150
Southampton	Liverpool	0	0	0	300

Figure 5.6. Predicted results for matchweek 38 games for SVM

The model seems to have high confidence in some of its predictions, like for matches between Leeds and Tottenham, Arsenal and Wolves, Chelsea and Newcastle, and Southampton and Liverpool where it settles for 1 result. Ironically, for half of those matches, the predicted result was wrong which indicates that statistical context is only part of a good prediction.

For other matches, the models seems to be split between 2 outcomes never settling for a 3rd one. This might indicate that a match is played between relatively equal opposition or that disparities between the sides are neglected by home field advantage.

5.3. K-Nearest Neighbors results



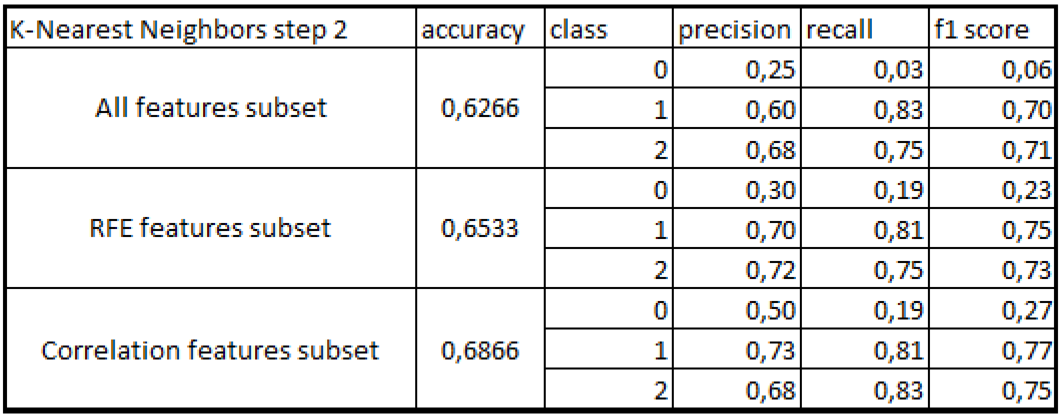
K-Nearest Neighbors step 1	accuracy	class	precision	recall	f1-score
2 seasons of data	0,6151	0	0,24	0,10	0,14
		1	0,65	0,74	0,69
		2	0,66	0,79	0,72
1 season of data	0,6266	0	0,25	0,03	0,06
		1	0,60	0,83	0,70
		2	0,68	0,75	0,71
10 matchweeks of data	0,3863	0	0,00	0,00	0,00
		1	0,40	0,43	0,41
		2	0,39	0,79	0,52

Figure 5.7. Results for K-Nearest Neighbors for different data splits

2-season data: The KNN model's accuracy is 61.52%, which is lower than that of the SVM model. The model predicts away wins (class "2") the best, correctly identifying 125 out of 158 instances. It performs poorly for draws (class "0"), correctly predicting only 9 out of 92 instances. The precision, recall, and f1-score show that the model is better at predicting wins rather than draws.

1-season data: The KNN model's performance slightly improves with an accuracy of 62.67%.

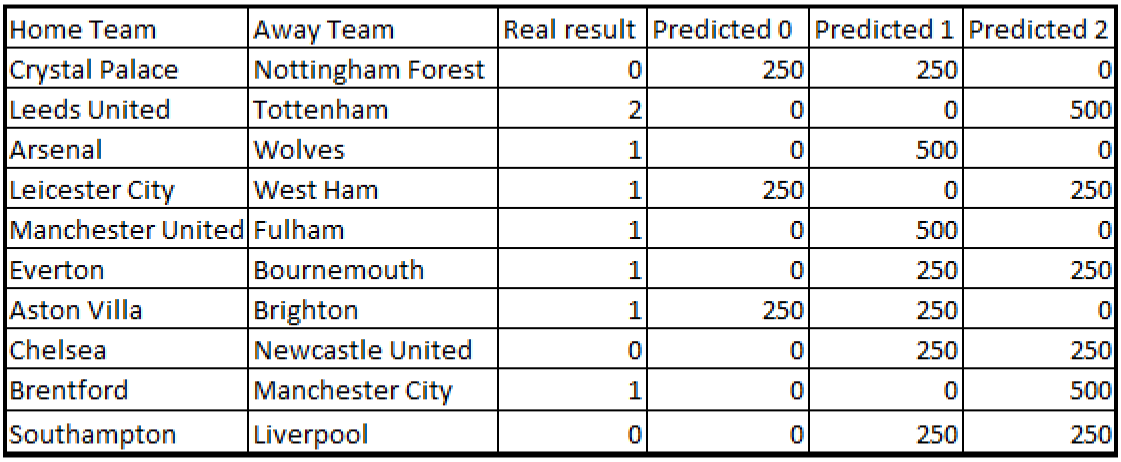
Recent matches: The KNN model's accuracy drops significantly to 38.64% for this data set. The precision, recall, and f1-score are low for all classes, which indicates the model is not predicting well. As with the SVM, this could be due to the smaller size and increased volatility of recent matches.



K-Nearest Neighbors step 2	accuracy	class	precision	recall	f1 score
All features subset	0,6266	0	0,25	0,03	0,06
		1	0,60	0,83	0,70
		2	0,68	0,75	0,71
RFE features subset	0,6533	0	0,30	0,19	0,23
		1	0,70	0,81	0,75
		2	0,72	0,75	0,73
Correlation features subset	0,6866	0	0,50	0,19	0,27
		1	0,73	0,81	0,77
		2	0,68	0,83	0,75

Figure 5.8. Result for KNN for different sets of features

A pattern is noticeable for all models, no matter what data split is applied and what features the model is being trained on, it struggles with predicting draws. Using all features for KNN yields 62,67% accuracy which is lower than SVM and RFC. Similar behaviour could be observed for features selected with RFE where accuracy stands at 65,33%. Interestingly, as opposed to the previous 2 models, K-Nearest Neighbors seems to work best with the feature subset chosen by the correlation matrix.

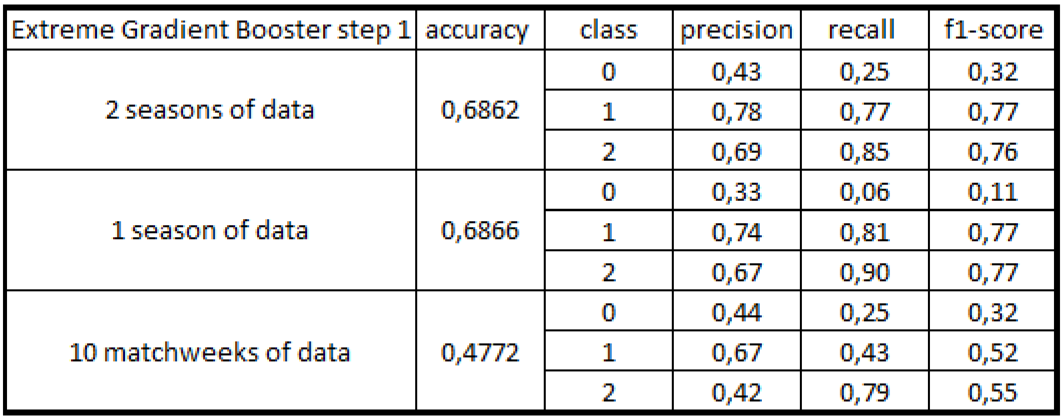


Home Team	Away Team	Real result	Predicted 0	Predicted 1	Predicted 2
Crystal Palace	Nottingham Forest	0	250	250	0
Leeds United	Tottenham	2	0	0	500
Arsenal	Wolves	1	0	500	0
Leicester City	West Ham	1	250	0	250
Manchester United	Fulham	1	0	500	0
Everton	Bournemouth	1	0	250	250
Aston Villa	Brighton	1	250	250	0
Chelsea	Newcastle United	0	0	250	250
Brentford	Manchester City	1	0	0	500
Southampton	Liverpool	0	0	250	250

Figure 5.9. Predicted results for Matchweek 38 games for KNN

It is important to note that the KNN model has no option of changing the seed for simulations which means that there is much less randomness in its predictions. In the case of SVM, the clustering of predictions meant that model is confident and if it wasn't the case, a similar situation like for RFC could've happened - a much more even spread of predictions. In the case of KNN, there is a strong case to be made that rerunning the model is pointless since it will either settle on 1 of the 2 answers or chose 1 of them and never change its opinion (allowing more randomness). For the sake of consistency, it was decided that the result table for KNN should be included too.

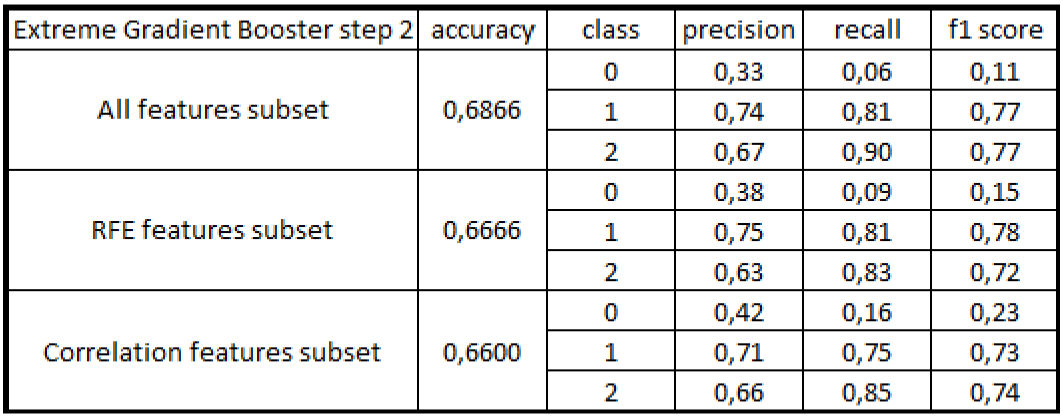
5.4. Extreme Gradient Boost results



Extreme Gradient Booster step 1	accuracy	class	precision	recall	f1-score
2 seasons of data	0,6862	0	0,43	0,25	0,32
		1	0,78	0,77	0,77
		2	0,69	0,85	0,76
1 season of data	0,6866	0	0,33	0,06	0,11
		1	0,74	0,81	0,77
		2	0,67	0,90	0,77
10 matchweeks of data	0,4772	0	0,44	0,25	0,32
		1	0,67	0,43	0,52
		2	0,42	0,79	0,55

Figure 5.10. Results for Extreme Gradient Booster for different data splits

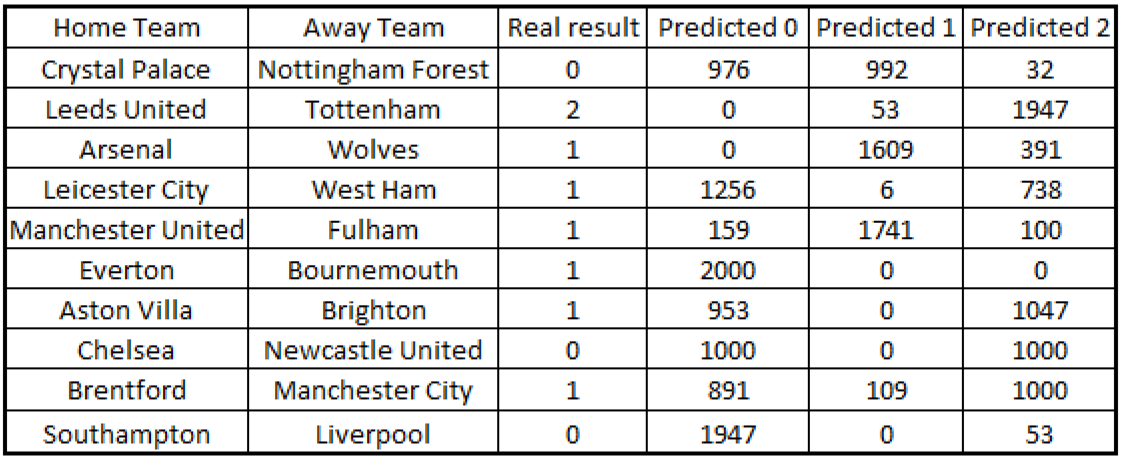
Extreme Gradient Boost showed similar behaviour to previous models. Accuracies for 2 seasons of data and 1 season of data were similar and in the range of 6̃5-70%.



Extreme Gradient Booster step 2	accuracy	class	precision	recall	f1 score
All features subset	0,6866	0	0,33	0,06	0,11
		1	0,74	0,81	0,77
		2	0,67	0,90	0,77
RFE features subset	0,6666	0	0,38	0,09	0,15
		1	0,75	0,81	0,78
		2	0,63	0,83	0,72
Correlation features subset	0,6600	0	0,42	0,16	0,23
		1	0,71	0,75	0,73
		2	0,66	0,85	0,74

Figure 5.11. Results for XGB for different sets of features

Interestingly, the XGB model similarly to KNN showed the best improvement in accuracy when working on selected hyperparameters. An increase of around 6% in accuracy indicated that for this model it is vital to perform the feature and hyperparameters selection.



Home Team	Away Team	Real result	Predicted 0	Predicted 1	Predicted 2
Crystal Palace	Nottingham Forest	0	976	992	32
Leeds United	Tottenham	2	0	53	1947
Arsenal	Wolves	1	0	1609	391
Leicester City	West Ham	1	1256	6	738
Manchester United	Fulham	1	159	1741	100
Everton	Bournemouth	1	2000	0	0
Aston Villa	Brighton	1	953	0	1047
Chelsea	Newcastle United	0	1000	0	1000
Brentford	Manchester City	1	891	109	1000
Southampton	Liverpool	0	1947	0	53

Figure 5.12. Predicted results for matchweek 38 games for XGB

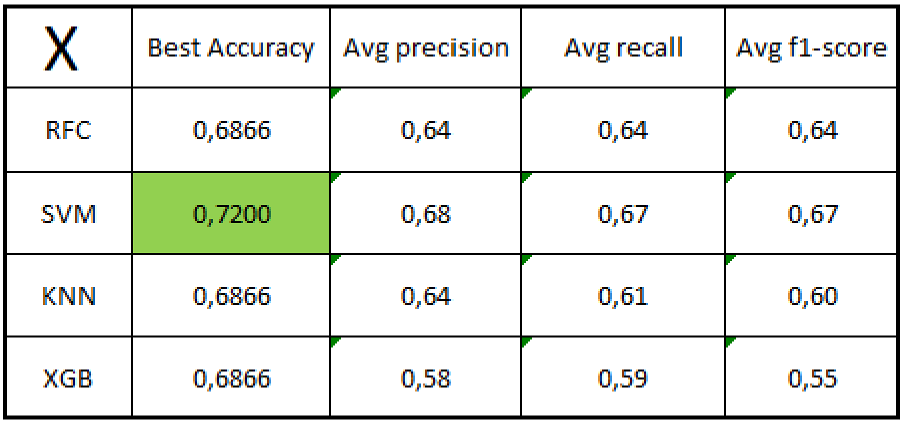
XGB is the second of the trained models that were capable of adding much-needed randomness to its results. Even though it is much more polarized when compared to RFC (if the 3rd class has records they are much lower than the other 2 classes in most cases). Interestingly, there are some matches for which classes are split equally (Chelsea vs. Newcastle) or all results were clustered into one class (Everton - Bournemouth)

6. Lessons learned

In this chapter, we transition from the objective analysis of results to an interpretative exploration of the results. It will serve as an opportunity to reflect upon the outcomes of the investigation, allowing the gain of meaningful insights and draw valuable conclusions. Each section of this chapter will be anchored by responses to the guiding questions outlined in Chapter 4. Through this process, we not only elucidate the significance of our findings but also consider their implications in the broader context of football match result prediction. This approach empowers us to distill important lessons from our work, leading to a richer understanding and providing a foundation for future research in this domain. As you read this chapter, consider each question and its corresponding response as a piece of a larger puzzle, contributing towards a comprehensive overview of our journey in this study.

6.1. Most accurate model

Comparison of multiple machine learning models on the same dataset has revealed unique strengths and weaknesses in each approach. Evaluating the model's performance should not be confined only to analysing its accuracy but also other metrics such as precision, recall, and f1-score [13].



X	Best Accuracy	Avg precision	Avg recall	Avg f1-score
RFC	0,6866	0,64	0,64	0,64
SVM	0,7200	0,68	0,67	0,67
KNN	0,6866	0,64	0,61	0,60
XGB	0,6866	0,58	0,59	0,55

Figure 6.1. Best Accuracy and average precision, recall and f1-score for each model from step 2 described in Chapter 5

The four models under investigation showed varying performances. These differences highlight the diversity of algorithms and their unique approach in tackling varying types of classification tasks [14]. Not only in terms of accuracy but also other metrics lead to Support Vector Machine emerging as the leading model [15].

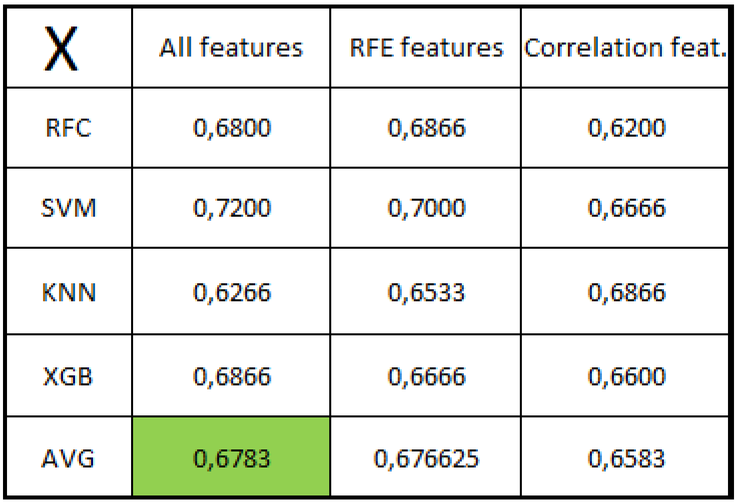
RFC and KNN demonstrated similar levels of average precision and f1-score with RFC tipping the contender in regards to recall. This suggests that KNN might struggle more with false negatives which might be caused by its sensitivity to the choice of neighbors and the dimensionality of the data [17] and points to enhanced overall generalization attributed to RFC's ensemble nature [16].

Interestingly XGBoost model, even though it exhibits similar accuracy to RFC and KNN, lags in other metrics. XGB is typically robust, however, its performance might be affected by the choice of loss function and regularization parameters [18].

In summary, these findings corroborate the multidimensional nature of model evaluation in machine learning [19]. Each model offers unique capabilities, and the choice depends on the specific requirements of the prediction task.

6.2. Most important features

Feature selection is a critical step in machine learning that has direct implications on the model's performance [20]. In conducted experiment, the accuracy of the models was compared when using all features from the dataset, features selected by Recursive Feature Elimination (RFE), and features with the highest correlation.



X	All features	RFE features	Correlation feat.
RFC	0,6800	0,6866	0,6200
SVM	0,7200	0,7000	0,6666
KNN	0,6266	0,6533	0,6866
XGB	0,6866	0,6666	0,6600
AVG	0,6783	0,676625	0,6583

Figure 6.2. Model accuracies for each algorithm based on selected features.

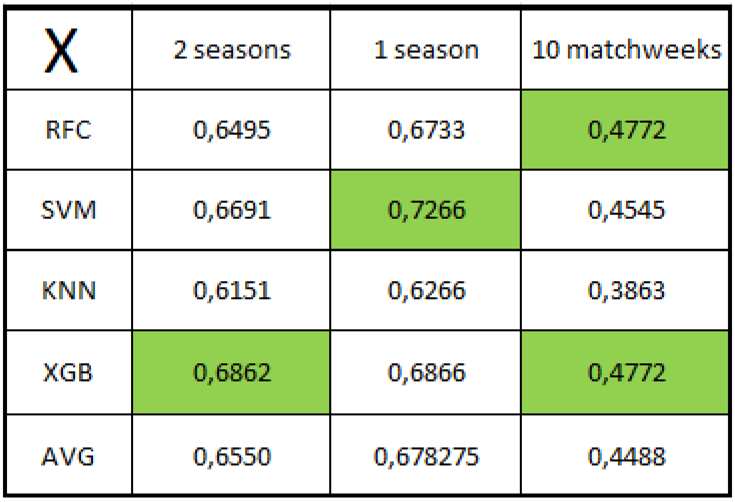
The RFC performed best when using features selected by RFE slightly outperforming the simulation using all features. SVM achieved the best accuracy when using all features and so did XGB. Extreme gradient boost noted the smallest fluctuation of accuracy between models trained on different sets of features with the disparity of 0,0266 between highest and lowest. KNN model showed the best prediction capabilities for features selected by correlation [21].

On average, models performed marginally better when using all 34 features instead of subsets of 10 features chosen by selectors. However, the differences were so minimal that no method proved to be universally superior. A conclusion can be drawn that feature selection is not a universal "must-have" when performing machine learning predictions, it is recommended to search for different and best solutions when using different models. Also, it is worth mentioning that the size of the subsets is a topic worth exploring.

Finally, it was observed that features such as expected goals - xg, expected goals against - xga, shot-creating actions - sca, and goal-creating actions - gca were consistently ranked as the most important by feature selectors. It is often stated that in football the emphasis should always be on attacking and keeping possession of the ball is crucial in winning a match [23] which corresponds with the aforementioned features. Therefore, while the feature selection method can influence model performance, the characteristics of the features themselves are an equally critical factor in determining predictive success. Further analysis of characteristics of match statistics is a step worth taking.

6.3. Impact of the time window

In accordance with the assessment of the machine learning models, it is observed that the choice of time window (e.g. train and test split) significantly affects the prediction accuracy of the outcomes.



X	2 seasons	1 season	10 matchweeks
RFC	0,6495	0,6733	0,4772
SVM	0,6691	0,7266	0,4545
KNN	0,6151	0,6266	0,3863
XGB	0,6862	0,6866	0,4772
AVG	0,6550	0,678275	0,4488

Figure 6.3. Accuracies of models for different splits of train and test data

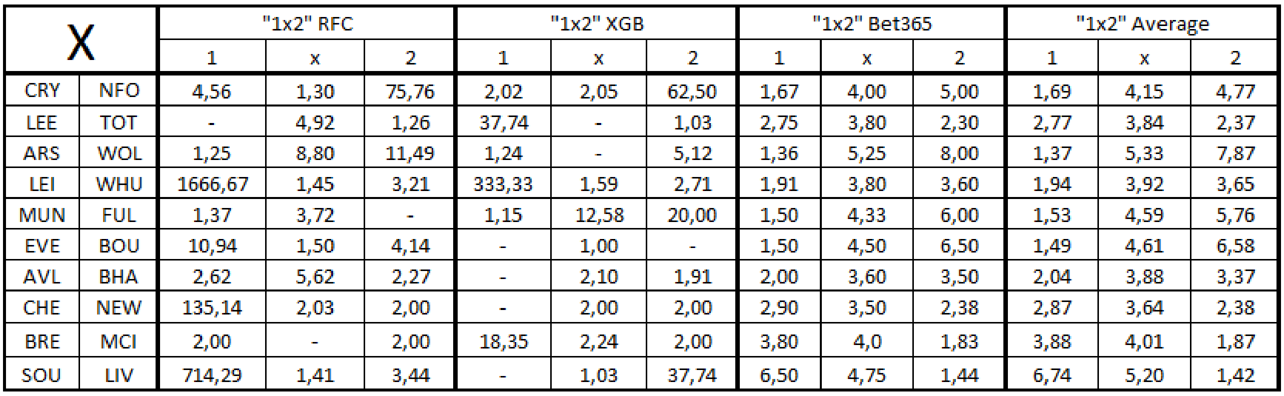
The highest average accuracy of the models was achieved with one season's worth of data. The results indicate that while the recent form is commonly acknowledged as a key factor in predicting match outcomes, it doesn't necessarily translate to better performance in machine learning. This could be due to the limited sample size provided by 10 matchweeks which further emphasises the importance of having a sufficiently large dataset to achieve reliable classification [21].

Following up on Chapter 5, the models performed quite poorly in predicting draws, specifically for the 10 matchweeks dataset where some models couldn't predict a single draw correctly. As highlighted by statistics [8] [24] draws are the least common outcome in football, thus leading to a conclusion that when one of the classes in a multiclass problem is much less common than others, it is essential for features to be detailed and vast to accurately capture instances of that class [23].

The fact that one season of data yielded the best results across the majority of the models suggests that a balance between historical data and recent form is necessary to achieve optimal predictive accuracy. Historical data can provide a good background and context but recent performance metrics show more information about the current state of a team [25] [26]. It also suggests the possibility that the optimal time window for the dataset might lie somewhere between 10 matchweeks and a full season.

6.4. Accuracy of "1x2" odds calculated with machine learning

The final section of this analysis dives into the accuracy of "1x2" odds.



X		"1x2" RFC			"1x2" XGB			"1x2" Bet365			"1x2" Average		
		1	x	2	1	x	2	1	x	2	1	x	2
CRY	NFO	4,56	1,30	75,76	2,02	2,05	62,50	1,67	4,00	5,00	1,69	4,15	4,77
LEE	TOT	-	4,92	1,26	37,74	-	1,03	2,75	3,80	2,30	2,77	3,84	2,37
ARS	WOL	1,25	8,80	11,49	1,24	-	5,12	1,36	5,25	8,00	1,37	5,33	7,87
LEI	WHU	1666,67	1,45	3,21	333,33	1,59	2,71	1,91	3,80	3,60	1,94	3,92	3,65
MUN	FUL	1,37	3,72	-	1,15	12,58	20,00	1,50	4,33	6,00	1,53	4,59	5,76
EVE	BOU	10,94	1,50	4,14	-	1,00	-	1,50	4,50	6,50	1,49	4,61	6,58
AVL	BHA	2,62	5,62	2,27	-	2,10	1,91	2,00	3,60	3,50	2,04	3,88	3,37
CHE	NEW	135,14	2,03	2,00	-	2,00	2,00	2,90	3,50	2,38	2,87	3,64	2,38
BRE	MCI	2,00	-	2,00	18,35	2,24	2,00	3,80	4,0	1,83	3,88	4,01	1,87
SOU	LIV	714,29	1,41	3,44	-	1,03	37,74	6,50	4,75	1,44	6,74	5,20	1,42

Figure 6.4. "1x2" odds calculated using machine learning models compared with real-life betting companies

The results of calculations based on machine learning predictions are juxtaposed against the odds given by a real-life betting company Bet365, as well as the average odds calculated using data from several betting companies, including: Blue Square, Gamebookers, Ladbrokes, Pinnacle, Stanleybet, and William Hill [27].

Upon analysis, it can be noted that machine learning models occasionally predict zero outcomes for certain classes in certain matches. Such instances hinder the calculation of a part of the "1x2" bet, as it results in division by 0. Furthermore, some calculations result in the generation of extremely high odds which are unrealistic but reflect on how the models "see" matches with strong favourites. Generally speaking the more even the match was the better the result of the "1x2" odds calculation. Unfortunately, even this statement has a flaw since matches that are too even also resulted in models ignoring certain classes thus leading to failure in calculating parts of the bet.

Despite the apparent shortcoming, the machine learning approach to calculating odds offers a promising avenue for exploration. The insights derived from these models can provide valuable support to bookies while establishing odds. Some of the calculations provided similar results to those offered by established betting companies suggesting that with enough data and proper selection of model's characteristics, models can capture intricacies of a football match [28]. Nonetheless, the process should be viewed as an adjunct rather than a replacement for traditional methods [29]. As such, continuous efforts should be invested to improve and optimize this process. In conclusion, machine learning offers a complementary tool for bookmakers and bettors alike to consider alongside traditional means of calculating odds. By incorporating a wider array of data sources, such as historical odds from multiple betting companies, more comprehensive and accurate predictions can potentially be achieved.

7. Summary

The objective of this thesis was to compare the performance of various machine learning models based on a specified dataset and to determine the most effective feature selection and time window for predictions. As the analysis unfolded, it was evident that the performance of each model varied significantly, and their effectiveness was contingent on the chosen features and the selected time window. Some models performed exceedingly well with specific features, while others were more versatile, indicating the complexity and adaptability of machine learning algorithms in predicting football match outcomes.

Another notable area of focus was the exploration of Explainable Artificial Intelligence (xAI). The value of interpretability in machine learning models cannot be overstated, particularly in complex domains such as football match prediction. The integration of xAI principles not only enhances transparency in the model's decision-making process but also provides a clear understanding of how different factors influence the prediction of match outcomes. This deepened understanding could contribute significantly to the field of sports analytics, leading to more informed and accurate predictions.

Subsequent chapters focused on a more nuanced analysis, delving deeper into aspects like feature importance, time window significance, and accuracy of "1x2" odds calculated with machine learning. The findings, while offering unique insights into the machinations of machine learning predictions, also underlined areas that could benefit from further research. For example, the high variability of predictions based on different splits of data suggests that ML models could be optimized further to handle recent data more effectively. Similarly, the calculation of "1x2" odds revealed the need for more robust algorithms that could handle edge cases better and avoid extreme values.

While this thesis offers insight into the world of machine learning, it also should be emphasised that it is a rapidly evolving field. The results of the experiments carried out for this work were based on implementations of simple machine-learning models. The reason those models were chosen was to focus on aspects like aforementioned feature selection, hyperparameters search, train/test split, and more, and highlight the differences that a proper establishment of those parameters has on the outcomes of calculations.

Recognizing the potential for continuous evolution and growth, several improvements are proposed:

1. Enlargement of the dataset to include more descriptive features, more entries
2. More data preprocessing, combining pairs of rows describing the same match into 1 row
3. Wider hyperparameters grids
4. Testing different sizes of subsets of features selected by RFE and Correlation and how they affect the accuracy
5. Inclusion of more sophisticated machine learning algorithms, perhaps even neural networks
6. More numerous simulations, due to limited computing power only a handful of simulations were carried out
7. Assuming different targets for models like the number of goals scored and conceded and determining the result of a match based on those.

8. Calculating different bets

Lastly, the contents of this thesis could be used to improve the bettor's chances of succeeding. Even though models on their own aren't capable of beating the bookies it is essential to understand that they can prove a helpful tool in noticing trends and statistical advantages that certain teams may have. The best predictors of football matches are the ones who find the proper balance between data analysis and applying their knowledge of the sport.

In conclusion, this thesis has accomplished its main objective - comparing different machine learning models and their effectiveness in predicting football match outcomes. However, it also opens the door for further exploration in this exciting intersection of machine learning and sports analytics. The integration of xAI and the proposed improvements set the stage for a more comprehensive, accurate, and transparent system of predicting football match outcomes.

Bibliography

[1] *Football*, https://en.wikipedia.org/wiki/Football, 18.06.2023

[2] *Betting and Gaming Act 1960*, https://en.wikipedia.org/wiki/Betting_and_Gaming_Act_1960, 18.06.2023

[3] *Charles Reep*, https://en.wikipedia.org/wiki/Charles_Reep, 18.06.2023

[4] *How Math and Data Science Made Liverpool the Best Team on the Planet*, https://medium.com/the-spekboom/how-math-and-data-science-made-liverpool-the-best-team-on-the-pla 18.06.2023

[5] A. Joseph, N.E. Fenton, M.Neil, Predicting football results using Bayesian nets and other machine learning techniques

[6] A. Majumdar, R. Bakirov, D. Hodges, S. Scott, T. Rees, Machine Learning for Understanding and Predicting Injuries in Football

[7] R. Beal, S. E. Middleton, T. J. Norman, S. D. Ramchurn, Combining Machine Learning and Human Experts to Predict Match Outcomes in Football: A Baseline Model

[8] *Premier League percentage of points*, https://www.soccerstats.com/table.asp?league=england&tid=1, 19.06.2023

[9] *2022-2023 Premier League Stats* https://fbref.com/en/comps/9/Premier-League-Stats, 19.06.2023

[10] F.A. Moura, L.E.B. Martins, S.A. Cunha, Analysis of football game-related statistics using multivariate techniques

[11] R.S. Mendes, L.C. Malacarne, C. Anteneodo - Statistics of football dynamics

[12] Premier League table search in google

[13] Provost, F., Fawcett, T., & Kohavi, R. (2012). The case against accuracy estimation for comparing induction algorithms. In Machine Learning (pp. 445-453). Springer.

[14] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160, 3-24.

[15] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

[16] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[17] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[18] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[19] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437.

[20] Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3(Mar), 1157–1182.

[21] Kukar, M., & Kononenko, I. (1998). Reliable Classifications with Machine Learning. Proceedings of the 13th European Meeting on Cybernetics and Systems Research, 2, 1095–1100.

[22] Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in

bioinformatics. Bioinformatics, 23(19), 2507–2517.

[23] Hughes, M., & Franks, I. (2005). Analysis of passing sequences, shots and goals in soccer. Journal of Sports Sciences, 23(5), 509–514.

[24] How common are draws in football, https://www.ukgamblingsites.com/sports-betting/football/how-common-are-draws-in-football/, 21.06.2023

[25] Anderson, C., & Sally, D. (2013). The Numbers Game: Why Everything You Know About Soccer Is Wrong. Penguin.

[26] Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., & Giannotti, F. (2019). PlayeRank: Data-driven performance evaluation and player ranking in soccer via a machine learning approach. ACM Transactions on Intelligent Systems and Technology (TIST), 10(5), 1-27. This study discusses the use of recent data in evaluating soccer players' performance and ranking them accordingly.

[27] Football-data.co.uk. (n.d.). Historical Football Betting Odds and Results. Retrieved from https://www.football-data.co.uk/englandm.php, 21.06.2023

[28] Hubacek, O., Zeileis, A., & Leitner, C. (2019). Overround in football betting markets. The Journal of Gambling Business and Economics, 12(2), 51-69.

[29] Davidson, B., Leung, D., Perdomo, O., & Wong, J. (2020). Prediction markets, mechanism design, and fair betting odds. Economic Theory, 69, 761–784.