UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Football Match Result Prediction Using Machine Learning Techniques

Diploma Thesis

Anastasiadis Georgios

Supervisor: Dimitrios Rafailidis

February 2023

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Football Match Result Prediction Using Machine Learning Techniques

Diploma Thesis

Anastasiadis Georgios

Supervisor: Dimitrios Rafailidis

February 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εύρεση Αποτελέσματος Ποδοσφαιρικού Αγώνα Με Την Χρήση Τεχνικών Μηχανικής Μάθησης

Διπλωματική Εργασία

Αναστασιάδης Γεώργιος

Επιβλέπων: Ραφαηλίδης Δημήτριος

Φεβρουάριος 2023

Approved by the Examination Committee:

Supervisor **Dimitrios Rafailidis**

Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Dimitrios Katsaros**

Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Michael Vasilakopoulos**

Professor, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

Firstly, I would like to thank Prof. Dimitrios Rafailidis, for his guidance and support. His expertise and insight shaped my research and helped me achieve my goals.

I would also like to thank Prof. Dimitrios Katsaros, and Prof. Michael Vasilakopoulos, for their time and effort in evaluating my paper.

Finally, I would like to thank my family and friends for their support during my academic years.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Anastasiadis Georgios

Diploma Thesis

Football Match Result Prediction Using Machine Learning Techniques

Anastasiadis Georgios

Abstract

In recent years, football match predictions have become increasingly popular. With the help of data and machine learning techniques, predictions have become more accurate. The availability of data and the rise of social media have also made it easier for fans and sports betting companies to access it. In this thesis, we developed machine learning methods that we trained using publicly available data for the 4 main European football leagues to predict the outcome of football matches. To investigate the performance of the classifiers, we considered 2 scenarios. In the first scenario, we tried to predict the match outcomes of the last 2 seasons for the 4 leagues. In the second scenario, we considered each season separately and tried to predict the game results of the second round of each season. To evaluate the results of the experiments, we used the following metrics: accuracy, precision, recall, and F1-score. We found that Random Forrest outperformed the other classifiers and produced better results in most cases.

Keywords:

football match result prediction, machine learning, neural networks, sports analytics

Διπλωματική Εργασία

Εύρεση Αποτελέσματος Ποδοσφαιρικού Αγώνα Με Την Χρήση Τεχνικών Μηχανικής Μάθησης

Αναστασιάδης Γεώργιος

Περίληψη

Τα τελευταία χρόνια, οι προβλέψεις των αποτελεσμάτων ποδοσφαιρικών αγώνων γίνονται όλο και πιο δημοφιλείς. Με την βοήθεια των δεδομένων και την χρήση τεχνικών μηχανίκής μάθησης οι προβλέψεις γίνονται ολοένα και πιο ακριβείς. Επιπρόσθετα, η διαθεσιμότητα των δεδομένων και η άνοδος των μέσων κοινωνικής δικτύωσης έχουν καταστήσει την προσβασιμότητα των οπαδών και των στοιχηματικών εταιρειών σε αυτά πιο εύκολη. Στην παρούσα διπλωματική εργασία, χρησιμοποιήσαμε διάφορες μεθόδους μηχανικής μάθησης τις οποίες εκπαιδεύσαμε με δεδομένα τα οποία είναι διαθέσιμα στο διαδίκτυο, ώστε να προβλέψουμε τα αποτελέσματα των ποδοσφαιρικών αγώνων των 4 μεγάλων Ευρωπαϊκών πρωταθλημάτων. Προκειμένου να τεστάρουμε την απόδοση των ταξινομητών, θεωρήσαμε 2 σενάρια. Στο πρώτο σενάριο, προσπαθήσαμε να προβλέψουμε τα αποτελέσματα των αγώνων από τις τελευταίες 2 σεζόν των 4 πρωταθλημάτων. Στο δεύτερο σενάριο, θεωρήσαμε κάθε σεζόν ξεχωριστά και προσπαθήσαμε να προβλέψουμε τα αποτελέσματα των αγώνων από τον 2ο γύρο της κάθε σεζόν. Για την αξιολόγηση των πειραμάτων χρησιμοποιήσαμε τις παρακάτω μετρικές αξιολόγησης: accuracy, precision, recall, F1-score. Από τα αποτελέσματα προκύπτει ότι το μοντέλο Random Forest λειτούργησε καλύτερα από τα υπόλοιπα στις περισσότερες περιπτώσεις.

Λέξεις-κλειδιά:

πρόβλεψη αποτελέσματος ποδοσφαιρικού αγώνα, μηχανική μάθηση, νευρωνικά δίκτυα, ανάλυση δεδομένων στον αθλητισμό

Table of contents

List of figures

List of tables

Abbreviations

e.g	for example
i.e	in other words
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
HomeTeamID	Unique ID of home team
AwayTeamID	Unique ID of away team
FTR	Full time result
league	Name of league
LSTM	Long short-term memory
AVGH	Average home win odds
AVGA	Average away win odds
AVGD	Average draw odds
HT_wins	Number of wins for the home team
AT_wins	Number of wins for the away team
HT_draws	Number of draws for the home team
AT_draws	Number of draws for the away team
HT_losses	Number of losses for the home team
AT_losses	Number of losses for the away team
HTGS	Number of goals scored by home team
HTGC	Number of goals conceded by away team
ATGS	Number of goals scored by home team
ATGC	Number of goals conceded by away team
HToverall	Difference between goals scored and conceded by home team
AToverall	Difference between goals scored and conceded by away team
l5_ravg_HTST	5 games rolling average of home team's shots on target

l5_ravg_ATST	5 games rolling average of away team's shots on target
l5_ravg_HTCR	5 games rolling average of home team's conversion rate
l5_ravg_ATCR	5 games rolling average of away team's conversion rate
l5_ravg_HTxG	5 games rolling average of home team's expected goals
l5_ravg_ATxG	5 games rolling average of away team's expected goals
l5_ravg_HTxpts	5 games rolling average of home team's expected points
l5_ravg_ATxpts	5 games rolling average of away team's expected points
l5_ravg_HTdeep	5 games rolling average of passes home team exchanged with an estimated 20 yards from goal
l5_ravg_ATdeep	5 games rolling average of passes away team exchanged with an estimated 20 yards from goal
l5_htdiff	5 games rolling average of difference between the goals home team scored and conceded
l5_atdiff	5 games rolling average of difference between the goals away team scored and conceded
l5_ravg_HTp	5 games rolling average of home team's points
avgHTP	Average of home team's points from fixture to fixture
avgATP	Average of away team's points from fixture to fixture
diff_points	Difference between points of home and away team
diff_ATT	Difference between attack rating of home and away team
diff_MID	Difference between midfield rating of home and away team
diff_DEF	Difference between defense rating of home and away team
diff_OVA	Difference between overall rating of home and away team

Chapter 1

Introduction

According to data creation statistics for 2022, 2.5 quintillion bytes of data are created every day [7]. With such volumes of data, more people have access to information that enables them to make better decisions. There is no wonder then that data is playing an increasingly important role in almost every industry. Among the areas thriving in the new era of data is sports analytics [8].

In sports analytics, statistics are inserted into mathematical models to predict the outcome of a particular play or game. The coach uses analytics to scout opponents and optimize their in-game instructions, while their front office uses it to prioritize player development. Analytics also plays a significant role off the field, offering fans insights into sports betting and fantasy sports [9].

One of the most crucial areas where sports analytics is used is player evaluation and scouting. Analysts use data on a player's past performance, physical attributes, and scouting reports to predict future performance and potential. This helps teams identify underrated players and make better draft, trade, and free-agency decisions. As described in the book ”Moneyball: The Art of Winning an Unfair Game” (by Michael Lewis), the use of analytics in baseball has enabled Oakland Athletics to compete with teams that have much higher budgets [10].

Another area that sports analytics can be utilized is player development. Using the data, clubs can measure improvements or deterioration in their players' physical, tactical, and technical performance. Individual plans can then be created to help players focus on the areas of their game that need more work [11].

It is also possible for fans to make use of sports analytics. They can find information that can give them an edge over their competitors. That may include information about past matches, player statistics, and current odds. Understanding team and player performance, and the odds of different outcomes, allows you to make more informed betting decisions [12]. They can also exploit this information to choose better drafts for their team in the fantasy league.

The main contributions of this thesis are summarized below:

- We focus on machine-learning approaches for predicting football match outcomes for the 4 major European leagues, namely Logistic Regression, Random Forest, Support Vector Machines, K-Nearest Neighbors, XGBoost, and Artificial Neural Networks.
- We explain the data used to create the attributes that we later trained the models with.
- We present two different test scenarios that we later used to evaluate the results of our experiments using the following metrics: accuracy, precision, recall, and F1- score.
- Based on our results, we propose the most promising machine learning algorithm.

The remainder of the thesis is structured as follows. In Chapter 2, we review the current literature. In Chapter 3, we discuss the various algorithms we use. In Chapter 4, we provide details of the datasets, scenarios, and metrics used to evaluate the model. Then a comparison of the different classifiers based on the results of the experiments is presented in Chapter 5. In the final chapter, we summarize the research and discuss how we can improve this.

Chapter 2

Related Work

When it comes to football match result prediction, Josip Hucaljuk and Alen Rakipović investigated the combination of features and classifiers that would be the most accurate in predicting the results of Champions League matches. They then compared their results with those of other articles and achieved better accuracy [13].

Enes Eryarsoy and Dursun Deler developed a model to predict football match outcomes and determine the factors influencing games in the Turkish Super League using data from the 2007 to 2017 seasons. The predictions were made in two levels: win\draw\loss and points\no points. They achieved their best accuracy with Random Forest after oversampling the minority class [14].

Jan Koszak and Szymon Glowania to better predict the results of the German Bundesliga games created a heterogeneous ensemble of classifiers. They used data about Bundesliga seasons 2010/2011 to 2019/2020 and achieved an accuracy of 56% using the last 17 fixtures of the final season as the test set [15].

Roman Nestoruk and Grzegorz Slowinski trained 3 machine-learning models with a dataset of 50, 100, and 200 games collected from the 5 most popular European leagues. Unlike the other works, they tried to predict the number of goals a team should score [16].

After some experimentation, the authors in [17] decided to use a four-layer neural network to predict binary football outcomes (e.g., over\under). During the training process, they used data from 12 different countries. Then they utilized Return on Investment as a metric to see if they could make a profit.

Yunfei Li and Yubin Hong proposed a method for predicting soccer match outcomes based on edge computing and machine learning technology. They used data from the Chinese Super

League for the 2008 to 2018 seasons. They figured out that increasing the number of nodes of the neural network improves the accuracy of the model [18].

Ekansh Tiwary, Prasanjit Sardar, and Sarika Jain developed an RNN with LSTMs to predict the outcomes of English Premier League football matches. They used data from the 2011 to 2018 seasons. According to their work, an RNN with LSTMs appears to have an advantage over simple neural networks and machine learning algorithms [19].

The authors in [20] attempted to predict the outcome of football matches using various machine learning algorithms based on the number of goals scored by a team. They used data about the seasons of the Mexican football league from 2012 to 2020. They tested the following two scenarios: results including draws and without draws. In the second scenario, they achieved better performance. Table 2.1 shows the models used in the current literature.

Linear Regression	[20]
Logistic Regression	[17], [18]
KNN	[13], [17], [18]
SVM	[14], [15], [17], [20]
Random Forest	[13], [14], [15], [17]
Boosting methods	[13], [15]
Artificial Neural Networks	[13], [14], [16], [17], [18], [19]
Naive Bayes	[13], [14], [17], [20]
Decision Trees	[20]
Others	[15]

Table 2.1: Algorithms used in the literature

Chapter 3

Examined Models

In this chapter, we will explain the different machine learning algorithms we trained to predict football game outcomes.

3.1 Logistic Regression

Despite the term ”regression”, Logistic Regression is a machine learning algorithm used for the solution of classification problems. One of the main advantages of this algorithm is that it can be used for both classification and class probability estimation as it is linked to logistic data distribution. It takes a linear combination of attributes and applies a non-linear sigmoidal function (logistic function) to them [21]. There are 3 types of Logistic Regression [22]:

- Binary Logistic Regression: this is a binary classification problem (e.g whether an e-mail is spam or not)
- Multinomial Logistic Regression: the target label can have three or more non-ordinal classes (e.g the prediction of a football game result into a home win, draw, and away win)
- Ordinal Logistic Regression: the target label can have three or more classes in a defined order (e.g pain scale from 0 to 10).

The hyperparameters that we have adjusted to improve the accuracy of our models are the following: ’solver’, ’C’, ’class_weight’, ’fit_intercept’, and ’multi_class’. To better optimize these hyperparameters, we performed GridSearchCV. Their function is listed below [23]:

- C: the inverse of the regularization strength. It can only take positive float values. Higher values of C instruct the model to give more weight to the data as they constitute a good representation of real-world data.
- Solver: the algorithm used to solve the optimization problem. Since our problem is a multiclassification one, the solver can take the following values: ’newton-cg’, ’sag’, ’saga’, and ’lbfgs’.
- Class_weight: specifies the weights for each class in the format {class_label: weight}. Unless specified, all classes have a weight of one.
- Fit_intercept: if set to False, then y_intercept is set to 0. Otherwise, the y_intercept is determined by the line of best fit.
- Multi_class: specifies the problem to be solved. The values that multi_class can take are the following {‘auto’, ‘ovr’, ‘multinomial’}.

3.2 Random Forest

Random Forest is an ensemble [24] algorithm of decision trees first introduced by Breiman [25]. The basic idea behind Random Forest is that many uncorrelated models combined into a group perform better than individual models. Random Forest is a special bagging technique that creates uncorrelated trees to achieve greater variance reduction. This is done by considering only a subset of the available predictors for each split [26]. Figure 3.1 shows the function of the Random Forest classifier.

Due to the large number of hyperparameters to be tuned and the wide range of values they can take, we first implemented RandomisedSearchCV to narrow down the options and then GridSearchCV. The hyperparameters we tuned to improve the performance of our model are the following: ’criterion’, ’n_estimators’, ’max_features’, ’max_depth’, ’min_samples_split’, ’min_samples_leaf’, and ’bootstrap’. The function of each tuned hyperparameter is listed below [23]:

- N_estimators: indicates the number of decision trees in the forest.
- Criterion: the function used to determine the quality of the splits.

- Max_depth: the maximum depth the decision trees can reach. If we do not set a specific value, then the trees are expanded until the purity is reached.

- Max_features: specifies the number of attributes to be used for a node split.

- Min_samples_split: determines the minimum number of instances to split a node.

- Min_samples_leaf: declares the minimum number of samples that need to exist at a leaf node.

- Bootstrap: if set to True, bootstrap samples are used to build the tree. Otherwise, the whole dataset is used for the construction of each tree.

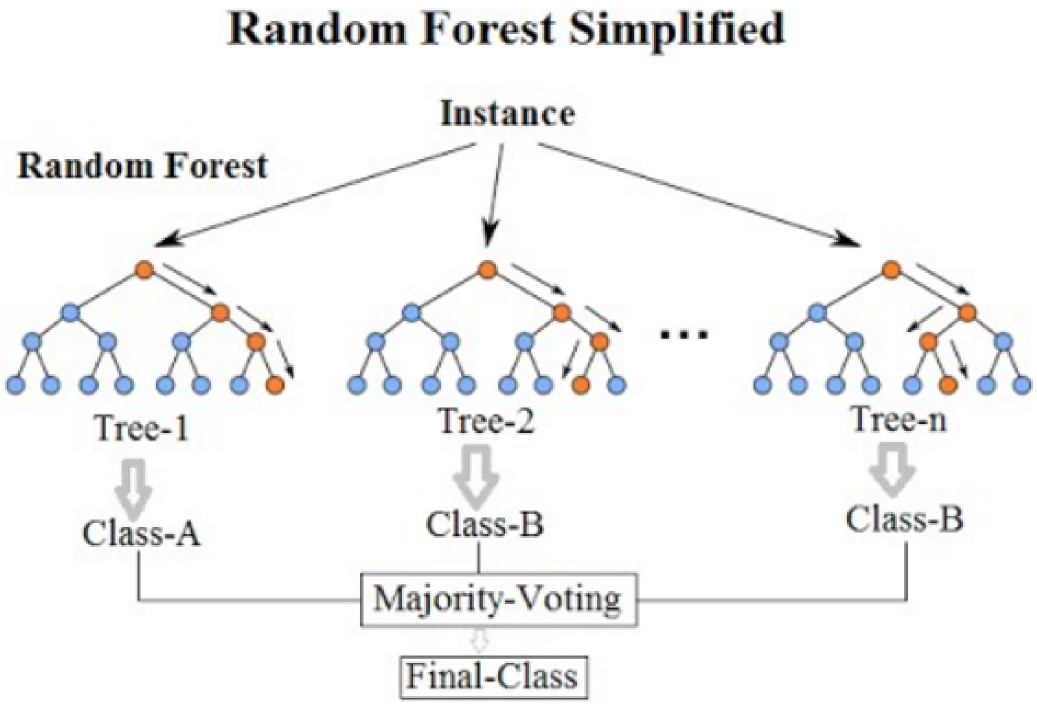


Figure 3.1: Random forest. (2022, December 24). In Wikipedia [1]

3.3 K-Nearest Neighbors

The KNN algorithm is a nonparametric supervised learning classifier that uses proximity to make classifications or predictions about the clustering of a single data point. For classification problems, a class label is assigned based on majority voting, i.e., the label that is most frequently displayed around a given data point is used [27]. Figure 3.2 shows the KNN procedure.

To improve the accuracy of the model, we tuned the following hyperparameters: ’leaf_size’, ’n_neigbors’, ’p’, and ’weights’, using GridSearchCV. The function of each tuned hyperparameter is listed below [23]:

- Leaf_size: affects the speed of construction and querying, as well as the memory required to store the tree.
- N_neighbors: number of neighbors to consider.
- P: the formula used to calculate the distance. For p=1 the Manhattan distance is used and for p=2 the Euclidean distance.
- Weights: if ’weights’ is set to uniform, all points in each neighborhood have the same weight. Otherwise, if ’weights’ is equal to ’distance’, closer neighbors of a query point have a higher impact than more distant neighbors.

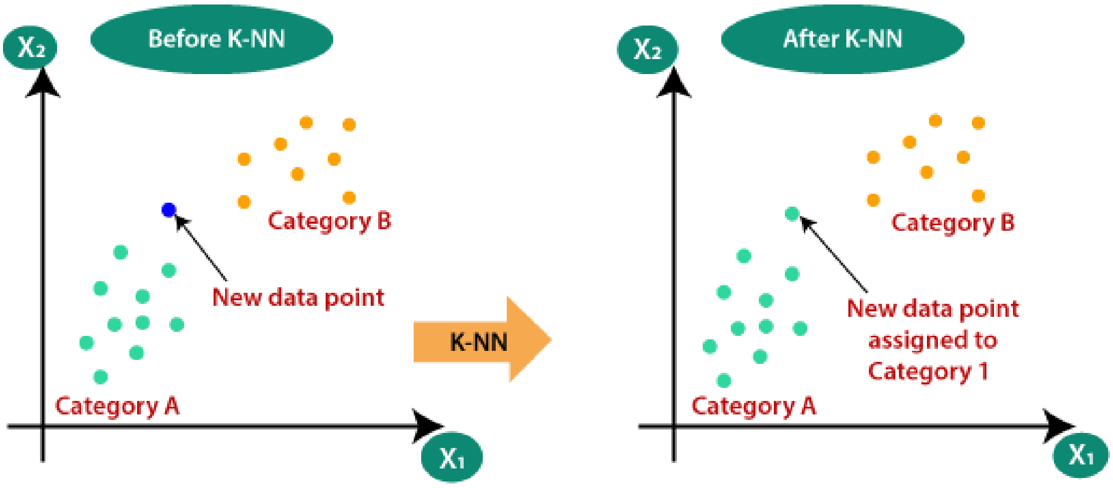


Figure 3.2: KNN procedure [2]

3.4 Support Vector Machines

Support Vector Machines [28] are supervised machine learning algorithms commonly used for classification tasks. The goal of SVM is to find the maximum margin between hyperplanes. When dealing with data that are not linearly separable, SVMs use the kernel trick.

The kernel is a function that maps the non-linearly separable data into a higher-dimensional space where it is easy to find a linear separating hyperplane [29]. Figure 3.3 shows an example of SVM using the kernel trick. To better improve the performance of our model we implemented GridSearchCV. The hyperparameters we tuned and their function are listed below [23]:

- Kernel: the kernel type to be used. We test the model for three kernels, namely 'rbf', 'sigmoid', and 'linear'.
- C: tells the SVM optimizer how much you want to avoid misclassifying each training sample. Larger values of C lead to smaller margins. Conversely, higher values of C lead to a bigger margin.
- Gamma: kernel coefficient for non-linear hyperplanes. High values of gamma may lead to overfitting.

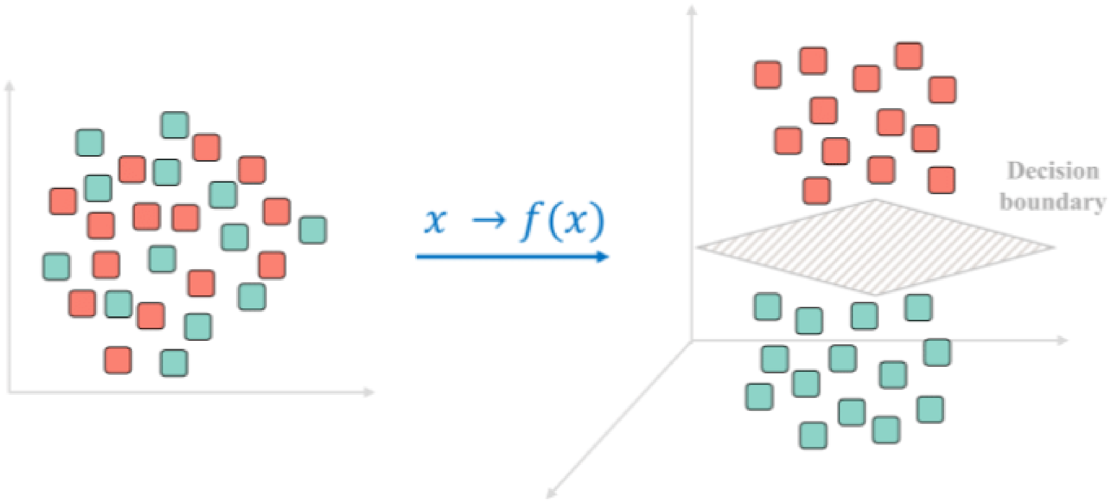


Figure 3.3: SVM example for non-linearly separable data with kernel trick [3]

3.5 Artificial Neural Networks

An artificial neural network is a computing system that attempts to mimic the function of the human brain. An ANN consists of a large number of nodes connected by weighted links. These nodes are also referred to as 'neurons'. The output of each node depends only on the information available locally at the node, either stored internally or arriving through the weighted links. Each unit receives inputs from many other nodes and sends its output to even more nodes [30]. The basic architecture of an artificial neural network consists of 3 layers:

- Input layer: data is accepted from this layer and passed to the rest of the network.
- Hidden layer: between the input and the output layer there can be 1 or more hidden layers. This is where the calculations are performed.
- Output layer: the final result estimation.

Figure 3.4 shows the architecture of a simple neural network.

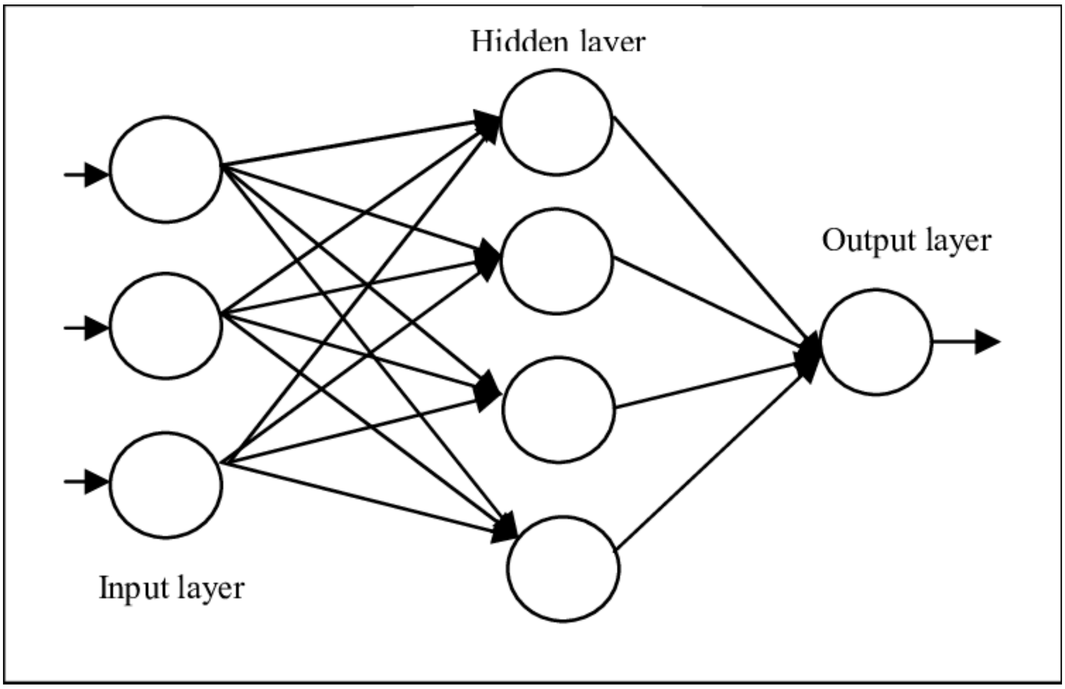


Figure 3.4: Feed-forward Artificial Neural Network [4]

The hyperparameters of ANN, which we adjusted to improve the performance of the model, and their function are listed below:

- Hidden layer: the number of hidden layers of the ANN
- Number of neurons: the number of neurons each layer has.
- Activation function: determines whether a neuron should be activated or not.
- Batch size: the number of training examples being used per iteration.
- Epochs: cycles in which all the training data is used to train the neural network.

- Learning_rate: specifies how our network adjusts its weights with respect to the loss gradient descent.

3.6 XGBoost

XGBoost is an ensemble algorithm based on decision trees that uses a gradient boosting framework. XGBoost was first developed by Tianqy Chen [31] and now is part of a larger collection of open-source libraries developed by the Distributed Machine Learning Community (DMLC). The purpose of its creation was to enhance the boosted tree algorithms' performance and computational speed. It can perform the three main gradient boosting techniques: Gradient Boosting, Stochastic Gradient Boosting, and Regularized Gradient Boosting. This library stands out from others for its ability to add and tune regularization parameters. The algorithm reduces the computing time very effectively and offers optimal use of memory resources. It is sparse aware or can take care of missing values, supports parallel structures in tree construction, and has the unique quality of boosting on added data already present in the trained model (Continued Training) [32, 33].

Due to the large number of hyperparameters to be tuned and the wide range of values they can take, we first implemented RandomisedSearchCV to narrow down the options and then GridSearchCV. The hyperparameters and their function are listed below [34]:

- Subsample: subsample ratio of training instances. Setting it to 0.5 means that XGBoost will randomly sample half of the training data before trees grow, and this prevents overfitting. Subsampling occurs once in each boosting iteration.

- Reg_lambda: L2 regularization parameter on leaf weights.

- N_estimators: indicates the number of decision trees.

- Max_depth: the maximum depth that the decision trees can reach. If the value is 0, there is no limit to the depth. The higher the number, the more likely the model is to overfit.

- Learning_rate: after each boosting step, we can get the weights of new features directly. Learning_rate decreases the feature weights to make the boosting process more conservative.

- Gamma: a minimum reduction in loss required for a new tree split to be created.
- Colsample_bytree: the subsample ratio of columns when building each tree. The subsampling is done once for each tree constructed.

Chapter 4

Evaluation Protocol

In this chapter, we will describe the datasets, scenarios, and evaluation metrics we used to conduct the project. In Section 4.1, we inform the readers about the datasets we utilized to create the features we later used to train the models. In Section 4.2, we explain the two scenarios that we tested. In Section 4.3, we describe the features that we have created and the method we used to choose the most relevant ones. In the last section, we explain the metrics we used to evaluate the different models.

4.1 Data Sets

We used three publicly available datasets to develop the features we later used to train our models. The first dataset contained information on game statistics and betting odds. The second dataset contained information on team dynamics for each league, and the third dataset contained information on advanced football statistics.

4.1.1 Match History Data

The data relating to match statistics were provided to us by football-data [35]. For our research, we looked at seasons from 2014/2015 to 2019/2020, covering the Premier League, Bundesliga, Serie A, and La Liga. The data from this website was available in CSV format. The Premier League, Serie A, and La Liga consist of twenty teams that play with each other on home and away grounds. Each season consists of 380 matches, so the datasets for these three leagues consist of 2080 rows. Unlike the other leagues, the Bundesliga consists of 18 teams, which means that each season consists of 306 games. From the dataset related to the

Bundesliga, we have obtained 1836 rows. Table 4.1 contains the features of the first dataset. A sample of this dataset can be found in Appendix A.

League Division	Match Date (dd/mm/yy)	Referee
Home Team	Away Team	Full Time Home Team Goals
Full Time Away Team Goals	Full Time Result (H=Home Win, A=Away Win, D=Draws)	Half Time Home Team Goals
Half Time Away Team Goals	Half Time Result	Home Team Shots
Away Team Shots	Home Team Shots on Target	Away Team Shots on Target
Home Team Corners	Away Team Corners	Home Team Fouls Committed
Away Team Fouls Committed	Home Team Yellow Cards	Away Team Yellow Cards
Home Team Red Cards	Away Team Red Cards	Bet365 home win odds
Bet365 away win odds	Bet365 draw odds	Bet&Win home win odds
Bet&Win away win odds	Bet&Win draw odds	Interwetten home win odds
Interwetten away win odds	Interwetten draw odds	Ladbrokes home win odds
Ladbrokes away win odds	LadBrokes draw odds	Pinnacle home win odds
Pinnacle away win odds	Pinnacle draw odds	William Hill home win odds
William Hill away win odds	William Hill draw odds	Stam James home win odds
Stam James away win odds	Stam James draw odds	VC Bet home win odds
VC Bet away win odds	VC bet draw odds	Betbrain max over 2.5 goals
Betbrain max under 2.5 goals	Betbrain avg over 2.5 goals	Betbrain avg under 2.5 goals

Table 4.1: Football-data stats

4.1.2 FIFA Index Team Rating

FIFA Index [36] provided us with data related to team dynamics. To obtain the data from that website, we had to manually create CSV files for each season of the four leagues. The CSV files for Premier League, Serie A, and La Liga consist of 20 rows and 5 columns. Unlike the other leagues, the CSV files for the Bundesliga comprised of 18 rows and 5 columns due to the smaller number of teams. These 5 columns are team name, attack, defense, midfield, and overall team rating. An example of this dataset can be found in Appendix A.

4.1.3 Understats Data

The advanced match statistics for each team were obtained from Kaggle [37]. This dataset contains statistics for Premier League, Serie A, Bundesliga, La Liga, Ligue 1, and Russian EPL from 2014/2015 to 2019/2020. The data from this website were available to us in a CSV format, consisting of 24580 rows and 16 columns. Table 4.2 contains the features of the Understats dataset. An example of this dataset can be found in Appendix A.

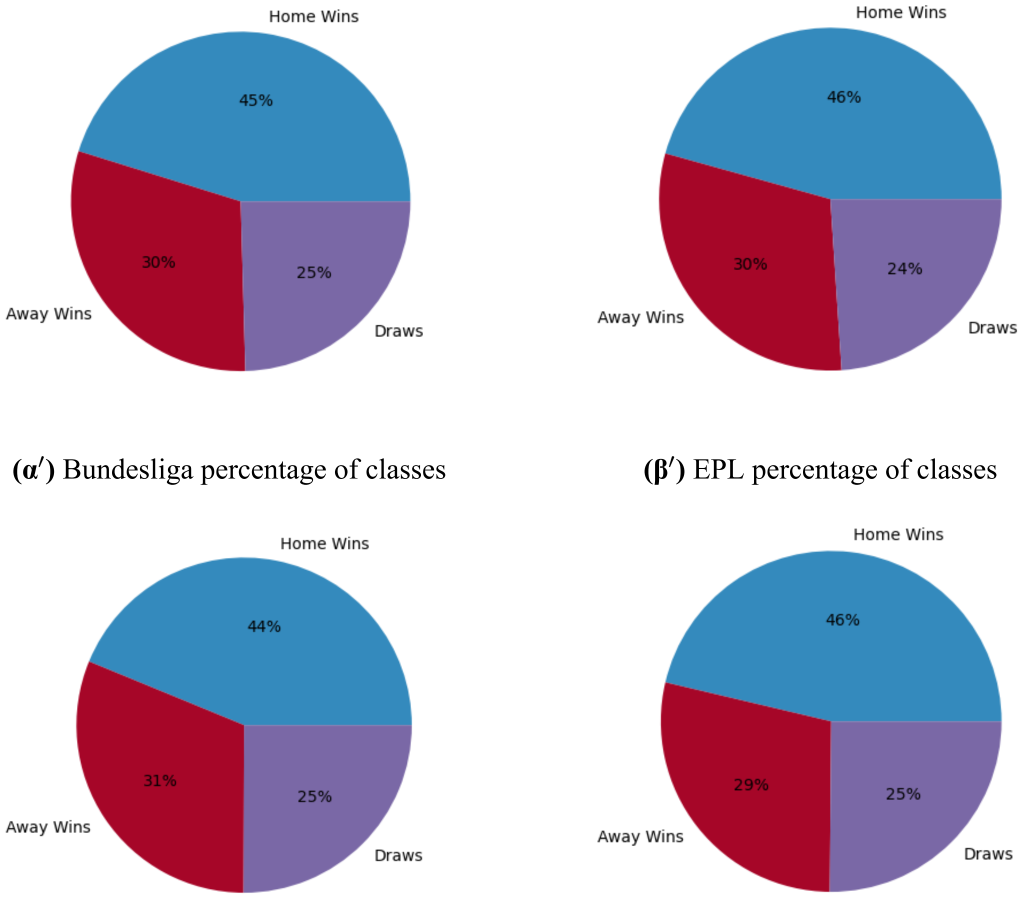
Year	League
Expected goals	Difference between actual and expected goals
Expected goals without penalties and own goals	Expected goals against
Difference between actual goal conceded and expected goals against	Expected goals against without penalties and own goals
Power of pressure	Power of opponents pressure
Passes completed within an estimated 20 yards of goal	Opponent passes completed within an estimated 20 yards of goal
Expected Points	Difference between actual and expected points
Team	Date

Table 4.2: Understats data

4.2 Testing Scenarios

To study the classifiers’ performance we considered 2 scenarios. In the first scenario, we are attempting to predict the game results of the last 2 seasons for the aforementioned leagues. In order to do that, we performed a simple train test split on the dataset, using the first 4 seasons as the training set and the last 2 as the test set. Figure 4.1 depicts the ratio of each class in the four datasets.

In the second scenario, we decided to consider each season separately. The Premier League, Serie A, and La Liga seasons consist of 2 rounds of 19 fixtures each. Unlike the other 3 leagues, the Bundesliga seasons consist of 2 rounds of 17 fixtures each due to the smaller number of teams. In this scenario, we use the first round of each season as training data and the second round as test data. Figure 4.2 shows the ratio of each class for the 4 leagues in season 2018/2019.



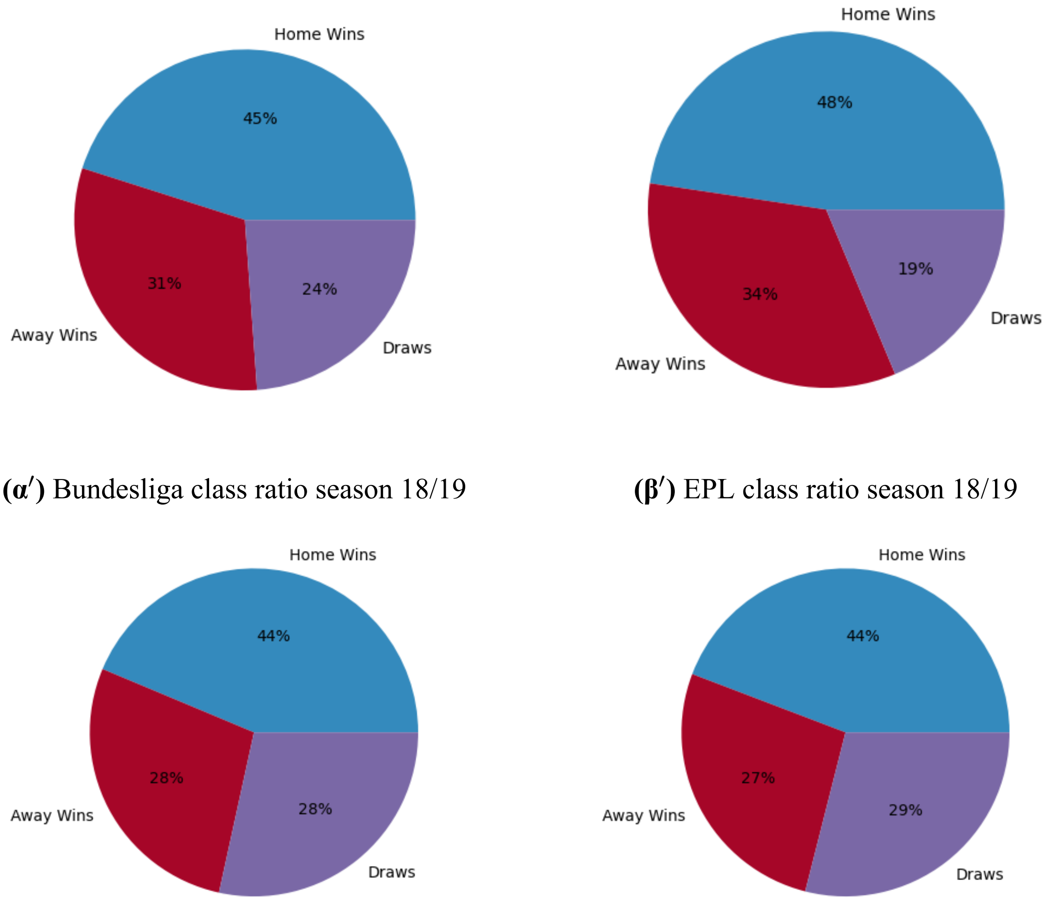
(α′) Bundesliga percentage of classes

(β′) EPL percentage of classes

(γ′) Serie A percentage of classes

(δ′) La Liga percentage of classes

Figure 4.1: Scenario 1 classes distribution



(α′) Bundesliga class ratio season 18/19

(β′) EPL class ratio season 18/19

(γ′) Serie A class ratio season 18/19

(δ′) La Liga class ratio season 18/19

Figure 4.2: Scenario 2 classes distribution in season 2018/2019

4.3 Data Preprocessing

After data collection, the data were reviewed to resolve any issues that might exist. Because we didn't want to delete any rows from the dataset that contain null values we had to manually retrieve the missing values using the websites Flashcore and Understats. We then proceeded to the creation of new features.

4.3.1 Feature Creation

A distinction can be made between game-related and external features [38]. Game-related features are known to us after the game is over. Such features include shots on target by both teams, fouls committed by both teams, etc. In order to use these features to predict the outcome of a football match, we first had to transform these data. The method we used to do this was to calculate the rolling average of these features over a five-game period. For each team, the rolling average value of the following attributes was computed: the number of goals scored and conceded, shots on target, conversion rate, the difference between goals scored and conceded, expected goals, expected points, power of pressure, and the number of passes exchanged within an estimated 20 yards of goals.

External features are known in advance of the upcoming game. These features could be the team's value, the team's points, etc. In order to make better use of the external features of the FIFA Index [36] dataset, we decided to create a new feature, namely the subtraction of home and away team ratings. We also decided to combine the team points into a single feature, resulting from the subtraction of the home and away team points. Furthermore, we computed the average of points a team earns from fixture to fixture. As for the betting odds, we have calculated the average between the prices of the betting providers. Finally, we calculated the number of wins, losses, and draws for every team before the beginning of the next game. Table 4.3 shows the features that we created using the three datasets. An example of this dataset can be found in Appendix B.

Home Team ID	Away Team ID	Average Home Win Odds
Average Draw Odds	Average Away Win Odds	Each team's wins so far
Each team draws so far	Each team's losses so far	Rolling avg. of each team's shots on target
Rolling avg. of each team's conversion rate	Rolling avg. each team's expected points	Rolling avg. of passes that each team exchanged with an estimated 20 yards of goal
Rolling avg. of each team's power of pressure	Rolling avg of the difference between each team's goals scored and conceded	Average team's points from fixture to fixture
Rolling avg. of each team's points	Difference between home and away team's points	Difference between home and away team attack rating
Difference between home and away team midfield rating	Difference between home and away team defense rating	Difference between home and away team overall rating

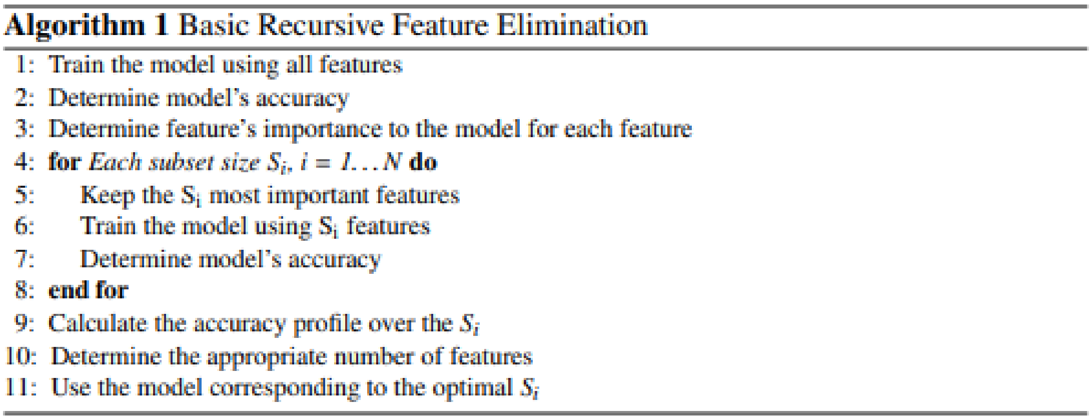
Table 4.3: Features

4.3.2 Feature Selection

Since we ended up with a total of 36 features, we performed feature selection to find the most relevant ones. The three main advantages of feature selection are [39]:

- Decreases over-fitting.
- Improves accuracy.
- Reduces training time.

The method we proceed with is Recursive Feature Elimination (RFE). Based on an estimator of our choice (e.g Decision Trees) that assigns weights to the features, the objective of RFE is to select features by recursively considering smaller and smaller sets of attributes [23]. Figure 4.3 breaks down the process of recursive feature elimination step by step.



Algorithm 1 Basic Recursive Feature Elimination

```
Train the model using all features
Determine model's accuracy
Determine feature's importance to the model for each feature
for Each subset size $S_i$, $i = 1 \ldots N$ do
    Keep the $S_i$ most important features
    Train the model using $S_i$ features
    Determine model's accuracy
end for
Calculate the accuracy profile over the $S_i$
Determine the appropriate number of features
Use the model corresponding to the optimal $S_i$
```

Figure 4.3: RFE (Joanna Goscik, Tomasz Łukaszuk) [5]

An interesting fact is that for the various leagues and seasons, the features of great importance were not the same. Furthermore, different estimators for the same scenario also provided us with different features of great importance. Table 4.4 shows the different features that were selected after implementing RFE using Logistic Regression as the estimator.

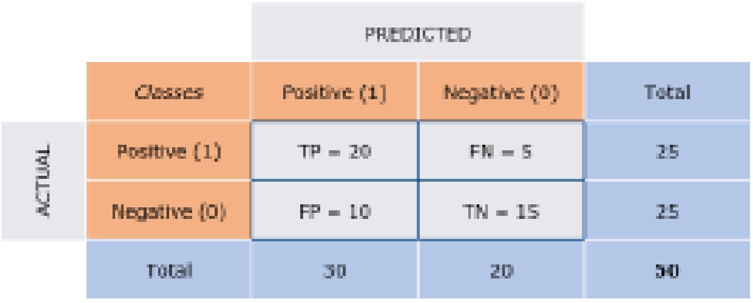
4.4 Evaluation Metrics

The metrics that we used to assess the performance of the various models are precision, recall, F1-score, and accuracy [6]. To better describe them we first are going to break down the main elements of the confusion matrix using Figure 4.4. The components of the confusion matrix are:

- True Positive (TP): Observation is positive and the prediction is positive.
- True Negative (TN): Observation is negative and the prediction is negative.
- False Positive (FP): Observation is negative, but the prediction is positive.
- False Negative (FN): Observation is positive, but the prediction is negative.

RFE using Logistic Regression Scenario 1			
Bundesliga	Premier League	La Liga	Serie A
AVGH	AVGH	AVGH	AVGH
AVGD	AVGD	AVGD	AVGD
AVGA	AVGA	AVGA	AVGA
HT_draws	HT_draws	HT_wins	l5_ravg_ATCR
AT_draws	AT_draws	AT_wins	avgHTP
HT_losses	HT_losses	HT_losses	avgHTP
AT_losses	AT_losses	AT_losses	
l5_ravg_HTST	l5_ravg_HTST	l5_ravg_ATST	
l5_ravg_HTCR	l5_ravg_HTCR	l5_ravg_HTCR	
l5_ravg_ATCR	l5_ravg_ATCR	l5_ravg_ATCR	
l5_ravg_HTxG	l5_ravg_HTxG	l5_ravg_HTxG	
l5_ravg_ATxG	l5_ravg_ATxG	l5_ravg_ATxG	
l5_ravg_HTxpts	l5_ravg_HTxpts	l5_ravg_HTxpts	
l5_ravg_ATxpts	l5_ravg_ATxpts	l5_htdiff	
l5_ravg_HTdeep	l5_ravg_HTdeep	l5_atdiff	
l5_htdiff	l5_atdiff	avgHTP	
avgHTP	avgHTP	avgATP	
avgATP	avgATP	l5_ravg_HTp	
diff_points	diff_points	diff_points	
	diff_MID	diff_MID	
		diff_OVA	

Table 4.4: Important features after RFE



		PREDICTED		
	Classes	Positive (1)	Negative (0)	Total
ACTUAL	Positive (1)	TP = 20	FN = 5	25
	Negative (0)	FP = 10	TN = 15	25
	Total	30	20	50

Figure 4.4: Two-Class Confusion Matrix [6]

Based on the aforementioned elements, the definitions of the metrics we used for the experiments are:

- Precision: the fraction of the True Positive samples divided by the total number of positive predicted samples. Precision can be calculated by the following equation:

$$Precision = \frac{TP}{TP + FP}$$

- Recall: the ratio of True Positive specimens divided by the actual number of positive specimens. The formula for calculating the recall is:

$$Recall = \frac{TP}{TP + FN}$$

- Accuracy: the amount of correct predictions divided by the total number of samples. The equation below defines the accuracy of the model.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

- F1-score: describes the harmonic mean of recall and precision. The formula for calculating F1-score is:

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Chapter 5

Experiments

The main focus of our research was to investigate how the different classifiers perform in predicting soccer match results. As we can see from Table 5.1, the selection of the best classifier varies depending on the metric we use to evaluate the model.

If we want to maximize the accuracy of the model, the best algorithm based on the results of our experiments is Random Forest. Random Forest outperformed the other classifiers and had the best accuracy in 12 out of 28 cases. The highest accuracy we could get was 0.57 for the English Premier League for scenario 1 and 0.63 for the English Premier League season 2016/2017 for scenario 2.

For the other metrics, the best model depends on the outcome we want to predict. In terms of predicting home wins, XGBoost performed the best regarding precision, outperforming the other classifiers 10 out of 28 times. In terms of recall, SVM performed the best. It achieved better results than the various classifiers in 18 out of 28 cases. Random Forest was the best concerning F1-score in 13 out of 28 cases. In general, all the classifiers had zero problems predicting the home wins, a result we anticipated because it was the majority class.

As for the draw, the models had difficulty predicting it accurately. That is something we expected because it is the minority class in a highly imbalanced dataset. According to a study published in the Journal of Quantitative Analysis in Sports, draws in soccer are generally less than 20% of all matches [40]. Consequently, the quantity of data for model training is limited, which can lead to poor model performance. XGBoost is the best classifier concerning our experiments, as it outperforms the other classifiers on all metrics.

For the away victories, SVM performed better regarding model precision. It performed better than the other classifiers in 13 out of 28 cases. Random Forest was the best classifier concerning recall and F1- score. Compared to the various classifiers, it performed better on 12 and 9 occasions, respectively. Tables 5.2-5.29 show the results of our experiments.

Generally, our model did a great job predicting home and away victories but had problems predicting draws. That makes perfect sense when we consider that ties are also difficult for humans to predict. Players' motivation, game style, referee decisions and other factors can influence the result of a draw. Random Forest seems to outperform the various classifiers and is probably the best method to choose concerning our results, as it had the best performance in predicting the outcomes of a football game in both test scenarios. Of the various leagues, we obtained the best results for the EPL although it is known as the most unpredictable football league.

As for the scenarios, both have their advantages and disadvantages. If we choose the first scenario, we have more data to train the models, but the drawback is that the dynamics of each team change from year to year. Also, in the second scenario, we have better results in some cases, but it depends a lot on the number of home and away wins. So if the number of draws in the second round is high, the models will probably not perform well.

Finally, it is worth noting that unlike the other studies, ANN did not give us the best results. That probably happened because we do not have much data to train the neural network. That is one of the main disadvantages of neural networks over other machine learning algorithms. Neural networks need at least thousands, if not millions, of labeled samples to produce good results. Although neural networks can sometimes perform well when trained with a small amount of data, most of the time they do not. In such cases, a simple machine-learning algorithm is better suited [41].

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	25%	7%	14%	25%
	D	**29%**	18%	14%	
	A	18%	18%	25%	
Random Forest	H	29%	14%	**46%**	**43%**
	D	18%	3%	0%	
	A	3%	**43%**	**32%**	
SVM	H	10%	**64%**	25%	32%
	D	14%	3%	0%	
	A	**46%**	7%	18%	
KNN	H	7%	0%	7%	10%
	D	3%	25%	18%	
	A	29%	21%	10%	
XGBoost	H	**36%**	3%	14%	18%
	D	**29%**	**50%**	**54%**	
	A	10%	14%	18%	
ANN	H	14%	21%	39%	25%
	D	7%	3%	7%	
	A	18%	18%	7%	

Table 5.1: Ratio of the number of cases in which a particular classifier performed the best with respect to a particular metric.

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.58	0.84	**0.69**	**0.56**
	D	**0.50**	0.01	0.02	
	A	0.49	0.52	0.50	
Random Forest	H	**0.60**	0.79	0.68	**0.56**
	D	0.00	0.00	0.00	
	A	0.47	**0.62**	0.53	
SVM	H	0.55	**0.91**	0.68	**0.56**
	D	0.00	0.00	0.00	
	A	**0.61**	0.41	0.49	
KNN	H	0.56	0.74	0.64	0.51
	D	0.19	**0.02**	**0.04**	
	A	0.43	0.48	0.45	
XGBoost	H	0.58	0.81	0.68	**0.56**
	D	0.00	0.00	0.00	
	A	0.52	0.59	**0.55**	
ANN	H	0.59	0.85	**0.69**	**0.56**
	D	0.00	0.00	0.00	
	A	0.48	0.51	0.49	

Table 5.2: Bundesliga scenario 1 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.57	0.85	0.68	0.56
	D	0.20	0.01	0.01	
	A	0.52	0.53	0.52	
Random Forest	H	0.57	0.87	**0.69**	**0.57**
	D	0.00	0.00	0.00	
	A	**0.57**	**0.57**	**0.57**	
SVM	H	0.56	**0.88**	0.68	0.56
	D	0.00	0.00	0.00	
	A	**0.57**	0.50	0.53	
KNN	H	0.57	0.81	0.67	0.55
	D	0.14	**0.02**	0.03	
	A	0.52	0.55	0.54	
XGBoost	H	**0.58**	0.85	**0.69**	**0.57**
	D	**0.38**	**0.02**	**0.04**	
	A	0.55	0.56	0.56	
ANN	H	0.57	0.85	0.68	**0.57**
	D	0.00	0.00	0.00	
	A	0.55	**0.57**	0.56	

Table 5.3: Premier League scenario 1 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.57**	0.79	0.66	0.53
	D	0.43	0.04	0.07	
	A	0.45	**0.51**	**0.48**	
Random Forest	H	**0.57**	0.81	0.67	0.53
	D	0.00	0.00	0.00	
	A	0.44	0.50	0.47	
SVM	H	0.55	0.84	0.66	0.52
	D	**0.57**	0.03	0.05	
	A	0.43	0.40	0.42	
KNN	H	0.55	0.53	0.54	0.44
	D	0.25	**0.18**	**0.21**	
	A	0.39	**0.51**	0.44	
XGBoost	H	**0.57**	0.80	0.02	0.52
	D	0.29	0.01	0.02	
	A	0.44	0.49	0.47	
ANN	H	**0.57**	**0.86**	**0.69**	**0.54**
	D	0.00	0.00	0.00	
	A	**0.47**	0.47	0.47	

Table 5.4: La Liga scenario 1 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.55	0.86	0.67	**0.56**
	D	0.00	0.00	0.00	
	A	**0.58**	0.62	0.60	
Random Forest	H	**0.59**	0.75	0.59	0.55
	D	0.00	0.00	0.00	
	A	0.49	**0.75**	**0.66**	
SVM	H	0.56	**0.86**	**0.68**	**0.56**
	D	**0.50**	0.01	0.02	
	A	0.55	0.61	0.58	
KNN	H	0.57	0.76	0.65	0.54
	D	0.37	**0.17**	**0.23**	
	A	0.55	0.55	0.55	
XGBoost	H	0.58	0.80	0.67	0.55
	D	0.39	0.04	0.07	
	A	0.51	0.65	0.57	
ANN	H	0.57	0.83	0.67	**0.56**
	D	0.00	0.00	0.00	
	A	0.53	0.66	0.59	

Table 5.5: Serie A scenario 1 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.54	0.74	0.62	0.48
	D	0.18	0.05	0.08	
	A	0.42	0.42	0.42	
Random Forest	H	0.54	0.70	0.61	0.48
	D	0.25	0.03	0.05	
	A	0.40	**0.53**	**0.46**	
SVM	H	0.52	0.92	0.66	0.52
	D	0.00	0.00	0.00	
	A	0.52	0.28	0.36	
KNN	H	0.50	0.69	0.58	0.45
	D	0.16	0.10	0.12	
	A	**0.54**	0.35	0.42	
XGBoost	H	**0.55**	0.66	0.60	0.48
	D	**0.32**	**0.21**	**0.25**	
	A	0.44	0.42	0.43	
ANN	H	0.54	**0.93**	**0.69**	**0.54**
	D	0.00	0.00	0.00	
	A	**0.54**	0.35	0.42	

Table 5.6: Bundesliga season 14/15 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.47	**0.92**	0.63	0.49
	D	0.00	0.00	0.00	
	A	**0.62**	0.22	0.32	
Random Forest	H	0.53	0.70	0.60	**0.52**
	D	**0.50**	0.05	0.10	
	A	0.49	**0.63**	**0.60**	
SVM	H	**0.55**	0.78	**0.64**	**0.52**
	D	0.00	0.00	0.00	
	A	0.48	0.57	0.52	
KNN	H	0.51	0.84	0.63	0.51
	D	0.00	0.00	0.00	
	A	0.57	0.43	0.49	
XGBoost	H	0.52	0.78	0.63	0.50
	D	0.30	**0.08**	**0.12**	
	A	0.50	0.43	0.47	
ANN	H	0.49	0.77	0.60	0.46
	D	0.33	0.03	0.05	
	A	0.38	0.35	0.36	

Table 5.7: Bundesliga season 15/16 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.54**	0.87	0.67	0.51
	D	0.00	0.00	0.00	
	A	0.39	0.27	**0.32**	
Random Forest	H	0.53	0.84	0.64	0.48
	D	0.00	0.00	0.00	
	A	0.34	**0.29**	**0.32**	
SVM	H	0.50	**0.97**	0.66	0.50
	D	0.00	0.00	0.00	
	A	0.33	0.02	0.05	
KNN	H	0.53	0.95	**0.68**	**0.52**
	D	0.00	0.00	0.00	
	A	**0.50**	0.17	0.25	
XGBoost	H	**0.54**	0.56	0.55	0.42
	D	**0.25**	**0.29**	**0.27**	
	A	0.36	**0.29**	**0.32**	
ANN	H	0.53	0.83	0.65	0.49
	D	0.00	0.00	0.00	
	A	0.37	0.27	0.31	

Table 5.8: Bundesliga season 16/17 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.49	0.90	0.63	**0.50**
	D	**0.38**	0.08	0.13	
	A	0.64	0.20	0.31	
Random Forest	H	**0.54**	0.72	0.62	0.46
	D	0.19	0.08	0.11	
	A	0.40	0.39	0.39	
SVM	H	0.48	**0.99**	**0.65**	**0.50**
	D	0.00	0.00	0.00	
	A	**0.86**	0.14	0.24	
KNN	H	0.50	0.75	0.60	0.44
	D	0.21	0.16	0.18	
	A	0.53	0.20	0.30	
XGBoost	H	0.52	0.48	0.50	0.42
	D	0.26	**0.32**	**0.29**	
	A	0.44	**0.41**	**0.42**	
ANN	H	0.33	0.03	0.05	**0.50**
	D	0.33	0.03	0.05	
	A	0.70	0.16	0.26	

Table 5.9: Bundesliga season 17/18 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.55	0.86	0.67	0.56
	D	**0.50**	0.03	0.05	
	A	0.57	0.51	0.54	
Random Forest	H	**0.62**	0.77	**0.68**	**0.58**
	D	0.00	0.00	0.00	
	A	0.52	**0.71**	0.60	
SVM	H	0.53	**0.91**	0.67	0.56
	D	0.00	0.00	0.00	
	A	**0.63**	0.45	**0.62**	
KNN	H	0.53	0.84	0.65	0.52
	D	0.33	0.11	0.17	
	A	0.56	0.37	0.44	
XGBoost	H	0.55	0.71	0.62	0.52
	D	0.29	**0.14**	**0.19**	
	A	0.53	0.51	0.52	
ANN	H	0.53	0.88	0.66	0.54
	D	0.00	0.00	0.00	
	A	0.57	0.43	0.49	

Table 5.10: Bundesliga season 18/19 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.40	0.98	0.56	0.44
	D	0.00	0.00	0.00	
	A	0.74	0.22	0.34	
Random Forest	H	**0.41**	0.98	**0.58**	**0.48**
	D	0.00	0.00	0.00	
	A	0.84	0.33	0.47	
SVM	H	0.39	0.96	0.56	0.42
	D	0.00	0.00	0.00	
	A	0.65	0.20	0.31	
KNN	H	0.40	0.94	0.57	**0.48**
	D	0.00	0.00	0.00	
	A	0.81	**0.34**	**0.48**	
XGBoost	H	0.39	**1.00**	0.56	0.44
	D	0.00	0.00	0.00	
	A	**1.00**	0.20	0.34	
ANN	H	0.38	**1.00**	0.55	0.41
	D	0.00	0.00	0.00	
	A	0.90	0.24	0.14	

Table 5.11: Bundesliga season 19/20 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.56**	0.83	**0.67**	**0.56**
	D	0.00	0.00	0.00	
	A	0.58	0.54	**0.56**	
Random Forest	H	0.53	0.83	0.65	0.53
	D	0.00	0.00	0.00	
	A	0.51	0.42	0.46	
SVM	H	0.53	**0.90**	0.66	0.55
	D	0.00	0.00	0.00	
	A	**0.62**	0.39	0.48	
KNN	H	0.55	0.80	0.65	0.55
	D	0.00	0.00	0.00	
	A	0.54	0.54	0.54	
XGBoost	H	0.54	0.84	0.66	0.54
	D	0.00	0.00	0.00	
	A	0.57	0.46	0.51	
ANN	H	0.54	0.74	0.63	0.53
	D	0.00	0.00	0.00	
	A	0.51	**0.58**	0.54	

Table 5.12: Premier League season 14/15 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.52	0.80	0.63	0.48
	D	**0.50**	0.08	0.13	
	A	0.40	0.40	0.40	
Random Forest	H	0.48	**0.85**	0.61	0.48
	D	0.41	0.13	0.20	
	A	0.52	0.25	0.33	
SVM	H	**0.56**	0.71	0.63	**0.51**
	D	**0.50**	0.02	0.04	
	A	0.44	**0.66**	**0.53**	
KNN	H	0.51	0.80	0.62	0.49
	D	0.26	0.09	0.14	
	A	**0.53**	0.40	0.45	
XGBoost	H	0.51	0.65	0.57	0.45
	D	0.37	0.19	0.25	
	A	0.36	0.38	0.37	
ANN	H	**0.56**	0.81	**0.66**	**0.51**
	D	0.36	**0.30**	**0.33**	
	A	0.52	0.23	0.32	

Table 5.13: Premier League season 15/16 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.61	0.94	0.74	0.62
	D	0.00	0.00	0.00	
	A	0.66	0.55	0.60	
Random Forest	H	**0.64**	0.92	**0.75**	**0.63**
	D	0.00	0.00	0.00	
	A	0.62	**0.62**	**0.62**	
SVM	H	0.59	**0.95**	0.73	0.61
	D	0.00	0.00	0.00	
	A	**0.70**	0.49	0.58	
KNN	H	0.61	0.92	0.73	0.61
	D	0.00	0.00	0.00	
	A	0.60	0.55	0.57	
XGBoost	H	0.59	0.94	0.72	0.61
	D	0.00	0.00	0.00	
	A	0.68	0.49	0.57	
ANN	H	0.61	0.94	0.74	0.62
	D	0.00	0.00	0.00	
	A	0.66	0.55	0.60	

Table 5.14: Premier League season 16/17 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.71**	0.50	0.59	0.53
	D	0.39	**0.60**	**0.47**	
	A	**0.54**	0.52	0.53	
Random Forest	H	0.57	0.74	0.64	0.52
	D	0.29	0.13	0.18	
	A	0.51	0.54	0.52	
SVM	H	0.54	**0.81**	**0.65**	0.51
	D	0.00	0.00	0.00	
	A	0.49	0.50	0.50	
KNN	H	0.57	0.68	0.62	**0.54**
	D	**0.55**	0.12	0.19	
	A	0.49	**0.72**	**0.58**	
XGBoost	H	0.53	0.80	0.64	0.52
	D	0.29	0.04	0.07	
	A	0.50	0.52	0.51	
ANN	H	0.55	0.80	**0.65**	0.52
	D	0.27	0.06	0.10	
	A	0.51	0.52	0.51	

Table 5.15: Premier League season 17/18 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.60	0.71	0.65	**0.57**
	D	**1.00**	**0.03**	**0.06**	
	A	0.53	0.66	**0.59**	
Random Forest	H	0.58	0.78	0.67	**0.57**
	D	0.00	0.00	0.00	
	A	0.56	0.56	0.56	
SVM	H	0.58	**0.84**	**0.68**	**0.57**
	D	0.10	**0.03**	0.05	
	A	**0.66**	0.44	0.52	
KNN	H	**0.66**	0.61	0.63	0.56
	D	0.50	**0.03**	**0.06**	
	A	0.47	**0.76**	0.58	
XGBoost	H	0.57	0.74	0.64	0.55
	D	0.00	0.00	0.00	
	A	0.52	0.55	0.54	
ANN	H	0.62	0.69	0.66	0.56
	D	0.00	0.00	0.00	
	A	0.48	0.65	0.55	

Table 5.16: Premier League season 18/19 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.60**	0.71	0.65	**0.57**
	D	**1.00**	0.03	0.06	
	A	**0.51**	**0.65**	**0.57**	
Random Forest	H	0.57	0.78	**0.66**	0.53
	D	0.00	0.00	0.00	
	A	0.46	0.58	0.52	
SVM	H	0.52	**0.83**	0.64	0.49
	D	0.00	0.00	0.00	
	A	0.42	0.36	0.39	
KNN	H	0.54	0.74	0.62	0.51
	D	0.50	0.02	0.04	
	A	0.46	0.55	0.50	
XGBoost	H	0.57	0.76	0.65	0.54
	D	0.38	**0.07**	**0.11**	
	A	0.49	0.56	0.53	
ANN	H	0.57	0.79	**0.66**	0.54
	D	0.00	0.00	0.00	
	A	0.49	0.60	0.54	

Table 5.17: Premier League season 19/20 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.60	0.04	0.07	0.34
	D	0.30	**0.81**	**0.44**	
	A	0.46	0.32	0.38	
Random Forest	H	0.57	**0.78**	**0.66**	**0.53**
	D	0.00	0.00	0.00	
	A	0.46	**0.58**	**0.52**	
SVM	H	0.64	0.42	0.51	0.38
	D	0.27	0.59	0.37	
	A	0.38	0.11	0.17	
KNN	H	0.57	0.14	0.23	0.35
	D	0.29	0.74	0.41	
	A	**0.48**	0.26	0.34	
XGBoost	H	0.59	0.58	0.58	0.49
	D	**0.33**	0.41	0.37	
	A	0.55	0.43	0.48	
ANN	H	**0.68**	0.25	0.37	0.39
	D	0.30	0.52	0.38	
	A	0.39	0.49	0.43	

Table 5.18: Serie A season 14/15 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.59	0.71	0.64	0.49
	D	0.00	0.00	0.00	
	A	0.36	0.62	0.46	
Random Forest	H	0.44	**1.00**	0.61	0.44
	D	0.00	0.00	0.00	
	A	0.00	0.00	0.00	
SVM	H	0.62	0.58	0.60	0.49
	D	0.20	0.04	0.06	
	A	0.42	0.83	**0.56**	
KNN	H	**0.70**	0.44	0.54	0.45
	D	0.00	0.00	0.00	
	A	0.34	**0.96**	0.51	
XGBoost	H	0.68	0.65	0.67	**0.56**
	D	**0.45**	**0.36**	**0.40**	
	A	**0.46**	0.60	0.52	
ANN	H	0.58	**0.73**	0.65	0.52
	D	0.00	0.00	0.00	
	A	0.44	0.71	0.54	

Table 5.19: Serie A season 15/16 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.56	0.81	0.66	0.55
	D	0.00	0.00	0.00	
	A	0.53	**0.56**	0.55	
Random Forest	H	0.50	**1.00**	0.61	0.50
	D	0.00	0.00	0.00	
	A	0.00	0.00	0.00	
SVM	H	0.54	0.89	0.67	0.55
	D	0.00	0.00	0.00	
	A	**0.60**	0.47	0.53	
KNN	H	0.56	0.87	**0.68**	0.57
	D	0.00	0.00	0.00	
	A	0.58	0.55	0.56	
XGBoost	H	**0.58**	0.83	**0.68**	**0.58**
	D	**0.56**	**0.12**	**0.20**	
	A	**0.60**	**0.56**	**0.58**	
ANN	H	0.56	0.89	**0.68**	0.57
	D	0.00	0.00	0.00	
	A	**0.60**	0.52	0.55	

Table 5.20: Serie A season 16/17 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.76**	0.43	0.55	0.43
	D	0.23	**0.51**	**0.32**	
	A	0.48	0.38	0.42	
Random Forest	H	0.72	0.71	**0.71**	0.58
	D	0.00	0.00	0.00	
	A	0.47	0.78	0.58	
SVM	H	0.74	0.64	0.69	**0.59**
	D	**0.64**	0.17	0.27	
	A	0.49	**0.81**	**0.61**	
KNN	H	0.74	0.50	0.60	0.49
	D	0.26	0.29	0.28	
	A	0.45	0.62	0.52	
XGBoost	H	**0.76**	0.58	0.66	0.52
	D	0.26	0.24	0.25	
	A	0.45	0.60	0.51	
ANN	H	0.67	**0.72**	0.69	0.57
	D	0.38	0.27	0.31	
	A	**0.53**	0.57	0.55	

Table 5.21: Serie A season 17/18 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.64	0.65	0.65	0.54
	D	0.00	0.00	0.00	
	A	0.46	0.75	0.57	
Random Forest	H	**0.73**	0.70	**0.71**	**0.58**
	D	**0.50**	0.02	0.05	
	A	0.46	**0.78**	**0.58**	
SVM	H	0.51	**0.88**	0.65	0.51
	D	0.44	0.15	0.23	
	A	0.53	0.29	0.38	
KNN	H	0.61	0.65	0.63	0.49
	D	0.31	**0.44**	0.36	
	A	**0.59**	0.29	0.39	
XGBoost	H	0.60	0.64	0.63	0.49
	D	0.35	**0.44**	**0.39**	
	A	0.49	0.31	0.38	
ANN	H	0.54	0.86	0.66	0.53
	D	0.43	0.17	0.25	
	A	0.57	0.38	0.46	

Table 5.22: Serie A season 18/19 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.64	0.65	0.65	0.43
	D	0.00	0.00	0.00	
	A	0.46	0.75	0.57	
Random Forest	H	0.60	0.74	0.66	0.57
	D	**0.62**	0.12	0.20	
	A	0.53	0.66	0.59	
SVM	H	0.58	**0.80**	**0.67**	**0.58**
	D	0.60	0.07	0.12	
	A	**0.59**	0.66	**0.62**	
KNN	H	0.54	0.74	0.62	0.52
	D	0.00	0.00	0.00	
	A	0.49	0.60	0.54	
XGBoost	H	**0.68**	0.64	0.66	0.56
	D	0.54	**0.16**	**0.25**	
	A	0.48	0.73	0.58	
ANN	H	0.67	0.66	**0.67**	0.55
	D	0.00	0.00	0.00	
	A	0.47	**0.78**	0.58	

Table 5.23: Serie A season 19/20 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.59	0.80	0.68	0.57
	D	0.31	0.11	0.17	
	A	0.60	0.55	0.57	
Random Forest	H	0.65	0.82	**0.73**	**0.61**
	D	**0.33**	0.05	0.08	
	A	0.55	**0.70**	**0.61**	
SVM	H	0.53	**0.97**	0.69	0.57
	D	0.00	0.00	0.00	
	A	**0.81**	0.38	0.51	
KNN	H	0.65	0.79	0.71	0.58
	D	0.22	0.05	0.08	
	A	0.53	0.68	0.59	
XGBoost	H	**0.66**	0.74	0.70	0.59
	D	0.30	**0.20**	**0.24**	
	A	0.63	0.66	0.64	
ANN	H	0.61	0.76	0.68	0.57
	D	0.25	0.05	0.08	
	A	0.54	0.68	0.60	

Table 5.24: La Liga season 14/15 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.58	0.87	**0.70**	**0.58**
	D	**1.00**	0.04	0.09	
	A	0.55	**0.53**	**0.54**	
Random Forest	H	0.57	0.86	0.68	0.52
	D	0.22	0.16	0.18	
	A	0.63	0.23	0.33	
SVM	H	0.50	**1.00**	0.67	0.51
	D	0.40	0.04	0.08	
	A	**1.00**	0.04	0.07	
KNN	H	0.57	0.87	0.69	0.54
	D	0.25	0.07	0.11	
	A	0.53	0.38	0.44	
XGBoost	H	**0.60**	0.84	**0.70**	0.56
	D	0.32	**0.22**	**0.26**	
	A	0.67	0.38	0.48	
ANN	H	0.58	0.89	**0.70**	0.57
	D	0.50	0.02	0.04	
	A	0.57	0.49	0.53	

Table 5.25: La Liga season 15/16 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	**0.78**	0.45	0.57	0.48
	D	**0.22**	**0.57**	0.31	
	A	**0.66**	**0.48**	**0.55**	
Random Forest	H	0.65	0.80	**0.72**	**0.58**
	D	0.21	0.17	0.19	
	A	0.62	0.48	0.54	
SVM	H	0.56	**0.86**	0.69	0.57
	D	0.00	0.00	0.00	
	A	0.62	0.41	0.50	
KNN	H	0.65	0.54	0.59	0.47
	D	0.17	0.34	0.23	
	A	**0.66**	0.44	0.53	
XGBoost	H	0.62	0.80	0.70	0.55
	D	0.17	0.14	0.16	
	A	0.62	0.41	0.50	
ANN	H	0.62	0.79	0.69	0.57
	D	**0.22**	0.17	**0.69**	
	A	**0.66**	**0.48**	**0.55**	

Table 5.26: La Liga season 16/17 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.58	**0.77**	0.66	0.54
	D	0.00	0.00	0.00	
	A	0.46	0.57	0.51	
Random Forest	H	0.59	0.76	**0.67**	**0.55**
	D	0.00	0.00	0.00	
	A	0.47	0.62	0.54	
SVM	H	**0.61**	0.62	0.62	0.52
	D	0.00	0.00	0.00	
	A	0.43	0.77	0.55	
KNN	H	0.40	0.47	0.43	0.52
	D	0.00	0.00	0.00	
	A	**0.57**	**0.78**	**0.66**	
XGBoost	H	0.60	0.54	0.56	0.51
	D	**0.39**	**0.16**	**0.23**	
	A	0.45	0.75	0.57	
ANN	H	0.59	**0.77**	**0.67**	0.54
	D	0.00	0.00	0.00	
	A	0.46	0.58	0.52	

Table 5.27: La Liga season 17/18 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.55	0.70	0.62	0.47
	D	0.33	0.33	0.33	
	A	0.36	0.18	0.24	
Random Forest	H	0.61	0.75	**0.67**	0.47
	D	0.28	**0.44**	0.34	
	A	0.00	0.00	0.00	
SVM	H	0.53	**0.90**	0.66	0.49
	D	0.22	0.12	0.16	
	A	**0.80**	0.08	0.15	
KNN	H	0.59	0.60	0.59	0.44
	D	0.29	0.38	0.33	
	A	0.32	0.22	0.26	
XGBoost	H	**0.62**	0.72	**0.67**	**0.54**
	D	**0.36**	0.42	**0.39**	
	A	0.55	**0.32**	**0.41**	
ANN	H	0.57	0.80	**0.67**	0.49
	D	0.27	0.29	0.28	
	A	0.56	0.10	0.17	

Table 5.28: La Liga season 18/19 experiment results

Abbreviations: H: Home Win, D: Draws, A: Away Win

Classifier	Result	Precision	Recall	F1-score	Accuracy
Logistic Regression	H	0.50	0.93	**0.65**	0.51
	D	0.00	0.00	0.00	
	A	0.56	0.27	0.36	
Random Forest	H	0.55	0.80	**0.65**	0.51
	D	0.00	0.00	0.00	
	A	0.41	**0.50**	0.45	
SVM	H	0.50	**0.94**	**0.65**	**0.52**
	D	0.00	0.00	0.00	
	A	**0.62**	0.29	0.39	
KNN	H	0.56	0.68	0.61	0.47
	D	0.33	**0.28**	**0.30**	
	A	0.38	0.29	0.33	
XGBoost	H	0.54	0.74	0.62	0.48
	D	0.17	0.04	0.06	
	A	0.41	0.46	0.44	
ANN	H	**0.57**	0.74	0.64	0.51
	D	**0.35**	0.12	0.18	
	A	0.44	**0.50**	**0.47**	

Table 5.29: La Liga season 19/20 experiment results

Chapter 6

Conclusion

In this thesis, we presented several models that could help people predict the outcome of football games. To do so, we used data from the 4 major European football leagues for the 2014 to 2020 seasons. While other studies usually test their models for a small number of matches (e.g., 18 fixtures of the last season), we decided to predict the following two scenarios: i) the results of the last two seasons of the dataset and ii) the results of the second round of each season.

In our research, Random Forest achieved the best results in terms of overall accuracy, F1-score for home and away wins, and recall for away wins. XGboost performed best in terms of accuracy, recall, and F1-score for draws and precision for home wins. SVM did well concerning precision for away wins and recall for home wins. If someone wants to use a model for a betting activity, we would recommend Random Forrest, as it achieved the highest accuracy in both scenarios.

Although our model has shown promising results in predicting the outcome of football games, it could be further improved if we had more data. That data could include information about each team's players, weather conditions, motivation to win the game, injuries, etc. That could also help improve the performance of our ANN, which did not perform well due to the small amount of data. In addition, in the future, we might try to implement an RNN with LSTMs, which according to various studies has good potential for predicting football game results.

Bibliography

[1] Random forest. (2022, december 24). in wikipedia. https://en.wikipedia.org/wiki/Random_forest. Accessed: 15-01-2023.

[2] K-nearest neighbor(knn) algorithm for machine learning. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning. Accessed: 15-01-2023.

[3] Mohammadreza Sheykhmousa and Masoud Mahdianpari. Support vector machine vs. random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10 2020.

[4] Chang Vui, Gan Soon, Chin On, Rayner Alfred, and Patricia Anthony. A review of stock market prediction with artificial neural network (ann). pages 477–482, 11 2013.

[5] J. Gościk and T. Łukaszuk. Application of the recursive feature elimination and the relaxed linear separability feature selection algorithms to gene expression data analysis. *Advances in Computer Science Research*, (10):39–52, 2013.

[6] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020.

[7] Jason Wise. How much data is generated every day in 2023? (new stats). https://earthweb.com/how-much-data-is-created-every-day/. Accessed: 22-01-2023.

[8] Kurtis Pykes. Sports analytics: How different sports use data analytics. https://www.datacamp.com/blog/sports-analytics-how-different-sports-use-data-analysis. Accessed: 22-01-2023.

[9] Alyssa Schroer. How sports analytics are used today, by teams and fans. `https://builtin.com/big-data/big-data-companies-sports`. Accessed: 22-01-2023.

[10] William Shughart II and William Shughart. Moneyball: The art of winning an unfair game, by lewis, m. new york and london: Norton, 2003, xv + 288 pp., usd 24.95 (cloth). *Managerial and Decision Economics*, 25:550–552, 12 2004.

[11] Performance analysis in football. `https://analyisport.com/performance-analysis-in-football/`. Accessed: 28-01-2023.

[12] Using sports betting analytics, does it help you win? `https://www.alloysports.com/blog?p=using-sports-betting-analytics-does-it-help-you-win`. Accessed: 28-01-2023.

[13] Josip Hucaljuk and Alen Rakipović. Predicting football scores using machine learning techniques. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1623–1627, 2011.

[14] Enes Eryarsoy and Dursun Delen. Predicting the outcome of a football game: A comparative analysis of single and ensemble analytics methods. 2019.

[15] Jan Kozak. Szymon głowania. *Heterogeneous ensembles of classifiers in predicting Bundesliga football results, Procedia Computer Science*, 192, 2021.

[16] Roman Nestoruk and Grzegorz Slowinski. Prediction of football games results. In *CS&P*, pages 156–165, 2021.

[17] Luca Carloni, Andrea De Angelis, Giuseppe Sansonetti, and Alessandro Micarelli. *A Machine Learning Approach to Football Match Result Prediction*, pages 473–480. 07 2021.

[18] Yunfei Li and Yubin Hong. Prediction of football match results based on edge computing and machine learning technology. *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, 13(2):1–10, 2022.

[19] Ekansh Tiwari, Prasanjit Sardar, and Sarika Jain. Football match result prediction using neural networks and deep learning. In *2020 8th International Conference on Reliabil-*

ity, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pages 229–231, 2020.

[20] Alba Gálvez, Ricardo Gonzalez, Sully Gálvez, and Mario García. Model to predict the result of a soccer match based on the number of goals scored by a single team. *Computación y Sistemas*, 26, 03 2022.

[21] Aleksandra Bartosik and Hannes Whittingham. Chapter 7 - evaluating safety and toxicity. In Stephanie Kay Ashenden, editor, *The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry*, pages 119–137. Academic Press, 2021.

[22] Saishruthi Swaminathan. Logistic regression — detailed overview. `https://www.ibm.com/topics/knn`. Accessedç: 13-01-2023.

[23] F. Pedregosa, 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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] Wikipedia contributors. Ensemble learning — Wikipedia, the free encyclopedia, 2022. [Online; accessed 24-January-2023].

[25] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[26] Rahul Baboota and Harleen Kaur. Predictive analysis and modelling football results using machine learning approach for english premier league. *International Journal of Forecasting*, 35(2):741–755, 2019.

[27] What is the k-nearest neighbors algorithm? `https://www.ibm.com/topics/knn`. Accessed: 13-01-2023.

[28] Corinna Cortes and Vladimir Naumovich Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 2004.

[29] Martin Hofmann. Support vector machines-kernels and the kernel trick. *Notes*, 26(3):1–16, 2006.

[30] AD Dongare, RR Kharde, Amit D Kachare, et al. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194, 2012.

[31] Tianqi Chen and Carlos Guestrin. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2016. ACM.

[32] Sukhpreet Singh Dhaliwal, Abdullah-Al Nahid, and Robert Abbas. Effective intrusion detection system using xgboost. *Information*, 9(7):149, 2018.

[33] Ilan Reinstein. Xgboost, a top machine learning method on kaggle, explained. `https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html`. Accessed: 16-01-2023.

[34] Xgboost documentation. `https://xgboost.readthedocs.io/en/stable/parameter.html`. Accessed: 16-01-2023.

[35] Football betting|football results|free bets|betting odds. `https://football-data.co.uk/`. Accessed: 10-01-2023.

[36] Fifa index. `https://www.fifaindex.com/`. Accessed: 10-01-2023.

[37] Football data: Expected goals and other metrics. `https://www.kaggle.com/datasets/slehkyi/extended-football-stats-for-european-leagues-xg?resource=download`. Accessed: 10-01-2023.

[38] Fadi Thabtah Rory P Bunker. A machine learning framework for sport result prediction. In *Applied Computing and Informatics*, volume 15, pages 27–33. Emerald Publishing Limited, 2019. https://doi.org/10.1016/j.aci.2017.09.005.

[39] H20 wiki. `https://h2o.ai/wiki/feature-selection/`. Accessed: 10-01-2023.

[40] F. Caron, G. Bontempi, and J. M. Renders. Learning from imbalanced data for football outcome prediction. *Journal of Sports Analytics*, 5(1).

[41] Niklas Donges. 4 disadvantages of neural networks and deep learning. `https://builtin.com/data-science/disadvantages-neural-networks`. Accessed: 28-01-2023.

APPENDICES

Appendix A

Dataset

A.1 Football-data

Div	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR
E0	16/08/14	Arsenal	Crystal Palace	2	1	H	1	1	D
E0	16/08/14	Leicester	Everton	2	2	D	1	2	A
E0	16/08/14	Man United	Swansea	1	2	A	0	1	A
E0	16/08/14	QPR	Hull	0	1	A	0	0	D

Referee	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR	B365H
J Moss	14	4	6	2	13	19	9	3	2	2	0	1	1.25
M Jones	11	13	3	3	16	10	3	6	1	1	0	0	3.2
M Dean	14	5	5	4	14	20	4	0	2	4	0	0	1.36
C Pawson	19	11	6	4	10	10	8	9	1	2	0	0	2.5

B365D	B365A	BWH	BWD	BWA	IWH	IWD	IWA	LBH	LBD	LBA	PSH
6.5	15	1.25	5.5	12	1.3	5	9	1.25	6	13	1.26
3.4	2.4	2.9	3.2	2.4	2.9	3.3	2.3	3.25	3.4	2.25	3.14
5	11	1.4	4.75	9	1.33	5	8	1.36	5	10	1.37
3.3	3.1	2.5	3.1	2.85	2.3	3.3	2.9	2.4	3.25	3.1	2.48

PSD	PSA	WHH	WHD	WHA	SJH	SJD	SJA	VCH	VCD	VCA
6.45	14.01	1.25	5.5	12	1.25	5.75	12	1.25	6.25	10.5
3.38	2.46	3.1	3.1	2.4	3	3.3	2.38	3.2	3.4	2.4
5.1	10.6	1.36	4.5	9	1.36	5	8	1.36	5.2	10
3.26	3.22	2.6	3	2.9	2.5	3.25	2.88	2.55	3.2	3.12

BbMx	BbAv>2.5	BbMx<2.5	BbAv<2.5
1.77	1.72	2.26	2.1
2.1	2	1.9	1.8
1.77	1.71	2.3	2.13
2.52	2.36	1.65	1.58

A.2 Fifa-Index

Team	ATT	MID	DEF	TOVA
Chelsea	82	82	81	83
Man City	83	82	81	82
Man United	85	80	75	81
Arsenal	80	80	77	80
Liverpool	79	78	75	79
Everton	78	78	78	78
Tottenham	78	77	77	78
Southampton	77	75	76	76
Stoke	76	75	74	76
Swansea	77	74	74	75
West Ham	75	74	72	75
Newcastle	74	75	74	75
QPR	74	74	71	75
Aston Villa	72	74	74	75
Sunderland	76	73	72	74
West Brom	74	74	71	74
Crystal Palace	73	73	73	73
Leicester	71	72	68	72
Burnley	71	70	71	70
Hull	75	73	70	73

A.3 Understats

Year	League	xG	xGA	npxG	npxGA	deep	deep_allowed	ppda_coef
Bundesliga	2014	2,570	1,198	2,570	1,198	5	4	9,625
Bundesliga	2014	1,503	1,312	1,503	1,307	10	1	4,756
Bundesliga	2014	1,229	0,31	1,229	0,310	13	3	5,06
Bundesliga	2014	1,035	0,203	1,035	0,203	6	2	4,423

team	xpts	date	oppda_coef	xG_diff	xGA_diff	xpts_diff
Bayern Munich	2,3486	2014-08-22 19:30	21,85	0,570	0,198	-0,651
Bayern Munich	1,5143	2014-08-30 17:30	17,695	0,503	0,307	0,514
Bayern Munich	2,1588	2014-09-13 14:30	16,961	-0,770	0,310	-0,841
Bayern Munich	2,1367	2014-09-20 14:30	9,446	1,035	0,203	1,136

Appendix B

Features

Date	HomeTeamID	AwayTeamID	FTR	round	league	AVGH	AVGD	AVGA
2015-01-10	14	22	H	2	EPL	2,2	3,26	3,7
2015-01-10	20	29	A	2	EPL	4,45	3,47	1,93
2015-01-10	16	32	D	2	EPL	2,26	3,38	3,41
2015-01-10	18	19	H	2	EPL	2,05	3,3	4,18

season	HT_wins	AT_wins	HT_draws	AT_draws	HT_losses	AT_losses	HTGS
2014/2015	3	5	5	7	12	8	19
2014/2015	3	8	11	5	6	7	18
2014/2015	8	9	5	5	7	6	25
2014/2015	4	4	6	7	10	9	19

ATGS	HTGC	ATGC	l5_ravg_HTST	l5_ravg_ATST	l5_ravg_HTCR	l5_ravg_ATCR
11	33	22	3,2	3	0,268	0,05
28	30	27	3,4	7,8	0,282	0,244
31	24	24	3,6	3,4	0,316	0,34
20	29	26	4,8	3,6	0,206	0,2

HToveral	AToveral	l5_ravg_HTxG	l5_ravg_ATxG	l5_ravg_HTxpts	l5_ravg_ATxpts
-14	-11	0,753	0,859	0,679	1,29
-12	1	1,340	1,984	1,51	2,006
1	7	1,286	1,005	1,086	0,924
-10	-6	1,572	1,159	1,47	1,414

l5_ravg_HTdeep	l5_ravg_ATdeep	l5_ravg_HTppda	l5_ravg_ATppda	l5_ravg_HTgs	l5_htdiff
5,8	6,2	10,19	15,257	0,8	-0,6
4,2	9,4	12,299	12,381	1	-0,4
4,4	5,8	12,589	13,991	1	-0,4
7,2	4,4	9,077	9,558	1	-0,8

l5_ravg_ATgs	l5_ravg_HTgc	l5_ravg_ATgc	l5_atdiff	avgHTP	avgATP	l5_ravg_HTp
0,2	1,4	0,6	-0,4	0,05	0,05	10,6
1,8	1,4	1,6	0,2	0	0,05	17,8
1	1,4	1,2	-0,2	0,05	0,04	25
1	1,8	1	0	0,04	0,14	16,4

l5_ravg_ATp	diff_points	diff_ATT	diff_MID	diff_DEF	diff_OVA
19,8	-8	-1	-2	-6	-3
23,4	-9	-3	-5	-3	-5
29,6	-3	2	0	2	0
14,2	-1	-1	1	1	1