INTERNATIONAL
HELLENIC
UNIVERSITY

Sport Analytics Algorithms for Football Performance Prediction

Christina Markopoulou

SID: 3308220015

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2024

THESSALONIKI – GREECE

Sport Analytics Algorithms for Football Performance Prediction

Christina Markopoulou

SID: 3308220015

Supervisor:	Assoc. Prof. Christos Tjortjis
Supervising Committee Members:	Dr. Christos Berberidis
	Dr. Paraskevas Koukaras

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2024

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in Data Science at the International Hellenic University.

The field of Sports Analytics has been experiencing rapid growth, with applications that are extremely beneficial to sports clubs. These organizations can use data to their advantage by gathering information about players and the game itself, and then using that knowledge to get important insights to improve the player's performance. Dozens of matches are played each week, thus resulting in a variety of available data.

The primary objective of this dissertation is to predict a player's performance within the domain of football. Most specifically, it aims to predict the number of goals a football player is likely to achieve in the upcoming season based on historical data.

To achieve this, football data were collected from reliable online sources [1]. Following that, feature engineering techniques were applied to shape the acquired data into the appropriate format. To place the data in a historical context, an approach called as season lag features was used. Additionally, a strategic approach was taken to enhance the analysis, where the original dataset was split to focus on the top 30% of players based on goal performance.

Six different machine learning algorithms were developed using python in order to achieve low metric values. The results were analyzed and compared in order to establish which model performed better in each case. Remarkably, the XGBoost algorithm emerged as the standout performer, achieving the lowest metric values with a Mean Absolute Error of 1.29 when applied to Serie A's dataset. Notably, an unusual observation emerged during the Bundesliga dataset study. It was discovered that using a reduced dataset resulted in better outcomes.

KEY WORDS: Python, Sports Analytics, Machine Learning, Data analytics, Linear Regression, Ridge Regression, Random Forest, Gradient Boosting, XGBoost, Multilinear perceptron, Octoparse

Acknowledgments

At this point, I would like to thank my supervisor, Professor Christos Tjortjis, for the trust he showed me, as well as his valuable help and guidance throughout the completion of this thesis. In addition, I would like to thank PhD candidate, Georgios Papageorgiou, who was providing efficient feedback and suggestions to the upcoming problems.

Additionally, I would like to thank my family and friends for their valuable support throughout my studies.

Christina Markopoulou

08-01-2023

Contents

List of Figures

List of Tables

1 Introduction

This dissertation consists of eight chapters, with each chapter serving a specific purpose. The initial chapter is the introduction of this topic. Following this, the second chapter, Background Information, discusses some general terms of machine learning, data mining and sport analytics. The third chapter, Literature Review, analyzes earlier studies in performance prediction as well as the historical evolution of sports analytics across various sports. Chapter four, Prediction Models, discusses the theory of the machine learning algorithms that were developed in this dissertation. Moving on to chapter five, Methodology, the methodologies applied during the experiments are descripted. In chapter six, the outcomes of the models used are presented. Chapter seven, Discussion, compares and analyzes the results. Finally, the eighth and concluding chapter, Conclusions, wraps up the dissertation and presents proposals for future study.

Sports analytics has transformed the world of sports nowadays. Sport analytics provides organizations and coaches additional information about a player's performance through new tracking technologies. This vast amount of data assists coaches in improving their decision-making and strategy. This innovation has boosted team competitiveness while also making sports more exciting for fans, who now have real-time access to a wealth of statistics.

This dissertation focuses on football. Football is one of the most famous sports globally, with a massive fanbase and significant financial investments. The scope of this dissertation is to predict a player’s performance in terms of goals using historical data. To achieve this, the models will be trained on information from the four seasons before, with the final evaluation taking place during the last season (2022-2023). To be more specific, players from four different leagues are included. Specifically, Bundesliga, Premier League, La Liga and Serie A. Furthermore, the final case includes a dataset where players from all leagues are added.

There is a big variety of available online data and repositories which contain statistics about players. However, it was decided to collect data from Sports Reference, which is a valid source of information. Data were collected for more than 5000 players from

season 2017-2018 to 2022-2023. Preprocessing and feature engineering techniques were needed in order to transform the dataset into the appropriate format, to be inserted in the prediction models. Furthermore, season lag features were implemented, alongside with the split of the dataset into the top 30% of players in each league.

Various implementations were tested. To begin with, unique approaches for each league and for the overall dataset were used. Each implementation was further divided into two versions: the first encompassed the entire dataset, while the second focused exclusively on the top 30% of players, determined by their goal performance. Ultimately, each version was further subdivided into three cases, determined by the attributes utilized in the training process, as elaborated in the subsequent chapter. For each case, various machine learning algorithms were tested. Specifically, these were: Linear and Ridge Regression, Random Forest, Gradient Boosting, XGBoost and Multilayer Perceptron.

The last step was to measure the effectiveness of the models. Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and R-squared were the five-evaluation metrics for each algorithm. The results depicted that the best metric values had XGBoost algorithm using the dataset of Serie A and the attributes of case 1. Case 1 attributes contained the 10 most correlated features related to the target variable ‘Gls’.

Certainly, a variety of external factors can have an effect on a player's performance. These variables include the player's mental state, injuries, weather, team dynamics etc. However, this research indicates that accurate predictions about the performance of a football player are possible. The added value of this dissertation is the use of advanced statistics, as well as the transformation of these statistics to historical data. Additionally, this dissertation penetrates into the world of cultural and gameplay variances, offering a layer of insight into how predictions differ between football leagues. Strategy and performance patterns are deeply connected with the distinctive characteristics of each league.

2 Background Information

In this section, general terms related to the topic will be analyzed. The first subsection refers to Machine Learning (ML), a branch of artificial intelligence that enables computers to learn from data and make predictions. There are three main categories of ML: Supervised, Unsupervised and Reinforcement Learning, as it will be explained below [2]. The next subsection is focused on Data Mining, the process of identifying hidden patterns and important insights within massive datasets to help decision-making [3]. Lastly, Sport Analytics is analyzed. Sport Analytics involves the collection and analysis of data related to athletic performance, team strategies and fan involvement in order to enhance decision-making and improve overall outcomes in the sports industry [4].

2.1 Machine Learning

Machine learning (ML) is a subset of artificial intelligence that empowers computers to learn from training data and find the patterns in data. ML uses algorithms to mimic how humans learn, eventually boosting their accuracy. As a result, the models can generate accurate predictions and become more experienced in decision making [2] [5]. ML has many applications, as shown in the diagram below.

https://www.ibm.com/topics/machine-learning

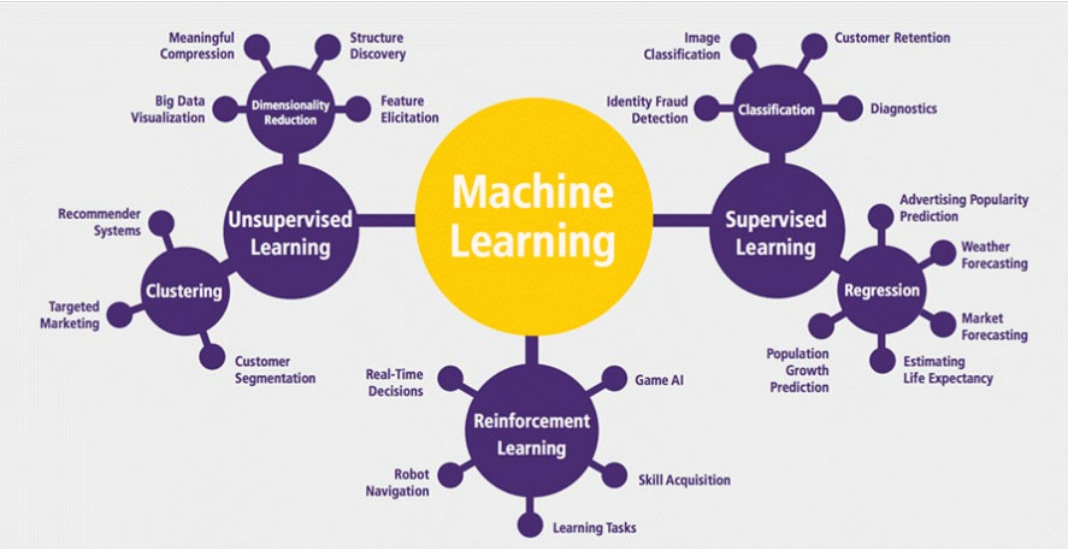

Figure 1: Machine Learning Algorithms [6]

There are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning [7]. The differences will be analyzed in the following chapters.

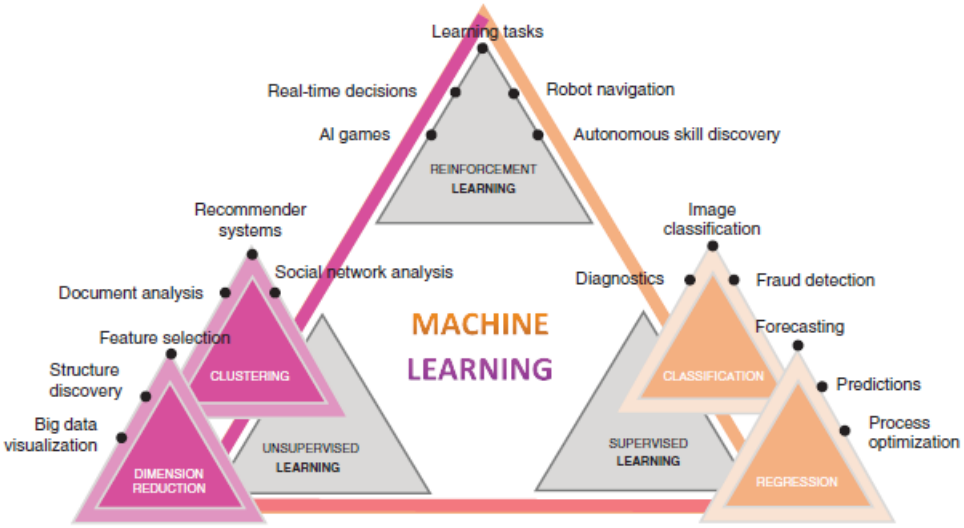

Figure 2: Machine learning approaches [7]

2.1.1 Supervised Learning

In supervised learning, the algorithms are trained on data that are labeled. The algorithm divides the dataset into two parts, using one for training and one for testing. The model continues to be trained until it recognizes the correlations between the input and the output data [7].

There are two categories of supervised learning problems: classification and regression. Classification classifies a set of data into specific categories. The output takes discrete values or classes. Some of the most known classification methods used are Linear classifier, Support Vector Machines (SVM), Decision Trees and Random Forest. On the other hand, regression is used to find the relationship between dependent and independent variables [7]. Some of the most well-known regression methods include Linear, Logistical and Polynomial Regression. The models employed in this dissertation are examined in the chapter Prediction Models.

2.1.2 Unsupervised Learning

On the other hand, algorithms in unsupervised learning are trained on unlabeled data with as little human supervision as possible and because of this, they are also known as self-organizing algorithms [8]. Unsupervised learning is divided into two major catego-

ries: clustering and dimensionality reduction. The goal of clustering is to divide a set of points into groups, where points in the same group are more comparable to those in other clusters. Dimensionality reduction reduces the number of input variables in a dataset [7].

Some of the most commonly used algorithms are [2][9]:

- Association Rules: if-then statements that aid in demonstrating likelihood of relationships among data. For example, if certain items are present, there is a high probability that other items will also be present.
- Clustering: a method of grouping similar data points or objects based on their characteristics or attributes. There are various clustering algorithms, and the method chosen relies on the data and the specific problem. Here are some examples of common clustering methods:
    - K-means clustering (Exclusive and overlapping clustering)
    - Ward's linkage (Hierarchical clustering)
    - Average linkage (Hierarchical clustering)
    - Complete (or maximum) linkage (Hierarchical clustering)
    - Single (or minimum) linkage (Hierarchical clustering)
    - Gaussian Mixture Models (Probabilistic clustering)
- Apriori Algorithms: the goal is to identify frequent item sets, meaning finding items that are often purchased together.
- Dimensionality Reduction: a data preprocessing technique used in ML to reduce the number of variables or features in a dataset. The most common algorithm is Principal Component Analysis (PCA).
- Singular Value Decomposition (SVD): a mathematical approach for expressing a matrix as a sequence of linear approximations that reveal the inherent structure and significance within the matrix [10].

2.1.3 Reinforcement Learning

Reinforcement learning is quite similar to the human learning mechanism. Reinforcement Learning is trained by a trial-and-error process. Rewards and punishments are employed in response to the model's feedback. The goal is to develop a model that maximizes the agent's overall cumulative reward [11].

The elements that describe a Reinforcement Learning problem are [11]:

1. Policy: Indicates how the learning agent behaves.
2. Reward function: Feedback from the environment. Positive rewards indicate desirable actions, while negative rewards signal undesirable outcomes.
3. Value function: A state's value reflects the total expected reward that an agent can acquire over time, beginning with that state [12].
4. Environment model: They provide a way to simulate or predict how the environment will evolve based on the agent's actions [12].

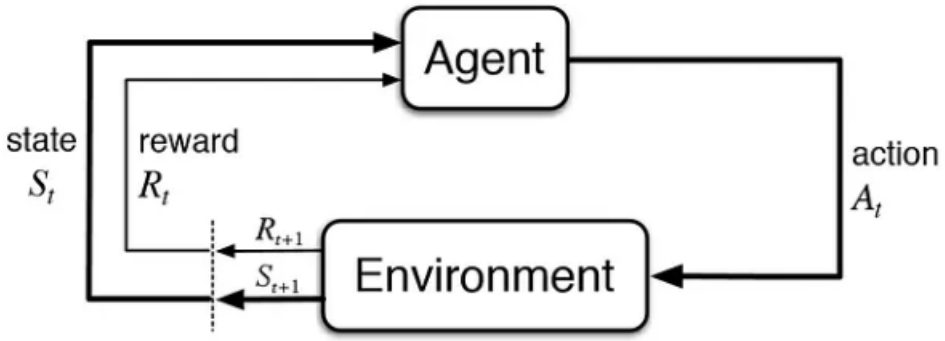

Figure 3: Action reward feedback loop of a Reinforcement Learning model [13]

The most used Reinforcement Learning algorithms are:

- Markov Decision Process (MDP)
- Dynamic Programming
- Monte Carlo method
- Q - Learning [11]

2.2 Data Mining

The rapid collection of data resulted in the development of structured databases and database management systems (DBMS). Data mining is the transformation of data to insightful information, the process of extracting useful patterns or knowledge from massive amounts of data. Knowledge discovery in databases, knowledge extraction, and data dredging, are some of the other terms that are used to describe it. Different types of information repositories are able to benefit from data mining, such as: business transac-

tions, scientific data, medical and personal data, surveillance video and pictures, satellite sensing and text reports [3].

The process of Data Mining is illustrated below.

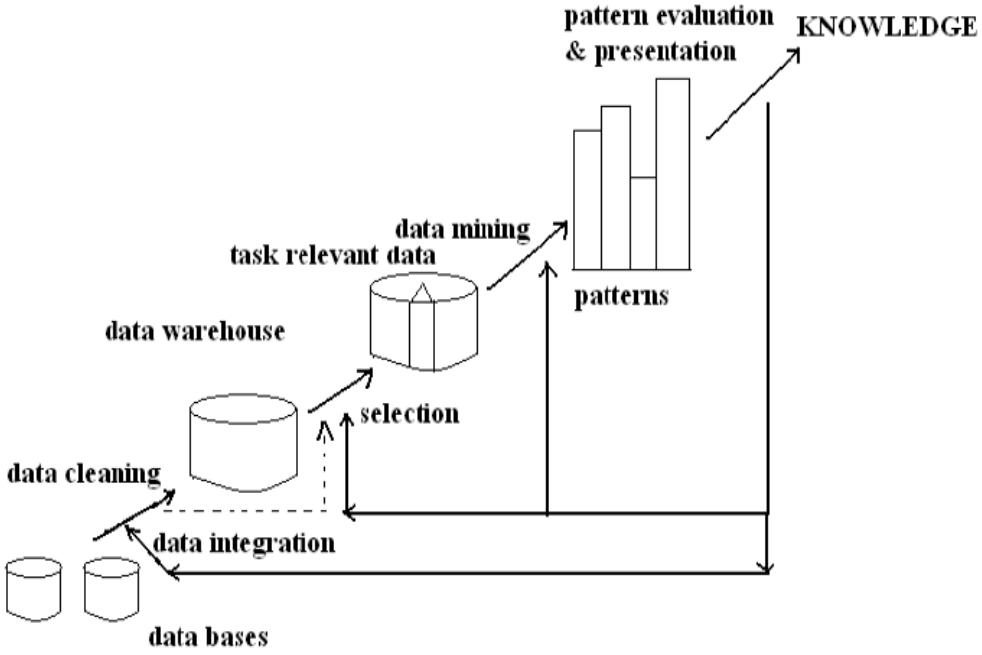

Figure 4: Data mining as a process of knowledge discovery [3]

Steps like data selection, data cleaning are vital for all the data mining algorithms to work. The next step is data transformation, where the selected data are turned into suitable forms in order for the algorithms to find the patterns and extract useful information [14][15].

2.3 Sport Analytics

Sport analytics refers to the data collection, analysis, and interpretation of data related to sports. It involves applying statistical approaches, data visualization, and technology to find patterns, trends, and insights that can improve sports decision-making, by optimizing team tactics, improving player performance, or increasing fan engagement [4]. As a result, sport analytics has become an essential tool for teams, coaches, and organizations [4].

2.3.1 Sport Analytics' Categories

There are two main categories of sport analytics: on-field and off-field analytics. The first one contains several concepts like helping teams in game strategy, methods for boosting the performance of the players. Of-field analytics is concerned with increasing the fan engagement, monitor ticket sales and cut expenses [16].

2.3.2 Sport Analytics' Applications

The emergence of sports analytics in recent years has been spectacular. Data-driven insights have radically changed the way sports are played, coached, and understood. Inspired by the revolutionary method presented in the film "Moneyball," teams from all disciplines have adopted a more analytical perspective. Based on Michael Lewis' book, this film depicts how the Oakland Athletics baseball organization used statistical analysis to construct a competitive squad despite financial restraints. The success of "Moneyball" indicated a new era in which analytics have become a vital tool for improving player performance, refining strategy, and gaining a competitive advantage, as analyzed further below [17].

One of the applications of sport analytics is **performance optimization**. By evaluating data from sensors and trackers, data science improves athlete performance in sports. This information aids in the identification of patterns, improving the quality of training, Furthermore, this has a tremendous impact on **injury prevention**. As a result, coaches can adapt training and create recovery plans for each individual athlete so as to maintain peak performance. Sport analytics is also becoming increasingly important in the operations of **betting** organizations. These organizations can deliver more accurate odds and suggestions to their consumers by analyzing data. This improves the whole sports betting experience by making it more informative and interesting for fans, resulting in bigger **fan engagement**. Sport analytics also helps in sports scouting and recruitment by recognizing talent based on performance data and ultimately producing competitive teams [4].

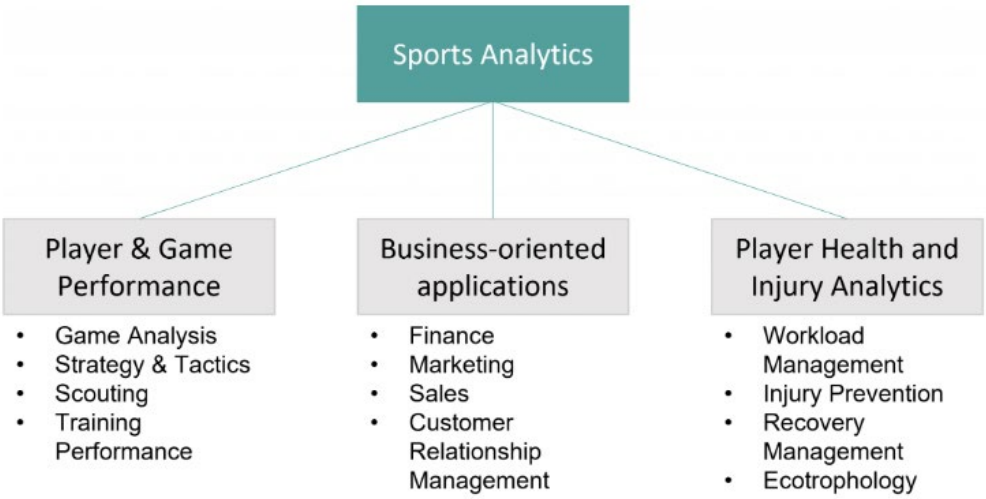

Figure 5: Usefulness of sport analytics [4]

2.3.3 Wearable Devices

Wearable devices have transformed sport analytics by delivering real-time, detailed data for each athlete's performance. These technologies have altered the way teams and coaches assess and optimize numerous aspects of the athlete's game.

These wearable devices monitor a wide range of attributes, including heart rate, speed, distance covered, acceleration, and even biomechanical data like stride length. Some advanced wearables can also track player positioning on the field, which is extremely useful for evaluating team formations and strategy [18].

By using wearable devices into training and performance analysis, teams can gain various advantages. For starters, they obtain a better awareness of individual player skills and limitations, allowing for more customized training programs. Second, these devices assist teams in refining game plans by providing information on player positions, fatigue levels, and tactical efficiency. Finally, they are very helpful in injury prevention by measuring player workload and detecting early symptoms of exhaustion. In this way teams can lower the chance of a player's injury and can maintain players in peak condition. As demonstrated, wearable devices have become vital tools in modern sport analytics, improving player performance, team plans, and overall game dynamics [19].

3 Literature Review

In this section of the dissertation, previous works and significant research about the historical background of sports analytics will be presented. This investigation will allow us to comprehend the progress made in the field up to the present day.

3.1 Historical Background

Sport analytics has a rich historical background that dates back to the early 20th century. Technology and data gathering methods have changed sport analytics over the years, spreading its effect across numerous sports, from basketball with the development of player tracking technology to football with the widespread use of performance analysis tools.

3.1.1 Tennis

Tennis is one of the most known sports around the world. It is played by a racket between two players (singles) or between two teams of two players (doubles). Anytime the opponent fails to return the ball, points are collected by a player or a team. Its origin begins from Britain, were Victorian gentlemen and ladies were playing it in Birmingham in the 19th century. Nowadays, tennis is played in a variety of surfaces (grass, hard courts and clay) and also, it consists 4 major tournaments, the Grand Slams (Wimbledon, US Open, Australian Open, Roland Garros) [20].

A tennis match is defined by a large number of factors, thus making the sport both entertaining and unexpected [21]. Professional players originate from a varied range of countries, having different playing styles and specialties. The development of a player can be influenced by many factors such as playing technique, tactics, psychology and physical fitness [22].

Tennis has millions of fans in the globe. The popularity of the sport has benefited it and more records of games and players data have been stored. As a result, sport analytics in tennis has been introduced, as well as the development of machine learning algorithms. The possibility to forecast tennis match outcomes prior to match start is of great interest to gamblers and betting businesses, and for that reason all and more data scientists are being involved [23].

In 2017, Chen and Zhong tried to predict the outcome of a tennis match based on historical data. The models that they were used were Logistic regression, Support Vector Classification and Naïve Bayes classification. This research provided some interesting observations. Firstly, total player's points won in the previous match can identify who is more likely to win. Secondly, the possibility of winning is influenced by the player's performance when faced with break points. Generally, the best model was Support Vector Machine with an accuracy bigger than 80% [24].

Gao and Kowalczyk conducted similar study in 2021, and managed to predict the outcome of a tennis match by using three machine learning algorithms (Support Vector Machine, Logistic regression, Random Forest). Out of the three models, Random Forest was the one with the most sensitivity to input parameter selection. Overall, this algorithm produced the higher accuracy, up to 83.18%. One of the achievements of this research is that they managed to produce better accuracy than the accuracy of betting odds. The second one is that they identified serve strength as a crucial predictor of a match outcome [23].

3.1.2 Basketball

Basketball was created in 1991 by James Naismith, a Canadian physical education teacher at Springfield in Massachusetts. It all started with the problem that his students were bored with all the games they instructed to do. As a result, Naismith tried to invent a game that would meet some criteria. Some of the most important were that this new game must be easy to learn and should be played indoors. In the end, he eventually conceived the concept of what we now universally recognize as 'basketball'. As the popularity of the sport expanded, the basketball as we know it today was formed. The rules changed, making the sport faster and more enjoyable. Within a few years, professional leagues were founded and basketball became an Olympic sport in 1936. Finally, the National Basketball Association (NBA) was formed in 1946 [25].

Nowadays, the vast amount of data generated by basketball has increased the interest of researchers. Data analysis in basketball is a game-changer, offering numerous advantages to the sport. It enables teams to optimize player performance, make informed decisions, and gain strategic advantages. Additionally, it aids in player development, injury prevention, and scouting new talent. This data-driven approach improves the fan experience and ensures the sport remains dynamic and competitive.

Thabtah, Zhang and Abdelhamid, in 2019, applied machine learning algorithms to predict NBA game results based on historical data. Furthermore, another goal of the research was to discover the most important factors influencing game results. The three models that were developed are: Naïve Bayes, artificial neural network and Decision tree. The results showed that the attribute defensive rebounds (DRB) is the most crucial factor in determining the outcome. One strange observation was that the dimensionality reduction did not result in achieving better accuracy. Accuracy fluctuated from 73% to 83% [26].

Another interesting research was conducted by Chen in 2020. The aim was to predict the regular season Most Valuable Player (MVP) award for season 2019-2020. Top 50 players during seasons 1979-2019 were used and their statistics were standardized. Multivariate correlation and recursive partitioning were used to reduce overfit risk. The prediction model was built using neural algorithms. The accuracy was greater than 90%. Another key point to note is that the model displayed Giannis Antetokounbo would win the MVP award, which he did [27].

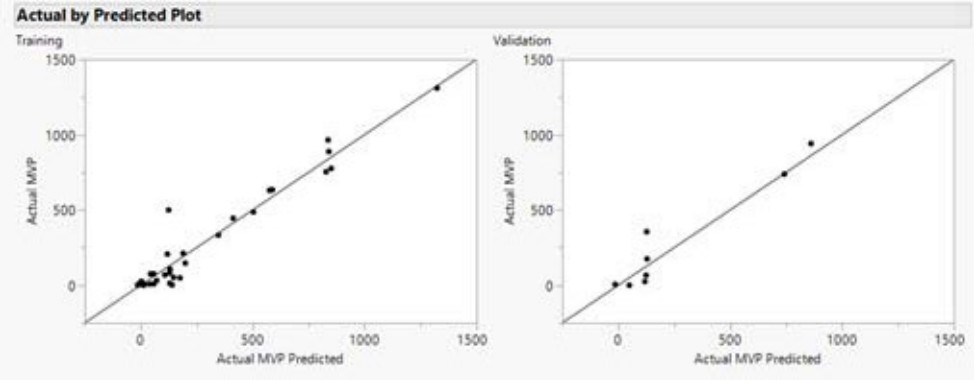

Figure 6: Neural MVP Model Prediction Accuracy [27]

In 2021, Sarlis, Tjortjis et al., tried to analyze the impact of injuries on basketball players and team performance in NBA, using data from 2010 to 2020. The researches had 4 different research questions. The 1st one was about the most common injury in the NBA league. According to the findings, there is a weak positive association between performance and injuries, with musculoskeletal injuries being the most frequent regarding lower performance. The 2nd research question was about the team that presented the largest number of injuries and it was San Antonio Spurs. Overall, several models were tested and the researchers managed to achieve almost 100% accuracy using XGBoost Tree Ensemble, XGBoost Linear Ensemble (Regression) and Linear Regression [28].

Lastly, Papageorgiou and Tjortjis in 2022 presented a topic about daily NBA player performance prediction. The project aimed to predict the fantasy points each player would achieve in a game by developing different ML models. Furthermore, they presented the Daily Lineup Optimizer (DLO), which may be utilized for NBA Fantasy Tournaments. They used historical data and specifically data from season 2010-2011 to season 2020-2021. The results showed that Voting Regressor performed best for the vast majority of players [29].

3.1.3 Baseball

Baseball is frequently recognized as one of the most beloved sports on the planet and its worth is millions of dollars [30]. It is a team sport played by two teams of nine players each on a diamond-shaped field. In this game a pitcher, from one team throws the ball to a batter from the opposing team, whose objectives is to hit the ball and progress through a sequence of bases including second and third base until eventually reaching home plate to score runs. Baseball is known for its rich history, strategic play, and the combination of individual skills such as pitching and batting [31].

In 2021, Huang and Li applied several maching learning algorithms to predict the outcome of Major League Baseball matches. The data that were used was 30 teams in season 2019. All the models -one-dimensional convolutional neural network, artificial neural network, support vector machine- achieved an accuracy bigger than 90%. The prediction findings showed that when all pitchers' data was included, the models produced higher forecast accuracies than when only the starting pitches data were used [32].

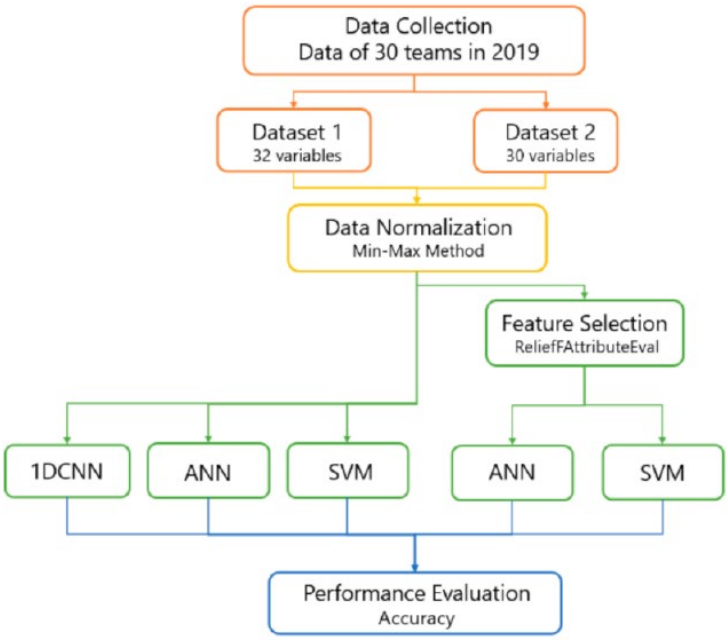

Figure 7: The research framework [32]

Karnuta et al., in 2020, developed a prediction system that can forecast future injuries and pinpoint the particular anatomical location of injuries in Major League Baseball (MLB) players. The study looks at data from 1931 position players and 1245 pitchers from 2000 to 2017. According to the statistics, 44.0% of position players and 43.6% of pitchers have previously been injured. Back injuries were particularly common among both position players and pitchers. When employing the top three ensemble classifications, advanced ML models beat logistic regression, with an average AUC (Area under the ROC curve) of 0.76 for position players and 0.65 for pitchers [33].

In the following diagram the process of the development is presented:

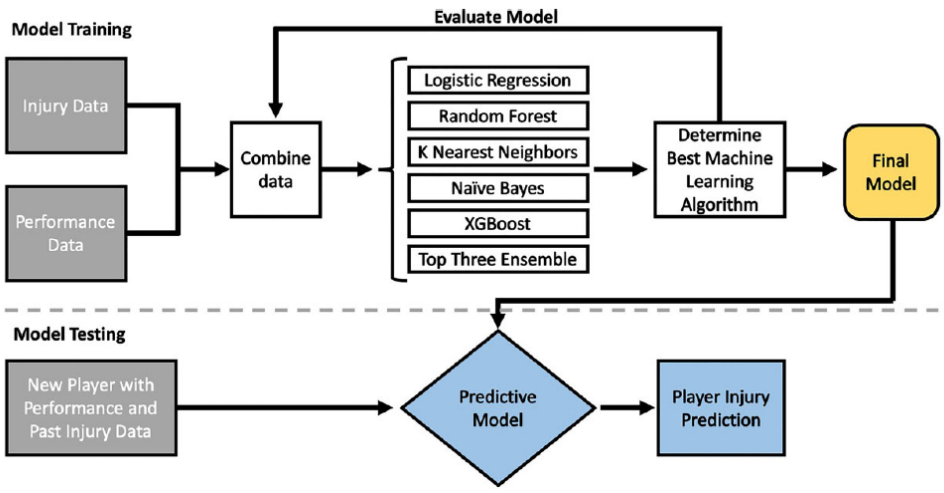

Figure 8: Schematic demonstrating of the ML algorithms development and testing [33]

Furthermore, the accuracy of all the models was approximately around 60% [33]. Figure 9 shows the top 20 variables for predicting future injury.

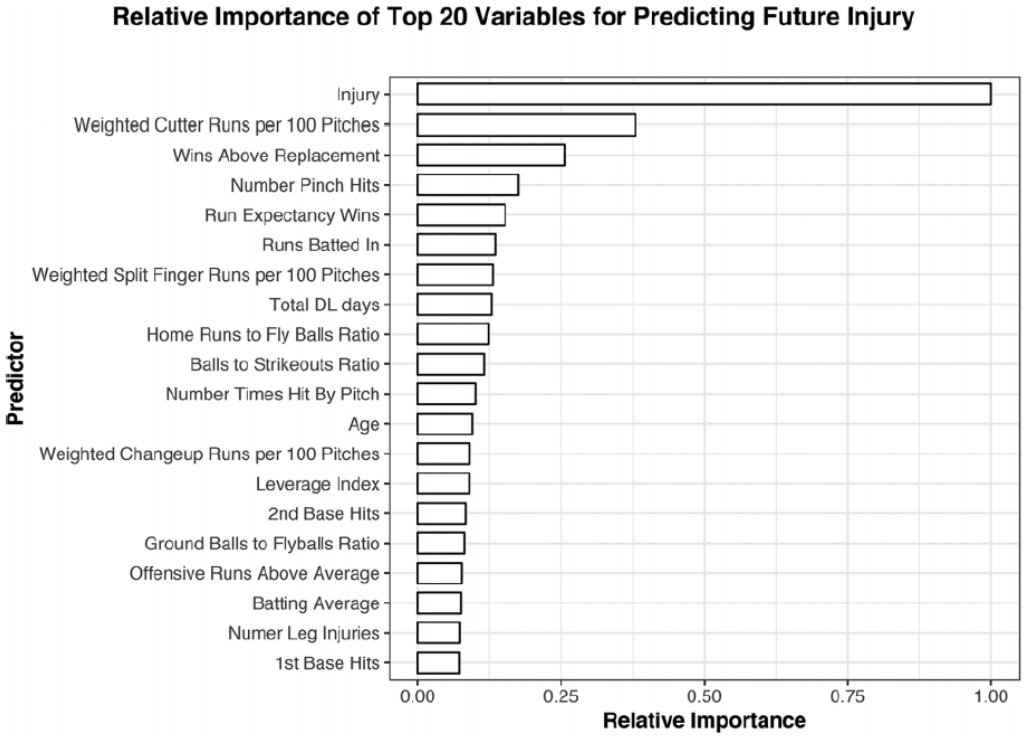

Figure 9: Variables ranked by relative importance for predicting future injuries among position players [33]

In conclusion, this study demonstrates that advanced ML models have the potential to predict future injuries and determine their anatomical sites in MLB position players [33].

Finally, in the research of Manoj, Prashant and Parikh, in 2018, a prediction was made to predict the champion of 2017 American League Baseball Championship. They used four different attributes: home/away, day/night, division, ranking. The results depicted that out of the 15 teams in the league, the one that had the most chances to win it according to the Analytical hierarchy process (AHP) was Kansas City Royals with an accuracy of approximately 62% [30].

3.1.4 Volleyball

Volleyball stands out as a globally beloved sport. This team game involves two opposing teams, each comprising six players, separated by a net. The primary goal of volleyball is to score points by hitting the ball effectively onto the opposing team's court. To

beat their opponents and gain victory, players use a combination of serving, passing, setting, spiking, and blocking skills [34].

In 2021, Leeuw et al used some machine learning algorithms to monitor injuries in elite volleyball players. Due to the danger of possible injuries, athletes must maximize their training in order to improve their physicality. This study implements the technique Subgroup Discovery to predict the injury risk based on wellness indicators and training load. The study shows that the most important factor in preventing injuries is the tracking of jump load [35].

Suda et al, in 2019, developed a method for predicting the trajectory of a volleyball toss 0.3 seconds before the actual toss by observing the setter player's action. The approach compares 3D data from Kinect and OpenPose and is evaluated between two players. The technique can be utilized for live broadcasts as well as analyzing opponent player characteristics. The error for the ball trajectory for the two players was 20.0 cm and 24.4 cm in the training data and 24.7 cm and 29.2 cm in the test data [36].

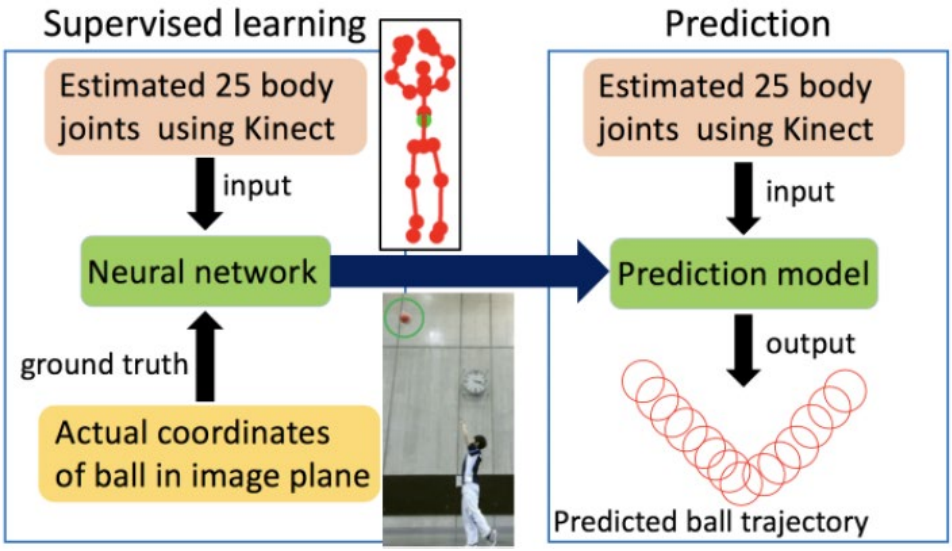

Figure 10: Process and outline of the proposed system [36]

Furthermore, Tian in 2021 used machine vision and wearable devices to optimize a volleyball motion estimation system. The program tries to improve volleyball tracking by minimizing motion blur and player misunderstanding. According to the research, the tracking accuracy of the different models were: 58% for Support Vector Machine (SVM), 69% for Conventional Videography Method (CVM), 76% for Mel-frequency cepstral coefficients (M-FCC) and 89% for the proposed Volleyball Motion Estimation Algorithm. Additionally, the use of machine vision and wearable sensors allows pre-

dicted imagery to be layered on live broadcasts, boosting the sports viewing experience even further. The paper also emphasizes the importance of data analysis and visualization in understanding opponent strategy in team sports. [37]

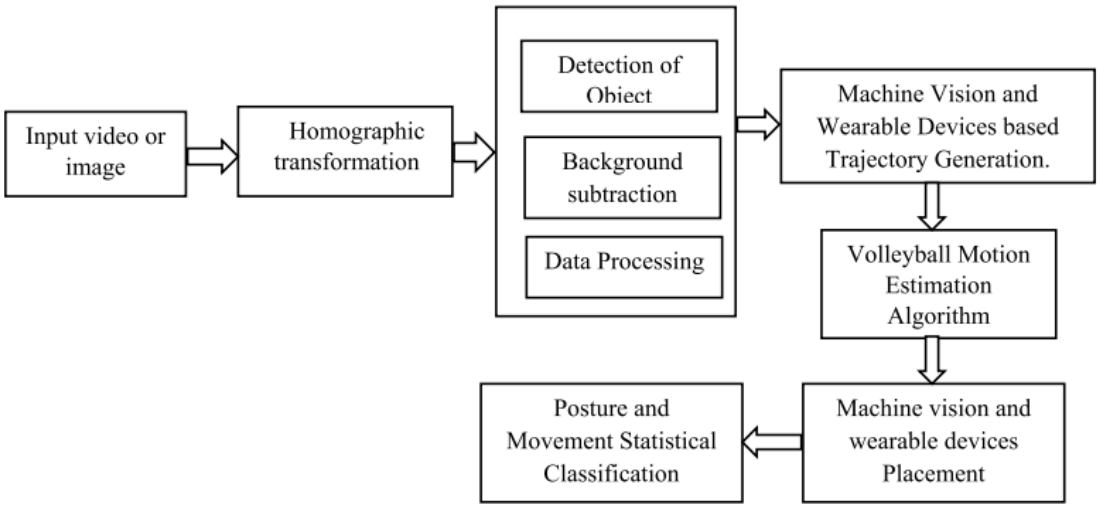

Figure 11: System process diagram [37]

Another interesting study was produced by Tümer and Koçer in 2017. They created an Artificial Neural Network (ANN) model to predict future team rankings in professional male volleyball leagues. They used data from league tables from 2013 to 2015, where the input parameters were wins, defeats, home wins, and away wins and the output parameter was the team rankings. The researchers examined 12 distinct models of 4, 6, 8, and 10 neurons in logsig, purelin, and tansig functions. A single hidden layer 4-neuron model with a 98% accuracy rate was the best ANN model [38].

3.1.5 Football

This section dives into the history of sports analytics in the world of football. It examines prior studies through time, providing details on the technique utilized. By reviewing these older papers, we intend to provide a comprehensive perspective on the growth of Sports Analytics in football, ultimately leading to a better understanding of its evolution and influence on the sport.

Because of the growing amount of data available in the sport, researchers have increasingly focused their efforts on football in recent years. This volume of data is an excellent opportunity to increase coaching staff decision-making capabilities, making football decisions more reliable. Following, there are some interesting football research publications.

Pariath, Shah, et al., in 2018, conducted research about player performance prediction in football related to overall performance value. For the first approach, separate models were produced depending on the position of the player. For this challenge, the linear regression algorithm obtained 84.34% accuracy, whereas for predicting a player's future market value based on performance, the algorithm achieved 91% accuracy [39].

The research of Baboota and Kaur in 2018 aimed at predicting football outcomes for English Premier League. The dataset contained 11 seasons, 9 used for training (2005 to 2014) and 2 for testing (2014 to 2016). One of the most influencing factors was the home/away attribute, which means if a team is playing in its stadium or not. Overall, some of the characteristics that makes predicting football outcomes difficult is the high occurrence of draws (25% in the dataset of testing). A variety of models was tested: Gaussian Naïve Bayes, Support Vector Machine, Random Forest, Gradient boosting. The last one produced the best results as the results are shown in the following figure [40]:

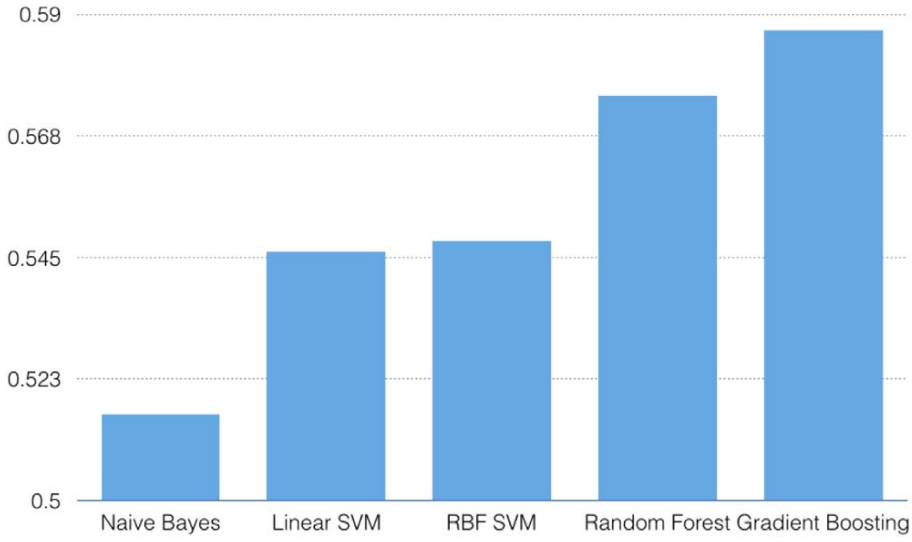

Figure 12: Mean test accuracy percentage of the different ML models [40]

In 2019, Apostolou and Tjortjis executed several experiments. One of them was to predict a player's position on the field. Using Random Forest and Sequential Minimal Optimization (SMO) with 10-fold cross-validation, the accuracy achieved was 81.5% by picking the most significant features. In the following figure the relevant confusion matrix is depicted [41]:

```
=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 24  0  9  0 |  a = FOR
  0 17  0  0 |  b = GK
  6  0 32  3 |  c = MID
  0  0  4 24 |  d = DEF
```

Figure 13: Confusion matrix [41]

Another experiment was that they tried to predict the goals a player will achieve next season based on the previous year data and specifically for 2 famous players, Lionel Messi and Luiz Suarez. Data was acquired by scraping while 4 ML algorithms were tested: Random Forest, Logistic Regression, MLP classifier and Linear SVC. As it is understood from the picture below, Random Forest was the best model in both players because it was closer to the actual number of goals of season 2017-2018 (Messi: 34 goals, Suarez: 25 goals) [41].

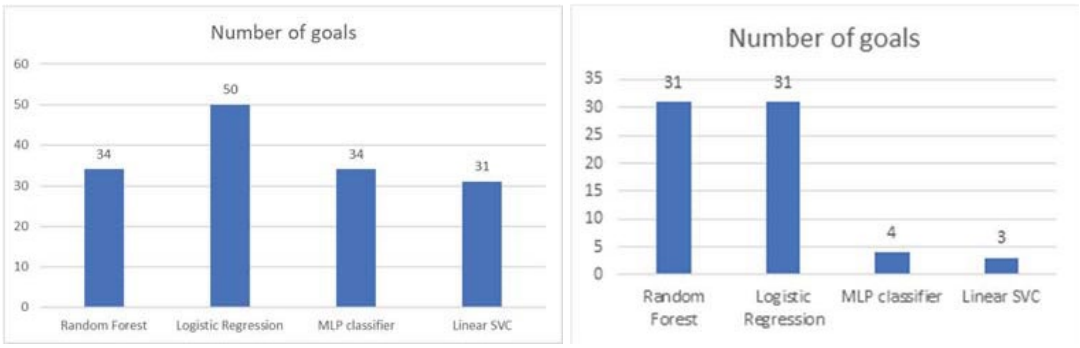

Figure 14: Number of goals for Messi and Suarez [41]

Finally, similar study was conducted to determine how many shots a player would take during a match. Once again, Random Forest delivered the greatest results, particularly for a specific match in which Messi made two shots and the model predicted 2.133 [41].

Pantzalis and Tjortjis, in 2019, experimented with two approaches: team performance prediction and player performance prediction for football. For the first experiment, they took two alternative tactics. The goal of the first technique was to predict whether a team would have a better place in the table for season 2017-2018 compared with the previous two seasons. Using Random Forest, the accuracy of this approach reached 70%. The second strategy involved simulating football matches for the 2018-2019 season with the purpose of categorizing the results as home victory, away win, or draw. The English Premier League had the highest match outcome accuracy (57%), while the Spanish La Liga had the lowest RMSE [42].

ACTUAL TABLE		PREDICTED TABLE	
1. Barcelona	87	1. Barcelona	83
2. Atletico Madrid	76	2. Atletico Madrid	75
3. Real Madrid	68	3. Real Madrid	64
4. Valencia	61	4. Valencia	58
5. Sevilla	59	5. Sevilla	57
6. Getafe	59	6. Getafe	57
7. Espanyol	53	7. Real Betis	57
8. Athletic Bilbao	53	8. Eibar	57
9. Real Sociedad	50	9. Celta Vigo	57
10. Real Betis	50	10. Villarreal	55
11. Alaves	50	11. Athletic Bilbao	54
12. Eibar	47	12. Real Sociedad	54
13. Leganes	45	13. Leganes	54
14. Villarreal	44	14. Espanyol	51
15. Levante	44	15. Alaves	51
16. Celta Vigo	41	16. Levante	51
17. Valladolid	41	17. Valladolid	51
18. Girona	37	18. Girona	51
19. Huesca	33	19. Vallecano	50
20. Vallecano	32	20. Huesca	49

Figure 15: Actual vs Predicted table for Spanish La Liga, season 2018-2019 [42]

In the second experiment, they figured out which characteristics and moves throughout a game can influence a defender's rating. The dataset contained 59 central defenders of English Premier League for season 2016-2017. The model that was used was Multiple Linear Regression with Backward Elimination and it achieved 0.867 in R-Squared metric. The features that influenced more the performance of the denders were interceptions and clearances [42].

The public dataset from Wyscout is used in the study of Zeng and Pan, in 2021, to predict player positions based on sports performance and physiological characteristics. Six indicators (accuracy of shot, accuracy of simple pass, accuracy of glb, accuracy of defending duel, accuracy of air duel, accuracy of attacking duel) are chosen as input for training into a BP neural network. To evaluate hyperparameter pairings, the model employs k-fold cross-validation. The model achieved 77% accuracy [43].

Moreover, injuries in sports are a source of concern not only for individuals but also for teams and organizations. These injuries can have long-term effects on an athlete's career, team accomplishments, and general competitiveness. They frequently necessitate rehabilitation and recovery periods, which can have an impact on team chemistry and strategic decision making. Injuries can also have a big impact on the outcomes of matches and entire seasons, emphasizing their importance in the world of sports.

In 2020, Oliver et al. conducted research to determine how effective ML was in identifying injury risk characteristics in elite male youth football players. The sample included 355 athletes who performed a neuromuscular test, which included anthropometric measurements, single leg countermovement jump, and tuck jump assessments. According to the findings, the most common factors to injury were asymmetry in the SLCMJ, 75% Hop, Y-balance, tuck jump knee valgus, and anthropometrics [44].

Martins et al. published a study in 2022 that used body composition characteristics and physical fitness assessments to predict injury risk in professional football players. The study included 36 male players from the First Portuguese Soccer League in the 2020-2021 season. There were 22 different attributes with the number of injuries every season as the target variable. Sectorial positions, body height, sit-and-reach performance, one minute number of push-ups, handgrip strength, and 35 minutes linear speed were the strongest indicators of injury risk, according to the net elastic analysis. Ridge was the most accurate model, with an error of RMSE = 0.591 [45].

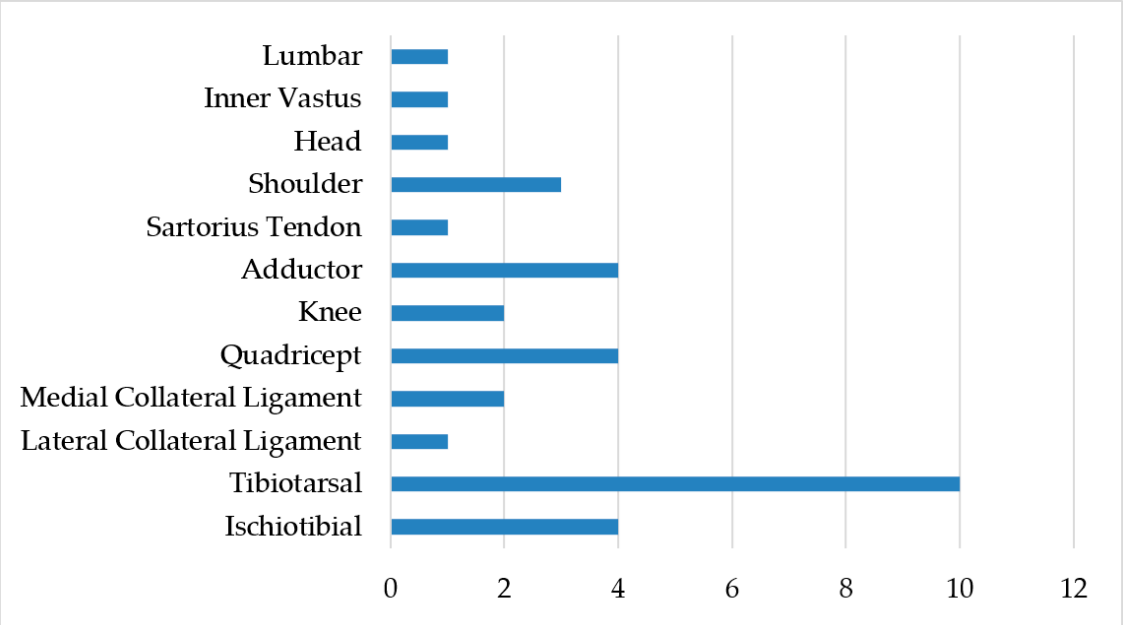

Figure 16: Injury frequency by specific location [45]

As previously stated, football clubs frequently use wearable gadgets during training and matches, a trend fueled by their data-gathering potential for players' physical attributes. Professional firms conduct statistical analyses and share their insights with football clubs to help with player tracking and strategic decision-making. Wearable gadgets are becoming increasingly important in football decision-making.

Using wearable devices and recurrent neural networks, the research of Feng et al. in 2021, proposes a smart football player health prediction algorithm. The algotithm analyzes health data from 100 players to extract deep patterns and predict health consequences. The accuracy rate is 81%, demonstrating its usefulness and supremacy in the competitive sports market [46].

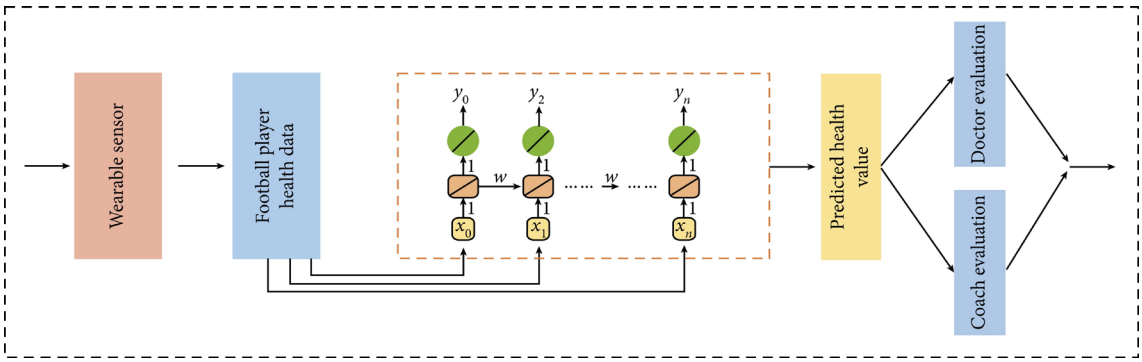

Figure 17: Schematic diagram of the algorithm [46]

Pilka et al in 2023 tried to create a decision-making model to predict lower-body injuries in male football players due to over- or undertraining using wearable devices. Injury prediction remains a tough subject due to individual biological variances in the body and each player's phychophysical condition. Catapult wearable global positioning trackers [47] were used to collect data during both exercise and game activities [48].

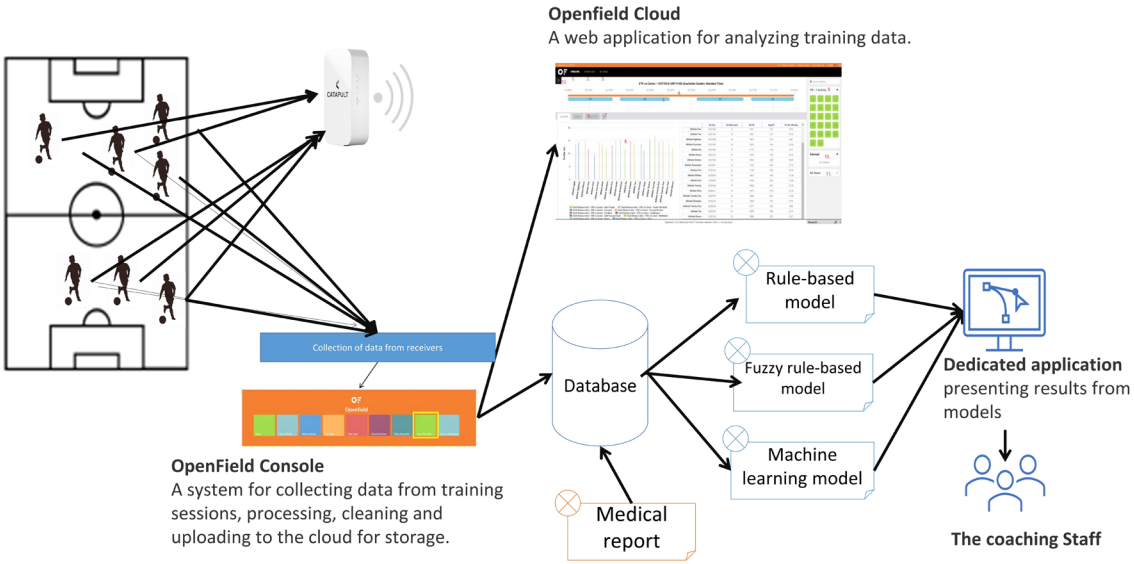

Figure 18: The process of collecting and processing training activity data from Catapult GPS wearable devices [48]

Furthermore, a rule-based method, a fuzzy rule-based method, and the XGBoost algorithm were also examined as decision-making models. The XGBoost algorithm produced the highest accuracy of 90% [48].

4 Predictive Modeling Techniques in Sport Analytics

In this chapter, the prediction models that are the core of the research will be analyzed. These are: Linear Regression, Ridge Regression, Random Forest, Gradient Boosting, XGBoost and Multilinear Perceptron. These models are critical in achieving our goal on predicting player performance using football data.

4.1 Linear Regression Algorithm

Linear Regression is the most commonly used algorithms for predictive analysis. It investigates the relationship between two or more variables in machine learning. It is used for predicting the linear relationship between a dependent variable (target variable) and one or more independent variables. The aim is to find the best straight line or hyperplane in higher dimensions that captures the connection between the variables [49][50].

The most basic version is simple linear regression, in which there is one dependent and one independent variable. The straight line that shows the connection is known as regression line and it has two parameters: the slope - coefficient and the intercept. The equation is [50]:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

where y = the dependent variable, x = the independent variable, β_0 = the intercept, β_1 = the slope coefficient and ε = the error term.

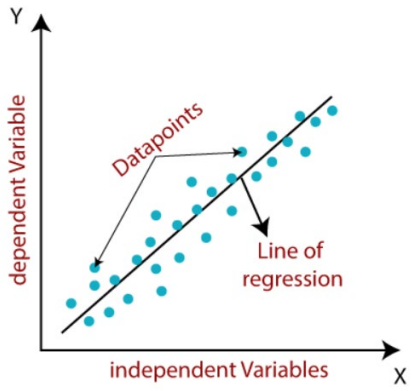

Figure 19: Linear regression schema [50]

The main objective of linear regression is to determine the values β_0 and β_1 that minimize the sum of squared deviations, between the actual and predicted values of the dependent variable. This approach is commonly known as " squares" regression [50].

In general, Linear Regression can be used for a variety of applications, including predicting sales and determining the relationship between temperature and energy use. Multiple linear regression (with several independent variables) or polynomial regression (for nonlinear relationships) can be used to adapt linear regression to tackle more complex issues.

4.2 Ridge Regression Algorithm

Ridge Regression is a linear regression algorithm which is capable of dealing with the issue of multicollinearity, which occurs when independent variables are highly correlated. It is also known as Tikhonov regularization named by its creator [51][52].

The algorithm uses the ordinary least squares approach by adding a regularization component to the cost function, commonly known as L2 regularization. This term penalizes large coefficients, preventing the model from learning from one variable and provides a balance. It works by decreasing the coefficients toward zero, making the model more stable and capable of dealing with correlated variables. Although this results in some bias, it increases the model's capability to work well with unseen data. Ridge Regression is especially useful in cases where ordinary linear regression may fail [52][53].

4.3 Random Forest Algorithm

Random forest belongs to the supervised ML algorithms and it is used for regression or classification. It was first introduced by Leo Breiman from University of California in

2001[54]. It's an ensemble learning method that combines multiple decision trees independent from each other to create a more robust and accurate model [55].

The steps for developing the random forest algorithm are analyzed below [55]:

1) Randomly select samples from the dataset.
2) Creation of decision trees for each sample by the algorithm. By this, the prediction result from each decision tree will be obtained.
3) Voting will be done for each anticipated result in this phase.
4) The most popular prediction result will be the outcome.

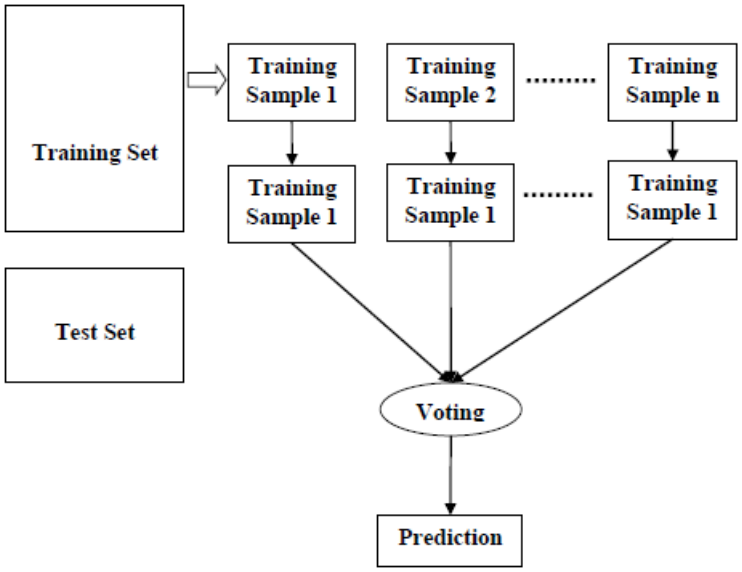

Figure 20: Random Forest algorithm [55]

To solve regression problems, random forest uses the mean of the individual trees. On the other hand, the output for the classification problems is the clss selected by the majority of decision trees. Random Forest has various advantages. Because it averages out errors and minimizes variance, it is less prone to overfitting than a single decision tree. Furthermore, tt is capable of handling a wide range of features, including numerical and categorical data. It offers efficient performance, high accuracy, and resilience to missing data, making it a valuable choice for a wide range of applications [54][56].

4.4 Gradient Boosting Algorithm

Gradient Boosting is an ensemble machine learning technique that develops a model by combining the predictions of numerous weak learners, usually decision trees. It is used for both regression and classification problems. The algorithm minimizes the errors of prior models by fitting new models to their residuals. The combination of these weak learners produces a more robust and accurate model. Furthermore, the algorithm gives focus on decreasing the bias and the variance, resulting in a good resistance to overfitting [57][58].

Further optimization techniques of Gradient Boosting have been introduced using libraries such as XGBoost, LightGBM and CatBoost. Gradient boosting has become a popular machine learning technique due to its adaptability, good prediction accuracy, and application across multiple domains.

4.5 XGBoost Algorithm

XGBoost, or Extreme Gradient Boosting, is a machine learning technique that is both efficient and powerful. It belongs to the gradient boosting algorithm family, which works by combining the predictions of numerous "weak" models (decision trees), to generate a significantly stronger predictive model [59][60].

XGBoost relies on decision trees as its foundational learners. These trees are frequently shallow, implying that they have limited depth, making XGBoost computationally efficient. XGBoost algorithm is lowering the danger of overfitting, which occurs when the model performs well on training data but poorly on new, unknown data. XGBoost provides several advantages. One of its primary characteristics is the ability to control overfitting using regularization approaches, it includes L1 and L2 penalties and biases of each tree [59][61]. In addition, XGBoost provides parallel computing. It makes it easier to scale up by utilizing multi-core systems or clusters [61][62].

XGBoost has a wide range of applications, including binary and multi-class classification, regression, ranking, recommendation systems, and anomaly detection.

4.6 Multilayer Perceptron Algorithm

A Multilayer Perceptron (MLP) is a form of artificial neural network constructed with several layers of artificial neurons. It is a feedforward neural network, which

means that information goes in only one direction, from the input layer to the output layer via the hidden layers. The following figure shows one hidden layer MPL [63].

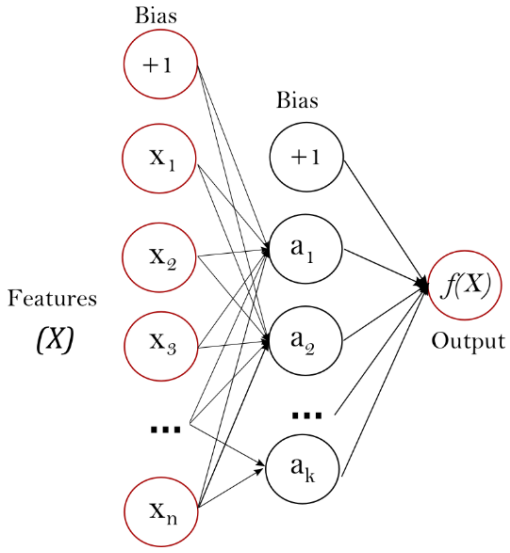

Figure 21: MLP with a single hidden layer [64]

The architecture of MLP consists of an input layer, one or more hidden layers and one output. The data is received by the input layer, with each node representing a feature or input variable. Then, the data flows through the hidden layers, to the output layer, where predictions or classifications are formed. The distinction between regression and classification is that the former uses a linear activation function for the output layer nodes, while the latter uses a softmax activation function. Weights are used to create connections between neurons. The alteration of the weights is achieved through the training phase of the algorithm. The most often utilized learning approach is backpropagation [63][64].

Some examples of the MPL applications are: image and speech recognition, natural language processing, regression analysis.

5 Methodology

This chapter describes the methodology's procedures and complications. It examines the full data collection process, from data scraping to data cleansing and feature engineering. The final objective is to demonstrate how the dataset was modified prior to the application of ML algorithms. The comparison of the models is based on the evaluation metrics of Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage (MAPE) and R squared that provide a thorough evaluation of our predictive model's performance.

5.1 Process Description

In order to achieve the major purpose of this dissertation, the analysis investigates various machine learning models. Linear Regression, Ridge Regression, Random Forest, Gradient Boosting, XGBoost and Multilinear Regression are the algorithms used. This chapter delves deeper into the steps that contributed to the effectiveness of the process.

The first step is the data collection. A suitable dataset must be developed in order for this prediction to work. For this reason, the data should be scraped from a reliable source in order to have more accurate findings. All of the data was collected from Sports Reference, which gives a wealth of statistics on athletes for a number of sports [1]. Football players from the 2017-2018 season through the 2022-2023 season were included in the extraction. The number of players exceeded 5000, and the total number of features were 36. The seasons 2018-2019 to 2021-2022 were used to train the ML algorithms, while the season 2022-2023 was used as a testing dataset. Each season was edited so that it contained data from the previous season, as it will be explained further in chapter Feature engineering.

The second step involved the pre-processing of the dataset. The dataset had a number of important adjustments to assure data cleanliness and dependability. To begin with, duplicates, null values, and noise were carefully deleted, resulting in a more refined dataset. Furthermore, some of the attributes were considered unnecessary for the analysis and they were systematically removed from the dataset. The dataset included a diverse range of football players from various countries who played in various clubs, leagues, and positions. However, the only criterion was the players' league. In this study, it was

chosen to only include players from four leagues: Bundesliga, Premier League, La Liga and Serie A. Each league had its own dataset. Afterwards, it was necessary to apply some prediction models in order to achieve the goal. As mentioned, python was implemented and specifically jupyter notebooks. More details about preprocessing are discussed in the relevant chapter.

Finally, the evaluation of the results was achieved by three metrics: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage (MAPE) and R squared. These metrics give us valuable insights for the reliability and efficacy of the models and they are analyzed in the next chapter.

The steps of the whole process are depicted in the diagram below:

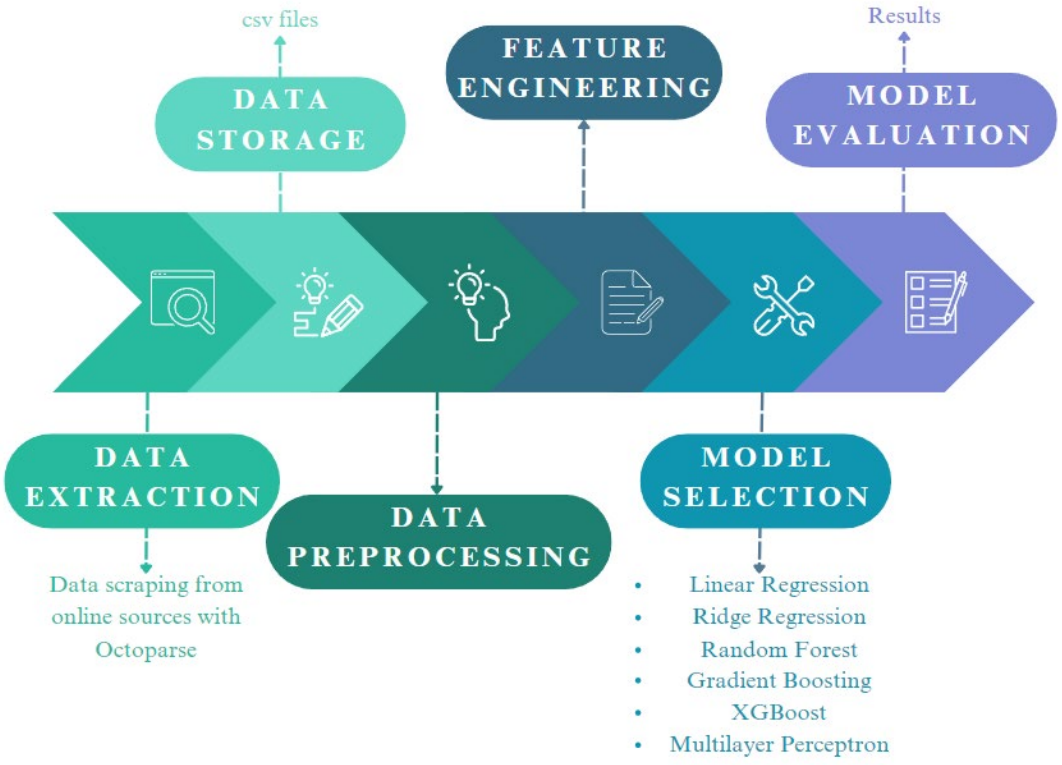

Figure 22: The flowchart of the process

5.2 Evaluation Metrics

5.2.1 Mean Absolute Error (MAE)

MAE is one of the most known accuracy metrics used for the evaluation of the performance of the ML algorithms. It is calculated as the average absolute difference among the model's predicted values and the data's true values [65][66]. Its mathematical expression is:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

where y_i = prediction, x_i = true value, n = total number of observations in the dataset

5.2.2 Mean Squared Error (MSE)

MSE is calculated by taking the squared difference between each predicted and actual value, adding all of these squared differences, and then dividing by the total number of data points. MSE's equation is the following [67]:

$$MSE = \frac{\sum_{i=1}^{n} |y_i - x_i|^2}{n}$$

5.2.3 Root Mean Square Error (RMSE)

RMSE is the square root of the mean squared error between the predicted and actual values of a model. The mathematical expression of this metric is [68][69]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |y_i - x_i|^2}{n}}$$

5.2.4 Mean Absolute Percentage (MAPE)

MAPE is calculated by the average percentage difference between predicted and actual values. A smaller MAPE indicated a more accurate model, whereas a larger indicates a less accurate model. However, MAPE has some restrictions, such as that it is sensitive to outliers and that is invalid when numbers are zero. The expression is the following [70]:

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{x_i - y_i}{x_i} \right| * 100\%$$

In this study MAPE has some limitations. As it is already mentioned, MAPE faces challenges in the presence of null values within the target variable. This limitation can result in inaccurate calculations. As a result, MAPE will be selectively depicted only in the results containing the top 30% players.

5.2.5 R-squared

R^2 (coefficient of determination) is used to assess the goodness of fit of a regression model. It expresses how effectively the independent variable(s) explain the variation in the dependent variable. R = 1 shows that the model explains all of the dependent variable's variability around its mean [71][72].

$$R^2 = 1 - \frac{RSS\ (sum\ of\ squares\ of\ residuals)}{TSS\ (total\ sum\ of\ squares)}$$

5.3 Data Collection

The crucial feature of this dissertation is the process of data collection. Numerous websites provide football statistics for clubs and players. As a result, it is critical to ensure the data's legitimacy, as any inaccuracies would compromise the precision of the results.

The dataset was obtained from Sports Reference, a well-known organization for having massive amounts of data across various sports such as football, basketball, baseball and hockey. It is constantly updated and contains a wide range of information about approximately 100,000 players across more than 100 competitions, including scores, statistics, and historical context [1].

The scraping tool that was used was Octoparse [73]. Octoparse is a web scraping tool and platform that allows users to extract data from websites. In the following figure, the steps of scraping are shown:

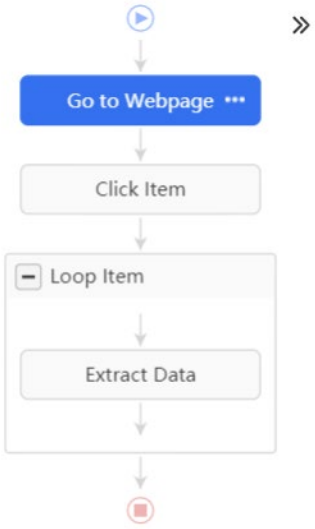

Figure 23: Steps for scraping - Octoparse

To delve deeper, the dataset used in this dissertation were scraped from this platform and concerned football players who competed in seasons 2017-2018 to 2022-2023. Each season featured a large number of players. This procedure resulted in four unique CSV files for each league having information on over 5000 football players. The dataset includes the following features, which are listed below:

Table 5-1: The attributes of the dataset

GLOSSARY	
COLUMN	**DESCRIPTION**
Player	Name of the player
Nation	Nationality of the player
Pos	Position most commonly played by the player
Squad	Club the player is currently playing
Age	Age of the player at season's start
Born	Player's year of birth
MP	Matches played by the player
Starts	Game or games started by the player
Min	Minutes played by the player
90s	Minutes played divided by 90
GLs	Goals scored or allowed

Ast	Assists
G+A	Goals and Assists
G-PK	Non-penalty goals
PK	Penalty Kicks made
Pkatt	Penalty Kicks attempted
CrdY	Yellow cards
CrdR	Red cards
xG	Expected goals
npxG	Non-Penalty Expected Goals
xAG	Expected Assisted Goals
npxG+xAG	Non-Penalty Expected Goals plus Assisted Goals
PrgC	Progressive Carries
PrgP	Progressive Passes
PrgR	Progressive Passes Rec
GLs per 90'	Goals scored per 90 minutes
Ast per 90'	Assists per 90 minutes
G+A per 90'	Goals and Assists per 90 minutes
G-PK per 90'	Goals minus penalty kicks made per 90 minutes
G+A-PK per 90'	Goals plus Assists minus Penalty Kicks made per 90 minutes
xG per 90'	Expected Goals per 90 minutes
xAG per 90'	Expected Assisted Goals per 90 minutes
xG+xAG per 90'	Expected Goals plus Assisted Goals per 90 minutes
npxG per 90'	Non-Penalty Expected Goals per 90 minutes
npxG+xAG per 90'	Non-Penalty Expected Goals plus Assisted Goals per 90 minutes

The shape of the dataset without applying any techniques is the following:

Rank	Player	Nation	Pos	Squad	Age	Born	MP	Starts	Min	s90	Gls	Ast	G_PLUS_A	G_MINUS_PK	PK	PKatt	CrdY	CrdR	xG	npxG	xAG	npxG_PLUS_xAG
1	Issah Abbas	gh GHA	FW	Mainz 05	19	1998	1	0	9	0.1	0	0	0	0	0	0	0	0	0	0	0	0
2	David Abraham	ar ARG	DF	Eint Frankfurt	32	1986	17	17	1,420	15.8	0	0	0	0	0	0	2	0	0.8	0.8	0.4	1.1
3	Amir Abrashi	al ALB	MF	Freiburg	28	1990	10	9	790	8.8	0	0	0	0	0	0	5	0	0.2	0.2	0.1	0.3
4	Tyler Adams	us USA	MF	RB Leipzig	19	1999	10	8	721	8	0	2	2	0	0	0	0	0	0.1	0.1	1	1.1
5	Antonis Aidonis	de GER	DF	Stuttgart	17	2001	2	1	80	0.9	0	0	0	0	0	0	1	0	0	0	0	0
6	Manuel Akanji	ch SUI	DF	Dortmund	23	1995	25	25	2,205	24.5	1	0	1	1	0	0	1	0	0.5	0.5	0.6	1.1
7	Chadrac Akolo	cd COD	MF,FW	Stuttgart	23	1995	16	5	525	5.8	0	1	1	0	0	0	0	0	0.9	0.9	0.5	1.4
8	Kevin Akpoguma	ng NGA	DF	Hoffenheim	23	1995	8	5	499	5.5	0	0	0	0	0	0	2	0	0	0	0	0
9	Kevin Akpoguma	ng NGA	DF	Hannover 96	23	1995	4	4	313	3.5	0	0	0	0	0	0	2	0	0	0	0	0
10	David Alaba	at AUT	DF	Bayern Munich	26	1992	31	29	2,530	28.1	3	3	6	3	0	0	4	0	3.2	3.2	4.4	7.6

Figure 24: Final database (some of the features)

However, the dataset used for the ML algorithms has a different shape than this. Many changes were made to transform the dataset to the proper format. These changes will be analyzed in the next chapters.

5.4 Pre-processing

The primary focus during the early stages of data preprocessing was thorough data cleaning. This demanded a thorough study of the data's format in order to eliminate any potential sources of "noise." Duplicates, null values, missing entries, and outliers are all part of the noise. However, the dataset that was scraped did not contain any of these things.

Following that, the object type columns were converted to strings, so that can more easily be used in the prediction models. Attributes ‘Rank’ and ‘s90’ were also deleted because they did not add useful information.

Next, it was observed that some players appeared in more than one football clubs for the same season. For this reason, it was decided to calculate the average value of a player in this situation for the arithmetic columns and create a combined string name in the column ‘Squad’ with the names of the teams. One crucial step of this phase was that the dataset should include players who had participated in all of the seasons. So, players that have not played in all of the 6 seasons were removed from the dataset. As a result, the data had a significant reduction. Specifically:

- **Bundesliga:** from 1185 unique players to 109 players
- **Premier League:** from 1298 unique players to 112 players
- **La Liga:** from 1431 unique players to 97 players
- **Serie A:** from 1441 unique players to 106 players

On top of that, an additional implementation, was a dataset that contained the players from all leagues, 424 players in total. To have a distinction between the players and their league, a new column ‘League’ was added, which had coding numbers for each

league (League = 1 for Bundesliga, League = 2 for Premier League, League = 3 for La Liga, League = 4 for Serie A).

After this distinction, two different versions were used. The first one contained all the players and the second contained the 30% top players that have played in the last season based on goal performance.

Then, the cases that developed were regarding which features were included in the algorithms. The process of reducing the number of features by keeping the most important information is called dimensionality reduction [74].

In **case 1**, there are 10 columns selected, representing the attributes most strongly correlated with the target variable 'Gls.' These selections are based on the Pearson correlation coefficient. Pearson correlation coefficient assesses the strength of a two-variable linear relationship. It has a value ranging from -1 to +1. -1 indicates total negative correlation between the features, 0 indicated no correlation and +1 indicates total positive correlation [75]. The features of case 1 are showed in the following figure.

```
x_columns =  ['xG', 'npxG', 'npxG_PLUS_xAG', 'xG_90', 'Previous_Gls', 'npxG_90', 'G_MINUS_PK', 'xG_PLUS_xAG_90',
              'G_PLUS_A', 'PrgR']
```

Figure 25: Case 1 attributes

Then, for **case 2**, we calculated for each pair of attributes the percentage of correlation. So, from each highly correlated pair, we kept one of the two columns.

```
x_columns = ['Nation','Pos','Squad', 'Age', 'MP','Ast','PK','CrdY','CrdR','PrgC','PrgP','PrgR','Previous_Gls']
```

Figure 26: Case 2 attributes

Final, **case 3** contains all the available columns of the dataset.

```
x_columns = ['Nation', 'Pos', 'Squad', 'Age', 'Born', 'MP','Starts', 'Min', 'Ast', 'G_PLUS_A', 'G_MINUS_PK', 'PK',
             'PKatt', 'CrdY', 'CrdR', 'xG', 'npxG', 'xAG', 'npxG_PLUS_xAG', 'PrgC','PrgP', 'PrgR', 'Gls_90', 'Ast_90',
             'G_PLUS_A_90', 'G_MINUS_PK_90', 'G_PLUS_A_MINUS_PK_90', 'xG_90', 'xAG_90', 'xG_PLUS_xAG_90', 'npxG_90',
             'npxG_PLUS_xAG_90','Previous_Gls']
```

Figure 27: Case 3 attributes

Notable observation is that in the dataset that contained the total number of players for all leagues, the additional column ‘League’ was added to its case.

To better understand the cases the following diagram is presented:

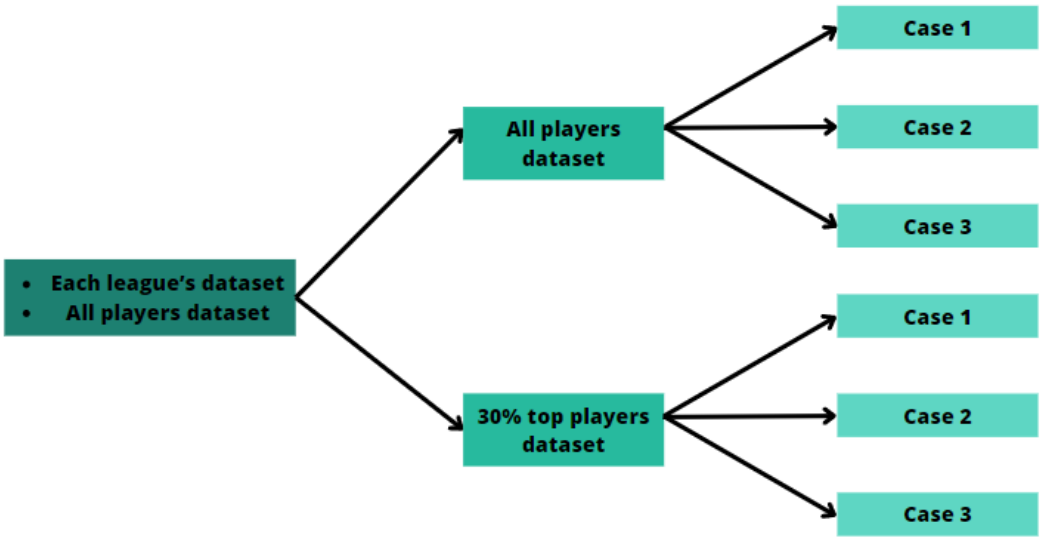

Figure 28: Diagram of the cases applied for each league

5.5 Feature Engineering

As explained earlier, the scope was to train the algorithms using the 4 seasons (2018-2019 to 2021-2022) and then evaluate them using as test the last season (2022-2023). However, if we kept this format, then the results would be exceptionally good and not realistic because we would keep statistics that have a big influence in how many goals a football player will achieve, like expected goals or assists etc.

For this reason, a different approach was applied. In order to avoid using statistics from current season, the dataset was converted to contain historical data. Each row contained past statistics. For example, the algorithm will be trained to predict the number of goals a player will achieve for season 2018-2019 using statistics from the previous season which is 2017-2018. Furthermore, an additional column 'Previous_Gls' was created, that shows the number of goals a player achieved for season 2017-2018, while column 'Gls' indicates the number of goals of a player for season 2018-2019.

The goal is to predict the number of goals of a player for season 2022-2023 using data from season 2021-2022. This is known as season lag-features. Lag features are used to identify patterns in data that can be used to make accurate forecasts or explain the behavior of a time series. They use the value of a variable at a previous time point into the model at the current time point.

The code applied is shown below:

```
#sort dataset by player name and age in order the dataset to be sorted by player name and by season in ascending order
final_data = final_data.sort_values(by=['Player','Age'])
```

```
#create a new column with name previous goals
# For example we have season 2018-2019: it will contain data from 2017-2018
# Previous_Gls -> from season 2017-2018
# Gls -> from season 2018-2019

final_data['Previous_Gls'] = final_data['Gls']
```

```
for i in range(len(final_data) - 1):
    current_player = final_data.iloc[i]['Player']
    next_player = final_data.iloc[i + 1]['Player']

    if current_player != next_player:
        final_data.iat[i, 9] = -1
    else:
        final_data.iat[i, 9] = final_data.iat[i + 1, 9]
```

```
#remove the records for the sixth season
cropped = final_data.loc[final_data['Gls']!=-1]
```

Figure 29: Code for feature engineering

A further refinement involved categorizing the dataset into what we called the '30% top players.' This categorization was determined based on the players' scoring performance in the last season. Consequently, the final dataset comprised players who fell within the top 30% quartile. The threshold for this classification was stored as a variable. The corresponding code is available in Figure 30.

```
# Display unique player IDs and player names from the cropped DataFrame
unique_players = cropped[['player_id', 'Player']].drop_duplicates()
```

```
# Keep only the last appearance for each player
last_appearance_by_player = cropped.groupby('Player').last().reset_index()

# Step 1: Determine the threshold for the top 50% based on total goals
total_goals_by_player_last = last_appearance_by_player.groupby('Player')['Gls'].sum()
threshold = total_goals_by_player_last.quantile(0.7)

# Step 2: Filter the DataFrame to include only data for the top players
top_players_last = total_goals_by_player_last[total_goals_by_player_last >= threshold].index
cropped_top_players_last = last_appearance_by_player[last_appearance_by_player['Player'].isin(top_players_last)]
```

```
print(threshold)
```

```
# Count the number of rows in model_data_top_players
num_rows_model_data_top_players = cropped_top_players_last.shape[0]
print(num_rows_model_data_top_players)
```

```
# Assuming 'Player' is the column representing player names
# You may need to adjust the column names based on your actual DataFrame
filtered_cropped = cropped[cropped['Player'].isin(cropped_top_players_last['Player'])]
cropped = filtered_cropped
```

Figure 30: Code for the 30% top players dataset

On top of that, the split of the dataset is depicted in the following figure.

```
# Columns for x_train

# x_columns =  ['xG', 'npxG', 'npxG_PLUS_xAG', 'xG_90', 'Previous_Gls', 'npxG_90', 'G_MINUS_PK', 'xG_PLUS_xAG_90',
#              'G_PLUS_A', 'PrgR']

# x_columns = ['Nation','Pos','Squad', 'Age', 'MP','Ast','PK','CrdY','CrdR','PrgC','PrgP','PrgR','Previous_Gls']

x_columns = ['Nation', 'Pos', 'Squad', 'Age', 'Born', 'MP','Starts', 'Min', 'Ast', 'G_PLUS_A', 'G_MINUS_PK', 'PK',
             'PKatt', 'CrdY', 'CrdR', 'xG', 'npxG', 'xAG', 'npxG_PLUS_xAG', 'PrgC','PrgP', 'PrgR', 'Gls_90', 'Ast_90',
             'G_PLUS_A_90', 'G_MINUS_PK_90', 'G_PLUS_A_MINUS_PK_90', 'xG_90', 'xAG_90', 'xG_PLUS_xAG_90', 'npxG_90',
             'npxG_PLUS_xAG_90']

# Create empty dataframes for x_train and y_train
x_train = pd.DataFrame(columns=x_columns)
y_train = pd.DataFrame(columns=['Gls'])

# Columns for x_test
x_test = pd.DataFrame(columns=x_columns)
y_test = pd.DataFrame(columns=['Gls'])

# Iterate over unique player_ids in the dataframe
for player_id in cropped['player_id'].unique():
    # Extract rows for the current player
    player_rows = cropped[cropped['player_id'] == player_id]

    # Take the first 4 rows for x_train and y_train
    x_train = pd.concat([x_train, player_rows.head(4)[x_columns]])
    y_train = pd.concat([y_train, player_rows.head(4)[['Gls']]])

    # Take the last row for x_test and y_test
    x_test = pd.concat([x_test, player_rows.tail(1)[x_columns]])
    y_test = pd.concat([y_test, player_rows.tail(1)[['Gls']]])

# Reset index for the resulting dataframes
x_train.reset_index(drop=True, inplace=True)
y_train.reset_index(drop=True, inplace=True)
x_test.reset_index(drop=True, inplace=True)
y_test.reset_index(drop=True, inplace=True)
```

Figure 31: Code for splitting the dataset

Furthermore, Grid search was used for all the algorithms to find the best possible results. Grid search is a machine learning hyperparameter technique method that systematically searches for the best combination of hyperparameter values for a given model [76].

Moreover, feature importance was an additional part, which refers to determining the contribution of each input variable to the model's predictions. In summary, the positive and negative values represent the strength of each feature's influence on the expected outcome. Positive values suggest a positive impact, while negative values suggest a negative impact [77]. All algorithms provided feature importance scores, except for MLP.

```
# Get feature importances
feature_importance = model.coef_

print("\nFeature Importance (Coefficients):")
for feature, importance in zip(x_train.columns, feature_importance):
    print(f"{feature}: {importance}")
```

Figure 32: Code for feature importance

Also, for each algorithm's prediction we rounded the results, because the number of goals a player achieves is always an integer. Finally, metrics were computed for both the training and testing sets to facilitate a comprehensive comparison.

6 Results

This chapter examines the results of each algorithm for each different dataset. Following that, we will do a comparative analysis to discover which one performs better. The most essential indicator is MAE, which shows how close the predictions are to the actual numbers. The influence of the rest of the metrics is equally important. However, MAPE encounters issues when there are null values in the target variable, leading to inaccurate calculations. Consequently, it will only be showcased in the results of the dataset of the top 30% players. Lastly, the feature importance values of the best algorithm for each scenario are depicted in different tables in the Appendix.

6.1 Bundesliga Results

In this chapter, the results of Bundesliga are presented. This dataset only includes players who remained members of Bundesliga teams for the whole six-season period, from 2017-2018 to 2022-2023.

6.1.1 All players dataset

In this sub-chapter, the three different cases based on attributes selection will be presented. Notably, the standard deviation for this dataset was calculated to 3.24 goals, indicating how much the goal-scoring performances vary. The total number of players sums up to 109.

Case 1

We can obtain the following findings by using all the algorithms. These results are obtained by using the features that are available in Figure 26: Case 1 attributes.

Table 6-1: Performance results for Bundesliga case 1 (all players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*R^2 test / R^2 train*
Linear Regression	1.80 / 1.50	6.74 / 4.96	2.60 / 2.23	0.38 / 0.52
Ridge Regression	1.80 / 1.50	6.37 / 4.99	2.52 / 2.23	0.41 / 0.52
Random Forest	1.84/ 1.27	7.01 / 3.20	2.65 / 1.79	0.35 / 0.69
Gradient Boosting	1.90 / 1.21	7.41 / 2.59	2.72 / 1.61	0.32 / 0.75
XGBoost	1.81 / 1.35	6.74 / 3.73	2.60 / 1.93	0.38 / 0.64
MLP	1.82 / 1.51	6.56 / 5.03	2.56 / 2.24	0.40 / 0.52

An initial observation is that error values in Table 6-1 are quite small, implying an overall good performance for our models. As it seems, the best algorithm based on all metrics is the Ridge Regression algorithm. Furthermore, by observing MAE and RMSE values between train and test results, it is understood that they are relatively close. This means that the models predict well even in cases where they do not know.

Lastly, feature importance needs to be discussed. General observations about the Ridge Regression algorithm are:

- Expected goals per 90 min. (xG_90) appears to be the most crucial predictor, while non-penalty expected goals per 90 min. (npxG_90) is the second one.
- Attributes non-penalty expected goals (npxG), non-penalty goals (G_MINUS_PK) and goals plus assists (G_PLUS_A) have a negative influence in the results.

Case 2

Case 2 contains the attributes that are depicted by Figure 26: Case 2 attributes. The following results were obtained for this case.

Table 6-2: Performance results for Bundesliga case 2 (all players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*R^2 test / R^2 train*
Linear Regression	1.76 / 1.65	4.98 / 5.69	2.23 / 2.39	0.54 / 0.45
Ridge Regression	1.80 / 1. 65	5.08 / 5.74	2.25 / 2.40	0.53 / 0.45
Random Forest	1.76/ 1.18	5.29 / 2.65	2.30 / 1.63	0.51 / 0.75
Gradient Boosting	1.82 / 1.21	5.25 / 2.64	2.29 / 1.63	0.52 / 0.75
XGBoost	1.71 / 1.03	5.41 / 2.28	2.33 / 1.51	0.50 / 0.78
MLP	1.78 / 1.65	5.04 / 5.71	2.24 / 2.39	0.54 / 0.45

In this case, the best algorithm based on MAE is XGBoost. However, Linear and Ridge Regression have lower values on metrics MSE and RMSE. Generally, in this case it seems that the results for all the metrics are a bit better than the results of the previous case. Additionally, if we observe the values of the metrics between the train and test results, we will see that they are close to one another. This complies that the algorithms do not overfit.

Finally, the significance of each feature must be explored. The most essential feature is previous goals (Previous_Gls) with a big difference in comparison to others. Also, the attribute that does not carry any significant importance is nation.

Case 3

This case contains almost all the features of the original dataset (Figure 27: Case 3 attributes). The results are depicted below:

Table 6-3: Performance results for Bundesliga case 3 (all players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*R^2 test / R^2 train*
Linear Regression	1.89 / 1.45	8.29 / 4.59	2.88 / 2.14	0.24 / 0.56
Ridge Regression	1.80 / 1. 48	6.77 / 4.77	2.60 / 2.18	0.38 / 0.54
Random Forest	1.81/ 0.86	6.40 / 1.37	2.53 / 1.17	0.41 / 0.87
Gradient Boosting	1.95 / 1.30	7.04 / 2.89	2.65 / 1.70	0.35 / 0.72
XGBoost	1.80 / 1.28	6.37 / 3.25	2.52 / 1.80	0.41 / 0.69
MLP	1.96 / 1.60	6.19 / 5.55	2.49 / 2.36	0.43 / 0.47

In this case, the best algorithm based on MAE is Ridge Regression. However, it's worth mentioning that MLP presents good results in MSE and RMSE. In this scenario, it is evident that certain algorithms produced exceptionally low error values, signaling a potential issue of minor overfitting in these specific models. These values reflect the difference between the training and test sets.

Finally, the three most important features in this dataset are: expected goals per 90 min. (xG_90), expected plus assisted goals per 90 min. (xG_PLUS_xAG_90) and non-penalty expected goals per 90 min. (npxG_90).

6.1.2 Top 30% players dataset

The original dataset was reduced to 34 players from 109 players that was originally. The outcomes of this implementation revealed that a player needed to score a minimum of 3 goals (threshold) to secure a spot in the final dataset. Additionally, in this implementation, the MAPE metric was employed, where a lower value signifies enhanced performance. The dataset's standard deviation, calculated at 3.79 goals, provides a measure of the variability within the data.

Case 1

The subsequent outcomes were derived from this scenario.

Table 6-4: Performance results for Bundesliga case 1 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	2.38 / 1.96	8.84 / 7.19	2.97 / 2.68	0.40

Ridge Regression	2.27 / 1.97	8.85 / 7.24	2.86 / 2.69	0.38
Random Forest	2.58/ 1.78	10.2 / 5.63	3.20 / 2.37	0.39
Gradient Boosting	2.43 / 1.63	9.27 / 4.49	3.04 / 2.12	0.36
XGBoost	2.83 / 1.25	11.7 / 2.99	3.43 / 1.73	0.44
MLP	2.35 / 1.96	8.50 / 7.25	2.92 / 2.69	0.40

By observing the results, it becomes evident that the Ridge Regression algorithm consistently outperforms others. The small difference between the values of the train and test metrics suggests the absence of overfitting. Moreover, based on Table 8-1: Feature importance values for case 1, it is notable that the model heavily relied on the expected plus assisted goals per 90 min. (xG_PLUS_xAG_90) to make the predictions. Important negative influence to the predictions, had the non-penalty expected goals attribute (npxG).

Case 2

The results for this case were as follows:

Table 6-5: Performance results for Bundesliga case 2 (30% top players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***MAPE test***
Linear Regression	1.90 / 2.04	5.84 / 7.13	2.42 / 2.67	0.31
Ridge Regression	1.93 / 2.02	5.53 / 7.36	2.35 / 2.71	0.32
Random Forest	1.71/ 1.02	4.38 / 1.73	2.09 / 1.31	0.29
Gradient Boosting	1.83 / 1.62	5.46 / 4.12	2.34 / 2.03	0.29
XGBoost	1.96 / 0.56	5.72 / 0.50	2.39 / 0.71	0.35
MLP	2.03 / 2.04	6.47 / 7.31	2.54 / 2.70	0.34

The metrics linked with Random Forest consistently display superior values across the majority of parameters. A MAPE of 0.29 indicates that, on average, the predictions made by the model deviate from the actual values by approximately 29%. In other words, the model's predictions, on average, have an error rate of 29% when compared to the true values. Lower MAPE values generally indicate better accuracy. Moreover, the comparison of results between the training and testing sets suggests that the models perform effectively on new, unseen data, with the exception of the XGBoost algorithm, which exhibits signs of overfitting.

Like previously, in case 2, previous goals (Previous_Gls) is the most influenced attribute.

Case 3

The results of this case are depicted below:

Table 6-6: Performance results for Bundesliga case 3 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	2.67 / 1.74	11.24 / 5.37	3.35 / 2.32	0.45
Ridge Regression	2.20 / 1.98	7.07 / 7.04	2.66 / 2.65	0.37
Random Forest	2.28 / 0.85	7.25 / 1.28	2.69 / 1.13	0.38
Gradient Boosting	2.17 / 1.60	8.60 / 4.18	2.93 / 2.05	0.30
XGBoost	2.78 / 1.22	11.16 / 2.76	3.94 / 1.66	0.44
MLP	2.19 / 2.02	6.79 / 7.48	2.61 / 2.73	0.37

The metrics presented in the table highlight that Gradient Boosting model stands out as the top performer in terms of MAE. Additionally, the MLP algorithm demonstrates particularly strong results in both MSE and RMSE metrics. Moreover, the comparison of outcomes between the training and testing datasets depicts good performance on unseen data for the majority of the models. Nevertheless, it's important to highlight that both the XGBoost and Random Forest algorithms exhibit signs of overfitting.

Concerning feature importance, around 50% of the influence is attributed to the expected plus assisted goals per 90 min (xG_PLUS_xAG_90) feature. The remaining 50% is distributed among the other 32 features.

6.2 Premier League Results

This chapter unveils the outcomes of the Premier League, providing a comprehensive analysis of the results. The dataset exclusively includes players who remained in a Premier League's team throughout all six seasons.

6.2.1 All players dataset

The standard deviation for this particular dataset was calculated to 4.93 goals. Following, we present all the different cases depending on feature selection. There are a total of 112 players.

Case 1

Case 1 uses the 10 most correlated features regarding the target variable calculated by Pearson correlation coefficient. These are: xG, npxGnpx, G_PLUS_xAG, xG_90, Pre-

vious_Gls, npxG_90, G_MINUS_PK, xG_PLUS_xAG_90, G_PLUS_A, PrgR. The first feature (expected goals - xG) was the most influential factor.

We can obtain the following findings by using all the algorithms.

Table 6-7: Performance results for Premier League case 1 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	2.10 / 1.94	11.04 / 8.18	3.32 / 2.86	0.50 / 0.67
Ridge Regression	2.03 / 2.01	10.72 / 8.72	3.27 / 2.95	0.51 / 0.65
Random Forest	2.04 / 1.76	10.79 / 6.34	3.28 / 2.52	0.51 / 0.74
Gradient Boosting	2.14 / 1.74	10.80 / 5.63	3.29 / 2.37	0.51 / 0.77
XGBoost	1.93 / 1.66	10.35 / 6.17	3.22 / 2.48	0.53 / 0.75
MLP	1.99 / 1.98	10.82 / 8.81	3.29 / 2.97	0.64 / 0.51

A first observation reveals that the outcomes in this scenario fall short when compared to those in the Bundesliga. Among all the algorithms assessed, the XGBoost algorithm emerges as the most effective based on various metrics. Notably, the differences between the training and test values are minimal for MAE and RMSE, yet slightly bigger for MSE.

Case 2

Case 2 contains the following features: Nation, Pos, Squad, Age, MP, Ast, PK, CrdY, CrdR, PrgC, PrgP, PrgR, Previous_Gls. Key insights regarding the evaluation of feature importance indicate that previous goals (Previous_Gls) was the feature with the highest importance.

Using all algorithms, we achieve these results:

Table 6-8: Performance results for Premier League case 2 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	2.19 / 2.08	11.43 / 8.73	3.38 / 2.95	0.48 / 0.65
Ridge Regression	2.19 / 2.09	11.44 / 8.79	3.38 / 2.97	0.48 / 0.65
Random Forest	2.14 / 2.03	10.85 / 8.44	3.29 / 2.90	0.50 / 0.66
Gradient Boosting	2.11 / 1.85	9.82 / 6.29	3.13 / 2.51	0.55 / 0.75
XGBoost	1.93 / 1.75	9.88 / 6.73	3.14 / 2.59	0.55 / 0.73
MLP	2.20 / 2.08	11.57 / 8.78	3.40 / 2.96	0.47 / 0.65

Referring to the provided table, XGBoost emerges as the optimal algorithm when evaluating Mean Absolute Error (MAE). It's noteworthy to highlight that Gradient Boosting

showcases strong performance in terms of metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

Case 3

In this case, we're using all the features in the dataset after going through pre-processing. We can achieve the following results by combining all algorithms.

Table 6-9: Performance results for Premier League case 3 (all players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*R^2 test / R^2 train*
Linear Regression	2.09 / 1.87	11.07 / 7.36	3.33 / 2.71	0.49 / 0.70
Ridge Regression	2.18 / 2.05	11.00 / 8.69	3.32 / 2.95	0.50 / 0.65
Random Forest	2.15 / 1.90	11.32 / 7.24	3.36 / 2.69	0.48 / 0.71
Gradient Boosting	2.17 / 1.67	10.91 / 4.99	3.30 / 2.23	0.50 / 0.80
XGBoost	1.95 / 1.59	10.60 / 5.57	3.26 / 2.36	0.52 / 0.78
MLP	2.14 / 1.98	11.13 / 8.38	3.34 / 2.90	0.49 / 0.66

Examining the presented table, XGBoost stands out as the top-performing algorithm when assessed against the MAE. Notably, it is worth emphasizing that Ridge Regression demonstrates robust performance, particularly excelling in metrics like MSE and RMSE.

6.2.2 Top 30% players dataset

The original dataset, comprising 112 players, was refined to include only 35 players who met the specified goal criteria. Consequently, the results of this implementation indicate that a player must achieve a minimum of 3 goals (threshold) to qualify for inclusion in the final dataset. The dataset's standard deviation, calculated at 6.04 goals.

Case 1

The results of the metrics are depicted in the following table.

Table 6-10: Performance results for Premier League case 1 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	3.78 / 2.88	25.08 / 13.70	5.01 / 3.70	0.54
Ridge Regression	3.75 / 2.93	25.41 / 14.09	5.04 / 3.75	0.53
Random Forest	3.72 / 2.42	25.90 / 9.38	5.33 / 3.06	0.53
Gradient Boosting	3.97 / 2.13	28.43 / 7.22	5.33 / 2.69	0.60
XGBoost	3.67 / 2.29	27.34 / 9.74	5.23 / 3.12	0.49
MLP	3.70 / 2.95	24.86 / 14.20	4.99 / 3.77	0.53

The metrics presented in the table, emphasize the better performance of the XGBoost model, particularly in the MAE. However, MLP model has the lowest values in MSE and RMSE. Regarding feature importance the features that are appearing as the most important are expected Goals (xG), previous goals (Previous_Goals) and non-penalty expected goals (npxG) collectively accounting for a significant 0.60 of the overall importance.

Case 2

The outcomes of the algorithms are presented below.

Table 6-11: Performance results for Premier League case 2 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	3.64 / 2.84	25.55 / 13.59	5.06 / 3.69	0.52
Ridge Regression	3.69 / 2.91	26.38 / 14.12	5.14 / 3.76	0.54
Random Forest	3.41 / 2.84	23.30 / 13.38	4.83 / 3.66	0.47
Gradient Boosting	3.27 / 2.23	20.38 / 7.48	4.51 / 2.73	0.48
XGBoost	3.32 / 2.31	22.00 / 9.55	4.69 / 3.09	0.43
MLP	3.75 / 2.86	26.26 / 13.67	5.12 / 3.70	0.55

The best performed algorithm based on all metrics is Gradient Boosting, with previous goals (Previous_Gls) emerging as the most crucial feature in predicting goal-scoring across the algorithm.

Case 3

Table 6-12: Performance results for Premier League case 3 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	4.00 / 2.59	27.90 / 11.20	5.28 / 3.35	0.62
Ridge Regression	3.65 / 2.98	25.86 / 15.49	5.09 / 3.94	0.53
Random Forest	3.61 / 2.26	25.39 / 8.04	5.04 / 2.83	0.50
Gradient Boosting	3.80 / 0.52	28.05 / 0.43	5.30 / 0.65	0.52
XGBoost	4.07 / 0.01	29.53 / 0.00	5.43 / 0.01	0.59
MLP	3.70 / 2.82	26.02 / 13.41	5.10 / 3.66	0.54

Table 6-13 contains the results of the metrics for all the models. According to all metrics, Random Forest emerges as the top-performing algorithm. Also, upon observing the error metric values for Random Forest and Gradient Boosting, it becomes evident that they exhibit indications of overfitting.

Lastly, the algorithm highlights three features contributing to a sum of nearly 0.7 importance. These are expected goals (xG), previous goals (Previous_Gls) and goals scored per 90 min. (Gls_90).

6.3 La Liga Results

Here, the outcomes and findings from La Liga are presented. The league consists of 97 players, who have only played for La Liga teams from 2017-2018 until 2022-2023.

6.3.1 All players dataset

In this sub-section, we discuss three distinct cases, each utilizing different attributes to train the algorithms. The attributes that are used, are presented in Figure 25 for case 1, in Figure 26 for case 2 and in Figure 27 for case 3. The dataset's standard deviation was computed to 4.35 goals.

Case 1

Upon reviewing the metrics table for both the training and testing datasets, it suggests an overall good performance. The proximity of values across the metrics indicates a lack of overfitting. MLP emerges as the optimal algorithm based on MAE, while Gradient Boosting excels notably in MSE and RMSE.

Table 6-13: Performance results for La Liga case 1 (all players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*R^2 test / R^2 train*
Linear Regression	1.83 / 1.86	7.67 / 8.36	2.77 / 2.89	0.42 / 0.59
Ridge Regression	1.78 / 1.88	7.22 / 8.47	2.69 / 2.91	0.45 / 0.58
Random Forest	1.83 / 1.50	7.31 / 4.70	2.70 / 2.17	0.45 / 0.77
Gradient Boosting	1.86 / 1.57	6.78 / 4.79	2.60 / 2.19	0.49 / 0.76
XGBoost	1.82 / 1.55	7.76 / 5.44	2.79 / 2.33	0.41 / 0.73
MLP	1.72 / 1.88	6.91 / 8.66	2.63 / 2.94	0.48 / 0.57

It is worth noticing that there are no feature importance values for MLP.

Case 2

The results by applying case 2 are presented in the following table.

Table 6-14: Performance results for La Liga case 2 (all players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*R^2 test / R^2 train*
Linear Regression	1.92 / 1.96	7.55 / 8.95	2.75 / 2.99	0.43 / 0.56

Ridge Regression	1.92 / 1.96	7.42 / 8.97	2.72 / 3.00	0.44 / 0.55
Random Forest	1.99 / 1.75	8.49 / 6.55	2.91 / 2.56	0.36 / 0.68
Gradient Boosting	2.09 / 1.68	8.41 / 5.70	2.90 / 2.39	0.36 / 0.72
XGBoost	1.89 / 1.65	7.49 / 6.33	2.74 / 2.52	0.43 / 0.69
MLP	1.95 / 1.96	7.70 / 8.96	2.78 / 2.99	0.42 / 0.56

It appears that XGBoost algorithm outer performs other algorithms based on MAE, while Ridge Regression has good results in MSE and RMSE. Additionally, the close proximity of MAE and RMSE values between training and testing results indicates effective prediction even in unseen data.

Furthermore, following a close examination of the highlighted feature importance values for XGBoost, we see that attributes related to previous goal-scoring performance (Previous_Gls) and penalty kicks (PK) are those with the bigger influence.

Case 3

Based on the results presented in the table, it appears that XGBoost is the best algorithm across all metrics. In this case, it is observed that more features play an important role to the predictions as it is observed from Table 8-3: Feature importance values for case 3.

Table 6-15: Performance results for La Liga case 3 (all players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***R^2 test / R^2 train***
Linear Regression	2.05 / 1.84	8.81 / 7.68	2.97 / 2.77	0.33 / 0.62
Ridge Regression	1.82 / 1.88	7.30 / 8.45	2.70 / 2.91	0.45 / 0.58
Random Forest	1.88 / 1.26	7.90 / 3.08	2.81 / 1.75	0.40 / 0.85
Gradient Boosting	1.94 / 1.50	7.41 / 4.41	2.72 / 2.10	0.44 / 0.78
XGBoost	1.78 / 1.48	6.94 / 4.91	2.63 / 2.21	0.48 / 0.76
MLP	1.95 / 1.89	7.77 / 8.19	2.79 / 2.86	0.41 / 0.59

6.3.2 Top 30% players dataset

The initial dataset, comprising 97 players, was reduced to 32 players. The outcomes of this process indicated that for a player to be included in the final dataset it had to score a minimum of two goals (threshold). The dataset's variability is indicated by the calculated standard deviation of 5.55. Another observation is that the values between the MAE and RMSE for the training and testing sets do not surpass 1.00, suggesting a lack of significant overfitting in the data.

Case 1

The results for this case were as follows.

Table 6-16: Performance results for La Liga case 1 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	2.96 / 3.18	15.97 / 17.51	4.00 / 4.18	0.68
Ridge Regression	2.53 / 3.20	12.34 / 18.15	3.51 / 4.26	0.59
Random Forest	2.37 / 2.78	10.99 / 12.92	3.31 / 3.59	0.50
Gradient Boosting	2.49 / 2.09	10.53 / 7.39	3.25 / 2.72	0.55
XGBoost	3.05 / 2.17	15.73 / 8.46	3.97 / 2.91	0.55
MLP	2.61 / 3.25	12.73 / 18.85	3.57 / 4.34	0.60

Random Forest is the best model regarding the outcomes of MAE and MAPE. As explained before, a MAPE of 0.50 depicts that the model's predictions differ from the actual values by around 50% on average, which is not ideal.

Concerning feature importance, the 4 features with greater than 0.1 significance are: non-penalty expected plus assisted goals (npxG_PLUS_xAG), goals plus assists (G_PLUS_A), expected goals per 90 min. (xG_90) and expected plus assisted goals per 90 min. (xG_PLUS_xAG_90).

Case 2

Table 6-17: Performance results for La Liga case 2 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	3.10 / 3.14	15.43 / 17.44	3.93 / 4.18	0.73
Ridge Regression	2.83 / 3.15	13.89 / 17.61	3.73 / 4.20	0.65
Random Forest	2.39 / 3.05	10.67 / 15.10	3.27 / 3.89	0.53
Gradient Boosting	2.68 / 2.38	13.69 / 8.53	3.70 / 2.92	0.30
XGBoost	2.92 / 2.15	16.08 / 10.00	4.01 / 3.16	0.54
MLP	3.05 / 3.14	15.07 / 17.45	3.88 / 4.18	0.71

In this scenario, the Random Forest algorithm stands out as the superior performer, attaining the lowest value in MAE and thus establishing its excellence. Significantly, the combined influence of previous goals (Previous_Gls) and progressive passes records (PrgR) surpasses the limit of 0.5 influence.

Case 3

Table 6-18: Performance results for La Liga case 3 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	3.85 / 2.99	24.47 / 14.39	4.95 / 3.79	0.97
Ridge Regression	2.42 / 3.20	11.17 / 17.74	3.34 / 4.21	0.53
Random Forest	2.60 / 1.39	12.38 / 3.42	3.52 / 1.85	0.58
Gradient Boosting	2.73 / 1.94	12.97 / 5.92	3.60 / 2.43	0.60
XGBoost	2.75 / 2.09	13.28 / 8.01	3.64 / 2.83	0.55
MLP	3.46 / 3.01	16.97 / 15.91	4.12 / 3.99	0.76

In this case, the Ridge Regression algorithm outperforms all others across multiple criteria, demonstrating its better performance.

6.4 Serie A Results

In this chapter, the results of Serie A are analyzed. The information was reduced to focus solely on players in Serie A teams from the 2017-2018 season to the 2022-2023 season. As a result, the total number of players considered reaches 106.

6.4.1 All players dataset

In this section, we look at three distinct scenarios, each of which uses a different set of features to train the algorithms. Moreover, the standard deviation of the dataset was calculated to be 4.35 goals, offering insight into the diversity of goal-scoring performances evaluated.

Case 1

For Case 1, the model is trained using the subsequent attributes: expected goals, non-penalty expected goals, non-penalty expected plus assisted goals, expected goals per 90 min., previous goals, non-penalty expected goals per 90 min., non-penalty goals, expected plus assisted goals per 90 min., goals plus assists, progressive passes records. Out of these features the most important to the predictions is expected goals (xG). By observing the following table with the results, it is understood that XGBoost is the best performed algorithm. The error values are relatively small indicating an overall good performance of our models. Moreover, an examination of MAE and RMSE values between training and testing results demonstrate effective predictions in unknown scenarios.

Table 6-19: Performance results for Serie A case 1 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	1.44 / 1.67	5.47 / 7.43	2.34 / 2.73	0.28 / 0.62
Ridge Regression	1.43 / 1.66	5.32 / 7.45	2.31 / 2.73	0.30 / 0.62
Random Forest	1.40 / 1.35	4.91 / 3.91	2.22 / 1.98	0.35 / 0.80
Gradient Boosting	1.49 / 1.55	4.45 / 5.02	2.11 / 2.24	0.41 / 0.74
XGBoost	1.29 / 1.39	3.96 / 4.68	1.99 / 2.16	0.48 / 0.76
MLP	1.44 / 1.67	5.41 / 7.44	2.33 / 2.73	0.29 / 0.62

Case 2

Case 2 features are: nation, squad, position, age, minutes played, assists, penalty kicks made, yellow and red cards, non-penalty goals, expected plus assisted goal per 90 min., goals and assists, progressive passes records. Table 6-23 indicates that the algorithm with the lowest values in all metrics is XGBoost. The nearby values of the metrics show that there is no important sign of overfitting.

Table 6-20: Performance results for Serie A case 2 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	1.60 / 1.89	5.01 / 8.77	2.24 / 2.96	0.34 / 0.55
Ridge Regression	1.60 / 1.89	5.01 / 8.77	2.24 / 2.96	0.34 / 0.55
Random Forest	1.63 / 1.65	5.16 / 6.08	2.27 / 2.47	0.32 / 0.69
Gradient Boosting	1.62 / 1.69	4.17 / 5.96	2.04 / 2.44	0.45 / 0.69
XGBoost	1.40 / 1.49	3.66 / 5.59	1.91 / 2.36	0.52 / 0.71
MLP	1.63 / 1.89	5.09 / 8.80	2.26 / 2.97	0.33 / 0.55

Several fundamental insights concerning the significance of features in this case include:

- All features have a positive impact.
- The attribute with the biggest value is previous goals (Previous_Gls), while penalty kicks made (PK) is the second.

Case 3

Case 3 uses all the attributes referred to Figure 27: Case 3 attributes.

Table 6-21: Performance results for Serie A case 3 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	1.59 / 1.62	7.95 / 6.80	2.82 / 2.61	0.05 / 0.65

Ridge Regression	1.41 / 1.65	5.48 / 7.18	2.34 / 2.68	0.28 / 0.63
Random Forest	1.45 / 1.47	5.05 / 4.67	2.25 / 2.16	0.33 / 0.76
Gradient Boosting	1.68 / 1.74	4.66 / 6.41	2.16 / 2.53	0.39 / 0.67
XGBoost	1.33 / 1.32	4.12 / 4.23	2.03 / 2.06	0.46 / 0.78
MLP	1.49 / 1.79	5.14 / 8.03	2.27 / 2.83	0.32 / 0.59

By analyzing the above results, XGBoost algorithm continues to be the best of all. Notably, there is not a sign of overfitting within the dataset. Expected goals (xG) is the feature that was mostly used to make the predictions as accurate as possible.

6.4.2 Top 30% players dataset

From the initial dataset of 106 players, only 38 were selected based on meeting a minimum goal criterion. This process revealed that players needed to score at least 2 goals to be included in the final dataset. The dataset's standard deviation, calculated at 5.08 goals, measures its variability.

Case 1

In reference to the presented table, MLP stands out as the best algorithm, in all metrics. Moreover, train and test metric values are closely enough meaning that the algorithms perform good in unseen data.

Table 6-22: Performance results for Serie A case 1 (30% top players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***MAPE test***
Linear Regression	2.01 / 2.71	9.28 / 14.95	3.05 / 3.87	0.59
Ridge Regression	1.93 / 2.79	8.36 / 15.75	2.89 / 3.97	0.52
Random Forest	1.87 / 2.44	7.55 / 11.35	2.75 / 3.37	0.50
Gradient Boosting	1.86 / 2.47	6.69 / 10.87	2.59 / 3.30	0.55
XGBoost	1.85 / 2.15	7.43 / 11.15	2.73 / 3.34	0.46
MLP	1.69 / 2.90	5.98 / 17.25	2.45 / 4.15	0.45

Case 2

The outcomes reveal that the Gradient Boosting algorithm consistently exhibits the lowest values across all metrics. However, in this case it is implied that overfitting exists in XGBoost algorithm, due to the high difference between train and test values.

Table 6-23: Performance results for Serie A case 2 (30% top players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***MAPE test***

Linear Regression	2.09 / 2.90	8.32 / 16.44	2.88 / 4.06	0.58
Ridge Regression	2.09 / 2.90	8.26 / 16.44	2.87 / 4.06	0.58
Random Forest	1.87 / 2.52	6.29 / 11.72	2.51 / 3.42	0.49
Gradient Boosting	1.66 / 2.00	4.88 / 6.57	2.21 / 2.56	0.47
XGBoost	2.44 / 0.25	11.49 / 0.10	3.39 / 0.32	0.64
MLP	2.31 / 2.99	8.03 / 20.40	2.83 / 4.52	0.63

Regarding the feature importance of all algorithms, the conclusion is that feature Previous Goals (Previous_Goals) remains the most influential factor.

Case 3

XGBoost emerges as the optimal algorithm, showing superior performance based on both MAE and MAPE metrics. Concerning the significance of features across the outcomes, it can be concluded that expected Goals (xG) and non-penalty expected goals (npxG) are the most highlighted in the model.

Table 6-24: Performance results for Serie A case 3 (30% top players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***MAPE test***
Linear Regression	2.78 / 2.54	15.97 / 13.64	4.00 / 3.69	0.82
Ridge Regression	2.10 / 2.71	9.51 / 14.94	3.08 / 3.86	0.59
Random Forest	1.96 / 2.38	8.71 / 10.41	2.95 / 3.23	0.53
Gradient Boosting	1.93 / 2.33	7.83 / 9.22	2.80 / 3.04	0.60
XGBoost	1.83 / 2.02	7.65 / 9.97	2.77 / 3.16	0.47
MLP	2.00 / 2.75	8.51 / 15.14	2.92 / 3.89	0.55

6.5 All Players Dataset Results

As mentioned earlier, this case includes all players from the four leagues. An additional column, 'league,' has been introduced to denote the origin of each player (the league that he plays). The subsequent sub-chapters delve into a detailed analysis of the results.

6.5.1 All players dataset

The dataset contains 424 players in total. The standard deviation of this dataset was calculated to 4.23 goals. As previously emphasized, the attributes used in the algorithms for each case are depicted in the Pre-processing chapter.

Case 1

Given the diversity of this dataset, the results depict that the models produced very good results. The best algorithm is XGBoost with lowest values in all metrics. Features like

expected goals (xG) and non-penalty expected goals (npxG) are the most influential. Finally. the league feature, while positive, it did not get a high value.

Table 6-25: Performance results for All players dataset case 1 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	1.76 / 1.78	7.30 / 7.55	2.70 / 2.75	0.46 / 0.60
Ridge Regression	1.74 / 1.78	7.02 / 7.57	2.65 / 2.75	0.49 / 0.60
Random Forest	1.76 / 1.63	7.28 / 5.95	2.70 / 2.44	0.47 / 0.68
Gradient Boosting	1.84 / 1.73	7.06 / 6.43	2.66 / 2.54	0.48 / 0.66
XGBoost	1.69 / 1.66	6.68 / 6.63	2.58 / 2.57	0.51 / 0.65
MLP	1.76 / 1.79	7.06 / 7.58	2.66 / 2.77	0.48 / 0.60

Case 2

The following table shows that the best performed algorithm is Random Forest. Furthermore, the values between train and test indicate that the models predict well in unknown data. The feature with the highest feature importance reaching the value of 0.76 is previous goals (Previous_Goals), while league does not have an important influence.

Table 6-26: Performance results for All players dataset case 2 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	1.86 / 1.92	7.15 / 8.26	2.67 / 2.87	0.48 / 0.56
Ridge Regression	1.84 / 1.92	7.05 / 8.29	2.66 / 2.88	0.48 / 0.56
Random Forest	1.78 / 1.58	6.76 / 5.52	2.60 / 2.35	0.50 / 0.71
Gradient Boosting	1.78 / 1.69	6.80 / 5.81	2.61 / 2.41	0.50 / 0.69
XGBoost	1.80 / 1.51	7.20 / 4.97	2.68 / 2.23	0.47 / 0.74
MLP	1.91 / 1.93	7.35 / 8.31	2.71 / 2.88	0.46 / 0.56

Case 3

Once again, the XGBoost algorithm outperforms all other algorithms, attaining the lowest values in this case. One observation about feature importance is the following. Expected goals (xG) appears as the most important feature.

Table 6-27: Performance results for All players dataset case 3 (all players)

Models /Metrics	MAE test / MAE train	MSE test / MSE train	RMSE test / RMSE train	R^2 test / R^2 train
Linear Regression	1.76 / 1.77	7.44 / 7.33	2.73 / 2.71	0.45 / 0.61
Ridge Regression	1.81 / 1.80	7.02 / 7.56	2.65 / 2.75	0.49 / 0.60
Random Forest	1.77 / 1.50	7.13 / 4.70	2.67 / 2.17	0.48 / 0.75

Gradient Boosting	1.85 / 1.58	7.19 / 4.97	2.68 / 2.23	0.47 / 0.74
XGBoost	1.67 / 1.51	6.48 / 5.19	2.55 / 2.28	0.52 / 0.72
MLP	1.90 / 1.86	7.35 / 7.86	2.71 / 2.80	0.46 / 0.58

6.5.2 Top 30% players dataset

The initial dataset of 424 players, had a reduction to 157 players. The results of this process showed that a player had to achieve a minimum of 2 goals (threshold) to be included in the final dataset. The calculated standard deviation of the dataset is 5.18.

Case 1

In this case, Gradient Boosting is the algorithm that produced the lowest MAE value. However, Random Forest performs extremely well based on MSE and RMSE metrics. The feature displaying the highest significance is expected goals (xG). Remarkably, the values between the training and testing datasets exhibit close proximity, indicating the absence of overfitting in the data.

Table 6-28: Performance results for All players dataset case 1 (30% top players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***MAPE test***
Linear Regression	2.48 / 2.69	12.78 / 13.63	3.57 / 3.69	0.53
Ridge Regression	2.44 / 2.69	12.37 / 13.66	3.52 / 3.70	0.52
Random Forest	2.32 / 2.65	11.41 / 12.85	3.38 / 3.59	0.47
Gradient Boosting	2.30 / 2.44	11.90 / 10.76	3.45 / 3.28	0.47
XGBoost	2.42 / 2.40	12.13 / 11.43	3.48 / 3.38	0.47
MLP	2.49 / 2.69	12.66 / 13.65	3.56 / 3.69	0.53

Case 2

The best performed algorithm is Gradient Boosting like previously. In this case, previous goals (Previous_Goals) is the attribute that has the highest value.

Table 6-29: Performance results for All players dataset case 2 (30% top players)

Models /Metrics	***MAE test / MAE train***	***MSE test / MSE train***	***RMSE test / RMSE train***	***MAPE test***
Linear Regression	2.51 / 2.77	12.54 / 14.21	3.54 / 3.77	0.54
Ridge Regression	2.46/ 2.78	12.43 / 14.30	3.53 / 3.78	0.53
Random Forest	2.40 / 2.37	11.33 / 10.00	3.37 / 3.16	0.51
Gradient Boosting	2.31 / 2.47	11.04 / 10.83	3.32 / 3.29	0.48
XGBoost	2.43 / 2.17	11.47 / 9.41	3.39 / 3.07	0.48
MLP	2.52 / 2.77	12.58 / 14.21	3.55 / 3.77	0.54

Case 3

Finally, for another time Gradient Boosting algorithm is the winner regarding the different error metrics.

Table 6-30: Performance results for All players dataset case 3 (30% top players)

Models /Metrics	*MAE test / MAE train*	*MSE test / MSE train*	*RMSE test / RMSE train*	*MAPE test*
Linear Regression	2.64 / 2.65	13.79 / 13.06	3.71 / 3.61	0.57
Ridge Regression	2.57 / 2.72	12.85 / 13.52	3.58 / 3.68	0.55
Random Forest	2.42 / 2.02	12.08 / 6.91	3.48 / 2.63	0.53
Gradient Boosting	2.28 / 2.32	11.17 / 9.42	3.34 / 3.07	0.48
XGBoost	2.37 / 2.29	11.64 / 10.20	3.41 / 3.20	0.48
MLP	2.542/ 2.73	12.33 / 13.69	3.51 / 3.70	0.54

Discussion

In this chapter, all the results of each different case and league are analyzed. The final sub-chapter provides a comprehensive comparison of these results.

7.1 Overview of Cross-League Analysis

Here, summaries of individual leagues and an overview of the dataset which includes players from all leagues are provided.

7.1.1 Bundesliga Implications

For the comprehensive Bundesliga's dataset encompassing all players, the findings indicated that case 2 and XGBoost algorithm outperformed others, as evidenced by its lower Mean Absolute Error (MAE). In contrast, when focusing on the top 30% of players, case 2 emerged as the optimal choice, with Random Forest standing out as the most effective algorithm in delivering good results. The attributes in case 2 were chosen by evaluating the correlation for each pair of features. Specifically, one feature was selected from each highly correlated pair.

The feature that has the highest importance value for both algorithms is the previous goals (Previous_Gls), suggesting that a player's previous goal-scoring performance is a critical factor in predicting future outcomes. Another important feature for Random Forest is Age. This might suggest that the performance and impact of a player vary with age, and older or younger players might have distinctive characteristics that affect their overall contribution to the team.

Table 7-1: Performance results for Bundesliga - comparative analysis

Models /Metrics	*MAE*	*MSE*	*RMSE*	*MAPE*	*R^2*
XGBoost (all dataset – case 2)	1.70	5.41	2.33	-	0.50
Random Forest (30% top players – case 2)	1.70	4.38	2.09	0.29	0.57

While the anticipation of variations in results across different dataset versions was foreseen, the noteworthy observation lies in the exceptional performance of the reduced da-

taset in comparison. Contrary to expectations, an algorithm trained on a significantly reduced dataset demonstrated superior results.

To further recognize their differences, we will offer an example of two football players and their predictions. Generally, if we use center backs or defenders as examples, the predictions would be more accurate than forwards, because a defender normally scores zero goals in a season. As a result, to check the results of the algorithms, Thomas Müller and Joshua Kimmich were chosen. Thomas Müller is a forward and his actual goals for season 2022-2023 were 7. On the other hand, Joshua Kimmich is a mid-fielder and he scored 5 goals in the season 2022-2023.

The following bar-charts depict the results.

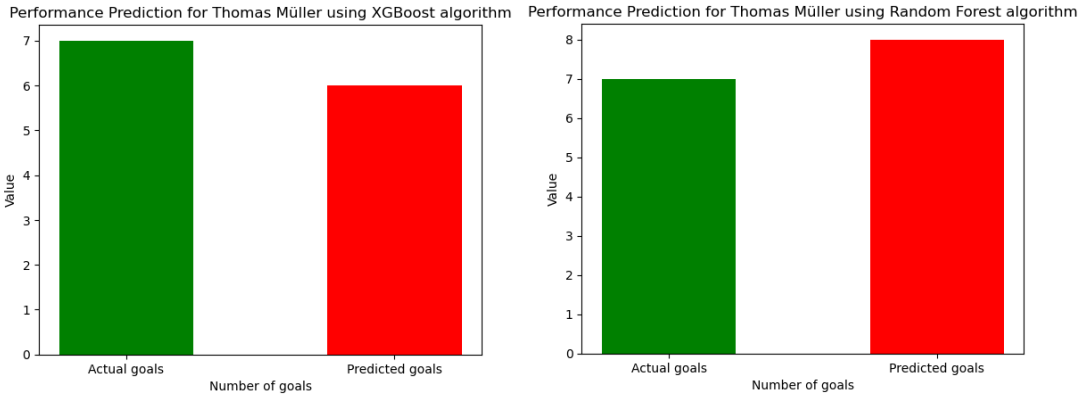

Figure 33: Thomas Müller's performance prediction

In the case of Thomas Müller, the XGBoost algorithm, applied to the entire dataset, forecasted 6 goals, with a discrepancy of just 1 goal. Meanwhile, the Random Forest algorithm, when employed on the top 30% of players, predicted 8 goals, exhibiting a similar 1-goal difference.

For Joshua Kimmich, the XGBoost algorithm, utilizing the complete dataset, predicted 2 goals with a deviation of 3 goals. In parallel, the Random Forest algorithm, applied to the top 30% of players, predicted 3 goals, displaying a 2-goals difference.

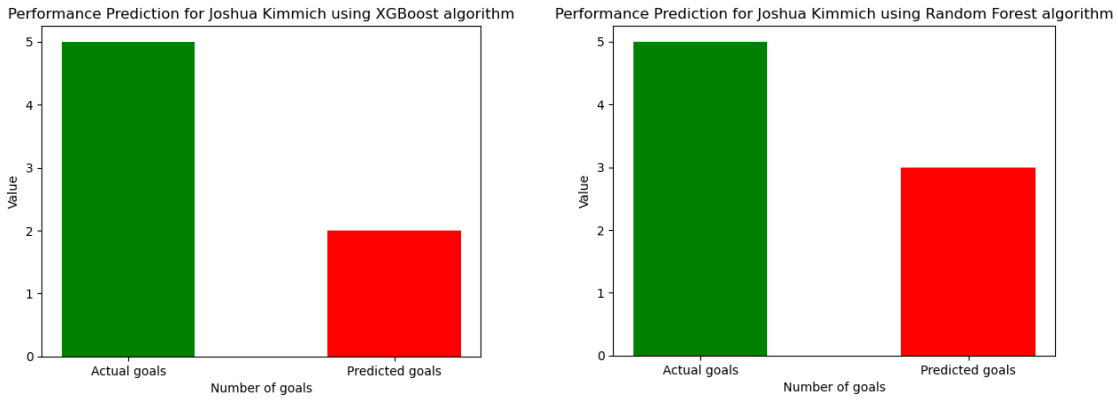

Figure 34: Joshua Kimmich's performance prediction

Obviously, some players' actual goals align precisely with the predictions made by all models, while conversely, there are numerous instances where the predicted goals for players deviate significantly from their actual achievements.

One more conclusion can come out of the calculation of standard deviation of the target variable 'Gls'. It was calculated at 3.24 goals, while MAE for the best model is 1.795 regarding the dataset with all the players. This means that, if a player scored 10 goals, the prediction would be 10 +/-3, while the error is at 1.795, which is a good outcome. Regarding the dataset with the top 30% players the standard deviation was calculated to 3.79.

7.1.2 Premier League Implications

In the analysis of the dataset encompassing all players of Premier League, the results revealed that case 1 and the XGBoost algorithm demonstrated superior performance, evident in their lower MAE compared to other approaches. It is worth mentioning also, that all cases regarding this dataset had XGBoost as their best training model. Conversely, when narrowing the focus to the top 30% of players, case 2 emerged as the preferred option, with Gradient Boosting emerging as the most effective algorithm.

In this league, different features were used for each implementation. In the domain of XGBoost, expected goals (xG), non-penalty expected goals (npxG) and previous goals (Previous_Goals) consistently emerge as the most important features. Their consistent importance emphasizes their dependability in impacting the target variable. However, for Gradient Boosting, previous goals (Previous_Goals) is the only feature that stands out.

Table 7-2: Performance results for Premier League - comparative analysis

Models /Metrics	*MAE*	*MSE*	*RMSE*	*MAPE*	*R^2*
XGBoost (all dataset – case 1)	1.92	10.35	3.22	-	0.53
Gradient Boosting (30% top players – case 2)	3.27	20.38	4.51	0.48	0.37

In this case, it is understood based on the metrics that the dataset which contained all the players produced better results. Specifically, the values of all measures associated to the reduced dataset are noticeably higher, demonstrating the whole player dataset's superior performance.

Furthermore, it is vital to mention that generally the values between the train and test metrics contained the whole dataset have a small difference between them. This suggests that the occurrence of overfitting is relatively minor. However, for the reduced dataset, there is a notable increase in the difference between the metric values.

To illuminate these distinctions further, let's consider Danny Welbeck and his respective predictions. Danny Welbeck is a midfielder, who scored 6 goals in the actual season of 2022-2023.

The following bar charts vividly illustrate the outcomes, offering a comparative view of the predictions across different algorithms.

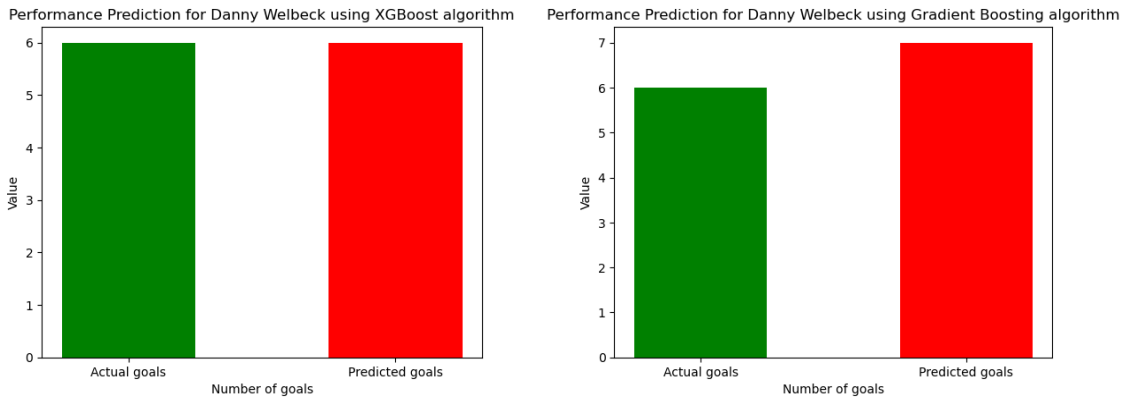

Figure 35: Danny Welbeck's performance prediction

In the case of Danny Welbeck, the XGBoost algorithm accurately forecasted the precise number of goals the player achieved during the 2022-2023 season. In contrast, the Gradient Boosting algorithm, applied to the top 30% of players, predicted a total of 7 goals, showcasing a minor 1-goal deviation.

7.1.3 La Liga Implications

The best outcome for this league was produced using MLP in case 1 for the entire dataset of La Liga. In contrast, for the top 30% of players in case 1, Random Forest emerged as the most effective algorithm. As the results indicate in the table below, the dataset with all the players produced better results.

Table 7-3: Performance results for La Liga - comparative analysis

Models /Metrics	*MAE*	*MSE*	*RMSE*	*MAPE*	*R^2*
MLP (all dataset – case 1)	1.72	6.91	2.63	-	0.48
Random Forest (30% top players – case 1)	2.37	10.99	3.31	0.50	0.39

The features that are most important for Random Forest algorithm is non-penalty expected plus assisted goals (npxG_PLUS_xAG).

To delve deeper into these differences, we will examine 2 cases: Luka Modrić and Karim Benzema. Luka Modrić is a mid-fielder and he scored 4 goals in season 2022-2023. Karim Benzema is a striker who achieved a total of 19 goals during this season.

In the scenario involving Luka Modrić, the MLP algorithm, applied to the complete dataset, predicted 2 goal, 2 goals difference from the actual ones. Conversely, the Random Forest algorithm, when deployed on the top 30% of players, predicted 3 goals, meaning 1-goal difference.

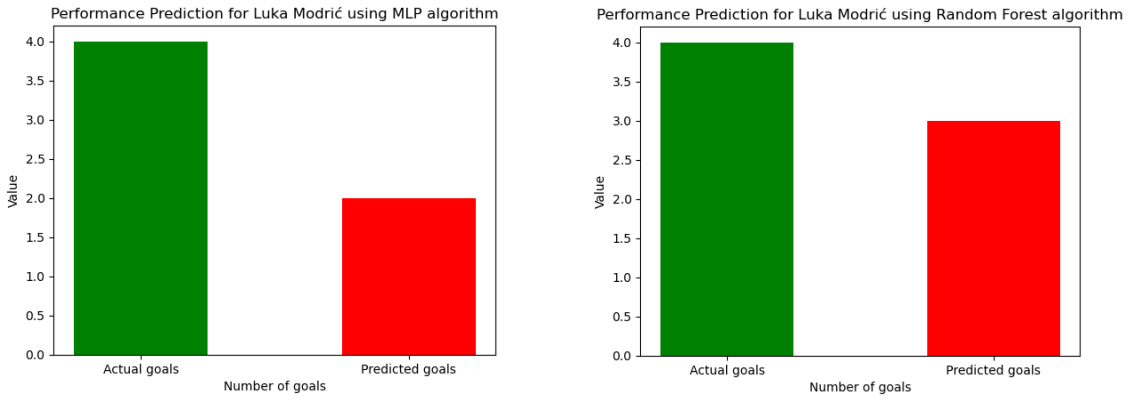

Figure 36: Luka Modrić's performance prediction

In the case of Karim Benzema, the MLP algorithm forecasted 19 goals, while the Random Forest predicted 21 goals. Notably, Random Forest prediction exhibited a 2-goal difference from the observed outcome.

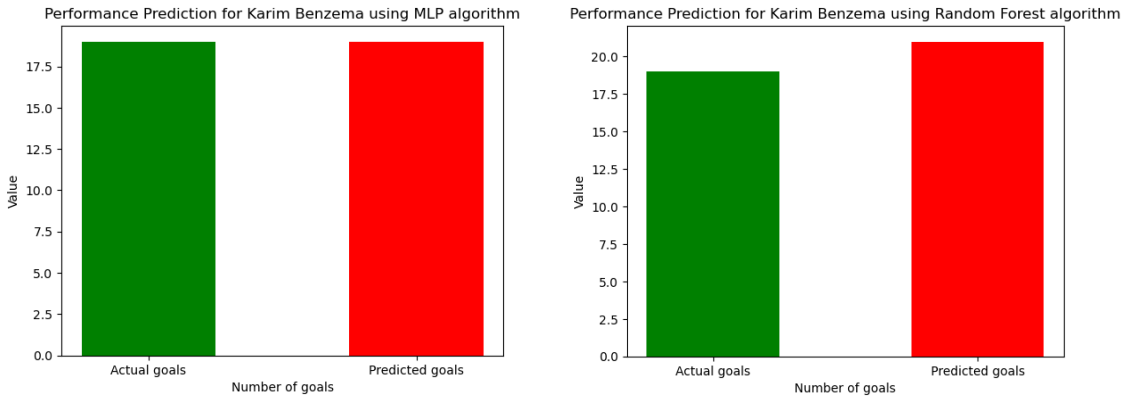

Figure 37: Karim Benzema's performance prediction

7.1.4 Serie A Implications

Table 6-28 illustrates that, for the entire dataset of Serie A, the XGBoost algorithm demonstrated the best performance, whereas in the context of the reduced dataset, Gradient Boosting claimed the top position. Overall, the best metric values produced the case with the entire dataset. An additional noteworthy observation is that the values between the training and testing sets in this league exhibit a closer proximity compared to any other league.

Table 7-4: Performance results for Serie A - comparative analysis

Models /Metrics	***MAE***	***MSE***	***RMSE***	***MAPE***	***R^2***
XGBoost (all dataset – case 1)	1.29	3.96	1.99	-	0.48
Gradient Boosting (30% top players – case 2)	1.67	4.89	2.21	0.47	0.47

An example to depict these distinctions is given by Nicolò Barella. Nicolò Barella is a mid-fielder who scored 6 goals in target season 2022-2023.

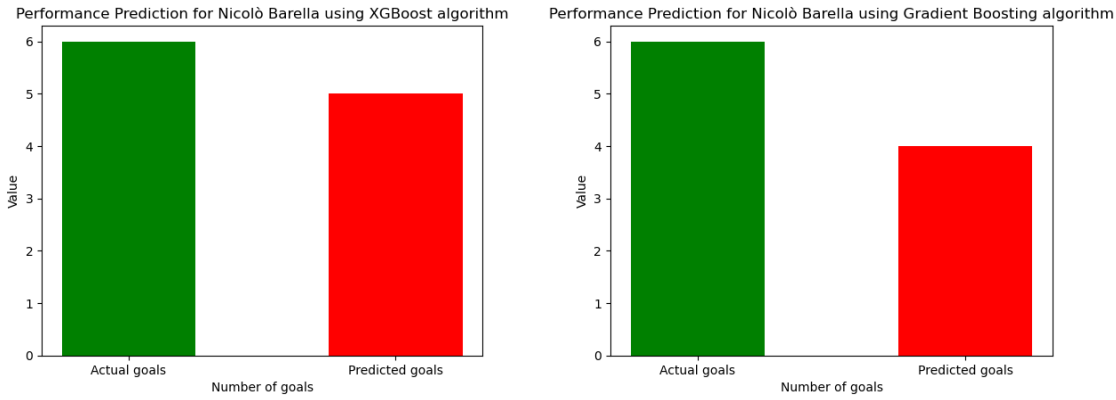

Figure 38: Nicolò Barella's performance prediction

The bar charts showed that XGBoost algorithm predicted with 1-goal difference from the actual number of goals, whereas Gradient Boosting exhibited a slightly larger 2-goal difference.

7.1.5 All dataset Implications

Based on the previous analysis, we conclude to the following epilogue. XGBoost was the best performed algorithm for the dataset that contained all the players. On the other hand, for the reduced dataset, Gradient Boosting produced the best metric values. Significantly, the 2 models that produced these results used all the features (case 3) for the training, with expected goals (xG) playing a key role in both instances. Like expected, the dataset contained all players produced better results.

Table 7-5: Performance results for All players dataset - comparative analysis

Models /Metrics	*MAE*	*MSE*	*RMSE*	*MAPE*	*R^2*
XGBoost (all dataset – case 3)	1.68	6.58	2.56	-	0.52
Gradient Boosting (30% top players – case 3)	2.29	11.23	3.35	0.48	0.39

Visual representations of the XGBoost algorithm's performance using the complete dataset in case 3 in comparison with the best model from each league are depicted through the following diagrams.

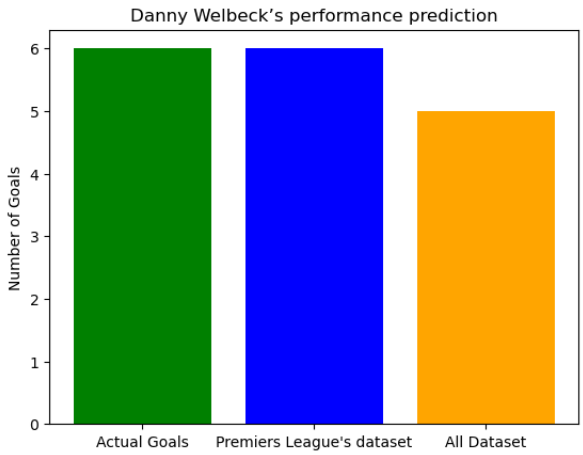

Figure 39: Performance prediction of Danny Welbeck using XGBoost algorithm

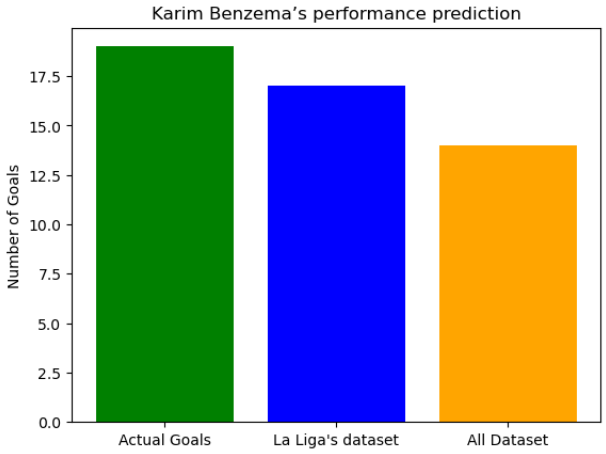

Figure 40: Performance prediction of Karim Benzema using XGBoost algorithm

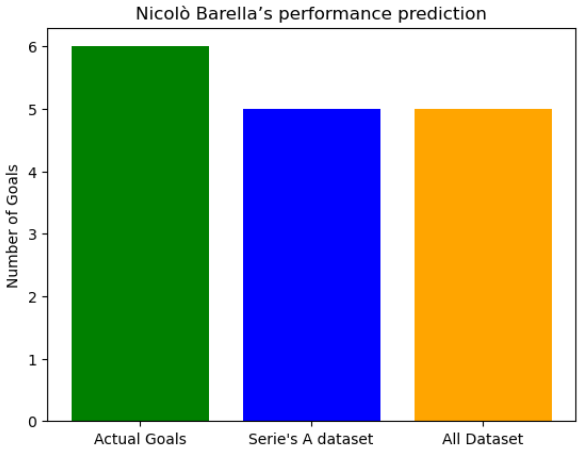

Figure 41: Performance prediction of Nicolò Barella using XGBoost algorithm

It is observed that for these specific players, the models developed in each league's dataset produced better predictions.

7.2 Comparative Insights Across Leagues

As previously stated, the comparison between the algorithms in all the possible cases is one of the most important factors of this dissertation. The results are gathered in the table below:

Table 7-6: Performance results all scenarios - comparative analysis

League	*Best Model /Metrics*	*MAE*	*MSE*	*RMSE*	*MAPE*	*R^2*
Bundesliga	***Random Forest (30% top players – case 2)***	1.70	4.38	2.09	0.29	0.57
Premier League	***XGBoost (all dataset – case 1)***	1.92	10.35	3.22	-	0.53
La Liga	***MLP (all dataset – case 1)***	1.72	6.91	2.63	-	0.48
Serie A	***XGBoost (all dataset – case 1)***	1.29	3.96	1.99	-	0.48

All dataset	***XGBoost (all dataset – case 3)***	1.68	6.58	2.56	-	0.52

The outcomes yielded intriguing insights. By observing Table 7-6: Performance results all scenarios - comparative analysis we have some conclusions. To begin with, Serie's A dataset produced the lowest error metric values than any other league or the total dataset. The most essential indicator between the metrics is MAE. MAE shows how close the predictions are to the actual number of goals. In this case, XGBoost the MAE was 1.29, meaning that if for example a player scored 10 goals, the prediction would be to score 9 to 11 goals. Certainly, not all predictions achieve this result. There are many instances where the predictions are not accurate, but overall, they are very good.

Secondly, only in Bundesliga, an algorithm using the reduced dataset produced better results. Delving into specific metrics, the mean absolute error (MAE) witnessed the second lowest value when the XGBoost algorithm was applied to the entire dataset across all leagues. This indicates that besides the different cultures and gameplays each league has, the whole dataset produced more accurate predictions. However, the other error metrics are not the lowest in that case.

Lastly and most important, it becomes apparent that XGBoost algorithm emerges as the overall winner in 3 out of 5 scenarios. As a result, this algorithm should be used by data scientists when these or similar datasets are used.

Furthermore, all researchers must realize the significance of their study. Our results are promising. They outperform most previous studies, despite the fact that some researches have slightly superior error metrics. It's important to realize that comparing results might be challenging because researchers usually use different datasets. For example, predicting points in basketball differs significantly from football due to the many fluctuations it has as a sport.

8 Conclusions

In this dissertation, we successfully predicted a player's performance regarding the number of goals based on historical data. This chapter is dedicated to summarize the methodology and the results of the ML algorithms.

8.1 Conclusion

It is understood that sport analytics are going to play a vital role in shaping the future of sports. Across the world, an increasing number of teams are hiring experts in this domain to enhance the performance of the players. Recognizing the potential for improvement, football clubs are giving the appropriate attention to improve each players performance and make them perform better.

In the scope of this dissertation, our primary aim was to forecast the performance of football players, specifically predicting the number of goals they would score, based on their historical statistics. To achieve this, we systematically evaluated the effectiveness of six distinct machine learning algorithms: Linear Regression, Ridge Regression, Random Forest, Gradient Boosting, XGBoost, and Multilayer Perceptron. Each implementation had 2 different versions. One containing the whole dataset and one with the 30% top players dataset. Then, the implementations were divided based on the features used for training.

Our dataset comprised over 5,000 professional football players originally, and the algorithms were trained on statistics from the 2018-19 to 2021-2022 seasons. The next season, 2022-23, served as the testing ground.

The results were quite intriguing. Out of all algorithms, XGBoost was the one that in the most of the cases produced better results and was the most accurate. Generally, the results were very good avoiding overfitting. The metrics that were used to measure the effectiveness of each model in general were: MAE, MSE, RMSE, MAPE and R-squared.

In the larger context, the relevance of sports analytics is escalating rapidly. The increasing flow of large amounts of data every day requires effective tools like Machine Learning and Data Mining for collecting and analyzing it efficiently. Looking ahead, we can expect sports clubs to form dedicated teams of data scientists for every sport.

In conclusion, this dissertation offers valuable insights for football clubs, managers, and coaching staff. The findings can empower decision-making processes by providing invaluable information on how well a player is anticipated to perform in the upcoming season, thereby contributing to collective squad improvement.

8.2 Future Work

Our dissertation findings suggest that accurate predictions of a player's performance for the upcoming season based on historical data is possible. Nonetheless, there is always room for further innovation for results to be more accurate and effective.

There are many ways that a data scientist can improve his work to achieve better results. First of all, an idea is to use more advanced statistics or even take the statistics of players for each match of the season to have more data to train the models. Moreover, a data scientist can use the results of this dissertation to calculate the total number of goals a team will achieve by summing all the players goal performance. If the total goals surpass the previous season, it might indicate a potential for the team to achieve a higher-ranking place.

Another idea, as it is mentioned in a separate chapter, is the use of wearable devices or cameras [18]. These devices provide a variety of statistics about a player's field movements and physical condition, such as heart rate and breathing patterns. Although difficult to gather, such data has significant potential for improving goal performance analysis.

Similar idea, referring to the phycological factors that can influence a player's performance is to analyze twitter data using sentiment analysis [78]. This information aids coaches and teams in making crucial decisions to boost morale or determine optimal player usage in critical matches [79].

Lastly, injury analytics is another big part of sports. Injury analytics are critical in optimizing a player performance and reducing injury risk. Teams may improve player well-being and sustain peak physical condition throughout the season by leveraging data on player fitness and movement patterns [80].

References

[1] Sports Reference. (2023). https://www.sports-reference.com/

[2] Nutan Farah Haq, Abdur Rahman Onik, Md. Avishek Khan Hridoy, Musharrat Rafni, Faisal Muhammad Shah and Dewan Md. Farid, “Application of Machine Learning Approaches in Intrusion Detection System: A Survey” International Journal of Advanced Research in Artificial Intelligence(ijarai), 4(3), 2015.

http://dx.doi.org/10.14569/IJARAI.2015.040302

[3] B. N. Lakshmi and G. H. Raghunandhan, "A conceptual overview of data mining," 2011 National Conference on Innovations in Emerging Technology, Erode, India, 2011, pp. 27-32, doi: 10.1109/NCOIET.2011.5738828.

[4] Analytics Vidhya, *All you need to know about sport analytics in 2023*, https://www.analyticsvidhya.com/blog/2023/08/sport-analytics/#h-how-do-fans-use-sports-analytics-data

[5] S. Shahriar, A. R. Al-Ali, A. H. Osman, S. Dhou and M. Nijim, "Machine Learning Approaches for EV Charging Behavior: A Review," in IEEE Access, vol. 8, pp. 168980-168993, 2020, doi: 10.1109/ACCESS.2020.3023388

[6] Daniel, J., 2021. *Machine Learning Tutorial for Beginners: What is, Basics of ML.*

https://www.guru99.com/machine-learning-tutorial.html

[7] Chatzilygeroudis, Konstantinos et al. “Machine Learning Basics.” Intelligent Computing for Interactive System Design, 2021

https://doi.org/10.1145/3447404.3447414

[8] H. U. Dike, Y. Zhou, K. K. Deveerasetty and Q. Wu, "Unsupervised Learning Based On Artificial Neural Network: A Review," 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS), Shenzhen, China, 2018, pp. 322-327, doi: 10.1109/CBS.2018.8612259.

[9] IBM. *What is unsupervised learning?* https://www.ibm.com/topics/unsupervised-learning

[10] Chapter 11 - Singular Value Decomposition in Text Mining,

Editor(s): Gary Miner, Dursun Delen, John Elder, Andrew Fast, Thomas Hill, Robert A. Nisbet, Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications, Academic Press, 2012, Pages 935-947, ISBN 9780123869791,

https://doi.org/10.1016/B978-0-12-386979-1.00039-6

[11] J. Jia and W. Wang, "Review of reinforcement learning research," 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Zhanjiang, China, 2020, pp. 186-191, doi: 10.1109/YAC51587.2020.9337653.

[12] GeeksforGeeks, *Reinforcement Learning*, https://www.geeksforgeeks.org/what-is-reinforcement-learning/

[13] Shweta Bhatt 2018. *Reinforcement Learning 101*
https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292

[14] C. Apte, "Data mining: an industrial research perspective," in IEEE Computational Science and Engineering, vol. 4, no. 2, pp. 6-9, April-June 1997, doi: 10.1109/99.609825.

[15] Ming-Syan Chen, Jiawei Han and P. S. Yu, "Data mining: an overview from a database perspective," in IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 6, pp. 866-883, Dec. 1996, doi: 10.1109/69.553155.

[16] Alyssa Schroer, 2022. *How Sports Analytics Are Used Today, by Teams and Fans*.

https://builtin.com/big-data/big-data-companies-sports

[17] Wikipedia, *Moneyball* https://en.wikipedia.org/wiki/Moneyball

[18] Li RT, Kling SR, Salata MJ, Cupp SA, Sheehan J, Voos JE. Wearable Performance Devices in Sports Medicine. Sports Health. 2016 Jan-Feb;8(1):74-8. doi: 10.1177/1941738115616917.

[19] Devox Software Team, 2023 https://devoxsoftware.com/blog/the-future-of-wearable-technology-for-athletes/

[20] Morrys et al, 2023, *tennis*, Encyclopedia Britannica. https://www.britannica.com/sports/tennis

[21] Cornman, Andre et al. “Machine Learning for Professional Tennis Match Prediction and Betting.” (2017).

[22] Panjan, Andrej & Sarabon, Nejc & Filipcic, Ales. (2010). Prediction of the successfulness of tennis players with machine learning methods. Kinesiology. 42. 98-106.

[23] Gao, Zijian and Kowalczyk, Amanda. ‘Random Forest Model Identifies Serve Strength as a Key Predictor of Tennis Match Outcome’. 1 Jan. 2021: 255 – 262.

[24] Chen, Yangjuin et al. “Final Project Report Real Time Tennis Match Prediction Using Machine Learning.” (2017).

[25] Myerscough, K. (1995). The game with no name: the invention of basketball.

Routledge. 12(1), pp.137-152, doi: 10.1080/09523369508713887

[26] Thabtah, F., Zhang, L. & Abdelhamid, N. NBA Game Result Prediction Using Feature Analysis and Machine Learning. Ann. Data. Sci. 6, 103–116 (2019). https://doi.org/10.1007/s40745-018-00189-x

[27] M. Chen and C. Chen, "Data Mining Computing of Predicting NBA 2019–2020 Regular Season MVP Winner," 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE), Las Vegas, NV, USA, 2020, pp. 1-5, doi: 10.1109/ICACCE49060.2020.9155038.

[28] Vangelis Sarlis, Vasilis Chatziilias, Christos Tjortjis, Dimitris Mandalidis,

"A Data Science approach analysing the Impact of Injuries on Basketball Player and Team Performance", Information Systems, Volume 99, 2021,

https://doi.org/10.1016/j.is.2021.101750.

[29] Papageorgiou, G. & Tjortjis, C. (2022). Data Mining in Sports: Daily NBA Player Performance Prediction [Master’s thesis, International Hellenic University].

International Hellenic University Repository.

https://repository.ihu.edu.gr/xmlui/handle/11544/29991

[30] M. Manoj, R. Prashant, V. Parikh and A. Chaudhary, "American League Baseball Championship 2017 Prediction using AHP," 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2018, pp. 469-473, doi: 10.1109/IC3IoT.2018.8668120.

[31] Holtzman et al, 2023, *baseball*, Encyclopedia Britannica.

https://www.britannica.com/sports/baseball

[32] Huang, M L & Li, Yun-Zhi. (2021). Use of Machine Learning and Deep Learning to Predict the Outcomes of Major League Baseball Matches. Applied Sciences. 11. 4499. 10.3390/app11104499.

[33] Karnuta JM, Luu BC, Haeberle HS, et al. Machine Learning Outperforms Regression Analysis to Predict Next-Season Major League Baseball Player Injuries: Epidemiology and Validation of 13,982 Player-Years from Performance and Injury Profile Trends, 2000-2017. Orthopaedic Journal of Sports Medicine. 2020;8(11).

doi:10.1177/2325967120963046

[34] Britannica, *volleyball,* https://www.britannica.com/sports/volleyball/The-game

[35] De Leeuw, A.-W., Van Der Zwaard, R. & Knobbe, A. (2021). Personalized

Machine Learning Approach to Injury Monitoring in Elite Volleyball Players. European

Journal of Sport Science, pp. 1-10. doi: 10.1080/17461391.2021.1887369

[36] Suda, Shuya, Yasutoshi Makino and Hiroyuki Shinoda. “Prediction of Volleyball Trajectory Using Skeletal Motions of Setter Player.” Proceedings of the 10th Augmented Human International Conference 2019

[37] Yang Tian, Optimization of volleyball motion estimation algorithm based on machine vision and wearable devices, Microprocessors and Microsystems, Volume 81, 2021, ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2020.103750.

[38] Abdullah Erdal Tümer & Sabri Koçer (2017) Prediction of team league’s rankings in volleyball by artificial neural network method, International Journal of Performance Analysis in Sport, 17:3, 202-211, DOI: 10.1080/24748668.2017.1331570

[39] R. Pariath, S. Shah, A. Surve and J. Mittal, "Player Performance Prediction in Football Game," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 1148-1153, doi: 10.1109/ICECA.2018.8474750.

[40] Baboota, Rahul & Kaur, Harleen. (2018). “Predictive analysis and modelling football results using machine learning approach for English Premier League”. International Journal of Forecasting. 35. 10.1016/j.ijforecast.2018.01.003.

[41] K. Apostolou and C. Tjortjis, "Sports Analytics algorithms for performance prediction," 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 2019, pp. 1-4, doi: 10.1109/IISA.2019.8900754.

[42] V. C. Pantzalis and C. Tjortjis, "Sports Analytics for Football League Table and Player Performance Prediction," 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA, Piraeus, Greece, 2020, pp. 1-8, doi: 10.1109/IISA50023.2020.9284352.

[43] Zeng, Z. and Pan, B. (2021). A Machine Learning Model to Predict Player's Positions based on Performance. In Proceedings of the 9th International Conference on Sport Sciences Research and Technology Support; SciTePress, pages 36-42. DOI: 10.5220/0010653300003059

[44] Oliver, Jon & Ayala, Francisco & De Ste Croix, Mark & Lloyd, Rhodri & Myer, Gregory & Read, Paul. (2020). Using machine learning to improve our understanding of injury risk and prediction in elite male youth football players. Journal of Science and Medicine in Sport. 23. 10.1016/j.jsams.2020.04.021.

[45] Martins F, Przednowek K, França C, Lopes H, de Maio Nascimento M, Sarmento H, Marques A, Ihle A, Henriques R, Gouveia ÉR. Predictive Modeling of Injury Risk Based on Body Composition and Selected Physical Fitness Tests for Elite Football Players. J Clin Med. 2022 Aug 22;11(16):4923. doi: 10.3390/jcm11164923.

[46] Healthcare Engineering JO. Retracted: Wearable Device-Based Smart Football Athlete Health Prediction Algorithm Based on Recurrent Neural Networks. J Healthc Eng. 2023 May 24;2023 doi: 10.1155/2023/9879826.

[47] Catapult. Catapult: Provider of Wearable Technology for the Management of Athletes. Available online: https://www.catapultsports.com

[48] Majumdar, A., Bakirov, R., Hodges, D. et al. Machine Learning for Understanding and Predicting Injuries in Football. Sports Med - Open 8, 73 (2022). https://doi.org/10.1186/s40798-022-00465-4

[49] javaTpoint, *Linear Regression in Machine Learning,* https://www.javatpoint.com/linear-regression-in-machine-learning

[50] IBM, *What is linear regression?*

https://www.ibm.com/topics/linear-regres-si-on#:~:text=Linear%20regression%20analysis%20is%20used,is%20called%20the%20independent%20variable.

[51] Wikipedia, *Ridge Regression* https://en.wikipedia.org/wiki/Ridge_regression

[52] Pavan Vadapalli, *What is Ridge Regression in Machine Learning?* ,2023, upGrad, https://www.upgrad.com/blog/what-is-ridge-regression-in-machine-learning/

[53] McDonald, G. C. (2009). Ridge regression. Wiley Interdisciplinary Reviews: Computational Statistics, 1(1), 93–100. doi:10.1002/wics.14

[54] Parmar, A., Katariya, R., Patel, V. (2019). A Review on Random Forest: An Ensemble Classifier. In: Hemanth, J., Fernando, X., Lafata, P., Baig, Z. (eds) International Conference on Intelligent Data Communication Technologies and Internet of Things

[55] tutorialspoint, *Classification Algorithms - Random Forest* https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm

[56] JavaTPoint. *Random Forest Algorithm.* https://www.javatpoint.com/machinelearning-random-forest-algorithm

[57] Wikipedia, *Gradient Boosting* https://en.wikipedia.org/wiki/Gradient_boosting

[58] Bentéjac, C., Csörgő, A. & Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. Artif Intell Rev 54, 1937–1967 (2021). https://doi.org/10.1007/s10462-020-09896-5

[59] GeeksforGeeks, *XGBoost*, https://www.geeksforgeeks.org/xgboost/

[60] Hao Mo, Hejiang Sun, Junjie Liu, Shen Wei, Developing window behavior models for residential buildings using XGBoost algorithm, Energy and Buildings, Volume 205, 2019, 109564, ISSN 0378-7788, https://doi.org/10.1016/j.enbuild.2019.109564.

[61] R. Jain and A. Nayyar, "Predicting Employee Attrition using XGBoost Machine Learning Approach," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2018, pp. 113-120, doi: 10.1109/SYSMART.2018.8746940.

[62] Simplilearn*, What is XGBoost? An Introduction to XGBoost Algorithm in Machine Learning,*https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article#:~:text=XGBoost%20is%20a%20boosting%20algorithm%20that%20uses%20bagging%2C%20which%20trains,with%20many%20features%20to%20consider.

[63] Devadoss, Anitha and T. Antony Alphonnse Ligori. “Forecasting of Stock Prices Using Multi Layer Perceptron.”, Volume: 02, December 2013, Pages: 440-449.

[64] 1.17. Neural network models (supervised). Scikit-learn. https://scikitlearn.org/stable/modules/neural_networks_supervised.html

[65] Computer Science wiki, *Mean absolute error (MAE)* https://computersciencewiki.org/index.php?title=Mean_absolute_error_(MAE)

[66] Stephen Allwright, *What is a good RMSE value? Simply explained*, https://stephenallwright.com/good-rmse-value/

[67] Simplilearn, *Mean Squared Error : Overview, Examples, Concepts and More* https://www.simplilearn.com/tutorials/statistics-tutorial/mean-squared-er-ror#:~:text=The%20Mean%20Squared%20Error%20measures,it%20relates%20to%20a%20function.

[68] Statistics by Jim, *Root Mean Square Error (RMSE)* https://statisticsbyjim.com/regression/root-mean-square-error-rmse/

[69] Stephen Allwright, *What is a good RMSE value? Simply explained* https://stephenallwright.com/good-rmse-value/

[70] Wikipedia, *Mean absolute percentage error,* https://en.wikipedia.org/wiki/Mean_absolute_percentage_error

[71] Jim Frost, *How to interpret R-squared ub Regression Analysis,* Statistics by Jim, https://statisticsbyjim.com/regression/interpret-r-squared-regression/

[72] *Coefficient of Determination, R-squared,* https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html

[73] Octoparse https://www.octoparse.com/

[74] GeeksforGeeks, *Introduction to Dimensionality Reduction,* https://www.geeksforgeeks.org/dimensionality-reduction/

[75] Bianca Williams, Caroline Halloin, Wiebke Löbel, Ferdous Finklea, Elizabeth Lipke, Robert Zweigerdt, Selen Cremaschi, Data-Driven Model Development for Cardiomyocyte Production Experimental Failure Prediction, Computer Aided Chemical Engineering, Elsevier, Volume 48, 2020, Pages 1639-1644, ISBN 9780128233771, https://doi.org/10.1016/B978-0-12-823377-1.50274-3.

[76] Rahul Shah, *Tune Hyperparameters with GridSearchCV,* 2023, https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

[77] Elastic, *Feature importance,* https://www.elastic.co/guide/en/machine-learning/current/ml-feature-importance.html

[78] Chen, Z., & Kwak, D. H. (2023). It's Okay to be Not Okay: An Analysis of Twitter Responses to Naomi Osaka's Withdrawal due to Mental Health Concerns. Communication & Sport, 11(3), 439-461. https://doi.org/10.1177/21674795221141328

[79] Frank Dreyer, Jannik Greif, Kolja Günther, Myra Spiliopoulou, and Uli Niemann. 2022. Data-Driven Prediction of Athletes' Performance Based on Their Social Media Presence. In Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, https://doi.org/10.1007/978-3-031-18840-4_15

[80] Hecksteden A, Schmartz GP, Egyptien Y, Aus der Fünten K, Keller A, Meyer T. Forecasting football injuries by combining screening, monitoring and machine learning. Sci Med Footb. 2023 Aug;7(3):214-228. doi: 10.1080/24733938.2022.2095006

Appendix

Here, we include the feature importance values for the optimal algorithm in each scenario. We have enhanced the presentation with color-coded significance. Green highlights the most crucial feature, blue denotes the second most important, and yellow signifies the third.

Table 8-1: Feature importance values for case 1

Category	Bundesliga		Premier League		La Liga		Serie A		All dataset	
Dataset	all players	top 30%	all players	top 30%	all players	top 30%	all players	top 30%	all players	top 30%
Features / Models	*Ridge Regression*	*Ridge Regression*	*XGBoost*	*XGBoost*	*XGBoost*	*Random Forest*	*XGBoost*	*MLP*	*XGBoost*	*Gradient Boosting*
xG	0.2246	0.1723	0.4222	0.3754	0.3691	0.0624	0.4768	-	0.4451	0.3892
npxG	-0.4984	-0.8313	0.1240	0.0626	0.0231	0.0979	0.1030	-	0.0602	0.0039
npxG_PLUS_xAG	0.1773	0.2195	0.0376	0.0764	0.1969	0.3100	0.0799	-	0.0344	0.1623
xG_90	6.0603	3.6796	0.0585	0.0599	0.0576	0.1330	0.0742	-	0.1293	0.1775
Previous_Gls	0.2859	0.2076	0.1456	0.1519	0.0553	0.0245	0.0157	-	0.1186	0.0851
npxG_90	4.6114	3.2392	0.0634	0.0485	0.0468	0.0329	0.0378	-	0.0323	0.0083
G_MINUS_PK	-0.0135	0.3947	0.0278	0.0464	0.0639	0.0421	0.0128	-	0.0189	0.0055
xG_PLUS_xAG_90	1.7540	6.1006	0.0484	0.0626	0.0977	0.1189	0.1232	-	0.0665	0.0572
G_PLUS_A	-0.1112	-0.1545	0.0501	0.0813	0.0580	0.1588	0.0328	-	0.0472	0.0650
PrgR	0.0030	0.0013	0.0223	0.0351	0.0315	0.0196	0.0437	-	0.0227	0.0358
league	-	-	-	-	-	-	-	-	0.0248	0.0101

Table 8-2: Feature importance values for case 2

Category	Bundesliga		Premier League		La Liga		Serie A		All dataset	
Dataset	all players	top 30%	all players	top 30%	all players	top 30%	all players	top 30%	all players	top 30%
Feature / Models	*XGBoost*	*Random Forest*	*XGBoost*	*Gradient Boosting*	*XGBoost*	*Gradient Boosting*	*XGBoost*	*Gradient Boosting*	*Random Forest*	*Gradient Boost-ing*
Nation	0.0250	0.0224	0.0416	0.0284	0.0128	0.0096	0.0610	0.2017	0.0079	0.0049
Pos	0.0395	0.0259	0.0913	0.0215	0.0650	0.0102	0.0661	0.0040	0.0308	0.0570
Squad	0.0330	0.0363	0.0368	0.0215	0.0466	0.0326	0.0256	0.0295	0.0131	0.0140
Age	0.0378	0.1001	0.0322	0.0216	0.0152	0.0176	0.0485	0.0100	0.0101	0.0189
MP	0.0476	0.0853	0.0338	0.0458	0.0345	0.0677	0.0429	0.0228	0.0295	0.0391
Ast	0.0448	0.0375	0.0409	0.0144	0.0695	0.0400	0.0526	0.0329	0.0130	0.0029
PK	0.0625	0.0371	0.0642	0.0619	0.1050	0.0135	0.1542	0.0009	0.0146	0.0813
CrdY	0.0828	0.0538	0.0477	0.0094	0.0110	0.0389	0.0432	0.0114	0.0113	0.0052
CrdR	0.0263	0.0079	0.0298	0.0024	0.0000	0.0006	0.0212	0.0017	0.0009	0.0003
PrgC	0.0488	0.0728	0.0398	0.0343	0.0605	0.0909	0.0413	0.0202	0.0161	0.0238
PrgP	0.0532	0.0485	0.0449	0.0484	0.0839	0.0351	0.0326	0.1232	0.0318	0.0299
PrgR	0.0909	0.0778	0.0422	0.0347	0.0500	0.1444	0.0878	0.1028	0.0527	0.0717
Previous_Gls	0.4078	0.3944	0.4548	0.6557	0.4460	0.4990	0.3231	0.4389	0.7649	0.6503
league	-	-	-	-	-	-	-	-	0.0031	0.0007

Table 8-3: Feature importance values for case 3

Category	Bundesliga		Premier League		La Liga		Serie A		All dataset	
Dataset	all players	top 30%	all players	top 30%	all players	top 30%	all players	top 30%	all players	top 30%
Feature / Models	*Ridge Regression*	*Gradient Boosting*	*XGBoost*	*Random Forest*	*XGBoost*	*Ridge Regression*	*XGBoost*	*XGBoost*	*XGBoost*	*Gradient Boosting*
Nation	0.0132	0.0034	0.0115	0.0050	0.0057	-0.0046	0.0157	0.0173	0.0157	0.0027
Pos	0.0148	0.0115	0.0227	0.0009	0.0093	0.0072	0.0257	0.0131	0.0133	0.0214
Squad	0.0030	0.0072	0.0142	0.0186	0.0097	0.0803	0.0117	0.0183	0.0153	0.0105
Age	-0.0791	0.0058	0.0091	0.0038	0.0117	0.0401	0.0154	0.0179	0.0107	0.0108
Born	-0.1159	0.0144	0.0199	0.0130	0.0144	-0.0402	0.0260	0.0234	0.0138	0.0010
MP	0.0275	0.0198	0.0133	0.0011	0.0121	-0.0244	0.0138	0.0093	0.0122	0.0064
Starts	-0.0107	0.0174	0.0168	0.0040	0.0120	0.0138	0.0137	0.0128	0.0098	0.0017
Min	-0.0003	0.0146	0.0175	0.0096	0.0135	0.0000	0.0114	0.0137	0.0146	0.0289
Ast	0.0135	0.0264	0.0076	0.0035	0.0162	0.0252	0.0121	0.0099	0.0064	0.0003
G_PLUS_A	0.0574	0.0277	0.0256	0.0402	0.0572	0.1166	0.0391	0.0069	0.0101	0.0356
G_MINUS_PK	0.0516	0.0053	0.0115	0.0091	0.0199	0.0675	0.0115	0.0138	0.0107	0.0004
PK	-0.0076	0.0065	0.0245	0.0306	0.0097	0.0239	0.0358	0.0259	0.0209	0.0299
PKatt	0.2039	0.0045	0.0231	0.0067	0.0197	0.0267	0.0116	0.0175	0.0162	0.0119
CrdY	-0.1716	0.0347	0.0177	0.0019	0.0091	-0.0190	0.0154	0.0126	0.0082	0.0000
CrdR	0.6074	0.0065	0.0089	0.0023	0.0126	-0.0010	0.0000	0.0241	0.0047	0.0001
xG	0.0479	0.0004	0.2478	0.4390	0.1822	0.0878	0.2107	0.1322	0.3425	0.3582
npxG	-0.8396	0.0019	0.0743	0.0372	0.0291	0.0666	0.0518	0.1286	0.0126	0.0166
xAG	-0.7863	0.0104	0.0140	0.0193	0.0269	0.0220	0.0207	0.0312	0.0230	0.0264
npxG_PLUS_xAG	0.7867	0.0317	0.0249	0.0267	0.0682	0.0889	0.1111	0.0940	0.0215	0.1597
PrgC	-0.0001	0.0086	0.0135	0.0137	0.0194	0.0189	0.0109	0.0105	0.0117	0.0136
PrgP	0.0031	0.0039	0.0107	0.0071	0.0140	-0.0149	0.0111	0.0108	0.0099	0.0060
PrgR	0.0018	0.0365	0.0145	0.0077	0.0097	0.0018	0.0202	0.0216	0.0150	0.0080

Gls_90	1.2369	0.0559	0.0540	0.0574	0.1098	0.0032	0.0281	0.0618	0.0242	0.0306
Ast_90	-1.8926	0.0153	0.0118	0.0037	0.0129	0.0006	0.0162	0.0195	0.0099	0.0021
G_PLUS_A_90	-0.7462	0.0154	0.0199	0.0032	0.0098	0.0037	0.0236	0.0259	0.0099	0.0063
G_MINUS_PK_90	0.6126	0.0091	0.0495	0.0246	0.0283	0.0023	0.0120	0.0159	0.0161	0.0073
G_PLUS_A_MINUS_PK_90	-1.1783	0.0027	0.0179	0.0025	0.0142	0.0029	0.0171	0.0236	0.0148	0.0008
xG_90	3.1888	0.0215	0.0325	0.0129	0.0582	0.0034	0.0346	0.0383	0.0890	0.1018
xAG_90	0.1041	0.0189	0.0107	0.0086	0.0203	0.0004	0.0086	0.0159	0.0087	0.0094
xG_PLUS_xAG_90	3.2073	0.5094	0.0232	0.0098	0.0833	0.0038	0.0785	0.0511	0.0443	0.0285
npxG_90	2.3668	0.0094	0.0357	0.0046	0.0257	0.0026	0.0225	0.0239	0.0258	0.0111
npxG_PLUS_xAG_90	2.1432	0.0114	0.0209	0.0026	0.0269	0.0030	0.0354	0.0239	0.0147	0.0030
Previous_Gls	0.0439	0.0320	0.0802	0.1695	0.0285	0.0914	0.0281	0.0349	0.1090	0.0482
league	-	-	-	-	-	-	-	-	0.0149	0.0008