Universiteit Leiden
The Netherlands

Opleiding Informatica

Machine Learning for
Football Player Scouting

Bram van Ommen

Supervisors:
Bas Kruiswijk & Walter Kosters

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

05/12/2023

Abstract

Football is one of the biggest sports in the world, with millions of people practicing or watching the sport. Clubs are constantly trying to improve their teams and new techniques keep begin developed. The growing amount of data that is available creates more and more opportunities. This allows for machine learning to be used in this process as well.

The purpose of the model proposed in the research is to examine how machine learning can be implemented to support football player scouting. This process is an important part of building and improving the team, however, it can be very time consuming and costly. By implementing machine learning it is possible to make this process quicker and less costly.

By doing literature research on multiple important subjects like data collection and multiple machine learning algorithms, this research will propose a proof of concept of a system that uses machine learning to scout football players.

This system will be defined in an UML use case diagram and an UML sequence diagram. These models will function as guidelines to ultimately build the model in further research.

Contents

1 Introduction

Football is one of the most popular sports in the world. According to (Toma & Campobasso, 2023) the European football competition alone already made €28.4 billion in revenues. The biggest five competitions in Europe made €15.6 billion in revenues, which is 6% more than the year before. The revenues in football are growing, and so is growth of number of fans. The last group is accounting for the biggest part of the revenue expansion, indicating that the growth of football is still going.

To keep the sport and their fanbase growing, tournaments and competitions must be thrilling enough to watch. This necessarily means that tournaments and competitions must have an uncertain outcome to guarantee for an exciting tournament or competition. To keep a chance of winning, a football club must keep up with the level of their opponents in the competition. Even better is for them to keep improving so they pass this level and win the competition. Therefore, each year a football club starts the competition by revaluating their team and coach, possibly hiring a new coach or new players. Essentially, they are building a new team, and while building a team the football club relies mainly on their financial assets, but this can also be influenced by specific players or other higher employees at the football club. It is important to know that building a team has a certain influence in creating future corporate value for a football club. Thus, building a team is a very decisive process and must be considered very carefully (Toma & Campobasso, 2023).

To support this process of building a team, football clubs have made more and more use of data analytics. Since recent years football teams started to use computer technologies and sensors to measure different aspects of football games. This is divided into team statistics and individual player statistics. Football can also rely on professional statistical analysis firms like *ProZone* and *OPTASports* by *STATS* (ProZone is now part of *STATS*) (StatsPerform, n.d.), which record and provide data about football clubs, individual players and football leagues like the English Premier League and the Spanish La Liga. These bureaus collect big data and provide statistics which can be used to analyse a football game or player, and based on these statistics teams or players can improve themselves. These statistics can also be used to collect data on potential new players and talents and help football clubs in their decision on buying or selling new players (Cintia, Rinzivillo, & Pappalardo, 2015).

However, nowadays football clubs still make use of a physical scouting team that is active all over the world. Usually, they have one head of scouting, leading a team of other scouts. These scouts visit games from players they potentially want to buy. They watch closely and make notes about the player. If these players match their wishes, they will start the bidding process, eventually buying the player for a certain amount of money. However, these scouts have a salary, so they cost money for the club. Next to that, they are humans, so they cannot work without resting. Using data to replace this process is a great opportunity.

This research is proof of concept and will ultimately provide a UML model. The purpose of the concept is to be an addition to a football's scouting procedure. The user can input a player profile according to their desires, which then will be processed in the model to find players that fit this profile. This model will implement machine learning algorithms to make use of the available data about football players. Eventually, the research will answer the following question: how can machine learning be implemented to support financial decision making on player transfers in football using big data.

In Section 3.1, this research will examine how data is collected, how it is available, and how it can be used. In Section 3.2, the most valuable features of football players will be researched. These features can be used to find players with similar physique and similar performance, but also to estimate a player's market value. In Section 3.3, the research will find out which machine learning algorithms are the best option for the model's purpose. This research is a bachelor project, done under supervision of Bas Kruiswijk and Walter Kosters, at Universiteit Leiden.

2 Method

To acquire the information to build the model, literature research will be done. The literature review will exist of three steps that eventually will be combined to design the model. For each step, several related articles will be compared to each other, to finally give a better view on the corresponding part. A more schematic overview of the research process is shown in Figure 1.

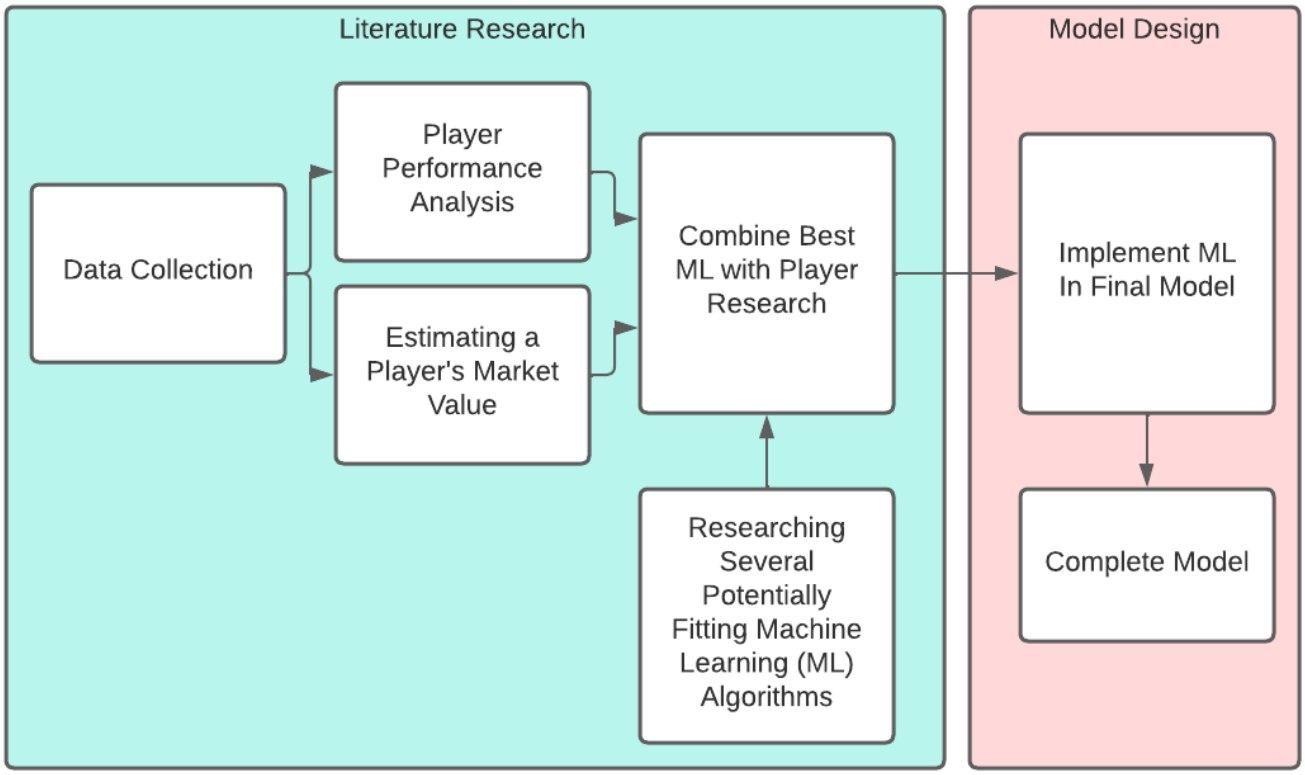

Figure 1: Schematic view of the research process

2.1 Collecting Data

First step in the process is to find out how the data is collected. Knowing how the data is collected gives a better insight, making it easier to understand each feature that is kept track of. It is also important that it is possible to examine the data, so that the most useful features can be extracted. To make this possible, the data must be accessible, therefore it is necessary to know who is collecting the data and where it is stored. Eventually, the research will extract the features that will be used in the model. To complete these steps, the research will be done by examining other related articles about the subject, and by looking into the data.

2.2 Player Features

The next step is to research football players. In this phase, this work finds out what makes players unique, what makes them similar and what determines their market value. This will be researched by reading related work. Football players are defined by numerous features, and the goal is to find the most important features. These features define a major part of the football player and its capabilities, giving insight in what makes a player stand out. When that is achieved, it is possible to compare players to each other. The most important features will be defined and used in the model.

2.3 Machine Learning Algorithms

The last step is to find the best fitting machine learning algorithms to process the data. Several machine learning algorithms that might be useful for the model will be proposed. To find these algorithms, work that is related to the research question will be examined. This will be supported by doing research about the algorithms on their own, and also through literature research. The algorithms will be compared, eventually leading to a set of algorithms that are best fitted for the model.

2.4 Designing the Models

The actual purpose of this research is to design a model that can be used to be an addition to, or even replace, the scouting process in football. The literature review that has been done will be combined in this model. The model will be described by a UML use case diagram and UML sequence diagram. By designing the model, the research question will be answered. The model will show how machine learning can be implemented in the financial decision making to support player transfers in football using data. See Figure 1.

3 Literature

To design the model, it is important to research the multiple aspects that must be considered. It is necessary to know how the data is collected, where it can be acquired, what player features are important, and which machine learning algorithms will be best practice for this case.

3.1 Data Collection

The analysis of team and player data by tracking systems is already present in many professional sports such as football. These systems provide large amounts of player and ball tracking data however, the large-scale mining of this data is very limited due to the difficulty in representing dynamic multi-agent trajectories (Bialkowski et al., 2016). Therefore, research is done using spatio-temporal data to discover team structures and analyse individual player actions.

Spatio-temporal data is a sequence of samples that contain a timestamp and a location of actions that happen on the field. For team sports, there are typically two types of spatio-temporal data. One of them is object trajectories, which describe the movement of players or the ball. The other data are event logs of time and location of match events, which describe actions such as passes, shots at goal or fouls (Gudmundsson & Horton, 2017). These datasets, when combined, can provide a richer explanation of a certain football game. When used, individual player actions can be analysed, resulting in a clear overview of a player's statistics during a single game, or even a whole season.

The data is collected using optical tracking systems. These systems contain cameras that are used to capture player movement. These images are then used to compute the trajectories (Gudmundsson & Horton, 2017). Next to that, device tracking systems are used as well. These are little devices that are attached to a player's clothing or embedded into the football. This data is used by the clubs themselves, however, nowadays it is becoming way more usual to have external parties involved in this process. These parties can also use the data to do research about numerous aspects in football.

An example is *OPTASports* by *STATS*, a renowned British sports analytics company. They collect data from football matches all around the world. They work with 500 teams, supporting them in data collection so that they can improve their performance on the field based on this data. They use the described tracking systems, as well as teams of employees to collect, check and validate the data. This enables them to give a detailed match report in 24 hours (StatsPerform, n.d.). To acquire this data, Stats Perform must grant permission to use it. It is possible to buy the data from Stats Perform, thus this won't be free. However, there are sites like *whoscored.com* that use this data as well and these sites are freely accessible.

A quick look on *whoscored.com* shows the available data. It is possible to look for many different player statistics. It contains defensive features such as tackles, interceptions, fouls, and clearances, offensive features such as goals,

assists, shots, and key passes, and passing statistics such as pass completion, crosses, long balls and through balls. Next to that, it is also possible to find the player's physical characteristics, like age and height. This site will be used for this research, as well as available data from *OPTASports*.

3.2 Player Performance Analysis

To find similar players across different or same leagues all over the world, it's important to consider different features of players. These features will be divided into four categories: player position on the field, player roles, which can be numerous different playing styles, player performance, like goals and assists and player characteristics, like height and age.

3.2.1 Player Position

Player position defines the location of the player on the field. For example, a striker is playing in the opponent's part of the field, usually close to the opponent's goal. On the other hand, a defender will most likely play close to its own team's goal. Each position requires different abilities. Strikers must be able to score goals, hence their shot accuracy and shot power will probably be greater than the defender's shot accuracy and shot power (García-Aliaga, Marquina, López, Rodríguez-González, & Luengo-Sanchez, 2020).

When finding similar players across the world, comparing positions is an obvious step in the process of finding the best-fitting profiles. This is because we are looking for similar players with similar capabilities. Hence, a striker is not the player we want when we are looking for a defender. Moreover, dividing players in smaller clusters will make it easier and less time-consuming for the algorithms used later in the model, since there are fewer samples.

To keep track of the players' positions, their spatio-temporal data can be used. In the previous section, this is explained as data that consists of a sequence of samples that contain location and timestamps of events that happen on the field. This includes the location of ball or player movements on the field. It also contains data that keeps track of player events in football matches. Therefore, the the x and y coordinates of football players are stored as well. This data can be used to calculate the average location of a player on the field (Bialkowski et al., 2016).

3.2.2 Player Roles

Each player is assigned to a position, but this does not mean every player is playing this position in the same way. Therefore, each player can be assigned certain roles on the field. For example, according to (Aalbers & Van Haaren, 2018), there are five different roles for midfielders:

1. Ball-Winning Midfielder: this player is mainly tasked with regaining the possession of the ball. This player's focus will be on pressing the opponent's team and collecting the ball from them. This player is constantly trying to disturb the opponent's build-up. When in possession, or after regaining it, this player will play a simple passing game.

2. Holding Midfielder: this player is mainly tasked with protecting the defensive line. Opposed to the ball-winning midfielder, this player is defences passively and keeps the defensive line compact. When in possession, this player dictates the pace of the game. This player requires to be highly intelligent.

3. Deep-Lying Playmaker: this player is also tasked with dictating the pace of the game; however, this player is less tasked with defending. His main task is more focused on build-up. These players usually have high level of vision and timing and are technically capable of giving precise and excellent passes.

4. Box-To-Box: this player is more dynamic. This player's focus is on excellent positioning. He must be in defence as in offense as well. These players collect the ball, dribble, or pass forward and create chances. These players usually have high stamina; hence they are called box-to-box midfielders.

5. Advanced Playmakers: these players are the most important creators of the team. They are highly intelligent, technically skilled and have excellent vision. They can give excellent short- and long-range passes, create chances and score goals as well.

These are the different roles that midfielders can adapt. However, players playing in other positions can adapt different roles as well.

3.2.3 Player Performance and Player Characteristics

Player performance is measured by their statistics through the seasons, or even their whole careers. These statistics include goals and assists, but also yellow and red cards. The features that will be considered are selected according to the highlighted statistics shown on *www.whoscored.com*. These statistics are collected from the *OPTA* database. For these statistics see Table 1, 2, 3, and 4.

Table 1: General Statistics of football layers.

FEATURES	**DESCRIPTION**
Appearances	How many games the player has played.
Minutes Played	How many minutes the player has played for a team.

Table 2: Defensive statistics of football players.

FEATURE	**DESCRIPTION**
Tackles	Tackles can be sliding or standing tackles, it tells how many times the player has dispossessed an opponent.
Interceptions per Game	An interception is when a player successfully intercepts an opponent's pass to another player.
Fouls per Game	Average of how many times a player made an illegal action.
Clearances per Game	Average of how many times the player successfully kicked a ball away from his own goal.
Dribbled Past per Game	Average of how many times the opponent successfully dribbled past the player.
Blocks per Game	Average of how many times a player blocked an opponent's shot on his own goal.

Table 3: Offensive statistics of football players.

FEATURE	**DESCRIPTION**
Goals per Season	How many times the player has scored per season.
Assists per Season	How many times the player provided an assist for a teammate per season.
Shots per Game	Average of how many times the player shot on the opponent's goal per game.
Key Passes per Game	Average of how many times the player successfully delivered the final pass to a teammate, who then shot on goal. This is disregarding the outcome. Statistic is per game.
Dribbles per Game	Average of how many times the player made a successful dribble past an opponents per game.
Offsides per Game	Average of how many times the player has stood offside per game.
Dispossessed per Game	Average of how many times the player lost the ball to an opponent per game.

Table 4: Pass statistics of football players.

FEATURE	DESCRIPTION
Average Number of Passes per Game	Average of how many times a player passes the ball into a teammate's direction, per game.
Pass Success Percentage	Percentage of how many times the player's passes reach their teammate.
Crosses per Game	Average of how many times a player successfully delivers a cross to a teammate. A cross is a ball passed from a wide position to a teammate standing around the opponent's goal. Statistic is per game.
Long Balls per Game	Average of how many times the player passed the ball to a teammate over a medium/long distance per game.
Through Balls per Game	Average of how many times the player has passed the ball to a teammate through the opponent's last line of defence per game.

In addition to player performance, we will also consider player characteristics. These characteristics are about the physical features of a player. This allows for the physical comparison between two football players. This comparison will exclusively be to compare the two players based on physical features, this will not include features like speed and agility. Table 5 contains the considered features.

Table 5: Physical characteristics of football players.

FEATURE	DEFINITION
Height	Physical length of the player in centimetres
Age	Age of the player in years
Footedness	Preferred foot when playing football. Could be left, right or both.
Weight	Physical weight of the player in kilograms

3.3 Estimating a Football Player's Value

According to the research of (Al-Asadi & Taşdemir, 2022) the most common indicators for determining a player's market value can be put into three different categories: *player characteristics*, *player performance* and *player popularity*. Player characteristics will be described as in the previous section, however, for the next step demographic attributes will be considered as well, since this can influence the player's market value. Player performance will be described in the same way. In addition, player popularity will be included as well. This describes the popularity of a football player in the football community.

Following the research of (Al-Asadi & Taşdemir, 2022), as well as research done by (Müller, Simons, & Weinmann, 2017) and (Behravan & Razavi, 2021). this paper defines multiple attributes that will be considered while estimating a player's market value. In Table 6, 7 and 8, the features are listed with their corresponding definitions. Some features include an explanation to better understand why these features are included.

Table 6: Physical characteristics of a football player

FEATURE	DESCRIPTION
Age	Age of the player. Older players are more sensitive to injuries. They also have fewer years left to play football.
Height	Physical height of the player. Longer players have a bigger likelihood of heading a ball away defensively or scoring a header. This can affect the value, depending on the preference of the team and manager.

Table 7: Player performance features of a football player.

FEATURE	DESCRIPTION
Minutes Played	How many minutes a player has played for his team. Players with more minutes played are more likely to be very important for a certain team than players that have played zero minutes. These players are more likely to be benched because the team has a better player for his position, therefore they are will probably cost less.
Goals	How many goals the player has scored. This includes headers, penalties, free-kicks, and open-play goals. A player that scores more goals is of great value for a football team.
Assists	How many assists are provided by a player. Players with higher assist ratings have higher goal contribution statistics. These players are more likely to help teams scoring goals, which can be of great value.
Average Number of Passes per Game	Average of how many times a player passes the ball into his teammate's direction, per game.
Pass Success Percentage	Percentage of how many times the player's passes reach their teammate.
Dribbles per Game	Average of how many times the player made a successful dribble past his opponents.
Interceptions per Game	Average of how many interceptions a player makes per game. An interception is when a player successfully intercepts an opponent's pass to another player.
Fouls per Game	Average of how many times a player made an illegal action.
Clearances per Game	Average of how many times the player successfully kicked a ball away from his own goal.
Fouls per Game	Average of how many times a player made an illegal action
Yellow cards and red cards	The more cards a player receives, the higher the chance will be that this player will be suspended, and the coach cannot put him in the squad. Therefore, the player value can be lower if the player receives more cards.

Table 8: Popularity feature of football players.

FEATURE	DESCRIPTION
Popularity	More popular players have a higher commercial value. These players attract more fans, which can enlarge a club's revenue.

Note that player performance differs for different positions. Goals scored is a more important statistic for attacking players than for defensive players. The other way around, for a defender it is more important that his duelling statistics are higher. There lays much more value in quality tackles, good clearances, and interceptions for them than for attacking players. An exception in this research is goalkeepers. Since goalkeepers are no in-field players, they require a different approach. Therefore, this paper will not consider goalkeepers.

3.4 Machine Learning Algorithms

Machine Learning (ML) exists of many different algorithms and techniques (Mahesh, 2020). This section of the paper will describe some of the most interesting classification, regression, and clustering algorithms.

3.4.1 Classification Algorithms

Classification algorithms are used to sort data objects into different qualitative classes (Zhang, Liu, Zhang, & Almpanidis, 2017). Three classification algorithms are explained in the section below.

RIPPER-algorithm
The RIPPER (for Repeated Incremental Pruning Produce Error Reduction) Algorithm (Asadi & Shahrabi, 2016) is a rule-based classification algorithm that derives a set of rules from the training set. The algorithm can be divided

into three stages: rule set building, rule set pruning, and rule optimization.

In the first stage RIPPER is learning the rules. The classes are initially ordered according to their prior probabilities and the algorithm is learning those class-based rules according to this order. When a rule set is defined, this set will be pruned, and finally the obtained rules will be improved. In this process, each rule that is learned will be grown, and subsequently pruned.

In the second stage, RIPPER uses the MDL (Minimum Description Length) measure. This is done to stop rule learning for a class. Whenever the description length of a learned set of rules is at least 64 bits longer than the MDL obtained so far, the algorithm stops adding a rule to the rule set for this class. This stops the rule sets from keeping on growing. When this is achieved, the existing set of learned rules is pruned according to description length. Rules with high costs are eliminated from the learning process.

The third stage is the optimization phase. In this phase the algorithm will replace each rule in the decision list by two new rules. These rules are learned based on the entire set of instances that the rule covers in the training data. The one whose decision list has the shortest length will remain; the others will be eliminated.

When the process is finished, the algorithm ends up with a list of rules per existing class. The dataset from which the training set was acquired can then be put into the algorithm to be finally classified. The final output will be a classified dataset according to the RIPPER algorithm (Asadi & Shahrabi, 2016).

Decision Tree

Decision Tree classifies groups through a tree. The tree consists of nodes, branches, and leaves. This algorithm is a relatively simple supervised learning algorithm. Each node contains a question for the algorithm, and each branch that is connected to this node represents an answer to this question. That means, that by putting in an object, the algorithm will go down the decision tree, passing through the nodes until it eventually reaches a leave (Mahesh, 2020). This leave contains the class in which the object will be placed. This algorithm can also be used for regression tasks, an example is random forest regression (Kotsiantis, 2011) however this will be explained in more detail in a later section.

Support Vector Machine

The Support Vector Machine (SVM) algorithm is a supervised machine learning algorithm that can be used for classification, regression, and outlier detection. The focus in this part is on classification by SVM. SVM can be divided into linear and nonlinear models. These models are linear if the original data can be divided linearly (e.g., straight line). However, the model is nonlinear if the original data cannot be divided linearly but can be transformed to a different feature space that allows the data to be divided linearly (Suthaharan, 2016).

For two-class classification using linear SVM we need to create a data domain and divide this domain into two subdomains. With the straight-line equation $ax + b = 0$, we can create these subdomains by simply drawing a line in the original data domain (Suthaharan, 2016). This line is called the hyperplane, and the algorithm ultimately looks for the best hyperplane. This is the hyperplane that maximizes the distance between the two nearest objects in both subdomains. Generally, the higher this distance is, the lower the generalization error is. When this error is lower, the algorithm is less likely to experience overfitting (Gold & Sollich, 2003).

Not all classification can be done linearly. Sometimes, it is not possible to simply draw a line to create two subdomains each containing objects that fit a type of class. In these cases, the set of features needs to be transformed using a polynomial kernel. By doing this, a set of p features may be transformed to a new set of q features. This new space is called the feature space, and the advantage of this is that non-separable classes can be made separable by choosing the right kernel. Finding the right kernel is hard, however, when done correctly, the feature space of q features will ultimately allow for linear division between objects (Suthaharan, 2016).

3.4.2 Regression Algorithms

Regression algorithms are used for regression analysis. This is explained by (Ter Braak & Looman, 1995) as "a statistical method that can be used to explore the relation between species and environment, on the basis of observations on species and environmental variables at series of sites." Three regression algorithms are explained in the section below.

Linear Regression
Linear regression models are supervised machine learning models based on the principle of finding the best fit linear line between dependent and an independent variable (Rong & Bao-Wen, 2018). The formula the algorithm uses is expressed as $y = a_0 + a_1x + e$. In this formula, y is the dependent variable, a_0 the constant term, a_1 the regression coefficient, x the independent variable, and e is the random error. This random error is used to express the influence of random factors on the dependent variable (Rong & Bao-Wen, 2018).

Linear regression can be used for simple, two-variable linear regression but it can also be expanded to be used for multiple linear regression. This can be useful for regression analysis of objects with more than two features. The formula will simply be extended by adding more independent variables, in combination with their coefficient, to the formula.

Random Forest Regression
Random Forest regression is a supervised machine learning algorithm that uses ensemble learning method for regression (Li et al., 2018). Ensemble learning means that the algorithm makes use of multiple machine learning methods. It combines the predictions of those algorithms to make a single, more accurate prediction.

Random forest generates multiple decision trees. All those decision trees act as regression functions on their own. The final output is the average of all those decision trees combined. A decision tree does not assume any prior parameters, nor defines any fixed tree structure. Each decision tree consists of decision nodes and leaf nodes. During the training process of the decision tree, all the input data will be split at each decision node in the tree until a stopping criterion is satisfied or splitting any further will not give any more advantage. This will create a set of optimized split functions. This splitting will go on until the final nodes are reached, which are the leaf nodes. The final step of the training process is to create a prediction function based on the set of split functions (Kotsiantis, 2011) and (Li et al., 2018).
The decision trees that are created in the random forest training process are not correlated in any way, and each tree has a random subset of predictors, hence it is called random forest. The random forest is built by combining those decision trees using the bagging algorithm. This algorithm randomly samples a feature subset and/or training data subset for each decision tree. This generates an output per decision tree. This output will then be added to a set with all the other outcomes of the other decision trees in the random forest. The outcome of the random forest is the average of this set (Li et al, 2018).

K-Nearest Neighbours Regression
K-Nearest Neighbours (k-NN) is a supervised machine learning algorithm. The algorithm is given an unlabelled object and will then search for the set of k most similar objects. These objects are called neighbours. This selection process will be done according to the features of the objects. Similar features will cause for similar objects, meaning these neighbours lie close to the main object in the data space. It is a strong algorithm in case of large datasets and low dimensions (Krämer, 2013). This algorithm can be used for both classification and regression. For this section, the focus will lie on utilizing the algorithm for regression tasks.

In k-NN regression, this target object is usually a sequence of values. The algorithm searches for objects that contain similar sequences and combines these sequences together to predict the next value of the target sequence. By repeating this process, the algorithm will eventually complete the prediction. Essentially, the underlying intuition is that any consistent linear or nonlinear data generating process causes certain patterns of behaviour. The algorithm is learning by observing the outcome over time, eventually creating a certain set of rules which can be used for calculating any further predictions (Alqahtani & Crone, 2013).

To find the nearest neighbours in the data space, the algorithm calculates the Euclidian distance between the target

object and the objects around it. Essentially, the k objects that lie within the closest radius of the target object are selected as the target objects nearest neighbours, implying that these objects show the biggest similarities. These objects will be used for to predict the outcomes (Alqahtani & Crone, 2013).

3.4.3 Clustering Algorithms

Clustering algorithms are used to creates clusters of data with similar characteristics. The goal is to find hidden data structures, rather than accurate classification of objects (Xu & Wunsch, 2005). Two clustering algorithms are explained in the section below.

K-Means Algorithm
K-means is an unsupervised clustering algorithm. The k-means algorithm depends on the value k. This value must always be specified when using the algorithm. The value k defines how many clusters will be generated. All the data that is put into the algorithm will be spread across an n-dimensional space, where n will be defined by how many features each data point has. Then, the algorithm will first randomly select k centres, where each centre represents a data point in the n-dimensional space. The next step in the algorithm is that each data point will be taken to the nearest centre. This is usually done by using the Euclidian distance between the centre and another data point. This is done until each data point is connected to a cluster. The final step is to recalculate the average of the early formed clusters. This step continues repeatedly until the criterion function reaches the smallest distance. The outcome of the algorithm will be k clusters with data points, each cluster forming a group of similar data (Na, Xumin, & Yong, 2010).

DBSCAN Algorithm
DBSCAN is an unsupervised, density-based, clustering algorithm. It is designed to cluster data with arbitrary shapes. The idea of DBSCAN is that for each cluster centre, a given radius contains a minimum number of points (Khan, Rehman, Aziz, Fong, & Sarasvady, 2014). The difference with k-means is that this algorithm does not need a pre-specified value to determine the number of clusters. However, DBSCAN requires a specified radius and a specified minimum number of points.

DBSCAN searches for clusters by checking the radius of each point in the dataset. This radius defines the neighbourhood of a certain point p. If this neighbourhood contains fewer than the number of minimal other points that is pre-specified, point p will not be a centre point. Eventually points with enough other points in their neighbourhood will be defined as centre point and clusters will be formed. This may also involve the merge of two clusters, which will result in one bigger cluster. This process will go on until no more points can be added to another cluster. The outcome of the algorithm is an unspecified number of clusters, each representing a group of similar objects (Rehman et al., 2014).

4 Model

The next step in this research to combine the previous sections and propose a model. The purpose of this model is to give more clarity about how machine learning can be used in the modern-day scouting process of football. This includes searching for players, but also to value these players according to their performance and their characteristics. The model will combine the previous sections about player performance and player valuation, with the various machine learning techniques that have been described. The data that will be put into the model is a large set of football players, each player having a collection of features, described in Section 3.2, that will be considered by the model. When speaking of a data point, this point represents a football player. The model exists of two sections, mainly because the training of the machine learning algorithm does not need to be done every time the system is used.

4.1 Textual Description of the Model

Setting up the model

1. *The Dataset*
   The dataset that will be used at the start of the process will consist of 29 separate features. These features can be split up into four different groups, each group being important for a certain phase in the model (values in parentheses represent the number of features):

   (a) *Market value (1)*
   For most players, the dataset contains the market value on before-hand, however, for some players this data is missing. These missing gaps will be filled in by the model.

   (b) *Spatial data (2)*
   This data describes the location of the players on the field. In this dataset, the x and y coordinates are already averaged out, creating the average x-coordinate $\bar{x}$, and the average y-coordinate $\bar{y}$. These average coordinates will be used in the model to determine player positions.

   (c) *Physical characteristics and performance (24)*
   These features are explained in Section 3.2. This section contains two tables in which the features are addressed in more detail. These features will be important in the final step of the model, the identification of similar types of players.

   (d) *Player popularity (2)*
   This data describes the player's nationality and popularity. These extra features can be used to determine a player's market value, which will be a necessary feature to provide a complete output.

2. *Pricing the Players and Training the Algorithm*
   The next phase in the model is to assign to each player a market value. To do this, the model will combine the database that is collected from (*Transfermarkt*, n.d.), which contains a large database of players and their market values, with the original database mentioned earlier in this paper. Since 2000, *Transfermarkt* is specialized in assigning market values to players. The valuation is done by experts, who continuously discuss a player's market value online and in real life. They are supported by thousands of users that are allowed to discuss these prices as well. They also have the possibility to suggest their opinion on player's their market value, this can be done always.

   However, there is a chance that some players are not found in this database, meaning the model has no value assigned to these players. Therefore, the model will implement a machine learning method to assign a value to these players. (Al-Asadi & Taşdemir, 2022) tested four algorithms for this: linear regression, multiple linear regression, regression tree and random forest regression. The last one came out as the most efficient.

   With the random forest algorithm, the dataset will be split into a train and a test set. The training set is 30% of the original dataset, the test set will be 70% of the original dataset. The training set will be used to train this algorithm, and when it is put into the algorithm, this will create a random forest. Each decision tree in this forest will be formed by randomly selecting a sample of rows, and at each node it will randomly select a different sample of features for splitting. In the end, all the outcomes of the decision trees will be added up, and this number will be averaged out. The output will be the estimated market value of the player.

The reason for this phase coming first, is that random forest regression works better on larger groups of data. In the next steps, the model will start grouping the data in smaller datasets, making the random forest regression less efficient (Li et al. 2018).

In the next steps, the model will make use of the data that is prepared in this section. The dataset now contains the 24 variables mentioned in Section 3.2, plus the extra market value variable. This means the dataset contains a total of 25 variables. This complete dataset will be passed on to the next phase of the model.

Using the System to Scout a Football Player

3. *Insert a player profile*
   The first step in using the system is to insert a profile that fits the desired player. The user fills in the values for the variables mentioned in Section 3.2. For example, one of the variables is goals. The user can fill in a value for how many goals he wants the player to have scored, and the model will take this value into account when it is searching for a player. The main idea is that, by filling in the variables, the user creates an imaginary player that fits the desires of the user. An example of how this works for a player like Greg Leigh, who played for NAC Breda in the Eredivisie 2018/2019 season, is shown in Table 9.

Table 9: Example of how a player profile will be created.

Appear- ances	**Minutes Played**	**Tackles per Game**	...	**Footedness**	**Weight**	**Position***
14	1302	1.2		Left	73	Left fielder

(a) Example of Greg Leigh's 2018/2019 season for NAC Breda in the Dutch Eredivisie. The dots represent the remaining 19 features mentioned in Table 1, 2, 3, 4 and 5 from Section 3.2. * Position will be explained in more detail in phase 4 of the model.

When the player profile is created, and thus the imaginary player, this player will be added to the dataset that will be used by the model. When the model is processing, the imaginary player is the main target of the model. This means, that each time when the model splits the data, it will continue with the dataset containing this imaginary player. This will be further explained in the next steps.

4. *Grouping Players According to Their Positions*
   The next step in the process is to create smaller clusters from the dataset. Each cluster will represent a different position on the field. In total, eight clusters will be created, according to the following possible positions: 1) left/central back, 2) right/central back, 3) central fielder, 4) left fielder, 5) right fielder, 6) left forward, 7) right forward and 8) central forward. This is almost equal to what (García-Aliaga et al., 2020) suggest in their research, however they mention defensive midfielders and attacking midfielders. These two positions are combined under the central defender position.

   For all the players that are in the dataset, the centre of performance u will be considered. The centre of performance describes the position of a player relative to its teammates. The centre of performance u in match m is denoted as $c_u^m = (\bar{x}_u^m, \bar{y}_u^m)$, and $\bar{x}_u^m$ and $\bar{y}_u^m$, where $\bar{x}_u^m$ and $\bar{y}_u^m$ are the average coordinates of the player in the field. The average coordinates are simply the means of the x and y coordinates that are tracked per player, per match. This is hard to implement by hand, therefore, the user can select one out of the eight positions that have been mentioned before.

   Now, the model contains a dataset in which each player object contains its centre of performance u. To find out which player belongs to which cluster, and thus to which position, the model will make use of the k-means algorithm. Here, k denotes for how many clusters must be created, therefore $k = 8$. Each cluster has its own centre, and this centre is used to connect to data points that are nearby. This means that each centre of performance u is connected to its nearest cluster centre. This process will keep on going until the eight different clusters have been formed and no data point is left out. The outcome of this process will be eight clusters, also known as groups, containing players that have the same position on the field.

   Now that the model is left with smaller groups of players, the model can select the cluster that will be used in the next step. The cluster that will be used is the cluster that contains the imaginary player that has been

put in by the user. This cluster is simply a smaller part of the dataset that is used at the start of the model. All the features are the same, however, the number of objects has been narrowed down to a group of players that play in the same position. This smaller dataset, containing the imaginary player, will be used in the next phase of the model.

5. *Grouping Players According to Performance and Physical Characteristics*
   In the last step, the model will process the remaining dataset to find a list of players with similar performance and physical characteristics. This process will output a list of players that can be used in the final step of the process of scouting new players.

   To obtain this list of players, the model will execute another clustering algorithm. The algorithm that will be used is the DBSCAN algorithm. This algorithm, mentioned in Section 3.4.3, is an algorithm that can be used on datasets of arbitrary shapes. It will fit well with the high-dimensional dataset that will be used, which consists of 24 distinct features that will be evaluated (Khan et al., 2014). These features are mentioned in Section 3.2. The difference with the k-means algorithm is that this algorithm does not need any prespecified numbers of clusters. This is ideal in this situation, since it is not known how many lists of similar players will be created. This ultimately depends on how similar the model wants the players to be. This will be explained in further detail.

   DBSCAN is an unsupervised algorithm, meaning it will train itself. The algorithm requires a minimum number of points p in a cluster centre's neighbourhood n. This p can be defined as the minimum number of players that the final list must contain. For example, when $p = 10$, the list will contain at least 10 different, but highly similar players. To determine a cluster centre, this centre must have at least p data points (thus players) in its neighbourhood n. This n is also known as the radius from a data point in which the algorithm will look for other data points. In this process, the algorithm will carefully analyse every point in the dataset. It will keep on going, until there is no more data point to be assigned to a cluster. In this process it is also possible for clusters to merge, eventually creating bigger clusters.

   The output of this algorithm is an unknown number of clusters, each containing players with a similar profile. In the previous steps, these players had already assigned their market values and their positions, the last step is now to choose the cluster that also contains the imaginary player created by the user.

6. *Output: The Final List of Players*
   Eventually, the model will output a list of j players that are like the player profile that is put in by the user of the model. This list contains players with similar physical characteristics, performance, and market value. When removing the imaginary player from this list, the user ends up with a list of players that contains a detailed statistical description of the player, Table 10 shows an example of how the output looks. This gives the user more insight in how much a player costs and what the player can do in the field. From there on, the user can select interesting players and do, for example, more in-depth research on the player. This selection can be done in different ways, for example by the user's budget or a minimum number of goals a player must have on its name.

Table 10: Example of how the model will output the list of players.

Name	**Appearances, Minutes Played, ..., Weight**	**Market Value (€)**
Player *1*	Player *1 Characteristics*	Player *1 MarketValue*
...	...	...
Player *j*	Player *j Characteristics*	Player *j MarketValue*

To give a more visual explanation of all the different parts of the process, Figure 2 shows a step-by-step visual representation of the described system.

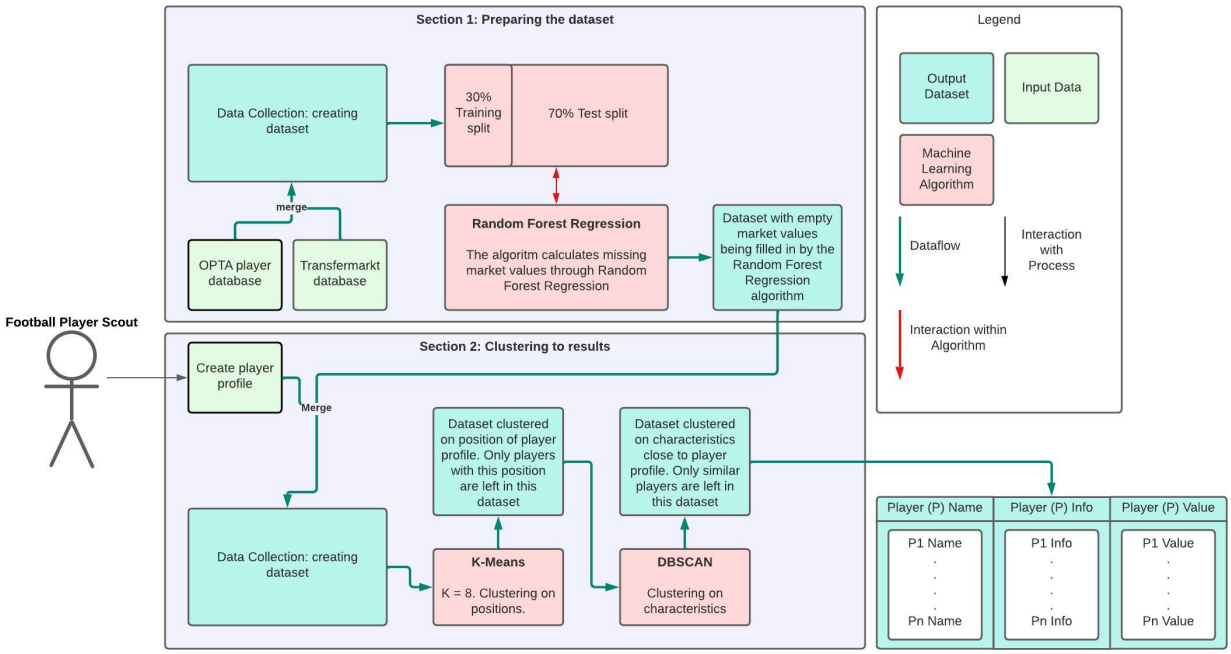

Figure 2: Visual representation of the system's process.

4.2 UML Use Case Diagram

Figure 3 shows a UML use case diagram of the system. The left actor is the primary actor, which is the football player scout. As told, this person will ultimately use the system. The secondary actors are placed on the right side. These, again, represent *OPTASports* (StatsPerform, n.d.) and *Transfermarkt* (*Transfermarkt*, n.d.). These actors will provide the necessary datasets. In this diagram, different actions of the system are presented within the oval figures. Each use case included in create complete dataset does not happen independently, meaning that if one use case happens, all the others will happen as well. The chronological order in which the use cases will happen is from top to bottom respectively

4.3 UML Sequence Diagram

Figure 4 shows a UML sequence diagram of the system. Following the use case diagram in Section 4.2, this sequence diagram includes all the use cases that are described in chronological order. This is because, when the system is used, all the use cases that are described happen in chronological order. Therefore, all use cases depend on each other.

The actors represent both parties that are included in the system. The first actor is the football player scout, who is ultimately going to use the system to scout players. The secondary actor includes, again, both the *OPTASports* and the *Transfermarkt* databases, who are going to provide the necessary player datasets. The red boxes represent the machine learning algorithms that are used by the system. The green boxes represent datasets or data objects that are created in the system by either the user or the system. The first step is to prepare the dataset that will be used by the system, this step is completed in the data collection part. From there on, the system will respectively call the algorithms to be executed on this dataset. The last step is to prepare the output, by making the data readable. The template for this output is discussed in step 6 of Section 4.1.

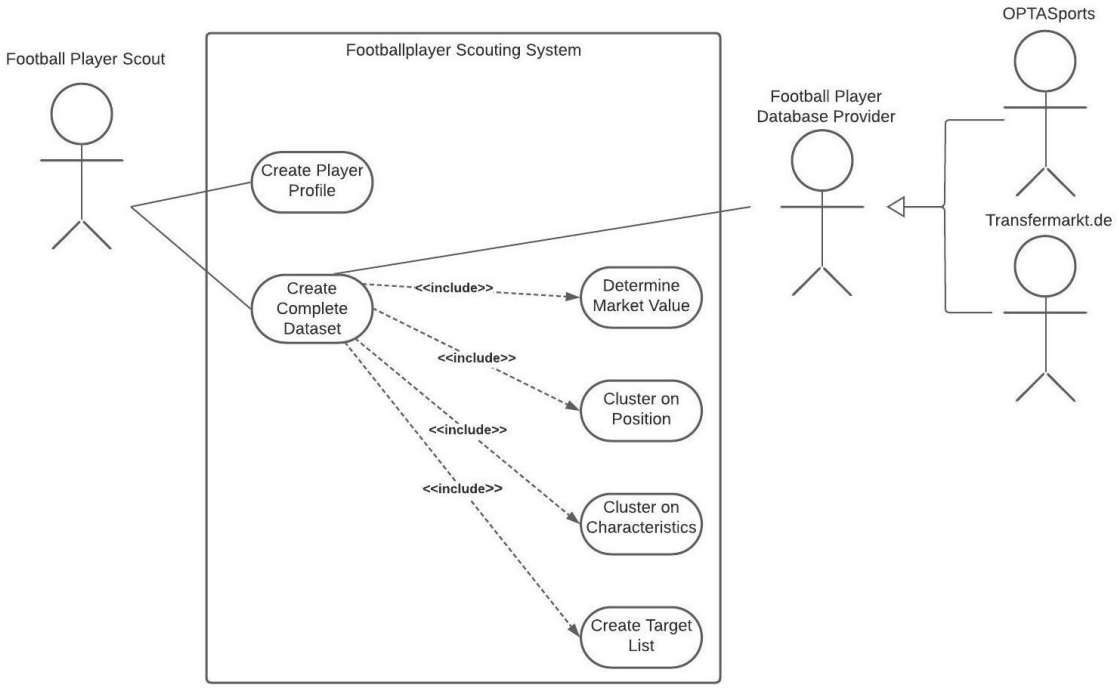

Figure 3: UML use case diagram of the system.

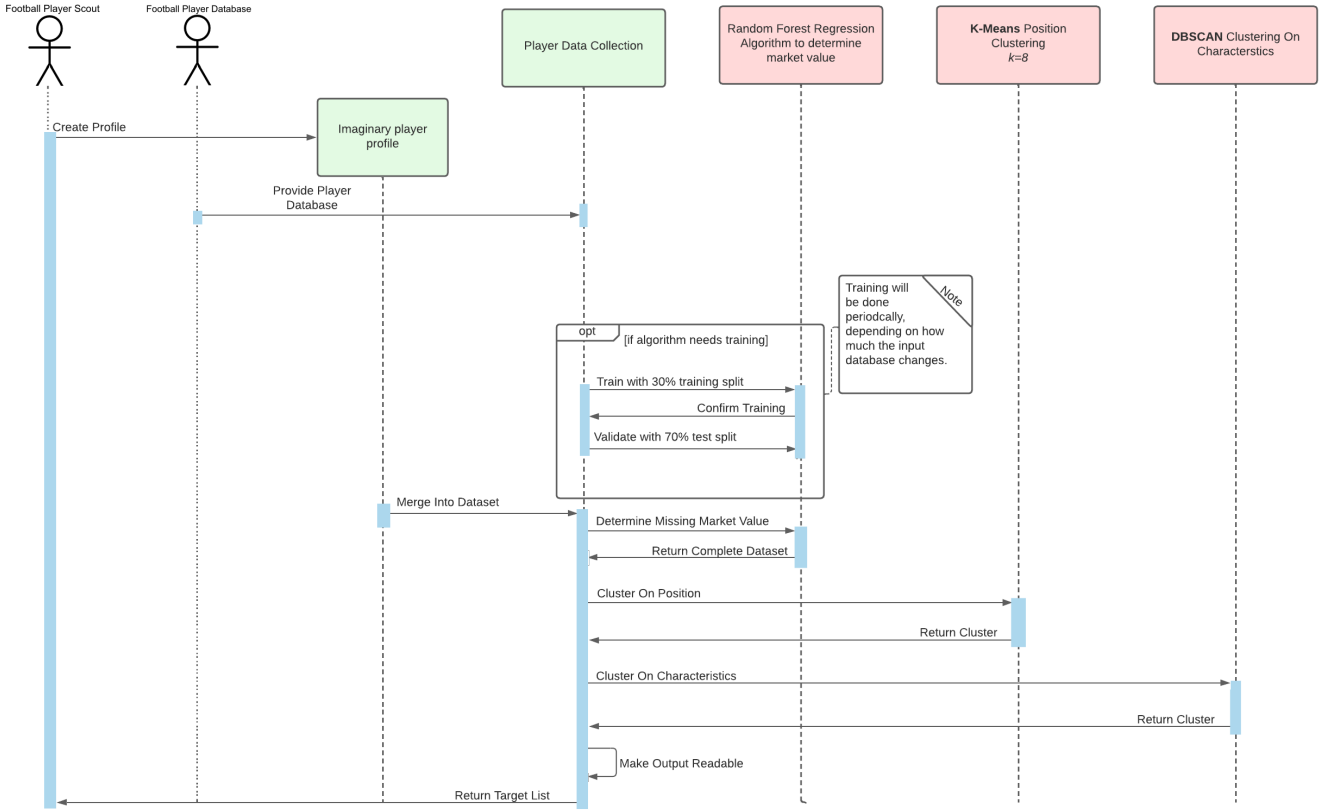

Figure 4: UML sequence diagram of the system.

5 Discussion

The system, summarized in the provided UML use case and UML sequence diagrams, is designed to help football clubs in their scouting process. It consists of two different phases, each phase containing several smaller phases. These phases have been defined by doing extensive literature research on machine learning algorithms, but also on player performance and player valuation. Each phase has been researched separately to find what the best fitting algorithms and features would be, to be implemented in the model. The model works on a dataset containing football players from all over the world, each with corresponding features mentioned in Section 3.2. Each phase has a different machine learning algorithm. By using random forest regression, k-means and DBSCAN in that order respectively, it is possible to find similar players according to a given player profile.

As stated, the model works mainly with the features that are mentioned in Section 3.2. These features describe player characteristics and their statistics. Those statistics can indicate the player's skills, but this is not guaranteed. In the research done by the authors of (Pappalardo et al., 2019), a framework is proposed that ranks players a certain level. Higher skilled players receive a higher level, thus delivering a ranking of players. This ranking could improve the model, as it compares players based on their skills. For example, the number of scored goals could tell if a player is very skilled, however, if the player plays in a lower division, it is likely to produce goals. This means that comparing players on goals and assists could possibly be misinforming. Thus, the model cannot tell the user on what level the player is, this must be decided by the user subjectively.

Goalkeepers are also not included in the model. This is because goalkeepers have different game rules than in-field players, (Berri, Rossi, Simmons, & Tordoff, 2023), (Pappalardo et al., 2019). Goalkeepers are usually not valued by goals and assists, but mainly on how good they are in protecting their goal. Since the in-field players are not valued for this, goalkeepers and in-field players should be looked at separately. To improve the model, in further research goalkeepers should be included as well. This would make the model more generally useful, and this means scouting for goalkeepers is included as well.

Defining player roles is also not included in the model. In Section 3.2, it has been explained that each player can adapt to a position differently. This means that a position is easily defined, however, this does not necessarily mean that each player with the same position has the same playing style.The authors of (Aalbers & Van Haaren, 2018) proposed an approach to distinguish players by their roles. It is uncertain if their approach would improve the model because the features they use in their approach are not defined. This could mean they make use of the same features as the model, but this is uncertain. If the approach would be implemented in the model, the features would have to be guessed. This gives too much uncertainty, which is not wanted. Next to that. it cannot be tested, for that a prototype is necessary.

The fact that it is only a model that is designed must be kept in mind. Because there is no prototype, it cannot be proven to work. With a prototype it will be possible to see the outcome of the model. This will also make it possible to try parameter tuning. These parameters include, e.g., the radius r of the DBSCAN algorithm that defines the neighbourhood, or the depth of the decision trees in the random forest regression. These parameters have not yet been given a value, for that a prototype is needed.

6 Conclusion

In the process of building a football team, a club uses scouts to find players. When the club finds a player that it wants, it needs to negotiate with the club that currently owns the player. This is a lengthy process and requires multiple individuals. This research proposes a model to support this process, or even replace it. By implementing multiple machine learning algorithms, this goal can be acquired. The whole process is defined in a model.

The aim of the model is to find the desired players by using machine learning. By combining several machine learning techniques, it is possible to make a complete algorithm that will do this for the user. This will help the user with financial decision making in the process of buying players. The main target audience of this model is football clubs, and obviously these clubs would still have to execute the transfer process themselves, however, the scouting procedure that comes beforehand can supported by, or even be replaced by the model. This depends on the club's willingness to use it, but it will allow the club to replace physical scouts. This means that they have less employees, thus fewer expenses on wages.

The research is done by individually researching football- and football player data, machine learning algorithms and how they can be combined to create the model. This means that the model relies on research that is already done by others. However, while doing the research, it became clear that the application of machine learning on football is still limited. It is a hot topic, however, many articles found on this subject are only recently published. If more research could be considered, the quality of the model would be more guaranteed.

This research has shown how machine learning can be implemented to support the financial decision making on player transfers in football. The outcome is a proof of. concept that can be used as guideline to build the prototype. For further research it is suggested to build a prototype, so that the model can be proven. With a prototype, a better understanding of the algorithm can be acquired. In further research, the model can also be improved by implementing the limitations that have been discussed in the discussion.

References

Aalbers, B., & Van Haaren, J. (2018). Distinguishing between roles of football players in play-by-play match event data. In *Proceedings of the 5th Workshop on Machine Learning and Data Mining for Sports Analytics* (p. 31-41).

Al-Asadi, M. A. M., & Taşdemir, (2022). Predict the value of football players using FIFA video game data and machine learning techniques. *IEEE Access*, *10*.

Asadi, S., & Shahrabi, J. (2016). RipMC: RIPPER for multiclass classification. *Neurocomputing*, *191*, 19–33.

Behravan, I., & Razavi, S. M. A. (2021). A novel machine learning method for estimating football players' value in the transfer market. *Soft Computing*, *3*(25), 2499–2511.

Berri, Rossi, G., Simmons, R., & Tordoff, C. (2023). Salary determination in professional football: empirical evidence from goalkeepers. *European Sport Management Quarterly*, 1–17.

Bialkowski, A., Lucey, P., Carr, P. W., Matthews, I., Sridharan, S., & Fookes, C. (2016). Discovering team structures in soccer from spatiotemporal data. *IEEE Transactions on Knowledge and Data Engineering*, *28*(10), 2596-2605.

Cintia, P., Rinzivillo, S., & Pappalardo, L. (2015, 09). A network-based approach to evaluate the performance of football teams.

García-Aliaga, A., Marquina, M., López, J. C., Rodríguez-González, A., & Luengo-Sanchez, S. (2020). In-game behaviour analysis of football players using machine learning techniques based on player statistics. *International Journal of Sports Science Coaching*, *16*, 148–157.

Gold, C. A., & Sollich, P. (2003). Model selection for support vector machine classification. *Neurocomputing*, *55*, 221–249.

Gudmundsson, J., & Horton, M. A. (2017). Spatio-temporal analysis of team sports. *ACM Computing Surveys*, *50*, 1–34.

Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014). DBSCAN: Past, present and future. *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 232–238.

Kotsiantis, S. (2011). Decision trees: A recent overview. *Artificial Intelligence Review*, *39*, 261–283.

Krämer, O. (2013). K-nearest neighbors. *Intelligent Systems Reference Library*, *51*, 13–23.

Li, Y., Zou, C., Berecibar, M., Nanini-Maury, E., Chan, J. C., Van Den Bossche, P., ... Omar, N. (2018). Random forest regression for online capacity estimation of lithium-ion batteries. *Applied Energy*, *232*, 197–210.

Mahesh, B. (2020). Machine learning algorithms — A review. *International Journal of Science and Research (IJSR)*, *9*, 381–386.

Müller, O. J., Simons, A., & Weinmann, M. (2017). Beyond crowd judgments: Data-driven estimation of market value in association football. *European Journal of Operational Research*, *263*, 611–624.

Na, S., Xumin, L., & Yong, G. (2010). Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics* (pp. 63–67).

Pappalardo, L., Cintia, P., Ferragina, P., Massucco, E., Pedreschi, D., & F., G. (2019). Playerank: Data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology*, *10*, 1–27.

Rong, S., & Bao-Wen, Z. (2018). The research of regression model in machine learning field. *MATEC Web of Conferences*, *176*, 1–33.

StatsPerform. (n.d.). *OPTA data from Stats Perform.* Retrieved from www.statsperform.com. (Accessed on 11-12-2023.)

Suthaharan, S. (2016). Support vector machine. *Integrated Series on Information Systems*, *36*, 207–235.

Ter Braak, C. J. F., & Looman, C. W. N. (1995). Regression. *Data Analysis in Community and Landscape Ecology*, 29–77.

Toma, P., & Campobasso, F. (2023). Using data analytics to capture the strategic and financial decision-making of europe's top football club. *Technological Forecasting and Social Change*, *186*.

Transfermarkt. (n.d.). Retrieved from www.transfermarkt.de. (Accessed on 11-12-2023.)

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Zhang, C., Liu, C., Zhang, X., & Almpanidis, G. (2017). An up-to-date comparison of state-of-the-art classification algorithms. *Expert Systems with Applications*, *82*, 128–150.