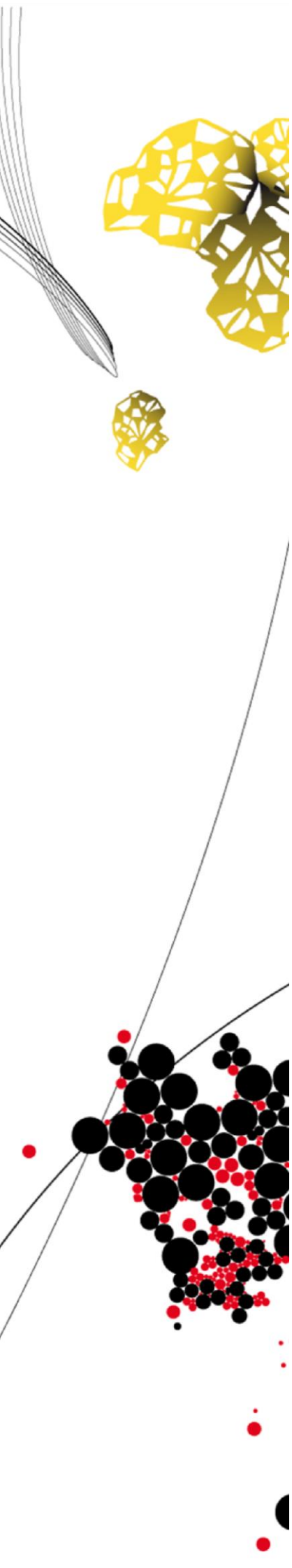

UNIVERSITY OF TWENTE.

Faculty of Behavioral, Management & Social Sciences

An elaborate approach to game winning strategies and player ratings in football

Casper Ritmeester

M.Sc. Thesis July 2018

Supervisors:
dr. R.A.M.G. Joosten
Dr. B. Roorda
M.J. Fledderus

Financial Engineering and Management
Faculty of Behavioral,
Management and Social Sciences
University of Twente
7500 AE Enschede
The Netherlands

Abstract

In an industry where the gap between the rich and the poor is growing apart to a ridiculous extent, outsmarting the big spender has never been more crucial. Making more informed decisions regarding the buying of football players is becoming more and more important. Therefore, player rating tools can be useful for football teams to support these decisions. In this master thesis we design a new method for rating football players. We call this the Exact Player Rating (EPR). This method will not only estimate a player's skill level, but will also incorporate the most important game winning aspects of football and link this to the players at hand. We have shown that the EPR method does not only show better results in terms of predictive power, but also provides more useful information and does not overvalue offensive skills, which most current player rating methods do. Because theses are made publicly available, we do not include privacy-sensitive data.

Keywords: football, player ratings, statistical modeling, game winning strategies, hierarchical Bayes, player valuation

Contents

Chapter 1: Introduction

In the football industry, each team has a different budget to spend. Due to recent developments in football, the budget gaps between the poor teams and rich teams have never been bigger. When looking at the budget of Real Madrid and Las Palmas, two teams in the same league, we see that this budget gap can easily be over a few hundred million euro's [1]. Many teams have stated that there has never been a more unequal playing field, and competing against the big spender seems impossible. In order for "David" to have a fighting chance against "Goliath", "David" has to make smart decisions regarding their small budget. One of these decisions is which players to buy. Statistical methods can be used by football teams to support these decisions, since plenty of necessary data are recorded during a football match and made available for usage. However, not many teams are using statistics and statistical models as a base for their decisions.

There are however a few examples showing us the need for statistical methods in sports. One of those examples is well documented in Michael Lewis' book Moneyball [2]. This book tells the story about a baseball team in the late 90's , the Oakland Athletics, which started to incorporate statistical methods in order to improve their decision making regarding player transactions and game strategies. They found out that certain player attributes were highly overvalued, such as the ability to hit homeruns and that other attributes were highly undervalued, such as the ability to steal bases. They found that it is better to concentrate on getting bases and hitting the ball at higher percentages, than hitting the ball at lower percentages and hope for a homerun. By adjusting their game plans and trading players in line with their game plans, they unexpectedly became a much better team and even outperformed many teams which had more money to spend. After the book Moneyball was published and other teams in U.S. baseball understood what Oakland Athletics was doing, their competitive edge was gone and the team went back to the lower end of the rankings. Nonetheless, this is a good example of how statistical methods can be beneficial for a sports team and can impact a team's performance.

Football and baseball are two completely different sports and they cannot be compared. Football is way more dynamic, there are fewer breaks and the strategies are completely different. However, the use of statistical methods can still be beneficial for football teams if used in a correct way. This demonstrated Danish football club FC Midtjylland, going from almost bankrupt, to winning their first Superliga title within just a couple of years due to the smart use of statistical models [3].

We take a look at a Dutch football team, Heracles, in the Dutch Eredivisie. We discuss some of the existing statistical methods Heracles can use to support their decision making regarding player transactions and game winning strategies. We also discuss the limitations of these statistical methods and propose improvements.

1.1 Literature review

The use of statistical models in football is not new in the industry. A lot of statistical models have been made in order to beat the bookmakers. An example is the model made by G. Baio et al. [4], using a Bayesian hierarchical model for the prediction of football results. However, player rating models are

a relatively recent development in the industry and there are only a few player rating models on the market. We discuss the player rating models available on the market, some that are absent when compared to other sports, and we conclude this literature review with a short summary of what we believe are some of the shortcomings that we could improve upon.

1.1.1 Player Rating Models

Individual player statistics are used extensively in player rating methods. They are widely available and easy to interpret. Individual player statistics are kept for every football match by data collectors. They record for each player the minutes played, goals scored, scoring percentage, assists, chances created, interceptions, tackles, fouls and percentage of duels won by the player for example. These statistics are then made widely available on the internet by companies such as Opta, Whoscored, Wyscout, Soccerlab and Squawka.

These statistics are quite easy to interpret, as they represent the direct output of a player on a match. Think in terms of goals, assist, interceptions, tackles, duels won, fouls etc. It is however hard to interpret how much effect some of these statistics have on the outcome of a match. Take for example a tackle or an interception of the ball. Making tackles and interceptions is important, as it will rob the opponent from an opportunity to score. However, they do not always lead to goals scored by the team making the interception or a tackle.

So what is the effect of making a tackle, interception or any of the other individual player statistics? We discuss several statistics that use regression analysis to measure the effect of individual player statistics on the outcome of a match.

Player Efficiency Rating Model

ESPN writer John Hollinger has invented The Player Efficiency Rating (PER) [5]. The Player Efficiency Rating is widely used in the basketball scene. The PER stat measures the per-minute performance a player has on average and can be used to compare player performances across seasons. The Player Efficiency Rating models a player contribution to a game and even adjusts them for the pace of each team. This accounts for the fact that teams that have more possession have more opportunities to score. Football is however a different sport and it is hard to find ways to translate tackles and interceptions to goals, since so few goals are scored compared to basketball and the assessment of off-ball movements are also difficult. It will thus take more data and breakthroughs to assign goal-numbers to actions and determine whether a person is 'efficient' or not. Data collectors Whoscored and Squawka have developed their own Player Efficiency Rating systems, where aspects of the game (goals,assist,passes, duels, fouls) are taken into account and translated into a number per player [6]. The accuracy of this number however, has never been determined.

Wins Produced Model

The Wins Produced statistic measures the wins a player produces and was invented by sports economist David Berri [7] [8]. The Wins Produced model first estimates the effect of statistics on two measures of attack and defense with regression analysis [9]. Then an individual player's contribution to his teams Offensive- and Defensive Efficiency can be measured by looking at his statistic. This model estimates an individual player's contribution to a win and is again highly implemented in basketball, but not much yet in football. The wins produced method can be found for world class play-

ers like Ronaldo and Messi and we thus conclude that this method is applicable for football. However, it is not being used for everyday players yet.

Player contribution per position

It is generally accepted that players in different positions have different roles and it thus makes sense that certain skills are more valued for some positions than others [10]. For example, dribbling is not as desirable for a center back as it is for an attacker. Losing the ball due to insufficient dribbling will usually lead to a big opportunity for the opponent to score, since the center back is usually the last line of defense. Dribbling for the attacker however is a desirable attribute, since it creates opportunities to score and losing the ball in the front line can still be recovered by his teammates. However, there is no research to determine which attributes are important for each positions and why in football. Page et al. [11] made a start by using a Bayesian hierarchical model in basketball to estimate how statistics affect the match outcome as measured in point differentials. Page et al. [11] found for example that making steals is more valuable for centers than for other positions, since their position requires them to be close to the basket. Football looks to have fallen behind in this department and although Page et al. [11] did not use their results to rate players, a similar research can be easily extended to do just that.

Adjusted Plus-Minus

Dan Rosenbaum is the first person that came up with the Adjusted Plus-Minus statistic [12] and it is based on the Plus-Minus statistic. The Plus-Minus rating is a relatively simple concept. This model identifies a player's implied effect on his team's score difference while he is on the field [12]. The Adjusted Plus-Minus model attempts to establish this contribution while accounting for opponents and teammates on the field [12]. A player's effect on his team's score differential will thus change as his teammates or opponents change during the match. If we take a look at a large number of scenarios, it should possible to measure how each player contributes to the game. It follows that if we know the player's contributions to a game, we can predict the expected margin of victory and thus the game outcome. The Adjusted Plus-Minus is thus not only a descriptive model, but a predictive one as well.

What makes the Adjusted Plus-Minus so attractive in theory is that the data inputs are relatively easy and already available on the web for usage. We only need to know the player line-ups, the substitutions, the expulsion records in combination with the times they occurred and the score In practice however, this is not as easy for football.

Howard Hamilton did research on the Adjusted Plus-Minus in football [13]. He found out that the value of Adjusted Plus-Minus in sports like basketball and ice hockey is substantial, since there are a lot of segments in both sports, which makes it easier to identify the impact of players [13]. The metric showed the top players everyone would expect- LeBron James, Dwight Howard, Sidney Crosby, Pavel Datsyuk, etc. In football however, there are fewer segments to measure. This means that there are fewer opportunities to identify the impact of players. Hamilton found out that the out-of-sample prediction for the football Adjusted Plus-Minus had a variance, R^2, of 0,03 [13]. This means that 3% of the variance in the goal difference data can be explained by the model. This is clearly not sufficient and we thus conclude that the Adjusted Plus-Minus is not a suitable predictor yet. It could become a useful metric in the future, but Hamilton states that this metric still requires a lot of care in its formulation, implementation, and interpretation [13].

Regularized Adjusted Plus-Minus

The next step from the Adjusted Plus-Minus is to try and reduce the errors by moving from a standard linear regression to a ridge regression [13]. Ridge regression can be seen as an extension of linear regression and the idea is that ridge regression helps minimize the errors associated with the player's plus-minus scores [13]. Howard Hamilton showed nonetheless that this model has still too much error to be useful.

Subspace Prior Regression

The Subspace Prior Regression (SPR) statistic made by Dapo Omidiran is arguably the most accurate statistical model in basketball [14]. The SPR statistic can largely be seen as an extension of Dan Rosenbaum's Adjusted Plus-Minus statistic [12]. D. Omridan's [14] criticism was that the Adjusted Plus-Minus statistic did not account enough for the skill disparity between players. He based his model on the NBA and noted that the NBA is a competition driven largely by star players. Better players contribute far more to the success than lesser players. He therefore penalizes large player ratings in order to create more model sparsity [14]. Furthermore, he adds another penalty term in his regression model where he penalizes the distance between a player's rating and some of his score outputs [14]. This penalty term is included, because a player's skill level should be reflected in his statistics [14].

D. Omridan [14] found that these two additions to the model increased the predictive performance of the Adjusted Plus-Minus model and is therefore a good model extension. However, since H. Hamilton [13] found out that the Adjusted Plus-Minus model is not a suitable predictor for football and D. Omidiran's [14] model can be seen as an extension of the Adjusted Plus-Minus model, it is not likely that the Subspace Prior Regression statistic is the answer for rating football players.

1.1.2 Upcoming companies

The most important player rating models have been discussed and there are already two big upcoming companies in the Netherlands that translate data into statistical models in order to give football clubs advice regarding buying players. We take a look at these companies in order to get a proper understanding of what is already being offered and used on the market. Furthermore, the football club FC Midtjylland has been very successful with the use of statistics. We also take a look at FC Midtjylland in order to get an understanding of how they have been so successful. Lastly, we will discuss the possible shortcomings of these models in Section 1.1.4.

Scisports

The first big upcoming company is Scisports, based in Enschede, the Netherlands. Scisports gives football clubs consultation regarding who to buy and also gives football players guidance in order for them to understand which team suits them best [15]. Usually, a football club comes to Scisports with the question which player to buy. Scisports will then take different criteria into consideration, such as salary, maximum transfer fee, and other technical criteria that the football club hands them. These criteria are the starting points for their advice [15].

In order for Scisports to give proper consultation, Scisports builds algorithms to present numerous scores [15]:

- The SciSkill score: The SciSkill score is composed of a combination of a player's offensive, defensive and resistance (dependent on the league) factors.
- The potential SciSkill score: The potential SciSkill score is an estimation of a player's potential based on the expected maximum Sciskill of a player at the age of 28, since it is unlikely that a player's skills increases after that age.
- SciSkill growth: The SciSkill growth is the increase in a player's SciSkill score over the past six months.
- The P-score (percentile score): Lastly, the P-score compares the current SciSkill score with the value of players in their age group who are six months older or younger than the player concerned. This means that is a player has a percentile score of 98 (out of 100), he is in the top 2% players in his age group with the highest SciSkillscore.

A simple example where players are being compared based on these scores can be found below.




	TUKO OHIYAMA	ZOJAN KORALICIC	IRWIN DENNINGHAM	CELVINA POZUELA
AGE	29	27	22	19
CURRENT CLUB	Keypass Athletic	FC Algebra (last club)	AC Stats	Maps United
LEAGUE	First Goal Division	Secunda Assist Division	First Goal Division	First Goal Division
COMPETITION STRENGTH SCISKILL	71,8	59,3	71,8	71,8
TOTAL MATCHES (All competitions season 2016-2017)	13	5	11	17
TOTAL MINUTENS (All competitions season 2016-2017)	903	294	781	1385
GOALS (All competitions season 2016-2017)	0	0	0	0
ASSISTS (All competitions season 2016-2017)	3	1	0	1
ASSISTS (Per 90 minutes)	0,3	0,3	0,0	0,1
% PLAYED	48%	12%	31%	73%
CURRENT SCISKILL	57,7	55,5	70,3	29,6
DEVELOPMENT SCISKILL (Difference SciSkill past half year)	-3	-2	+1,9	+11,2
POTENTIAL SCORE	62	64	106,8	67,4
P-SCORE	87	85,3	98,3	80,8

Figure 1.1: The comparison of SciSports [15].

Finally, when the football club has made a choice between the different players, SciSports will perform an extensive background check of the selected player or players. They will examine a player's roots, his club history, media profile, management, social media activities as well as an extensive data analysis of his performances on the pitch and his development throughout his career [15].

Remiqz

The second big upcoming company in the Netherlands is Remiqz. Remiqz is a predictive intelligence service for football clubs [16]. They use data from all games over the last decade, which they use to simulate the rankings of football clubs, the current and future added value of all players and corresponding team performance [16]. It supports professional clubs in scouting, coaching and general management by making profound decisions in their transfer policy [16].

Remiqz's starting point is the EuroClubIndex (ECI) [16]. The ECI is a ranking of football teams in the highest division of all European countries that show their relative playing strengths at a given point in time. The ECI also accounts for the development of playing strengths in time. The ECI makes it possible to calculate the probabilities of different match results (win, draw, loss) for football matches in the near future [17]. Remiqz states: "The EuroClubIndex (ECI) is the only objective ranking of clubs that shows an accuracy of 97.3% in its predictions " [18]. Based on the ECI, Remiqz can predict the outcome of the competition at the end of the season. If they see a football club coming short on their target for the season, Remiqz can show incentive to change and help the clubs to improve.

FC Midtjylland

One of the biggest stories in the rise of data analysis in football since 2013, and maybe so far, has to be FC Midtjylland. The club went from almost being bankrupt to winning the league in the highest Danish division in just a few years [3]. Club owner Rasmus Ankersen wanted to experiment and test the thesis that you can successfully run a football club based on statistical analysis of the game [19]. He wanted to stamp out subjective and emotional decision making and replace it with a scientific method. In doing so, smaller clubs like FC Midtjylland can get a competitive edge over the bigger and richer opponents. "We can't outspend our competitors, so we have to out think them" [19]. Rasmus Ankersen feels that careful analysis of data on leagues, teams, and players can give them an edge.

So how does FC Midtjylland do it? An important part of their success is player recruitment. Ankersen uses a model that ranks all clubs in Europe as if they are playing together in one big league [19]. For example, Greuther Fürth, a small German team playing in the second division of the German league, did not play against the clubs in the Premier League, but they did play HSV of Hamburg, which in turn played Bayer Munich, which, in the Champions' League, played Manchester United, which played the rest of the Premier League. Taking this into account, the model can cross-reference results from different leagues and use advanced statistical tools to rank every club on the continent. This allows Midtjylland to see through the aura that the Primera Division or Premier League project onto the clubs that play in them. "People see huge difference between the Premier League and the lower divisions in England," Ankersen explains. "We think this is not true. There is a big gap between the Premier League's number 7 and number 10. But the gap between the Premier League's number 10 and the Championship, or even League One, is far smaller" [19]. It also allows them to see the true value of clubs playing in less fashionable leagues. So when Greuther Fürth appeared surprisingly high on the ranking of all European football clubs, Midtjylland took an interest in the players who had the most appearances for the club, and were thus most responsible for Fürth playing like a Premier League team. Top of that list: Tim Sparv. This singing turned out to be very successful.

In order to rank every club on the continent, FC Midtjylland uses the expected goals method [20]. Ramus Ankersen feels like the number of goals scored is a poor reflection on a teams quality and thus a poor predictor of the results. He felt that analysts need data that strip out randomness and

luck and can predict future performance more accurately. "It has happened thousand of times before that your team dominates a game, is unable to score, whereas the opponent then pings one the last minute and you find yourself losing" [20]. Ankersen proposed a metric to calculate a conversion rate for a particular shot [20]. It is then possible to calculate the expected number of goals scored and the expected number of goals conceded. Subtracting the expected number of goals scored by the expected number of goals conceded you get the net expected goals. This gives the analyst a good indication of a team's quality and thus making it possible to create a ranking system of every club on the continent, where the team with the higher average net expected goals is the favourite to win.

1.1.3 Data collectors

We have discussed several player rating models, upcoming companies that create player rating models, and FC Midtjylland, where FC Midtjylland uses a team rating model in order for them to understand which players at which club can make a difference for them. In order for the companies and FC Midtjylland to make an accurate statistical model, a profound understanding of data is needed. Luckily for them, there are so called data collectors who keep track of data and update the most important data each week in order for football clubs to improve for example:

- Player development and performance.
- Communication with players, staff, parents.
- Internal organization.
- Results.

The data collectors sell their data to companies like Scisports and Remiqz, but also straight to football clubs like FC Midtjylland. Some of the biggest data-collectors in football are Opta, Wyscout, Soccerlab and Squawka.

1.1.4 Summary

Now that we have discussed the player rating models on the market, we are able to make an overview. When looking at these player rating models, we can see that statistical models in football can already be very beneficial. The models on the market today can already give an accurate score of how good a player is today and how good he can be in the future. FC Midtjylland created their own statistical model and it seems to be working for them. However, there are still some shortcomings to the current player rating methods.

There are not many player rating models on the market compared to other sports and the current models that are on the market are not able to make a link between a player's skill level and the type of player that clubs are looking to sign. If a team wants to acquire a new player, not only is it important to know how good that player is, but also what the effect is of signing that particular player. Data are available to determine whether the player is a good passer, shooter, dribbler, defender, etc., but this still does not say what a player's effective contribution to a team is. For example, a player that scores many goals would probably look like a good addition to most teams. However, if signing this player means that the offensive efficiency of his team-mates goes down, it might not be a good decision to sign that player.

Steven Houston, Head Analyst at Hamburger SV [21] was spot on: "It is no longer a case of saying a player has scored X goals or a midfielder has created X assists. You only have to look at something

simple like a goal. There are so many types of goals, the difficulty of the goal, the quality of the goal. And with passes there are passes and then passes in the final third. The hardest thing is to work out what is important and what isn't important, at a team level but also for individual players."

The companies Scisports and Remiqz both offer extensive advice in the quality of players, but these companies are not able to determine the effective contribution to a team. Therefore, their advice is incomplete.

Another point that the current statistical methods lack, is that they are unable to measure what kind of strategies and attributes are effective to win games. The model of FC Midtjylland made a start with their net expected goals method, but this is too simple. It has happened far too often that the team with the most goals scored and least amount of goals conceded is not crowned champion. It even happened in the year 2017-2018 in the Eredivisie, where Ajax, the team with the most goals and least amount of goals conceded, came in second. Of course goals scored and goals conceded are important, but it is more important for teams to know which attributes per position are important in order for teams to properly understand and determine which players can make an impact on their team and who they should buy. This has never been done before and determining which attributes per position are important will give football clubs also a better understanding of how players will fit into a team. Page et al. [11] made a start with their model that estimates how important the output of several statistics for each player position is. Although it was done for basketball and also did not measure the effect of certain attributes per position, it can be extended to just that for football.

1.2 Thesis contribution

The main goal of this thesis is to find an approach to correctly estimate how skillfull a player really is and what strategies are most important for winning. We are then able to simulate the effect of signing a particular player for Heracles better. Compared to the existing methods, which are only able to estimate a player's skill level, we believe this is a substantial improvement. This will be done in a 2-stage regression model. The first stage models the influence of players on several production statistics.

In the second stage we model score differentials with the estimated production statistics from the first stage as explanatory variables. With the results from the second stage we can see which production statistics affect score differentials most and are most important for winning. Because we estimate the influence of the players on score differentials indirectly through the first stage, we can see which players have the largest effects on score differentials, so which players are the best and the worst. We call this the Exact Player Rating.

We validate the model by comparing the forecast accuracy of our EPR method with the Bayesian hierarchical model of G. Baio et al. [4]. We would have liked to compare the accuracy of our EPR with the best player rating model in the literature, but this is impossible, since no algorithms are available for usage.

What this thesis contributes to existing methods, is that current methods estimate player's skill levels by simply regressing score differentials on a player's presence on the field. In other words, the current methods are only able to estimate player's skill levels. Our approach regarding the EPR is very different, because we use a two-stage model that estimates the relation between score-differentials

and the presence of players through production statistics. This will provide a lot of extra useful information about game winning strategies and players' strengths and weaknesses.

Lastly, in this thesis we propose a method that tries to determine a player's market value by their statistical skill level in order to determine which players should be bought. This has also not been done before.

Chapter 2: Data

In this section we give a short description of the data we used to estimate player ratings. We used data from the season 2017-2018 for ten different competitions in order to determine game winning attributes and thus the game winning strategies. The competitions are the Dutch Eredivisie, the French Ligue 1, the English Premier League, the German Bundesliga, the Italian Serie A, the Russian Premier League, the Spanish Primera Division, the Portuguese Primeira Liga, the Turkish Süper Lig and the Danish Superliga.

This may sound peculiar, since the competitions are different and comparing the results is difficult, which is a valid point when it comes to comparing team strength. The champion in the Danish Superliga with 90 points in total is not likely to be better than the English Premier League champion, who "only" obtained 88 points, since the English Premier League is considered a much more challenging competition. However, in determining the game winning attributes and strategies, the data of different competitions can be compared. The champion playing in a simpler competition, who has more points compared to the champions playing in a more difficult competition, is also likely to dominate more than the champion in the more difficult competitions. They will score more goals, create more chances, win more duels and since this is the case, the variables are thus comparable and useful in determining game winning strategies. The data used can be found on https://platform.wyscout.com/app/?.

Continuing with what is included in the dataset, we use all data available on the website in order to determine which attributes per position are most important. We first create a matrix with data per team, per player, per game. We are then able to summarize the total statistics of these players for an entire season. We elaborate more on this in Chapter 3. Furthermore, we make a distinction between the goalkeeper and field players, since the goalkeeper has completely different attributes compared to field players.

The variables included in the dataset for the field players are given in Table 2.1.

Field players			
Position	Age	Matches played	Market value
Minutes played	Goals	Assists	Expected goals
Expected assists	Birth country	Foot	Height
Weight	On loan	Succesful def duels per 90 min	Def duels per 90 min
Def duels won %	Aerial duels per 90 min	Aerial duels won %	Tackles per 90 min
Tackles won %	Shots blocked per 90 min	Interceptions per 90 min	Fouls per 90 min
Yellow cards	Yellow cards per 90 min	Red cards	Red cards per 90 min
Succesful attacking actions	Goals per 90 min	Non penalty goals	Non penatly goals per 90 min
Expected goals per 90 min	Head goals	Head goals per 90 min	Total shots taken
Shots per 90 min	Shots on target %	Goal conversion %	Assist per 90 min
Crosses per 90 min	Crosses from left per 90 min	Crosses from right per 90 min	Crosses accuracy %
Dribbles per 90 min	Dribbles succes %	Touches in box per 90 min	Passes per 90 min
Passes accuracy %	Forward passes per 90 min	Forward passes accuracy %	Back passes per 90 min
Back passes accuracy %	Short/middle passes per 90 min	Short/middle passes accuracy %	Avg long pass length
Expected assists per 90 min	Second assists per 90 min	Third assists per 90 min	Smart passes per 90 min
Smart passes accuracy %	Final 3rd passes per 90 min	Final 3rd passes accuracy %	Long passes per 90 min
Long passes accuracy %	Through passes per 90 min	Through passes accuracy %	Average pass length
Passes to penalty area per 90 min	Passes to penalty area accuracy %	Direct free kicks on target %	Penalties taken
Penalties conversion %	Rating		

Table 2.1: Variables field players.

The variables included in the dataset for the goalkeepers are given in Table 2.2.

Goalkeepers			
Age	Market value	Minutes played	Birth country
Height	Weight	On loan	Matches played
Goals per 90 min conceded	Clean sheets	Save %	Expected goals conceded
Exits per 90 min	Punches per 90 min	Punches %	Claims per 90 min
Aerial duels per 90 min	Aerial duels won %	Claim/punch ratio	Foot
Total goals conceded	Expected goals conceded per 90 min	Rating	

Table 2.2: Variables goalkeepers.

The Rating variable is the total points of their team.

For each event (match) there is information which players were on the field and at which position they were playing. These positions are:

- Striker: The striker usually requires good shooting abilities and/or heading abilities in order to score.
- Winger: The winger can be a left- or a rightwinger. The winger position usually requires good passing, dribbling and shooting abilities.
- Midfielder: A midfielder can be attacking, defending or balanced. An attacking midfielder is the playmaker and usually has good passing and dribbling abilities combined with excellent vision. A defensive midfielder is required to rob the opponent of the ball and therefore requires good tackling and interceptions abilities. The central midfielder is normally an all-rounder and requires both defensive and attacking abilities.
- Left- and rightback: The left- and rightback are arguably the most discussed positions on the field. Some say the left- and rightback just have to be good defensively and thus have to have good heading, strength and tackling abilities. Others say left- and rightbacks have to be dynamic and join the attacking line, making passing and dribbling also important abilities for the left- and rightback.
- Center back: The center backs are usually one of the tallest players on the team. They require good defensive abilities like heading, interceptions, tackling and strength.
- Goalkeeper: The last position is the goalkeeper. The goalkeeper has totally different attributes then the other positions and height, reflexes and positioning are arguably the most demanded attributes in looking for a goalkeeper.

Once the key attributes per position have been determined, we are able to link the weights per attribute per position in order to form the current EPR of players. The EPR will then be linked to the current market value of players from the Eredivisie, Jupiler League, Ligue 2, second- and third division of the Bundesliga, since these are the competitions that have potential players for Heracles.

Lastly, we also account for the fact that most home teams have an advantage as opposed to the away team. This data can be found on https://footystats.org/netherlands/eredivisie/detailed-stats/home-advantage-table. The mean of the home advantage is positive and a positive sign can easily be explained by the fact that home teams will have a motivational advantage by playing in front of their home crowd. Another point is that some teams play on artificial grass instead of real grass, making the bounces of the ball and passes different. Playing on artificial grass on a daily bases gives the team a better understanding of how the ball is going to behave differently. Heracles and PEC both have a huge home advantage, which can be explained by the fact that these teams play and train on artificial grass on a daily basis.

Chapter 3: Methods

In this section we formulate various models and the estimation procedures of these models. In Section 3.1 we formulate the famous Bayesian hierarchical model made by G. Baio et al. [4], which will be used as a benchmark model. In Section 3.2 we improve upon this method and formulate our Exact Player Rating (EPR). In Section 3.3 we explain how we validate and compare the different methods in terms of forecasting power.

3.1 Bayesian hierarchical model

Since we were unable to obtain the player rating methods of Scisports or Remiqz and no other player rating models are available for usage, we were forced to look at a method that is widely available. In this section, we formulate the Bayesian hierarchical model made by G. Baio et al. [4]. Even though this is a team rating model and not a player rating model, this model provided excellent outcomes in the prediction of football matches and these results can be compared with our EPR method.

G. Baio et al. [4] proposed two models in their paper. The first one is the base Bayesian hierarchical model and the second model is adjusted to overcome the issue of over shrinkage, which will be explained in Section 3.1.2. The second model specified a more complex mixture that results in better fit to the observed data. We first discuss the base Bayesian hierarchical model.

3.1.1 Base Bayesian hierarchical model

In the first model, the observed goals scored counts as independent Poisson:

$$y_{gj} \mid \theta_{gj} = Poisson(\theta_{gj})$$

"where the parameters θ = (θ_{g1}, θ_{g2}) represent the scoring intensity in the gth game for the team playing at home (j = 1) and away (j = 2), respectively" [4]. These parameters are modelled assuming a log-linear random effect model [4]:

$$log\theta_{g1} = home + att_{h(g)} + def_{a(g)}$$
$$log\theta_{g2} = att_{a(g)} + def_{h(g)}$$

Note that they are breaking out the total team strength into attacking and defending strength. A negative defense parameter will have a negative impact on the opposing team's attacking parameter. Furthermore, the parameter home represents the advantage for the team hosting the game and the home parameter is assumed to be constant for all the teams and throughout the season in this model. The prior on the home and intercept parameters is flat [4]:

$$home \sim Normal(0, .0001)$$
$$intercept \sim Normal(0, .0001)$$

The team-specific effects are modeled as exchangeable from a common distribution [4]:

$$att_t \sim Normal(\mu_{att}, \tau_{att})$$
$$def_t \sim Normal(\mu_{def}, \tau_{def})$$

In order to insure identifiability, they impose a sum-to-zero constraint on the attack and defense parameters [4]:

$$\sum_{t=1}^{t} att_t = 0$$
$$\sum_{t=1}^{t} def_t = 0$$

Finally, the hyper-priors of the attack and defense effects are modeled using again flat prior distributions [4]:

$$\mu_{att} \sim Normal(0, .0001)$$
$$\mu_{def} \sim Normal(0, .0001)$$
$$\tau_{att} \sim Gamma(.1, .1)$$
$$\tau_{def} \sim Gamma(.1, .1)$$

3.1.2 Reducing over shrinkage caused by a hierarchical model

A known possible drawback of Bayesian hierachical models is the phenomenon of over shrinkage, where some of the extreme observations tend to be pulled towards the mean of the overall observations [4]. The model discussed in Section 3.1 assumes that all the attack and defense propensities are drawn by a common process. This is characterised by the common vector of hyperparameters (μ_{att} τ_{att}, μ_{def} ,τ_{def}). It is clear that this might be not sufficient to capture the difference in quality of the different teams. In the model of Section 3.1, two possible outcomes might occur: a) extremely good teams are penalized; and b) the performance of poor teams is overestimated.

In order to avoid this problem, G. Baio et al. [4] introduced a more complicated structure for the parameters of the model discussed in Section 3.1.

Some aspects remain unchanged, the model for the likelihood, the prior specification for the θ_{gj} and for the hyper-parameter $home$ are unchanged [4]. The other hyper-parameters are modeled as follows. First they defined for each team t two variables $grp^{att}(t)$ and $grp^{def}(t)$, which can take on the values 1, 2 or 3, identifying the bottom-, mid- or top-table performances in terms of attack and defense [4]. These are given categorical distributions, depending on the following vectors of prior probabilities π^{att} = (π^{att}_{1t}, π^{att}_{2t}, π^{att}_{3t}) and π^{def} = (π^{def}_{1t}, π^{def}_{2t}, π^{def}_{3t}) [4]. Both π^{att} and π^{def} follow a Dirichlet distribution with parameters (1, 1, 1) [4].

G. Baio et al. [4] argued that over shrinkage can be limited by modeling the attack and defense parameters using a non central (nct) distribution. The distribution was set on ν= 4 degrees of freedom instead of the normal in Section 3.1 [4]. The non central distribution generalizes a probability distribution using a non centrality parameter [22]. Whereas the central distribution describes how a test statistic t is distributed when the difference tested is null, the non central distribution describes how t is distributed when the null is false [22]. The number of values in the final calculation of a statistic

that are free to vary is called the degrees of freedom [22]. The attack and defense effects are then modeled for each team t as [4]:

$$att_t \sim nct(\mu^{att}_{grp(t)}, \tau^{att}_{grp(t)}, \nu)$$
$$def_t \sim nct(\mu^{def}_{grp(t)}, \tau^{def}_{grp(t)}, \nu)$$

Since the values of $grp^{att}(t)$ and $grp^{def}(t)$ are unknown, the formulation of the mixture model on the attack and defense effects essentially boils down to the following [4]:

$$att_t = \sum_{k=1}^{3} \pi^{att}_{kt} \times \text{nct}(\mu^{att}_k, \tau^{att}_k, \nu)$$
$$def_t = \sum_{k=1}^{3} \pi^{def}_{kt} \times \text{nct}(\mu^{def}_k, \tau^{def}_k, \nu)$$

The model for the location and scale parameters of the nct distributions is specified as follows. If a team has poor performance, it is likely that this team will concede goals and it is unlikely that this team will score goals. In other words, the poor team has low (negative) propensity to score, and high (positive) propensity to concede goals. This can be represented using truncated Normal distributions [4]:

$$\mu^{att}_1 \sim \text{truncNormal}(0, 0.001, -3, 0)$$
$$\mu^{def}_1 \sim \text{truncNormal}(0, 0.001, 0, 3)$$

For the top teams, there is a symmetric situation [4]:

$$\mu^{att}_3 \sim \text{truncNormal}(0, 0.001, 0, 3)$$
$$\mu^{def}_3 \sim \text{truncNormal}(0, 0.001, -3, 0)$$

Finally, the model of the average teams assumes that the mean of the attack and defense effect have independent dispersed Normal distributions [4]:

$$\mu^{att}_2 \sim \text{Normal}(0, \tau^{att}_2)$$
$$\mu^{def}_2 \sim \text{Normal}(0, \tau^{def}_2)$$

For all teams k = 1, 2, 3, the precisions are modeled using Gamma distributions [4]:

$$\tau^{att}_k \sim \text{Gamma}(0.01, 0.01)$$
$$\tau^{def}_k \sim \text{Gamma}(0.01, 0.01)$$

The model of G. Baio et al. [4] discussed above is recreated with data of the Eredivisie, season 2017-2018, in order to properly compare our ERP method with this model. However, due to computational limitations we were unable to recreate the more complex mixture model. Nonetheless, in the paper of G. Baio et al. [4], the more complex mixture model did not show any significant improvement in terms of predictive power. Hence, the failed recreation of the more complex mixture model is not being found of great importance. The validation and comparison of the models will be discussed in Section 4 along with the results. The recreation of this model, along with the more complex mixture model is discussed with greater detail in Appendix A.

3.2 Exact Player Rating

In this section we formulate our Exact Player Rating (EPR), which improves upon the existing player rating models. The EPR method will not only allow us to estimate the skill level of players, but also what strategies are important for winning games. The ERP method is estimated in two stages. The first stage is about estimating the effect of a player's presence on his team's offensive and defensive output of certain production statistics. In the second stage we determine what strategies are most effective to winning. This is done by regressing score differentials on the estimated production statistics from the first stage. We first formulate this two-stage model. In Sections 3.2.1 and 3.2.2 we explain the estimation procedures of the first and second stage of the EPR model respectively.

First we formulate the first stage of the EPR model. Let Z be an $N \times M$ matrix build in the same way as Omidiran's SPR model [14], containing the elements $z_{i,m}$, such that

$$z_{i,m} = \begin{cases} 1 & \text{if player } m \text{ is on the field for the home team} \\ -1 & \text{if player } m \text{ is on the field for the away team} \\ 0 & \text{if player } m \text{ is not on the field} \end{cases}$$

for $m \in \{1, ..., M\}$ and $i \in \{1, ..., N\}$ where M is the number of players that are considered in our model and N is the number of events in our dataset. Let $x_{p,i,j}$ be the difference between the j-th production statistics of the home and away team's players, who play on the p-th position made during the i-th event for $p \in \{1, ..., P\}$, $j \in \{1, ..., K\}$ and $i \in \{1, ..., N\}$. Let $x_{p,j}$ be the $N \times 1$ vector containing the elements $x_{p,i,j}$ with i ranging from 1 to N for a given j and p. The P positions are the positions as described in Section 2, ST for striker, RWF for rightwinger, LWF for leftwinger, AMF for attacking midfielder, RCM for right midfielder, LCM for left midfielder, DMF for defending midfielder, LB for leftback, RB for rightback, CB for centerback and GK for goalkeeper. The K production statistics are discussed in Section 2. With regards to the goal related statistics, we would like to distinguish between different ranges of shots, because players will probably shoot less efficiently when shooting further away from the goal. These statistic however are not available and thus cannot be included in our model.

Continuing with the formulation of our model, the influence of the player's presence on the difference between the output of production statistics of his team and the opponent is estimated in Equation 3.1.

$$\text{Stage I: } x_{p,j} = \kappa + Z\theta_{p,j} + \eta \tag{3.1}$$

Large positive values in the estimated parameter vector $\hat{\theta}_{p,j}$ indicate that players cause a large output for the j-th production statistic of players on his team playing the p-th position, or causing their opponent to have a reduced output for these production statistics. The error terms η_i are assumed to be independently and identically distributed with the Normal distribution.

Equation 3.1 is estimated in the first stage for all positions $p \in \{1, ..., P\}$ and production statistics $j \in \{1, ..., K\}$. Now we define $\hat{\Theta}$ to be the $M \times (P \times K)$ estimated parameter matrix containing the estimated parameter vectors $\hat{\theta}_{p,j}$ for all $p \in \{1, ..., P\}$ and $j \in \{1, ..., K\}$. Furthermore we define $\hat{X}$ to be the $N \times (P \times K)$ matrix containing the vectors $\hat{x}_{p,i}$ estimated in Stage I (Equation (3.1)) for all $p \in \{1, ..., P\}$ and $j \in \{1, ..., K\}$. These results are later needed in Stage II when the Exact Player Rating is formed.

To make it easier to understand, we first summarize the data per team, per player, per game with the following matrix:

$$R_{Heraclesgame\#1} = \begin{array}{c} \\ Player_1 \\ Player_2 \\ \vdots \\ Player_n \end{array} \begin{array}{c} \begin{array}{cccc} Attribute_1 & Attribute_2 & \dots & Attribute_m \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \end{array}$$

We are then able to summarize the total statistics of these players for an entire season. One can imagine computing the aggregate statistics matrix in the following way, where t represents the total number of games played.

$$R_{Heracles} = \sum_{t=1}^{t} R_{Heraclesgame\#t}$$

Finally, we define the matrix R that vertically concatenates R_m across the 18 teams in the Eredivisie:

$$R = \begin{array}{c} \\ Team1 \\ Team2 \\ \vdots \\ Team18 \end{array} \begin{array}{c} \begin{array}{c} \end{array} \\ \begin{pmatrix} R_{Heracles} \\ R_{Ajax} \\ \vdots \\ R_{Twente} \end{pmatrix} \end{array}$$

Once we have an overview of the final matrix and thus an overview of the production statistics per player of the entire season, we are now able to estimate the influence of all production statistics on score differentials in Equation 3.2. In Stage II of our model we are trying to estimate which output of production statistics have the largest influence on score differentials. In other words, which what strategies are most effective to winning or losing. These effects are estimated in $(P \times K) \times 1$ parameter vector β. A large positive value in the parameter vector $\hat{\beta}$ indicate that the corresponding production statistic has a larger effect on winning games. Note that the influence of the presence of players on winning is captured indirectly through $\hat{X}$, which is estimated in Stage I (Equation (3.1)). The distribution for the error terms ε_i is discussed in Section 3.2.2.

$$\text{Stage II} : y = \alpha + \hat{X}\beta + \varepsilon \tag{3.2}$$

The Exact Player Rating is formed after the two stages from Equations (3.1) and (3.2) have been estimated. Let $\hat{\vartheta}_m$ be the m-th row of matrix $\hat{\Theta}$ with dimensions $1 \times (P \times K)$. The EPR for player m is then defined as the following in Equation 3.3.

$$EPR_m = \hat{\theta}_m \hat{\beta} \tag{3.3}$$

Values for EPR should be interpreted as the added value player m is delivering to his team when player m is on the field. The EPR rating can be broken down to see what the strengths and weaknesses of players are. This is done in a similar fashion as how the EPR is defined. The difference is that $\hat{\theta}_m$ and $\hat{\beta}$ are multiplied entrywise instead of multiplied as vectors, which is also known as the Hadamard product [23]. This breakdown is defined in Equation 3.4.

$$EPRBreakdown_m = \hat{\theta}_m \circ \hat{\beta}^T \tag{3.4}$$

This breakdown has the added information of how players contribute to their team in terms of certain

production statistics of players playing in a certain position. Note that summating all the elements of $EPRBreakdown_m$ will result in the EPR of Equation (3.3).

3.2.1 Estimating Stage I

In this section we explain the estimation procedure used to estimate the first stage of our EPR model. Equation 3.1 is estimated in quite a simple fashion where the weight given, $\hat{\beta}$, is +1 for a positive attribute (goals,assist,interceptions) and -1 for a negative attribute (fouls per 90 min, red cards, yellow cards). We choose this algorithm, since no algorithm is available and starting as simple as possible and working our way up in determining the weights in Stage II is usually a good starting point.

3.2.2 Estimating Stage II

In this section we explain the procedure used to estimate the second stage of our EPR model. The parameter vectors $\hat{\beta}$ estimated in Equation 3.3 can be interpreted as the influence that several production statistics have on score differentials.

We have included a variable in the model for each production statistic for each of the possible positions a football player could play. We believe these parameters are not equal over all positions, because different production statistics will not have the same effect on score differentials for the same position. However, there will be some similarity, since they estimate the effect on the same production statistics. Also due the fact that we have a sufficient amount of data, we are able to use some machine learning and regression techniques that are known for feature selection. For these reasons we believe that some of the most common machine learning algorithms, such as Multiple Linear Regression (MLR), ridge regression, lasso regression, XGBoost, random forest and decision tree are appropriate in order to estimate Equation 3.2. The machine learning algorithms are available in Python, a scripting language we use to program the different mathematical functions.

To further explain why the machine learning model estimations are so appropriate, we formulate the different models and functions it seeks to minimize.

Lastly, we provide a model specification for a simplified model in order to provide evidence that not every position has the same effect on score differentials.

Multiple Linear Regression

Before we discuss MLR, note that Equation 3.2 can also be written as:

$$y_i = \alpha + \sum_{p=1}^{P} \sum_{j=1}^{K} \hat{x}_{p,i,j} \beta_{p,j} + \varepsilon_i \tag{3.5}$$

Where $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)$ with $i \in \{1, ..., N\}$, $p \in \{1, ..., P\}$ and $j \in \{1, ..., K\}$.

MLR is one of the most widely known modeling techniques [24]. With a MLR analysis, we wish to predict a scalar response variable y_i, given a vector of predictors [24]. Our scalar response variable y_i, will be the "Rating" of a player and the predictions will be our set of j production statistics with $j \in \{1, ..., K\}$. The relationship between the dependent variable and the explanatory variables is represented by Equation 3.6 [24]:

$$y_i = \alpha + \beta_1 j_{i,1} + \beta_2 j_{i,2} + \beta_k j_{i,k} + \varepsilon_i \tag{3.6}$$

Note that Equation 3.6 is quite similar to our own Equation 3.5, since the only aspect missing is the position statistic P with $p \in \{1, ..., P\}$.

Recall that we defined $\hat{X}$ to be the $N \times (P \times K)$ matrix containing the vectors $\hat{x}_{p,i}$ estimated in the first stage (Equation (3.1)) for all $p \in \{1, ..., P\}$ and $j \in \{1, ..., K\}$. Equation 3.6 can therefore be written as:

$$y_i = \hat{X}\beta + \varepsilon \tag{3.7}$$

where $\hat{X}\beta$ represents the matrix-vector product.

In order to estimate $\hat{\beta}$, we take a least squares approach. That is, we want to minimize the following function over all possible values of the intercept and slopes [24]:

$$\sum_i (y_i - \alpha - \beta_1 \hat{X}_{i,1} - \beta_2 \hat{X}_{i,2} - \beta_k \hat{X}_{i,k})^2 \tag{3.8}$$

Equation 3.8 is minimized by setting [24]:

$$\hat{\beta} = (\hat{X}'\hat{X})^{-1}\hat{X}'Y \tag{3.9}$$

Equation 3.9 is a point estimate, but fitting different samples of data from the population will cause the best estimators to shift around. The amount of shifting can be explained by the variance-covariance matrix of $\hat{\beta}$ [24]:

$$Cov(\hat{\beta}, \hat{\beta}) = \sigma^2(\hat{X}'\hat{X})^{-1} \tag{3.10}$$

MLR seems to be a good fit for estimating Stage II. Equation 3.6 is quite similar to Equation 3.5, since the only aspect missing is the position statistic P with $p \in \{1, ..., P\}$, but this should be simple to enclose. Another point is that a sufficient amount of data is imperative, but also this is not an issue. Nonetheless, we need to be extremely careful with the issue of overfitting the data with MLR [25].

Overfitting the data is an extremely common issue in many machine learning problems and one of the most common instruments for avoiding overfitting is called regularization [25]. Regularized machine learning models are models where the loss function minimizes another element as well [25]. This second element sums over squared β values and multiplies it by the parameter λ [25]. The reasoning is to punish the loss function for high values of the coefficients β, making it a simpler model [25]. It is possible that simpler models obtain a better fit for our ERP method than complex models. Therefore, we also need to try and simplify the model as much as possible and compare the results. Two regularization models are ridge- and lasso regression, which is why we also specify these two models for our EPR method.

Ridge Regression

Ridge regression can be seen as an extension for MLR [26]. With ridge regression, we wish to minimize the following function:

$$\begin{aligned} &\text{Minimize}(Y - \beta'\hat{X})'(Y - \beta'\hat{X}) + \lambda\beta'\beta \\ &\text{Subject to} \sum_{j=1}^{K} \beta_j^2 <= t \end{aligned} \tag{3.11}$$

where t represents the specified free parameter that determines the amount of regularisation and λ represents the penalty coefficient, which can be any value [26]. Note that as t comes close to infinity,

the problem becomes an ordinary least squares and λ becomes 0, since the relation between λ and the upper bound t is a reverse relationship [27].

Ridge regression has one major advantage over MLR, as it penalizes the estimated β values, which is represented by the $\lambda\beta'\beta$ term [26]. Recall that the β values can be interpreted as the influence that several production statistics have on score differentials. Ridge regression does not penalize all the estimated β values in similar fashion [26]. If the estimated β values are very large, then the $(Y - \beta'\hat{X})'(Y - \beta'\hat{X})$ term in the above equation will minimize, but the penalty term will increase [26]. If the estimated β values are small however, then the penalization will be minimized, but the $(Y-\beta'\hat{X})'(Y-\beta'\hat{X})$ term will increase due to poor generalization [26]. Ridge regression thus chooses to penalize the estimated β values in such a way that less influential features undergo more penalization. Adding a penalty term reduces overfitting and since we have a lot of β values to estimate, ridge regression may be more beneficial for our model than MLR.

The ridge regression estimate is given by:

$$\hat{\beta}_{Ridge} = (\hat{X}'\hat{X} + \lambda I)^{-1}\hat{X}'Y \tag{3.12}$$

Lasso Regression

LASSO, Least Absolute Shrinkage and Selection Operator, is a regularization and variable selection method for statistical models [27]. Lasso regression seeks to minimize the sum of squared errors, which is comparable with ridge regression [25]. The only difference from ridge regression is that the regularization term is given in absolute value [25]. This means that the lasso regression method is not only punishing high values of the coefficients β, but actually sets them to zero if they are not relevant [25]. Therefore, we might end up with fewer features included in the model, making the model less complex, which can be a huge advantage. There are different mathematical formulations for lasso regression, but we will refer to the formulation used by Bühlmann and van de Geer [27].

The lasso estimate is defined by the solution to the optimization problem:

$$\begin{aligned} &\text{Minimize} \left(\frac{\|Y - \hat{X}\beta\|_2^2}{n} \right) \\ &\text{subject to} \sum_{j=1}^{K} |\beta_j| <= t \end{aligned} \tag{3.13}$$

where t, again represents the specified free parameter that determines the amount of regularisation [27].

This optimization problem is equivalent to the parameter estimation that follows:

$$\hat{\beta}(\lambda) = \underset{\beta}{\operatorname{argmin}} \left(\frac{\|Y - \hat{X}\beta\|_2^2}{n} + \lambda\|\beta\|_1 \right) \tag{3.14}$$

where $\|Y - \hat{X}\beta\|_2^2 = \sum_{i=0}^{N}(Y_i - (\hat{X}\beta)_i)^2$, $\|\beta\|_1 = \sum_{j=1}^{K}|\beta_j|$ and where $\lambda \geq 0$ is the parameter that controls the strength of the penalty [27]. In other words, the larger the value of λ, the greater the amount of shrinkage. Note again that as t comes close to infinity, the problem becomes an ordinary least squares and λ becomes 0, since the relation between λ and the upper bound t is a reverse relationship [27].

Lasso- and ridge regression are potentially a good fit for our EPR model, but there still may be better models. There are other machine learning models that are able to handle messier data and messier relationships better than regression models, which may lead to a better fit for our EPR model. These models are XGBoost, random forest and the decision tree, which is why we also specify these three models for our EPR method.

XGBoost

XGBoost is a form of gradient boosting [28]. Gradient boosting is a fairly new machine learning technique for regression and classification problems. Gradient boosting produces a prediction model in the form of decision trees and is comparable with random forest and the decision tree algorithm, which will be discussed next.

We follow the mathematical formulations of T. Chen et al. [29]. We first fit a model to the data, $F_1(\hat{X}) = y$ [29]. Then we fit a model to the residuals, $h_1(\hat{X}) = y - F_1(\hat{X})$ [29]. The third step is to create a new model, $F_2(\hat{X}) = F_1(\hat{X}) + h_1(\hat{X})$ [29]. It is possible to generalize this idea by creating more models that improve upon the previous models by correcting the errors, $F(\hat{X}) = F_1(\hat{X}) \rightarrow F_2(\hat{X}) = F_1(\hat{X}) + h_1(\hat{X}) ... \rightarrow F_M(\hat{X}) = F_{M-1}(\hat{X}) + h_{M-1}(\hat{X})$ where $F_1(\hat{X})$ is the initial model fitted to y [29].

Since the procedure is initialized by fitting $F_1(\hat{X})$, we wish to to find the gradient boosting solution at each step given by [29]:

$$h(\hat{X}) = y - F_m(\hat{X}) \tag{3.15}$$

In order to minimize the squared error, we initialize F with the mean of the training target values:

$$F_0(\hat{X}) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(\gamma - y_i)^2 = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{3.16}$$

where L represents the loss function , m the number of iterations and γ the step size [29].

Now we can define each subsequent $F_m(\hat{X})$:

$$F_m(\hat{X}) = F_{m-1}(\hat{X}) + \underset{h_m \in H}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(\hat{X}_i) + h_m(\hat{X}_i)) \tag{3.17}$$

where $h_m \in H$ represents the base learner function [29].

However, due to computational limitations, calculating h_m at each step is infeasible. We are forced to apply some sort of simplification, which will be in the from of applying a steepest descent step to this minimization problem. Equation 3.17 can then be written as [29]:

$$F_m(\hat{X}) = F_{m-1}(\hat{X}) - \gamma_m \sum_{i=1}^{n} \nabla F_{m-1} L(y_i, F_{m-1}(\hat{X}_i)) \tag{3.18}$$

This optimization problem is minimized by setting [29]:

$$y_m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(\hat{X}_i) - \gamma \nabla F_{m-1} L(y_i, F_{m-1}(\hat{X}_i))) \tag{3.19}$$

Random Forest and Decision Tree

The last machine learning models that we use for our EPR model are the random forest and decision tree algorithms.

Random forest is a supervised learning algorithm [30]. The algorithm builds multiple decision trees and merges them together in order to obtain a more accurate and stable prediction [30]. The random forest algorithm and the decision tree algorithm are comparable, with the exception that in random forest the processes of finding the root node and splitting the feature nodes is random [30].

Random forest and the decision tree algorithm focus on fully grown decision trees (low bias, high variance), which means that these algorithms tackle the error reduction by reducing variance [30]. XGBoost on the other hand is focused on weak learners (high bias, low variance), which means that XGBoosting reduces error mainly by reducing bias [27].

Both random forest and decision tree applies the general technique of bootstrap aggregating [30]. This is often called bagging, which is represented with the parameter B [30]. The basic idea of bagging is to resample the data continuously and train a new classifier for each sample [30]. Different classifiers will overfit the data in a different way and these differences are averaged out at the end [30]. Since the training algorithm for random forest and decision tree is similar to the approach discussed with the XGBoost algorithm, we only discuss the regression function, which is given below:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(\hat{X}') \tag{3.20}$$

where f_b is the classification or regression tree trained on the data [30]. Note that the regression function is very similar to Equation 3.16. The main difference is that we use the transpose of $\hat{X}$, which is logical, since random forest and decision tree works the other way around compared to XGboost.

Training and variable selection

In this section we provide further information regarding the training procedures for the methods discussed above in Section 3.2.2. We also describe how we have selected the variables that are included in our EPR model.

With regards to the training procedures, we start by dividing and using an 80% / 20% of training and validation data for every model. There is no rule of thumb for dividing datasets and 80% / 20% feels like a safe bet. The percentages can always be changed if the results may seem off. Continuing with the training procedures, for every model we start with the default values and look for better results. This is done by changing the sample size and fit-intercept value for MLR, changing the alpha value and random state value for the lasso- and ridge regression, changing the number of estimators and learning rate for XGBoost, changing the max depth, max leaf nodes, number of estimators and random state for random forest and finally, changing the max depth, number of estimators and random state for the decision tree algorithm.

With regards to the variable selection, the 95%-confidence interval for all parameters is checked and an Analysis of Variance (ANOVA) is considered. The 95%-confidence intervals and the ANOVA can provide some insight regarding the significance of a variable. If a confidence interval contains a zero,

it cannot be said with some certainty that that parameter is not equal to zero and that the variable should be removed from the model. With the ANOVA, we are able to calculate the F-value of a certain variable. The F-value can be compared with the t-statistic of a single variable. If this value comes close to 0, this shows that this variable is not significant and should be removed from the model [24]. Afterwards, we estimate Equation 3.2 once more without the removed variables.

Simplified model

In this section we provide a simplification of the second stage of our EPR model. Our goal with this model is to provide evidence that not every position has the same effect on score differentials.

We have already discussed that the MLR model might suffer from a drawback, namely that it might be overcomplicated. This model needs $(P \times K)$+ 1 variables to be estimated and overfitting the data is a common issue with MLR [25]. Using too many variables may lead to a poor predictive performance of our method. We therefore also consider a simplified model for MLR.

We pool all variables with respect to the position. So, we assume that the variables have the same effect for all positions. This decreases the amount of variables with a factor P. The simplified model thus becomes:

$$y_i = \alpha + \sum_{j=1}^{K} \hat{x}^*_{i,j}\beta_j + \varepsilon_i \tag{3.21}$$

We will not discuss the full mathematical formulations, since we follow the exact same steps as discussed in the MLR section. This less complicated model may give better results for MLR, but we still believe one of our other models will have better results in terms of forecasting power. With regards to the training and variable selection, we follow the exact same selection procedures as we have explained for MLR.

3.3 Model validation and comparison

We compare the performance of our EPR model to the Bayesian hierarchical model. First, we compare the different models discussed in Section 3.2.2 in terms of accuracy, cross validation score, mean absolute error, baseline mean squared error, mean squared error, the goodness of fit and the adjusted goodness of fit. This will already provide a considerable overview of the best fit for our EPR model.

Once we have established which of the models is the best fit for our EPR model, we are able to compare our EPR method with the Bayesian hierarchical model. The most important, but by any means not the only aspect, will be looking at the fraction of games guessed right in terms of wins/losses in order to form our prediction of the rankings. Another important criteria is the Bayesian information criterion (BIC) [31]. The BIC is an estimate of a function of the posterior probability of a model being true, under a certain Bayesian setup. The BIC is formally defined as:

$$BIC = ln(n)k - 2ln(\hat{L}) \tag{3.22}$$

where

- $\hat{L}$ is the maximized value of the likelihood function of the model M, $\hat{L} = p(x|\hat{\theta}, M)$, where $\hat{\theta}$ are the parameter values that maximize the likelihood function

- x is the observed data
- n is the number of data points in x
- k is the number of parameters estimated by the model

Among our models, the model with the lowest BIC is preferred [31].

Other scores such as the cross validation score, mean absolute error, baseline mean squared error, mean squared error, the goodness of fit and the adjusted goodness of fit are difficult to compare due to the fact that the Bayesian hierarchical model is not a machine learning model. It is therefore not possible to divide the data into training and validation data.

Chapter 4: Comparing the methods

In this section we compare the predictive accuracy and results of our EPR method with the recreated model of G.Baio et al. [4]

Before we discuss the various results of the models, we first discuss the rankings of the Eredivisie 2017-2018. The rankings are obtained from https://www.fcupdate.nl/stand/s1271/nederland-eredivisie-2017-2018/ where GS represents the amount of goals scored, GC represents the amount of goals conceded, GD represents the goal difference and the last column is the total points per team. The ranking of the Eredivisie 2017-2018 can be found in Table 4.1.

Team	P	W	D	L	GS	GC	GD	Pts
PSV	34	26	5	3	87	39	48	83
Ajax	34	25	4	5	89	33	56	79
AZ	34	22	5	7	72	38	34	71
Feyenoord	34	20	6	8	76	39	37	66
FC Utrecht	34	14	12	8	58	53	5	54
Vitesse	34	13	10	11	63	47	16	49
ADO Den Haag	34	13	8	13	45	53	-8	47
SC Heerenveen	34	12	10	12	48	53	-5	46
PEC Zwolle	34	12	8	14	42	54	-12	44
Heracles Almelo	34	11	9	14	50	64	-14	42
Excelsior	34	11	7	16	41	56	-15	40
FC Groningen	34	8	14	12	50	50	0	38
Willem II	34	10	7	17	50	63	-13	37
NAC Breda	34	9	7	18	41	57	-16	34
VVV Venlo	34	7	13	14	35	54	-19	34
Roda JC	34	8	6	20	42	69	-27	30
Sparta	34	7	6	21	34	75	-41	27
FC Twente	34	5	9	20	37	63	-26	24

Table 4.1: Ranking of the Eredivisie 2017-2018.

4.1 Bayesian hierarchical model

From the model described in Section 3.1.1, we can start to understand the different distributions of attacking strength and defensive strength per team. We have to take advantage of the fact that we can quantify our posterior uncertainty. We have written a code in order to look at the Highest Posterior Density intervals for the attack parameters. The code along with the figure is given below:

```
In [10]: df_hpd = pd.DataFrame(pm.stats.hpd(trace['atts']),
                               columns=['hpd_low', 'hpd_high'],
                               index=teams.team.values)
         df_median = pd.DataFrame(pm.stats.quantiles(trace['atts'])[50],
                                  columns=['hpd_median'],
                                  index=teams.team.values)
         df_hpd = df_hpd.join(df_median)
         df_hpd['relative_lower'] = df_hpd.hpd_median - df_hpd.hpd_low
         df_hpd['relative_upper'] = df_hpd.hpd_high - df_hpd.hpd_median
         df_hpd = df_hpd.sort_values(by='hpd_median')
         df_hpd = df_hpd.reset_index()
         df_hpd['x'] = df_hpd.index + .5


         fig, axs = plt.subplots(figsize=(10,4))
         axs.errorbar(df_hpd.x, df_hpd.hpd_median,
                      yerr=(df_hpd[['relative_lower', 'relative_upper']].values).T,
                      fmt='o')
         axs.set_title('HPD of Attack Strength, by Team')
         axs.set_xlabel('Team')
         axs.set_ylabel('Posterior Attack Strength')
         _= axs.set_xticks(df_hpd.index + .5)
         _= axs.set_xticklabels(df_hpd['index'].values, rotation=45)
```

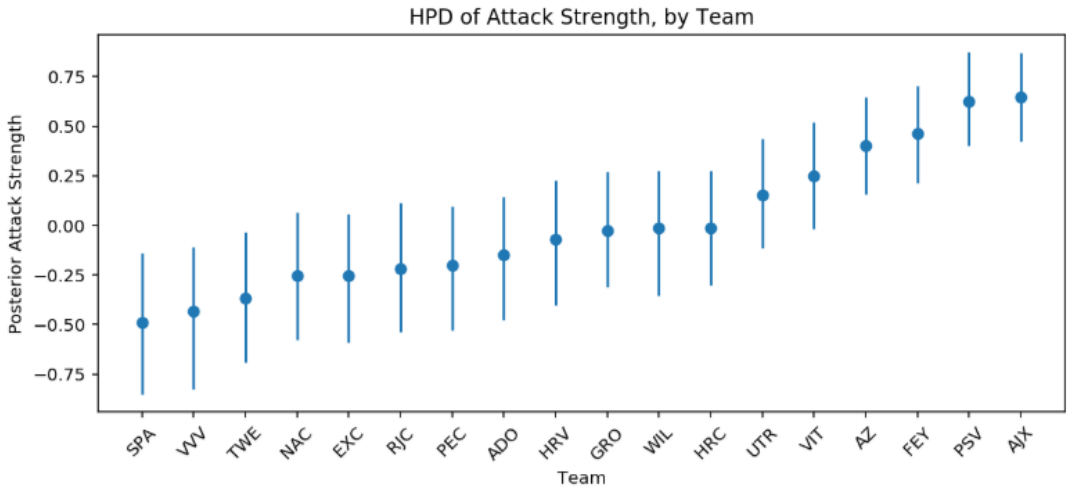

Figure 4.1: HPD of attacking strength.

From Figure 4.1, we see that the top 4 and maybe even the top 6 teams are clearly ahead of the other teams when it comes to attacking strength. Continuing with the top teams, we see that the top two teams, Ajax and PSV are in a league of their own and are by far the two best attacking teams. Furthermore, we see that the last three teams, Sparta, VVV Venlo and FC Twente are quite far behind the middle-ranking, whereas the teams in the middle-ranking have quite similar attacking strength. In order to obtain a more clear-cut overview, we have split the attacking and defensive strength per team, which is represented in the figures below.

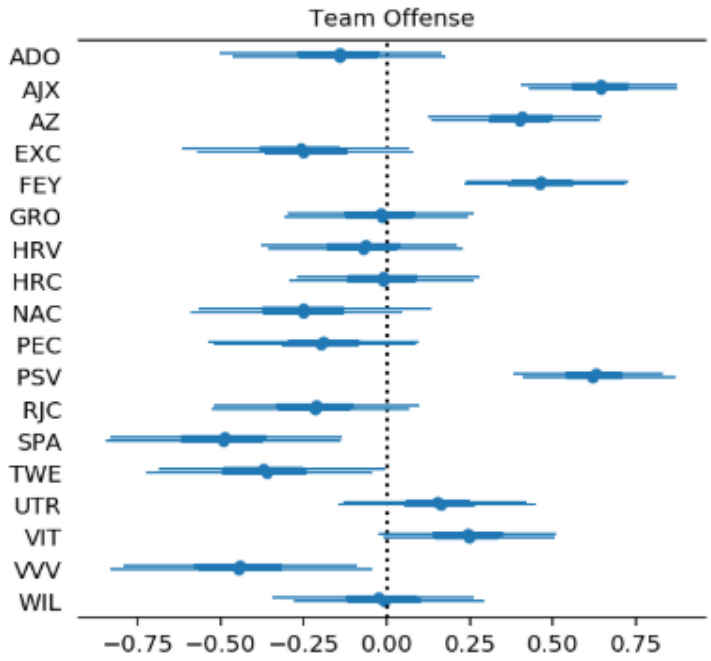

Figure 4.2: Splitting attacking strength per team.

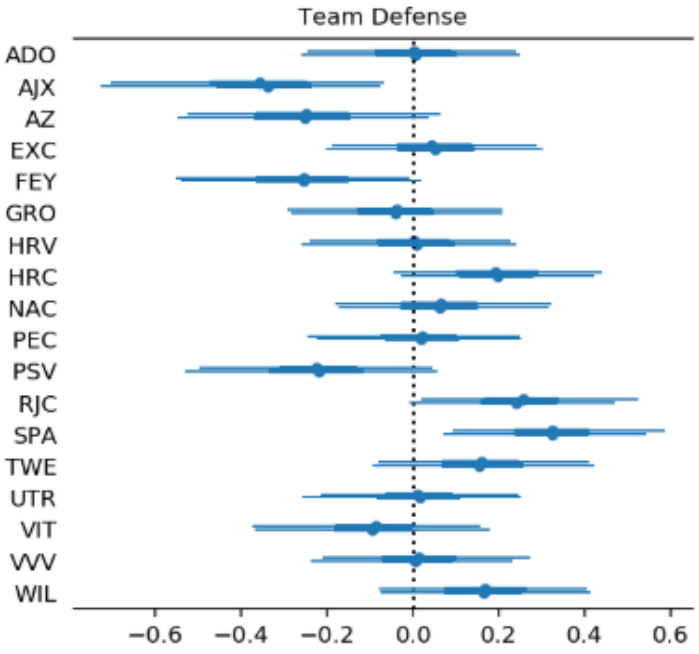

Figure 4.3: Splitting defensive strength per team.

We expect strong teams like Ajax, Feyenoord and PSV to have strong positive effects in attack and strong negative effects in defense. When looking at Figure 4.2 and 4.3, this seems to be correct. The parameters are thus in line with what is expected and look good.

Now let's give special attention to Heracles. Heracles is represented by team 7 in the model, so let's take a look at the defensive effects for Heracles.

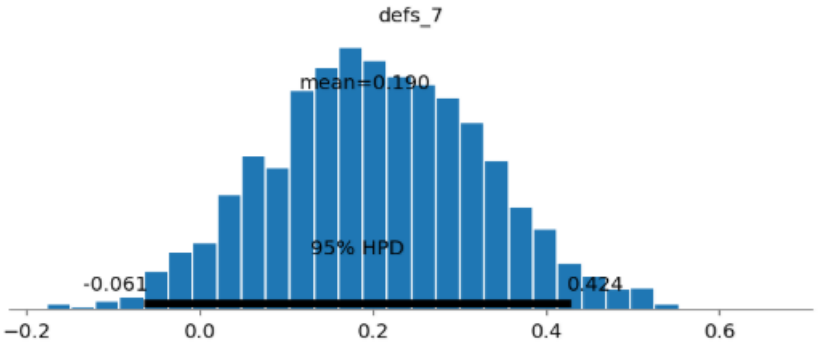

Figure 4.4: Defensive effects for Heracles.

From Figure 4.4, we see that the mean is 0.190. This can be interpreted as Heracles not having a strong defense. The mean is positive, which means that we expect Heracles to concede more goals than it scores. Looking at Table 4.1, where Heracles goal difference is -14, this seems to be correct.

Looking back at Figure 4.1, we see that Heracles is number 7 when it comes to attacking strength. Nonetheless, it seems that their defensive strength weights them down and is their bottleneck. From Figure 4.4, we see that the 95% HPD is between -0.061 and 0.424. Based on their defensive skills, Heracles would be placed number 14 in the rankings. A simple conclusion based on this model would thus be that Heracles has to upgrade their defense.

Now that we have established that the different parameters look promising, we are able to simulate the rankings and determine the winner over a 1000 seasons with the following code:

```
In [15]: with model:
             pp_trace = pm.sample_ppc(trace)

         100%|██████████| 1000/1000 [00:11<00:00, 88.74it/s]

In [17]: home_sim_df = pd.DataFrame({
             'sim_points_{}'.format(i): 3 * home_won
                 for i, home_won in enumerate(pp_trace['home_points'] > pp_trace['away_points'])
         })
         home_sim_df.insert(0, 'team', df['home'])

         away_sim_df = pd.DataFrame({
             'sim_points_{}'.format(i): 3 * away_won
                 for i, away_won in enumerate(pp_trace['home_points'] < pp_trace['away_points'])
         })
         away_sim_df.insert(0, 'team', df['away'])

In [18]: sim_table = (home_sim_df.groupby('team')
                                 .sum()
                                 .add(away_sim_df.groupby('team')
                                                 .sum())
                                 .rank(ascending=False, method='min', axis=0)
                                 .reset_index()
                                 .melt(id_vars='team', value_name='rank')
                                 .groupby('team')
                                 ['rank']
                                 .value_counts()
                                 .unstack(level='rank')
                                 .fillna(0)
                                 .div(1000))
```

Figure 4.5: Simulation code.

The simulation code generated the following table:

Out[18]:

rank	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0
team																		
ADO	0.000	0.003	0.013	0.022	0.051	0.071	0.085	0.097	0.102	0.087	0.091	0.089	0.068	0.073	0.043	0.063	0.029	0.013
AJX	0.503	0.267	0.142	0.047	0.024	0.010	0.004	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AZ	0.110	0.189	0.216	0.240	0.112	0.060	0.030	0.012	0.012	0.009	0.005	0.002	0.002	0.000	0.000	0.001	0.000	0.000
EXC	0.000	0.000	0.002	0.006	0.017	0.047	0.081	0.052	0.084	0.092	0.091	0.100	0.072	0.092	0.077	0.076	0.065	0.046
FEY	0.166	0.214	0.251	0.189	0.095	0.044	0.022	0.006	0.006	0.003	0.002	0.000	0.001	0.001	0.000	0.000	0.000	0.000
GRO	0.002	0.004	0.019	0.056	0.086	0.120	0.128	0.102	0.112	0.091	0.084	0.060	0.053	0.035	0.024	0.013	0.007	0.004
HRC	0.001	0.003	0.006	0.022	0.036	0.057	0.088	0.102	0.104	0.097	0.084	0.091	0.078	0.072	0.060	0.046	0.029	0.024
HRV	0.003	0.001	0.014	0.030	0.073	0.110	0.103	0.110	0.099	0.088	0.073	0.072	0.065	0.056	0.044	0.031	0.021	0.007
NAC	0.000	0.001	0.004	0.011	0.027	0.036	0.037	0.064	0.066	0.097	0.083	0.094	0.093	0.101	0.088	0.092	0.068	0.038
PEC	0.001	0.002	0.004	0.021	0.037	0.061	0.068	0.078	0.082	0.114	0.090	0.081	0.080	0.072	0.063	0.063	0.052	0.031
PSV	0.338	0.317	0.160	0.112	0.039	0.016	0.015	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RJC	0.000	0.000	0.000	0.004	0.007	0.016	0.032	0.051	0.079	0.071	0.077	0.085	0.087	0.102	0.107	0.112	0.089	0.081
SPA	0.000	0.000	0.000	0.000	0.000	0.005	0.004	0.008	0.010	0.021	0.029	0.045	0.060	0.073	0.089	0.138	0.191	0.327
TWE	0.000	0.000	0.002	0.003	0.005	0.014	0.031	0.029	0.058	0.064	0.072	0.080	0.096	0.090	0.105	0.128	0.123	0.100
UTR	0.004	0.021	0.059	0.109	0.152	0.178	0.111	0.103	0.068	0.047	0.049	0.039	0.024	0.013	0.006	0.008	0.006	0.003
VIT	0.018	0.046	0.097	0.175	0.232	0.151	0.088	0.056	0.047	0.037	0.023	0.007	0.013	0.004	0.003	0.001	0.002	0.000
VVV	0.000	0.000	0.002	0.004	0.014	0.017	0.031	0.030	0.053	0.056	0.068	0.090	0.091	0.092	0.123	0.134	0.117	0.078
WIL	0.000	0.004	0.013	0.023	0.044	0.083	0.103	0.098	0.098	0.085	0.085	0.088	0.071	0.072	0.056	0.041	0.023	0.013

Figure 4.6: Simulation of the rankings.

Based on the simulation table, we are able to compare the actual rankings with the predictions of the Bayesian hierarchical model, which is represented in Table 4.2.

Bayesian model	
Actual ranking	Prediction
PSV	Ajax
Ajax	PSV
AZ	Feyenoord
Feyenoord	AZ
FC Utrecht	Vitesse
Vitesse	FC Utrecht
ADO den Haag	FC Groningen
SC Heerenveen	SC Heerenveen
PEC Zwolle	ADO den Haag
Heracles	PEC Zwolle
Excelsior	Willem II
FC Groningen	Heracles
Willem II	Excelsior
NAC Breda	NAC Breda
VVV Venlo	VVV Venlo
Roda JC	FC Twente
Sparta	Roda JC
FC Twente	Sparta

Table 4.2: Ranking of the Eredivisie, 2017-2018.

Comparing the prediction with the actual ranking, we see that Ajax is +1 above the actual ranking, PSV is -1 below the actual ranking and Feyenoord is +1 above the actual ranking. Continuing in similar fashion, we are able to calculate the total deviations from the actual ranking. The total deviations from the actual ranking for the Bayesian hierarchical model and EPR model is compared in Table 4.5 and Table 4.9. Further comparison between the different models is discussed in Section 4.4.

4.2 Results of Stage I

Recall that Stage I (Equation 3.1) is estimated in quite a simple fashion where the weight, $\hat{\beta}$, is +1 for a positive attribute (goals,assist,interceptions) and -1 for a negative attribute (fouls per 90 min, red cards, yellow cards). From Stage I, we were able to determine the total score per team, the average score per player and the total score of the starting eleven. The results of Stage I can be found in Table 4.3.

Stage I			
Team	Total score	Average score	Score starting 11
PSV	99667	4333	54520
Ajax	116825	4028	53960
AZ	100085	3849	52568
Feyenoord	88104	6293	53183
FC Utrecht	93639	3901	54363
Vitesse	88148	3391	51534
ADO den Haag	71499	2749	40249
SC Heerenveen	78682	3278	46182
PEC Zwolle	76004	2714	46522
Heracles	77021	3080	44145
Excelsior	71900	2876	41871
FC Groningen	63904	3043	42489
Willem II	78634	3276	47859
NAC Breda	69540	2674	38414
VVV Venlo	69167	3143	45219
Roda JC	75165	3131	43518
Sparta	87902	3255	38544
FC Twente	80095	2966	42544

Table 4.3: Scores of Stage I, Eredivisie, 2017-2018.

From Table 4.3 we are able to predict the outcome of the Eredivisie 2017-2018. The predictions are summarized in Table 4.4.

Stage I			
Team	Prediction total score	Prediction average score	Prediction score starting 11
PSV	Ajax	Feyenoord	PSV
Ajax	AZ	PSV	FC Utrecht
AZ	PSV	Ajax	Ajax
Feyenoord	FC Utrecht	FC Utrecht	Feyenoord
FC Utrecht	Vitesse	AZ	AZ
Vitesse	Feyenoord	Vitesse	Vitesse
ADO den Haag	Sparta	SC Heerenveen	Willem II
SC Heerenveen	FC Twente	Willem II	PEC Zwolle
PEC Zwolle	SC Heerenveen	Sparta	SC Heerenveen
Heracles	Willem II	VVV Venlo	VVV Venlo
Excelsior	Heracles	Roda JC	Heracles
FC Groningen	PEC Zwolle	Heracles	Roda JC
Willem II	Roda JC	FC Groningen	FC Twente
NAC Breda	Excelsior	FC Twente	FC Groningen
VVV Venlo	ADO den Haag	Excelsior	Excelsior
Roda JC	NAC Breda	ADO den Haag	ADO den Haag
Sparta	VVV Venlo	PEC Zwolle	Sparta
FC Twente	FC Gronigen	NAC Breda	NAC Breda

Table 4.4: Prediction of the rankings, Eredivisie 2017-2018, Stage I.

Calculating the total deviations from Stage I in similar fashion as explained in Section 4.1, we are able to compare the predictive power of Stage I with the Bayesian hierarchical model. The results are given in Table 4.5.

	Deviations		
Bayesian hierarchical model	Total score	Average score	Score starting 11
24	59	64	48

Table 4.5: Total number of deviations, Bayesian vs Stage I.

From the results of the predictive power, it is clear that the score of the starting eleven is the most accurate, but by far not accurate enough, since the Bayesian hierarchical model is still the most preferable model. Stage I is clearly not sufficient and the weights have to be chosen correctly. We elaborate more on this in Section 4.3.

4.3 Results of Stage II

In order to determine which of the methods discussed in Section 3.2.2 is the best fit for our EPR model, we calculate the various scores between the models, which were also discussed in Section 3.2.2. Further explanation for the calculation of the codes can be found in Appendix B. The results of the different scores are given in Table 4.6.

	Accuracy	Cross validation	Mean absolute error	Baseline mean squared error	Mean squared error	Goodness of fit	Adjusted Goodness of Fit
MLR Default values	0,436	0,354	10,263	148,585	172,041	0,480	0,461
MLR Changing n_jobs	0,436	0,354	10,263	148,585	172,041	0,480	0,461
MLR Changing fit intercept value	-6,715	-9,246	46,304	2294,426	2346,551	-6,857	-7,156
Lasso Default values	0,431	0,352	10,399	148,972	172,748	0,477	0,458
Lasso Changing alpha	0,431	0,352	10,399	148,972	172,748	0,477	0,458
Lasso Changing random state	0,431	0,352	10,399	148,972	172,748	0,477	0,458
Ridge Default values	0,430	0,354	10,514	153,296	174,012	0,479	0,459
Ridge Changing alpha	0,435	0,353	10,264	148,796	173,167	0,477	0,460
Ridge Changing random state	0,430	0,354	10,514	153,296	174,012	0,479	0,459
XGBoost Default values	0,360	0,273	10,907	153,584	196,593	0,794	0,787
XGBoost Changing n_estimators and learning rate	0,001	-0,139	13,758	277,006	304,843	0,807	0,802
Random Forest Default values	0,340	0,270	11,080	142,530	201,296	0,767	0,759
Random Forest Changing n_estimators and random state	0,384	0,309	10,576	122,185	188,468	0,793	0,788
Random Forest Changing max depth, max leaf nodes and random state	0,364	0,262	10,866	134,775	195,152	0,755	0,745
Random Forest Changing max depth, n_estimators and random state	0,378	0,320	10,695	120,750	190,290	0,791	0,783
Decision Tree Default values	-0,090	-0,380	13,870	291,370	330,085	0,750	0,740
Decision Tree Changing max depth, n_estimators and random state	-0,105	-0,330	14,040	295,750	337,183	0,743	0,735
Simplified model Default values	0,000	0,318	143,846	17697,600	13344223,590	0,339	0,337
Simplified model Changing n_jobs	0,000	0,318	143,846	17697,600	13344223,590	0,339	0,337
Simplified model Changing fit intercept value	0,000	-6,910	178,715	7826,040	13334563,390	-6,674	-6,700

Table 4.6: Scores between the different models.

Let's start with comparing the multiple linear-, lasso- and ridge regression. Although the scores of the different regression models are very similar, MLR has the best score in every statistic compared to lasso- and ridge regression. We can thus conclude that MLR is the best fit when comparing the different regression models.

Continuing with the comparison, we see that the XGBoost, random forest and the decision tree algorithms have excellent outcomes when it comes to the goodness of fit and the adjusted goodness of fit. However, these methods have lesser results when it comes to the accuracy, cross validation, mean absolute error and mean squared error. Hence, we need to make a decision. Does the goodness of fit and the adjusted goodness of fit outweigh the other shortcomings? We came to the conclusion that this is not the case. The goodness of fit and the adjusted goodness of fit are important statistical measures, since it will give us insight of how well the regression line approximates the real data points [23], but this does not outweigh the other statistical shortcomings. The other statistical measures provide insight in the variance and consistency of the model, which is also highly relevant. Every statistic is significant and since MLR is winning 4 out of the 7 measures, MLR is still the most preferred model.

The last comparison is between MLR and the simplified model. It is clear that the simplified model is by far the worst fit for our model, since it scores worse on every aspect compared to the other models. We can thus conclude that MLR is the best fit for our EPR model.

Now that we have established that MLR is the best fit for our EPR model, we are able to make a prediction of the rankings based on the total score per team, the average score per player and the total score of the starting eleven. The scores and the prediction of Stage II can be found in Table 4.7 and Table 4.8.

Stage II			
Team	Total score	Average score	Score starting 11
PSV	498,92	21,69	254,88
Ajax	645,06	23,04	256,53
AZ	516,62	22,46	253,52
Feyenoord	452,29	20,56	246,83
FC Utrecht	496,15	21,57	230,55
Vitesse	444,77	19,34	229,67
ADO den Haag	462,64	18,51	216,82
SC Heerenveen	450,95	19,61	225,99
PEC Zwolle	424,32	16,32	210,56
Heracles	495,27	19,81	209,61
Excelsior	422,45	18,39	207,49
FC Groningen	344,88	18,15	213,20
Willem II	444,98	18,54	205,02
NAC Breda	405,45	17,63	202,39
VVV Venlo	415,99	18,91	201,89
Roda JC	423,74	18,42	201,99
Sparta	525,79	17,53	200,69
FC Twente	393,46	17,11	199,47

Table 4.7: Scores of Stage II, Eredivisie, 2017-2018.

Stage II			
Team	Prediction total score	Prediction average score	Prediction score starting 11
PSV	Ajax	Ajax	Ajax
Ajax	Sparta	AZ	PSV
AZ	AZ	PSV	AZ
Feyenoord	PSV	FC Utrecht	Feyenoord
FC Utrecht	Feyenoord	Feyenoord	Utrecht
Vitesse	Heracles	Heracles	Vitesse
ADO den Haag	ADO den Haag	SC Heerenveen	SC Heerenveen
SC Heerenveen	Feyenoord	Vitesse	ADO den Haag
PEC Zwolle	SC Heerenveen	VVV Venlo	FC Groningen
Heracles	Willem II	Willem II	PEC Zwolle
Excelsior	Vitesse	ADO den Haag	Heracles
FC Groningen	PEC Zwolle	Roda JC	Excelsior
Willem II	Roda JC	Excelsior	Willem II
NAC Breda	Excelsior	FC Groningen	NAC Breda
VVV Venlo	VVV Venlo	NAC Breda	Roda JC
Roda JC	NAC Breda	Sparta	VVV Venlo
Sparta	FC Twente	FC Twente	Sparta
FC Twente	FC Gronigen	PEC Zwolle	FC Twente

Table 4.8: Prediction of the rankings, Eredivisie 2017-2018, Stage II.

The predictive power along with further comparison between the Bayesian hierarchical model and our EPR model is discussed in Section 4.4. An extensive elaboration and an analysis of the results and weights, $\beta_{p,j}$, are discussed in Section 5.

4.4 Bayesian Hierarchical vs EPR

Now that we have discussed the results of both the Bayesian hierarchical model and our EPR model, we are able to make a comparison between the predictive power of both methods. Recall that the Bayesian hierarchical model had better predictive power than Stage I of the EPR model. We therefore not discuss the predictive power of Stage I, but only Stage II. The comparison between the predictive power of both methods is given in Table 4.9.

Deviations			
Bayesian hierarchical model	Stage II Total score	Stage II Average score	Stage II Score starting 11
24	55	46	11

Table 4.9: Total number of deviations, Bayesian vs Stage II.

In Table 4.9, we see a similar situation as in Table 4.5. The score of the starting eleven is by far the most accurate and in this case, even better than the Bayesian hierarchical model.

We do not observe many deviations compared to the actual ranking. There are three teams that have switched places. These teams are Ajax and PSV, SC Heerenveen and ADO den Haag and Roda JC and VVV Venlo. Furthermore, FC Groningen has been placed significantly higher (+3) than their actual ranking. An explanation can be found by the looking at the total score of FC Groningen. FC Groningen is placed last when it comes to the total score. This can indicate that FC Groningen does not have many players in their selection. An injury can therefore have significant consequences for the results, since the substitute is not used to playing time, or is simply worse than the other player

that is injured. Since we are basing our prediction of the starting eleven, a few injuries during the season can explain why FC Groningen only came in 12th place.

Overall we believe that the predictive power of Stage II is in line with the overall ranking, which indicates that the player rating model can be beneficial for player recruitment.

In Table 4.10 the BIC scores between the Bayesian hierarchical model and Stage II of our EPR model are compared.

BIC score	
Bayesian hierarchical model	EPR Stage II
-457	3150

Table 4.10: Comparing the BIC scores.

Even though the EPR model clearly outperforms the Bayesian hierarchical model in terms of forecasting power, we see that the Bayesian hierarchical model has a far lower BIC compared to the EPR model. We however place more value on the deviations and the prediction of the rankings than the BIC score. The BIC score is relevant to get an indication of the information loss, but the BIC also penalizes model complexity. This may explain the high difference between the scores. Furthermore, the EPR model may have lost a significant amount of information, but this information was found to be insignificant. After removing certain insignificant variables, we see that the BIC score of the EPR model goes down.

For these reasons, we prefer the EPR model over the Bayesian hierarchical model.

Chapter 5: Analyzing the EPR results

In the previous section we made several arguments for MLR being the best fit for our EPR model and to be the favourite compared to the Bayesian Hierarchical model. In this section we look at some results of our EPR model applied to the 2017-2018 Eredivisie dataset. We show what strategies are most effective for teams to win and how to interpret the results for players' strengths and weaknesses.

5.1 Best Strategies

From the $\beta_{p,j}$'s calculated in Equation 3.2, we are able to determine which strategies are most effective for winning. After a few runs, it became clear that most of the passing variables are highly significant according to the model. For example, the total number of passes per game, the number of back passes per game and the number of short middle passes per game were found to be significant. This can easily be explained by the fact that better teams will have more possessions and more chances created. This means that players playing for a better team will automatically have more passes. We choose to discard these quantitative variables and focus more on the qualitative variables, since otherwise it will give players playing for a worse team a disadvantage. Furthermore, since we have a lot of variables per position, we only discuss the most noteworthy insignificant variables along with the most significant.

We start with the goalkeeper. The variables height, age, expected goals against, total shots on goal and the total goals conceded were not found to be significant. Especially the expected goals against and the total goals conceded are two noteworthy variables. An explanation could be that the expected goals against and the total goals conceded are not only dependable on the goalkeeper, but also highly dependent on the defence and on the whole team. Variables that proved to be significant are given in Table 5.1.

Variable	Coefficient
Clean sheets	x
⋮	⋮
y	z

Table 5.1: Most significant variables goalkeeper.

The second position we discuss are the fullbacks. The most significant variables are given in Table 5.2.

Variable	Coefficient
Expected goals	x
⋮	⋮
y	z

Table 5.2: Most significant variables fullbacks.

Apparently, not only defens minded variables, such as defensive duels per game, defensive duels won, aerials duels won and the number of tackles are important. More attack minded variables are also significant according to the model. From Table 5.2, we see that the number of assists, the number of crosses per game and the accuracy of different types of passes are also highly significant. From the model, we thus conclude that better left- and rightbacks also give attacking impulses to a team and not only take care of their defensive work.

The most noteworthy insignificant variable is perhaps the number of interceptions per game. An explanation could be that worse teams are more in the defense, which will cause players playing for a worse team to have more opportunities to make interceptions.

The third position we discuss are the center backs. The most significant variables are given in Table 5.3.

Variable	Coefficient
Expected goals	x
⋮	⋮
y	z

Table 5.3: Most significant variables center backs.

We observe a similar situation as discussed with the fullbacks. Not only defens minded variables, but also more attack minded variables seem to be significant. The most important distinction from the fullbacks is that the number of crosses per game, which was found to be highly significant, is replaced by the number of dribbles per game. This can be explained by the fact that center backs play in a central position, which means that they do not come in a position to deliver crosses. Furthermore, a center back that dribbles towards midfield creates space for his team players, which explains that the number of dribbles and successful dribbles are significant. We thus conclude that better center backs are not only good defensively, but also have excellent passing and dribbling abilities, give attacking impulses to a team and not only take care of their defensive work.

The most noteworthy insignificant variables are the tackle percentage and number of interceptions. A higher tackling percentage should not be perceived as negative, but apparently, a higher tackle percentage has an insignificant effect on score differentials. The number of interceptions could have the same explanation as discussed with the fullbacks.

The fourth position we discuss is defensive midfielder. The most significant variables are given in Table 5.4.

Variable	Coefficient
Expected goals	x
⋮	⋮
y	z

Table 5.4: Most significant variables defensive midfielders.

The most important distinction from the center backs is that the passes accuracy and the expected

assists are found to be highly significant. An explanation for this could be that the defensive midfielders are the link between the midfielders and defense. The defensive midfielders are the first step in creating opportunities and getting the ball from the defensive to their fellow midfielders and attackers. A defensive midfielder that does not have excellent passing abilities may not only cause the team to create fewer opportunities and thus goals, but losing the ball will also create a dangerous counter-attack for the opponent. We thus conclude that better defensive midfielders are not only good defensively, but also have excellent passing abilities, which may lead to assists.

The most noteworthy insignificant variables are the number of tackles won and shots blocked. From Table 5.3 it becomes clear that the number of tackles won and shots blocked are significant variables for the center back. An explanation could thus be that not making a tackle or blocking a shot can be repaired by the center backs.

The fifth position we discuss are the left- and rightmidfielder. The most significant variables are given in Table 5.5.

Variable	Coefficient
Expected goals	x
⋮	⋮
y	z

Table 5.5: Most significant variables left- and rightmidfielders.

Left- and rightmidfielders are generally seen as all-rounders and this thinking seems to correspond with Table 5.5. Left- and rightmidfielders have to be good defensively, score goals, win aerial duels, give assists and have excellent passing abilities. Comparing the left- and rightmidfielders to the defensive midfieler, we see that the left-and rightmidfielders have more attacking minded variables that are significant, where non penalty goals is the most significant variable. We thus conclude that better left- and rightmidfielders are all-rounders.

The most noteworthy insignificant variables are the total number of goals scored and the accuracy of forward passes, through passes and short middle passes. The goals variable should not be perceived as an negative variable, since scoring goals is indeed positive. An explanation for this could be that the effect of scoring goals on score differentials is probably already mostly captured in the expected goals and non penalty goals variable. Furthermore, apparently the accuracy of some passes will have an insignificant effect on score differentials. Again, an explanation for this could be that the effect of several passes on score differentials is probably already mostly captured in the other variables, such as the passes accuracy, final 3rd passes and penalty area passes variables.

The sixth position we discuss is the attacking midfielder. The most significant variables are given in Table 5.6.

Variable	Coefficient
Expected goals	x
⋮	⋮
y	z

Table 5.6: Most significant variables attacking midfielder.

One of the most important distinctions compared to the left- and rightmidfielders is that the passing accuracy is far less significant, but a lot of passing variables are new to Table 5.6. The number of forward passes, smart passes and the accuracy of these passes for example. Furthermore we see that the number of interceptions per game, dribble percentage and successful attacking actions are new variables, where the number of interceptions per game is the most remarkable. This could be explained by the fact that intercepting the ball high up the pitch may lead to a dangerous counter-attack and a potential goal. The last important distinction is that the number of penalty area passes per game is far more significant compared to Table 5.5. A pass to the penalty area can result in a dangerous situation for the opponent. We thus conclude that better attacking midfielders score goals, make interceptions, and give a lot of penalty area passes per game, which may lead to assists.

The most noteworthy insignificant variables is the total number of goals scored. The same explanation can be used as explained for left- and rightmidfielders.

The seventh position we discuss is the wingers. The most significant variables are given in Table 5.7.

Variable	Coefficient
Expected goals	x
⋮	⋮
y	z

Table 5.7: Most significant variables wingers.

In Table 5.7 we see a couple of new variables. The shots percentage, goal conversation rate, crosses per game and average pass length are new variables that we have not seen before with other positions. Especially the number of crosses per game is found to be highly significant. Furthermore, we see that the amount of non penalty goals, passes accuracy, aerial duels won and assists are four other variables that are highly significant, with the amount of non penalty goals scored as clear winner. We thus conclude that better wingers score goals, give assists, win aerial duels and give a lot of crosses.

The most noteworthy insignificant variables are the total number of goals scored, which already has been explained at previous sections.

The last position we discuss is the striker. The most significant variables are given in Table 5.8.

Variable	Coefficient
Goals	x
⋮	⋮
y	z

Table 5.8: Most significant variables strikers.

The results from Table 5.8 are not surprising. Most of the variables are related to scoring goals, where the amount of non penalty goals scored is by far the most significant. The interceptions per game is another variable that we have seen before with the attacking midfielder and the same explanation could be used for the striker as for the attacking midfielder. The most important distinction compared to other positions is the number of touches in the box per game variable. An explanation could be that stronger strikers are able to hold the ball, which may lead to an opportunity. Another explanation could be that strikers who are more calm are able to postpone their shot, which again may lead to a better opportunity. The number of touches in the box per game could be an indication of how strong or how calm a striker is. We thus conclude that better strikers score goals, give assists, win aerial duels, make interceptions and make more touches in the box.

There are no noteworthy insignificant variables for the striker position.

5.2 Best Players

In the following tables are the 5 best and worst players in the Eredivisie season 2017-2018 according to the EPR ranking for each position. The players in these tables have played a minimum of 10 matches.

Best 5		Worst 5	
Player	EPR	Player	EPR
J. Zoet	28,29	J. Brondeel	11,78
M. Bizot	26,77	J. Houwen	12,12
A. Onana	25,48	R. Kortsmit	13,62
H. Jurjus	23,5	T. Wellenreuther	14,38
B. Jones	22,66	B. Castro	14,51

Table 5.9: 5 best and worst goalkeepers.

Best 5		Worst 5	
Player	EPR	Player	EPR
J. Veltman	29,49	T. Goppel	17,16
J. Svensson	29,43	F. Holst	22,96
S. Arias	29,16	V. Karavaev	23,03
N. Tagliafico	28,83	G. Wijnaldum	23,47
D. Zeefuik	28,71	T. David	23,54

Table 5.10: 5 best and worst fullbacks.

Best 5		Worst 5	
Player	EPR	Player	EPR
F. de Jong	19,71	N. Kuipers	12,89
W. Janssen	18,00	Y. van Nieff	13,14
M. Wöber	17,89	D. Bulthuis	14,24
M. de Ligt	17,61	J. Promes	14,28
S. Wuytens	17,43	B. Vriends	14,40

Table 5.11: 5 best and worst center backs.

Best 5		Worst 5	
Player	EPR	Player	EPR
L. Schöne	33,11	D. Haspolat	26,10
K. El Ahmadi	32,22	D. Post	27,11
M. Vejinovic	31,92	C. Colkett	27,38
S. Schaars	30,45	D. Gorter	27,94
J. Hendrix	30,41	D. Bakker	28,50

Table 5.12: 5 best and worst defensive midfielders.

Best 5		Worst 5	
Player	EPR	Player	EPR
D. van de Beek	25,17	N. Rutjes	17,23
M. van Ginkel	24,16	M. Veenhoven	17,25
T. Vilhena	24,10	M. El Makrini	18,70
M. Thorsby	23,49	J. van der Heyden	18,77
J. Monteiro	22,67	O. Velanas	19,01

Table 5.13: 5 best and worst lef- and rightmidfielders.

Best 5		Worst 5	
Player	EPR	Player	EPR
H. Ziyech	24,40	K. Vermeulen	15,74
M. Mount	24,21	B. Vliet	16,51
G. Til	23,46	P. van Amersfoort	19,01
G. Pereiro	23,25	R. Seuntjens	19,09
J. Toornstra	23,18	R. Mühren	19,68

Table 5.14: 5 best and worst attacking midfielders.

Best 5		Worst 5	
Player	EPR	Player	EPR
S. Berghuis	24,31	I. Alhaft	13,53
A. Jahanbakhsh	23,44	U. Antuna	14,07
H. Lozano	23,39	P. Fernandes	14,76
D. Neres	22,86	J. Croux	14,93
B. Linssen	21,81	T. Verhaar	15,05

Table 5.15: 5 best and worst wingers.

Best 5		Worst 5	
Player	EPR	Player	EPR
W. Weghorst	16,91	E. Riis	6,43
B. Johnsen	14,50	Z. El Azzouzi	7,70
L. de Jong	14,20	E. Amenyido	7,80
F. Sol	14,06	L. Castaignos	7,91
K. Huntelaar	13,64	S. Nijland	7,93

Table 5.16: 5 best and worst strikers.

In Section 5.1 we discussed the variables and the coefficients per position. Each position has different significant attributes and coefficients per attribute. The attackers and attacking midfielders have high coefficients with attacking minded variables and the more defensive minded players however have more defensive minded variables with a high coefficient. Many player rating models from journalists, but also Scisports, claim that they are able to determine the best player in a certain league. This does not make sense. For example, the number of tackles per game is important for the left- and rightback position. When comparing this variable to the number of assists, which is significant for the attackers, it is clear that the number of tackles per game is greater than the number of assists. We thus see that the more defensive minded players obtain a higher score. It is thus not possible to compare players in different positions in our opinion.

Comparing our best players per position with the team of the season of Voetbal International [32], we see a lot of interesting results and especially the results of the center backs is surprising. M. de Ligt is often found to be the best center back in the last year by highly esteemed journalist and regular watchers of the game. However, we see M. de Ligt only coming in 4th place in our EPR model. This can be explained by the fact that there are two types of positions in the center back. There is a player who constructs the build up, and there is a player that marks the striker, who is called the marker. No distinction is made between the two in our dataset. From Table 5.3, we see that the most significant attributes for the center backs are a mix between passing, dribbling and defensive abilities. Comparing the passing and dribbling variables with the defensive variables, we see that most of the passing and dribbling variables are greater than the defensive variables. In other words, players who construct the build will be placed higher in the ranking than markers. M. de Ligt is the first real marker who shows up in the top 5 of best center backs, which means he was the best marker in the Eredivisie last season. A limitation of this model is thus that we have to manually check which type the center back is.

Other surprising results are that of the wingers, striker and fullbacks, but we will take a closer look at the fullbacks. Three fullbacks that were often praised last season were D. Dumfries, Angeliño and S. Arias, whereby S. Arias was in the team of the season of Voetbal International [32]. The winner of our EPR model however is J. Veltman, a player who was often criticised. The ERP rating of J. Veltman and S. Arias is relatively close, but it is surprising to find out that D. Dumfries and Angeliño are respectively placed 28th and 49th. In order to get an understanding why J. Veltman is placed number one, we take a look at what J. Veltman's strengths and weaknesses and compare them to those of S. Arias in Table 5.17. The EPR is broken down according to Equation 3.4. Note that both players are fullbacks, which means that all numbers for the variables corresponding with the fullback position are the result of combination of his own output and his defensive skill on his direct opponent. Note again, that adding all elements in this breakdown will result in his original EPR.

Player	$Attribute_1$	$Attribute_2$	$Attribute_3$	$Attribute_4$	$Attribute_5$	$Attribute_6$	$Attribute_7$	$Attribute_8$	$Attribute_9$
J. Veltman	?	?	?	?	?	?	?	?	?
S. Arias	?	?	?	?	?	?	?	?	?

	$Attribute_{10}$	$Attribute_{11}$	$Attribute_{12}$	$Attribute_{13}$	$Attribute_{14}$	$Attribute_{15}$	$Attribute_{16}$	$Attribute_{17}$	$Attribute_{18}$	$Attribute_{19}$
J. Veltman	?	?	?	?	?	?	?	?	?	?
S. Arias	?	?	?	?	?	?	?	?	?	?

Table 5.17: Breakdown of J. Veltman and S. Arias EPR.

From Table 5.17 it becomes clear that most of the variables are relatively close. Nonetheless, it seems that J. Veltman is better defensively and is a better passer. S. Arias has scored more goals, provides more assists and has more crosses per game, which are also highly significant variables according to Table 5.2. We thus conclude that S. Arias is better offensively, but combining every variable, we see that S. Arias just comes short when comparing the EPR of both players. Valuing S. Arias better than J. Veltman may seem to indicate biased opinions based on the goals and assist of players. The offensive abilities seem to be overvalued and the defensive abilities seem to be undervalued.

We have to note that a drawback of the EPR method is that we are unable to tell what the mix of offensive and defensive efforts is that result in a player's contributions. It is therefore important to discuss the results with scouts and other experts to correctly identify a player's contributions before buying a new player.

5.3 Shortcomings Heracles

Now that we have formulated our EPR model and discussed the best strategies, we are able to determine the shortcomings of Heracles last season and begin to understand what is needed in order to achieve a place in the Top 8. Not only will the Top 8 will guarantee the play-offs for European football, but looking back at Table 4.7, we see that the score of Heracles and the number 8 is relatively close, making the Top 8 an achievable goal. The target score of Heracles last season should have been 216,82, whereas their actual score was 209,61. We look at the EPR of the starting 11 of Heracles along with the norm of each position. The norm is based on the EPR score of the player represented by the 8th club on the rankings. Note that this is different from the 8th best player of the rankings per position. Better teams like Ajax and PSV are likely to have better substitutes than most of the starting 11 players of other teams. Since these clubs can only start with one player in every position, the number 8th club is the norm. The EPR's of the starting 11 of Heracles along with the norm are given in Table 5.18.

Player	Position	EPR	Norm	Difference	Ratio Norm/EPR
B. Castro	Goalkeeper	14,51	21,29	6,78	1,46
R. Baas	Leftback	25,10	27,47	2,37	1,09
R. Propper	Center back	16,76	16,96	0,20	1,01
D. Wuytens	Center back	15,99	16,96	0,97	1,06
T. Breukers	Rightback	25,61	27,65	2,04	1,08
P. van Ooijen	Rightmidfielder	19,65	22,20	2,55	1,13
S. Jakubiak	Central midfielder	21,38	22,20	0,82	1,04
J. Monteiro	Leftmidfielder	22,67	22,20	-0,47	0,98
K. Peterson	Left winger	18,75	19,10	0,35	1,02
V. Vermeij	Striker	11,64	13,05	1,41	1,12
B. Kuwas	Right winger	17,56	18,84	1,28	1,07
Total		209,61	227,92	18,31	

Table 5.18: EPR's of Heracles players vs norm.

From Table 5.18, it becomes clear that the goalkeeper, the fullbacks, the rightmidfielder and the striker were the bottlenecks last season. The striker, V. Vermeij, has a difference of 1.41 compared to the norm, which seems reasonable. However, the striker has a lower EPR compared to other positions. Taking this into account and after calculating the ratio between the norm and EPR, we can conclude that also the striker is far away from the norm. Furthermore, note that the total score of the norm is higher than the total score needed last season in order to secure a place in the Top 8th. This can be explained due the fact that it is irregular for the number 8th club to have the 8th best player in every position, which apparently leads to a higher score.

From Table 5.18 we conclude that five positions were clearly far off the norm in order to achieve a spot in the Top 8. Improving the goalkeeper to the norm value is expected to have a huge impact on the team, making the total EPR score of Heracles almost equal to the target score. Another improvement, regardless of this being one of the fullbacks, midfielder or striker will result in achieving the target score for Heracles.

Various changes have already happened to Heracles and other teams. It is therefore of great importance to continuously update the norm and EPR's of the players. The goalkeeper and the leftback of Heracles have already left the club and several new players have joined the club. The evaluation of these new signings along with advice of other potential signings will be discussed in Section 6.

Chapter 6: Market value analysis

Now that we have formed our EPR model, there still remains the question how much market value players are really worth. It is very important to put an accurate estimate of a player's market value, since this market value will have direct impact regarding the decisions of player recruitment for Heracles. To gain a competitive edge over opponents, Heracles has to make sure they do not overpay for players. In this section we try to find a method which rewards each player a fair market value. This will be done for the Eredivisie, Jupilerleague, Ligue 2 and the second- and third Bundesliga, since the players playing in these competitions are the potentially affordable players for Heracles. Afterwards, we analyze which players are the most over- and undervalued for each competition and form an advice regarding potential improvements for Heracles. All the data regarding the market value of players is taken from https://www.transfermarkt.com/.

6.1 Fair market value Eredivisie

We start with determining the fair market value of the Eredivisie. In Figure 6.1 we ranked the market values for all Eredivisie players from the 2017-2018 season. We have to note that this market value is an estimation and the actual market value is often defined by the demand for a certain player.

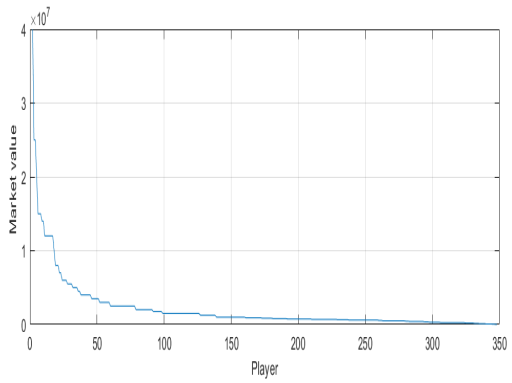

Figure 6.1: Market value of Eredivisie players 2017-2018

From the figure it becomes clear that the teams do not reward players proportionally to their rank. The market value seems to scale exponentially with the rank of players. This corresponds with the thought that the league is driven by star players. This means that a select few players dominate the rest of the league and their market value is awarded accordingly.

Furthermore, note that the observed market values in Figure 6.1 do not follow a completely smooth curve, which is an indication that some players are still slightly over-or undervalued. These deviations can be explained by various factors, such as image in the media, negotiating skills and experience. Our goal is to find a player's market value, without accounting for these various factors. If the players are ranked correctly according to skill, their market values should follow their EPR skill level. Therefore, we have plotted the market values corresponding to the EPR skill ranking in Figure 6.2.

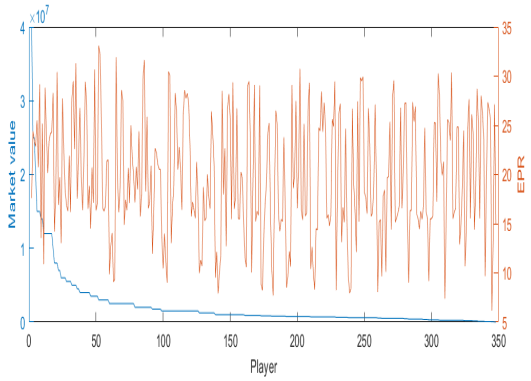

Figure 6.2: EPR vs market value, Eredivisie,2017-2018.

From Figure 6.2 we see that the market values generally do not follow the EPR skill level, which makes the figure look chaotic. Singling out one position, we observe the same.

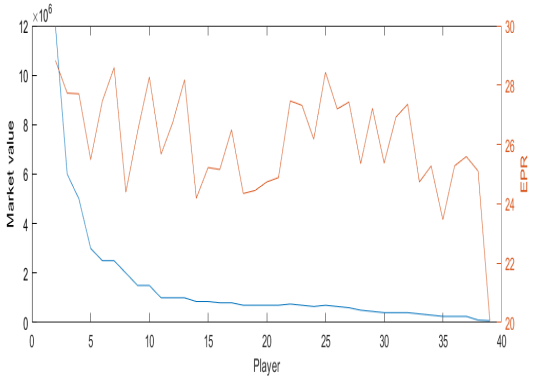

Figure 6.3: EPR leftbacks vs market value, Eredivisie,2017-2018.

There does seem to be a slight declining trend in market value as the players get less skillful, as we would expect, but it is safe to say that most players are either very over- or undervalued, since no correlation can be found.

In the following tables, we have listed the top 5 over- and undervalued players according to the EPR skill ranking for each position.

Top 5 overvalued goalkeepers			
Player	EPR	Price per skillpoint	Amount overvalued
A. Onana	25,48	€588.687	€13.038.644
J. Zoet	28,29	€353.463	€7.882.262
S. Padt	19,73	€202.790	€2.481.679
M. Bizot	26,77	€149.430	€1.939.503
D. Jensen	19,59	€127.630	€992.234

Top 5 undervalued goalkeepers			
Player	EPR	Price per skillpoint	Amount undervalued
D. van Crooij	24,23	€6.183	€1.717.196
R. Zwinkels	21,45	€11.652	€1.401.469
W. van der Steen	21,29	€11.742	€1.388.869
T. Zwarthoed	19,91	€12.566	€1.282.621
R. Pasveer	19,78	€25.273	€1.022.845

Table 6.1: Top 5 over- and undervalued goalkeepers according to the EPR skill ranking.

Top 5 overvalued fullbacks			
Player	EPR	Price per skillpoint	Amount overvalued
S. Arias	29,16	€514.381	€13.472.393
N. Tagliafico	28,83	€416.206	€10.489.647
R. Haps	27,73	€216.362	€4.547.303
J. Brenet	27,71	€180.457	€3.548.556
J. Veltman	29,49	€169.555	€3.455.227

Top 5 undervalued fullbacks			
Player	EPR	Price per skillpoint	Amount undervalued
T. Van de Berg	27,33	€2.744	€1.356.763
D. Zeefuik	28,71	€5.224	€1.353.970
F. Sporkslede	26,62	€2.817	€1.319.432
H. Asmelash	26,31	€3.801	€1.278.099
E. Korkmaz	25,54	€2.936	€1.263.025

Table 6.2: Top 5 over- and undervalued fullbacks according to the EPR skill ranking.

Top 5 overvalued center backs			
Player	EPR	Price per skillpoint	Amount overvalued
M. de Ligt	17,61	€2.271.384	€37.654.177
M. Wöber	17,89	€447.077	€5.616.399
F. de Jong	19,71	€355.098	€4.374.115
S. van Beek	16,19	€339.830	€3.344.112
J. St. Juste	16,28	€337.769	€3.330.954

Top 5 undervalued center backs			
Player	EPR	Price per skillpoint	Amount undervalued
B. Meijers	14,98	€1.668	€1.970.680
D. Werker	15,54	€6.435	€1.969.927
M. Breuer	15,53	€12.874	€1.869.242
T. Oude Kotte	14,39	€10.421	€1.767.285
D. Van den Buijs	15,97	€25.041	€1.727.754

Table 6.3: Top 5 over- and undervalued center backs according to the EPR skill ranking.

Top 5 overvalued defensive midfielders			
Player	EPR	Price per skillpoint	Amount overvalued
J. Hendrix	30,41	€263.039	€6.615.438
F. Midtsjo	31,32	€159.636	€3.574.128
T. Koopmeiners	30,69	€114.059	€2.103.051
K. El Ahmadi	32,22	€93.107	€1.533.164
L. Schöne	33,11	€90.619	€1.492.894

Top 5 undervalued defensive midfielders			
Player	EPR	Price per skillpoint	Amount undervalued
J. Bruijn	27,13	€921	€1.210.142
E. Lieftink	28,02	€3.569	€1.175.441
G. Nijholt	30,36	€8.234	€1.132.150
D. Gorter	27,94	€8.948	€1.021.782
D. Haspolat	26,10	€7.662	€988.185

Table 6.4: Top 5 over- and undervalued defensive midfielders according to the EPR skill ranking.

Top 5 overvalued left- and right midfielders			
Player	EPR	Price per skillpoint	Amount overvalued
D. van de Beek	25,17	€556.164	€10.577.154
T. Vilhena	24,10	€497.887	€8.703.925
M. van Ginkel	24,16	€496.688	€8.695.968
B. Ramselaar	22,20	€270.234	€2.963.613
Y. Ayoub	21,91	€251.036	€2.503.781

Top 5 undervalued left- and rightmidfielders			
Player	EPR	Price per skillpoint	Amount undervalued
S. Jakubiak	21,38	€14.031	€2.624.016
R. Sanusi	20,39	€24.529	€2.287.575
P. Van Ooijen	19,65	€25.439	€2.187.855
H. Vučkič	20,67	€36.285	€2.076.708
N. Rutjes	17,23	€17.408	€2.056.682

Table 6.5: Top 5 over- and undervalued left- and rightmidfielders according to the EPR skill ranking.

Top 5 overvalued attacking midfielders			
Player	EPR	Price per skillpoint	Amount overvalued
H. Ziyech	24,40	€1.024.539	€21.992.910
G. Pereiro	23,25	€430.166	€7.135.175
J. Toornstra	23,18	€258.896	€3.143.987
G. Til	23,46	€255.753	€3.108.886
M. Mount	24,21	€165.234	€1.016.716

Top 5 undervalued attacking midfielders			
Player	EPR	Price per skillpoint	Amount undervalued
M. Osman	21,58	€13.904	€2.358.922
T. Agyepong	19,80	€12.628	€2.189.679
P. van Moorsel	20,25	€19.753	€2.095.484
A. Messaoud	21,47	€27.940	€2.046.368
R. Vloet	21,08	€33.205	€1.897.907

Table 6.6: Top 5 over- and undervalued attacking midfielders according to the EPR skill ranking.

Top 5 overvalued wingers			
Player	EPR	Price per skillpoint	Amount overvalued
H. Lozano	23,39	€1.068.652	€22.204.064
David Neres	22,86	€874.821	€17.023.460
J. Kluivert	20,75	€722.888	€12.298.398
S. Bergwijn	20,15	€595.515	€9.376.451
A. Jahanbakhsh	23,44	€512.006	€8.948.548

Top 5 undervalued wingers			
Player	EPR	Price per skillpoint	Amount undervalued
S. Spierings	18,57	€8.077	€2.267.785
J. Lelieveld	18,63	€13.419	€2.175.455
R. Castelen	16,57	€15.088	€1.907.268
D. George	16,18	€15.466	€1.857.207
D. Malen	19,99	€37.517	€1.852.718

Table 6.7: Top 5 over- and undervalued wingers according to the EPR skill ranking.

Top 5 overvalued strikers			
Player	EPR	Price per skillpoint	Amount overvalued
N. Jørgensen	13,55	€1.033.447	€11.587.649
K. Dolberg	10,88	€1.102.733	€10.062.189
L. de Jong	14,20	€563.386	€5.471.376
W. Weghorst	16,91	€413.975	€3.988.905
Z. Labyad	13,05	€459.940	€3.676.994

Top 5 undervalued strikers			
Player	EPR	Price per skillpoint	Amount undervalued
D. Schahin	12,62	€55.461	€1.547.520
B. Ogbeche	12,20	€61.498	€1.421.696
R. Ache	9,17	€32.693	€1.334.054
N. Proschwitz	9,66	€51.725	€1.220.446
M. Kvasina	9,01	€44.419	€1.203.570

Table 6.8: Top 5 over- and undervalued strikers according to the EPR skill ranking.

The most undervalued player in the Eredivisie is S. Jakubiak, who should be worth €2.624.016 more. Not only was his market value determined to be one of the lowest by transfermarkt.com, his EPR placed him quite high on the middle-ranking of midfielders.

The most overvalued player was M. de Ligt, who should be worth €37.654.177 less. There are several reasons for this huge difference. The first one is that M. de Ligt plays for Ajax. Ajax is a top- and rich team in the Eredivisie and has a huge name when it comes to delivering talents. This makes it possible for Ajax to ask a significant amount for M. de Ligt. Another reason is that M. de Ligt has already been captain of Ajax and made his debut for the Dutch national team, while being just 18 years old. M. de Ligt may not be a world-class defender yet, but he could become one. Teams are thus paying for his potential.

Now that the fair market value of the Eredivisie has been established, we are able to determine which players Heracles should buy. Note that the most undervalued players in the Eredivisie are not necessarily improvements for Heracles, since these players could also have an EPR below the norm. We filter the players to have an EPR above the norm and a market value of maximum €1.000.000, making the player not too expensive for Heracles. In Table 6.9 we summarize potential improvements for Heracles. The age of the player and total matches played last season have been added to Table

6.9 in order for Heracles to make a well considered decision. Recall that the goalkeeper, fullbacks, right- or leftmidfielder and the striker were the positions that should be prioritized by Heracles in finding new players. Absence of certain positions in Table 6.9 indicates that there are no players with the minimum required EPR in that position or are to expensive.

Potential improvements						
Player	Position	Age	Matches played	EPR	Actual market value	Amount undervalued
Anonymous	Goalkeeper	Anonymous	Anonymous	24,26	€150.000	€1.717.196
Anonymous	Goalkeeper	Anonymous	Anonymous	23,50	€900.000	€908.932
Anonymous	Goalkeeper	Anonymous	Anonymous	21,46	€250.000	€1.401.469
Anonymous	Goalkeeper	Anonymous	Anonymous	21,29	€250.000	€1.388.869
Anonymous	Rightback	Anonymous	Anonymous	28,71	€150.000	€1.353.970
Anonymous	Leftback	Anonymous	Anonymous	28,18	€1.000.000	€476.268
Anonymous	Right- and leftback	Anonymous	Anonymous	27,66	€200.000	€1.248.791

Table 6.9: Recommendations to Heracles Eredivisie.

6.2 Fair market values other leagues

The fair market valuation of other leagues is determined in similar fashion. We will not discuss these other leagues to the same extend, but just resort to giving our final recommendations for these competitions. An addition for these recommendations will be the adjusted EPR, where the initial EPR of a player has been altered to the strength of the Eredivisie. This is necessary in order to properly determine whether or not players are also good enough for a top 8th ranking in the Eredivisie. Information about the strength of each competition is obtained through Heracles and is based on the skill of the champion, teams in the mid-ranking and relegation teams.

We noticed that not many players passed the target for the adjusted EPR in the other competitions, leaving only a recommendation of a couple players. This is due the fact the competition strength of some competitions is considerable less than the Eredivisie, making it difficult for the adjusted EPR to stay above the target value. However, every player reacts differently to a transfer and competition strength of 0,75 does not necessarily mean that a player will perform 0,75 less. It is therefore important to discuss the results with scouts and other experts to correctly identify whether or not that player is still good enough for the ambitions of Heracles.

In order to still present Heracles with an overview of the best players per competition for their bottlenecks, we choose to include several players who's adjusted EPR did not pass the target score. We however still urge Heracles to account for the adjusted EPR.

Fair market value Jupilerleague

Potential improvements based on the fair market value are given in Table 6.10.

Potential improvements							
Player	Position	Age	Matches played	EPR	Adjusted EPR	Actual market value	Amount undervalued
Anonymous	Goalkeeper	Anonymous	Anonymous	25,82	19,62	€200.000	€192.862
Anonymous	Goalkeeper	Anonymous	Anonymous	25,37	19,28	€800.000	€-413.948
Anonymous	Goalkeeper	Anonymous	Anonymous	25,21	19,16	€250.000	€133.585
Anonymous	Goalkeeper	Anonymous	Anonymous	25,12	19,10	€200.000	€182.302
Anonymous	Leftback	Anonymous	Anonymous	28,96	22,01	-	-
Anonymous	Rightback	Anonymous	Anonymous	28,83	21,91	€200.000	€-19.438
Anonymous	Rightback	Anonymous	Anonymous	28,62	21,75	€150.000	€29.221
Anonymous	Rightback	Anonymous	Anonymous	28,34	21,56	€125.000	€52.479
Anonymous	Rightmidfielder	Anonymous	Anonymous	24,28	18,45	€350.000	€-84.914
Anonymous	Rightmidfielder	Anonymous	Anonymous	23,83	18,11	€300.000	€-39.791
Anonymous	Left-or rightmidfielder	Anonymous	Anonymous	23,64	17,97	-	-
Anonymous	Defensive-or rightmidfielder	Anonymous	Anonymous	23,10	17,56	€1.250.000	€-997.770
Anonymous	Striker	Anonymous	Anonymous	17,74	13,48	€350.000	€36.369
Anonymous	Striker	Anonymous	Anonymous	15,10	11,58	€300.000	€28.913
Anonymous	Striker	Anonymous	Anonymous	13,74	10,44	€400.000	€-100.717
Anonymous	Striker	Anonymous	Anonymous	13,55	10,30	€175.000	€120.171

Table 6.10: Recommendations to Heracles Jupilerleague.

From Table 6.10 we see that the initial EPR of many players is excellent. However, there are not many players whose adjusted EPR still exceeds the target value. This is due the fact that the competition strength is only 0,76, making only Anonymous able to stay above the target value. Whether or not other players in Table 6.10 are improvements for Heracles have to be discussed with scouts and other experts.

Fair market value Ligue 2

Potential improvements based on the fair market value are given in Table 6.11

Potential improvements							
Player	Position	Age	Matches played	Competition strength	Adjusted EPR	Actual market value	Amount undervalued
Anonymous	Goalkeeper	Anonymous	Anonymous	26,41	21,76	€150.000	€482.539
Anonymous	Goalkeeper	Anonymous	Anonymous	25,72	21,19	€150.000	€465.833
Anonymous	Goalkeeper	Anonymous	Anonymous	25,71	21,19	-	-
Anonymous	Rightback	Anonymous	Anonymous	27,68	22,81	-	-
Anonymous	Leftback	Anonymous	Anonymous	27,53	22,69	€800.000	€-317.866
Anonymous	Leftmidfielder	Anonymous	Anonymous	22,36	18,42	€800.000	€-230.285
Anonymous	Leftmidfielder	Anonymous	Anonymous	22,30	18,38	€800.000	€-231.680
Anonymous	Striker	Anonymous	Anonymous	13,54	11,15	€900.000	€-80.331
Anonymous	Striker	Anonymous	Anonymous	13,00	10,71	€600.000	€186.967

Table 6.11: Recommendations to Heracles Ligue 2.

From Table 6.11 we see that again, there are few players whose adjusted EPR exceed the target value. This is partly caused by the fact that 0,824 was calculated to be the competition strength. However, we see that a new goalkeeper from the Ligue 2 looks promising. All three goalkeepers have an adjusted EPR around the target value and have market values that Heracles can afford. The two goalkeepers Anonymous and Anonymous are slightly under the target value, but a lot younger than Anonymous , making their potential higher.

The goalkeeper is the only position we recommend in the Ligue 2. We do not propose a new fullback, midfielder or a new striker from this league. Players who exceeded the target value were either too

expensive or non-existent.

Fair market value 2nd Bundesliga

Potential improvements based on the fair market value are given in the table below. Our recommendation is given in green.

Potential improvements							
Player	Position	Age	Matches played	EPR	Adjusted EPR	Actual market value	Amount undervalued
Anonymous	Goalkeeper	Anonymous	Anonymous	30,47	29,04	€800.000	€123.695
Anonymous	Goalkeeper	Anonymous	Anonymous	26,44	25,20	€900.000	€-98.465
Anonymous	Goalkeeper	Anonymous	Anonymous	26,07	24,84	€1.000.000	€-209.832
Anonymous	Goalkeeper	Anonymous	Anonymous	24,04	22,91	€150.000	€578.740
Anonymous	Goalkeeper	Anonymous	Anonymous	22,39	21,34	€400.000	€278.747
Anonymous	Goalkeeper	Anonymous	Anonymous	22,39	21,34	€300.000	€378.571
Anonymous	Goalkeeper	Anonymous	Anonymous	21,36	20,36	€900.000	€-252.430
Anonymous	Leftback	Anonymous	Anonymous	27,43	26,14	€250.000	€519.235
Anonymous	Rightback	Anonymous	Anonymous	27,05	25,78	€500.000	€258.481
Anonymous	Rightmidfielder	Anonymous	Anonymous	22,90	21,83	€700.000	€188.777
Anonymous	Striker	Anonymous	Anonymous	13,34	12,71	€350.000	€659.473
Anonymous	Striker	Anonymous	Anonymous	11,98	11,41	-	-

Table 6.12: Recommendations to Heracles 2nd Bundesliga.

From Table 6.12 we see that there are a few players whose adjusted EPR exceed the target value. The strength of the competition is calculated to be 0,953, which explains the small difference between the EPR and the adjusted EPR.

Recruiting a new goalkeeper from the 2rd Bundesliga looks promising. There are six goalkeepers who have an adjusted EPR above the target value and especially the adjusted EPR of Anonymous looks impressive. His adjusted EPR of 29,04 will place him above J. Zoet in the Eredivisie, who is found to be the best goalkeeper last season and has a market value of €10.000.000. €800.000 thus looks like a steal for Anonymous. The cheaper alternative will be Anonymous, a goalkeeper who is only 22, has an adjusted EPR of 22,91, and has an market value of €150.000.

Further investigation has to be done for Anonymous and Anonymous. These players have an adjusted EPR slightly below the target value, but are still found to be direct improvements for the starting 11 of Heracles. Furthermore, they are considered cheap. Anonymous has an adjusted EPR lower than Anonymous, but he is only 18 years old and played 31 games last season. Making him a player with high potential.

We do not propose a new fullback or midfielder from this league. Players in these position who exceeded the target value were either to expensive or non-existent.

Fair market value 3. Liga

Potential improvements based on the fair market value are given in the table below.

Potential improvements							
Player	Position	Age	Matches played	EPR	Adjusted EPR	Actual market value	Amount undervalued
Anonymous	Goalkeeper	Anonymous	Anonymous	29,88	24,33	€300.000	€1.629
Anonymous	Goalkeeper	Anonymous	Anonymous	29,35	23,89	€275.000	€62.629
Anonymous	Goalkeeper	Anonymous	Anonymous	28,08	22,86	€150.000	€151.629
Anonymous	Goalkeeper	Anonymous	Anonymous	27,58	22,45	€300.000	€1.629
Anonymous	Goalkeeper	Anonymous	Anonymous	27,01	21,99	€275.000	€26.629
Anonymous	Leftback	Anonymous	Anonymous	28,49	23,19	€250.000	€22.590
Anonymous	Leftback	Anonymous	Anonymous	27,16	22,11	€1.000.000	€-740.079
Anonymous	Leftback	Anonymous	Anonymous	27,10	22,06	€400.000	€-140.656
Anonymous	Leftback	Anonymous	Anonymous	26,86	21,87	€250.000	€7.047
Anonymous	Striker	Anonymous	Anonymous	14,73	11,99	€550.000	€-159.319
Anonymous	Striker	Anonymous	Anonymous	14,70	11,97	-	-
Anonymous	Striker	Anonymous	Anonymous	14,68	11,95	€450.000	€-60.667
Anonymous	Striker	Anonymous	Anonymous	14,39	11,71	€425.000	€-43.493

Table 6.13: Recommendations to Heracles 3. Liga.

From Table 6.12 we see that again a new potential goalkeeper looks promising. Anonymous and Anonymous are the two best goalkeepers in the league and are affordable.

The fullbacks and strikers have again high initial EPR's, but adjusted EPR's below the target value. These players have to be discussed with scouts and other experts. The strength of the competition is calculated to be 0,814.

We do not propose a midfielder from this competition. Players in the midfield position who exceeded the target value were either too expensive or non-existent.

6.3 Evaluation of new signings

Now that the fair market values have been determined, we are able to evaluate the new signings of Heracles. The new signings are Adrián Dalmau (Villarreal), Silvester van der Water (Almere City), Janis Blaswich (Hansa Rostock), Maximilian Rossmann (Sportfreunde Lotte), Tarik Kada (Eindhoven), Zeki Erkilinc (FC Twente), Yoëll van Nieff (FC Groningen) and Joey Konings (PSV). Unfortunalty, no data were available for Zeki Erkilinc. The EPR's of the other signings can be found in Table 6.14.

New signings							
Player	Position	Age	Matches played	EPR	Adjusted EPR	Actual market value	Amount undervalued
J. Blaswich	Goalkeeper	27	35	27,58	22,45	€300.000	€1.629
Yoëll van Nieff	Center back	25	21	13,14	13,14	€650.000	€1.100.080
M. Rossmann	Center back	23	19	14,08	11,46	€100.000	€160.007
S. Sama	Center back	25	14	15,43	14,70	€200.000	€512.438
S. van der Water	Rightwinger	21	32	20,66	15,70	€200.000	€233.697
T. Kada	Rightwinger	22	19	13,03	9,90	€150.000	€123.538
A. Merkel	Attacking midfielder	26	17	20,31	18,53	€350.000	€1.085.109
J. Konings	Striker	20	16	7,99	6,07	€25.000	€149.023
A. Dalmau	Striker	24	40	11,04	8,75	€200.000	€57.059

Table 6.14: Evaluation of the new signings.

Janis Blaswich is the only player who's adjusted EPR exceeds the target value. Note that Janis Blaswich was also recommended in Table 6.13 when the 3. Liga was discussed. Silvester van der Water is another singing who's EPR looks promising. His initial EPR of 20,66 is higher than the EPR's of K. Peterson and B. Kuwas, the current wingers of Heracles. How this player will adapt to the strength of the Eredivisie has to be seen. The other signings have EPR's below the current starting players of Heracles. However, these players could always develop.

Chapter 7: Conclusions

In this thesis we have proposed a new method to rate football players. The purpose of this new method was to deal with some of the limitations of existing player rating methods. The existing methods and MVP awards of journalists overvalue offensive skills. This is because statistics are being used in some way or capacity, which are unable to correctly capture the defensive skill of a player. Furthermore, the current methods are too one-dimensional. The current methods are able to determine how good a player is, but not what his strengths and weaknesses are.

Our Exact Player Rating (EPR) improves upon these limitations by also estimating what strategies are effective for winning and what the strengths and weaknesses are of each player. Furthermore, data has been used in order to fully capture player's defensive capabilities.

Our EPR method consists of a two-stage regression. The first stage models the influence of players on several production statistics. The second stage is a regression of score differentials on several production statistics such as goals, assists, shots and interceptions. In other words, in the second stage we model score differentials with the estimated production statistics from the first stage as explainable variables. For these production statistics, the difference between the production output of these statistics of the home and away team are taken. Furthermore, the distinction between the various possible positions of football players is made for all production statistics. The results of the second-stage regression will allow us to say which tactics are effective to win.

We used match data from the 2017-2018 Eredivisie season in order to compare our EPR method with the Bayesian hierarchical model made by G. Baio et al [4]. These methods have been compared in terms of forecasting accuracy and BIC score. The EPR performed better in terms of forecast accuracy and the results came close to the actual ranking. The EPR however performed worse when it comes to the BIC score. We have placed more value on the forecast accuracy. Among our several EPR methods, we found that Multiple Linear Regression was the best fit for the data in the second stage, outperforming a simplified model and several machine learning models, such as XGBoost, random forest and lasso regression.

We have compared player rankings of our EPR method with other player rating models, such as the team of the season of Voetbal International. A place in the team of the season is awarded by highly esteemed journalists. We noticed that Voetbal international rank many of the players similarly. Many of the offensive players have been awarded. This makes sense, since offensive skills will stand out more to the experts than defensive skills. However, it seems like the journalists have overvalued players with good offensive and undervalued defensive skills.

We have used the EPR method to analyze what strategies are effective for winning and found some interesting results. The fullbacks should for example also focus on more attacking minded variables, such as the number of assists, number of crosses and the accuracy of different types of passes. Significant variables for the left- and rightmidfielder are the number of defensive duels per game, the percentage of defensive duels won and the percentage of aerial duels won. The attacking midfielder and and the striker should also focus on interceptions.

We have used the EPR method in order to determine the shortcomings of Heracles last season. We discussed the target score, the norm for each position and the EPR score of the starting 11 players. We found that the goalkeeper, the fullbacks, the rightmidfielder and the striker were the bottlenecks last season. Improving the goalkeeper along with another improvement, regardless the position, will result in the target score for Heracles. Various changers however have already happened to the team of Heracles and other teams. It is therefore of great importance to continuously update the norm and EPR's of players.

Finally, we tried to come up with a fair market value and find out which players were over- and undervalued according to the EPR method in order to propose possible improvements for Heracles. This was done by first ranking all market values from high to low and subsequently fitting our EPR skill level through this data. We found that some players were severely over- or undervalued. This can be explained by various factors, but again, we believe this is hugely a result from a poor judgement of offensive and defensive skill. Also the new signings of Heracles have been evaluated.

Overall we found that our EPR method is a good addition to current player rating models and the current literature. It improves upon some of the limitations that the current methods have, namely that they overvalue offensive skills and undervalue defensive skills. Furthermore, the EPR method provides more useful information besides a mere player rating. It provides teams insight in game winning strategies, provides teams with advice regarding the buying and selling of players and our EPR method has proven to be better compared to the current best methods when it comes to predictive accuracy.

7.1 Suggestions for further research

In this thesis we compared our model with the Bayesian hierarchical model made by G. Baio et al [4]. Although the prediction of the rankings and the BIC score can be compared, it is difficult to compare other results, such as the accuracy and cross validation score. Furthermore, the Bayesian hierarchical model is a team rating model and not a player rating model. In order to properly compare our EPR model with another player rating model, it is preferred to get our hands on the models of Scisorts and Remiqz. This however could be difficult to achieve.

Furthermore, lasso regression puts constraints on the size of the coefficients associated to each variable. However, this value will depend on the magnitude of each variable. It is therefore necessary to standardize the variables. This applies equally to ridge regression. This is not done for both regressions. Another suggestion for further research is thus to standardize the variables in our data, which may lead to better results for lasso- and ridge regression.

Lastly, data of only one season is used and we did not incorporate the variance of certain aspects. For example, we did not calculate the variance of the explanatory variables per game or the variance of the EPR's of players per game. This could be done, which may help in calculating the growth of a player or how stable a player is throughout the season or seasons. In other words, more data is needed and preferred for risk management and further research.

Bibliography

[1] “Website for accurate market and budget data for football teams” [Online]. Available: https://www.transfermarkt.de/

[2] Michael Lewis. Moneyball: "The art of winning an unfair game". WW Norton & Company, 2004.

[3] Sean Ingle. How Midtjylland took the analytical route towards the Champions League, 2015 [Online]. Available: https://www.theguardian.com/football/2015/jul/27/how-fc-midtjylland-analytical-route-champions-league-brentford-matthew-benham

[4] Gianluca Baio and Marta A. Blangiardo. Bayesian hierarchical model for the prediction of football results. Journal of Applied Statistics , 37 (2) 253 - 264, 2010.

[5] John Hollinger. Pro Basketball Prospectus: 2003 EDITION (Pro Basketball Forecast). Brassey’s, Washington DC, 2003.

[6] “Website for the explanation of their player efficiency ratings in football” [Online]. Available: https://www.whoscored.com/Explanations

[7] David J. Berri. A simple measure of worker productivity in the national basketball association. The Business of sport, 3:1-40, 2008.

[8] David J. Berri and Martin B. Schmidt. Stumbling On Wins in Basketball. Pearson Education, 2010.

[9] Justin Kubatko, Dean Oliver, Kevin Pelton, and Dan T. Rosenbaum. A starting point for analyzing basketball statistics. Journal of Quantitative Analysis in Sports, 3.3, 2007.

[10] S. R. Bray and L. R. Brawley. Role efficacy, role clarity, and role performance effectiveness. Small Group Research, 33:233-253, 2002.

[11] Garritt L. Page, Gilbert W. Fellingham, and C. Shane Reese. Using box-scores to determine a positions contribution to winning basketball games. Journal of Quantitative Analysis in Sports, 3.4, 2007.

[12] Dan T. Rosenbaum. Measuring how NBA players help their teams win, 2004. [Online]. Available: http://www.82games.com/comm30.htm.

[13] Howard Hamilton. Adjusted Plus/Minus in football. Why it’s hard, and why it’s probably useless, 2014. [Online]. Available: http://www.soccermetrics.net/player-performance/adjusted-plus-minus-deep-analysis

[14] Dapo Omidiran. Penalized regression models for the nba. arXiv preprint arXiv:1301.3523, 2013.

[15] “Website of Scisports” [Online]. Available: http://www.scisports.com/

[16] “Website of Remiqz” [Online]. Available: http://www.remiqz.com/about-us/

[17] “Website about the Euroclubindex” [Online]. Available: https://www.euroclubindex.com/methodology/

[18] Witold Kipenski. Football intelligence platform Remiqz wint SAP HANA Innovation Special Award, 2017 [Online]. Available: https://dutchitchannel.nl/576204/football-intelligence-platform-remiqz-wint-sap-hana-innovation-special-award.html

[19] Michiel de Hoog. How data, not people, call the shots in Denmark, 2015 [Online]. Available: https://thecorrespondent.com/2607/how-data-not-people-call-the-shots-in-denmark/230219386155-d2948861

[20] Jack Coles. Rise of data analytics in football, 2016 [Online]. Available: http://outsideoftheboot.com/2016/07/21/rise-of-data-analytics-in-football/

[21] Adam Bate. Scouting revolution, 2012 [Online]. Available: http://www.skysports.com/football/news/11096/8293811/scouting-revolution

[22] James O. Berger. Robust Bayesian analysis: sensitivity to the prior. Journal of Statistical Planning and Inference, 25 (1990) 303-328, 1987.

[23] Elizabeth Million. The Hadamard Product, 2007 [Online]. Available: http://buzzard.ups.edu/courses/2007spring/projects/million-paper.pdf

[24] Mark Tranmer, Mark Elliot. Multiple Linear Regression, 2008. [Online]. Available: http://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2008/2008-19-multiple-linear-regression.pdf

[25] Ofir Chokan. Practical machine learning: "Ridge Regression vs. Lasso", 2017

[26] Balaji Pitchai Kannu. What are the benefits of using ridge regression over ordinary linear regression?, 2017. [Online]. Available: https://www.quora.com/What-are-the-benefits-of-using-ridge-regression-over-ordinary-linear-regression

[27] Peter Bühlmann, Sandra van de Geer. Statistics for High-Dimensional Data:"Methods, Theory and Applications", Springer-Verlag Berlin Heidelberg, 2011

[28] Ben Gorman. A Kaggle Master Explains Gradient Boosting, 2017. [Online]. Available: http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/

[29] Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System, 2017

[30] Niklas Donges. The Random Forest Algorithm, 2017. [Online]. Available: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[31] G. Schwarz. Estimating the dimensions of a model. The annals of statistics, 6(2):461-464, 1978.

[32] “Website of Voetbal International” [Online]. Available: https://www.vi.nl/nieuws/kampioen-psv-hofleverancier-van-vis-elftal-van-het-jaar

[33] James M. Flegal, Murali Haran and Galin L. Jones. Markov Chain Monte Carlo: Can We Trust the Third Significant Figure? Statistical Science, 23(2):250-260, 2008.

Appendix A: Recreating the Bayesian hierarchical model

We recreated the model in python with data of the Eredivisie season 2017-2018. We have done this in order to properly validate this method with our own EPR method. In this section, we discuss the basics of the model. The results of the model and the simulations are discussed in Section 4.

The first step in recreating the model is loading the different packages and data into the model.

In [1]:
```
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
```

In [2]:
```
import os
import math
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
try:
    from StringIO import StringIO
except ImportError:
    from io import StringIO
import pymc3 as pm, theano.tensor as tt
import matplotlib.pyplot as plt
from matplotlib.ticker import StrMethodFormatter
import seaborn as sns
```

```
WARNING (theano.configdefaults): g++ not available, if using conda: `conda install m2w64-toolchain`
WARNING (theano.configdefaults): g++ not detected ! Theano will be unable to execute optimized C-implementations (for both CPU and GPU) and will default to Python implementations. Performance will be severely degraded. To remove this warning, set Theano flags cxx to an empty string.
WARNING (theano.tensor.blas): Using NumPy C-API based implementation for BLAS functions.
```

In [3]:
```
DATA_DIR = os.path.join(os.getcwd(), 'data/')
CHART_DIR = os.path.join(os.getcwd(), 'charts/')
```

In [4]:
```
data_file = DATA_DIR + 'eredivisie_17_18.txt'
df = pd.read_csv(data_file, sep='\t', index_col=0,)
df.head()
```

Figure A.1: Loading the packages and the data

Clearly, loading the Eredivisie 2017-2018 text file with just the scores per games is not sufficient. We have to turn this into a long dataframe, and split the score into two numeric columns, which is done in input box 5. This is better, but still not done. In order to have an easy way to refer to teams, we created a lookup table which maps a team name to a unique integer i. This is done in input box 6.

In [5]:
```
df.index = df.columns
rows = []
for i in df.index:
    for c in df.columns:
        if i == c: continue
        score = df.ix[i, c]
        score = [int(row) for row in score.split('-')]
        rows.append([i, c, score[0], score[1]])
df = pd.DataFrame(rows, columns = ['home', 'away', 'home_score', 'away_score'])
df.head()
```

Out[5]:

	home	away	home_score	away_score
0	ADO	AJX	0	2
1	ADO	AZ	0	1
2	ADO	EXC	4	1
3	ADO	FEY	0	1
4	ADO	GAE	3	0

In [6]:
```
teams = df.home.unique()
teams = pd.DataFrame(teams, columns=['team'])
teams['i'] = teams.index
teams.head()
```

Out[6]:

	team	i
0	ADO	0
1	AJX	1
2	AZ	2
3	EXC	3
4	FEY	4

Figure A.2: Creating the lookup table

We are now able to merge the last table into our main dataframe in order to create the columns i_home and i_away. It is now possible to extract the data into arrays, so that PyMC3, a package of Python, is able to process the data. Note that each of the arrays (observed_home_goals, observed_away_goals, home_team, away_team) are the same length, and that the ith entry of each refers to the same game. The last step of the code in box 7 is to come up with some decent starting values for the attacking and defense parameters.

In [7]:
```
df = pd.merge(df, teams, left_on='home', right_on='team', how='left')
df = df.rename(columns = {'i': 'i_home'}).drop('team', 1)
df = pd.merge(df, teams, left_on='away', right_on='team', how='left')
df = df.rename(columns = {'i': 'i_away'}).drop('team', 1)

observed_home_goals = df.home_score.values
observed_away_goals = df.away_score.values

home_team= df.i_home.values
away_team= df.i_away.values

num_teams = len(df.i_home.drop_duplicates())
num_games = len(home_team)

g = df.groupby('i_away')
att_starting_points = np.log(g.away_score.mean())
g = df.groupby('i_home')
def_starting_points = -np.log(g.away_score.mean())
```

Figure A.3: Merging the table

We now build the basic model in PyMC3, specifying the global parameters, the team-specific parameters and the likelihood function of the observed data.

In [8]:
```
with pm.Model() as model:
    # global model parameters
    home = pm.Normal('home', 0, 0.001)
    sd_att = pm.Gamma('sd_att', mu=.1, sd=.1)
    sd_def = pm.Gamma('sd_def', mu=.1, sd=.1)
    intercept = pm.Normal('intercept', 0, .001)

    # team-specific model parameters
    atts_star = pm.Normal("atts_star", mu=0, sd=sd_att, shape=num_teams)
    defs_star = pm.Normal("defs_star", mu=0, sd=sd_def, shape=num_teams)

    atts = pm.Deterministic('atts', atts_star - tt.mean(atts_star))
    defs = pm.Deterministic('defs', defs_star - tt.mean(defs_star))
    home_theta = tt.exp(intercept + home + atts[home_team] + defs[away_team])
    away_theta = tt.exp(intercept + atts[away_team] + defs[home_team])

    # likelihood of observed data
    home_points = pm.Poisson('home_points', mu=home_theta, observed=observed_home_goals)
    away_points = pm.Poisson('away_points', mu=away_theta, observed=observed_away_goals)
```

In [9]:
```
with model:
    trace = pm.sample(1000, tune=1000, cores=3)
    pm.traceplot(trace)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (2 chains in 2 jobs)
NUTS: [defs_star, atts_star, intercept, sd_def_log__, sd_att_log__, home]
```

Figure A.4: Creating the model

We have also written the code for the more complex mixture model. In order to replace the basic model with the more complex mixture model, input box 8 should be replaced. However, due to the fact the truncated normal distribution is no longer available in PyMC3, we were unable to properly run this code, but it should still be correct. The code is given below.

In [4]:
```
with pm.Model() as model:
    def ex_turnover_piecewise_exponential_model():

        NCT_DOF = 4
    # hyperpriors for team-level distributions
    std_dev_att1 = pm.Uniform('std_dev_att1', lower=0, upper=50)
    std_dev_def1 = pm.Uniform('std_dev_def1', lower=0, upper=50)
    std_dev_att2 = pm.Uniform('std_dev_att2', lower=0, upper=50)
    std_dev_def2 = pm.Uniform('std_dev_def2', lower=0, upper=50)
    std_dev_att3 = pm.Uniform('std_dev_att3', lower=0, upper=50)
    std_dev_def3 = pm.Uniform('std_dev_def3', lower=0, upper=50)

    sd_att1 = pm.TruncatedNormal('sd_att1', 0, .001, -3, 0, value=-.2)
    sd_def1 = pm.TruncatedNormal('sd_def1', 0, .001, 0, 3, value=.2)
    sd_att3 = pm.TruncatedNormal('sd_att3', 0, .001, 0, 3, value=.2)
    sd_def3 = pm.TruncatedNormal('sd_def3', 0, .001, -3, 0, value=-.2)

    pi_att = pm.Dirichlet("grp_att", theta=[1,1,1])
    pi_def = pm.Dirichlet("grp_def", theta=[1,1,1])

    # team-specific model parameters
    group_att = pm.Categorical('group_att', pi_att, size=num_teams)
    group_def = pm.Categorical('group_def', pi_def, size=num_teams)

    @pm.Deterministic
    def sd_atts(group_att=group_att,
                sd_att1=sd_att1,
                sd_att3=sd_att3):
        sds_by_group = tt.array([sd_att1, 0, sd_att3])
        return sds_by_group[group_att]

    @pm.Deterministic
    def sd_defs(group_def=group_def,
                sd_def1=sd_def1,
                sd_def3=sd_def3):
        sds_by_group = np.array([sd_def1, 0, sd_def3])
        return sds_by_group[group_def]
```

Figure A.5: Complex mixture model part 1

```
@pm.Deterministic
def tau_atts(group_att=group_att,
             std_dev_att1=std_dev_att1,
             std_dev_att2=std_dev_att2,
             std_dev_att3=std_dev_att3):
    taus_by_group = np.array([std_dev_att1**-2, std_dev_att2**-2, std_dev_att3**-2])
    return taus_by_group[group_att]


@pm.Deterministic
def tau_defs(group_def=group_def,
             std_dev_def1=std_dev_def1,
             std_dev_def2=std_dev_def2,
             std_dev_def3=std_dev_def3):
    taus_by_group = np.array([std_dev_def1**-2, std_dev_def2**-2, std_dev_def3**-2])
    return taus_by_group[group_def]

atts_star = np.empty(num_teams, dtype=object)
defs_star = np.empty(num_teams, dtype=object)

for i in range(num_teams):
    atts_star[i] = pm.NoncentralT("att_%i" % i, mu=mu_atts[i], lam=tau_atts[i], nu=NCT_DOF)
    defs_star[i] = pm.NoncentralT("def_%i" % i, mu=mu_defs[i], lam=tau_defs[i], nu=NCT_DOF)

# home
mu_home = pm.Normal('sd_home', 0, .0001)
std_dev_home = pm.Uniform('std_dev_home', lower=0, upper=50)

@pm.Deterministic(plot=False)
def tau_home(std_dev_home=std_dev_home):
    return std_dev_home**-2

home = pm.Normal('home',
                 mu=mu_home,
                 tau=tau_home, size=num_teams)

atts = pm.Deterministic('atts', atts_star - tt.mean(atts_star))
defs = pm.Deterministic('defs', defs_star - tt.mean(defs_star))
```

Figure A.6: Complex mixture model part 2

```
@pm.Potential
def limit_sd(std_dev_att1=std_dev_att1,
             std_dev_att2=std_dev_att2,
             std_dev_att3=std_dev_att3,
             std_dev_def1=std_dev_def1,
             std_dev_def2=std_dev_def2,
             std_dev_def3=std_dev_def3,
             std_dev_home=std_dev_home):
    if std_dev_att1 < 0 or std_dev_att2 < 0 or std_dev_att3 < 0:
        return -np.inf
    if std_dev_def1 < 0 or std_dev_def2 < 0 or std_dev_def3 < 0:
        return -np.inf
    if std_dev_home < 0:
        return -np.inf
    return 0

@pm.Potential
def keep_mu_within_bounds(sd_att1=sd_att1,
                          sd_def1=sd_def1,
                          sd_att3=sd_att3,
                          sd_def3=sd_def3):
    if sd_att1 < -3 or sd_att1 > 0 or sd_def3 < -3 or sd_def3 > 0:
        return -np.inf
    if sd_def1 < 0 or sd_def1 > 3 or sd_att3 < 0 or sd_att3 > 3:
        return -np.inf
    return 0

    return()
```

Figure A.7: Complex mixture model part 3

Running the model returned the following parameters:

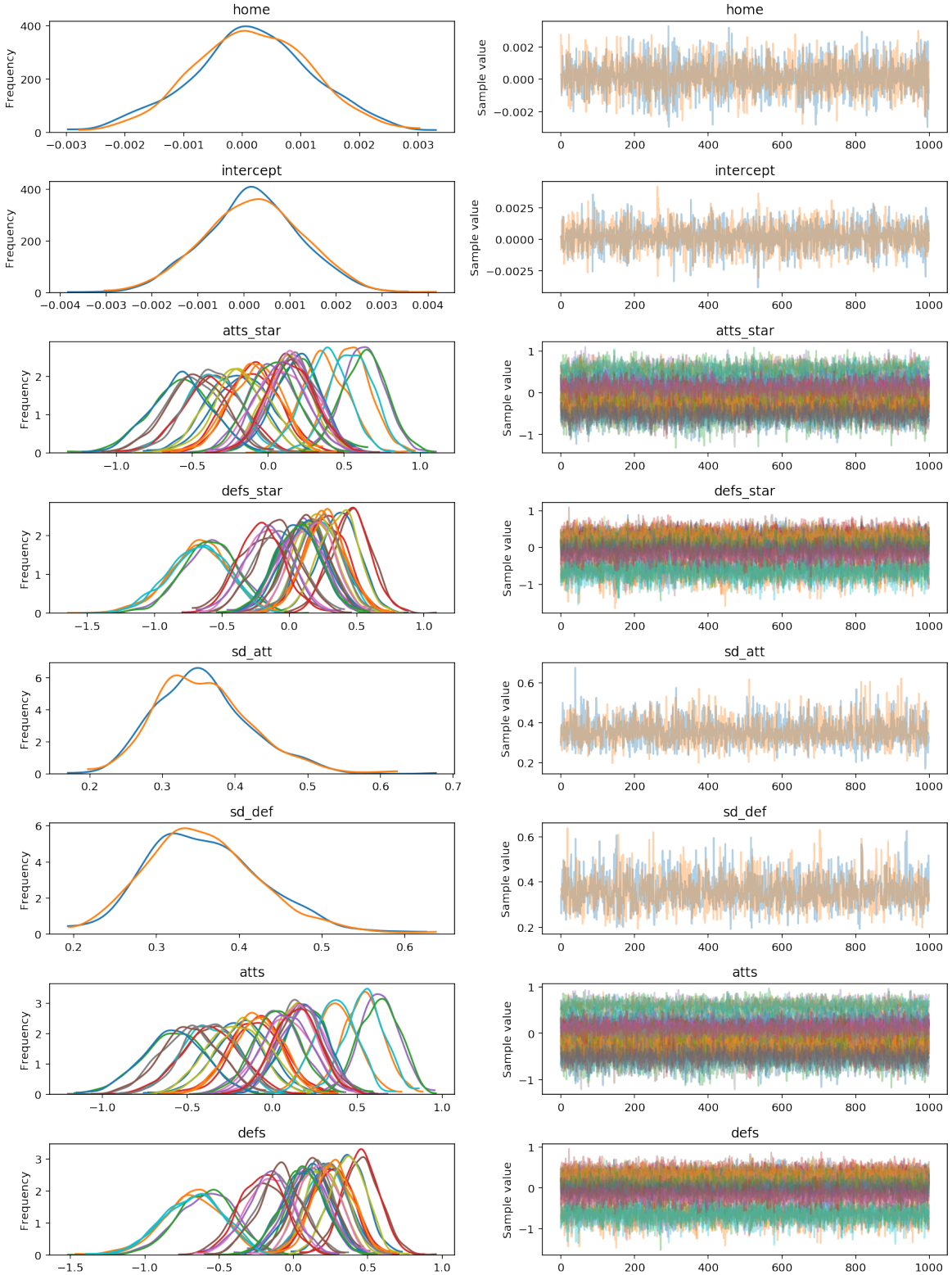

Figure A.8: Results of the different parameters

If we look at various evaluation metrics, just to verify that our model has returned the correct attributes, we can see that some teams are stronger than others. This is exactly as expected.

```
In [22]: pm.stats.hpd(trace['atts'])

Out[22]: array([[-0.55278236,  0.08036665],
                [ 0.30414901,  0.77041859],
                [-0.08099171,  0.44873319],
                [-0.36285101,  0.20314607],
                [ 0.40245475,  0.82996888],
                [-0.69865498, -0.04479631],
                [-0.08530634,  0.44626376],
                [-0.14164303,  0.41027503],
                [-0.14015759,  0.4002427 ],
                [-0.73665044, -0.07276946],
                [-0.49153616,  0.1134271 ],
                [ 0.11310533,  0.61054767],
                [-0.95434049, -0.25860224],
                [-0.41729521,  0.18141201],
                [-0.26672026,  0.27846882],
                [-0.12623247,  0.40544498],
                [-0.18819263,  0.32751454],
                [-0.82245731, -0.14224315]])

In [23]: pm.stats.quantiles(trace['atts'])[50]

Out[23]: array([-0.23048597,  0.53232688,  0.18295877, -0.07696361,  0.62963098,
                -0.36564956,  0.17246497,  0.15011454,  0.13497942, -0.37219452,
                -0.16665054,  0.3720573 , -0.57779884, -0.10472129,  0.02472094,
                 0.13940765,  0.08172587, -0.48069183])
```

Figure A.9: Team strength

Furthermore, a major consideration in markov chain monte carlo simulations is that of convergence [33]. Has the simulated Markov chain fully explored the target posterior distribution so far, or do we need more simulations. Taking a quick look at the Gelman-Rubin statistic, we see that our model is converged well and there is no indication that we need to do more simulations.

```
In [20]: bfmi = pm.bfmi(trace)
         max_gr = max(np.max(gr_stats) for gr_stats in pm.gelman_rubin(trace).values())

In [21]: (pm.energyplot(trace, legend=False, figsize=(6, 4))
            .set_title("BFMI = {}\nGelman-Rubin = {}".format(bfmi, max_gr)));
```

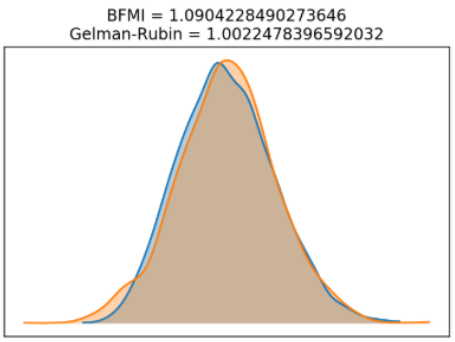

Figure A.10: Gelman-Rubin Statistic

Results are further discussed in Section 4.

Appendix B: Calculating the scores

In this section we discuss how we calculated the different scores in Table 4.6. We use the simplified model as an example. Other models are calculated in similar fashion. Furthermore, we only give the code for the default values for every model. The other models are alterations on the default model and changing the alpha and random state for lasso- and ridge regression for example, is simply done by chancing the alpha and random state in the code, which is quite straight forward.

We started with importing the packages needed and the dataset in the model.

```
In [1]: import pandas as pd
        import csv
        import sqlite3
        import matplotlib.pyplot as plt
        %matplotlib inline
        import numpy as np
        import xgboost as xgb
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LogisticRegression
        from sklearn import linear_model
        from sklearn.metrics import accuracy_score
        from sklearn.metrics import mean_squared_error
        from sklearn import metrics
        from sklearn.model_selection import cross_val_score
        from sklearn.metrics import mean_absolute_error
        from sklearn.model_selection import cross_val_score
        from sklearn.ensemble import RandomForestRegressor
        from sklearn.tree import DecisionTreeRegressor
        from sklearn import preprocessing
```

```
In [2]: df = pd.read_csv("Simplifiedmodel.csv")
        df.head()
```

Out[2]:

	Player_id	Age	Matches_played	Minutes_played	Goals	Expected_goals	Assists	Expected_assist	Height	Succ_def_90	...	Final_3rd_90	Final_3rd_acc	P
0	361	27	15	1005	0	0.61	0	0.26	182	6.81	...	10.84	71.90	
1	74	19	14	1021	0	0.37	0	0.12	181	5.51	...	6.83	80.70	
2	246	22	15	1023	1	0.45	0	0.42	178	5.28	...	4.84	76.36	
3	295	20	22	1023	0	0.26	0	0.01	180	6.60	...	5.98	64.71	
4	299	38	26	1027	0	0.11	0	0.10	182	8.41	...	7.89	70.00	

Figure B.1: Importing the packages and data

The next step is to split the dataframe in a target data, Y, which will be the rating in our case, and, X, which are all the other features/attributes of the players. This is done in input box 9. After splitting the dataframe, we can start dividing the dataset into training and validation data. Recall that we divided and used an 80% / 20% of training and validation data for every model. This is represented in input box 15.

```
In [9]: #Split dataframe into df_x and df_y
        df_x = soccer_data.drop(['Rating'], 1)
        df_y = np.array(soccer_data['Rating'])
```

```
In [13]: from sklearn.preprocessing import StandardScaler, Normalizer
         df_x = StandardScaler().fit_transform(df_x)
```

```
In [14]: from sklearn.decomposition import PCA

         # on non-standardized data
         df_x = PCA(n_components=10).fit_transform(df_x)
```

```
In [15]: x, x_test, y, y_test = train_test_split(df_x,df_y,test_size=0.2,train_size=0.8, random_state = 55)
         x_train, x_cv, y_train, y_cv = train_test_split(x,y,test_size = 0.20,train_size =0.80, random_state = 55)
```

Figure B.2: Splitting the dataframe and dividing the dataset

Now that the dataset has been divided, we can start building the different models. The code for MLR is given below:

```
In [13]: #Linear Regression
         #Default values
         n=np.mean(y_test)
         #print(n)
         clf = linear_model.LinearRegression(fit_intercept=True, normalize=False, copy_X=True, n_jobs=1)
         clf.fit(x_train,y_train)

         y_pred = clf.predict(x_test)
         #print(mean_squared_error(Ytest,y_pred))
         print('cross val score: %f' %np.mean(cross_val_score(clf, x_train, y_train, cv=10)))
         print('Accuracy: %.2f' % clf.score(x_test, y_test))
         print('mean absoluter error: %f' % (mean_absolute_error(y_test, y_pred)))
         print('Baseline Mean squared error: %.2f' % np.mean((clf.predict(x_test) - n) ** 2))
         print('Mean squared error: %.2f' % np.mean((clf.predict(x_test) - y_test) ** 2))
         clf.score(x, y), 1 - (1-clf.score(x, y))*(len(y)-1)/(len(y)-x.shape[1]-1)
```

Figure B.3: Multiple Linear Regression

The code for the lasso regression is given below:

```
In [31]: #Lasso
         #Default values
         n=np.mean(y_test)
         #print(n)
         clf=linear_model.Lasso(alpha=0.1, fit_intercept=True, normalize=False,
                                precompute=False, copy_X=True, max_iter=1000,
                                tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic')
         clf.fit(x_train,y_train)

         y_pred = clf.predict(x_test)
         #print(mean_squared_error(Ytest,y_pred))
         print('cross val score: %f' %np.mean(cross_val_score(clf, x_train, y_train, cv=10)))
         print('Accuracy: %.2f' % clf.score(x_test, y_test))
         print('mean absoluter error: %f' % (mean_absolute_error(y_test, y_pred)))
         print('Baseline Mean squared error: %.2f' % np.mean((clf.predict(x_test) - n) ** 2))
         print('Mean squared error: %.2f' % np.mean((clf.predict(x_test) - y_test) ** 2))
         clf.score(x, y), 1 - (1-clf.score(x, y))*(len(y)-1)/(len(y)-x.shape[1]-1)
```

Figure B.4: Lasso regression

The code for the ridge regression is given below:

```
In [19]: #Ridge
         # Default values
         n=np.mean(y_test)
         #print(n)
         clf = linear_model.Ridge(alpha=1.0, fit_intercept=True, normalize=False,
                                  copy_X=True, max_iter=None, tol=0.001, solver='auto', random_state=None)
         clf.fit(x_train,y_train)

         y_pred = clf.predict(x_test)
         #print(mean_squared_error(Ytest,y_pred))
         print('cross val score: %f' %np.mean(cross_val_score(clf, x_train, y_train, cv=10)))
         print('Accuracy: %.2f' % clf.score(x_test, y_test))
         print('mean absoluter error: %f' % (mean_absolute_error(y_test, y_pred)))
         print('Baseline Mean squared error: %.2f' % np.mean((clf.predict(x_test) - n) ** 2))
         print('Mean squared error: %.2f' % np.mean((clf.predict(x_test) - y_test) ** 2))
         clf.score(x, y), 1 - (1-clf.score(x, y))*(len(y)-1)/(len(y)-x.shape[1]-1)
```

Figure B.5: Ridge regression

The code for the XGBoost algorithm is given below:

In [28]:
```
# Apply xgboost model on the dataset
n=np.mean(y_test)
clf = xgb.XGBRegressor()
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
print('Accuracy: %.2f' % clf.score(x_test, y_test))
print('cross val score: %f' %np.mean(cross_val_score(clf, x_train, y_train, cv=10)))
y_pred = clf.predict(x_test)
print('mean absoluter error: %f' % (mean_absolute_error(y_test, y_pred)))
print('Baseline Mean squared error: %.2f' % np.mean((clf.predict(x_test) - n) ** 2))
print('Mean squared error: %.2f' % np.mean((clf.predict(x_test) - y_test) ** 2))
clf.score(x, y), 1 - (1-clf.score(x, y))*(len(y)-1)/(len(y)-x.shape[1]-1)
```

Figure B.6: XGBoost algorithm

The code for the random forest algorithm is given below:

In [23]:
```
#Random Forest
#Default values
n=np.mean(y_test)
clf = RandomForestRegressor(n_estimators=10, criterion='mse', max_depth=None,
                            min_samples_split=2, min_samples_leaf=1,
                            min_weight_fraction_leaf=0.0, max_features='auto',
                            max_leaf_nodes=None, min_impurity_split=1e-07, bootstrap=True,
                            oob_score=False, n_jobs=1, random_state=None, verbose=0, warm_start=False)
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
print('Accuracy: %.2f' % clf.score(x_test, y_test))
print('cross val score: %f' %np.mean(cross_val_score(clf, x_train, y_train, cv=10)))
y_pred = clf.predict(x_test)
print('mean absoluter error: %f' % (mean_absolute_error(y_test, y_pred)))
print('Baseline Mean squared error: %.2f' % np.mean((clf.predict(x_test) - n) ** 2))
print('Mean squared error: %.2f' % np.mean((clf.predict(x_test) - y_test) ** 2))
clf.score(x, y), 1 - (1-clf.score(x, y))*(len(y)-1)/(len(y)-x.shape[1]-1)
```

Figure B.7: Random forest algorithm

The code for the decision tree algorithm is given below:

In [26]:
```
#Decision Tree
#Default values
n=np.mean(y_test)
clf = DecisionTreeRegressor(criterion='mse', splitter='best', max_depth=None,
                            min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
                            max_features=None, random_state=None, max_leaf_nodes=None,
                            min_impurity_split=1e-07, presort=False)
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
print('Accuracy: %.2f' % clf.score(x_test, y_test))
print('cross val score: %f' %np.mean(cross_val_score(clf, x_train, y_train, cv=10)))
y_pred = clf.predict(x_test)
print('mean absoluter error: %f' % (mean_absolute_error(y_test, y_pred)))
print('Baseline Mean squared error: %.2f' % np.mean((clf.predict(x_test) - n) ** 2))
print('Mean squared error: %.2f' % np.mean((clf.predict(x_test) - y_test) ** 2))
clf.score(x, y), 1 - (1-clf.score(x, y))*(len(y)-1)/(len(y)-x.shape[1]-1)
```

Figure B.8: Decision tree algorithm