Rating football teams of all amateur levels based on performance

An approach using implementation specific knowledge

K.C. van Noortwijk
3867129

University of Utrecht

Master Thesis
Artificial Intelligence
Faculty of Science

Supervisors:
Dr. M.J.S. Brinkhuis
Dr. A.J. Feelders

March 27, 2018

Abstract

The Royal Dutch Football Association is responsible for dividing all amateur teams in the Netherlands into groups of similar strength each season. The teams in these groups play games against each other, which forms the basis for future placement in higher or lower level groups. This dividing of teams is currently mostly done by hand. To facilitate the partial automation of this process, the current study presents several ways of creating a rating for all standard football teams in the Netherlands. This includes analysis of the Elo, Glicko2 and Elo++ rating systems and their relevance in the context of rating amateur football teams. The influence of implementation specific parameters on the rating is investigated and the analysis shows that home field advantage and goal difference are relevant in this context. Thirdly the closeness factor stemming from graph theory was analyzed for all teams, which showed a positive correlation between a low closeness factor (and thus a high connectedness with all other teams) and the performance of teams. Parameters describing the population density, club density and average disposable income in the area where a team is located, were not found to correlate with the rating of teams. Various models were trained based on the Elo, Glicko2 and Elo++ algorithms, which all showed an accuracy of about 66%. The addition of factors for the three relevant implementation specific parameters to the models improved their accuracy by maximally 0.5%.

Acknowledgements

During the writing of this thesis I became increasingly aware of the approaching end of my time as a student. As much as I loved being a student and being involved in university life as a whole, even good things must come to an end.

I would like to thank my supervisors Dr. Matthieu Brinkhuis and Dr. Ad Feelders for letting me conduct this research any way I wanted and always getting me excited about my own research anytime we spoke about it.

Secondly, thanks to Isabel, Annette, Jacques and Rosa, for proofreading my work and providing a different perspective on things whenever I needed it.

Above all, I would like to thank my parents, for their relentless love and support. Without them, none of this would have been possible.

Contents

1 Introduction

Technological advancements have made a big impact on football in recent years. One of the most recent examples of that is the implementation of goal line technology. Where previously a team of football referees had to judge the validity of a goal by themselves in a very short time frame, nowadays goal line technology is put in place to take over this controversial duty in some competitions. Not only does this technology judge the validity of goals more accurately, it also has the nice property of taking away the controversy around such a decision.

Just like goal line technology is taking over the responsibility for judging goal validity, many other aspects of football are being automated. The nice side effect of these automations is that the impartiality of technology is generally beyond dispute. No one will argue it is more fair to let football referees decide whether a goal was valid based on their own observations, versus letting an automated system that has sensors in both the goals and the ball assess the validity of a goal.

Apart from automations like goal line technology, which affect processes that take place at the time of playing a game, automations can also be implemented in organizational processes involved with football. One such process that can be automated, is the process around dividing teams over leagues and divisions. Generally national football associations have too many teams associated with them to have every team compete with every other team. On top of that, the associated teams might be of varying tiers of strength. By dividing the teams into groups which only play among themselves in a given season, these problems can be solved. The process involved in this dividing of teams into groups was originally done by hand. To facilitate the automation of this process, a rating for all teams has to be created such that the strength of each team can be assessed in an impartial way. This rating can then be used to create groups of teams with similar playing strength.

Rating systems have been around for quite some time, but never really found their way into local level football in the Netherlands. Perhaps their relatively high level of abstraction so far did not make them very accessible for usage in the context of football.

1.1 Problem outline

The Royal Dutch Football Association (Koninklijke Nederlandse Voetbal Bond, or KNVB) is responsible for organizing football leagues and divisions in the Netherlands. Part of this responsibility is making sure all teams end up in performance appropriate leagues (a set of teams of similar strength, usually of size 8-12). Analyzing the strength of teams is done in two basic ways, namely: (1) if a team finished first in its league in the previous season, it is promoted, if it finishes last, it is demoted and (2) the result of the play offs (usually the second to last and the third to last team play each other for demotion and the second and third team play each other for promotion). Which teams are directly promoted or demoted, as well as which teams get sent to the play offs, can differ

per division (a set of leagues of similar strength) [6].

Using these basic performance indicators, human insight and preference from the teams, the KNVB produces new leagues and divisions each season.

This way of working has several downsides, the most important one being that it is very well possible for teams to get assigned to leagues in which the other teams perform at a different level. While only a predefined number of teams from the same league can be promoted (for example one directly and one through play offs), it is easily imaginable that there might be more teams eligible for promotion from a single league, when their performance is compared to teams from other leagues in the same division. Or perhaps a league exists where not a single team should be promoted when their performance is compared to teams from other leagues in the same division. The result of which is, that these teams end up in the wrong league and either lose or win almost all matches the following season. Lastly, the creation of new leagues and divisions each year takes a serious amount of time, since all this is processed by hand.

The aim of this project is to investigate different rating algorithms that could facilitate the partial automation of this process by creating a rating for all teams that is updated after each football season. With the help of this rating, it is expected that the creation of more equal and exciting leagues and divisions can be facilitated, and that the required amount of man hours to create the leagues and divisions for coming seasons can be reduced. Using techniques commonly used in machine learning, an AI-solution will be investigated and compared to more traditional approaches. The results of this research project can be used as a decision support mechanism: the rating systems can help solidify the reasoning behind promotion and demotion of teams as well as provide measurable parameters to substantiate decisions involving league creation.

Furthermore, indications exist that parameters such as social economic status, geographic position and club tradition affect the performance of teams [23] [20] [8]. These parameters can be assessed using the resulting ratings and can be incorporated into the rating algorithms, in search of better performance.

Lastly, an extension is proposed to the known rating systems. Since so many football teams participate in the Netherlands, the rating of many teams is going to be very loosely connected to the entire pool of teams (i.e. The rating of a team in one corner of the country might not be very accurate when compared to a team from another corner of the country, because both teams operate in mostly separated playing pools). However, this problem is not covered by existing literature about rating systems. A probable reason for this absence, is that most literature about rating systems covers their usage in large tournament settings, where the separation between players or teams is relatively small, and generally the same for all players. However, in this particular implementation the teams are not free to play against any opponent, which could make the separation between teams much larger. A 'calibration' of ratings will be proposed as a solution, inspired by a similar calibration sequence as proposed by Sonas [25], but adapted to improve loosely connected ratings, instead of Sonas' original use in comparing historical chess ratings (for example to compare the playing strength of a 1920s chess master to a current player).

1.2 Research questions

The aim of the current study is to create ratings for standard amateur football teams in the Dutch amateur football competition. The creation of this rating will be the first step towards full automation of the creation of new leagues and divisions each season. To evaluate the advantages and disadvantages of several candidate rating systems, a rating will be created in three different systems, namely the Elo system, Glicko2 system and Elo++ system. The implementations of these three systems will have two additions. Firstly, the rating of poorly connected or disconnected teams is calibrated towards the rating scale of more well-connected teams. Secondly, the addition of implementation specific parameters into the systems will help the prediction of game outcomes and thus the fitting of ratings to the playing ability of teams, as playing ability is expected to correlate with a number of properties belonging to a particular team.

The main research question to be answered in this study will be:

Is it possible to improve the performance of rating algorithms in the context of Dutch amateur football using implementation specific knowledge?

And the three sub questions following from that are:

(1) **How does the performance of relevant existing rating algorithms compare?**

In this study, the Elo, Glicko2 and Elo++ rating systems will be compared by assessing how well the expected outcome of games can be predicted by each system, after training them on several previous seasons of Dutch amateur football game data. The performance of the different rating systems will be compared using metrics from machine learning.

The second sub question is:

(2) **Can rating systems be improved by adding implementation specific parameters, such as factors that model home field advantage or club density in the area where a team is located?**

Using the output of the rating algorithms, the correlation between different implementation specific parameters and the rating score of teams can be assessed and correlations could be found between the parameters and performance of teams. The correlations that were found to be significant, can be used to improve the rating algorithms by adding factors for the values of the parameters to them. By assessing the performance of the rating systems both before and after the addition of the factors for the parameters to them, a comparison can be made between the different versions of the rating systems.

The final sub question is:

(3) **Can rating systems be improved by adding a calibration sequence to level out differences in the closeness of teams?**

By calculating the closeness factor, which is a metric stemming from graph theory that describes how tightly nodes are connected in a graph, secluded teams can be identified and the accuracy of their rating can be improved. Again, versions of the rating systems will be compared with and without the proposed calibration sequence, to assess the difference in their performance.

2 Theory

In this chapter the theoretical background of various rating systems will be examined. Two types of rating systems can be identified, namely traditional systems, and machine learning systems. Four traditional systems will be discussed and one machine learning system. Furthermore an example of a calculation in the Elo system will be given, to give an idea of how traditional rating systems work.

2.1 Traditional rating systems

In the following section a rather odd subgroup of traditional rating systems is discussed, of which the members do not necessarily need to follow clearly defined rules, because the actual ratings of players, or the rules upon which the ratings of players are based, are defined solely by expert opinion. The sections after that will discuss their counterpart: the non-expert based systems.

2.1.1 Expert based systems

In many sports, player or team ratings are determined by expert based principles, like giving specific amounts of points (determined by experts) to winners of specific tournaments (ATP tennis ranking [4]), or plain expert opinion (NCAA basketball [5]). Advocates of these types of systems point out the trust that has been widely granted to them over the years of their existence, but often heard criticisms about these rating systems are that they do not represent the true skill of specific teams or players and that the lack of alternative ways of rating players or teams is an underestimated factor in the acceptance of these types of systems.

Their non-expert based counterparts make use of principles from statistics, rather than expert opinions, to determine the rating of teams or players. These non-expert based rating systems form the main focus of this thesis. They can update ratings as frequently as once after every game played, but in this thesis, a batch update approach per season will be used for all non-expert based rating systems, for two main reasons: (1) Less updates are needed which provides more smooth results, since erratic behaviour of teams is averaged out, and (2) the data set does not contain time stamps for all games. Because of this the order in which the games are played is not always clear, which would influence the rating of teams. By employing a batch update approach, this is conveniently no longer an issue, since all games in a batch can be considered to have happened at the same time.

The first widely accepted non-expert based rating system was implemented by the international chess federation (FIDE). The so called Elo system would turn out to be the benchmark for non-expert based rating systems for decades to come.

2.1.2 Elo

In 1978 Elo published his magnum opus on chess ratings and his ideas about the different ways it can benefit the understanding of playing ability of chess players [11]. The proposed rating system would become the industry standard when it comes to non-expert based rating of performance in sports players and teams. The genius of this system, and probably the most important reason for its success, lies in the fact that calculating changes in rating is fairly easy and doable by hand. A desirable property in itself, and a direct consequence of this relative simplicity is that it is always fairly simple to calculate the amount of rating points that are at stake for any specific match.

In favor of this simplicity some predictive power has been traded off, which is where alternative systems propose improvements, as the next two sections will show.

The algorithm utilized to calculate the change in rating can be summarized in three steps:

1. Assign a predefined start rating to all new players in the player pool.
2. Have the player pool play games against each other (this can be more than one game per player) and record the game outcomes.
3. calculate the new ratings using Formula 1.

where Formula 1 is given by:

$$R' = R + K(S - E) \tag{1}$$

Where R' is the players new rating, R is the players current rating, K is a correction factor, S is the average outcome of the played games by the player in question, where 1 encodes a won game, 0.5 a drawn game and 0 a lost game. E is the expected outcome of those same games in the same encoding (1 for a won game, 0.5 for a draw and 0 for a loss) and the value of the K-factor determines the speed at which ratings can change. The value of E can be determined in various ways, one of them being given in Formula 2:

$$E = \frac{10^{R_a/400}}{10^{R_a/400} + 10^{R_b/400}} \tag{2}$$

Where R_a stands for the current rating for player A, and R_b stands for the current rating for player B. Using this formula for E makes every 400 points difference between players be expressed as one player being '10 times as good'. Note the arbitrarity of the specific amount of points in this rule, one could just as well have chosen 200 points to reflect the same difference.

Using this rating system, Elo found it possible to identify factors of success, since it was now possible to quantify and track changes in rating over a period of time [11]. In other words, with the use of his rating system it was possible to quantify growth or decline in playing ability.

As briefly mentioned above, by employing a relatively simple formula to calculate rating changes, Elo traded away some predictive ability. Intuitively one can understand that by simplifying the formula that 'fits' the recorded player rating to the actual playing ability of a player, the fit becomes more crude and the approximation of the rating to the actual playing ability becomes less precise [12]. On the other hand, by utilizing this simplification, it becomes much easier to understand exactly what is at stake for each game. Since the Elo system is essentially an economic system (meaning that, given a stable pool of players, points can only ever be redistributed, never created anew), each player can always calculate the points that can be won or lost when facing off with any other rated player.

Apart from this fitting issue, a problematic feature of this system is the inflation effect that its implementations suffer from. When looking at Elo rating data about chess players from different time periods it seems clear some rating inflation is happening. Although some controversy on this topic exists, a discrepancy between inflow and outflow of players is sometimes accused of being the reason for this effect. Since players are constantly added to, and removed from the system, the player pool is not stable. If more players are added than there are players that are removed from the system, this could mean there is an increase in the total amount of points in the system. This causes an inflation effect, much like the yearly inflation effect in the monetary world (because new money is constantly added to the system). Another factor in this inflation effect could be the initial rating that is assigned to new players [25]. By overrating new players on average, even if it is only very slightly, the net effect could be that rating points are donated to the entire player pool (since these overrated new players would lose their rating points to players with the same or higher rating). In any case, even when the cause of this inflation is still somewhat up for debate, the fact that it is happening points towards a flaw in the system.

Now that the basics of the Elo rating system have been covered, let us conclude this section with an example, since it aides understanding of the Elo system, which is considered to be the basis for the rating systems that are discussed in the next sections.

Let us define player A and player B. Player A has a pre-rating-period rating of 2000 and player B has a pre-rating-period rating of 1800. Player A and player B play 10 games against each other, of which player A wins 5 and draws 2 games. Consequently player B wins 3 and draws 2 games. The average outcome of the games for player A then is:

$$S_a = 5 + \frac{1}{2} + \frac{1}{2} = 6, and\ \frac{6}{10} = 0.6$$

The average outcome of the games for player B then is:

$$S_b = 3 + \frac{1}{2} + \frac{1}{2} = 4, and\ \frac{4}{10} = 0.4$$

Secondly the expected outcome can be calculated by using Formula 2:

$$E_a = \frac{10^{R_a/400}}{10^{R_a/400} + 10^{R_b/400}} = \frac{10^{2000/400}}{10^{2000/400} + 10^{1800/400}} = 0.76$$

and:

$$E_b = \frac{10^{R_b/400}}{10^{R_b/400} + 10^{R_a/400}} = \frac{10^{1800/400}}{10^{1800/400} + 10^{2000/400}} = 0.24 = 1 - 0.76$$

Formula 1 can now be used to calculate the new rating for player A. Let us take $K = 30$, which is the official K-factor used by FIDE for players rated below 2400 that have played 30 or more recorded games:

$$R'_a = R_a + K(S_a - E_a) = 2000 + 30(0.6 - 0.76) = 1995$$

and for player B:

$$R'_b = R_b + K(S_b - E_b) = 1800 + 30(0.4 - 0.24) = 1805$$

Comparing the values of E and S, player B outperformed his expected ability, and player A under-performed according to his expected ability. This is the reason that, even though player A won the majority of the games, he still lost rating points, and conversely, player B won rating points even though he lost the majority of the games. A second observation is the fact that the amount of points lost by player A is the exact same amount of points won by player B. As mentioned above, this is the case because the Elo system can be considered an economic system, and the total amount of points, given a stable player pool, always remains the same.

2.1.3 Glicko

In an attempt to fix some of the known issues with the Elo rating system, and acknowledging the advancement of computer technology, which greatly improved accessibility of cheaper, more powerful computers, Glickman published a new, stand alone rating system [13]. Upon closer inspection of his early publications it is obvious he is heavily inspired by the Elo system, and with good reason. By the time of his publication the Elo system had been successfully implemented by FIDE for about 30 years. However some of the discussed fundamental problems the Elo system exhibits had become clear over these years. Glickman decided to focus on shifting the simplicity/precision trade-off with his solution, by using computationally more expensive formulas to calculate changes in rating for individuals over a rating period. Secondly he introduced precision intervals for each rating. A pleasant property of these precision interval ratings is the fact that it allows us to present confidence intervals to users. Lastly, by introducing a so-called 'Rating Deviation' (or RD) factor, a not previously discussed problem with rating systems can now be tackled, namely behaviour of players focused on

maximizing their rating. Let's imagine a player who is somewhat overrated because he won a tournament on a fluke. Based on this win he would be rewarded with a rating increase, which would rank him above his true playing ability. The player could then game the system, by only playing very few games. By doing this, the system would never get the chance to correct his rating enough, so that his true skill would always remain overrated. The RD factor is introduced precisely to address this problem. Simply put, the RD factor models inactivity by giving more fluctuation power to a rating, the longer the player it is associated with is inactive. The idea being that the true skill of inactive players changes, but that these changes are not reflected in their rating, so that they have to be included at the next opportunity for changing the rating in order to make the system adapt more quickly.

Glickman improved his Glicko system by introducing a new 'volatility' factor in the Glicko2 rating system [14]. This new factor aims to model the erratic behaviour players can exhibit when they perform above or below their expected performance based on their rating. The idea is that players that perform erratically were hard to model in the original Glicko system as this system was not able to swing their ratings up or down enough per rating period. This volatility factor can be seen as the Glicko analogous version of the K-factor in the Elo rating system. It is worth noting that the introduction of this new factor increases the complexity of the rating calculations further, which was, as mentioned before, already much higher than the complexity level of the calculations needed to run the Elo rating system.

In this section the mathematics behind the Glicko2 system are discussed. Note that Glicko2 and Glicko use distinctly different values for rating and rating deviation, and they use different symbols for rating (μ in Glicko2) and deviation (ϕ in Glicko2). Formulas that translate between the two systems exist, but will be omitted here, since this section focuses mainly on the workings of Glicko2. The following formulas are of importance in the Glicko2 system:

The estimated variance of the rating can be calculated with Formula 3:

$$v = \frac{1}{[\sum_{j=1}^{m} g(\phi_j)^2 E(\mu,\mu_j,\phi_j)\{1 - E(\mu,\mu_j,\phi_j)\}]} \tag{3}$$

Where:

$$g(\phi) = \frac{1}{\sqrt{1 + 3\phi^2/\pi^2}} \tag{4}$$

$$E(\mu,\mu_j,\phi_j) = \frac{1}{1 + e^{(-g(\phi_j)(\mu-\mu_j))}} \tag{5}$$

m is the number of opponents played against (note that the number of opponents does not have to equal the number of games played, since multiple games can be played against the same opponent) and E denotes the expected (average) game outcome.

The estimated improvement can then be calculated by:

$$\Delta = v \sum_{j=1}^{m} g(\phi_j)\{s_j - E(\mu, \mu_j, \phi_j)\} \tag{6}$$

Where s_j denotes the (average) actual game outcome.
To calculate the new volatility σ', an iterated computation is needed in which the following formula is used:

$$f(x) = \frac{e^x(\Delta^2 - \phi^2 - v - e^x)}{2(\phi^2 + v + e^x)^2} - \frac{x - ln(\sigma^2)}{\tau^2} \tag{7}$$

where σ denotes the old volatility and τ is a system constant which constrains the volatility.

The iterated computation can be described with the following sequence:

Algorithm 1 Glicko2 iterated computation of volatility

Require: σ, Δ, ϕ, v, τ
  $A = ln(\sigma^2)$
  if $\Delta^2 > \phi^2 + v$ **then**
    $B = ln(\Delta^2 - \phi^2 - v)$
  else
    $k = 1$
    while $f(A - k\tau) < 0$ **do**
      $k = k + 1$
    end while
    $B = ln(\sigma^2) - k\tau$
  end if
  $f_A = f(A)$
  while $|B - A| > \epsilon$ **do**
    $C = \dfrac{A + (A - B)f_A}{f(B) - f_A}$
    if $f(C)f(B) < 0$ **then**
      $A = B$
      $f_A = f(B)$
    else
      $f_A = f_A/2$
    end if
    $B = C$
  end while
  $\sigma = e^{A/2}$

Where ϵ is a prechosen error tolerance value.

After executing this sequence, the old ϕ can be updated to a new post-rating-period ϕ_*:

$$\phi_* = \sqrt{\phi^2 + \sigma'^2} \tag{8}$$

And finally, the rating and rating deviation can be updated:

$$\phi' = \frac{1}{\sqrt{\frac{1}{\phi_*^2} + \frac{1}{v}}} \tag{9}$$

$$\mu' = \mu + \phi'^2 \sum_{j=1}^{m} g(\phi_j)\{s_j - E(\mu, \mu_j, \phi_j)\} \tag{10}$$

To write out the calculations of an example in this system would require several pages of uninteresting iterative calculations. Instead, some remarks about the rating calculation process can be made, since some clear similarities and differences with the Elo system can be identified. It should for example be noted that Formula 10 shows clear resemblance to Formula 1, used in the Elo system. Furthermore, the Glicko2 system requires a lot more computational power than the Elo system. Although too awkward to do by hand, because of the iterated process involved, it is fortunately very doable with the help of a computer. This complication has brought two big advantages, namely the inclusion of volatility and rating deviation factors. Some efforts have been made to implement variants of the Glicko system in the context of American Football and Beach Volleyball, demonstrating the generalizability of this system [16] [15].

2.1.4 Sonas

Although not publishing his ideas in scientific publications like Elo or Glickman, Sonas has had influential ideas on the topic of rating systems. Sonas proposed a system, much like the Elo system, which was mostly set up to compare historical chess masters' ratings to the rating of current players. Sonas specifically meant his system to be an improvement over Elo's historical rating comparison, by including a gradual 'weighing and padding' factor for each rating. Neither the Elo or Glicko systems use explicit temporal weighting, which Sonas defined as linearly weighting game outcomes, such that games from the past 2 months have a 50% weight and games that happened 2 years ago have only a 2% weight. Both the Elo and Glicko system have a weighting equivalent (the k factor and the combination of the player and opponent RD factors respectively), but they are both factors that do not include a time component. Secondly, Sonas introduces a so called padding of 7 games against a player with a predefined rating to reward players that play more. Recall that Glicko, in a similar effort, specifically punishes players that play less games by increasing their rating deviation factor. Thirdly the Sonas system makes use of calibration, which Sonas defines as taking the average rating of the players ranked third to twentieth. He then uses this presumably constant rating value to compare ratings of chess grand masters of different eras, and calibrating them so that they are of equal offset [25].

Especially this last part of the Sonas system is an inspiration for this project, since badly connected or even completely disconnected sets of teams in the KNVB competition are likely to occur. Making use of a calibration sequence

much like the one Sonas proposes, the rating comparison of teams that are badly connected or disconnected with the rest of the football teams can be improved. To add some more weight to this idea, the calibration of ratings like this can also be found in the context of comparing results of tests in education between digital and on-paper versions of the same test [18].

As mentioned, the documentation for this rating system is somewhat sketchy, since there are no scientific publications of it. However judging by the number of references in related articles in rating system research, Sonas' system is being classified as important.

2.2 Machine learning approach

To involve machine learning into calculating the rating of sports teams or players was a logical step. Most notably because this kind of problem is very similar to problems that can be solved using regression algorithms. Generally one can think of it like this: Regression algorithms try to fit a curve to a set of samples. By going over training data, the algorithm tries to learn which combination of parameter values contribute to which particular point in the vector space. This is more or less exactly what we want to do in the application of rating sports teams or players. Many different algorithms exist, mostly varying in rigidity of the curve the algorithm is using to fit towards the data. Generally more rigid and simple algorithms, perform better on simple or noisy data, where as more elaborate algorithms employ much more adaptable and flexible curves perform better on more elaborate and extensive input data. One can imagine that the problem of predicting football game outcomes has a lot of relevant data missing, or at least not readily available. Simply because it is not recorded, or because it might not be known to be of influence on the game outcome.

It is because of a similar realization that a variation of the fairly simple and rigid optimization algorithm of stochastic gradient descent was implemented under the name Elo++ in an effort to predict chess game outcomes. This algorithm is used as a part of well known machine learning algorithms, such as support vector machines and logistic regression.

It was in fact this implementation that turned out to perform the best compared to implementations of much more elaborate algorithms in a recent Kaggle competition, in which the specific goal was to find an algorithm that could outperform the Elo system in predicting chess game outcomes.

2.2.1 Elo++

Sonas' commitment to the problem of rating chess players, and his open mind towards any possible solution is shown by the fact that he had a Kaggle competition organized named: 'Elo vs. The World', in which participants were challenged to design a rating system that outperformed the Elo system. The winning submission of this competition was, somewhat surprisingly, not a clean implementation of Glicko, Sonas or Elo but an algorithm that can be described as a combination of the Elo and Sonas systems, named Elo++. Characterized by employing stochastic gradient descent to fit ratings, and including white advantage, time weighting and rating regularization, Elo++ is by far not the most elaborate algorithm that was submitted, as even the author admits he put most of his time in adding complex parameters, which only turned out to overfit on the training set [24]. However a factor to capture white advantage is included, which is one of its main differences when compared with the two ratings systems discussed in the previous sections. By employing stochastic gradient descent [24], a technique commonly used in machine learning algorithms, we enter the realm of machine learning, which is an important playing field in this story. Although often machine learning algorithms provide important

insights and sometimes show incredible results in classification and regression tasks, their 'black-box'-like functionality can be identified as a fundamental flaw in rating applications. Especially applied in sports rating assignment, it might be hard for sports teams and spectators alike to swallow performance ratings of which the calculation is unknown or unclear.

The Elo++ rating update algorithm can be written out like this:

Algorithm 2 Elo++

Require: G, γ, λ, P
  for all players i, $r_i = 0$
  for all games in G, compute temporal weight w
  for p=1 to P **do**
    for all players i, compute average opponent rating a_i
    $\eta = ((1 + 0.1P)/(p + 0.1P))^{0.602}$
    for all shuffled games g in G **do**
      $E = 1/(1 + e^{r_a - (r_h + \gamma)})$
      $r_h = r_h - \eta(w(E - o)E(1 - E) + \frac{\lambda}{|N_h|}(r_h - a_h))$
      $r_a = r_a - \eta(w(E - o)E(1 - E) + \frac{\lambda}{|N_a|}(r_a - a_a))$
    end for
  end for
  return all ratings r_i

Where G is the set of games, γ the predefined home advantage, λ the predefined regularization constant, P is a predefined number of iterations that the algorithm is allowed to go through, r_a is the rating of the away team, r_h is the rating of the home team, a_a is the weighted average opponent rating of the away team, a_h is the weighted average opponent rating of the home team, N_a is the number of opponents of the away team this season, N_h is the number of opponents of the home team this season and o denotes the actual game outcome.

The implementation of the time weighting algorithm, definition of λ and η, and the weighting of the average opponent rating will be omitted here, but can be found in the referenced article [24].

A few things separate this implementation from a standard implementation of stochastic gradient descent, namely: the inclusion of the average rating of opponents of all teams, and the addition of a factor that denotes home advantage (γ). The idea behind the inclusion of average opponent rating is that teams generally play against teams that have similar playing ability as themselves. Therefore their rating should be similar as well. The home advantage factor is actually a white advantage factor in the original publication as the algorithm was created to rate chess players. The idea was that playing white carries a distinct advantage, as these players are allowed to move first and thus have initiative from the beginning of the game. Although playing white in a game of chess and playing at home in a football game have nothing to do with each other, the idea of having an advantage when playing white in a game of chess perhaps somewhat surprisingly carries over to playing at home in football game,

since, statistically, playing at home carries a significant advantage over playing at an away field [22] [21]. The relevance of the home advantage factor will be more thoroughly discussed later on in this thesis.

Summarizing, Elo++ relies on three fundamental assumptions which made it outperform its competitors, namely:

- most chess games are played at tournaments at which players of similar skill play each other.
- playing white is advantageous.
- some smoothing must be applied to iron out irregular performance.

Fortunately they are expected to carry over to an application in football team rating, as discussed above. Maybe with the exception of the first assumption, which the addition of a calibration step like described in the Sonas rating system might help mitigate. Furthermore its rating capabilities are expected to be the best of the presented systems so far, since it won the Kaggle competition mentioned earlier.

3 Additional insights

Although the discussed rating systems, and most notably the Elo system, have been successfully implemented, the question rises what can still be improved about them. One of the motivations of this thesis was the realization that the traditional rating systems take into account only very little data about the actual games that they base their ratings on. The simple idea being that, depending on the domain the systems are implemented in, additional data exists about the players and games that is expected to correlate with the outcome of games. Instead of only trying to model wins and losses by abstract teams or players, it is not a stretch to claim that modelling tennis players could be very different from modelling football teams, since both applications carry their own set of circumstances that influence the outcome of games. For example, one could imagine that the surface a tennis game is played upon (grass, gravel or hard court) might be a big influence on the outcome of the game. However this parameter might not even exist in an implementation concerned with football teams. Taking this idea a bit further, it can even be stated that a rating system in the context of tennis played in the professional Australian league can have different modifiers compared to tennis games played in the professional English league, since the temperature might be a bigger influence in the Australian setting then it might be in the English setting, for example.

The next step is to utilize specialized implementations of rating systems for each specific application. In this section several modifications are presented for the use of rating systems in the particular context of Dutch amateur football.

3.1 Home field advantage

The existence of home field advantage in football is confirmed by several studies [22] [21]. While the exact amount of the advantage should ideally be determined per team, its existence is statistically proven in football. The cause for this difference in performance on away or home fields is argued to be found in the ability to (subconsciously) sway referees and game officials when making game influencing decisions and, to a certain extend, the size of the spectator crowd.

3.2 Goals scored

While traditionally not covered by rating algorithms, the difference in goals scored between teams in a game is a parameter to take into account. Implementations of non-traditional rating systems have incorporated this parameter with some success [16], and intuitively it is quite clear that a game with a final score of 8-0 displayed a larger difference in playing ability between opposing teams than a game with a final score of 2-1.

3.3 Geographical location of football club

The influence of the geographical position of a football club can be broken down into several factors. The first one is the size of the city or town the club is based in. Generalizing the findings in a study about university related American football teams, increasing the size of the potential applicant pool, increases the number of players signing up, which increases the performance of the highest teams in clubs [20]. It should be noted that this statistic should not be misinterpreted into meaning all teams of a club get a performance boost with an increase of applicant pool size, since players of lower levels are still accepted into lower level teams at the same club.

The second factor of influence in the geographical position of football clubs can be found in the socio-economic background of the players. Generalizing a study done on national football team performance and the socio-economic background of the nation they represent, a positive correlation can be found between the economic prowess of a nation and the performance of its national football team [23]. Intuitively, the more money a club invests in its facilities, the more growth individual players will show, and the better its teams will perform.

Lastly the performance of a club is dependent on the playing ability of its competition. Supporting evidence of this claim suggests that the performance of cyclists improves when confronted with competitive opponents [17]. This kind of research is suspected to not have been done in the particular field of football analytics, because it is more or less impossible to create a control group. Intuitively, however the study does translate to football, since one can imagine that players learn the most when playing against better teams. Teams belonging to clubs located in areas with a lot of other clubs therefore would display stronger playing ability and growth than teams from clubs located in areas with fewer competing clubs. Additionally Elo++ successfully included a factor modelling competition strength, which is another clue for its merit [24].

3.4 Closeness

Like discussed in the section about Sonas' calibration sequence, an analysis of closeness is included in this research. The idea is that teams that operate in a secluded set of teams have a rating that can be over- or understated, when compared to a team outside of the secluded group, since a specific rating inside the group can signify a different playing strength than the same rating outside of the group. To identify this flaw, an approximation of the closeness measure from graph theory is calculated for all teams. The closeness measure describes the average number of steps it takes to go from a certain node in a graph to any other node in that graph. By calculating the closeness for all nodes, it becomes possible to quantify how well connected each node is compared to each other node. The data set that is used in this thesis can be viewed as a graph as well, where the teams form the nodes and the games they play form the connections between them. Since this graph is very large, an approximation was made to shorten the run time of the closeness analysis. This approximation

can be described as follows: instead of checking each possible path between every two nodes to find the shortest path between them, certain 'bridge nodes' were identified. These bridge nodes all have the special property of having played games in more than one region (The KNVB divisions are divided into six regions, such that teams do not have to compete against teams that are located very far away). This property makes the bridge nodes uniquely able to connect the different regions. The approximation of the closeness then consists of calculating for each node, the average distance to the closest bridge nodes that connect with all regions that can be connected to from this region. The entire path is no longer being searched, but only the path from each node to its relevant bridge nodes.

To illustrate: say there are 4 regions A, B, C and D, where A is connected to B and C, and C is connected to D. If the closeness of team 1 in region A needs to be calculated, the distance to the nearest bridge teams (one distance to a bridge team that connects A to B and one distance to a bridge team that connects A to C) needs to be calculated and averaged (in this case there were two bridges, so the summed distances need to be divided by two). The resulting average distance approximates the closeness for team 1.

Since teams within regions are generally fairly well connected to each other (between 2-5 handshakes) this approximation sequence can be calculated much faster than the complete approach where all nodes have to be visited. However, the proposed approach using bridge nodes has a strong greedy nature, since it always and only takes into account the bridge nodes that are closest to the node from which it starts to search and not all paths to all other nodes are explored.

Approximations for the closeness measure usually involve leaving large numbers of nodes out of the analysis. Typically, the depth of the search is limited for each node. However, since the closeness within the regions is fairly similar for all regions and the majority of teams is going to be outside of the current region when calculating the closeness for a particular team, the distance to the bridge nodes matters most in this context, which is why this approximation variant was chosen. The algorithm can be described using the following pseudocode:

Algorithm 3 Closeness approximation

for all t in teams **do**
  Find region r, where t is located in.
  Find regions R, which are connected to r.
  $N_{bt} = 0$
  for $targetR$ in R **do**
    Find nearest bridge team, bt that connects $targetR$ and r, using a breath first search starting from t, exploring all opponents.
    Find distance d_{t-bt}, between t and bt.
    $N_{bt} = N_{bt} + 1$
  end for
  $closeness_t = (\sum d)/|N_{bt}|$
end for
return all $closeness_t$

4 Data

For this project historical data will be used, containing all standard amateur teams in the Netherlands of the competition seasons 2013-2014, 2014-2015, 2015-2016, 2016-2017 and the first half of season 2017-2018. This data set contains all results of the first teams as signed up by their clubs and notably does not contain results of lower teams (also called 'reserve' teams). The teams are divided over 6 regions, each with up to 5 levels, where level 1 has the best teams, and level 5 has the lowest level teams for that region. This data set contains 1804 teams that are present over all seasons, with an average of 2172 teams competing each season, a total of 2376 unique teams and 81,342 unique games played over the course of these seasons.

Originally, the plan was to include data from all amateur teams in the Netherlands, including so called 'reserve' teams. However, the KNVB ended up not willing to provide this data. Since the data corresponding to 'standard' teams was made readily available online by *hollandsevelden.nl*, it was decided to run the experiments for this research project on that data set. The set of standard teams consists of all football teams in the Netherlands that are considered the best team of their football club. These teams operate in their own competition. A larger data set generally brings forward better models, but considering the number of unique teams and amount of played games, the standard teams data set is viewed as sufficient to train rating systems on. It is expected that the majority of findings carries over to the competition of reserve teams, or any other football competition for that matter. The exact influence of specific variables might however be different. Like the argument made in Chapter 3, any competition would benefit from a tailor made rating system.

The central bureau of statistics (Centraal Bureau voor Statistiek, or CBS) was used as a second source to complement this data set. This Dutch government agency provides independent research, which in turn provides reliable data about many aspects of the Netherlands and its people. More specifically, the data from two studies has been used, namely one containing the average disposable income per household per zip code area in 2014 in the Netherlands and one containing the population count per zip code area in 2016 in the Netherlands. This data was combined with the data set containing the information about the performance of the standard football teams.

4.1 Scraping method

As mentioned above the majority of the data set was compiled out of data as provided by *hollandsevelden.nl*. Data about all standard football teams concerning the past four seasons as well as the current season were scraped from their publicly accessible pages. This data includes: team names, game results, number of goals scored per game, the address of teams and the level and region a team operated in during any given season. The website is constructed in such a way, that all relevant information about a specific team during a specific season can be found on a page dedicated to said specific team. Therefore, the needed

information could be accessed by visiting all these pages for all teams for all seasons and recording the mentioned variables by parsing the HTML pages.

The CBS has data from census studies publicly available for download on their website [7]. The studies on average disposable income per household in 2014 and the population count per zip code area in 2016 were available for download as an excel file. Therefore minimal parsing was necessary, and the data from the census studies was fairly easy to combine with the team information, because the address of teams was part of the scraped data.

4.2 Data set general metrics

To get a general feel for the data set, a few descriptive statistics will be discussed in this section.

Table 1: Number of teams

Unique teams in data set	2,376
Unique teams in all KNVB competitions	63,514

In Table 1, the number of teams in the data set that is used for analysis is offset against all Dutch football teams according to the KNVB in 2014 [2]. So the data set at hand is only describing a fairly small sample of all available data. One would think that a better rating system could be trained by including these teams into the analysis, since more training data generally leads to better trained models. But, considering the standard teams, which are contained in the data set that is used for analysis in this study, are all fairly similar to each other and the teams that are not included in this data set might be operating under very different circumstances, it is expected that a generalized model will perform worse. Instead of adding these teams into the same model, a completely new model should be created, with its own parameter values, to create the best fitting rating system for several distinguishable subsets of teams (for example teams with players under 14 years of age could benefit from having their own model).

In Table 2 the average number of standard teams per region per season is displayed. Noteworthy are the size of region West-2 and Zuid-2. While the other regions all contain more or less 400 teams, West-2 and Zuid-2 contain only 287 and 301 respectively. The calibration of the rating according to the closeness of the teams especially in these regions will become important, since they naturally have fewer teams to compete with, within their region.

The content of the last column of Table 2 can be somewhat confusing, however the idea behind it is fairly simple: it contains the number of teams per region that played a season in another region (also called bridge teams, as discussed in Section 3.4) during the five seasons of which the data is present for this research. This information says something about the closeness of the teams

within the regions. If there are very few teams in a region that changed regions over the five past seasons, it means the region is fairly secluded, and the teams within it rarely play against teams that have played against teams in different regions. For regions with a high number of bridge teams, the opposite is true. Notably the Noord region is fairly secluded and the Zuid-1 region is fairly well connected, when comparing them to the other regions.

Table 2: Average number of teams per region per season

Region	Teams	'Bridge' teams
Noord	380	28
Oost	428	72
West-1	375	95
West-2	287	85
Zuid-1	404	144
Zuid-2	301	72

Table 3: Number of goals per game and opponents per season

Region name	goals per game	goal difference per game	opponents per season
Full Data set	4.1	2.1	10.7
Noord	4.3	2.2	10.7
Oost	4.1	2.1	10.7
West-1	4.3	2.1	10.6
West-2	4.1	2.1	10.8
Zuid-1	4.0	2.0	10.8
Zuid-2	3.9	2.0	10.3

In Table 3, three measures can be found that tell something about the distribution of teams over the regions. If the average number of goals per game in a region is much higher in a specific region, it could be argued that the playing style in that region is very different. Or if the number of opponents per season differs greatly between regions, the closeness could be affected. Fortunately all regions generally follow the data set wide averages, where Zuid-2 is slightly trailing behind in both measures. However with the differences being so small, the influence is considered negligible. The same can be said of the average goal difference per game. No large differences exist between the regions, only region Noord is slightly more productive than the other regions, but again the difference is so small its influence is considered negligible.

5 Experiments

This chapter is divided into several sections, each of which details how the experiment is built up for the different rating systems. All rating systems are implemented in Python 3.5, using the original publications of the different rating systems as guidance for the formulas that were used.

All rating systems incorporate a step where new teams are awarded a start rating. In different contexts, new teams are able to play against any opponent, which allows the rating of the new team to quickly adapt to their true playing ability. However, because all teams only play against a small subset of other teams which are not chosen at random or by the team itself (they are chosen based on the performance of teams in the previous seasons), it would not be a good idea to give the same start rating to all teams at the start of the first season in the data set. For example, the following situation could occur if such an approach was used: let us define two teams: team A and team B, that are both competing for the first time in season 2013-2014, as are all other teams, since the data set starts at this season. This means none of the teams have a rating yet, and they are all awarded the same start rating. team A is competing on the highest level, against teams with the highest playing ability in the data set and team B is competing at the lowest level, competing with the teams that show the lowest playing ability in the data set. Team A performs mediocrely and has therefore not gained or lost any rating points after the first season. Team B on the other hand performed very well and has won a lot of rating points at the end of the first season. If the rating system were a leading parameter in the decision of what level team A and team B would play next season, team A would be demoted and team B would be promoted. This would allow the rating of team A to increase again if it showed to perform better at the lower level and it would allow the rating of team B to decrease again if it showed that they performed worse at a higher level. After a few seasons the ratings would then have approximated the true playing skill of the teams. However the rating of the teams was not a considered parameter when the decision was made which team should be promoted and which team should be demoted. The effect of which is that team A could have a rating of 1200 after season one, and team B could have a rating of 1500 after season one, but team A would remain at the highest level and team B would remain at the lowest level. This results in neither team being able to correct their rating and team A keeping a score of 1200 and team B keeping one of 1500, even though team A might be better than team B.

It is therefore necessary to adapt the start rating of the teams at the very first season in the data set to a rating representing a better estimation of the playing ability of a team than the standard start rating. This counter acts the described problem because team A from the example would have a higher rating at the end of season one then team B. The idea being that team A should have a higher estimated start rating because it plays against high performing teams and team B should have a lower estimated start rating because it plays against low performing teams. Both implementations of Elo and Glicko2 have

been adapted in the described way, by awarding different start ratings to teams from different levels. The Elo++ algorithm already incorporates a procedure to counter act this problem, by influencing the rating of a particular team by the average rating of the teams that it plays against (see Section 2.2.1) so it is not necessary to make any changes in the awarding of start ratings for this rating system.

5.1 Elo

The implementation used to create the Elo rating is analogous to the implementation described in Section 2.1.2. A set value for the K factor of 26 was chosen. Other implementations, most notably the implementation used by FIDE for the rating of chess players define adaptable K factors, such that new players can adjust their rating more quickly with a higher K factor and players that have played more games have a lower K factor since their rating is considered to be representing their actual playing ability better. A non-adaptive K factor was used in this implementation, because the teams receive a start rating in line with the playing level they are operating at, at the beginning of the first season in the data set. Therefore some information is already present in the start rating about the playing ability and there should be no need to quickly adapt the start rating, since it is already representing the actual playing ability a lot better than a set start rating for all teams would.

The start rating is defined as follows:

$$r = 1500 - (level * 100) \tag{11}$$

So each new team is awarded 1500 rating points, minus 100 times the level that it operates at. Thus teams in the highest level receive 1400 points, and teams at the lowest level receive 1000 points. The values have been derived using a grid search. This technique from machine learning can be summarized as a brute force way of determining the best parameter values by training models on a training set using many possible combinations of parameter values and comparing the performance of the resulting model on the test set. The division of the data set into the training set and test set for this analysis is the same as the division used for the rest of the experiments and is further explained in the next chapter.

Since it is customary to give new teams a rating of 1200, it was decided that regular teams, with middle of the road performance, should receive a rating of 1200. In the current data set, these teams should play at level 3 (out of the 5 possible levels, since that is the middle level). Assuming the skill difference between the consecutive levels is the same, the following rating difference between adjacent levels were tried: 0, 50, 75, 100, 125, 150, 200, 250 and 300. A standard K-value of 10 was used to compare the performance. Out of these options, a rating difference of 100 between adjacent levels resulted in the best performing model. After this analysis, all natural numbers between 10 and 40 were tried

as K-value in combination with the resulting model from the previous analysis, of which a K-value of 26 resulted in the best performing model.

5.2 Glicko2

The implementation used to create the Glicko2 rating is again exactly the same as described in Chapter 2 (Section 2.1.3). The start rating is, using similar reasoning as for the adaptable start rating in the Elo system, adaptable instead of the customary 1500 points.

The start rating is defined as:

$$r = 2100 - (level * 200) \tag{12}$$

Each new team is awarded 2100 rating points, minus 200 times the level it operates at. This means the teams at the highest level receive 1900 points and teams at the lowest level receive 1100 points. The choice of the particular values for the parameters have been chosen in a similar fashion to the parameters in the Elo system, namely though a grid search sequence, where the parameters were chosen that resulted in the best performing model on the test set.

This time it is customary to give new teams a rating of 1500. In a similar fashion as the analysis done for the Elo system, start rating differences of: 0, 50, 100, 150, 175, 200, 225, 250 and 300 between adjacent levels were tried, of which a difference of 200 between adjacent levels resulted in the best performing model in combination with an RD of 350 and a volatility of 0.06.

It is customary to choose $RD = 350$ and $volatility = 0.06$, but because the start rating is adaptable it is argued that such values should also be adaptable. The relatively high values for RD and volatility are such, because new teams should be given the opportunity to change their rating quickly so that it represents their true playing ability quicker. However because of the nature of the data set, the start rating is a little bit more uncertain, since it is possible for teams to be assigned to a level that is really not in line with their playing ability. Therefore $RD = 400$ and $volatility = 0.07$ are chosen as start values for these parameters so that this uncertainty is reflected. Note that both will change during the rating periods, unlike the K factor.

Again, the values for RD and volatility were found using a grid search. All combinations of the following values were tried: RD = 250, 300, 350, 375, 400, 425 and 450, and volatility = 0.04, 0.05, 0.06, 0.07 and 0.08.

5.3 Elo++

The Elo++ implementation used for the experiments is directly taken from its publication [24] and exactly as described in Chapter 2 (Section 2.2.1). Fortunately this algorithm includes a recursive sequence which relates the rating of a team to the ratings of the teams around it. Because of this it is not necessary to make the start rating adaptable like in the Elo or Glicko2 implementations. In the original publication the number of iterations for the algorithm to go

through, is set to 50, but the author admits that the ratings would converge starting from iteration 5. Because of computational limitations, the number of iterations for the algorithm for this experiment is set to 5. The λ factor, which denotes the influence of the average rating of the opponents of a team on the rating of that team was set to 0.2, as opposed to 0.77 in the literature, and the learning rate was set to 0.602, which is the same as in the literature. These values were found in a brute force grid search and are expected to be different for each implementation.

To start the grid search off, the value of λ was set to 0.77, and the learning rate to 0.602 after the values in the originally published implementation. After this, the following values for λ were tried: 0, 0.15, 0.2, 0.25, 0.3, 0.5, 0.77, 1 and 1.25, out of which 0.2 resulted in the best performing model. The following values were tried for the learning rate, while keeping the value for λ at its original value of 0.77: 0.4, 0.5, 0.55, 0.602, 0.65, 0.7 and 0.8, of which 0.602 resulted in to best performing model.

5.4 Additions

The discussed additions from Chapter 3 were implemented in two different ways. All three algorithms make use of an expectancy function which calculates the expected outcome of a game purely based on the rating the teams had before the game was played. Factors for goals difference, home field advantage, population density, disposable income and club density were implemented by altering the expectancy functions of the respective algorithms. The idea to implement them this way came from the original publication of the Elo++ algorithm and from a publication by Glickman on adapting the Glicko system for use in the context of American football teams in the NFL [24] [16]. Both publications use a similar set up, where the expectancy function is altered by a factor that represents the home field advantage, or in the case of the original publication of Elo++, white advantage for chess games. These parameters are all included by adding weights to the expectancy function outcomes. The original expectancy functions look like this:
for Elo:

$$E = \frac{10^{R_a/400}}{10^{R_a/400} + 10^{R_b/400}} \tag{13}$$

for Glicko2:

$$E(\mu, \mu_j, \phi_j) = \frac{1}{1 + e^{(-g(\phi_j)(\mu - \mu_j))}} \tag{14}$$

for Elo++:

$$E = \frac{1}{1 + e^{r_o - (r_i + \gamma)}} \tag{15}$$

They were altered to include the sum of the various weights like so:
for Elo:

$$E = \frac{10^{(R_a + \sum_{i=1}^{n}(W_i))/400}}{10^{(R_a + \sum_{q=1}^{n}(W_q))/400} + 10^{(R_b - \sum_{q=1}^{n}(W_q))/400}} \tag{16}$$

for Glicko2:

$$E(\mu, \mu_j, \phi_j) = \frac{1}{1 + e^{(-g(\phi_j)((\mu + \sum_{q=1}^{n}(W_q)) - (\mu_j - \sum_{q=1}^{n}(W_q))))}} \tag{17}$$

for Elo++:

$$E = \frac{1}{1 + e^{(r_o - \sum_{q=1}^{n}(W_q)) - (r_i + \sum_{q=1}^{n}(W_q))}} \tag{18}$$

Where W is a list of length n, containing the weights for the different factors.

The weights are all constructed in the same way, namely: the value of the relevant parameter for the current team, minus the value of that same parameter for the opposing team, times the factor that was established in a grid search to add the most information value. So for example, the weight for the home field advantage in the Elo algorithm for a team that is playing a home game, is calculated as follows: The home field advantage (the average number of times the team won a game at their home field that season) is deducted with the away field advantage of the opposing team (the average amount of times the opposing team won a game on an away field that season), times the factor that was established during a grid search to add the optimal amount of information value. This grid search consisted of individually optimizing the parameter values while keeping the other parameter values at zero. The parameter values that were tried can be found in Appendix 3.

The adaptation to include the closeness of teams entails a bit more, since the hypothesis is that teams with a high closeness value (in other words, the teams that are fairly isolated) are systematically underrated, and teams with a low closeness value are systematically overrated. Therefore, after training the systems or algorithms each season, the rating of poorly connected teams (those with a high closeness value) is raised with the average amount of points difference between the well connected and badly connected teams. The tried parameter values for this adaptation can also be found in Appendix 3.

The amount of influence a particular factor should have or, in the case of closeness, the boundary that decides which teams are considered badly connected and which are considered well connected are all up for training and have been established during a brute force grid search to be optimal.

6 Testing rating results

Testing how well rating scores represent the true playing ability of teams is not a trivial task. After all, an absolute measure of playing ability does not exist. Fortunately all rating systems incorporate a step where the expected outcome of a game is calculated based on the difference in rating for the two teams that are playing against each other. This expectancy function can then be used to assess how well the rating system is able to predict game outcomes based on team ratings. By assessing how well each expectancy function is able to predict game outcomes, the performance of the different rating systems can be compared. To facilitate this comparison of expectancy functions and because of the time sensitive nature of the data set, the data set, consisting of a total of 114,139 games, was divided into a training set, test set and validation set. The data from the final season is used as validation set (6,736 games) and the data from season 2016-2017 is used as test set (26,065 games), while the remaining seasons populate the training set (81,338 games). The rating systems are trained on the training set, the performance of the parameter weight values assessed on the test set, and the validation set is used to guard against overfitting on the data set.

In projects like these, the results of the automation are usually compared to the results as they would be without the automation. Unfortunately there is no such non-automated version of rating implemented at this time, apart from the level that teams are classified at. Each region is divided into 4 or 5 playing levels. This classification can be seen as a very crude form of rating, and can be compared to the output of the tested rating systems. This will then give a general idea of how the output of the rating systems compare to the existing levels, however it is expected that not a lot of information will be gained, since the division into levels is so much less precise when compared to ratings on an individual basis.

Therefore the performances of the rating systems need to be evaluated in another way, using more precise metrics. For this, metrics from machine learning will be used. Because output of the expectancy functions of the rating systems in this project can be seen as similar to outputs of regression models, a metric from regression model analysis will be included, namely the Root-Mean-Square Error (RMSE), which provides the standard deviation for the predicted values versus the actual values in a test set. However, the output of the expectancy function can also be seen similar to classification model output, since the only actually possible game outcomes are win (1), draw (0.5) or loss (0). Therefore it could be argued that the output of the expectancy functions has to be rounded to the values 0, 0.5 or 1. Using this philosophy the use of metrics from machine learning classification would make more sense, which is why the accuracy, precision, recall and f1 score will be included into the assessment of the expectancy functions as well.

To produce results with which the performance of the expectancy functions can be tested and compared, all home games and all away games in the validation set are put through the expectancy function of each algorithm. Effectively, this means all games are assessed twice by each algorithm, since each home game is an away game of another team. Because all expectancy functions only have three possible outcomes per game (team 1 is expected to win and team 2 is expected to lose, team 2 is expected to win and team 1 is expected to lose, or both teams are expected to draw), the produced metrics on which the algorithms are scored and compared are the same as if each game was evaluated once. Had this not been the case and for example both teams could have been predicted to win a game, evaluating each game twice would have given a more accurate representation of the performance of the algorithms, since both teams cannot actually win the same game, so at least one prediction would have been false. This false prediction could then have gone unnoticed if it was only assessed whether the home team was expected to win or not for that specific game.

For reasons that will be explained in Section 7.3, it is hard to model drawn games, using just the expectancy function of the rating systems. Therefore, drawn games are always classified as correctly predicted games, no matter the predicted game outcome.

7 Results

In this chapter the results from the various experiments will be presented. In the first two sections the performance and differences between the three rating systems (Elo, Glicko2 and Elo++) will be discussed. In the third section the validity of the proposed implementation specific parameters is discussed on the basis of the trained algorithms. Finally the performance gains or losses of the addition of the parameters like goal difference and home field advantage into the rating systems will be discussed. To keep the results uniform and easily comparable the rating systems will be trained on the training set as defined in Chapter 6, and the presented results come from comparing the output of the different rating systems on the validation set as defined in Chapter 6 as well.

7.1 Rating systems performance

The results for the three rating systems on the validation set (consisting out of 6,736 games), while being trained on the training set can be found in Table 4 (the corresponding confusion matrices can be found in Appendix 4). This table contains the accuracy, f1 score, precision, recall and RMSE for each rating system. The top 25 for the different rating systems including the rating for each team based on the training set can be found in Appendix 1 and a histogram for each rating system is depicted in Appendix 2. These appendices give the reader an idea of the differences between the outputs of the three rating systems and the differences between the ratings of the teams within each system.

Table 4: Performance of Rating systems without additions

	Acc.	F1 won	F1 lost	Pre. won	Pre. lost	Rec. won	Rec. lost	RMSE
Elo	0.656	0.655	0.656	0.655	0.657	0.656	0.656	0.491
Glicko2	0.657	0.656	0.658	0.657	0.658	0.656	0.659	0.472
Elo++	0.651	0.652	0.654	0.650	0.652	0.655	0.658	0.453

As can be seen in Table 4, the different rating systems perform surprisingly similar on the validation set. The only two differences that are worth noting are found in the RMSE and the Accuracy. The difference in RMSE value is probably due to the fact that the predictions of the expectancy function of the Elo system produces predictions that are a little less close to the actual outcomes. Recall that actual game outputs can only take on the values 0, 0.5 and 1, where 0 notates a loss, 0.5 a draw and 1 a win. The expectancy functions produce an output in the range of 0 to 1, which can have any imaginable value in that range.

When looking at the accuracy, the performance of Elo++ stands out, because it is surprisingly lower than the accuracy of the other two algorithms, while it was expected to be the best performing algorithm.

When looking at the other classification performance measures however, it is striking how similar the different algorithm performances really are. The Glicko2

algorithm shows the best performance across the board, which is somewhat unexpected, but the difference in performance is relatively small.

In general, an accuracy score of about 65% does not sound fantastic, but it does show a correlation between the rating and the game outcomes. To put it into perspective, Glickman created a Glicko rating model for NFL games [16], which produced a prediction accuracy of 58.2% on individual game outcomes. The completely different domains and data sets and the fact that drawn games are always marked as correctly predicted, while there are virtually no draws in the NFL make the comparison between his model and the currently presented models unfair, but it shows that the performance of table 4 is fairly similar to this model from the literature.

The systems expectancy functions do not perform uniformly over the validation set, but they do better at some predictions than others. Notably the outcome of games that involve a relatively large difference in rating between the two opponents are generally easier to predict, than games that involve a relatively small difference in rating between the opponents. To visualize the performance of the expectancy functions of the three rating systems over the games that are easier to predict, the performance is measured only on the 10% of games with the biggest difference in rating between the opponents in Table 5.

Table 5: Rating system performance without additions on 10% of games with biggest rating difference

	Acc	F1 won	F1 lost	Pre. won	Pre. lost	Rec. won	Rec. lost	RMSE
Elo	0.744	0.740	0.745	0.739	0.747	0.742	0.744	0.524
Glicko2	0.728	0.721	0.731	0.722	0.731	0.721	0.732	0.516
Elo++	0.732	0.734	0.728	0.724	0.739	0.745	0.717	0.458

The classification metrics (all metrics, excluding the RMSE) in Table 5 show a much kinder image. Interestingly the accuracy of the Elo algorithm is higher on this subset of samples than the accuracy of Glicko2 and Elo++.

A second interesting thing is going on, when looking at the RMSE in particular. By increasing the performance threshold, the RMSE was also increased for both the Elo and Glicko2 algorithm, which is unexpected, because one would expect the predictions that the algorithms were most certain about to be closer to the actual game outcomes, which should then decrease the RMSE value. However, the opposite seems to be the case. The RMSE value is sensitive to outliers, which could be why it is seemingly performing worse. By only looking at the bolder predictions, the wrong predictions that remain are also more wrong (the deviation of the predicted value versus the actual game outcome must be larger by default for a wrong prediction). This could then make the RMSE take on a larger value. Interestingly, the more complex the model, the less the RMSE is affected. It is therefore probable that the predictions made by the Elo system are the most bold, followed by the Glicko2 system and the predictions by the Elo++ system are most careful (bold being predicting close to 1 for a win and careful being only predicting slightly higher than 0.5 for a win).

7.2 Average proportional frequencies of game outcomes

The performance of the three algorithms can be visualized by plotting the average proportional frequencies of the game outcomes against the expectancy value outputted by the algorithm. These plots, in other words, show how many won, lost and drawn games on average, have a specific expectancy value. An ideal model would show a plot where all the lost games get an expectancy value of 0, all drawn games get an expectancy value of 0.5 and all won games get an expectancy value of 1. These plots then give some idea of the performance of the rating algorithms. As can be seen in Figure 1,2 and 3, elo++ shows more 'plateauing' for extreme values of the expectancy function than the other two algorithms. This plateauing comes closer to emulating the ideal performance than the straight lines that the plots for the Elo and Glicko2 algorithms show. Proportionally, most lost games occur at the lowest expectancy values, and the least lost games occur at the highest expectancy values, and vice versa for won games, for all three plots, which is exactly as expected.

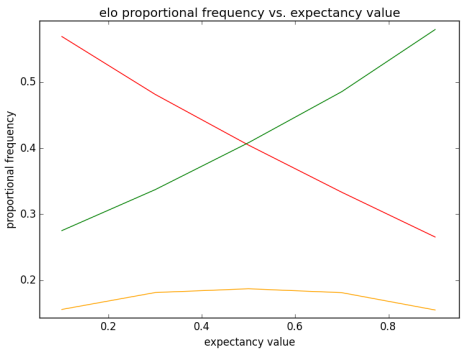


Figure 1: E function Elo

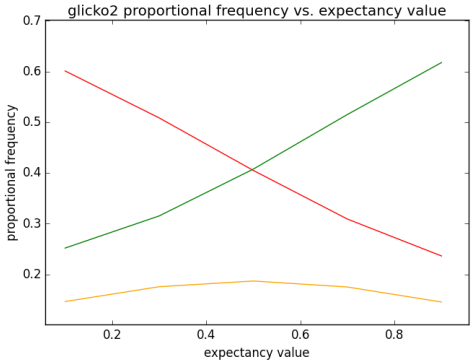


Figure 2: E function Glicko2

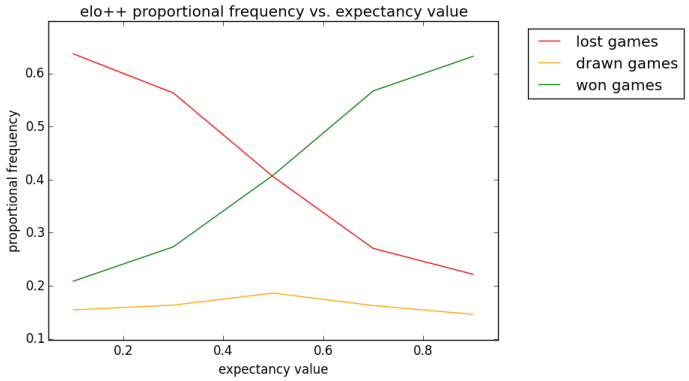


Figure 3: E function Elo++

Drawn games show different behaviour though. A small rise in the amount of drawn games can be seen in the middle ranges of the expectancy values, but it is very clearly not the case that when opponents are equal in rating, they are most likely to play a draw. A draw is always possible, but the chance of playing a draw slightly increases when the rating of opponents is closer. The consequence of this, is that it is very hard to model draws, because rating difference, and therefore the outcome on the expectancy functions of the rating systems, has only a minimal relation to the chances of the game ending in a draw. It will come as no surprise then, that it is particularly difficult for the rating systems to predict drawn games. Therefore the expectancy functions do not predict draws, but rather let outputs of 0.5 and above predict wins or draws and outcomes of 0.5 and below predict losses or draws. Draws will then be automatically classified as correctly predicted outcomes. This is reflected in classification metrics (accuracy, f1, precision and recall), which get a slight boost, but not in the RMSE, since it allows for prediction values to be of a continuous nature.

7.3 Validity of additional insights

In this section the statistical validity of the additional insights that are discussed in Chapter 3 will be investigated. This will be done by assessing whether the value of a specific parameter correlates with a standard Elo rating. Only the Elo rating will be used for this analysis, since the differences between the three algorithms is so small, and the Elo rating is easiest to run and interpret.

7.3.1 Home advantage

To confirm the validity of this parameter in the data set, and because the correlation between two groups of samples is tested, a statistical z-test has been carried out, comparing the average game outcome of home games with the average outcome of away games for each team. To get some idea of the distributions of the home win percentage and away win percentage, the probability distribution of both variables is displayed in Figure 4. It shows they both roughly follow a normal distribution.

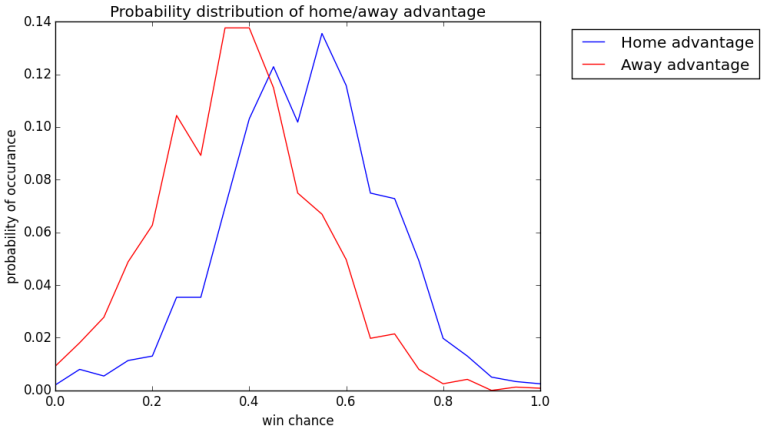


Figure 4: Probability distribution of home/away advantage

To test the hypothesis that the average home advantage is indeed higher than the average away advantage, a single tailed unpaired z-test is carried out, where the null hypothesis states that both averages are equal and the alternative hypothesis states that the average home advantage is greater than the average away advantage. The following formula was used to calculate the z-test:

$$z = \frac{(\bar{X}_2 - \bar{X}_1)}{\sqrt{\sigma_1^2/n + \sigma_2^2/n}} \tag{19}$$

Where X_2 denotes the mean value for home win percentage, X_1 the mean for away win percentage, and σ_2 the standard deviation corresponding to the home win percentage, σ_1 the standard deviation corresponding to the away win percentage and n denotes the sample size. In Table 6 the result of this z-test as well as some metrics about away games and home games can be seen.

Table 6: Home field advantage statistical test

Number of teams	2376
Average win percentage home	47.3 %
Average win percentage away	35.1 %
Average difference win percentage home/away	12.2 %
Standard deviation of home advantage	0.143
Standard deviation of away advantage	0.139
Z-value result	29.84

Looking up the z-value in a table with critical values for single tailed z-tests, the z-value calculated here drops of the table since it is so high. The chance that both home and away win percentage distributions are actually the same is therefore close to 0. It can confidently be stated that the venue of a game (either home or away) is indeed correlating with the outcome of a football game in this data set, since teams tend to win more games at their home field.

7.3.2 Goals scored vs. rating

To validate the correlation between the number of goals scored and the rating difference between opponents, a statistical z-test will not suffice, since, contrary to the validating of home field advantage, this time the correlation of two variables is being questioned. A Spearman correlation coefficient can be calculated to illustrate the existence or absence of a correlation between the difference in goals and the difference in rating for a particular game in the data set.

To facilitate the validation of this parameter a standard Elo rating was trained on the entire data set. After this, the two variables (goal difference and rating difference) were plotted and the Spearman correlation coefficient was calculated for the two variables, where the null hypothesis states there is no association between the two variables, and the alternative hypothesis states there is. Note that the final rating was used to compare the teams, and not the rating as it was for each team at the time that the game was played. This way we are testing whether a large difference in goals scored in a particular game predicts a team to become much better or not.

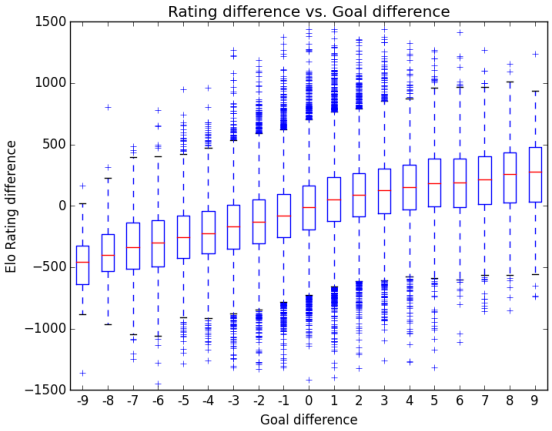


Figure 5: Elo rating difference vs goal difference

In Figure 5, the discussed rating difference is plotted versus the goal difference for all games in the data set. Note that the information is presented from the perspective of the team playing at home. So a goal difference of -8 means the home team lost by a difference of 8 goals. The same is true for the differ-

ence in rating. A rating difference of 250 means the home team has a rating advantage of 250 points over the away team. In Figure 5 a clear rise in rating difference can be found when the goal difference grows. A Spearman correlation of 0.39 and a corresponding p-value close to 0 ($p < 0.01$) indicate the null hypothesis can be rejected for $\alpha < 0.05$ and that there is indeed a correlation between the number of goals scored during a game and the difference in rating. The correlation itself can be designated as having a medium effect (according to Cohen [10]) as a value of 0.39 was found on a scale of -1 to 1. Intuitively this relationship makes sense, since one would expect that a large victory where the opponent was defeated with 8 goals difference, would predict that the two competing teams are not of the same strength.

7.3.3 Population density vs. rating

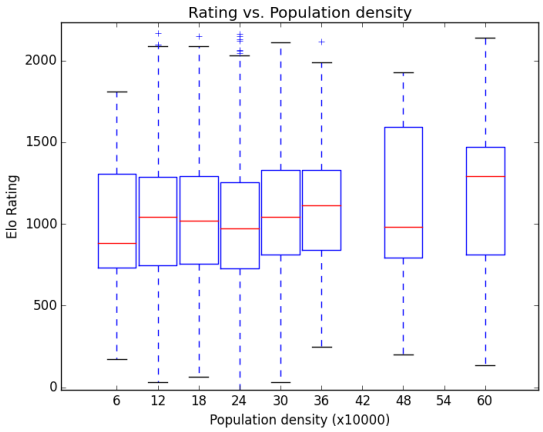


Figure 6: Elo rating vs. population density

In Figure 6, the Elo rating of all teams is visualized versus the population density of the area where the team is located. The data is divided into bins of size 60,000, where the first box contains the teams with populations in their area below 60,000, the second contains teams with populations below 120,000 and above 60,000, the third between 180,000 and 120,000 and so on. The population density of the area the club is located in is defined as the amount of people that live in the same zip code area as the zip code area where the club is located, according to information from Centraal Bureau voor Statistiek [3]. The Netherlands is divided into areas with a zip code of four numbers and two letters, where similar zip codes are located near each other (in other words: there is a logical order in zip codes). For this analysis, the population density of all zip codes with the same two starting numbers were added up, such that a much larger area is associated with it. Since a single exact zip code would be too specific, an averaging over a larger area is desirable.

As can be seen in Figure 6, there seems to be no clear correlation between Elo rating and population density. The Spearman correlation factor (again with a null hypothesis stating there is no correlation between the variables and the alternative hypothesis stating that there is) corresponding to this relationship is in line with this conclusion, with a value of 0.034 and a p-value of 0.11, indicating the alternative hypothesis should be rejected for $\alpha < 0.05$.

7.3.4 Club density vs. rating

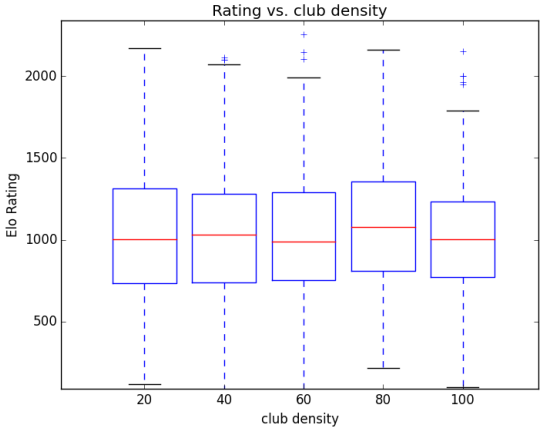


Figure 7: Elo rating vs. club density

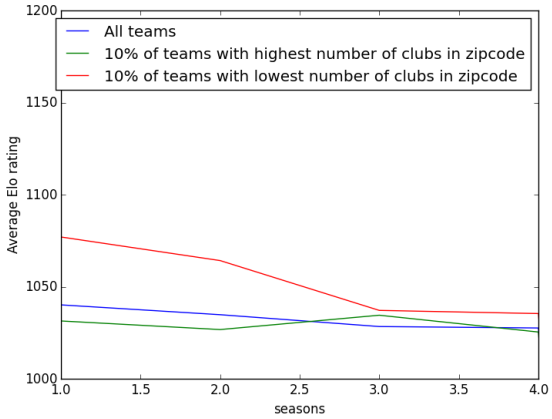


Figure 8: Elo rating vs. club density over time

In Figure 7, the Elo rating of all teams is visualized against the club density, where the data is divided into 5 boxes of size 20, namely the teams with a club density below 20, visualized by the first boxplot, the teams with a club density between 20 and 40 by the second, and so on. The club density per team is defined as the amount of clubs located in the same zip code area as the team itself. Again the zip code area is defined as the combination of the zip codes with the same two starting numbers, to increase the size of the area associated with the variable.

Similarly to the outcome of the analysis on population density, the correlation between the two variables is far from evident. Again a Spearman correlation test can be conducted using the null hypothesis stating there is no correlation between the variables and the alternative hypothesis stating that there is. The Spearman correlation factor associated with these two variables is 0.023 with a p-value of 0.28, solidifying this suspicion and indicating the alternative hypothesis should be rejected for $\alpha < 0.05$.

An interesting effect can be noticed in Figure 8, which displays the rating gain of teams over time. It shows that teams from areas with few football clubs probably tend to play at slightly too high a level on average. Because of this,

they generally get overrated at the beginning of season one and then gravitate towards the average Elo rating of the other groups.

7.3.5 Disposable income vs. rating

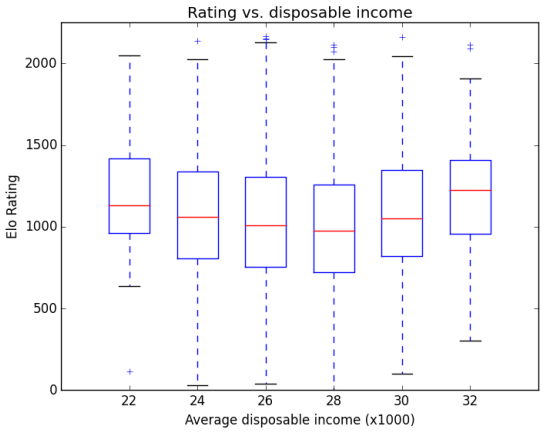


Figure 9: Elo rating vs. disposable income

In Figure 9 the Elo rating of all teams is visualized against the average disposable income of households in the area where the team is located. The average disposable income is defined as average income in euros before taxes per household in 2014. The average disposable income is published per zip code area, much like the population density, by Centraal Bureau voor Statistiek [1]. Again, the zip code areas with the same two starting numbers have been added up for this analysis, to get a larger area to be associated with it, and consequently a better averaged view of the area that a team is located in. The teams have been binned into 6 bins of size 2,000, with the first bin containing the teams with an average disposable income of lower than 22,000 euro per year, the second bin containing the teams with a disposable income between 22,000 and 24,000 euro per year and so on.

In Figure 9 a parabolic correlation seems to exist, where the teams with a disposable income of below 28,000 euro are experiencing a negative correlation with the Elo rating, while the teams with a disposable income of 28,000 and above seem to experience a positive correlation with the Elo rating. When examining the Spearman correlation factor, for the null hypothesis stating there is no correlation between the variables, and the alternative hypothesis stating that there is, a value of 0.11 can be found for the teams with a disposable income of 27,000 and above, which points to a weak correlation, with a corresponding p-value of 0.022, indicating the alternative hypothesis should be accepted for $\alpha < 0.05$. The Spearman correlation factor for the teams with a disposable income below 27,000 however, suggests that a correlation does not exist, with

a value of -0.04 and a corresponding p-value of 0.053, indicating the alternative hypothesis should be rejected for $\alpha < 0.05$.

7.3.6 Closeness vs. rating

The approximated closeness for each team was calculated using the approach described in Chapter 3 (Section 3.4), and visualized in Figure 10. The closeness is offset against the Elo rating of the corresponding team. The teams are again divided into several bins, this time of size 0.5, where the first bin contains the teams with a closeness of maximally 1, the second bin contains the teams with a closeness between 1 and 1.5, and so on.

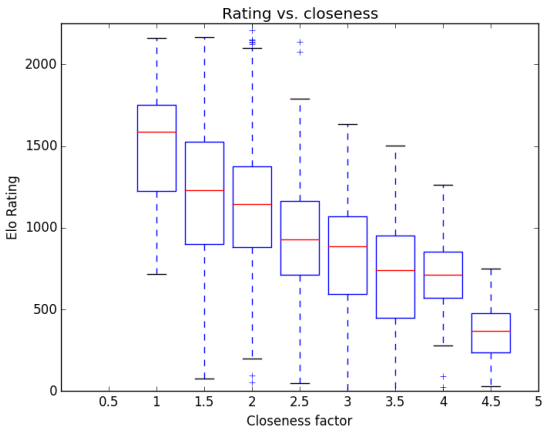


Figure 10: Elo rating vs. closeness factor

A clear pattern can be identified in Figure 10. The corresponding Spearman correlation factor (where the null hypothesis states there is no correlation between the variables, and the alternative hypothesis states that there is) is -0.39 with a p-value close to 0 ($p < 0.01$), which indicates the alterantive hypothesis should be accepted for $\alpha < 0.05$ and a negative correlation with medium effect exists. The more handshakes a team needs on average to get to other teams, the worse its rating is on average. Which intuitively makes sense. A lower closeness factor means the corresponding team played against more different opponents. Where as a team that only ever competes with the same opponents would not get the opportunity to learn as much as the better connected team. Secondly, something completely different could be happening. Namely the rating of the badly connected team could be understated, because it is not adequately updated. Like discussed in Chapter 2, a calibration sequence to calibrate the badly connected teams to the well connected teams would be necessary to counter act this problem.

7.4 Performance of additional insights

The same performance results from Table 4 for the different rating systems can be found in Table 7, this time the results of the three rating systems includes factors for the parameters: home field advantage, goal difference and closeness.

Table 7: Performance of Rating systems with various additions

	Acc.	F1 won	F1 lost	Pre. Won	Pre. lost	Rec. Won	Rec. lost	RMSE
Elo, home adv.	0.656	0.656	0.656	0.656	0.657	0.656	0.656	0.496
Elo, goal diff.	0.661	0.659	0.661	0.660	0.661	0.659	0.662	0.507
Elo, closeness	0.655	0.655	0.655	0.655	0.656	Section 0.655	0.655	0.492
Glicko2, home adv.	0.657	0.656	0.658	0.657	0.658	0.656	0.659	0.476
Glicko2, goal diff.	0.658	0.657	0.658	0.657	0.658	0.657	0.659	0.476
Glicko2, closeness	0.658	0.657	0.659	0.658	0.659	0.657	0.660	0.472
Elo++, home adv.	0.653	0.654	0.651	0.652	0.654	0.656	0.649	0.452
Elo++, goal diff.	0.654	0.655	0.653	0.653	0.655	0.657	0.651	0.471
Elo++, closeness	0.654	0.655	0.652	0.653	0.655	0.657	0.650	0.453

The values of the parameters such as home advantage and goal difference were multiplied with a factor, specific for each parameter, to control the influence that the parameter has on the expectancy functions of the algorithms, as described in Section 5.4. The values of the factors can be found in Appendix 3.

The three parameters that did not show a clear correlation in the analysis in 7.3 can indeed be classified as more or less irrelevant. The models which included factors for these parameters did not show an increase of performance, which is why they are not included in Table 7.

The general performance increase of the three algorithms with parameters included is also fairly underwhelming. The algorithms showing the most gain increased their performance with 0.5%, while most did not increase more than 0.1%.

The goal difference parameter had a clear correlation with winning and losing games in the analysis of Section 7.3. An increase in performance was also what was found when including a factor for this parameter in the calculations for all three algorithms. When added to the Elo algorithm it seemed to have an especially large positive effect on the accuracy of the predictions of the expectancy function.

The home advantage parameter also showed a correlation in the analysis of Section 7.3, which is again what was found when a factor for this parameter was included in the calculations of the algorithms. However the algorithms with this parameter included showed only a marginal increase in performance.

On the performance of the algorithms with a factor for closeness included, compared to the algorithms without such a factor, some words need to be said. After the grid search for the best weight of the influence of the closeness factor, it turned out that only a relatively small performance increase could be made by calibrating the very best connected teams (those with an approximated closeness

value of 1 or less) down by 5% of the average difference in rating between that group of teams and the rest of the teams for all algorithms. This means that only the very top connected teams are slightly overrated in this data set. It is expected that the calibration of the data set becomes more influential when a longer period of time is encapsulated by the data set. Meaning that, if the data set included data from much longer ago (say 10 seasons ago, instead of the maximal 4 seasons ago in the current data set) the difference in closeness between the best connected teams and the worst connected teams could potentially be much greater, and the amount of overrating or underrating as a result of this, could also potentially be much greater.

In Table 8, the performance of the algorithms with all the parameters added at the same time can be found. All algorithms show some improvement, but, in a similar manner to the previous analysis, only increase about 0.1% in accuracy each. This seems rather low, and indicates substantial overlap in the information value of the different parameters exists, since the gained accuracy is not only less than the sum of the accuracy gains in the analysis of the individual parameters, but about the same as the accuracy of any of the models trained on individual parameters.

Table 8: Performance of Rating systems with all additions

	Acc.	F1 won	F1 lost	Pre. won	Pre. lost	Rec. won	Rec. lost	RMSE
Elo	0.658	0.657	0.658	0.658	0.658	0.657	0.659	0.515
Glicko2	0.658	0.657	0.658	0.658	0.658	0.656	0.659	0.480
Elo++	0.652	0.653	0.652	0.651	0.654	0.655	0.650	0.470

8 Conclusion and discussion

Considering the first sub research question, which was: **How does the performance of relevant existing rating algorithms compare?**

The performance of the Elo, Glicko2 and Elo++ algorithms without the additions of home advantage, closeness and such, is surprisingly similar to each other. It was not the case that the more complex algorithms, and especially Elo++, perform better than the simpler algorithms.

The are a number of possible causes for the surprisingly disappointing performance of the Elo++ algorithm, one of which is the restriction in number of iterations that the algorithm was given, which was substantially lower than the number of iterations recommended by the original author of the algorithm (5 instead of 50 iterations). This could have prevented the ratings from converging properly during the execution of the algorithm. Secondly the non-adaptive start rating could have been a problem, even though the Elo++ algorithm contains a sequence that influences the rating of all teams by the rating of all other teams that it plays against. Implementing an adaptable start rating much like what was implemented in the Elo and Glicko2 algorithms did not prove successful. The fact that Elo++ employs a version of time weighting could also have been a problem. Because only 5 seasons of data were used, as opposed to the 100 time periods the original publication mentions, the time weighting might be too crude in this application. It is clear that more research needs to be done to investigate this disappointing performance.

The second sub research question was: **Can rating systems be improved by adding implementation specific parameters, such as factors that model home field advantage or club density in the area where a team is located?**

Based on the analysis of the various proposed parameters in Chapter 7, it seems like certain parameters, namely: home field advantage, goal difference and closeness do contain information corresponding to the performance of teams, while the others, namely: population density, club density and average disposable income in the area do not. However, training the algorithms with factors for the additional implementation specific parameters without overfitting on the data set proved to be challenging. Fortunately, this could be identified with the use of a validation set. The added value of the different parameters is smaller than the expected performance considering the analysis done in Section 7.3. The home field advantage, goal difference and closeness seem to have some influence on the performance of the three rating systems, while population density, club density and average disposable income of the area do not.

The absolute performance gained by adding these parameters to the three algorithms was not substantial. When creating future implementations of rating algorithms, one should always consider the amount of effort it takes to add these parameters, and the actual performance that can be gained. However, this lack of gained performance can be subscribed partially to the way the experiment is set up. Namely, by evaluating the performance on the validation set, as opposed to the test set, the performance measures go down somewhat. Because

the games in the validation set happen after the games in the test set, some information is missing in the rating used to predict the game outcomes of the games in the validation set. This idea is substantiated even more by the fact that all algorithms performed better on the test set, than on the validation set. Simply presenting the performance on the test set is not an option though, since there is no guarantee against overfitting. It is customary in the field of machine learning to 'roll' over time sensitive data by adding the test set to the training set after testing, training the algorithm on this new training set and using that model as input for the validation step. Perhaps this would be a preferred set up for future studies on the same subject. Note that this would not change how well the models are fitted, but it would show the performance of the same models more positively. Another option is to abandon the classification style of measuring performance all together, and measuring the performance with the help of analysis similar to the analysis done in Section 7.2, using distributions such as displayed in Figure 1 to 3.

The third sub research question was: **Can rating systems be improved by adding a calibration sequence to level out differences in the closeness of teams?**

All three algorithms showed improvement when calibrated for the closeness. However, only when the most well connected teams were calibrated downwards, which would mean that those teams are somewhat overrated. It is expected that including data from more seasons would increase the relevance of the closeness factor, since more variation in closeness would be possible.

Looking back at the main research questions posed at the beginning of this document, which was: **Is it possible to improve the performance of rating algorithms in the context of Dutch amateur football using implementation specific knowledge?**

Rating algorithms such as Elo, Glicko2 and Elo++ can be successfully trained on the type of input data available for this project. However the resulting ratings differ only very slightly from each other and therefore the advantage of more elaborate rating algorithms over simpler systems, such as the Elo algorithm, is debatable, especially since more elaborate algorithms can show black box-like behavior and are generally harder to understand. When adding implementation specific parameters to the analysis such as home field advantage and goal difference, the fit of the rating systems on the actual playing performance does improve, but the performance increase is heavily depending on the correlation of the parameter and the rating score, and even in the case of the best correlating parameters, the performance gain was only marginal at best. The addition of these parameters can only be recommended when absolute performance of the rating system is of key importance, or when a very high correlation between the parameter and the rating is known to exist. Addition of a closeness calibration sequence to the rating algorithms showed some success. It is expected that the influence of this parameter increases when the range of possible values for that parameter is extended, for instance by including data from much longer ago into the data set.

Looking back at the original motivation for this research, which was the partial automation of the creation of amateur football leagues and divisions, it can definitely be recommended that rating algorithms are incorporated in this process, and it is by no means necessary to start right away with a very sophisticated and adapted algorithm. If anything, this research showed that original implementations of rating algorithms such as Elo and Glicko2 already perform relatively well in the context of amateur football.

For future studies, the measuring of the performance for the different algorithms in this study could be improved. The set up in this study showed an image of the performance that was probably too bleak. This can be prevented by including the test set into the training set after testing, training the algorithm on the new training set and validating this model on the validation set. Another option to be investigated is to employ a different style of evaluating models all together. Furthermore, the performance of the Elo++ algorithm can possibly be improved by allowing it to run for more iterations and by including data from more time periods into the training set. Thirdly, it might be worthwhile to investigate the correlation of other metrics from graph theory with rating data, such as the data used in this study. Numerous graph metrics exist, such as the centrality and connectivity, which are related to the closeness metric in different ways, which could also have a correlation with the rating of teams.

Above all, in this study it has become clear that ratings of teams based on team performance alone, without inclusion of data per player, is inherently imperfect. Because players change teams throughout their career it seems logical that teams vary in strength per season, and even per game, based only on this ever changing body of players. To continue research in this area and improve the presented algorithms adding this information to the analysis is of key importance. Relying solely on this player based information though, would lack parameters that are team and club based. Parameters such as closeness of the team in the graph of all teams and home advantage could easily be overlooked, which is where the current study showed most of its success. Therefore a combination of these two approaches is worth investigating. Of course this is only possible when data on per player basis is recorded in the data set, which is not always the case. For those data sets that do not include this per player information, the trained algorithms such as described in this study will be difficult to improve.

References

[1] (2015). Besteedbaar inkomen per postcodegebied, 2004-2014. https://www.cbs.nl/nl-nl/maatwerk/2017/15/besteedbaar-inkomen-per-postcodegebied-2004-2014 [Accessed 2017-12-14].

[2] (2015). KNVB jaarverslag 2014/2015. http://knvb.h5mag.com/jaarverslag [Accessed 2018-1-5].

[3] (2016). Bevolking per viercijferige postcode op 1 januari 2016. https://www.cbs.nl/nl-nl/maatwerk/2016/51/bevolking-per-viercijferige-postcode-op-1-januari-2016 [Accessed 2017-12-14].

[4] (2017). Chapter 9 of the ATP rule book. http://www.atpworldtour.com/-/media/files/rulebook/2017/2017-atp-rulebook_chapter-ix.pdf [Accessed 2017-11-05].

[5] (2017). NCAA devision 1 men's basketball championship principles and procedures for establishing the bracket. http://www.ncaa.com/content/di-principles-and-procedures-selection [Accessed 2017-11-05].

[6] (2017). Regelingen promotie en degradatie. https://www.knvb.nl/assist/assist-bestuurders/wedstrijdzaken/competities/promotie-/degradatieregeling [Accessed 2017-12-05].

[7] (2017). Statline, public online data bank of CBS. https://opendata.cbs.nl/statline/#/CBS/nl/ [Accessed 2017-12-14].

[8] Cintia, P., G. F. P. L. P. D. and Malvaldi, M. (2015). The harsh rule of the goals: data-driven performance indicators for football teams. *IEEE conference on Data Science and Advanced Analytics proceedings*, pages 1–10.

[9] Cintia, P., R. S. and Pappalardo, L. (2015). A network-based approach to evaluate the performance of football teams. *Machine learning and data mining for sports analytics workshop at ECML/PKDD conference 2015, Porto, Portugal*.

[10] Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates.

[11] Elo, A. E. (1978). *Rating of chess players, past and present*. Arco Pub.

[12] Glickman, M. E. (1995). A comprehensive guide to chess ratings. *American Chess Journal*, 3:59–102.

[13] Glickman, M. E. (1999). Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics*, 48:377–394.

[14] Glickman, M. E. (2001). Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics*, 28:673–689.

[15] Glickman, M. E. and Hennessy J, B. A. (2018). A comparison of rating systems for competitive women's beach volleyball. *Italian Journal of Applied Statistics*, to appear:t.b.a.

[16] Glickman, M. E. and Stern, H. S. (1998). A state-space model for national football league scores. *Journal of the American Statistical Association*, 93:25–35.

[17] Jo Corbett, Martin J. Barwood, A. O. R. T. and Dicks, M. (2012). Influence of competition on performance and pacing during cycling exercise. *Medicine and Science in Sports and Exercise*, 44(3):509–515.

[18] Keizer-Mittelhaëuser, M.-A. (2014). *Modeling the Effect of Differential Motivation on Linking Educational Tests*. PhD thesis, Tilburg University.

[19] Lames, M. and McGarry, T. (2007). On the search for reliable performance indicators in game sports. *International Journal of Performance Analysis in Sport*, 7(1):62–79.

[20] Murphy, R. G. and Trandel, G. A. (1994). Relation between a university's football record and the size of its applicant pool. *Economics of Education Review*, 13(3):265–270.

[21] Nevill, A. M. and Holder, R. L. (1999). Home advantage in sport. *Sports Medicine*, 28(4):221–236.

[22] Nevill, A. M., N. S. M. and Gale, S. (1996). Factors associated with home advantage in english and scottish soccer matches. *Journal of Sports Sciences*, 14(2):181–186.

[23] Robert Hoffmann, L. C. and Ramasamy, B. (2002). The socio-economic determinants of international soccer performance. *Journal of Applied Economics*, V(2):253–272.

[24] Sismanis, Y. (2010). how i won the 'chess ratings–Elo vs the world competition'. `https://arxiv.org/pdf/1012.4571.pdf` [Accessed 2017-11-05].

[25] Sonas, J. (2005). Chessmetrics: a weighted and padded simultaneous performance rating. `http://www.chessmetrics.com` [Accessed 2017-10-10].

[26] Van Haaren, J. and Op De Beéck, T. (2014). Welke rol speelt sports analytics in het voetbal van morgen? `https://kuleuvenblogt.be/2014/03/31/welke-rol-speelt-sports-analytics-in-het-voetbal-van-morgen/` [Accessed 2017-10-11].

9 Appendices

9.1 Rating of top 25 teams

Table 9: Elo, no additions

Team	Rating
VVChevremont	2133
CVVOranjeNassauG	2131
IFC	2114
ASVDronten	2099
SCBemmel	2071
SVMeerssen	2033
VVHoogland	2027
PKC'83	2027
HVCH	2005
VVOudeMaas	2001
VVNunspeet	1993
VVSliedrecht	1991
SVArgon	1990
VVSJC	1985
VVGZ	1976
VVRijsoord	1972
VVBuitenpost	1961
RKSVMinor	1960
Heidebloem	1957
SVNootdorp	1956
VVBrielle	1946
VVBalk	1935
VVSVBO	1927
Purmersteijn	1926
d'OldeVeste'54	1916

Table 9: Glicko2, no additions

Team	Rating
IFC	2100
VVHoogland	2097
VVSliedrecht	2096
VVChevremont	2037
VVNunspeet	2021
SVArgon	2020
VVSJC	2019
VVNieuw-Lekkerland	2004
Purmersteijn	2003
HVCH	2000
CVVOranjeNassauG	2000
SVNootdorp	1999
PKC'83	1987
SVMeerssen	1986
Heidebloem	1980
VVRijsoord	1972
VVSVBO	1971
Wittenhorst	1969
VELO	1962
Zwaluwen'30	1959
AlphenseBoys	1946
VVBrielle	1943
VVBuitenpost	1939
SCBemmel	1933
DeMerino's	1930

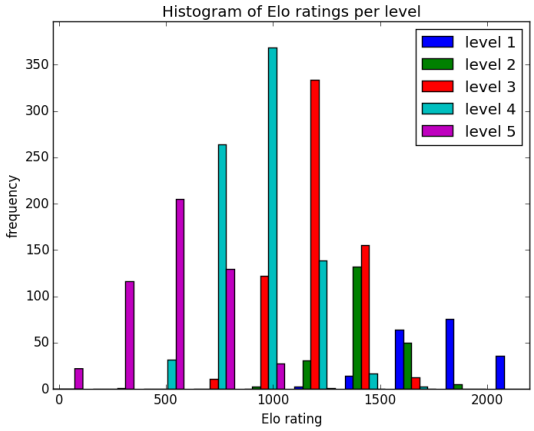
Table 10: Elo++, no additions

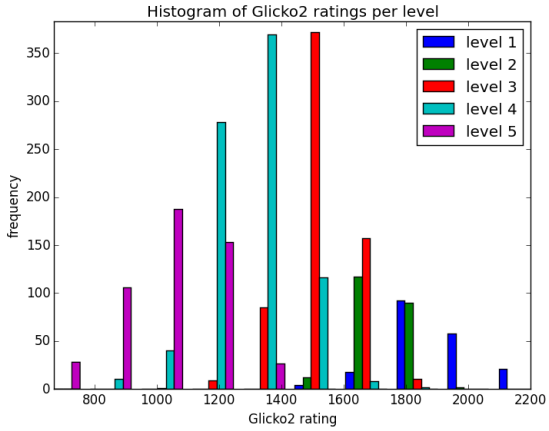
Team	Rating
VVChevremont	16.17
SVMeerssen	15.88
EHC	15.81
VVSchaesberg	15.73
Heidebloem	15.71
Wittenhorst	15.65
ZSV	15.65
RKSVMinor	15.63
VVDeValk	15.61
AMVJ	15.57
GeusseltSport	15.53
VVEijsden	15.51
Marvilde	15.51
Bekkerveld	15.50
Susteren	15.49
SSS'18	15.45
SVTOP	15.44
VenloscheBoys	15.44
SVVenray	15.43
SVDeurne	15.42
RKVVVolharding	15.40
Wilhelmina'08	15.39
SVArgon	15.39
VVSmitshoek	15.38
OirschotVooruit	15.36

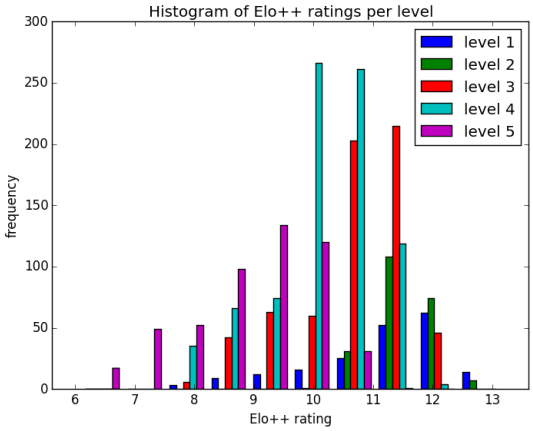
[1]

[1] At first glance a lot of differences seem to exist between the top 25s. However keep in mind that the top 40 teams could all come from different competitions and it is therefore probable that many teams never played against any other team in the top 40. This top 25 only shows a general agreement between the algorithms about which teams belong in the top 25 at all, and not so much about which specific rank each team should have.

9.2 Histogram of rating algorithms







[2]

[2]The histogram for Elo++ shows that the rating is not fully converged. Given more iterations, it is expected that the rating will converge further, however some spread over lower ratings is always expected.

9.3 Parameter values that were tried for all additions for the three algorithms (Elo, Glicko2 and Elo++)

In Table 10, 11 and 12, the parameter values that resulted in the best performing model are indicated with bold face.

In these tables, the factor values for the closeness parameter are defined as follows: the cut off value represents the closeness value that divides the well connected teams and the badly connected teams. The multiplication value represents the factor with which the average rating difference is multiplied before it is added to the rating of the badly connected nodes. The described factors are used to calculate the parameter weights as described in Section 5.4.

While executing the grid search for any specific parameter value, the other parameters were kept at 0, with the exception of the two closeness parameters, for which all combinations were tried.

Table 10: factor values for parameters in Elo system

	factor value
population dens.	-100, -50, -10, -5, **0**, 5, 10, 50, 100
club dens.	-100, -50, -10, -5, **0**, 5, 10, 50, 100
disp. Income	-100, -50, -10, -5, **0**, 5, 10, 50, 100
home adv.	-200, -175, -150, -135, -130, **-125**, -120, -115, -100, -75, -50, -10, -5, 0, 5, 10, 50, 100
goal diff.	-150, -125, -110, **-100**, -90, -50, -10, -5, 0, 5, 10, 50, 100
closeness cut off value	0, 0.5, 0.65, **0.75**, 0.85, 1, 1.25, 1.5, 2, 2.5, 3, 3.5
closeness mult. value	-0.5, -0.25, -0.1, 0, **0.05**, 0.1, 0.25, 0.5

Table 11: factor values for parameters in Glicko2 system

	factor value
population dens.	-1, -0.75, -0.5, -0.25, -0.1, **0**, 0.1, 0.25, 0.5, 0.75, 1
club dens.	-1, -0.75, -0.5, -0.25, -0.1, **0**, 0.1, 0.25, 0.5, 0.75, 1
disp. Income	-1, -0.75, -0.5, -0.25, -0.1, **0**, 0.1, 0.25, 0.5, 0.75, 1
home adv.	-1, -0.75, -0.6, **-0.5**, -0.4, -0.25, -0.1, 0, 0.1, 0.25, 0.5, 0.75, 1
goal diff.	-1, -0.75, -0.5, -0.25, -0.15, **-0.1**, -0.05, 0, 0.1, 0.25, 0.5, 0.75, 1
closeness cut off value	0, 0.5, 0.65, 0.75, 0.85, 1, **1.25**, 1.5, 2, 2.5, 3, 3.5
closeness mult. value	-0.5, -0.25, -0.1, 0, **0.05**, 0.1, 0.25, 0.5

Table 12: factor values for parameters in Elo++ system

	factor value
population dens.	-1, -0.75, -0.5, -0.2, -0.1, -0.05, **0**, 0.05, 0.1, 0.2, 0.5, 0.75, 1
club dens.	-1, -0.75, -0.5, -0.2, -0.1, -0.05, **0**, 0.05, 0.1, 0.2, 0.5, 0.75, 1
disp. Income	-1, -0.75, -0.5, -0.2, -0.1, -0.05, **0**, 0.05, 0.1, 0.2, 0.5, 0.75, 1
home adv.	-1, -0.75, -0.5, -0.2, -0.1, -0.05, 0, 0.05, 0.1, **0.12**, 0.15, 0.17, 0.2, 0.5, 0.75, 1
goal diff.	-1, -0.75, -0.5, -0.2, -0.17, -0.15, **-0.12**, -0.1, -0.05, 0, 0.05, 0.1, 0.2, 0.5, 0.75, 1
closeness cut off value	0, 0.5, 0.65, **0.75**, 0.85, 1, 1.25, 1.5, 2, 2.5, 3, 3.5
closeness mult. value	-0.5, -0.25, -0.1, 0, **0.05**, 0.1, 0.25, 0.5

9.4 Confusion matrices

Table 13: confusion matrix Elo

	True won	True lost
Predicted won	4415	2321
Predicted lost	2313	4423

Table 14: confusion matrix Glicko2

	True won	True lost
Predicted won	4411	2304
Predicted lost	2311	4446

Table 15: confusion matrix Elo++

	True won	True lost
Predicted won	4411	2371
Predicted lost	2325	4365

[3]

[3]Note that the total true won games and the total true lost games are not the same over the matrices. This is due to the fact that drawn games are always classified as predicted correctly. They can therefore sometimes pose as won games and other times as lost games. Confusion matrices for the other models exist, but have not been included, due to the lack of added informational value.