Predicting Sport Results by using Recommendation Techniques

Bram De Deyn

Supervisors: Prof. dr. ir. Luc Martens, Dr. ir. Toon De Pessemier
Counsellor: Dr. ir. Toon De Pessemier

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Department of Information Technology
Chair: Prof. dr. ir. Bart Dhoedt
Faculty of Engineering and Architecture
Academic year 2017-2018

Predicting Sport Results by using Recommendation Techniques

Bram De Deyn

Supervisors: Prof. dr. ir. Luc Martens, Dr. ir. Toon De Pessemier
Counsellor: Dr. ir. Toon De Pessemier

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Department of Information Technology
Chair: Prof. dr. ir. Bart Dhoedt
Faculty of Engineering and Architecture
Academic year 2017-2018

GHENT
UNIVERSITY

Preface

The research on the subject 'Predicting Sports Results by using Recommendation Techniques' was originally planned to be performed during the 2016/17 academic year. It was my favorite subject from the list last year when it was assigned to me and I could not find a more interesting subject this year. That is why I decided to ask the promoters for an opportunity to transfer the subject to this year. I want to thank Prof. dr. ir. Luc Martens and Dr. ir. Toon De Pessemier for giving me the chance to eventually work on this research subject. I also want to express my gratitude for guiding me through this semester, answering diverse questions and helping me in the application process for a student license at SportRadar.

SportRadar, and especially Malte Siegle, also deserve a spot in my acknowledgments. Without the data and technical support they provided me with, I could not have started this research. A data scientist without data is not really a data scientist, is he?

At last, I also want to thank my parents and some of my closest friends. They endured many moments where I unclearly tried to explain the problems I was facing or where I was mentally or physically absent. Thanks to my parents for giving me the opportunity to obtain a higher education and letting me ramble about my studies all the time. Thanks to Gert-Jan Storme for proofreading parts of my thesis. Thanks to everyone else for registering trial accounts on SportRadar or supporting me in general.

Bram De Deyn – January 15, 2018

Permission for use of content

Bram De Deyn – January 15, 2018

Abstract

This thesis research tries to maximize profit from sports betting on football outcomes in specific. Sports are mostly unpredictable and prone to human errors, with football being the worst case. A team consists of many players and there are three possible match outcomes (H/D/A). Obtaining profit would be feasible if one could correctly predict upsets and draws. A data provider was used for match summaries and probabilities, from which features were extracted. Models were optimized for accuracy as a basis for further research. Afterwards profit was the main goal. Approaches to limit losses and risks were come up with. Profitable ternary classifiers were found for each of the five considered major European leagues. No linear correlation between the accuracy and the profitability of a model was found.

Using these classifiers, a personal assistant for bettors was engineered in the form of a recommendation tool. It recommends the betting strategies and money management systems that were the most profitable in recent history and outputs the match outcome probabilities generated by the classifier.

KEYWORDS – football result prediction, recommendation techniques, profit maximization

Predicting Sport Results by using Recommendation Techniques

Bram De Deyn

Supervisor(s): Prof. dr. ir. Luc Martens, Dr. ir. Toon De Pessemier

Abstract—This thesis research tries to maximize profit from sports betting on football outcomes in specific. Sports are mostly unpredictable and prone to human errors, with football being the worst case. A team consists of many players and there are three possible match outcomes (H/D/A). Obtaining profit would be feasible if one could correctly predict upsets and draws. A data provider was used for match summaries and probabilities, from which features were extracted. Models were optimized for accuracy as a basis for further research. Afterwards profit was the main goal. Approaches to limit losses and risks were come up with. Profitable ternary classifiers were found for each of the five considered major European leagues. No linear correlation between the accuracy and the profitability of a model was found.

Using these classifiers, a personal assistant for bettors was engineered in the form of a recommendation tool. It recommends the betting strategies and money management systems that were the most profitable in recent history and outputs the match outcome probabilities generated by the classifier.

Keywords—football result prediction, recommendation techniques, profit maximization

I. Introduction

This research is a follow-up on a thesis written in the 2016/17 academic year by Robin Praet [1]. A different approach on the matter is taken. The focus lays on maximizing profit instead of accuracy and football is forecasted instead of tennis.

A lot of factors make football one of the most unpredictable sports there is. Sports in general are prone to human error; upsets are a possibility which enables the need for competition. Favorites do not consistently win every game of the league. Football teams consist of 11 base players and even more substitute players, each taking their own split second decisions. Coaches and referees also have a big impact on the team's performance and the match outcome. Home advantage is a proven concept [2].

The bookmakers' market is proven to be inefficient [2] and thus betting strategies exist for guaranteed profit in the long term. Even though arbitrage betting is still possible, the assumption is made that it is not. The risk to be caught by the bookmakers which could lead to major betting losses, is not worth the potential relatively small payouts.

II. Data exploration

A sports data provider is used. This comes with some challenges but also has its advantages. First of all, limited amounts of requests per month to collect data were available through trial accounts. Using several accounts, historical data was fetched for five major national professional European leagues: the Spanish LaLiga, the English Premier League, the Italian Serie A, the German Bundesliga and – mainly out of curiosity for the predictability of own national league – the Belgian Pro League. The collected data was cached in a local database to prevent duplicate requests, but can easily be updated with live data for the API. Each league has its matches since the 2011/2012 season fetched, which amounts to more than five seasons per competition. A couple of seasons of the international leagues – i.e. the Champions League and Europe League – were collected to improve the features explained below.

Probabilities without profit margins can be fetched from the same API. This allows to set a custom profit margin to be used during the research. After calculating margins for some of the most popular bookmakers, 7.5% is chosen as a loose upper bound.

III. Feature extraction

A lot of features were possible to be extracted from the collected data. Recent results for both teams, as well as derived features for form, season rankings, recent results separately and fatigue were immediately thought of. As much information as possible is put into these features. Wins or losses are expressed with goal differences, instead of using a nominal H/D/A feature. More complex features derived from historical statistics (e.g. ball possession, successful passes etc.) and team information (e.g. stadium, travel distance for away team, etc.) were also researched. Most features are considered for both the recent results of the home team and the away team separately, as well as for recent matches for both teams against each other (called the head to head matches).

The information gain, the correlation to the match outcome and the accuracy of a rule based one-feature classifier are calculated for each feature. The most important drivers turn out to be the most simple ones. The top 25 features for every evaluation metric only show goal derived features, e.g. form, rankings, average goals made, average goal difference, recent head to head results etc.

IV. Classifiers

Most of the commonly used classifier types were tested and support vector classifiers were found to be the most consistent to achieve higher accuracy values for each league. Models are evaluated on 20% test split basis with order preservation to prevent matches in the future to be present in the training set of classifiers. The predictor will obviously not have that that information outside of the training and evaluating phase.

League	Accuracy	Kernel	C	Reduction technique
LaLiga	55.30%	linear	0.5	PCA
Premier League	60.09%	radial	1	GainRatio / Correlation
Serie A	57.83%	linear	0.125	None
Bundesliga	51.87%	sigmoid	4.0	None
Pro League	49.52%	radial	2.0	GainRatio

TABLE I

The LibSVM parameters for the highest accuracy values per league.

League	Profit	Model	Betting strategy
Spanish LaLiga	27.43	SMO PukKernel C=8.0 with PCA	Home underdogs
English Premier League	29.48	SMO RBFKernel C=4.0	Playing the odds
Italian Serie A	17.90	LibSVM sigmoid kernel C=8.0 with PCA	Predicted safe favorites
German Bundesliga	42.00	SMO PukKernel C=8.0 with PCA	Playing the odds
Belgian Pro League	4.77	LibSVM linear kernel C=16.0 with PCA	Home underdogs

TABLE II

Final highest profits per league after executing the simulation.

A WEKA [3] LibSVM classifier was optimized to achieve precision values of up to 60% for some leagues (see Table I). Three reduction techniques were also evaluated together with the classifier: principal component analysis (PCA), a GainRatio ranker and a Correlation ranker. Each technique reduces the set of 113 features to 25 of them.

Since a precision of 54% is proposed to be the lower bound for a model to be profitable [4], hopes were good to find betting strategies that end up to be profitable in the long run for three of the considered leagues. The German and Belgian competitions were expected to be unprofitable due to disappointing precision results (51.87% and 49.52% respectively).

V. Maximize profit

The same optimization process for the LibSVM classifier is performed, but now the performance is profit instead of accuracy. LibSVM is able to use a ternary classifier, while the WEKA SMO classifier uses pair-wise classifiers to handle multi-class problems. As the researcher expected SMO to predict draws more frequently using the binary classifier, it is also included to the optimization process. Apart from the models, several betting strategies and money management systems were compared.

Betting strategies are simple rules that decide whether a match and what event of that matchup should be bet on. Five of them are compared: betting on the bookmakers' favorites, betting on the predictor's favorites (both a risky and safe version exist), playing the odds (i.e. playing on the event with the biggest difference in published and generated odds) and only betting on the underdog home teams.

A money management system calculates the stake size of the bet. Unit bets (i.e. same sized bets) and unit returns (i.e. same sized gross payouts) are simple. The Kelly ratio [5] on the other hand is more complex and compares published odds and predicted winning probabilities to come a conclusion.

The results of this optimization process were surprisingly good (see Table II). Each league has its own profitable configuration of classifier and betting strategy. The money management systems define how much risk a bettor allows for his bets. Unit bets have more risk but a higher potential profit, while unit returns and the Kelly ratio are somewhat equal in allowing less risk and lower potential payout.

Recommendation tool

The recommendation tool engineered during this research, uses this optimization process before predicting a given amount of matches. This way the best performing configuration for the recent history is used and the chances for profit are the highest. The tool also outputs the calculated event probabilities and the published odds which were basis of its predictions. Using these, an end user can make his own decisions and bet smartly.

VI. Conclusions

Each league is found to be profitable to bet on, in the assumption that the right model and strategy is chosen. The bookmaker's market is hence proven to be still inefficient, even when a high profit margin is used. A prototype recommendation tool to predict football outcomes has been created based on these results. A profitable configuration of classifier, betting strategy and money management system was found for each league. This makes the researcher assume that the predictor will be profitable in the long run, but will make incorrect predictions nevertheless.

Further work

The prototype could be made more user friendly with more predictions options and a GUI or Web Interface.

More features can also easily be thought of, but the question is whether those would be more important than the goal derived versions. These features could be weather information, coaches and managers, team formation, individual player information, fatigue due to national teams etc.

This research can also be applied to other areas. A recommendation tool for stocks to invest on can easily be derived from our prototype.

References

[1] Robin Praet, Predicting Sport Results by using Recommendation Techniques, Ph.D. thesis, Ghent University, 2017.

[2] Anthony Costa Constantinou, Norman Elliott Fenton, and Martin Neil, “Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks,” Knowledge-Based Systems, vol. 50, pp. 60–86, sep 2013.

[3] Eibe Frank, Mark A Hall, and Ian H Witten, “The WEKA Workbench,” Morgan Kaufmann, Fourth Edition, pp. 553–571, 2016.

[4] Martin Spann and Bernd Skiera, “Sports forecasting: A comparison of the forecast accuracy of prediction markets, betting odds and tipsters,” Journal of Forecasting, vol. 28, no. 1, pp. 55–72, jan 2009.

[5] J. L. Kelly, “A New Interpretation of Information Rate,” Bell System Technical Journal, vol. 35, no. 4, pp. 917–926, jul 1956.

Contents

Chapter 1

Introduction

This research is a continuation on Robin's work from last year [1] on the subject of predicting sport results. He specifically chose the sport tennis to forecast results for. Instead of trying to improve his results by finding better models or features, a different approach and focus was taken on the matter. I decided to switch up sports and chose football – also called association football or soccer in other regions of the world.

So why did I choose football? It seemed like the obvious choice to not continue Robin's research on tennis and take on a bigger challenge. Even though football is more popular – in both watching and betting –, it is obviously less predictable than tennis. I decided to use his research and methodology to improve the current progress in the academic world of forecasting football match outcomes.

1.1 Sports betting

First of all, it has to be pointed out that sport results are subject to human actions. Referees make mistakes. Players make mistakes. Injuries occurs and accidents happen. The favorite team or a key player in a match can have an off day. The underdog can be extra motivated to play against a much stronger team. Both teams start the match with the intention to take the three points home. If the top dogs always won, there would not be a competition in the first place. Luck also plays a considerable fact in the outcome of a match. Many goal chances do not necessarily convert into a win for the team – since the other team's goal keeper could be the hero of the day.

The point is that whatever the great idea behind the machine learning technique is, quasiperfect accuracy values or huge profits cannot be expected from a sports predictor. It should be possible to predict upsets – i.e. the underdog surprisingly winning the game – to some extent, but if everyone could predict the underdog winning it would not have been called an upset in the first place. Machine learning and recommendation techniques are statistics applied to practical life problems. It is not magic. No bettor will keep winning and never lose a bet. Even the most consistent team of the league loses or ties a game from time to time.

1.1.1 Predictability of football versus tennis

Several factors contribute to the fact that football is less predictable than other sports. One would be able to predict tennis match outcomes with a binary classifier, but a ternary classifier has to be build for football matches to handle the multi-class prediction problem. Apart from one of the two teams winning, a draw is also a possible outcome. This makes football outcomes a lot more unpredictable, as the amount of classes just increased with 50%. Added to that is the fact that the classes are not balanced at all (see Figure 1.1). Draws and away wins occur less. Draws are also hard to predict and many classifiers will just ignore draws in their predictions, capping the accuracy at about 60 to 70%. This upper bound can then only be reached when all upsets are correctly predicted.

Secondly, football is not an individual sport like tennis. Teams consist of eleven base players and even more substitute players can be sent onto the field during the duration of the game. Key players are sometimes given a rest for certain matches too, increasing the risk for upsets in games which are less important than usual. And let's not forget about the transfer periods twice a year. Teams are constantly changing their player base which leads to possible performance changes from time to time. The English football club Leicester is a good example. They shocked the world by winning the 2015/16 season Premier League title, after closing the ranking table a year before that. As of this moment (today's date is January 15, 2018) Leicester is in eighth place and has absolutely no chance of winning the title anymore – Manchester City is on a major winning streak.

Next, home advantage is proven to be a real concept [2, 3]. In tennis, the crowd has less effect on the players.

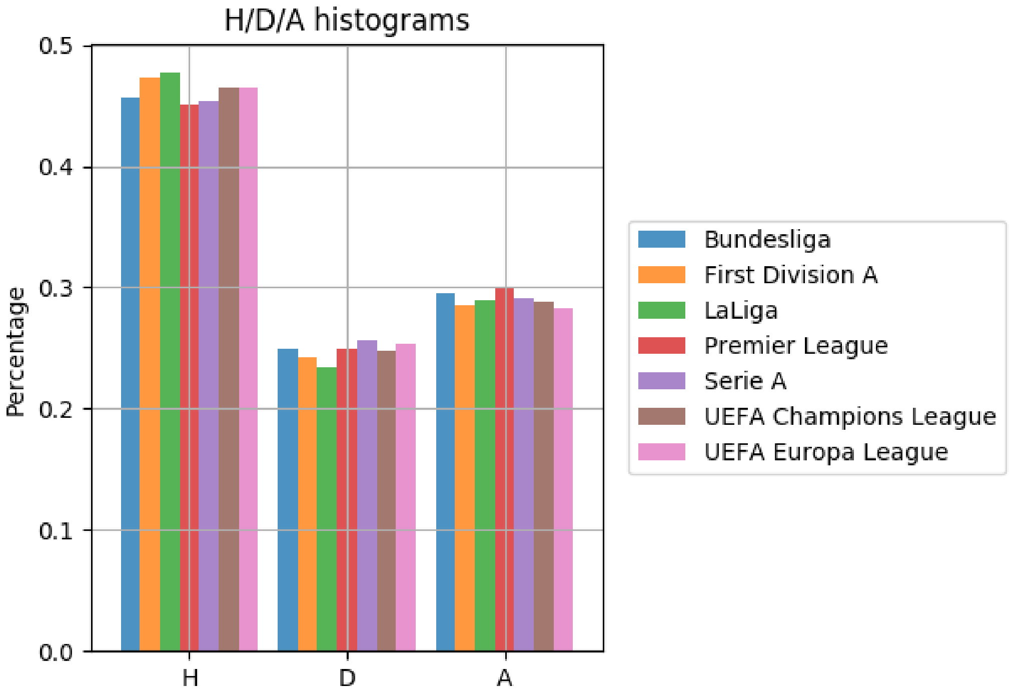


Figure 1.1: A plot with histograms depicting the occurrences of home wins, away wins and draws for several leagues.

Everyone has to be silent when the players are battling it out in a rally. Players should never be distracted by the crowd next to the tennis court. In football on the other hand, the crowd has a big impact. Some teams are feared for their home crowd and away games in those stadiums are always a big challenge.

Lastly, several other people surrounding the football field have an impact on the result. Apart from the players on the field each in control of their own actions and split second decisions, a coach guides them with tactics and has an important role in the performance of the team. Look at all those coaches being sacked every season due to a poor team performance. Added to that, the referee always takes some key decisions – whether those are correct or incorrect is left aside – that influence the end result. A wrongly handed out red card or penalty can give a match a major twist, without it being someone else's fault other than the referee's. Admitted, the effect of the referee's decisions has been decreasing lately due to the introduction of video assistance, but it is a whole other story in tennis. Coaches and referees have little to no effect. Coaches are not allowed to directly interact with their players (with the Davis and Fed Cup being the exceptions of the rule) and players can overrule a referee's decision using Hawk-Eye technology if they think it was incorrect.

Even though this section might look like a cover up for the results later in this thesis, nothing but facts were stated and the hopes for the reader skimming through this thesis report were simply kept realistic. Sports are unpredictable – especially football – so approaches to minimize risk and maximize profits have to be come up with.

1.1.2 Odds

To understand sports betting, a couple of technical terms have to be explained. First of all, probabilities are transformed into odds on which can bet. These odds represent the chance of the bet succeeding, but also indicate the possible profit if one were to stake money on that match outcome. During the years different strategies have been used to express odds all over the world. To explain these representations clearly (source [4, 5]), it is temporarily assumed that no bookmakers' profit margin is involved.

- Fractional odds represent the net profit you would make from a bet. Winning a bet with odds 2/1 (pronounced as two-to-one) implies that the underdog surprisingly won with a 33% percent change of winning. You would triple your money; getting back the amount of money you bet (also called the stake) and receiving a net profit of twice that amount. A possibility to gain half the bet stake – which means you would be betting on the top dog of the matchup with a 67% percent chance of winning – would be represented as 1/2 odds. This representation is also called the British or UK odds.

- Moneyline odds have two sides; one for the favorite and one for the under dog of the matchup. When the odds are positive, they represent the amount of net profit you would win on a 100$ bet. 2/1 odds would be represented as +200. On the other hand, negative odds show how big the stake has to be to gain a net profit of 100$. A top dog bet with odds 1/2 can hence also be expressed as -200. These are also called the American odds.

- In Europe, odds are mostly expressed in their decimal form. The odds 1/2 equal 1.50, while 2/1 equals 3.00. When you multiply these odds with the stake, it represents the gross profit you receive if you were to win the bet. This means that you would have to subtract the stake to get the net profit. In the course of this thesis, decimal odds will be used together with actual probabilities (in percentages). These representations are the easiest ones to use because they are each other's inverse.

Some useful and common probabilities have been calculated for later use (see Table 1.1).

Probabilities	0.10	0.25	0.50	0.75	0.90	p
Fractional odds	9/1	3/1	1/1	1/3	1/9	$\frac{1-p}{p}$
Moneyline odds	+900	+300	±100	-300	-900	$-\frac{100p}{1-p}; \frac{100(1-p)}{p}$
Decimal odds	10.0	4.0	2.0	1.33	1.11	$\frac{1}{p}$

Table 1.1: Probabilities and their gambling odds. The last column shows how to calculate the different odds.

1.1.3 Bookmakers

What about the profit margin bookmakers use? Without it, they would not make any money. Bookmakers usually take a cut before paying out the winning bets. That is why a fifty-fifty bet is not represented as two 2.00 odds. Depending on the size of the bookmakers' profit margin, published odds will probably be somewhere around 1.90 for both. The sum of the probabilities representing the outcomes of the bet will always be over 100% (e.g. 105.26% in our case). The excess is called the bookmakers' profit margin and will be their profit in the case of a balanced book.

Just like everyone, bookmakers cannot predict the outcomes of each bet perfectly themselves. A balanced book is thus hard to come by and that is why bookmakers sometimes pay out more than what was staked in total or earn more than was expected. To overcome this, bookmakers allow their odds to dynamically change in proportion to the amount of money staked on the possible outcomes to obtain a more robust book. Bettors may have inside information or personal experience, which is indirectly used by the bookmakers to derive better odds, i.e. closer to the true event probabilities. But what if this information is incorrect? This might open an opportunity to find faulty odds and possibly make profit out of it.

Due to the fact that bookmakers dynamically change their odds, different bookmaker websites offer (slightly) different odds. Selecting the best odds for a matchup can thus be done to maximize your profit. Even a guaranteed net profit is possible. Imagine for example an event with two possible outcomes for which odds over 2.00 for both events on two different websites can be found. No matter what the outcome will be, a positive net payout will be achieved if one were to bet on both outcomes. This phenomenon is called arbitrage betting. The returns are always relatively small and bookmakers usually notice this before payouts, which unrelentlessly results in a canceled bet, possible big losses and maybe a banned account. Added to that, bookmakers have been synchronizing their odds with each other to minimize the effects of arbitrage betting. The conclusion is that, as big amounts of money have to be used to get substantial profit, the risk is simply too high. There is hence no point in researching this. An assumption is made that arbitrage betting is not possible. Maximizing profit should be done by winning bets fair and square.

1.2 Current research

A lot of research has been done on the subject of predicting football results. Match results and scores have been modeled in an uncountable number of ways using all kinds of features since as early as the eighties. Maher [2] proposed a Poisson model, where home and away scores are modeled independently. A bi-variate Poisson model was proposed to improve the fit, because he found that the scores of both teams are not independent. A correlation coefficient of about 0.2 was estimated. Using variables for attacking strength, defending strength and home advantage he was able to get a good fit for the differences in scores. Lots of improvements on this model have been published since then, like incorporating time [6], giving weights to different kinds of scores [6] etc. Nevertheless, this paper is widely considered and cited to be the founding of football match result forecasting. Home advantage, for example, has been supported by everyone doing research into this subject [3]. The crowd effect is huge [7] and even the effect of biased referees has been proven [8, 9]. Referees are said to sometimes – unconsciously or not – prefer the home team, the title candidates or their personal favorite team of the matchup.

1.2.1 Statistics

Back in the eighties, little data and statistics were saved about sports matches. One was lucky if he was able to find the scores and key players of a football match. Since the creation of the world wide web and its massive growth, every little historic detail of a football match imaginable can be looked up online. Many research papers compare features and models to figure out what the dependencies between the historic statistics and match results are. Features and their relationship to the eventual match result have been studied over and over [10, 11, 12, 13]. The purpose of many of these researches is to find drivers of a match's outcome. These drivers then expose the weaknesses of a team, which then again can be used by a team's coach to focus on certain tactical aspects during training sessions. For this thesis research, features can be extracted out of historical match data, trying to mimic these drivers. Finding teams' strengths and weaknesses will result into better predictions.

1.2.2 Model comparison

Neural network, NaiveBayes, Random Forest and Multinomial Logistic Regression classifiers succeed to achieve accuracies up to 55% [14, 13]. It is shown that accuracies above 54% can lead to guaranteed net profit [15], if bettors use an adequate betting method and money management system – assuming that the bookmakers use a moderate profit margin. Bayesian Networks have been constructed and proven to work really well too [16]. Furthermore, a bivariate Poisson model [17], bivariate Weibul count models for score distributions [18], Random Forests [13] etc. have been used and proven to work as good forecast models.

It is necessary to consider both accuracy and profit to get the full informative and practical sense of a model's performance [19, 16]. To calculate the possible profit a model could generate, its predicted instance probabilities have to be compared to the published odds. To be able to perform this comparison, Shin probabilities [20] should be considered to transform odds with profit margins to probability values that sum up to 100% [21]. An equation was proposed to transform odds with margins to their original probabilities.

Bookmakers' published odds have multiple times been shown to be good forecasts for match outcomes [22, 15, 14] and have been called the golden odds for exactly that reason. Studies suggest that bookmakers acquire extra information that exceeds the historical match data available for the public, improving their published odds [16]. They have data engineers constantly focused on improving them. Just like everyone in this society, the primary goal of bookmakers is their own bank account. The focus of this research is hence to generate better odds than the bookmakers' ones and exploit their potential mistakes and differences in odds. As said by Dixon and Coles [6], "it requires a determination of probabilities that is sufficiently more accurate from those obtained by published odds" to generate profit.

Constantinou et al. published multiple papers [16, 3, 23, 8] in which better ways to forecast match results are researched. They showed that odds of a single bookmaker show deficiencies [23], implying that the gambling market is inefficient. This means that gambling strategies exist which can generate guaranteed profit. They demonstrated a high profitability on the basis of consistent odds biases and numerous arbitrage opportunities, plus the fact that the accuracy of bookmakers' odds had not been improved over the course of 6 seasons with 16 football leagues taken into consideration. One of the most remarkable results is that while betting on the favorites of matchups generates more returns in quantity, it is not profitable. On the contrary, betting on home playing underdogs generates higher returns and the bookmakers' bias towards favorites can be exploited into consistent profitability.

1.2.3 Money management

There are multiple ways one can go about betting and choosing the size of his stakes – so called betting strategies and money management systems –, e.g. fixed stake, fixed return, using the Kelly ratio [24] with a limited max stake, minimizing the difference between the expected profit and the variance of that profit [25], using the Markowitz portfolio management [26] etc. Some of these ideas were originally designed for portfolio management in stock investing, but can obviously also be used for sports betting. These systems have been compared a lot [27, 17, 18] and will be taken into consideration in the course of this thesis.

1.3 Research questions

At first, the focus lies on maximizing the accuracy of our predictor, but the actual goal of this research is to maximize profit. Aside from a different choice of sports, that is the main difference with Robin's research [1]. He only considered accuracy and used the published odds as features for his classifiers.

An betting strategy has to be found to filter out most of the bets the prediction model is not certain enough about. Football is less predictable than tennis and expectations, profit wise, are expected to be rather low to nonexistent if all matches were to be forecasted. Hence, instead of predicting every single match we can find, a recommendation tool – like pi-football [16] and FRES [12] – will be created as an assist for bettors. It should output matches for which the estimated odds are better than the published versions. Using the class distribution of our classifier results (i.e. the probability for each class) odds can be generated and used for recommendations. Potential significant differences can be found while comparing these chances to published odds and used for profit. This phenomenon is called 'playing the odds'.

This is best explained with an example. Imagine an intense football match between Manchester United and Chelsea, in which no team is the clear favorite (e.g. odds 3.29 - 3.50 - 2.29). Let's say that the prediction of our algorithm outputs the odds 1.25 for the home playing team. Using the difference in odds, United would be the recommendation to bet on. Assuming that our predictor does generate golden odds, Manchester United is almost certainly going to win. More concrete, Manchester United is going to win 4 in 5 games against Chelsea in the long run. If one were to bet on every match, this would result in a gross payout of 4 x 3.29 - 1 = 12.16 times the stakes – winning four bets with odds 3.29 and losing one of the five stakes. Net profit is thus guaranteed.

Then again, it is unreasonable to expect profits like this. The published odds are assumed to be the golden odds and are proven hard to improve. The goal is hence to find a profitable combination of features, prediction models and betting strategies to maximize profit. No research was found that combines these three factors to create a recommender for sports bets.

The details of this tool to help bettors are also considered to be research questions. Which features are the drivers of a match's outcome according to this research? Which model works best with those features? Where do we get enough (live) data to create the above mentioned tool? It would be great to start off with good data, quality and quantity wise, as most papers only consider a couple of seasons of a few leagues – if multiple leagues at all – to train their classifiers.

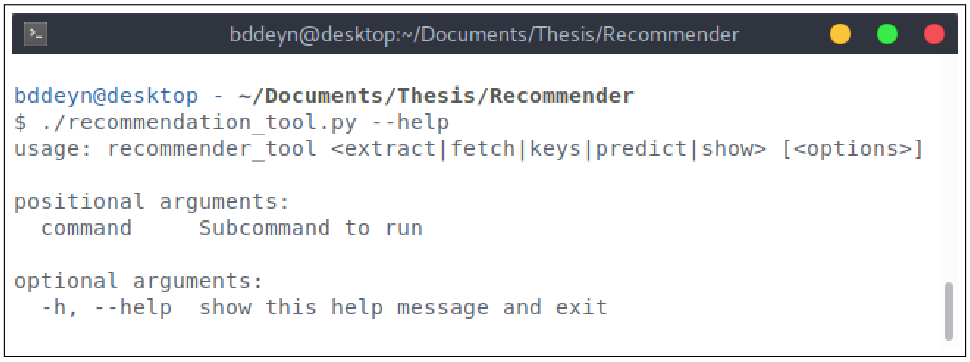
1.4 Next chapters

Next chapter is all about the collected data. Where did it come from and what does it consist of? This may be considered as the most important part of this dissertation, because a good machine learning research falls or stands by the quality of its data.

Afterwards, Chapter 3 is a summary of all the implemented features. The raw data from Chapter 2 is converted to usable features for the models in Chapter 4 and 5. What features can be extracted from the data and are thus implemented and tested features in our code? They are grouped into several categories and their importance is tested. In Chapter 4 the performance in accuracy of these features is tested and the best models for our research are found.

We take a step back in the last main chapter and use the metric profit to evaluate models in Chapter 5. Are there a lot of changes? What betting strategy is considered the best to use? Is there a clear winner whatsoever?

Each of these chapter also contains a section discussing its part of the recommendation tool's functionality that was engineered during this thesis research. The tool is written in Python3 and is accessible through the command line. The four subcommands `extract`, `fetch`, `predict` and `show` (see Figure 1.2)will be discussed below.



```
bddeyn@desktop - ~/Documents/Thesis/Recommender
$ ./recommendation_tool.py --help
usage: recommender_tool <extract|fetch|keys|predict|show> [<options>]

positional arguments:
  command      Subcommand to run

optional arguments:
  -h, --help   show this help message and exit
```

Figure 1.2: Help message showing the different subcommands of the recommendation tool.

Chapter 2

Data Exploration

Every data scientist knows that a good data set is key in a machine learning research. A big chunk of time was spent on obtaining the best possible data available. The used data was acquired from a sports data API instead of using a downloadable data set.

2.1 Sports data APIs

Using an API instead of a locally downloaded data set comes with some challenges. Requests will always be limited and thus duplicate data requests should be avoided by all means. Having a cache-like local database storing the raw responses for every request will directly lead to a bigger and better data set for the same amount of available requests.

Secondly, as the control of this third party data is out of our hands, the tool should support missing values and possible faulty data. Football matches can be canceled; less important matches could have less extensive statistics and player information could be missing. Probabilities are also not always recorded for some of the older matches. This is certainly something to keep in the back of your mind from now on.

Using an API also has some advantages. More data and live data can be dynamically fetched if needed. The data set – and eventually the end user of the tool – is thus not limited to the locally stored version. This is also advantageous for the recommendation tool, which can be used in real time on real world data, assuming access to the API is still present.

Eventually five national leagues and (about two seasons for) the two international ones (i.e. the Champions League and the UEFA League) were fetched all together. Every match since the 2011-2012 season was fetched for the English Premier League, the Spanish LaLiga, the Italian Serie A, the German Bundesliga and the Belgian Pro League. These are the biggest competitions of Europe, plus my own national league out of curiosity. The biggest competitions are the most popular, but are also assumed to be the most predictable. That is the biggest reason why they were considered for this research.

2.1.1 SportRadar

Many football data provider options are available, each with different sets of data aimed for several types of customers. The main focus while these options were compared, was to get as much information for every match as possible. The more extensive the information, the more features can be extracted and looked into. After contacting multiple sports data companies, SportRadar [28] seemed to be the best choice. Unlike most of the other options, they were willing to provide data to individuals and not only other companies. Added to that, they also have student programs in which they provide full data access for master and doctorate dissertations. The availability of trial accounts to kick start the thesis research was also a big plus. Apart from the obvious statistics – including scores, schedules and team information – SportRadar also provides match statistics and player information for most matches.

Due to the prolonged procedure to get a student license for full sports data access, several trial accounts – with a maximum limit of 1000 requests per month – were registered and used to insert data into the local database. This way, research progress was not hindered in any way.

2.1.2 Odds

Simple three way probabilities that sum up to a 100% are also supported through the SportRadar API, but unfortunately odds of major bookmakers are not supported. These probabilities can be converted to odds with a custom bookmaker's profit margin. Using these raw percentages instead of published odds actually comes with opportunities for more research. Different profit margins can be considered during model testing. This could be useful as a bookmaker's margin can easily be calculated from a couple of odds published on their website. These probabilities are also not subject to dynamic changes due to customer bets like the odds found on bookmakers' websites. These changes can hence be very easily observed by the user of our recommendation tool and used to their advantage.

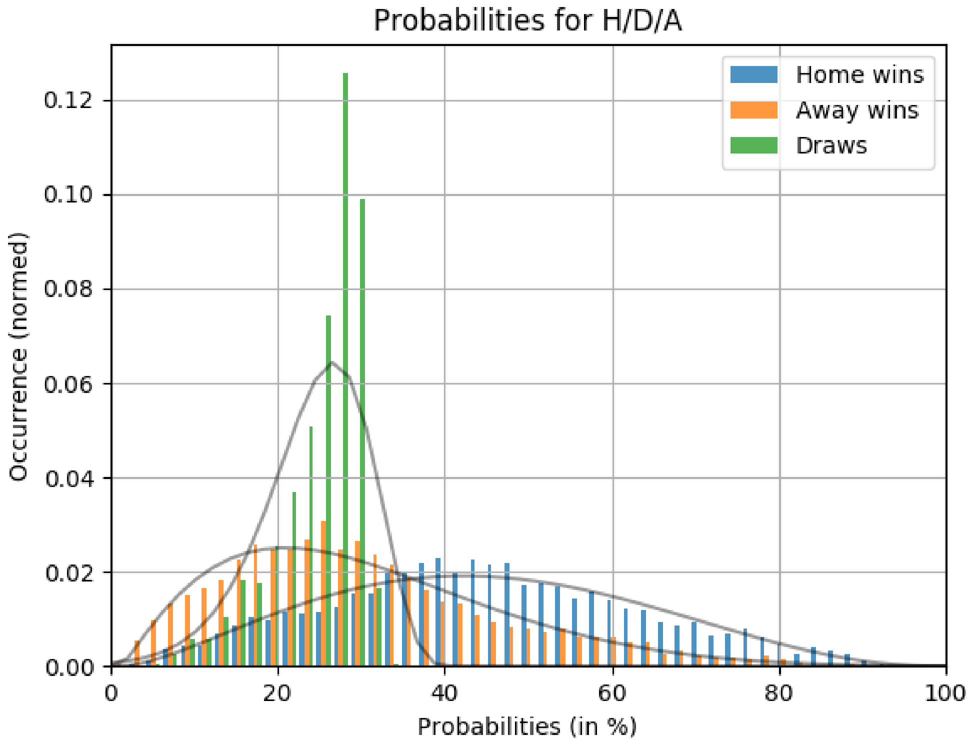


Figure 2.1: A plot with histograms depicting the occurrences of home wins, away wins and draws for several leagues. The beta distributed trend lines are added for clarity.

Looking at the histogram of the fetched probabilities (see Figure 2.1), some conclusions can be made. Home wins are again shown to be more probable than away wins and draws. Furthermore, there is a big spike around the 28% mark for draw outcomes and no matches have draw chances of more than 40%. This again shows that draws are hard to predict.

Although it is not stated how the probabilities were obtained in SportRadar's documentation – it could be their own predictions, as well as averages from published data – odds calculated from random probability samples seem to be really close to the published odds on respected bookmakers' websites. Other possible data sources were looked at for more accurate odds, but the time it would take to implement a second data source was not worth the insignificant improvement.

2.1.3 Dataflow

The dataflow of SportRadar does need some explanation. Every object – i.e. matches, teams, tournaments, seasons etc. – has a unique ID. As querying by name is not possible, these identifiers are needed to perform requests. Four steps thus are needed to get decent information for each match (see Figure 2.2).

1. To start off, get the tournament list consisting of basic tournament information and IDs. This also contains the season identifiers for each tournament. This takes one request per year, whenever the new seasons kick off. The new season IDs then have to be fetched to start feature extraction and result forecasting.
2. Afterwards, the match IDs and basic information of each desired season should be fetched. This takes only one request per season and returns the full season schedule.
3. Finally, now that the match identifiers are known, more information can be fetched for every match. Both match summaries and match probabilities have been fetched for this research, but more requests could be performed for further research.

The raw responses of all these requests are stored in a local MySQL database with tables for each of the above mentioned objects (see Figure 2.3). As mentioned already, the local database is used to prevent duplicate requests. If a match already has its summary fetched for example, the version found in the database can simply be reused, because its statistics and information will not change over time. The exact content of the request responses is documented in SportRadar's documentation [29] and each of their requests can be tested in their sandbox [30], assuming you applied for and have access to a trial API key.

2.2 Data fetching tool

User control of the local database is desired and can be done with the `show` subcommand (see Figure 2.4). The requests currently stored in the local database can be looked up easily. The main use of this is to quickly get the ID of a wanted object (e.g. tournament ID 17 for the Premier League). It is also possible to print out several match statistics per tournament or season, e.g. whether the match schedules have been fetched; how many seasons of each tournament are currently stored in the database; how many match summaries per season are available; how many matches have their probabilities pulled from the API etc.

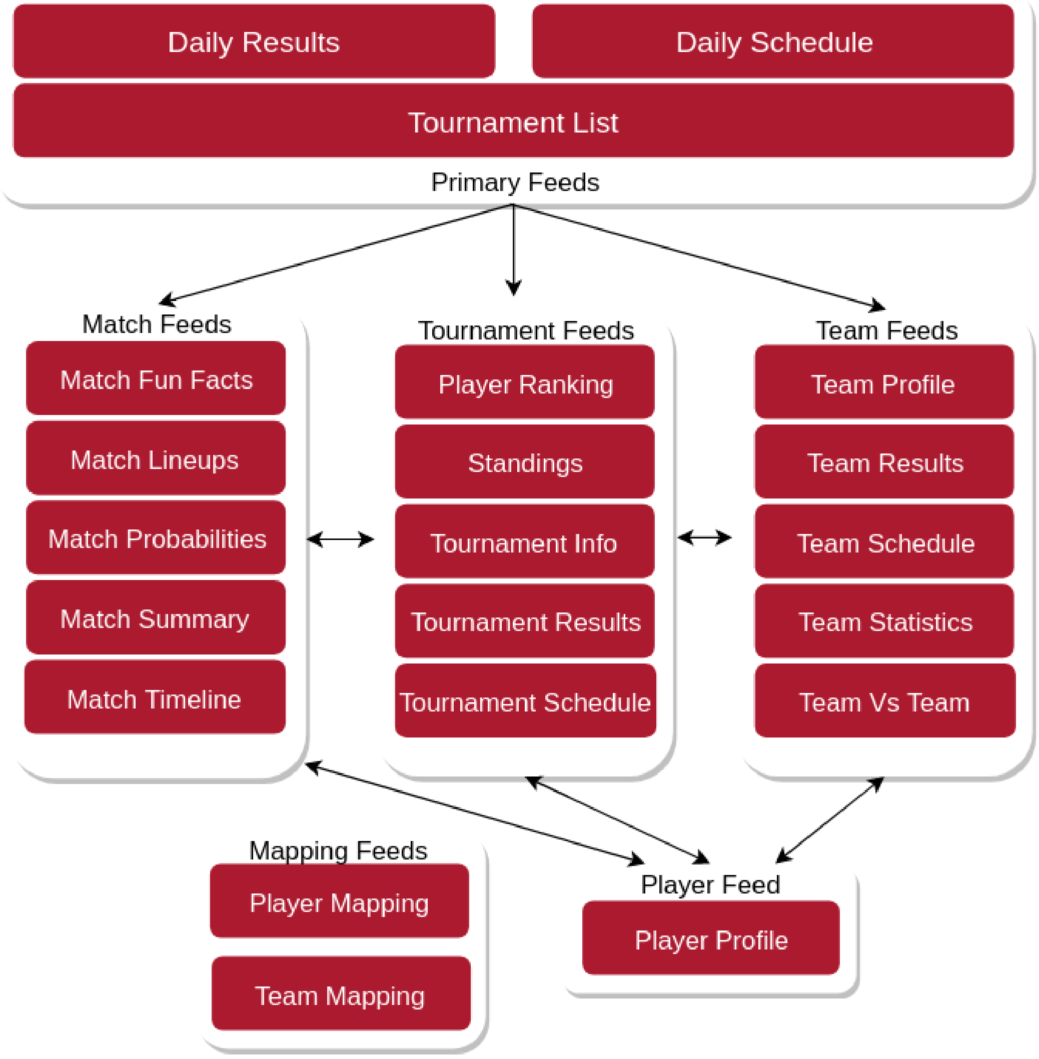


Figure 2.2: Dataflow for the SportRadar API [29].

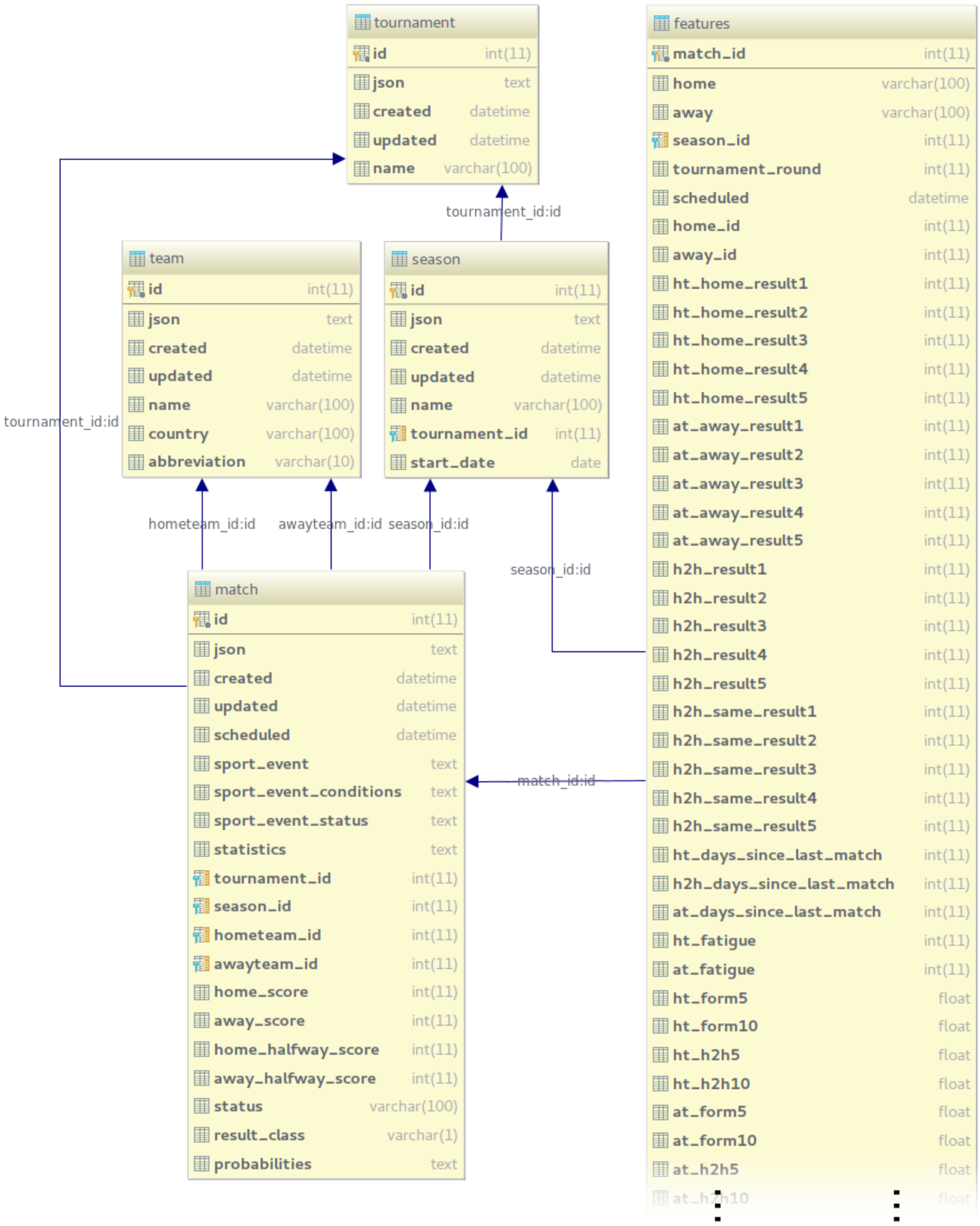
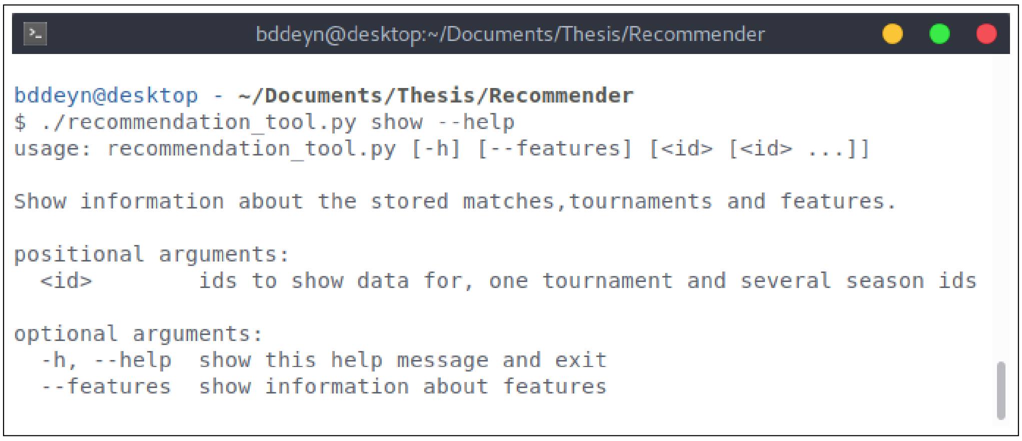


Figure 2.3: The schema of the local MySQL database. The 116 columns of the features table are not completely shown.

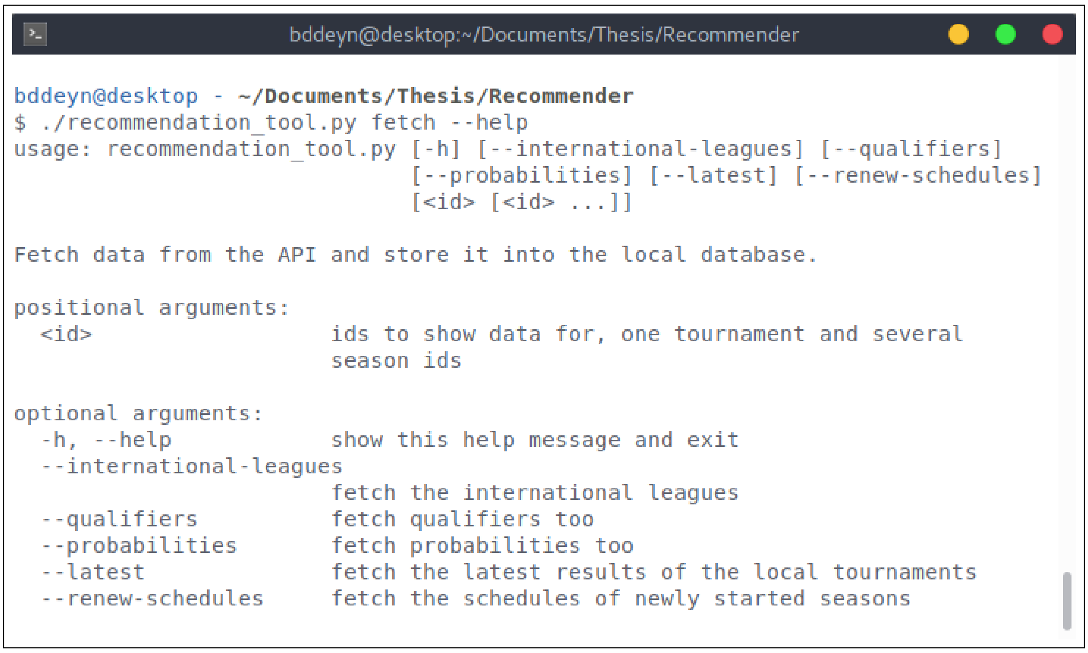
To make the functionality above show something useful, data has to be actually fetched and stored into the database first. This can be done with the `fetch` subcommand (see Figure 2.5). Matches of every tournament season can be fetched separately using its season ID, but there is also an option to add the summaries of the most recent completed matches. This is very convenient for frequent users of the recommendation tool who



```
bddeyn@desktop - ~/Documents/Thesis/Recommender
$ ./recommendation_tool.py show --help
usage: recommendation_tool.py [-h] [--features] [<id> [<id> ...]]

Show information about the stored matches,tournaments and features.

positional arguments:
  <id>        ids to show data for, one tournament and several season ids

optional arguments:
  -h, --help  show this help message and exit
  --features  show information about features
```

Figure 2.4: Help message for the show subcommand, with the tournament list output and the seasons of the Premier League afterwards.

can update their data set up until the current date with a single command. Furthermore, fetching matches of the international leagues is supported. These matches can be used to obtain more information about the performance and fatigue of a team in the national leagues. This could be key to predict an upset for example. Most teams competing in these leagues originate from major European leagues, which are the main focus of this research. The qualifiers of these leagues can also be manually fetched if wanted, but the teams playing in the qualifiers are mostly not of interest and will not add any information for the predictions in the major leagues. This is considered a poor use of limited requests and is hence disabled on default.



```
bddeyn@desktop - ~/Documents/Thesis/Recommender
$ ./recommendation_tool.py fetch --help
usage: recommendation_tool.py [-h] [--international-leagues] [--qualifiers]
                              [--probabilities] [--latest] [--renew-schedules]
                              [<id> [<id> ...]]

Fetch data from the API and store it into the local database.

positional arguments:
  <id>                  ids to show data for, one tournament and several
                        season ids

optional arguments:
  -h, --help            show this help message and exit
  --international-leagues
                        fetch the international leagues
  --qualifiers          fetch qualifiers too
  --probabilities       fetch probabilities too
  --latest              fetch the latest results of the local tournaments
  --renew-schedules     fetch the schedules of newly started seasons
```

Figure 2.5: Help message for the fetch subcommand.

Chapter 3

Feature Extraction

Now that a local data set is available, features can be extracted on which models can later be trained and evaluated. It is key to figure out what aspects of a match and the history of the two teams playing against each other is correlated with the eventual outcome of the match (= features or attributes). What features are used for the odds bookmakers calculate and publish?

3.1 Possible features

All of the features below are calculated for both the recent matches of the home playing team and the recent matches of the away playing team separately – referenced with prefixes `ht` and `at` respectively in the feature names. The two teams most probably also played some games against each other in the recent past, so separate features are also extracted on this sequence of recent (head to head) matches – referenced with the prefix `h2h`.

3.1.1 Recent matches

The simplest features are the plain results of the most recent matches, where each match is converted into one feature. The amount of matches considered is five, because the importance of individual results older than that is found to be close to irrelevant. At first the result class of the match (i.e. H, D or A) was considered as a nominal feature, but afterwards the difference in goals was used to create more informative versions. The bigger the difference in goals scored by the teams, the bigger the difference in performance between them. A strictly positive number means that the considered team won, while a strictly negative number indicates a loss. If that number is zero, the game tied and hence no team won.

3.1.2 Rankings

In national leagues, rankings are calculated by awarding points for each result. Wins award a team with three points, a draw is one point and a loss means the team will have to wait until the next game for a chance to increase their total ranking points. The more points a team has, the better it is thought to be performing. Averaging these obtained points over the number of played matches is a good indicator about the season wide performance of a team. Using these features that are derived from points instead of the actual spot in the ranking gives more information and will hopefully lead to better results.

A fun fact about the points system: wins originally only awarded a team with two points [31]. Over the years that got increased to three points to stimulate teams to go for a win instead of settling for a draw. This creates more exciting games, but also indirectly leads to more upsets and a less predictable competition.

3.1.3 Form

Features based on form are basically a contraction of the most recent match results mentioned in section 3.1.1 and the rankings mentioned in section 3.1.2. Instead of simply adding the points, they are also averaged to account for missing values. These features are hence positive real numbers between 0 and 3 with 3 having won all the considered recent matches. Separate features based on scoring abilities are also extracted. Both the amount of goals made and the differences in goals in the most recent matches is extracted. This will indicate whether the teams recently have been easily winning their matches or clutching them with only one goal to spare.

These form features are extracted for both the five most recent and the ten most recent matches of the considered match to extract features for. These numbers are also chosen on website like FlashScore [32] etc. to calculate the form of teams.

3.1.4 Fatigue

The more exhausted a team is, the higher the chance of a poor performance. Counting the number of matches in the last couple of weeks or months, together with the amount of days since the last match, might be good indicators for the fatigue of a team.

The amount of days until the next match was also considered, but due to missing data this was quickly disposed of to prevent training on faulty data. A lot of smaller competitions and cups are not being tracked by SportRadar and the next matches might not always be available online.

3.1.5 Historic match statistics

Loads of match statistics are available in the data set. The ones considered and thought to be useful are listed below. These statistics could be an indicator of a well playing team or on the contrary, a poorly performing one. All of them are averaged over the five most recent matches.

- Ball possession
- Free kicks
- Shots on target
- Shots off target
- Shots saved
- Offsides
- Yellow cards
- Yellow-red cards
- Red cards
- Corners
- Successful passes
- Successful crosses
- Successful duels
- Created chances

3.1.6 Team information

Some of the IDs of the above mentioned objects are very well possible features too. Every team has that lower ranked team against which they simply cannot win for a period in time. By adding the identifiers of the two playing teams to the features database, prediction models will hopefully be able to detect that. Venue IDs are also added as features. Simply using these might reflect the advantage some teams have in certain stadiums and tell the model when a team will not play in its own stadium. A derivative of this feature is the distance to travel for the away team. The reasoning behind this is that long travels might fatigue the away team, which enlarges the home advantages.

3.2 Attribute selection

Which of these features have the most impact? To find out, WEKA [33, 34] is used to execute several attribute selection algorithms on the features table in the local database.

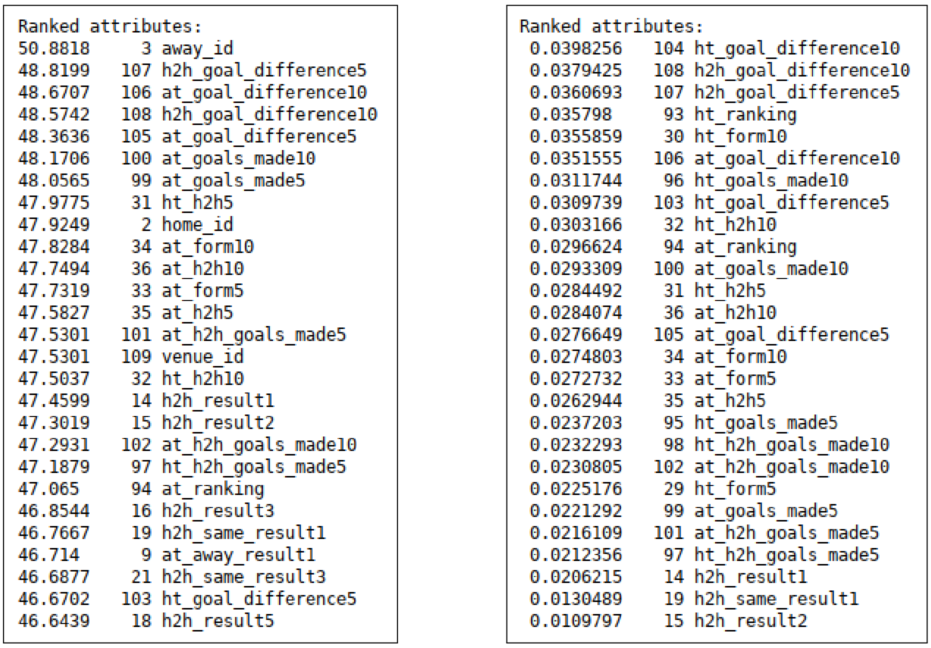
3.2.1 WEKA

WEKA is a collection of machine learning algorithms for data mining tasks, like stated on their website [33]. The algorithms, implemented in Java, can either be applied directly to a data set or called from Java code. The Explorer and WorkBench GUI they developed are easy and fast to use. It is particularly useful to explore a data set and quickly test some algorithms – e.g. attribute selection algorithms, clustering and classifiers. In the following chapters, a Python wrapper [35] for these Java classes is used to integrate WEKA into the recommendation tool. By using ready to use algorithms for our tool, valuable time is saved in compromise for some customizability.

3.2.2 Best features

Four algorithms are executed on the full data set over all competitions to determine how important each feature really is. First of all, the OneR algorithm (see Figure 3.1a) evaluates a feature's importance by evaluating it as a one feature classifier and ranks them by accuracy. Furthermore, InfoGain Ranker and GainRatio Ranker (see Figures 3.1b and 3.1c) both rank the features by the information they add to the model. They are thus evaluated by how much their contribution reduces the entropy of our problem. GainRatio Ranker is the more advanced form of the two as it normalizes the scores. Finally, the Correlation Ranker (see Figure 3.1d) algorithm's functionality is obvious, as its name pretty much gives it away. Features are ranked by its (Pearson's) correlation to the result class, i.e. how close they are to having a linear relationship to each other.

The conclusion of the results of these algorithms (see Figure 3.1) are pretty clear. Features derived from the goal differences in the recent past are the most important. The form features for the home team matches, the away team matches and the head to head matches are ranked on top. Added to that, IDs of the teams and the venue of the match seem to be good drivers for the result of the match, as well as the ranking features



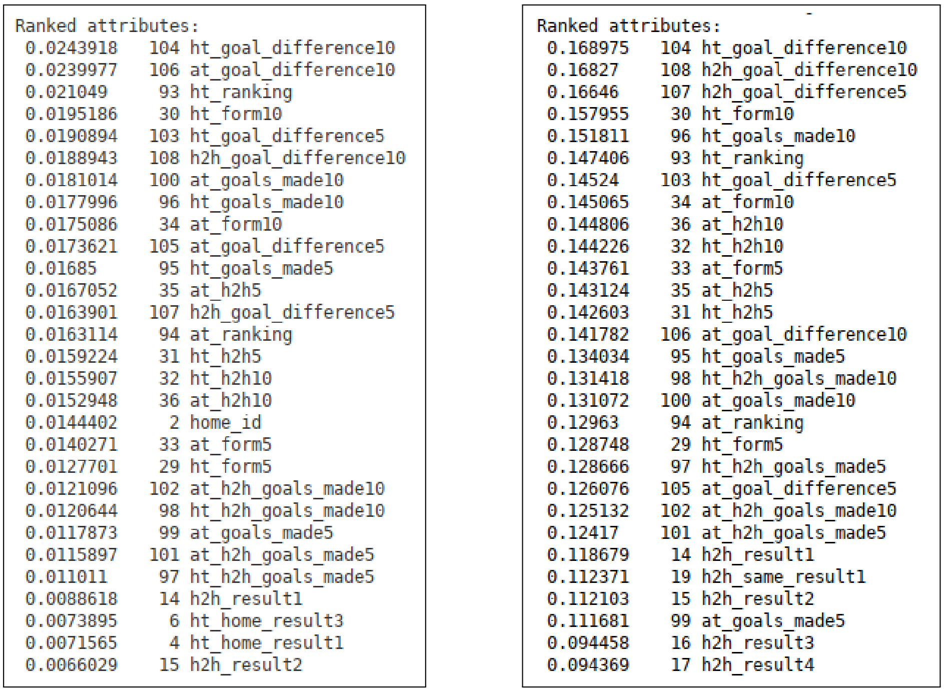
```
Ranked attributes:
50.8818      3 away_id
48.8199    107 h2h_goal_difference5
48.6707    106 at_goal_difference10
48.5742    108 h2h_goal_difference10
48.3636    105 at_goal_difference5
48.1706    100 at_goals_made10
48.0565     99 at_goals_made5
47.9775     31 ht_h2h5
47.9249      2 home_id
47.8284     34 at_form10
47.7494     36 at_h2h10
47.7319     33 at_form5
47.5827     35 at_h2h5
47.5301    101 at_h2h_goals_made5
47.5301    109 venue_id
47.5037     32 ht_h2h10
47.4599     14 h2h_result1
47.3019     15 h2h_result2
47.2931    102 at_h2h_goals_made10
47.1879     97 ht_h2h_goals_made5
47.065      94 at_ranking
46.8544     16 h2h_result3
46.7667     19 h2h_same_result1
46.714       9 at_away_result1
46.6877     21 h2h_same_result3
46.6702    103 ht_goal_difference5
46.6439     18 h2h_result5
```

(a) OneR

```
Ranked attributes:
 0.0398256   104 ht_goal_difference10
 0.0379425   108 h2h_goal_difference10
 0.0360693   107 h2h_goal_difference5
 0.035798     93 ht_ranking
 0.0355859    30 ht_form10
 0.0351555   106 at_goal_difference10
 0.0311744    96 ht_goals_made10
 0.0309739   103 ht_goal_difference5
 0.0303166    32 ht_h2h10
 0.0296624    94 at_ranking
 0.0293309   100 at_goals_made10
 0.0284492    31 ht_h2h5
 0.0284074    36 at_h2h10
 0.0276649   105 at_goal_difference5
 0.0274803    34 at_form10
 0.0272732    33 at_form5
 0.0262944    35 at_h2h5
 0.0237203    95 ht_goals_made5
 0.0232293    98 ht_h2h_goals_made10
 0.0230805   102 at_h2h_goals_made10
 0.0225176    29 ht_form5
 0.0221292    99 at_goals_made5
 0.0216109   101 at_h2h_goals_made5
 0.0212356    97 ht_h2h_goals_made5
 0.0206215    14 h2h_result1
 0.0130489    19 h2h_same_result1
 0.0109797    15 h2h_result2
```

(b) InfoGain

Figure 3.1: Results of the attribute selection algorithms.



```
Ranked attributes:
 0.0243918    104 ht_goal_difference10
 0.0239977    106 at_goal_difference10
 0.021049      93 ht_ranking
 0.0195186     30 ht_form10
 0.0190894    103 ht_goal_difference5
 0.0188943    108 h2h_goal_difference10
 0.0181014    100 at_goals_made10
 0.0177996     96 ht_goals_made10
 0.0175086     34 at_form10
 0.0173621    105 at_goal_difference5
 0.01685       95 ht_goals_made5
 0.0167052     35 at_h2h5
 0.0163901    107 h2h_goal_difference5
 0.0163114     94 at_ranking
 0.0159224     31 ht_h2h5
 0.0155907     32 ht_h2h10
 0.0152948     36 at_h2h10
 0.0144402      2 home_id
 0.0140271     33 at_form5
 0.0127701     29 ht_form5
 0.0121096    102 at_h2h_goals_made10
 0.0120644     98 ht_h2h_goals_made10
 0.0117873     99 at_goals_made5
 0.0115897    101 at_h2h_goals_made5
 0.011011      97 ht_h2h_goals_made5
 0.0088618     14 h2h_result1
 0.0073895      6 ht_home_result3
 0.0071565      4 ht_home_result1
 0.0066029     15 h2h_result2
```

(c) GainRatio

```
Ranked attributes:
 0.168975    104 ht_goal_difference10
 0.16827     108 h2h_goal_difference10
 0.16646     107 h2h_goal_difference5
 0.157955     30 ht_form10
 0.151811     96 ht_goals_made10
 0.147406     93 ht_ranking
 0.14524     103 ht_goal_difference5
 0.145065     34 at_form10
 0.144806     36 at_h2h10
 0.144226     32 ht_h2h10
 0.143761     33 at_form5
 0.143124     35 at_h2h5
 0.142603     31 ht_h2h5
 0.141782    106 at_goal_difference10
 0.134034     95 ht_goals_made5
 0.131418     98 ht_h2h_goals_made10
 0.131072    100 at_goals_made10
 0.12963      94 at_ranking
 0.128748     29 ht_form5
 0.128666     97 ht_h2h_goals_made5
 0.126076    105 at_goal_difference5
 0.125132    102 at_h2h_goals_made10
 0.12417     101 at_h2h_goals_made5
 0.118679     14 h2h_result1
 0.112371     19 h2h_same_result1
 0.112103     15 h2h_result2
 0.111681     99 at_goals_made5
 0.094458     16 h2h_result3
 0.094369     17 h2h_result4
```

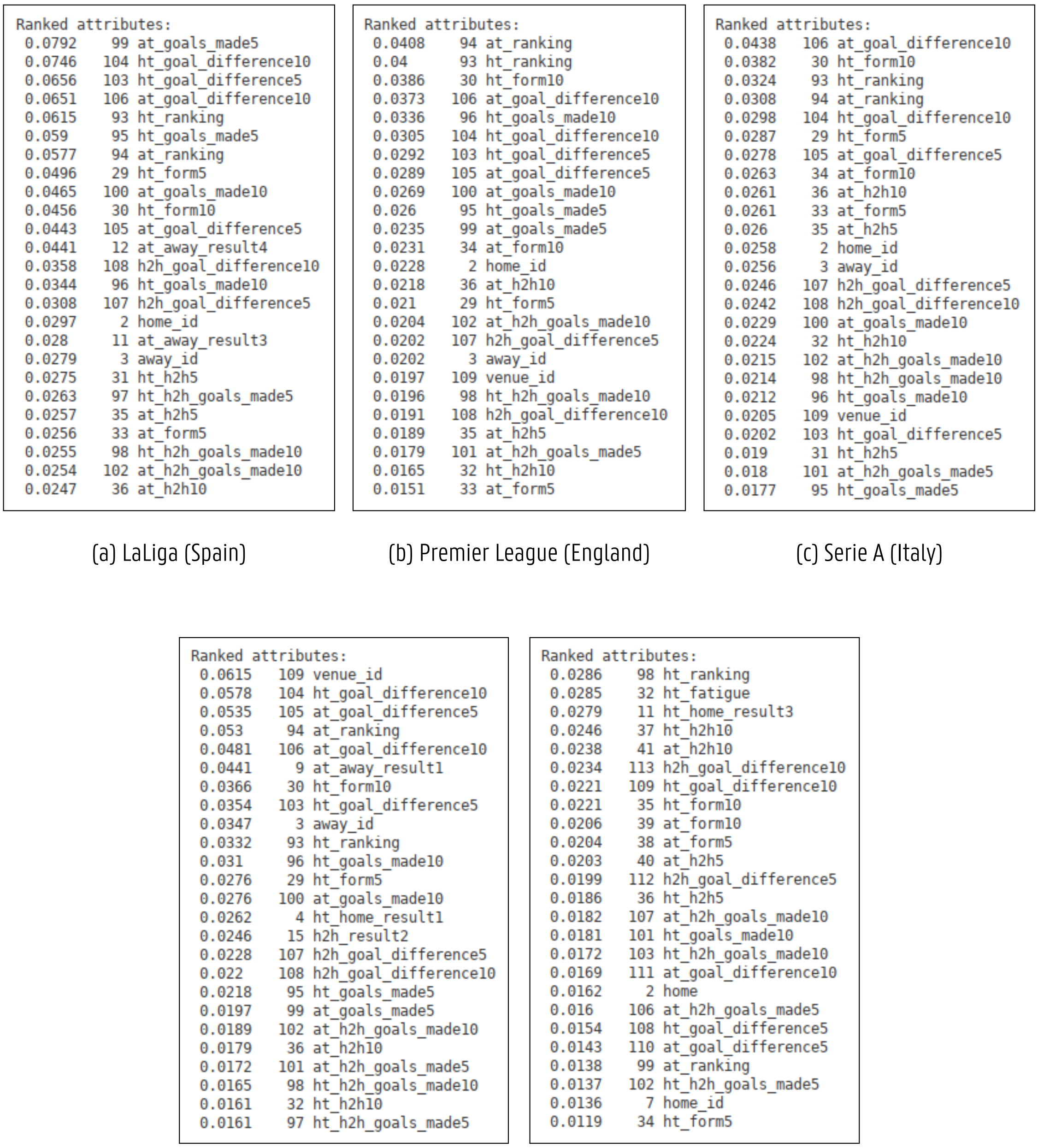
(d) Correlation

Figure 3.1: Results of the attribute selection algorithms.

for the two teams. The result features for the most recent head to head matches have a less important information addition than the derivative features, but seem to have a better correlation with the result class than any of the features derived from historical statistics. It is particularly surprising that none of the later features showed up in these rankings.

3.2.3 Results per league

What if an attribute selection research is performed per league? Can any big differences between the competitions be expected? No surprises are found in the results (see Figure 3.2 for the output of the GainRatio Ranker). Although there is a big difference in values and the rankings are a bit scrambled, the types of features in the top 25 are more or less the same. Notice the two ranking features on top for the Premier League. This suggests that it is the league with the least surprises and thus the most predictable league of the ones considered.



```
Ranked attributes:
 0.0792    99 at_goals_made5
 0.0746   104 ht_goal_difference10
 0.0656   103 ht_goal_difference5
 0.0651   106 at_goal_difference10
 0.0615    93 ht_ranking
 0.059     95 ht_goals_made5
 0.0577    94 at_ranking
 0.0496    29 ht_form5
 0.0465   100 at_goals_made10
 0.0456    30 ht_form10
 0.0443   105 at_goal_difference5
 0.0441    12 at_away_result4
 0.0358   108 h2h_goal_difference10
 0.0344    96 ht_goals_made10
 0.0308   107 h2h_goal_difference5
 0.0297     2 home_id
 0.028     11 at_away_result3
 0.0279     3 away_id
 0.0275    31 ht_h2h5
 0.0263    97 ht_h2h_goals_made5
 0.0257    35 at_h2h5
 0.0256    33 at_form5
 0.0255    98 ht_h2h_goals_made10
 0.0254   102 at_h2h_goals_made10
 0.0247    36 at_h2h10
```

(a) LaLiga (Spain)

```
Ranked attributes:
 0.0408    94 at_ranking
 0.04      93 ht_ranking
 0.0386    30 ht_form10
 0.0373   106 at_goal_difference10
 0.0336    96 ht_goals_made10
 0.0305   104 ht_goal_difference10
 0.0292   103 ht_goal_difference5
 0.0289   105 at_goal_difference5
 0.0269   100 at_goals_made10
 0.026     95 ht_goals_made5
 0.0235    99 at_goals_made5
 0.0231    34 at_form10
 0.0228     2 home_id
 0.0218    36 at_h2h10
 0.021     29 ht_form5
 0.0204   102 at_h2h_goals_made10
 0.0202   107 h2h_goal_difference5
 0.0202     3 away_id
 0.0197   109 venue_id
 0.0196    98 ht_h2h_goals_made10
 0.0191   108 h2h_goal_difference10
 0.0189    35 at_h2h5
 0.0179   101 at_h2h_goals_made5
 0.0165    32 ht_h2h10
 0.0151    33 at_form5
```

(b) Premier League (England)

```
Ranked attributes:
 0.0438   106 at_goal_difference10
 0.0382    30 ht_form10
 0.0324    93 ht_ranking
 0.0308    94 at_ranking
 0.0298   104 ht_goal_difference10
 0.0287    29 ht_form5
 0.0278   105 at_goal_difference5
 0.0263    34 at_form10
 0.0261    36 at_h2h10
 0.0261    33 at_form5
 0.026     35 at_h2h5
 0.0258     2 home_id
 0.0256     3 away_id
 0.0246   107 h2h_goal_difference5
 0.0242   108 h2h_goal_difference10
 0.0229   100 at_goals_made10
 0.0224    32 ht_h2h10
 0.0215   102 at_h2h_goals_made10
 0.0214    98 ht_h2h_goals_made10
 0.0212    96 ht_goals_made10
 0.0205   109 venue_id
 0.0202   103 ht_goal_difference5
 0.019     31 ht_h2h5
 0.018    101 at_h2h_goals_made5
 0.0177    95 ht_goals_made5
```

(c) Serie A (Italy)

```
Ranked attributes:
 0.0615   109 venue_id
 0.0578   104 ht_goal_difference10
 0.0535   105 at_goal_difference5
 0.053     94 at_ranking
 0.0481   106 at_goal_difference10
 0.0441     9 at_away_result1
 0.0366    30 ht_form10
 0.0354   103 ht_goal_difference5
 0.0347     3 away_id
 0.0332    93 ht_ranking
 0.031     96 ht_goals_made10
 0.0276    29 ht_form5
 0.0276   100 at_goals_made10
 0.0262     4 ht_home_result1
 0.0246    15 h2h_result2
 0.0228   107 h2h_goal_difference5
 0.022    108 h2h_goal_difference10
 0.0218    95 ht_goals_made5
 0.0197    99 at_goals_made5
 0.0189   102 at_h2h_goals_made10
 0.0179    36 at_h2h10
 0.0172   101 at_h2h_goals_made5
 0.0165    98 ht_h2h_goals_made10
 0.0161    32 ht_h2h10
 0.0161    97 ht_h2h_goals_made5
```

(d) Bundesliga (Germany)

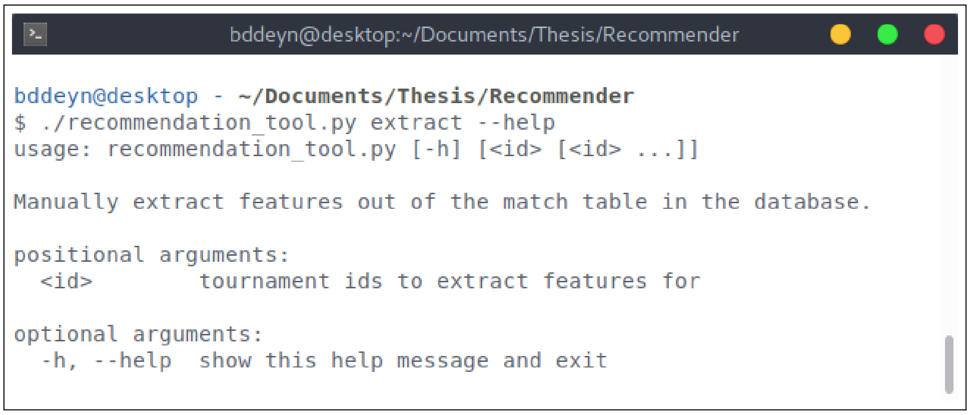
```
Ranked attributes:
 0.0286    98 ht_ranking
 0.0285    32 ht_fatigue
 0.0279    11 ht_home_result3
 0.0246    37 ht_h2h10
 0.0238    41 at_h2h10
 0.0234   113 h2h_goal_difference10
 0.0221   109 ht_goal_difference10
 0.0221    35 ht_form10
 0.0206    39 at_form10
 0.0204    38 at_form5
 0.0203    40 at_h2h5
 0.0199   112 h2h_goal_difference5
 0.0186    36 ht_h2h5
 0.0182   107 at_h2h_goals_made10
 0.0181   101 ht_goals_made10
 0.0172   103 ht_h2h_goals_made10
 0.0169   111 at_goal_difference10
 0.0162     2 home
 0.016    106 at_h2h_goals_made5
 0.0154   108 ht_goal_difference5
 0.0143   110 at_goal_difference5
 0.0138    99 at_ranking
 0.0137   102 ht_h2h_goals_made5
 0.0136     7 home_id
 0.0119    34 ht_form5
```

(e) First Division A (Belgium)

Figure 3.2: Results of the GainRatio attribute selection algorithm for competitions separately.

3.3 Feature extraction tool

While an end user ideally never has to extract the features for the locally stored matches manually, the option is still available in the tool. Fetching the latest results implies that only the features for those newly added matches have to be extracted. In contrary, fetching historic data means most of the features for matches of that tournament are invalidated and have to be extracted all over again. To give the end user full control over the data and features, the `extract` subcommand is also present in the recommendation tool (see Figure 3.3).



```
bddeyn@desktop - ~/Documents/Thesis/Recommender
$ ./recommendation_tool.py extract --help
usage: recommendation_tool.py [-h] [<id> [<id> ...]]

Manually extract features out of the match table in the database.

positional arguments:
  <id>        tournament ids to extract features for

optional arguments:
  -h, --help  show this help message and exit
```

Figure 3.3: Help message for the extract subcommand.

3.4 Further research

A lot more possible features can be thought of. Weather information could be useful. It is for example possible that some teams underperform in bad weather or actually prefer the rain or cold. More team information might improve predictions too, e.g. club budget, the total salary of the team, injuries of key players, performance of individual players, coaches and managers, whether some players recently played a national match or not, etc.

The question though remains: will there be features that contribute more information than the most obvious features considered above? Is it worth the extra effort to implement? Is it worth the extra execution time to extract features and train classifiers for? Only further research can determine that.

Chapter 4

Prediction models

Now that features are extracted from the match table of the local database, models can be assembled and the actual forecasting can begin. Just like in the previous chapter, WEKA [33] is used through a Python wrapper [35] for model training and evaluation. This chapter is focused on correctly predicting as much matches as possible, i.e. optimizing the model accuracy. The goal to achieve actual guaranteed profit is discussed in the following chapter. Published odds are then included into the prediction process.

After loading the features table of the local database into the WEKA Workbench GUI, the available classifier models can be explored. It instantly shows which algorithms support the imported data and allows full parameter customization before executing and evaluating a model. This way an idea can be formed about which model types fit the sports data best for our match outcome forecasting problem.

4.1 Model evaluation

While cross validation is considered to be the standard in model evaluation, it is not applicable to evaluate sports prediction models. Even though instance features are only extracted from matches that were scheduled prior to that instance, a classifier simply cannot use them as actual predictions will not have that information. Cross validation would allow the classifiers to find dependencies that cannot be replicated outside of the training and validation phase. A better way to evaluate the sports prediction models is to split the labeled data set while preserving the order the matches were played in. An 80% split is used in this research, which means the last 20% of the matches matches is predicted with models that were trained on the first 80%.

4.2 Baseline predictors

Let's start with baseline predictors. A baseline predictor basically is a simple classifier with which other more complex ones can be compared. The goal is to obtain more information about the problem itself and to be able to measure the improvement of the more complex versions afterwards.

What are the options in this case? Both ZeroR and OneR are provided by WEKA. Their names are actually an indication of how they function. ZeroR uses none of the features and predicts the majority result class for every instance. This boils down to predicting the home team to be the winner for every match. On the other hand, OneR chooses the attribute that produces the smallest error for the data set. It is actually proven that very simple classification rules perform well on most commonly used data sets [36]. It can hence be assumed that these baselines are good starting points.

4.3 Models

Now that two baseline predictors are available, more complex models can be researched. Their documentation pages [37] should be consulted for more information about the classifiers and how to use them. An experiment is set up with all the models that are thought to be well performing for the sports features. Default values are used for the parameters, which can later be optimized in a following section if needed. The models proposed below are executed once without dimensionality reduction and three more times to test the effect of different reduction techniques. GainRatio (GR), Correlation and Principal Components Analysis (PCA) [38] are all three used to reduce to 25 attributes per instance. While PCA transforms the attributes into uncorrelated features, both GainRatio and Correlation rank the attributes and take the best ones according to their specific metric. The accuracy results are shown in Table 4.1.

Reducing the dimensionality of the problem not only reduces the execution times for the classifiers, it also improves the performance of certain models which would otherwise be overfitting drastically (see Table 4.1). Another measure taken to prevent overfitting is dividing the features table into separate leagues. Instead of trying to predict all matches in one take, better accuracy is achieved by training models for each league separately. Most of the teams considered will almost never face a team of a different country, so no dependencies should be tried to trained. Plus, the models are going to be more simple and training time is significantly shorter.

The matches of the first five rounds are also not considered for features, although they are indirectly included into the features of the next matches. The start of the season is always quite unpredictable and a lot of upsets happen in that stage of the season. Using these matches to train on would potentially cause the classifier to overfit. That is also why classifiers should not train on matches that are too old. Many teams have a significantly big performance change over the years, for better or worse. Matches more than five years before the matches to predict should not be included in the training data set.

4.3.1 Support Vector Machines

Support Vector Machines or SVMs are debatably the most popular supervised learning models. An SVM is essentially a non-probabilistic binary linear classifier. Vanilla WEKA supports SVMs through the SMO classifier, called like algorithm used to solve quadratic problems in SVMs. Normalization of the attributes is enabled by default and multi-class problems are solved using pairwise classification, so three classifiers are trained in this case (H vs A, H vs D and D vs A). Non-linear learning can be achieved with different kernels. SMO supports polynomial kernels, both the standard and normalized versions, an RBF (radial basis function) kernel and the PUK (Pearsen function-based universal) kernel.

LibSVM on the other hand is a more popular Python library to solve the SVM problem. An extension wrapping this library is available for WEKA and its documentation page describes the classifier as being faster than SMO. It could thus not be left behind in this research. The biggest difference between the two implementations of support vectors is that LibSVM can solve multi-class problems with a single classifier. Two types are supported for our data set: C-SVC and nu-SVC. Nevertheless only C-SVC is tested, because both are proven to be mathematically the same (according to the LibSVM FAQ page [39]) and only use a different parameter range (for C and ν respectively). LibSVM supports four kernels: linear, sigmoid, polynomial and radial.

More information about the kernels – of both SMO and LibSVM – and how to use them can be found on the online WEKA documentation [40, 37].

4.3.2 Other models

MultiLayerPerceptrons or MLPs are feedforward neural networks utilizing a supervised learning technique called backpropagation. Without any dimensionality reduction, they are practically unfeasible to execute and even with reduction, they run significantly longer than the other classifiers. Although reduction techniques seem to limit the performance, neural networks are tested anyway, because they have been frequently used to obtain good results.

Naive Bayes classifiers are simple probabilistic classifiers with the assumption that features are independent. They apply the Bayes' theorem to find relations between features and the result class and have very fast execution times. It is considered to be a good competitor for support vectors in various domains and is hence also evaluated on this data set. It does not require any parameter configuration in WEKA which makes it really easy to implement.

RandomForests are actually an ensemble training technique. The classifier is used to limit the habit of its child decision trees to overfit to their training set. These performed really well on a small data set, but the execution time increased a lot once more seasons were fetched and the training set got bigger. These random forests seem to not scale very well on the football data. Training on features of multiple leagues together resulted in longer training times than the support vector classifiers. A RandomForests classifier is usually implemented with bagging as a bootstrap ensemble method. With the Bagging ensemble being supported on WEKA too, a default Bagging classifier is also set up and tested.

SimpleLogistic is the last classifier tested. It performs LogitBoost iterations to build linear logistic regression models which leads to automatic attribute selection. LogitBoost is an ensemble meta-algorithm that strengthens the logistic regression model and reduces bias and variance. It looked promising and is thus included in the experiment.

More models and ensembles are supported in WEKA, like Voting, MultiScheme and Bayesian Network classifiers. These only lead to higher execution times or simply did not improve any of the simpler models while testing them in the Explorer GUI. Concluding, there is no point in adding them to the experiment – which already took tens of minutes to finish executing.

	All leagues				**Spanish LaLiga**				**English Premier League**			
Model	**Full**	**GR**	**Corr.**	**PCA**	**Full**	**GR**	**Corr.**	**PCA**	**Full**	**GR**	**Corr.**	**PCA**
ZeroR	47.46%	-	-	-	48.16%	-	-	-	49.77%	-	-	-
OneR	52.29%	(`at_goal_diff10`)			49.77%	(`at_ranking`)			**54.36%**	(`at_ranking`)		
SMO (PolyKernel)	52.54%	52.95%	53.001%	53.81%	49.53%	52.53%	51.61%	**54.83%**	49.77%	**58.71%**	**58.02%**	**60.32%**
SMO (Norm.PolyKernel)	48.62%	53.15%	53.10%	52.84%	52.53%	52.30%	51.84%	51.15%	**59.86%**	**57.56%**	**57.79%**	**58.25%**
SMO (RBFKernel)	52.14%	52.54%	52.54%	47.46%	52.30%	51.38%	51.84%	48.15%	**56.42%**	52.98%	53.44%	49.77%
SMO (Puk)	48.16%	53.51%	**53.81%**	49.44%	50.9%	51.8%	50.9%	49.3%	52.0%	**58.2%**	**57.8%**	50.9%
C-SVC (sigmoid)	**53.71%**	53.20%	53.45%	**53.71%**	52.99%	52.76%	53.45%	52.76%	45.18%	**59.86%**	**59.40%**	**54.35%**
C-SVC (polynomial)	53.76%	52.54%	52.44%	46.38%	52.30%	53.45%	53.68%	**53.91%**	48.16%	**56.88%**	**57.11%**	50.45%
C-SVC (radial)	52.95%	**53.81%**	53.71%	**53.96%**	53.22%	52.30%	52.99%	49.30%	**57.79%**	**59.63%**	**60.09%**	44.49%
C-SVC (linear)	53.15%	52.79%	52.84%	**53.76%**	50.46%	51.38%	52.30%	**54.37%**	**55.50%**	**57.11%**	**57.79%**	**59.17%**
NaiveBayes	27.00%	49.79%	50.40%	26.75%	50.70%	**54.15%**	50.92%	31.10%	50.92%	53.67%	52.06%	41.51%
MLP	51.93%	53.30%	53.20%	52.89%	50.23%	51.61%	51.38%	50.46%	50.68%	**54.35%**	**55.04%**	51.37%
RandomForest	**53.96%**	**53.71%**	**54.37%**	52.54%	52.30%	51.38%	51.84%	52.99%	**58.71%**	**58.48%**	**58.02%**	**58.25%**
Bagging	47.45%	47.45%	47.45%	47.45%	50.69%	50.69%	50.69%	48.15%	**57.33%**	**57.33%**	**57.33%**	**57.11%**
SimpleLogistic	52.89%	52.89%	52.89%	52.84%	44.93%	50.46%	50.23%	49.07%	42.20%	53.44%	**54.35%**	**59.86%**

	Italian Serie A				**German Bundesliga**				**Belgian First Division A**			
Model	**Full**	**GR**	**Corr.**	**PCA**	**Full**	**GR**	**Corr.**	**PCA**	**Full**	**GR**	**Corr.**	**PCA**
ZeroR	45.39%	-	-	-	46.11%	-	-	-	46.03%	-	-	-
OneR	51.38%	(`at_goal_diff10`)			47.26%	(`at_goal_diff10`)			44.76%	(`ht_ranking`)		
SMO (PolyKernel)	**56.68%**	**55.06%**	**55.76%**	**54.37%**	47.26%	49.85%	49.28%	40.34%	44.76%	48.25%	48.25%	48.25%
SMO (Norm.PolyKernel)	**55.30%**	**55.30%**	**55.53%**	54.37%	48.99%	49.85%	49.56%	42.93%	48.88%	47.61%	48.57%	46.03%
SMO (RBFKernel)	45.62%	45.39%	45.39%	45.39%	47.26%	46.97%	46.68%	46.11%	46.03%	46.03%	46.03%	46.03%
SMO (Puk)	47.0%	53.0%	53.0%	47.9%	49.3%	49.0%	48.1%	41.8%	47.0%	50.8%	49.8%	47.3%
C-SVC (sigmoid)	**56.45%**	**53.91%**	**55.30%**	**54.14%**	51.58%	50.14%	48.99%	45.53%	47.30%	46.98%	46.98%	48.57%
C-SVC (polynomial)	46.31%	51.84%	51.84%	46.54%	50.72%	50.14%	49.85%	44.38%	48.88%	47.93%	47.61%	46.98%
C-SVC (radial)	**56.68%**	54.60%	54.37%	54.37%	51.87%	49.56%	50.43%	44.09%	47.30%	49.52%	48.88%	48.57%
C-SVC (linear)	**55.06%**	**55.06%**	**55.53%**	**54.60%**	47.55%	49.56%	49.28%	39.76%	46.03%	48.57%	48.25%	47.93%
NaiveBayes	26.27%	**55.53%**	**55.30%**	30.88%	43.51%	46.69%	44.38%	34.58%	44.76%	46.35%	45.08%	43.81%
MLP	**56.22%**	**56.22%**	**55.53%**	**54.60%**	48.70%	51.00%	51.29%	44.09%	46.03%	49.52%	49.52%	47.30%
RandomForest	**55.30%**	**55.30%**	**54.60%**	50.23%	50.72%	48.12%	47.55%	46.39%	49.52%	49.20%	49.52%	46.66%
Bagging	45.39%	45.39%	45.39%	45.39%	48.70%	48.70%	48.70%	50.14%	47.30%	46.66%	46.66%	46.66%
SimpleLogistic	52.99%	**55.30%**	**54.83%**	52.07%	50.14%	49.85%	51.29%	48.99%	46.66%	48.88%	48.57%	47.61%

Table 4.1: Accuracy results of the predictors for each competition separately.

4.4 Results discussion

The results (see Table 4.1) show that there are big differences between the leagues. All accuracy levels nearing the 54% mark are highlighted in bold, because it's proven that those classifiers can be used for guaranteed profit [15].

The synergy of the reduction algorithms with the models seems completely random. PCA works really well with the PolyKernel version of SMO, but performs awful with a NaiveBayes or RandomForest classifier. No big difference can be found between the GainRatio and Correlation algorithms, although it seems like a trial and error problem. The performance of these techniques also differ significantly per league.

The highest accuracy levels are achieved for the Premier League. Like suspected above in Chapter 3, it is found to be the most predictable. What's really remarkable is that the OneR baseline predictor also crosses the 54% accuracy level which indicates that this simple classifier can already be used for guaranteed profit. Just like the Premier League, the Spanish and Italian league should be profitable to bet on according to the experiment results.

Like personally expected and experienced while betting before starting this thesis research, the Belgian First Division seems to be unpredictable and possibly unprofitable to bet on. The Bundesliga has slightly better results but no models with accuracies over 54% have been found. Even the baseline predictor values for both leagues are lower than the other competitions. This indicates that many upsets or draws were seen during the course of the last couple of seasons. Upsets can be tracked using a certainty score based on the published odds, defined as the biggest difference in probabilities. The amount of underdog wins with a certainty score of over 0.10 (see Table 4.2) indicate the difference in predictability of the leagues. A difference of about 4 to 6% between the three more predictable and two (debatably) unpredictable leagues can be found.

No single model seems to stand out above the others for every league. Sure, the SVMs seem to be consistently good across the competitions, but the best kernels for each league differ. The same is true for the other classifiers. The neural network, for example, performs well for the Italian league, but Random Forests seem to result in better accuracy for the Premier League, while a simple NaiveBayes classifier performs quasi optimal for the Spanish league. The conclusion can be made that each league needs its own classifier for the next chapter.

League	Percentage underdog wins
Bundesliga	41.1%
Belgian First Division	39.6%
LaLiga	34.5%
Premier League	35.9%
Serie A	34.3%

Table 4.2: Underdog wins with a certainty score of over 10% on the published odds.

League	Kernel	C	Full	GR	Corr.	PCA
Spanish LaLiga	linear	0.5	49.07%	51.61%	51.84%	**55.30%**
English Premier League	radial	0.0625 / 0.125 / 1	56.65%	60.09%	**60.09%**	42.20%
Italian Serie A	linear	0.125	**57.83%**	54.83%	54.60%	53.68%
German Bundesliga	sigmoid	4.0	**51.87%**	50.14%	50.14%	44.09%
Belgian Pro League	radial	2.0	47.61%	**49.52%**	49.20%	47.61%

Table 4.3: The results for parameter optimization analysis of LibSVM, type C-SVC. The best kernel and C values are shown together with their accuracy values.

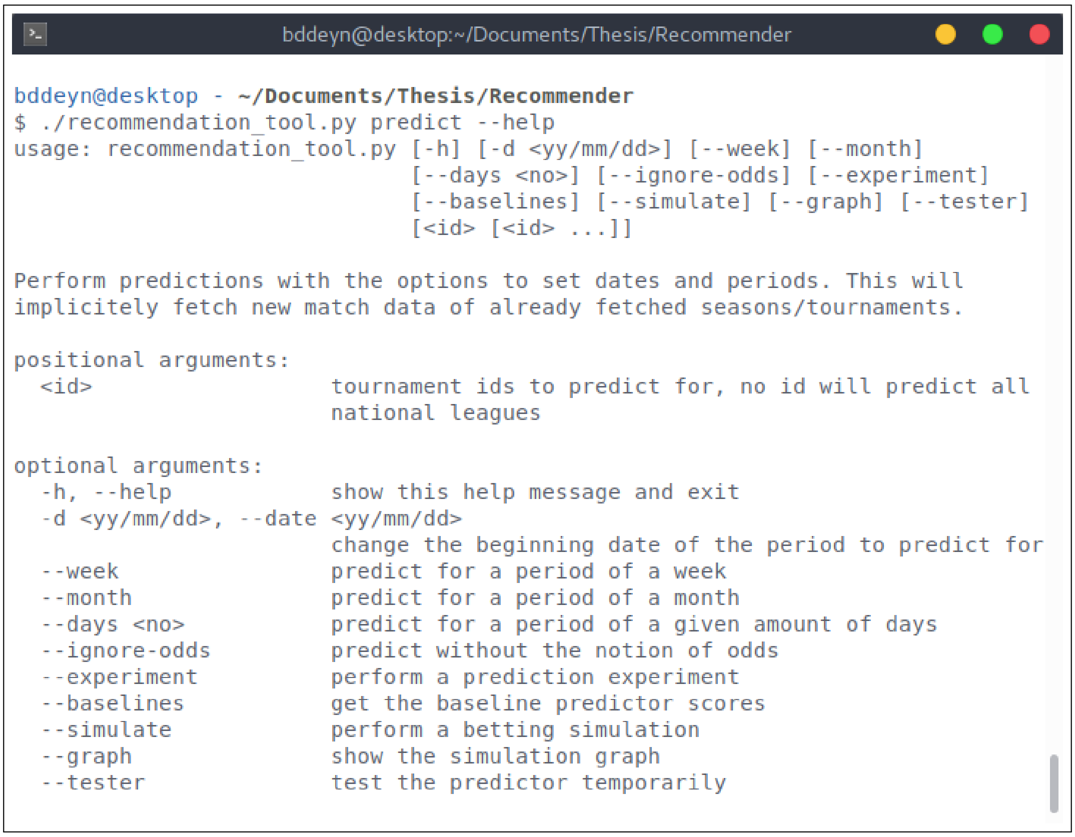
4.5 Parameter optimization

As the support vector classifiers are the most consistent models viewed over the leagues and usually take a couple of seconds to execute, a parameter optimization analysis is performed to possibly improve the accuracy. LibSVM is more popular and said to be faster, so this library is the obvious choice. Both the kernel and the complexity parameter are tried to be optimized. All four kernels are tested with a range of $2^{-4}, 2^{-3}, \ldots 2^{6}$ for the complexity parameter C. The results (see Table 4.3) are not surprising and take about 5 minutes per league to finish. Optimization did not succeed to improve the accuracy values significantly. These parameter configurations do not necessarily imply the best predictions too. Future predictions with these parameters could very well perform worse than configurations that are ranked slightly lower. Notice that the differences in between the leagues are again shown here. The Bundesliga and Pro League cannot break that 54% limit.

4.6 Recommendation tool

While the `predict` subcommand should be accounting for odds too, functionality to ignore odds and optimize for accuracy is available (see Figure 5.8). Predictions can thus be performed while explicitly disabling odds to reproduce the accuracy values for this chapter. When doing so, a LibSVM classifier is optimized for highest accuracy and trained on the matches that were played prior to the given period. This is done to obtain the highest accuracy and afterwards the predictions for every match in the given period will be generated. The output will include the calculated odds, so users are still able to compare these with published odds.

The experiment from above (see Table 4.1 where all considered models are executed with and without dimensionality reduction is supported. The baselines can also be executed manually again and could give a user some information about the predictability of a specific competition. These commands should be removed if this tool ever gets a production deploy, but is useful in this development phase.



```
bddeyn@desktop - ~/Documents/Thesis/Recommender
$ ./recommendation_tool.py predict --help
usage: recommendation_tool.py [-h] [-d <yy/mm/dd>] [--week] [--month]
                              [--days <no>] [--ignore-odds] [--experiment]
                              [--baselines] [--simulate] [--graph] [--tester]
                              [<id> [<id> ...]]

Perform predictions with the options to set dates and periods. This will
implicitely fetch new match data of already fetched seasons/tournaments.

positional arguments:
  <id>                  tournament ids to predict for, no id will predict all
                        national leagues

optional arguments:
  -h, --help            show this help message and exit
  -d <yy/mm/dd>, --date <yy/mm/dd>
                        change the beginning date of the period to predict for
  --week                predict for a period of a week
  --month               predict for a period of a month
  --days <no>           predict for a period of a given amount of days
  --ignore-odds         predict without the notion of odds
  --experiment          perform a prediction experiment
  --baselines           get the baseline predictor scores
  --simulate            perform a betting simulation
  --graph               show the simulation graph
  --tester              test the predictor temporarily
```

Figure 4.1: Help message for the extract subcommand.

4.7 Use cases

Showcasing the predictor is best done with examples. Rounds 20 and 21 of the most recent season of the Premier League and rounds 18 and 19 of the Serie A are predicted (see Table 4.4 and 4.5), resulting respectively in a 7/19 and 12/20 prediction score.

The major reason of the misclassifications for the Premier League is that in 9 out 19 matches no team was able to win. From the beginning it was known that draws are hard to predict and more than 50% of the matches resulted in a draw. Even though some of these faulty predictions were totally unexpected – Manchester United twice, Manchester City and Arsenal should have been able to win – some of the other matches would probably not have been bet on if refusing to bet was an option. A quick glance at the predicted event probabilities would indicates that the differences between them sometimes is not big enough to make a certain prediction. In this case, Everton has not been performing well this season and the differences for both matches was low (0.10 and 0.02 respectively). The same can be said for the Watford vs Leicester match with a difference of only 0.08. Only betting on matches with a certainty score bigger than 10% (marked with bold certainty scores) would have resulted in a better 7 out of 16 (=44%) prediction score.

Better results are observed for the predictions for the Italian competition. A good prediction precision is achieved, but again some of the errors could have been avoided. The 5 draws were not predicted as expected, but these losses would have been justified by the other correct predictions. It seems like two misclassified matches, the one in Verona and the away match of AC Milan, were not going to bet on by a smart bettor. Both Verona and Bologna could have easily won that match according to the generated probabilities and AC Milan has not been performing well this season. The differences in generated probabilities is 0.08 and 0.10 respectively. Using the same rule as for the Premier League would have prevented five bets here, including two of the correct predictions, resulting in a 10/15 (=66%) precision evaluation.

The conclusion is that better results, both accuracy and probably also profit wise, can be achieved by holding off on some of the more uncertain bets. A good betting strategy and money management system is still believed to be profitable for these weekends and especially for longer periods of time.

English Premier League – rounds 20 and 21						
Home team	Away team	H	D	A	Diff.	Result
Tottenham Hotspur	Southampton FC	**0.64**	0.29	0.06	0.35	**H**
Manchester United	Burnley FC	**0.59**	0.3	0.11	0.29	D
Chelsea FC	Brighton Hove Albion FC	**0.62**	0.29	0.08	0.33	**H**
AFC Bournemouth	West Ham United	**0.47**	0.23	0.3	0.17	D
Watford FC	**Leicester City**	0.33	0.26	**0.41**	**0.08**	H
West Bromwich Albion	**Everton FC**	0.33	0.24	**0.43**	**0.10**	D
Huddersfield Town	Stoke City	**0.48**	0.26	0.26	0.22	D
Liverpool FC	Swansea City	**0.64**	0.31	0.05	0.33	**H**
Newcastle United	Manchester City	0.08	0.27	**0.68**	0.34	**A**
Crystal Palace	Arsenal	0.19	0.27	**0.54**	0.27	**A**
Chelsea FC	Stoke City	**0.64**	0.3	0.06	0.34	**H**
Newcastle United	Brighton Hove Albion FC	**0.51**	0.21	0.27	0.24	D
Liverpool FC	Leicester City	**0.58**	0.27	0.14	0.31	**H**
AFC Bournemouth	**Everton FC**	0.37	0.24	**0.39**	**0.02**	H
Watford FC	Swansea City	**0.62**	0.25	0.13	0.37	A
Huddersfield Town	Burnley FC	**0.51**	0.29	0.2	0.22	D
Manchester United	Southampton FC	**0.59**	0.26	0.15	0.33	D
Crystal Palace	Manchester City	0.08	0.22	**0.7**	0.48	D
West Bromwich Albion	Arsenal	0.14	0.24	**0.63**	0.39	D

Table 4.4: Predictions and results for round 20 and 21 of the Premier League.

Italian Serie A – rounds 18 and 19						
Home team	Away team	H	D	A	Diff.	Result
AC Chievo Verona	Bologna FC	**0.41**	0.26	0.33	**0.08**	A
Cagliari Calcio	ACF Fiorentina	0.26	0.27	**0.47**	0.20	**A**
Lazio Roma	FC Crotone	**0.5**	0.27	0.23	0.23	**H**
SSC Napoli	Sampdoria Genoa	**0.51**	0.33	0.16	0.18	**H**
US Sassuolo	Inter Milan	0.28	0.3	**0.42**	0.14	H
Spal 2013	FC Torino	**0.38**	0.27	0.35	**0.03**	D
Udinese Calcio	Hellas Verona	**0.47**	0.28	0.25	0.19	**H**
Genoa FC	Benevento Calcio	**0.49**	0.26	0.24	0.23	**H**
AC Milan	Atalanta Bergamasca	0.31	0.28	**0.41**	**0.10**	**A**
Juventus Turin	AS Roma	**0.42**	0.31	0.26	0.11	**H**
FC Crotone	SSC Napoli	0.12	0.23	**0.66**	0.43	**A**
ACF Fiorentina	AC Milan	**0.48**	0.29	0.23	0.19	D
Atalanta Bergamasca	Cagliari Calcio	**0.55**	0.31	0.14	0.24	A
Benevento Calcio	AC Chievo Verona	**0.4**	0.25	0.35	**0.05**	**H**
Bologna FC	Udinese Calcio	0.29	0.28	**0.43**	0.14	**A**
AS Roma	US Sassuolo	**0.57**	0.32	0.11	0.25	D
Sampdoria Genoa	Spal 2013	**0.44**	0.28	0.28	0.16	**H**
FC Torino	Genoa FC	**0.39**	0.27	0.34	**0.05**	D
Inter Milan	Lazio Roma	**0.41**	0.3	0.29	0.11	D
Hellas Verona	Juventus Turin	0.19	0.27	**0.54**	0.27	**A**

Table 4.5: Predictions and results for round 18 and 19 of the Serie A.

Chapter 5

Maximizing profit

Evaluating models based on their accuracy has no meaning for the real betting world. A 60% accuracy predictor can still easily be unprofitable if it can only correctly predict bets with low odds. On the other hand, one with a 30% precision can be extremely profitable by forecasting the big upsets of a league. This chapter will focus on the practical side of betting which is profit maximization. A betting simulation is performed to determine which combination of classifier, betting strategy and money management is the most profitable per league over the last 20% of matches in the database. Per league, five full seasons plus the ongoing one are stored in the database. The number of evaluated predictions hence amounts to just over 1 season per league.

5.1 Betting strategies

A lot of similarities between betting and investing on stocks can be found. The first rule that any investor should know is to not lose. It sounds too simple, but it is not. Losses nullify profits that could become a significantly bigger amount due to compound interest. The same principle can be applied to gambling and sports betting. As concluded in the previous chapter, some of the losing bets can be prevented using simple rules or betting strategies to increase the total profits indirectly. Every bettor has his own set of rules and system to bet. Some of the most common are tested in this simulation.

5.1.1 Baselines

The simplest betting strategy included in the simulation does not even include a predictor. Always bet on the favorite according to the published odds and hope for the best. Other baselines – e.g. always predicting home wins, away wins or draws – were tested but quickly discarded after noticing that none of them were profitable for any of the leagues, using any of the money management systems. These baselines will show the profitability of a match and how inefficient the market is for each league.

5.1.2 Predicted favorites

The baseline betting strategy can be improved by using the probabilities generated by the model. Using the predictions will hopefully lead to bigger profits. Always bet on the favorite and again, hope for the best. This strategy can be profitable if the used model is able to predict some of the upsets and draws. Considering the profit margin bookmakers use, the assumption of an inefficient bookmakers' market can be made if it is possible to gain profit with this strategy in the long term. The model is then generated better odds than the published versions.

5.1.3 Predicted safe favorites

Making the previous strategy more robust against risk is done by assuring that a match is only bet on when the favorite has a 10% chance in winning probabilities. This way coin-flip matches are not bet on. The simulation will show if this betting strategy is able to outperform the previous one.

5.1.4 Playing the odds

Playing the odds is an actual term in the betting community. If one plays the odds, he thinks that he found a bet which the bookmakers did not predict correctly. Using the differences in estimated probabilities and probabilities calculated from published odds, a bet is made that should be profitable in the long run. Assuming that these estimated (or in our case predicted) probabilities are better, long term profitability should be guaranteed.

This is easily incorporated in the predictor as a betting strategy by betting on the event which is underestimated by the bookmakers. To prevent random bets due to very small differences, a minimum difference of 10% between the generated and published probabilities is needed to actually perform the bet.

5.1.5 Home underdogs

Since Constantinou et al. showed that betting on the home playing underdogs is profitable, this of course had to be inserted into the list of betting strategies. This uses the bias (if any) bookmakers have towards higher ranked teams, giving them overestimated odds and underestimating the effect of the home crowd. Home advantage will give the under dog an extra boost to win the game and increase their chances. Again at least a 10% difference is needed between the winning chances of the home and away playing teams to perform the bet. Big profits per bet can be expected, but many bets are also expected to fail canceling most of the profits.

5.2 Money management systems (MMs)

Money management systems (also called MMs from now on) determine the size of the stake for a bet. These systems can be really simple, but also very complex. The Markowitz portfolio management [26] would be really cumbersome to use for an end user and since the three considered systems below give enough options and perform well, it is not implemented in this simulation. A real numbered stake size between 0 and 1 is determined in the section, which can easily be multiplied by the amount of money an end user typically wants to pay per bet. Using bigger values increases the potential total profit, but also the potential total loss. The stake size is the potential loss when a bet fails for every MMs used.

5.2.1 Unit bet (UB)

The simplest way of determining each stake is to not differentiate. Every match that gets bet on receives the same sized stake. The simulations use unit bets (UBs) with a value of 1 to make it easily comparable with the other systems. This system is expected to be a high risk, high reward version, since higher risk matches get the same stake and thus a higher payout. The chances of winning that bet are also much lower. Bets with decimal odds B have a potential net payout of $B - 1$.

5.2.2 Unit return (UR)

Lowering the above mentioned risk can be done by including a correlation between the winning chances and the stake size. Instead of making sure the stakes are equal, stake size S is derived from the odds to have equal unit sized returns (URs). Potential net payout from those bets is thus $(1 - S)$, with a risk of losing S amount of money.

5.2.3 Kelly Ratio (KR)

Using the Kelly ratio [24] to determine the stake size is actually similar to playing the odds. It is typically used for long term growth in stock investing or gambling. Both the published odds in decimal form and the estimated win probability P for the match one are put into an equation, which is

$$S = \max\left(0, \frac{BP - Q}{B}\right) \quad \text{with} \quad Q = P - 1 \tag{5.1}$$

The potential net profit for this stake is then simply $S \cdot B$. The original Kelly ratio returns a negative value when the winning chances represented by the published odds are bigger than the estimated ones. While negative stakes are possible when investing in stocks – this is called shorting – negative bet sizes are not allowed in practice by bookmakers. Since betting on that event with any positive sized stake would not be profitable in the long run, a bet is simply prevented on that specific match.

Kelly originally also generalized the results of his findings to arbitrage betting, but the assumption that this is not possible anymore from Chapter 1 still stands.

Both the Kelly ratio and unit return MMs are expected to lead to less profits than the simple unit bet strategy, but will ideally also lead to a more consistent growth in the long term. A trade off between higher risk which has higher potential payouts and consistency has to be made by the end user of the tool.

5.3 Simulations

Since LibSVM was the most consistent model accuracy wise, it is also used in the profit simulation. The classifier is able to train a ternary classifier for our problem, which leads to draws mostly being ignored by the predictor. The SMO classifier handles the three classes by creating three binary predictors and combining them to achieve a ternary classifying model. This approach was expected to more frequently predict draws. Although lower accuracy values were achieved previously, it was still included in the simulation for potential bigger profits by correctly predicting draws – which typically have high odds.

The simulation hence includes two optimization procedures for both the LibSVM and SMO classifier. LibSVM's four kernels are listed again: linear, sigmoid, polynomial and radial. These are indexed from 0 to 3 respectively and will be referenced with that index in later figures. SMO's kernels PolyKernel RBF and Puk are tested, while NormalizedPolyKernel was quickly removed from the simulation because of its significantly longer execution times and disappointing profit results across all tournaments. Each model is also executed four times again to be able to thrive from the advantages of the three reduction techniques. After predicting each instance, the model is updated with that instance to improve the prediction for the following instances.

Figures 5.1, 5.2, 5.3,5.4 and 5.5 show the model with the highest profit achieved with one of the strategies for each of the five locally stored national leagues. The profit margin to calculate the odds from SportRadar's probabilities is set to 7.5%, which is good upper bound for the profit margins that most bookmakers actually use (see Table 5.1). The graphs show how much a bankroll would change over the course of x bet opportunities, using a specific betting strategy and money management system for the predictions of that model. Using another model would lead to completely different results and other strategies could possibly be more profitable for that other model.

Bookmaker	Published odds (H-D-A)	Profit margin
bet365	3.29 - 3.39 - 2.35	2.45%
Unibet	3.20 - 3.40 - 2.40	2.33%
bwin	3.10 - 3.30 - 2.40	4.23%
betfair	3.25 - 3.40 - 2.40	1.85%

Table 5.1: Odds from different bookmakers for the Arsenal vs Chelsea match of 03/01/2018 [41].

5.3.1 LaLiga

The classifier leading to the highest profit for the Spanish league is an SMO classifier with PukKernel and has a 50.44% accuracy. The results are disappointing as most of the strategies lead to tragic losses. Although most of the other strategies had a peak after 20 bets, a negative trend in the long term is noticeable. Only the strategy to bet on home playing underdogs would have led to profit. A lot of horizontal parts can be observed in its graph, which are periods of time where bets are prevented by the strategy. This indicates that the home team is mostly the favorite of the matchup. Some big payouts can also be observed where a big favorite team failed to win against one closing the rankings table.

5.3.2 Premier League

Although most strategies are profitable for the RBFKernel SMO classifier with a 58.48% accuracy for the Premier League, playing the odds with unit bets is definitely the way to go. The total profit after about 340 bet opportunities is 40 times the unit stake. Even the lower risk MMs outperform the other strategies. Both favorite betting strategies show the same peaks, indicating that the favorites were underestimated by the bookmakers for most of the matches of this league. Especially the away playing favorites won more than expected, considering that betting on the home underdogs does not seem profitable for this model and league. Betting on the favorites is also profitable for this model, but less than playing the odds. Even the baselines are profitable and were even outperforming every strategy after 150 matches. This indicates a highly profitable league.

5.3.3 Serie A

The best model for the Serie A is a sigmoid LibSVM classifier with 51% precision. The only profit would be achieved by safe betting on the favorites of the LibSVM classifier with the sigmoid kernel. Safe betting made a huge difference here, indicating a lot of close matches that turned out the wrong way for our model. Again all three MMs are profitable, with UB having more risk and higher losses at one point than the other two.

5.3.4 Bundesliga

All strategies – except for the home underdogs – are profitable for the bets classifier for the Bundesliga. Again a PukKernel SMO classifier seems to achieve the biggest profit. It is astonishing that this classifier only reaches a precision of 33.58% when predicting all matches, but produces such a big profit at the end of the simulation. Playing the odds achieves, just like for the Premier League, the most profit at the end of the simulation. UB more than quadruples the profit the other strategies achieve. Multiplying the stake with 42 after about 270 matches is huge and was not expected at all. The two other MMs perform almost as good as (safe) betting on the favorites with UBs.

5.3.5 Pro League

The same kind of result as for the Spanish League is obtained for the Belgian professional league. Although playing the odds is profitable at some points in the time line, a lot of losses cancel the temporary profits. Betting on the home playing underdogs seems to lead to consistent gain, even though only a couple of bets were placed judging by the low amount of spikes in the graph.

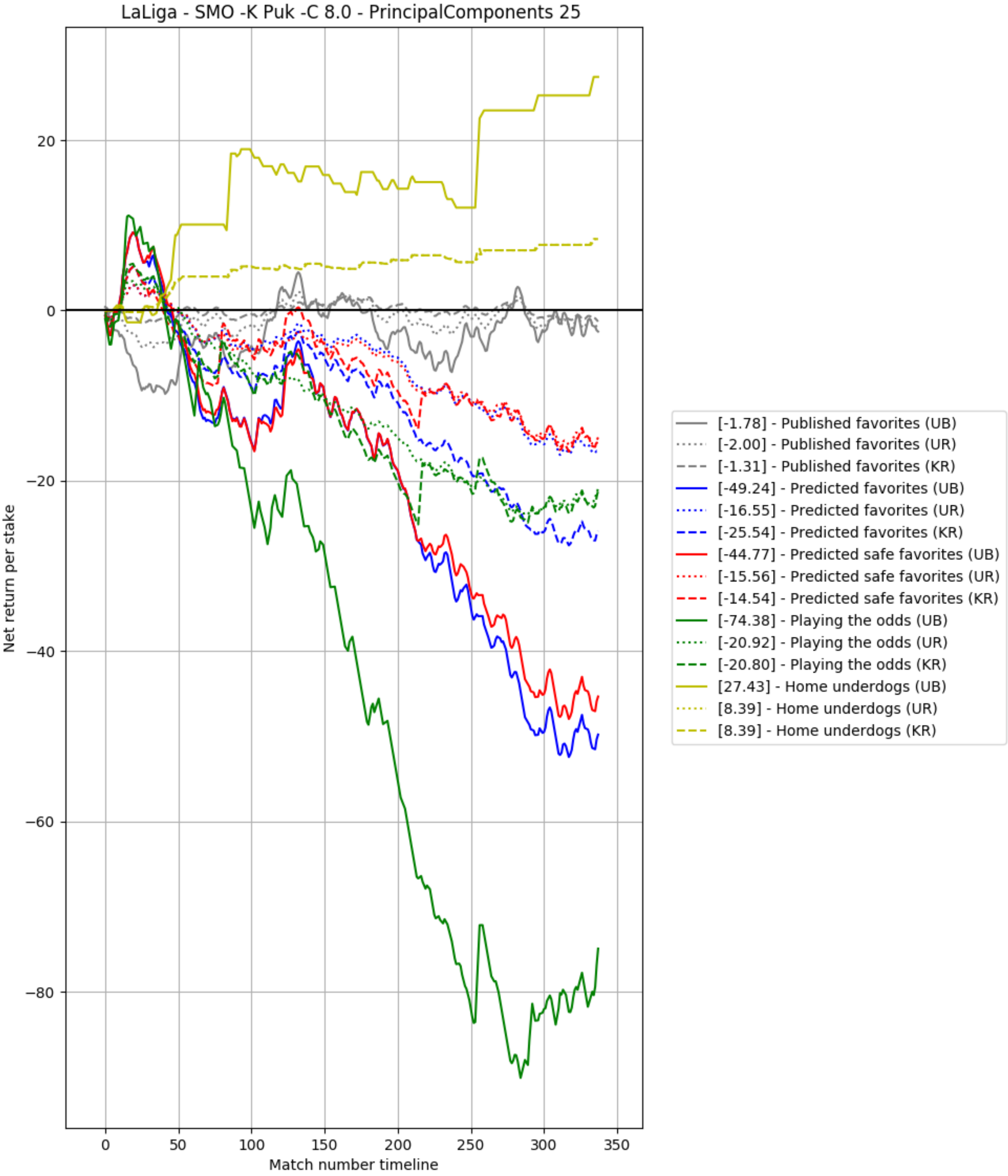


Figure 5.1: The evolution of different betting strategies for the best model on the LaLiga.

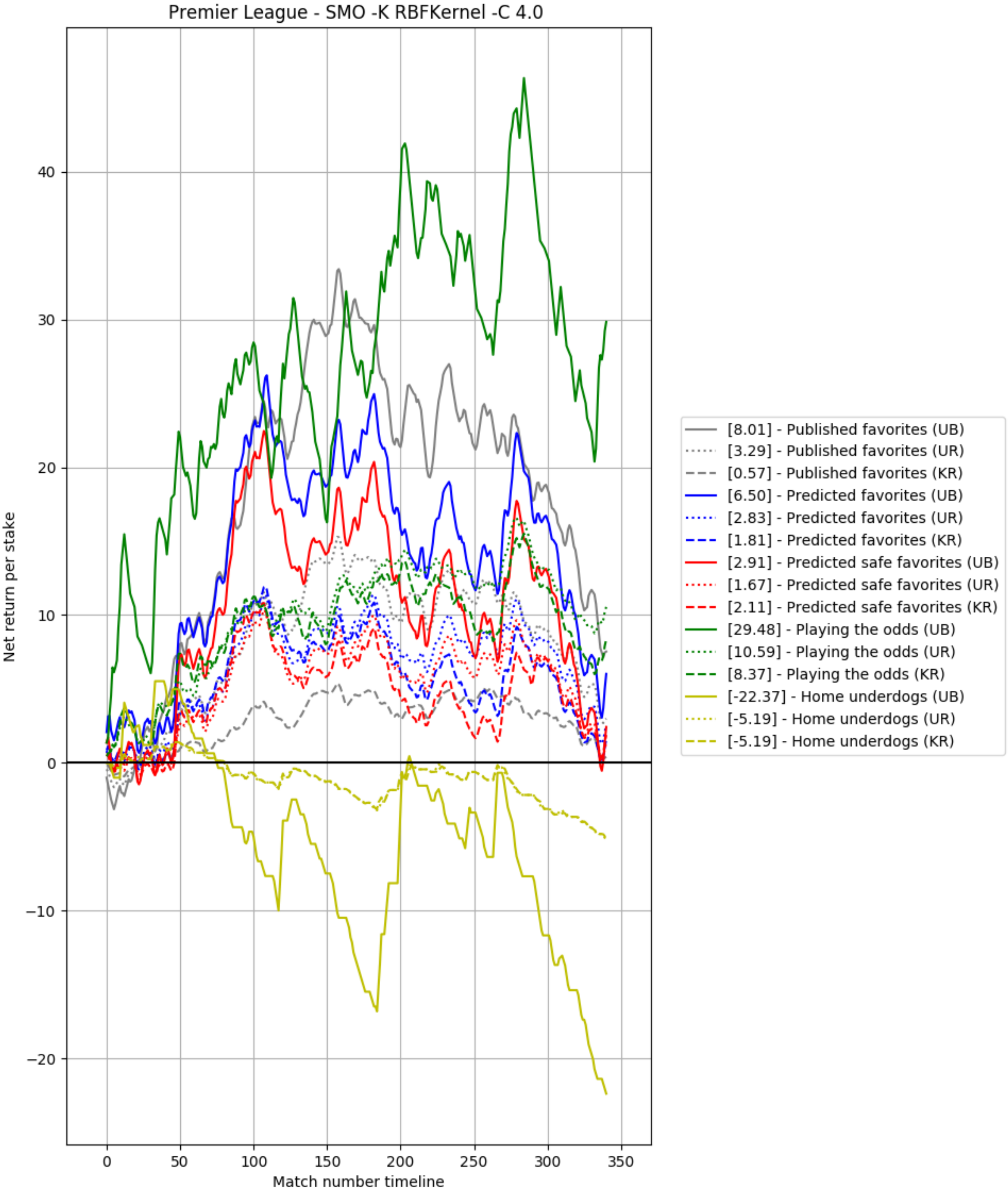


Figure 5.2: The evolution of different betting strategies for the best model on the Premier League.

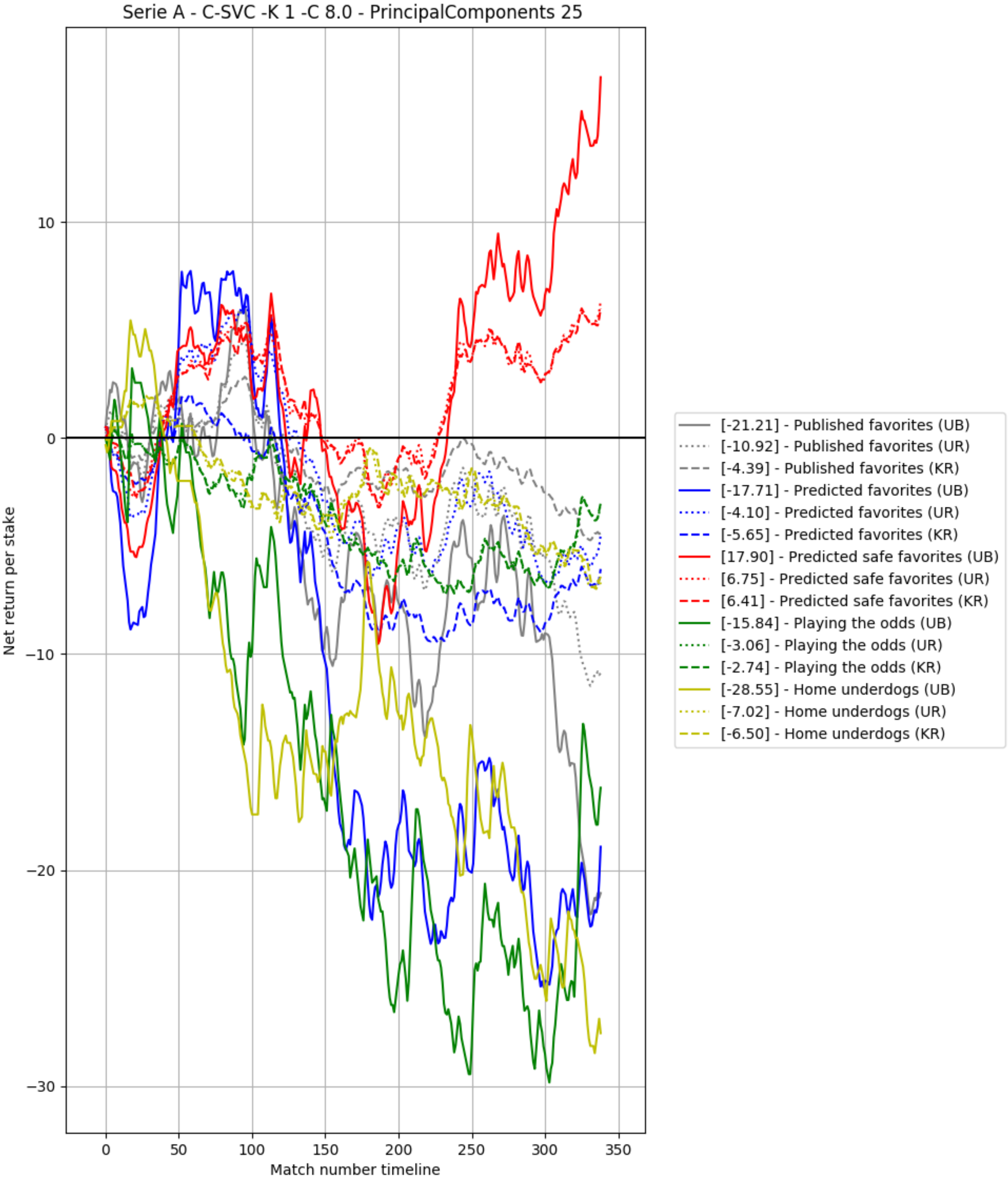


Figure 5.3: The evolution of different betting strategies for the best model on the Serie A.

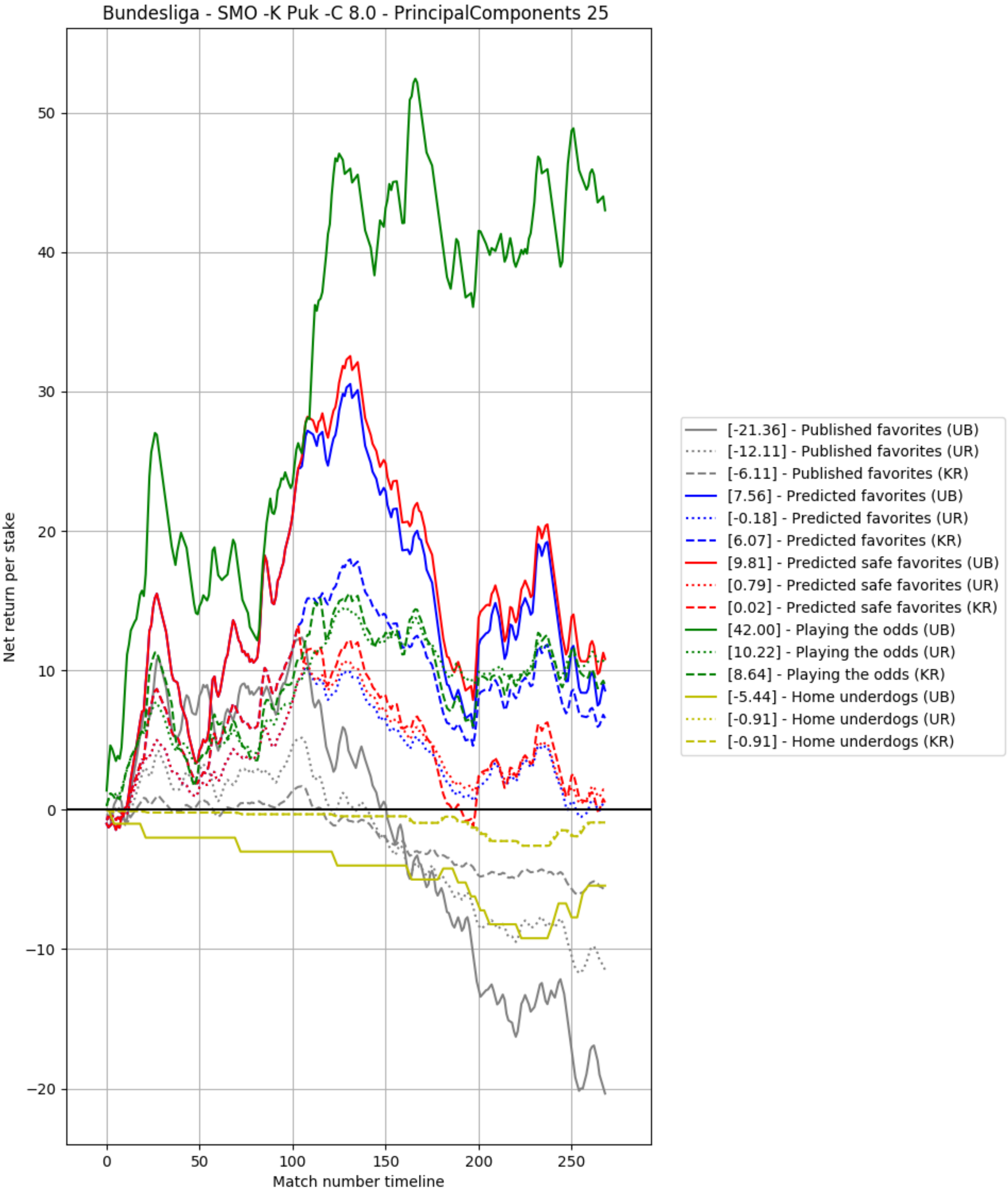


Figure 5.4: The evolution of different betting strategies for the best model on the Bundesliga.

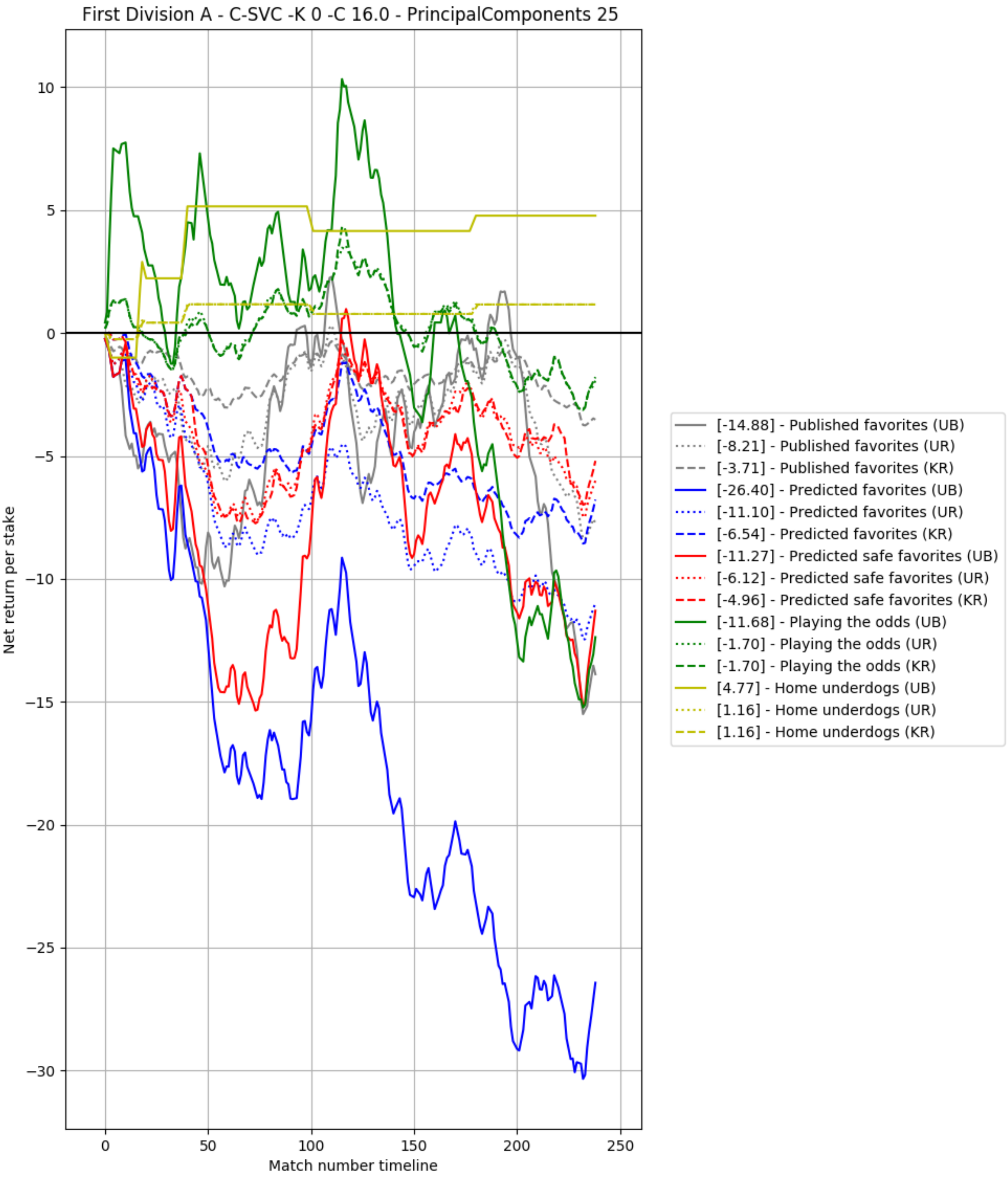
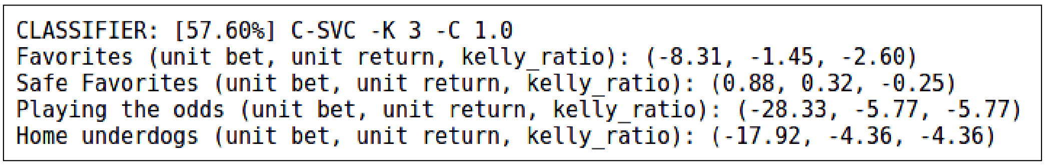


Figure 5.5: The evolution of different betting strategies for the best model on the Belgian Pro League.

5.3.6 Other results and remarks

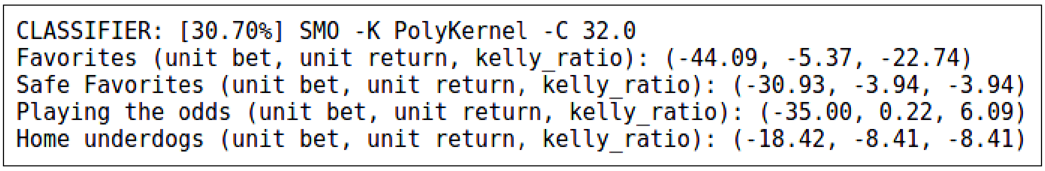
Totally unexpected, but a profitable combination of model and betting strategy is found for each league. Using the best performing model over the course of about a season to predict the next matches, gives the highest chances for profit gains in the near future. Results show that risk robust MMs prevent big losses and if used with the right model and betting strategy, the predictor shows steady growth. More model and strategy combinations turn out to be profitable and just like suspected in the introduction of this chapter, high accuracy values do not necessarily mean that a model will achieve high accuracy. Figure 5.6 shows a part of the output of the simulation for the Premier League. The 57.60% accuracy radial LibSVM classifier is only (barely) profitable using the safe favorites betting strategy.



```
CLASSIFIER: [57.60%] C-SVC -K 3 -C 1.0
Favorites (unit bet, unit return, kelly_ratio): (-8.31, -1.45, -2.60)
Safe Favorites (unit bet, unit return, kelly_ratio): (0.88, 0.32, -0.25)
Playing the odds (unit bet, unit return, kelly_ratio): (-28.33, -5.77, -5.77)
Home underdogs (unit bet, unit return, kelly_ratio): (-17.92, -4.36, -4.36)
```

Figure 5.6: Output for high accuracy model for Premier League simulation.

The contrary is also possible. A low accuracy model can surely be profitable if the right betting strategy is used, e.g. the best model for the Bundesliga (see Figure 5.4). Another example for the Premier League is shown in Figure 5.7. While the model can only predict 30% of the matches correctly, at the end of the simulation a 6.09 multiplier is achieved per stake by playing the odds and using the Kelly ratio MM.



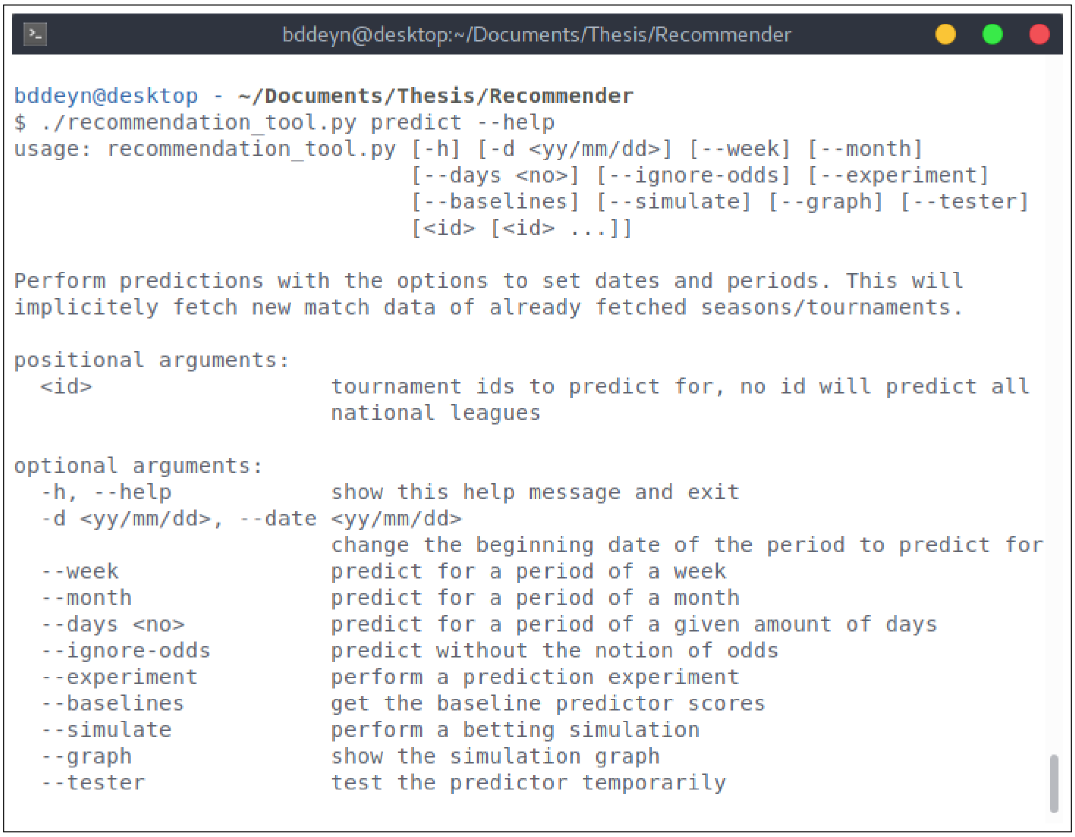
```
CLASSIFIER: [30.70%] SMO -K PolyKernel -C 32.0
Favorites (unit bet, unit return, kelly_ratio): (-44.09, -5.37, -22.74)
Safe Favorites (unit bet, unit return, kelly_ratio): (-30.93, -3.94, -3.94)
Playing the odds (unit bet, unit return, kelly_ratio): (-35.00, 0.22, 6.09)
Home underdogs (unit bet, unit return, kelly_ratio): (-18.42, -8.41, -8.41)
```

Figure 5.7: Output for low accuracy model for Premier League simulation.

The two examples above go to show that there is no linear correlation between accuracy and profit. The key for profit is using the right betting configuration for the right model for a specific league.

5.4 recommendation tool

Using the `predict` subcommand with the `--ignore-odds` flag, will use the profit simulation to determine the best model for the given tournament and predict the matches in the given period with it. The simulation can also be reproduced, as well as the time line graphs.



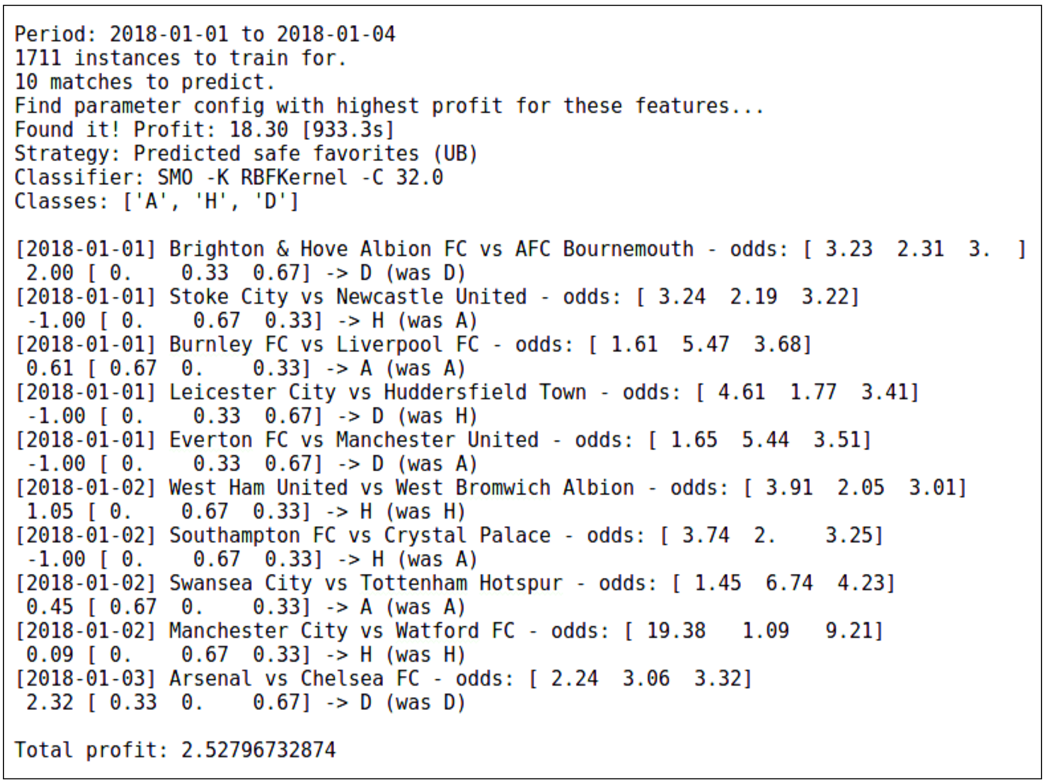
```
bddeyn@desktop - ~/Documents/Thesis/Recommender
$ ./recommendation_tool.py predict --help
usage: recommendation_tool.py [-h] [-d <yy/mm/dd>] [--week] [--month]
                              [--days <no>] [--ignore-odds] [--experiment]
                              [--baselines] [--simulate] [--graph] [--tester]
                              [<id> [<id> ...]]

Perform predictions with the options to set dates and periods. This will
implicitely fetch new match data of already fetched seasons/tournaments.

positional arguments:
  <id>                  tournament ids to predict for, no id will predict all
                        national leagues

optional arguments:
  -h, --help            show this help message and exit
  -d <yy/mm/dd>, --date <yy/mm/dd>
                        change the beginning date of the period to predict for
  --week                predict for a period of a week
  --month               predict for a period of a month
  --days <no>           predict for a period of a given amount of days
  --ignore-odds         predict without the notion of odds
  --experiment          perform a prediction experiment
  --baselines           get the baseline predictor scores
  --simulate            perform a betting simulation
  --graph               show the simulation graph
  --tester              test the predictor temporarily
```

Figure 5.8: Help message for the extract subcommand.

Predicting some matches of a league is easy. Figure on what dates the matches you want to bet on are going to take place, get the id of the tournament (17 in case of the Premier League) and execute the command like this:

```
./recommendation_tool.py predict 17 -d 18/01/01 --days 3
```

The result of this command will show up after about 15 minutes – it indeed takes a while to classify all models and pick the best performing one. Afterwards the actual betting and comparing by the end user can start. In our case (see Figure 5.9), matches from the recent history are predicted to show the performance of the predictor. The RBFKernel SMO classifier with $C = 32$ is chosen after the simulation of 20% of the 1711 historic instances finishes. Some mispredictions occurred, but together with the profits of the winning bets, net profit would have been made. A profit multiplier of 2.53 would have been achieved if one would have used this model and strategy again for these matches.



```
Period: 2018-01-01 to 2018-01-04
1711 instances to train for.
10 matches to predict.
Find parameter config with highest profit for these features...
Found it! Profit: 18.30 [933.3s]
Strategy: Predicted safe favorites (UB)
Classifier: SMO -K RBFKernel -C 32.0
Classes: ['A', 'H', 'D']

[2018-01-01] Brighton & Hove Albion FC vs AFC Bournemouth - odds: [ 3.23  2.31  3.  ]
 2.00 [ 0.    0.33  0.67] -> D (was D)
[2018-01-01] Stoke City vs Newcastle United - odds: [ 3.24  2.19  3.22]
 -1.00 [ 0.    0.67  0.33] -> H (was A)
[2018-01-01] Burnley FC vs Liverpool FC - odds: [ 1.61  5.47  3.68]
 0.61 [ 0.67  0.    0.33] -> A (was A)
[2018-01-01] Leicester City vs Huddersfield Town - odds: [ 4.61  1.77  3.41]
 -1.00 [ 0.    0.33  0.67] -> D (was H)
[2018-01-01] Everton FC vs Manchester United - odds: [ 1.65  5.44  3.51]
 -1.00 [ 0.    0.33  0.67] -> D (was A)
[2018-01-02] West Ham United vs West Bromwich Albion - odds: [ 3.91  2.05  3.01]
 1.05 [ 0.    0.67  0.33] -> H (was H)
[2018-01-02] Southampton FC vs Crystal Palace - odds: [ 3.74  2.    3.25]
 -1.00 [ 0.    0.67  0.33] -> H (was A)
[2018-01-02] Swansea City vs Tottenham Hotspur - odds: [ 1.45  6.74  4.23]
 0.45 [ 0.67  0.    0.33] -> A (was A)
[2018-01-02] Manchester City vs Watford FC - odds: [ 19.38   1.09   9.21]
 0.09 [ 0.    0.67  0.33] -> H (was H)
[2018-01-03] Arsenal vs Chelsea FC - odds: [ 2.24  3.06  3.32]
 2.32 [ 0.33  0.    0.67] -> D (was D)

Total profit: 2.52796732874
```

Figure 5.9: Example output predicting round 22 of the Premier League.

Actually predicting future matches for which the result is not known yet is not much different. The profits cannot be shown (yet), but suggestions are output: what betting strategy should be used and what are the stakes to put on each match? The published odds on which the suggestions are based are also presented. This way the end user can compare them to other bookmakers and change their betting behavior based on his findings.

5.5 Use cases

The use cases from last chapter are predicted again here with the focus on profit. The outputs of the predictions are again parsed into the tables for a better overview. These show the published odds for those matches and the predicted event probabilities by the classifiers. The odds that were bet on are highlighted with bold text. The payout/loss is represented in the last column.

The best model up until the 20th round of the Premier League was not able to turn around the bad accuracy for round 20 and 21 achieved previously (see Table 5.2). Unit bets on the favorites are thought to be the best MM. No draws were predicted and thus only 10 matches are left to be predicted with either home or away wins. Apart from these draw results, one big upset happened according to the published odds, i.e. the second considered home match of Watford (highlighted with bold text). The model did not see this coming, as well as the result of Watford's first match (also highlighted). A win for Leicester – which would have been an upset according to the published odds – was forecasted. Both hence resulted in a loss, totaling to a prediction accuracy of 7/19 matches for these two weekends. Using the safer strategy while betting on favorites would not have made a difference here, because the model always had a clear favorite for each of the matchups. A total profit multiplier of would be the result if one bet on all 19 matches. Choosing 19 1$ bets would leave you behind with a little more than 11 bucks.

The losses that would have occurred with bets on this weekend, would partially be reverted with bets on the next round of the Premier League (see Figure 5.9). Due to the many draws in the previous rounds, another model able to predict draws was chosen for the 22th round of the Premier League, resulting in 4/10 predictions to be draws. Two of them were correct, two of them were not. The overall profit of those draw predictions is 2.32, so the predictor made a good choice with the SMO classifier. Most of the favorites won their games, which leads to an overall profit of 2.53 for that round.

Contrary to above, the two rounds of the Italian league during the same period would have resulted in profit. The good accuracy achieved in Chapter 4 thus results in profit. 2.72 times the unit stakes on the favorites would have been the final profit after the two weekends. The algorithm was able to correctly predict three upsets (marked with bold text). The lost bets were either draws or the away playing favorites experiencing the home advantage and not being to take the win back home.

English Premier League – rounds 20 and 21							
Home team	**Away team**	**Published odds (H-D-A)**	**H**	**D**	**A**	**Result**	**Profit**
Tottenham Hotspur	Southampton FC	**1.36** - 4.72 - 7.88	0.64	0.29	0.07	H	**0.36**
Manchester United	Burnley FC	**1.17** - 6.24 - 15.77	0.6	0.3	0.11	D	-1.00
Chelsea FC	Brighton Hove Albion FC	**1.17** - 6.46 - 15.77	0.6	0.3	0.1	H	**0.17**
AFC Bournemouth	West Ham United	**2.36** - 3.20 - 2.95	0.49	0.23	0.27	D	-1.00
Watford FC	**Leicester City**	2.30 - 3.36 - **2.93**	0.3	0.27	0.43	H	-1.00
West Bromwich Albion	Everton FC	2.41 - 3.04 - **3.02**	0.31	0.26	0.44	D	-1.00
Huddersfield Town	Stoke City	**2.54** - 3.07 - 2.81	0.48	0.26	0.26	D	-1.00
Liverpool FC	Swansea City	**1.14** - 7.32 - 15.50	0.67	0.27	0.05	H	**0.14**
Newcastle United	Manchester City	10.82 - 6.16 - **1.22**	0.09	0.26	0.66	A	**0.22**
Crystal Palace	Arsenal	4.43 - 4.04 - **1.66**	0.14	0.3	0.56	A	**0.66**
Chelsea FC	Stoke City	**1.09** - 8.78 - 21.14	0.63	0.29	0.07	H	**0.09**
Newcastle United	Brighton Hove Albion FC	**2.04** - 3.10 - 3.83	0.57	0.2	0.23	D	-1.00
Liverpool FC	Leicester City	**1.21** - 6.64 - 10.45	0.59	0.26	0.15	H	**0.21**
AFC Bournemouth	Everton FC	**2.27** - 3.08 - 3.22	0.45	0.23	0.32	H	**1.27**
Watford FC	**Swansea City**	**1.77** - 3.43 - 4.58	0.63	0.25	0.12	A	-1.00
Huddersfield Town	Burnley FC	**2.30** - 2.88 - 3.41	0.48	0.32	0.19	D	-1.00
Manchester United	Southampton FC	**1.30** - 5.08 - 9.12	0.58	0.26	0.17	D	-1.00
Crystal Palace	Manchester City	7.95 - 5.60 - **1.30**	0.07	0.24	0.69	D	-1.00
West Bromwich Albion	Arsenal	5.29 - 4.01 - **1.57**	0.1	0.26	0.61	D	-1.00
						Total profit	**-7.87**

Table 5.2: Predictions and results for round 20 and 21 of the Premier League.

Italian Serie A – rounds 18 and 19

Home team	Away team	Published odds (H-D-A)	H	D	A	Result	Profit
AC Chievo Verona	**Bologna FC**	2.25 - 3.09 - **3.26**	0.34	0.21	0.45	A	**2.26**
Cagliari Calcio	ACF Fiorentina	3.10 - 3.38 - **2.19**	0.13	0.26	0.61	A	**1.19**
Lazio Roma	FC Crotone	**1.20** - 6.37 - 12.24	0.56	0.3	0.14	H	**0.20**
SSC Napoli	Sampdoria Genoa	**1.23** - 6.20 - 10.22	0.53	0.32	0.15	H	**0.23**
US Sassuolo	Inter Milan	4.47 - 3.78 - **1.70**	0.13	0.26	0.61	H	-1.00
Spal 2013	FC Torino	3.47 - 3.41 - **2.03**	0.24	0.29	0.47	H	-1.00
Udinese Calcio	Hellas Verona	**1.63** - 3.86 - 4.95	0.44	0.29	0.27	H	**0.63**
Genoa FC	Benevento Calcio	**1.39** - 4.45 - 7.44	0.46	0.28	0.26	H	**0.39**
AC Milan	**Atalanta Bergamasca**	2.43 - 3.24 - **2.82**	0.23	0.27	0.5	A	**1.82**
Juventus Turin	AS Roma	**1.77** - 3.58 - 4.37	0.36	0.31	0.33	H	**0.77**
FC Crotone	SSC Napoli	15.00 - 6.64 - **1.17**	0.04	0.14	0.82	A	**0.17**
ACF Fiorentina	AC Milan	**2.38** - 3.20 - 2.92	0.5	0.29	0.21	D	-1.00
Atalanta Bergamasca	Cagliari Calcio	**1.41** - 4.45 - 7.16	0.61	0.29	0.1	A	-1.00
Benevento Calcio	AC Chievo Verona	2.78 - 3.29 - **2.44**	0.36	0.27	0.36	H	-1.00
Bologna FC	**Udinese Calcio**	2.37 - 3.07 - **3.06**	0.22	0.25	0.52	A	**2.06**
AS Roma	US Sassuolo	**1.23** - 6.08 - 10.00	0.6	0.3	0.1	D	-1.00
Sampdoria Genoa	Spal 2013	**1.82** - 3.45 - 4.25	0.54	0.28	0.18	H	**0.82**
FC Torino	Genoa FC	**2.27** - 3.12 - 3.17	0.43	0.27	0.3	D	-1.00
Inter Milan	Lazio Roma	**2.25** - 3.48 - 2.91	0.48	0.35	0.16	D	-1.00
Hellas Verona	Juventus Turin	14.53 - 6.42 - **1.18**	0.1	0.22	0.68	A	**0.18**
						Total profit	**2.72**

Table 5.3: Predictions and results for round 18 and 19 of the Serie A.

Chapter 6

Conclusion

6.1 Main results

Only goal related statistics are found to be important for a teams' near future performance. It makes sense because a team cannot win without scoring in matches. The most important features represent the form, the recent results, the average goals in recent matches, the recent performance of the two teams playing against each other etc. Support vector classifiers were found to be the most consistent to achieve high accuracy values and were researched for profit afterwards.

A prototype of a recommendation tool has been engineered, which can help a bettor with betting suggestions. This can include the teams of the matchups to bet on, as well as the betting strategies or money management systems to use for a specific league. The predictor is based on two support vector machine models whose parameters get optimized for maximum profit. The best model on that moment for a specific league is used for predictions and stakes for each game to bet on are suggested. This tool should be used as an assist and should not be blindly followed by any end user. Insider information about a team could lead to a correctly predicted upset.

Every five of the national leagues researched have been proven profitable to bet if the correct model, betting strategy and money management system are used. The potential total payouts per league after a full season of betting are shown in Table 6.1. The bookmaker market is thus shown to still be inefficient. Better odds can

League	Profit	Model	Betting strategy
Spanish LaLiga	27.43	SMO PukKernel C=8.0 with PCA	Home underdogs
English Premier League	29.48	SMO RBFKernel C=4.0	Playing the odds
Italian Serie A	17.90	LibSVM sigmoid kernel C=8.0 with PCA	Predicted safe favorites
German Bundesliga	42.00	SMO PukKernel C=8.0 with PCA	Playing the odds
Belgian Pro League	4.77	LibSVM linear kernel C=16.0 with PCA	Home underdogs

Table 6.1: Final highest profits after executing the simulation.

be generated and discrepancies in generated and published odds can be abused for guaranteed profit in the long run.

6.2 Further work

A lot of ideas for further work come to mind. Both the research aspect and the engineering of the tool can be improved.

6.2.1 Research

More data and features can definitely be added. More data sets or data providers (e.g. Tonsser [42], which has been called the LinkedIn for football players) could lead to more information about matches and players, resulting in better features. A decent data license with SportRadar would have to be negotiated for this tool instead of using trial accounts too. Possible features to be added are discussed in Section 3.4. More strategies and money management systems could surely be thought of and researched. There really are a lot of ways to take this research to the next level.

The feature extractor and classifiers can also be used by football coaches whom are looking to train on the weak points of their team. After removing the obvious goal related features, indicators of a team's performance could be exposed with an attribute selection procedure or specifically designed classifier.

This research can also easily be extended to the stock market. A lot of similarities between investing in stocks and sports betting were discussed in this thesis. Adding a different data source and feature extractor should be enough for a basic stock recommender.

6.2.2 Tool development

There is a ton of small improvements that could improve the user friendliness of the sports bet recommendation tool. For starters, a GUI or web interface could make it commercially viable. Many performance updates are also possible (e.g. caching models that have been trained will reduce most of the execution times, etc) More prediction options could also make it more flexible (e.g. a tournament round parameter, a team id parameter etc.)

List of Figures

List of Tables

Bibliography

[1] R. Praet, “Predicting Sport Results by using Recommendation Techniques,” Ph.D. dissertation, Ghent University, 2017.

[2] M. Maher, “Modelling association football scores,” Statistica Neerlandica, vol. 36, no. 3, pp. 109–118, 1982.

[3] A. C. Constantinou, N. E. Fenton, and M. Neil, “Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks,” Knowledge-Based Systems, vol. 50, pp. 60–86, sep 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095070511300169X

[4] “Odds - Wikipedia.” [Online]. Available: https://www.wikiwand.com/en/Odds

[5] “Fixed-odds betting - Wikipedia.” [Online]. Available: https://www.wikiwand.com/en/Fixed-odds_betting

[6] M. J. Dixon and S. G. Coles, “Modelling Association Football Scores and Inefficiencies in the Football Betting Market,” Journal of the Royal Statistical Society: Series C (Applied Statistics), vol. 46, no. 2, pp. 265–280, 1997. [Online]. Available: http://doi.wiley.com/10.1111/1467-9876.00065

[7] T. J. Dohmen, “The influence of social forces: Evidence from the behavior of football referees,” Economic Inquiry, vol. 46, no. 3, pp. 411–424, jul 2008. [Online]. Available: http://doi.wiley.com/10.1111/j.1465-7295.2007.00112.x

[8] A. C. Constantinou, N. E. Fenton, and L. J. Hunter Pollock, “Bayesian networks for unbiased assessment of referee bias in Association Football,” Psychology of Sport and Exercise, vol. 15, no. 5, pp. 538–547, 2014.

[9] B. Buraimo, D. Forrest, and R. Simmons, “The 12th man?: Refereeing bias in English and German soccer,” Journal of the Royal Statistical Society. Series A: Statistics in Society, vol. 173, no. 2, pp. 431–449, apr 2010. [Online]. Available: http://doi.wiley.com/10.1111/j.1467-985X.2009.00604.x

[10] H. Liu, W. G. Hopkins, and M. A. Gómez, “Modelling relationships between match events and match outcome in elite football,” European Journal of Sport Science, vol. 16, no. 5, pp. 516–525, 2016.

[11] J. Oberstone, “Differentiating the Top English Premier League Football Clubs from the Rest of the Pack: Identifying the Keys to Success,” Journal of Quantitative Analysis in Sports, vol. 5, no. 3, 2009. [Online]. Available: https://www.degruyter.com/view/j/jqas.2009.5.3/jqas.2009.5.3.1183/jqas.2009.5.3.1183.xml

[12] B. Min, J. Kim, C. Choe, H. Eom, and R. I. (Bob) McKay, “A compound framework for sports results prediction: A football case study,” Knowledge-Based Systems, vol. 21, no. 7, pp. 551–562, oct 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705108000609

[13] M. Carpita, M. Sandri, A. Simonetto, and P. Zuccolotto, “Discovering the drivers of football match outcomes with data mining,” Quality Technology and Quantitative Management, vol. 12, no. 4, pp. 561–577, 2016.

[14] N. Tax and Y. Joustra, “Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approach,” Transaction on knowledge and data engineering, vol. X, no. SEPTEMBER, pp. 1–13, jan 2015.

[15] M. Spann and B. Skiera, “Sports forecasting: A comparison of the forecast accuracy of prediction markets, betting odds and tipsters,” Journal of Forecasting, vol. 28, no. 1, pp. 55–72, jan 2009. [Online]. Available: http://doi.wiley.com/10.1002/for.1091

[16] A. C. Constantinou, N. E. Fenton, and M. Neil, “Pi-football: A Bayesian network model for forecasting Association Football match outcomes,” Knowledge-Based Systems, vol. 36, pp. 322–339, 2012.

[17] S. J. S. Koopman and R. Lit, “A dynamic bivariate Poisson model for analysing and forecasting match results in the English Premier League,” Journal of the Royal Statistical Society. Series A: Statistics in Society, vol. 178, no. 1, pp. 167–186, 2015.

[18] G. Boshnakov, T. Kharrat, and I. G. McHale, “A bivariate Weibull count model for forecasting association football scores,” International Journal of Forecasting, vol. 33, no. 2, pp. 458–466, 2017.

[19] K. W. Chow and K. Tan, "The use of profits as opposed to conventional forecast evaluation criteria to determine the quality of economic forecasts," Applied Economics Research Series, vol. 1, pp. 187–200, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.1

[20] H. S. Shin, "Measuring the Incidence of Insider Trading in a Market for State-Contingent Claims," The Economic Journal, vol. 103, no. 420, p. 1141, 1993. [Online]. Available: http://www.jstor.org/stable/2234240?origin=crossref

[21] E. Štrumbelj, "On determining probability forecasts from betting odds," International Journal of Forecasting, vol. 30, no. 4, 2014.

[22] D. Forrest, J. Goddard, and R. Simmons, "Odds-setters as forecasters: The case of English football," International Journal of Forecasting, vol. 21, no. 3, pp. 551–564, jul 2005. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169207005000300

[23] A. C. Constantinou and N. E. Fenton, "Profiting from arbitrage and odds biases of the European football gambling market," Journal of Gambling Business and Economics, vol. 7, no. 2, pp. 1–22, 2013. [Online]. Available: http://www.constantinou.info/downloads/papers/evidenceOfInefficiency.pdf

[24] J. L. Kelly, "A New Interpretation of Information Rate," Bell System Technical Journal, vol. 35, no. 4, pp. 917–926, jul 1956. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6771227

[25] H. Rue and Ø. Salvesen, "Prediction and Retrospective Analysis of Soccer Matches in a League Prediction and Retrospective Analysis of Soccer Matches in a League Håvard Rue," Norges Teknisk, 1998. [Online]. Available: http://www.math.ntnu.no/preprint/statistics/1997/S10-1997.ps%5Cnhttp://www.math.ntnu.no/

[26] H. Markowitz, "Portfolio Selection," The Journal of Finance, vol. 7, no. 1, pp. 77–91, mar 1952. [Online]. Available: http://doi.wiley.com/10.1111/j.1540-6261.1952.tb01525.x

[27] H. Langseth, "Beating the bookie: A look at statistical models for prediction of football matches," in Frontiers in Artificial Intelligence and Applications, vol. 257, 2013, pp. 165–174.

[28] "Sports data services - SportRadar." [Online]. Available: https://sportradar.com/

[29] "Soccer Data Documentation - SportRadar." [Online]. Available: https://developer.sportradar.com/files/indexSoccer.html#soccer-extended-api-v3

[30] "API Sandbox - SportRadar." [Online]. Available: https://developer.sportradar.com/io-docs

[31] "Three points for a win - Wikipedia." [Online]. Available: https://www.wikiwand.com/en/Three_points_for_a_win#/History

[32] "FlashScore." [Online]. Available: https://www.flashscore.com/

[33] T. U. of Waikato, "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[34] E. Frank, M. A. Hall, and I. H. Witten, "The WEKA Workbench," Morgan Kaufmann, Fourth Edition, pp. 553–571, 2016. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

[35] Peter "fracpete" Reutemann, "Python Weka Wrapper 0.3.10," 2014. [Online]. Available: http://pythonhosted.org/python-weka-wrapper/

[36] R. C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets," Machine Learning, vol. 11, no. 1, pp. 63–90, 1993.

[37] "Classifiers Documentation - WEKA." [Online]. Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/Classifier.html

[38] "AttributeSelection Documentation - WEKA." [Online]. Available: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/package-summary.html

[39] "C-SVC vs nu-SVC | LibSVM FAQ." [Online]. Available: https://www.csie.ntu.edu.tw/$\sim$cjlin/libsvm/faq.html#f411

[40] "Kernels Documentation - WEKA." [Online]. Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/supportVector/Kernel.html

[41] "Odds Arsenal vs. Chelsea - OddsPortal." [Online]. Available: http://www.oddsportal.com/soccer/england/premier-league/arsenal-chelsea-Y1CoaD3J/

[42] "Tonsser." [Online]. Available: https://tonsser.com/