Amsterdam School of Economics
Faculty of Economics and Business

Improving the estimation of outcome probabilities of football matches using in-game information

Rogier Noordman

Student number:	12366315
Date of final version:	12th July 2019
Master's programme:	Econometrics
Specialisation:	Data Science & Business Analytics
Supervisor:	mw. dr. K. A. Lasak
Second reader:	dr. J. C. M. Van Ophem
SciSports supervisor:	dr. ir. J. Van Haaren
SciSports supervisor:	ir. B. J. Aalbers

Faculty of Economics and Business

Abstract

This thesis aims to improve the estimation of outcome probabilities of football matches using in-game information. Expected-goals is a widely used shot metric which is also used to estimate match outcome probabilities. The summed expected-goals values are used for this estimation, however only using this misses a lot of information. This thesis shows that by adding player-specific information to the expected-goals model the accuracy of the expected-goals model improves. Next to this, adding match statistics improves the estimation of the match outcome probabilities in comparison with models that only use expected-goals values. Exploiting the temporal aspect of the expected-goals values improves the estimation of the match outcome probabilities even further. This thesis also shows how these results can be used in practice and that the predictions on league rankings outperform predictions based on the the actual league ranking.

Statement of Originality

This document is written by Rogier Noordman who declares to take full responsibility for the contents of this document. I declare that the text and the work presented in this document is original and that no sources other than those mentioned in the text and its references have been used in creating it. The Faculty of Economics and Business is responsible solely for the supervision of the work, not the contents.

Contents

Chapter 1

Introduction

1.1 Context

Over the past decade, data and statistics have become more popular in the world of association football (better known as football). Football has been regarded as a conservative industry, but over the last couple of years important decisions as player recruitment (Burn-Murdoch, 2018) or match preparation (SciSports, 2018) are based on data instead of intuition. An important factor of this change in mindset is caused by the success of the use of data in baseball (Lewis, 2003), which was based on the research done by James (1986). Bill James is often regarded as one of the "founding fathers" of sports analytics because of his search for objective knowledge about baseball.

Football has a low-scoring nature, which makes that randomness and luck often have a big impact on the scoreline and the end result (Lucey, Bialkowski, Monfort, Carr & Matthews, 2015). As a result, the final score does not always reflect the actual performances of the two teams. It is interesting to gain insights in the actual performances of teams in order to see whether the obtained results are a result of the shown performance or on luck or randomness. In the long run, luck or randomness cancel out and for a football team this can for example make the difference between getting relegated or not. Because of this, over the past couple of years an increasing number of metrics (SciSports, 2016; Hamilton, 2017; Fernandez and Bornn, 2018) have been developed in order to get a better understanding of the game. Since shots are a more common event than goals, shot-based metrics are less based on randomness and hence considered as a better predictor than goal-based metrics (Goodman, 2018).

The shot-based metric that is most popular and used on television (Stanton, 2017) and by newspapers (Guardian-Sport, 2019) is expected-goals. The expected-goals metric reflects the probability of a given goal-scoring opportunity to be converted into a goal. Based on historic information about shots and its outcome under similar circumstances, it gives an indication of the quality of the opportunity. If, for example, 20% of the historic shots under similar circumstances resulted in a goal, then the corresponding expected-goals value is 0.20. Variables used to describe these circumstances are for example the distance to the goal, the angle to the

goal or whether it was a headed or a normal shot.

Aggregations of expected-goals over a match per team can also be used to estimate the outcome probabilities of a game (Eggels, 2016; Van den Hoek, 2019). If match outcomes are estimated in this way, it gives more insights in the shown performance of a team. If these estimated match outcomes are aggregated over multiple matches, it is also possible to see which teams have been over-performing or under-performing based on the expected-goals values that are calculated. These tables based on the expected-results are used by media (Stanley, 2019) in order to get an alternative table of the top football competitions. This information is also used in betting (Punter2Pro, 2017; bet.me, 2018) in order to explore possible interesting betting opportunities.

1.2 Problem statement

Expected-goals values have shown to be a good predictor of future performances (Ijtsma, 2015), but it also has its limitations (Cronin, 2019). These limitations are already in the definition of expected-goals: it is based on a big amount of historic shots and not on the context of a specific game or player. The goal of expected-goals is to approximate how many times a certain shot results in a goal and be able to assess performance of players over time, not to predict the outcome of one specific game.

Next to this, also the aggregation of expected goals values over a game is questionable (Page, 2015; Gurpinar-Morgan, 2015). Since expected-goals are probabilities, only summing up the values misses the story of variance. Two teams with equal summed expected-goals values could have different win probabilities because of the number of chances they had to get this summed expected-goals value. For example, if a home team creates no chances during a game except for one very big chance and the away team gets a lot of small chances which is summed up the same amount of expected goals over a game. The probability that the home team scores one goal is very big, but the probability of the home team scoring two goals is zero. On the other hand, the possible amount of goals the away team scored is very big, since it is possible that every little chance they got went in. This example shows the variance of expected-goals values: this sort of information is not captured if only the summed expected-goals values are used.

In this thesis the goal is to improve the estimation of outcome probabilities of football matches by adding both player-specific and match-specific information. Since the goal of estimating the probabilities of match outcomes is different from the original purpose of expected-goals, adding player specific information is allowed and could be beneficial. As shown before it is important to obtain a match-specific evaluation in order to get an estimation of the performance of a specific team instead of just looking at the final score. Next to this, in this thesis the temporal aspect of the expected-goals values is examined. As indicated before, simply adding up the expected-goals is not optimal, so it is examined whether considering time effects of these expected-goals values improve the estimation of outcome probabilities of football matches. For example, if a team gets a lot of big chances very early in the game and in the rest of the game

little to no chances, this might indicate that the team got in front and the rest of the game defended the result. By considering the temporal aspect of expected-goals values, this kind of information can be obtained without observing the amount of goals the teams actually scored.

1.3 Research questions

Following the problem statement, this thesis addresses the following research question:

\Does using more advanced in-game statistics improve the estimation of outcome probabilities of football matches?"

Currently, all models that estimate match outcome probabilities are solely based on summed expected-goals values. In the first step, an expected-goals model with player-specific information is compared to a model without the player-specific information. This analyzes whether the quality of a player has influence on the likelihood of a opportunity resulting in a goal. If adding the player-specific information is beneficial for the expected-goals model, it also makes the estimated probabilities of match outcomes more reliable.

Adding match-specific information could create extra context about a game which improves the estimation of match outcome probabilities based on the expected-goals values. In first instance, this could be general match statistics as possession, number of shots or number of red cards. Next to this, the temporal aspect of the expected-goals values could improve the estimation of outcome probabilities. By using the time labels of the expected-goals values, some patterns might be found that have an influence on the outcome of a game.

Finally, the choice of the learning algorithm is important (Dey, 2016). In modern-day machine learning it is very important to choose the proper model for each situation. Therefore in this thesis it is also investigated which learning algorithms are best to produce accurate values for individual shots and matches.

Based on this, the following sub-questions are also addressed:

1. *\Does the prediction accuracy of expected-goals models improve by adding player-speci c information?"*
2. *\Does the estimation of outcome probabilities of football matches improve by adding match statistics?"*
3. *\Does the estimation of outcome probabilities of football matches improve by considering the temporal aspect of expected-goals values?"*

1.4 Outline

The remainder of this thesis is structured in the following way. Section 2 provides a review of the existing literature and information regarding the learning algorithms that are used in this thesis. After that, in section 3 the data that is used in this thesis is considered per data source. In section 4 the methodology of the research process is explained and in section 5 the corresponding results are presented. Section 6 discusses and concludes the results from this thesis.

Chapter 2

Literature Review

This section describes the existing literature of the different subjects that are discussed in this thesis. Firstly, the literature and techniques in expected-goals models and match result evaluation are described. Secondly, the various machine learning algorithms than can be used for the expected-goals models and the match result evaluation are described. Finally, different methods to evaluate the performance of the machine learning models for the expected-goals models and the match result evaluation are discussed.

2.1 Expected-goals models

It is not exactly known who was the first to come up with expected-goals models, but Pollard, Ensum and Taylor (2004) were one of the first to indicate the estimation of the probability of a shot resulting in a goal. They use a logistic regression analysis with three different variables: the distance from the goal, the angle from the goalpost and the space from the nearest opponent at the time of the shot. Pollard et al. (2004) indicate that there are more factors that have an influence, but that this model is the basis for different usages such as quantifying the effectiveness of specific tactics or quantifying the capabilities of individual strikers and goalkeepers.

The first time the term "expected-goals" appeared was in a paper about ice hockey performance by Macdonald (2012). Macdonald (2012) uses ridge regression to estimate a player's contribution to his team's expected goals per 60 minutes. Ice hockey is a different sport than football, but in terms of the principles of expected goals models it is essentially very similar. In both sports a player tries to shoot and score a goal and a goalkeeper tries to prevent the ball or puck from going in.

In the years after this first notion of expected-goals, companies and people in the online community started building and improving their own expected-goals models (Green, 2012; Ijtsma, 2013; Trainor, 2013; Caley, 2013). Several important features, for example the body part and whether it is a set-piece or not, based on data on events in the game are introduced to the expected-goals models and different learning techniques are considered.

In recent years, so-called spatio-temporal data is getting more popular. Spatio-temporal

data is data which has information on the location of every player on the pitch instead of just on the location of the player who has the ball. Lucey et al. (2015) present a shot prediction model which uses spatio-temporal data to obtain strategic features such as defender proximity, speed of play and interaction of surrounding players. These features improve the measurement of the likelihood of each shot resulting in a goal. W. Spearman (2018) uses spatio-temporal data in order to quantify so-called off-ball scoring opportunities to enrich the expected goals values. In this way important opportunities during a match can be identified and analyzed.

2.2 Estimation of match outcome probabilities

Dixon and Coles (1997) use a technique based on a Poisson regression model in order to predict football scores. The goal of this research is to find inefficiencies in the football betting market and the models show that they have a positive return when this is used as the basis of a betting strategy. This methodology of predicting football games is different from the estimation of match outcome probabilities since it does not use in-game information, but this indicates that people started to gain interest in assessing outcome probabilities to football games. Rue and Salvesen (1997) and Langseth (2013) also look, among others, at statistical models to predict the outcome of football matches. Rue and Salvesen (1997) extend the methodology of Dixon and Coles (1997) by defining a random-walk model. Langseth (2013) compares different statistical models and betting strategies to obtain the best possible returns, which is according to Langseth (2013) the Gaussian motion model with an aggressive betting strategy.

Gurpinar-Morgan (2015) indicates a problem of using expected goals in evaluating single matches. In the article it is shown that the residual histogram of a single game (with an average number of shots) is very wide, which causes that the estimates can be inaccurate. It is noted that expected-goals are a very useful indicator of quality over a single match, but that because of the random nature of football one should be careful with the interpretation of these numbers.

Page (2015) indicates that the number of expected goals should not simply be added up but that also the variance should be taken into consideration. He indicates that expected-goals values are independent probabilities which can not be simply added up since this misses the story of variance. By simulation it is shown what the match probabilities for a specific game should be.

Eggels (2016) uses his expected goals models to explain match results using predictive analytics. In this paper the expected goals are converted to predicted number of goals, instead of converting into match probabilities. Eggels (2016) finds that the exact score is often not predicted. However, if a one goal difference is accepted the prediction accuracy is a lot higher. He concludes that the expected goals model is limited in evaluating match results, especially in tight games. Eggels (2016) also finds that the prediction results are not very different across leagues.

Van den Hoek (2019) uses his expected-goals model based on positional data to evaluate the expected outcome of a match. He uses a large number of simulations of a Poisson distribution

and then takes the mode to get the most likely number of goals scored for each team. The values for both teams then give the expected match outcome and is then compared to the actual match outcome to calculate the accuracy. Van den Hoek (2019) concludes that the outcome of a single match is often not in line with the outcome that would have been expected based on the scoring opportunities created by both teams.

2.3 Machine learning algorithms

Machine learning is among mathematical tools regarded as one of the most promising approaches. It is a method that builds analytical models based on the idea that a system can identify patterns and can make decisions with little human intervention. In this thesis some well-known machine learning algorithms will be used and hence a brief description of these algorithms is provided. The descriptions of the implementations are based on the corresponding description in the documentation of the library.

Logistic regression [1]

Logistic regression is one of the most basic and commonly used learning algorithms for binary classification problems (Navlani, 2018). Logistic regression uses a logit function to predict the probability of the occurrence of a binary event, which is the dependent variable. The estimation is done by maximum likelihood and the dependent variable follows a Bernoulli distribution.

XGBoost [2]

XGBoost is a gradient boosting framework which can be used for classification problems (Chen & Guestrin, 2016). Gradient boosting is a technique which produces predictions based on an ensemble of decision trees. The difference between XGBoost and random forests is that XGBoost builds new trees based on the errors of previous decision trees.

CatBoost [3]

CatBoost is a gradient boosting framework which differentiates itself from other gradient boosting frameworks by the implementation of ordered boosting and the possibility to process categorical features (Prokhorenkova, Gusev, Vorobev, Dorogush & Gulin, 2017). The ordered principle of boosting is inspired by online learning algorithms which get training samples sequentially in time. Prokhorenkova et al. (2017) describe how using this ordered boosting principle solves the so-called prediction shift which is present in other gradient boosting frameworks.

[1] From the sci-kit learn library: https://scikit-learn.org
[2] From the XGBoost library: https://xgboost.readthedocs.io/
[3] From the CatBoost library: https://catboost.ai/

Recurrent neural network [4]

The use of neural networks in machine learning has increased a lot over the last few years. Neural networks are based on biological neural networks and are general function approximations, which can be used for almost any machine learning problem with the goal of learning a complex mapping from the input to the output space. One of the possible neural network architectures is the recurrent neural network.

Elman (1990) introduced the recurrent neural network to try to find structure in time. A recurrent neural network processes sequences of data one element at a time, while it memorizes the preceding elements. "Recurrent" in this context means that the output of the previous time step is used as input for the next time step.

The most important part of a recurrent neural network is a layer which consists of memory cells. Hochreiter and Schmidhuber (1997) introduced the Long Short-Term Memory (LSTM) layer, which is currently the most popular memory cell in machine learning. An LSTM layer memorizes preceding elements and ensures that the signal is not lost as the sequence is processed. A common LSTM layer consists of a cell, an input gate, an output gate and a forget gate. The forget gate enables the LSTM layer to reset its own state.

Recurrent neural networks are often very powerful because it stores a lot of information about the past in an efficient way and the non-linear dynamics allows it to update the hidden state in complicated ways. How complicated the hidden state tries to compute the updating state is dependent on the number of neurons, which are little parts of a neural network which learn the weights based on the inputs and the desired outputs.

2.3.1 Ensembling

A problem of machine learning models is that they are likely to suffer variance in their predictions. This means that each time a model is fitted, it will give slightly different predictions which might be better or worse than expected. If the variance is lower, the model is less sensitive to the specifics of the training data. A well-known mathematical result shows that it is possible to reduce the variance without increasing bias:

$$Var\left(\frac{1}{N}\sum_{i}^{N} Z_i\right) = \frac{1}{N} Var(Z_i) \tag{2.1}$$

where $Z_1, \ldots, Z_N$ are i.i.d random variables. This gives that by making combinations of predictions, the overall accuracy of the predictions should be increasing. The most common ensemble algorithms are bagging, boosting and stacking. The strength of ensembling is shown by the fact that almost every winner of Kaggle competitions uses ensemble models (Vorhies, 2016).

[4] From the Keras library: https://keras.io/

2.4 Evaluation of classifiers

The goal of the model is to perform well on data that has not been seen in the training set. Different metrics are used in the literature to find out the performance of a specific model, which metric to use depends on the purpose of the study. For example, it is very unusual to use accuracy as a metric for the expected-goals values. An expected-goals value of 0.4 and 0.001 will be classified as a no-goal, but if it actually is a goal the error in the first case is a lot lower than the second one. Since this is not seen by accuracy, this is not a very good metric to use. Since probabilities are predicted in this thesis, it is very important that the models are well-calibrated. Calibration measures whether the predicted probabilities match the expected distribution of each class. The better calibrated a model is, the more reliable the forecast is out of sample. Described below are the metrics that are used in this paper to evaluate the performance of the expected-goals models and the match evaluations.

2.4.1 Log loss

Logarithmic loss (better known as log loss) measures the accuracy of a classifier by the probabilities obtained for every possible class instead of just the most likely class. Log loss is the cross entropy between the distribution of the predictions and the true labels. Cross entropy measures the unpredictability of the true distribution plus the extra unpredictability when a different distribution from the true distribution is assumed. The equation of the log loss is given by:

$$LL = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} log(p_{ij}) \tag{2.2}$$

which simplifies to the following if it is a binary classification problem:

$$LL = \frac{1}{N} \sum_{i=1}^{N} [y_i log(p_i) + (1 \quad y_i) log(1 \quad p_i)] \tag{2.3}$$

where N is the number of observations in the test set, M is the number of possible categories, y_{ij} is a binary indicator whether or not label j is the correct classification for observation i and p_{ij} is the model probability of label j to observation i. In Figure 2.1 the log loss functions of a binary classification problem are plotted. It can be seen that the lower the value for log loss, the better the prediction accuracy is. Because of the form of the logarithmic function, big errors in the predictions are relatively strongly penalized.

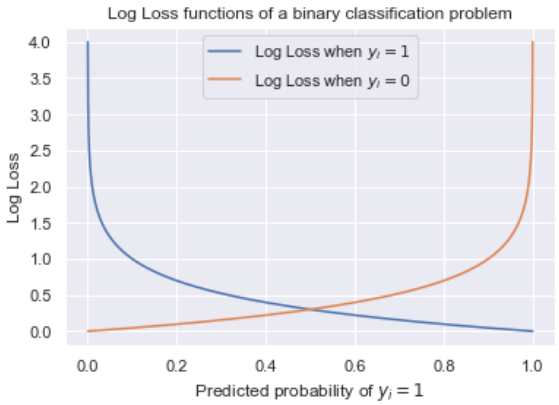


Figure 2.1: Graphical representation of log loss functions

2.4.2 Brier score

The Brier score is an alternative way of evaluating the accuracy of a probability forecast for problems with binary or categorical outcomes. In the case of more than two categories (for example home win, draw or away win), each possible category is treated as a binary outcome. The Brier score calculates the mean squared difference between the actual outcome of the event and the predicted probability for that specific outcome. This corresponds to the following equation:

$$BS = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} (p_{ij} \quad y_{ij})^2 \tag{2.4}$$

which simplifies to the following if it is a binary classification problem:

$$BS = \frac{1}{N} \sum_{i=1}^{N} [(p_i \quad y_i)^2 + ((1 \quad p_i) \quad (1 \quad y_i))^2] \tag{2.5}$$

where N is the number of observations in the test set, M is the number of possible categories, y_{ij} is a binary indicator whether or not label j is the correct classification for observation i and p_{ij} is the model probability of label j to observation i. In Figure 2.2 the Brier score functions of a binary classification problem are plotted. It can be seen that the lower the value for the Brier score, the better the prediction accuracy is. Because of the form of the graph, large mistakes are still penalized but not as strong as it was in the case of log loss.

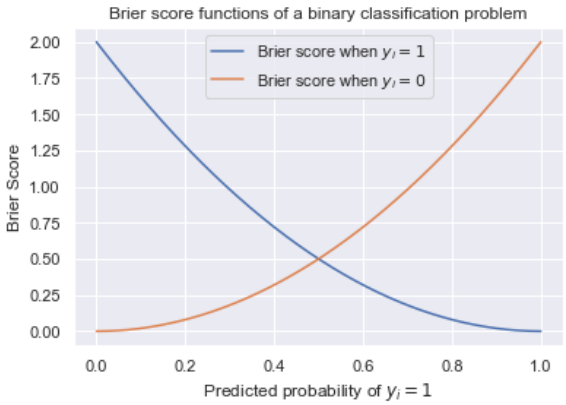


Figure 2.2: Graphical representation of Brier score functions

2.4.3 AUC-ROC

AUC-ROC is a performance measurement which tells how capable a model is in distinguishing between classes. AUC-ROC is a combination of AUC (Area Under the Curve) and ROC (Receiver Operating Characteristics). An ROC curve plots the True Positive Rate against the False Positive Rate at different classification thresholds. The AUC measures the area under the ROC curve and hence a higher AUC-ROC score indicates that the prediction accuracy is higher. An advantage of AUC is that it is invariant of the classification threshold that is chosen, it measures the quality of the model's predictions. In Figure 2.3 different examples of AUC-ROC curves are shown, with different scores in every graph. It shows that a higher value for AUC-ROC indicates a better prediction accuracy. The dotted line indicates the expectation of the AUC-ROC for random guessing, so it is even possible for the AUC-ROC to have a lower score than random guessing.

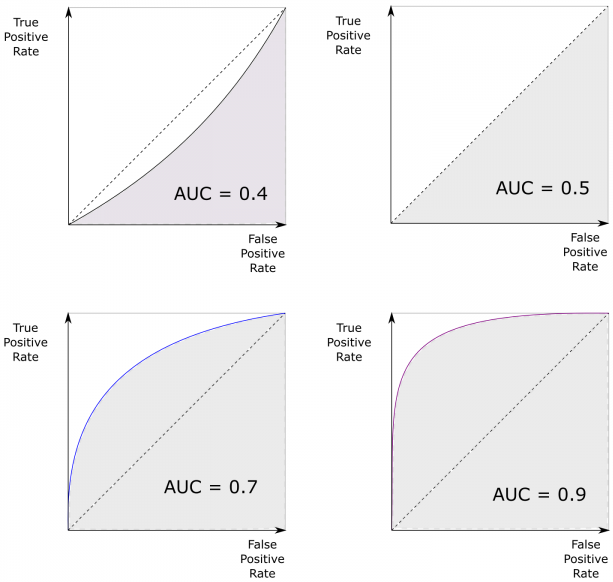


Figure 2.3: Graphical representation of different AUC-ROC curves. Source: Parkes (2018)

2.4.4 Calibration curve

An often used tool in machine learning to check how well calibrated a model is, is the calibration curve. An example of a calibration curve is shown in Figure 2.4. A calibration curve is a plot of the observed relative frequency on the vertical axis against the predicted probability on the horizontal axis, where the predicted probabilities are divided into a fixed number of bins. For example, for the values with a predicted probability of around 0.5, 50% of the sample should belong to the positive class. The dotted diagonal line from the bottom left to to top right of the plot in Figure 2.4 represents the perfectly calibrated model. The position of the curve relative to the dotted diagonal line helps interpreting the probabilities. If the curve is below the diagonal then the predicted probabilities are too large and vice versa. Following this, in Figure 2.4 the blue line is the best calibrated model.

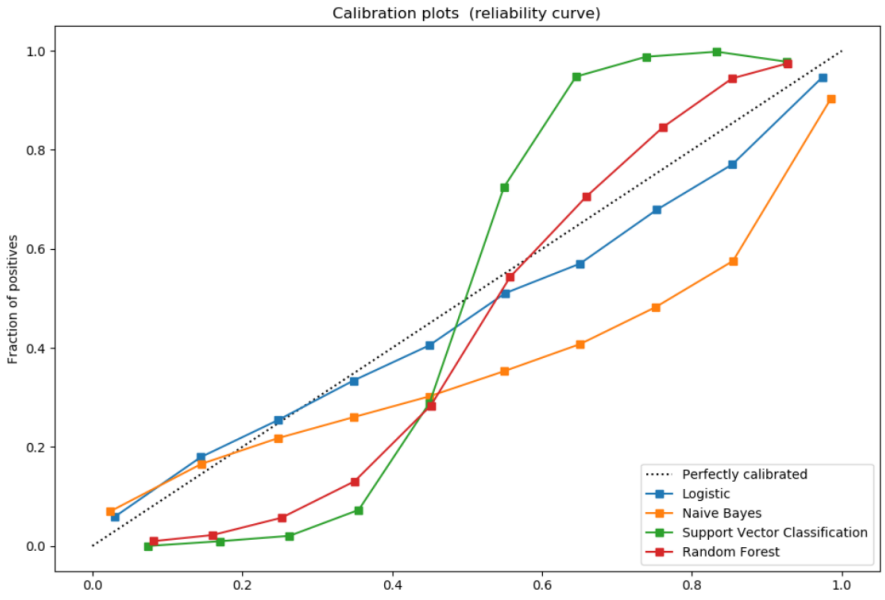


Figure 2.4: Example of calibration plots. Source: scikit-learn.org

Chapter 3

Data

In this section the data that is used in this thesis is presented. Since data from three different data sources are used, each of them will be presented separately. Next to this, the necessary data preparation steps will be discussed.

3.1 Data sources

3.1.1 Wyscout event data

The first data source that is used is so-called "event data" from a partner of SciSports, Wyscout. Wyscout manually annotates the events that happen during a game in over 250 competitions in 64 countries. In every match there are approximately 2,000 events, which for example could be a pass, a dribble or a shot. The Wyscout data is separated in two parts: match data and shot data.

Match data

The match data from the seasons that are contained in the data set for this thesis are presented in Table 3.1.

Competition (Country)	2016/2017	2017/2018	2018/2019*	# of matches
Premier League (England)	380	380	341	1,101
Bundesliga (Germany)	306	306	263	875
Serie A (Italy)	380	380	327	1,087
Ligue 1 (France)	380	380	324	1,084
La Liga (Spain)	380	380	321	1,081
Eredivisie (the Netherlands)	312	312	271	895
First Division A (Belgium)	0	269	239	508
Total	2,138	2,407	2,086	6,631

* Data up to April 2019

Table 3.1: Overview of competitions covered in the data set

In Table 3.1 it can be seen that in the data set used in this thesis over 6,500 matches are contained from 7 different leagues. The difference in the number of games per competition is caused by the different number of teams in a specific competition and the fact that in the 2018/2019 season a different number of games have been played in different competitions since the data in the data set is only up to April 2019.

In Table 3.2 a snapshot of the data obtained from Wyscout for a specific match can be found. For every of the approximately 2,000 events there are 73 variables which describes the context of the specific event. This can differ from the location of the event to whether the event was successful or not. All players, teams, matches and competitions have a specific identification number, in order to be able to for example distinguish two players with the same name. Next to this, every (sub-) type of event has its own identification number in order to easily distinguish different type of events. For example, type number 8 denotes a pass and type number 10 denotes a shot.

Match (id)	Period (id)	Time (ms)	Team (id)	Player (id)	Type (id)	Start loc* (x)	...	Accurate
2,723,110	1	2,636	5,068	99,479	8	51	...	True
2,723,110	1	3,995	5,068	163	8	36	...	True
2,723,110	1	5,366	5,068	129,523	8	31	...	True
...	...	...	...	...	...	...	...	...
2,723,110	2	2,916,673	5,081	134,715	1	53	...	False
2,723,110	2	2,922,624	5,068	163	3	50	...	False

* Every event has a normalized start (and end) x & y location.

Table 3.2: Snapshot of the Wyscout event data

To get a better overview of what is contained in the data, some descriptive statistics are given in Table 3.3. It can be seen that there are relatively a lot of passes in the data set and that on average 1 out of 8 shots results in a goal. Next to this, it can be seen that there relatively is a low amount of red cards. These descriptive statistics are useful in order to get some feeling on how often certain events happen in a game of football.

Type of event	# of occurrences	Average success rate
Shots	119,433	12.5%*
Passes	4,408,510	75.1%
Smart passes**	86,471	37.5%
Dangerous balls lost***	11,381	-
Fouls	140,509	-
Yellow cards	20,589	-
Red cards	543	-

* Percentage of shots resulting in a goal

** A "Smart pass" is a pass that leads the team in a good position to attack. The pass should be between 2-3 opposite players.[1]

*** A player loses possession and there is some dangerous counterattack for the opposite team. [1]

Table 3.3: Descriptive statistics of Wyscout data

Shot data

Next to the data set with all events for a specific number of games, there is a data set with only information about shooting opportunities. SciSports has enriched some of the shot data of Wyscout with more specific information about every shot. The corresponding data set contains 14 features which are more specific on the context of a certain shot. A data set with this information is useful for the training of expected-goals models. The format of the data set is presented in Table 3.4. In this data set there are a total of 146,540 shots which also contains shots from the matches that are in the match data.

Distance to goal (m)	Angle to goal ( )	Penalty	Head*	After cross**	After dribble**	...	Goal
30.34	37.24	False	False	False	False	...	False
10.85	14.52	False	False	True	False	...	False
28.32	27.14	False	False	False	False	...	False
...	...	...	...	...	...	...	...
6.86	23.35	False	False	False	False	...	False

* Indicates whether the shot was a headed shot or not

** Indicates whether the shot was after a shot/dribble

Table 3.4: Snapshot of the shot data from SciSports

[1] https://footballdata.wyscout.com/events-manual

3.1.2 FIFA 19 data

In order to measure the quality of the players, data from the football simulation video game FIFA 19 is used. The data set is scraped data from an online database[2] of FIFA 19 data and made publicly available for people interested in football analytics. A sample of the data is given in Table 3.5, which shows the FIFA 19 statistics of a few of the best football players in the world. The FIFA 19 data set consists of 18,207 players from 652 different clubs with information about the estimated quality for different aspects in the game of football.

Name	Age	Nation	Club	Overall	...	Finishing	Marking
L. Messi	31	Argentina	Barcelona	94	...	95	33
C. Ronaldo	33	Portugal	Juventus	94	...	94	28
Neymar Jr.	26	Brazil	PSG	92	...	87	27
...	...	...	...	...	...	...	...

Table 3.5: Snapshot of the FIFA 19 data

3.1.3 Bookmaker data

In order to get an intuition of the pre-game expectation of the result, it is interesting to look at the odds that the bookmakers offer. Bookmakers offer consumers the chance to bet on football games by providing odds. For example, if the odds of Ajax winning at home against PSV are 1.72 and you bet €1 on Ajax, then you receive €1.72 if they win and nothing if they do not win the game. A widely-used and simple method of obtaining outcome probabilities from bookmaker odds (Strumbelj, 2014) is given by:

$$p_h = \frac{{o_h}^1}{{o_h}^1 + {o_d}^1 + {o_a}^1} \tag{3.1}$$

where p_h is the winning probability of the home team, and o_h, o_d and o_a are the odds for the home team, a draw and the away team. This conversion is necessary since the bookmakers do not offer fair odds, they take an edge on the odds in order to be profitable.

The bookmaker data for the competitions of the Wyscout data are obtained from an online source[3] and manually merged into a CSV file. A sample of the bookmaker data set is given in Table 3.6.

[2] https://so fa.com/

[3] http://football-data.co.uk/

Date	Home team	Away team	Goals home	Goals away	...	Bwin H*	Bwin D*	Bwin A*
28-7-17	Antwerp	Anderlecht	0	0	...	5.00	4.00	1.67
20-10-18	Genk	Eupen	2	1	...	1.18	7.25	15.00
...	...	...	...	...	...	...	...	...
6-5-17	Barcelona	Villarreal	4	1	...	1.16	7.75	16.50
7-4-19	Valladolid	Sevilla	0	2	...	3.30	3.50	2.15

* The home odds, draw odds and away odds of Bwin, a popular bookmaker.[4]

Table 3.6: Snapshot of the bookmaker data

3.2 Data preparation

3.2.1 Data merging

In order to be able to combine the data sets of FIFA 19 and the Wyscout shot data, the mutual information that is contained in both data sources should be leading in the merging process. For every shot in the data set of Wyscout, the name of the player that took the shot is compared to the names of the players of the FIFA 19 data set. This comparison of the player names of Wyscout and FIFA 19 is done by approximate string matching, in the field better known as fuzzy string searching. Fuzzy string searching uses a distance function to calculate the differences between sequences, in this case the names of the players. In this thesis the Levenshtein distance (Levenshtein, 1966) is used to calculate this difference in the names, which is a simple but widely used method in string matching. The distance measure boils down to how many insertions, deletions or substitutions there are necessary to transform the first name to the other name. For example, the Levenshtein distance between Messi and Neymar is 5 since:

1. Messi *!* Nessi (substitution of "M" for "N")
2. Nessi *!* Neysi (substitution of "s" for "y")
3. Neysi *!* Neymi (substitution of "s" for "m")
4. Neymi *!* Neyma (substitution of "i" for "a")
5. Neyma *!* Neymar (insertion of "r")

This Levenshtein distance is calculated for every possible pair of players in the FIFA 19 data set and then ranked from the lowest value to the highest value. The match with the lowest Levenshtein distance is most likely the player that indicates the same player, but some extra checks are necessary to be made. It is possible that a player that took a shot in the Wyscout

[4] https://bwin.com

data is not in the FIFA 19 data, for example because the player has retired from football. In the FIFA 19 data set the age of the player is presented, while in the Wyscout data the birth date of the player with a specific identification number are known. If the player with the lowest Levenshtein distance also has the same age as the player in the Wyscout data, then the matching between the players is accepted. If this match is not accepted, then it is assumed that the player is not in the FIFA 19 data set and that no extra player-specific information can be gathered for this shot. In this case, the average value for a specific attribute will be used as value for this player.

3.2.2 Expected-goals model

Traditional expected-goals models only use information about the shot, while the expected-goals model in this thesis enriches the shot-specific information with player-specific information. In order to do this, the goal is to add the player-specific data of FIFA 19 to the shot data obtained from Wyscout. In this way, for every shot the likelihood of the shot resulting in a goal can also depend on the quality of the player who takes the shot. This is very important since expected-goals models aim to assess player performance over time instead of actually calculating the likelihood of a shot resulting in a goal.

Feature engineering

After all players between the Wyscout data set and the FIFA 19 data set are matched, the player-specific information can be added to every shot. Since the characteristics of every shot are very different, some types of shots are distinguished:

Headers

Long-range shots

Other shots

For every shot in the Wyscout data set there is one FIFA 19 rating stored, which is the finishing, heading accuracy or long shots statistic for the specific player. This is done since some players, for example defenders, are not very good in long shots but very good in heading. This differentiation of the statistics should be able to improve the model even further than just using for example the finishing statistic. In Table 3.7 all variables that are used for the expected-goals model and a short description are given.

Parameter	Description	Parameter	Description
After corner	Shot follows from a corner	Distance to goal line	Nearest distance to the goal line
After cross	Shot follows from a cross	Head	Headed shot or not
After dribble	Shot follows from a dribble	Free kick	Free kick shot or not
After duel	Shot follows from a duel	Penalty	Penalty kick or not
After pass	Shot follows from a pass	Rebound	Rebound shot or not
Angle of sight	Viewing angle of player relative to the goal	Left foot	Shot with left foot or not
Angle to goal	Angle of shot position and center of the goal	FIFA 19 Rating	Player specific rating for finishing, heading or long shots
Distance to goal	Distance to center of the goal		

Table 3.7: Parameter description of expected goals model

3.2.3 Estimation of match outcome probabilities

The goal of the model is to improve the estimation of match outcome probabilities by adding match statistics. In order to do this, the calculated expected-goals values for every match have to be merged with the other match statistics in the game. Since every match in Wyscout has a specific identification number, the expected-goals values can easily be combined with the match statistics.

The match statistics can be calculated by summing up every specific event that a team performs during a game for most statistics. A statistic which can not directly be obtained by the data is the percentage of possession for the home team. It is only necessary to obtain the percentage of possession for one of the teams, since the two possession percentages would be perfectly collinear. A method for an approximation of the possession percentage of the home team is given by Aberdeene (2012):

$$\text{possession}_{\text{home team}} = \frac{\#\text{ passes home team}}{\#\text{ total passes in game}} \tag{3.2}$$

Next to the match statistics that come from the Wyscout data, it is also interesting to look at the bookmaker odds for every specific game in the data set. The goal of a bookmaker is to make as much profit as possible and not to predict football games as precise as possible. Hence, after determining the starting odds, the odds of bookmakers change according to how people bet on the games. For example, if a lot of people bet on a home win, this odd will decrease and the odds of a draw and an away win will increase (Holland, 2018). This makes that bookmakers do not need to have perfect models, but just use the so-called “wisdom of the crowd” in order to

get information about the pre-game outcome probabilities of a football game. Because of this it is very interesting to use this information in the estimation of the match outcome probabilities, because it gives implicit information on the current strength of a team. If a team is in a very good form, the chances might be converted into goals more easily.

In order to merge the betting data from Bwin and the data from Wyscout, the team names are merged using the same fuzzy string searching as described in section 3.2.1. After merging all team names from Bwin and Wyscout, the betting odds for the home team, a draw and the away team are added to the Wyscout and expected-goals data. All variables that are now obtained and used in the model for the estimation of the match outcome probabilities can be found in Table 3.8.

Parameter	Description	Parameter	Description
xG home	Expected-goals values of home team	xG away	Expected-goals values of away team
# shots home team	Number of shots taken by home team	# shots away team	Number of shots taken by away team
# fouls home team	Number of fouls made by home team	# fouls away team	Number of fouls made by away team
# yellow cards home team	Number of yellow cards of home team	# yellow cards away team	Number of yellow cards of away team
# red cards home team	Number of red cards of home team	# red cards away team	Number of red cards of away team
# smart passes home team	*	# smart passes away team	*
# dangerous balls lost home team	**	# dangerous balls lost away team	**
Bwin home	Betting odds home team	Bwin draw	Betting odds draw
Bwin away	Betting odds away team	% possession home team	Percentage of time the home team has the ball

* A "Smart pass" is a pass that leads the team in a good position to attack. The pass should be between 2-3 opposite players.

** A player loses possession and there is some dangerous counterattack for the opposite team.

Table 3.8: Parameter description of model for estimation of match outcome probabilities

Chapter 4

Methodology

4.1 Expected-goals model

In this section the methodology of the expected-goals model will be explained. In order to observe the effect of adding player-specific information to the expected-goals model, two different models are estimated: a baseline model that only considers the shot characteristics and a model that also considers player-specific information in addition to the shot characteristics.

Feature set splitting

In machine learning it is common practice to split the data into a training set, a validation set and a test set. The goal in machine learning is to get a model that performs well on out-of-sample data and if a model is trained on all the data there is the danger of overfitting (Chollet, 2017). If a model is overfitted, it performs very good in sample but much worse out of sample. The data is splitted by taking the shots of the seasons of 2016/2017 and 2017/2018 as a set for training (90,000 shots) and validation (20,000 shots) and the season 2018/2019 as test set (50,000 shots). Since the data set only contains data for FIFA 19 and not data on previous editions of FIFA, note that shots taken before the 1st of September 2018 contain future information about how good a player becomes in the next year. However, since the season 2018/2019 is used as test set in order of correctness of the methodology this is not a problem.

Feature standardization

Next to the splitting of the data, normalization or standardization of the data has proved to have a positive influence on the performance of classification problems (KumarSingh, Verma & S. Thoke, 2015). In this thesis, standardization will be performed on the data since standardization is less sensitive to outliers compared to normalization. Standardization is the rescaling of features such that they have the properties of a Gaussian distribution with mean 0 and standard

deviation 1. The following formula is used to standardize the data:

$$z = \frac{x}{\quad} \tag{4.1}$$

where is the mean of the feature in the training set, is the standard deviation of the feature in the training set, x is the original value for the feature and z is the scaled value for the feature. The mean and standard deviation of the training set is used to standardize the data, since the validation set and the test set have to be treated as "unseen data" and hence can also not be used in the standardization process.

4.1.1 Model learning

The choice of the learning algorithms for the expected-goals models are based on previous research by Van den Hoek (2019) and Eggels (2016). Based on their findings on the most effective machine learning algorithms and the current trends in machine learning, the following algorithms (explained in section 2.3) are used:

Logistic Regression

XGBoost

CatBoost

Next to these learning algorithms, as mentioned in section 2.3.1, also an ensemble of these learning algorithms is considered.

4.1.2 Tuning hyperparameters

Hyperparameters are the settings of machine learning algorithms, which helps the algorithm to estimate the parameters. It is common practice in machine learning to tune these hyperparameters, since the optimal settings differ per problem and per algorithm. Two often used techniques for optimizing the hyperparameters are grid search or random search, but in this thesis the Bayesian optimization approach BayesSearchCV[1] is used. In contrast with grid search, not all parameter values are tried, but a number of parameter settings is sampled from the specified search spaces for a given number of iterations. This makes that BayesSearchCV is less time consuming since it spends less time on ineffective combinations of hyperparameters. The Bayesian optimization is applied in order to find the settings that give the best results over k folds in terms of AUC-ROC. The number of folds is set to the default of 3 and the number of iterations is set to 100. The results of this BayesSearchCV can be found in Table 4.1.

[1] https://scikit-optimize.github.io/#skopt.BayesSearchCV

Learning algorithm	Hyperparameter	Search space	Optimal settings
Logistic Regression			
	C	(0.001, 100)	42.0
	Penalty	[l1, l2]	l2
XGBoost			
	Max depth	(1, 8)	2
	Learning rate	(0.01, 0.20)	0.10
	Subsample	(0.2, 1.0)	0.84
	Colsample by tree	(0.2, 1.0)	0.77
	Min child weight	(2, 50)	42
CatBoost			
	Max depth	(1, 8)	2
	Learning rate	(0.01, 0.20)	0.06

Table 4.1: Optimal results of hyperparameter tuning

Note that in Table 4.1 the number of estimators used for XGBoost and Catboost are determined by an early stopping method. The number of estimators is set high enough and the algorithm decides when to stop training. If in the last 20 estimators the log loss of the validation set has not improved, the training of the model is stopped.

There are a lot of parameters that can be set in the different machine learning models, the choice of which parameters to optimize is based on previous research by Jain (2016), Peretz (2018) and Qiao (2019).

4.2 Estimation of match outcome probabilities

In this section the methodology of the estimation of the match outcome probabilities is presented. In order to observe the effect of adding more advanced in-game statistics, some baseline models are estimated next to the models with more advanced information. Next to this, the choice of learning algorithms for every model is explained.

Feature set splitting

As explained in section 4.1, in machine learning it is common practice to split the data into a training set, a validation set and a test set. The data for this problem is split by taking the matches of the seasons of 2016/2017 and 2017/2018 as a set for training (4,000 games) and validation (500 games) and the season 2018/2019 as test set (2,000 games). Next to the splitting of the data, the same standardization procedure as described in section 4.1 will be used.

4.2.1 Baseline models

The goal of the model is to estimate the probabilities of each possible match outcome. Since the "true" match probabilities are not observed (only the actual outcome), some baseline models are considered. These baseline models are based on historic information about football games or on the actual information from the game.

Baseline 1: Prior probabilities

One of the most simple and commonly used baseline models (Brownlee, 2018) is the prior distribution over the possible outcomes. Since in this thesis the model is trained on data of the matches of the seasons of 2016/2017 and 2017/2018, the outcome percentages of these matches are taken. This gives the following match probabilities:

Home win: 44%; Draw: 26%; Away win: 30%

These percentages can be used as a very simple match result estimation, where no team and match statistics are taken into consideration. It can be observed that home wins occur more often than away wins, this can be attributed to the existence of home advantage in football. Pollard et al. (2004) tries to find the possible causes of the home field advantage. He finds that for example factors like crowd effects, travel effects and psychological factors can be the explanation of the home field advantage in football.

Baseline 2: Poisson distribution

Estimation of match outcome probabilities with the help of expected-goals values can be very straightforward: the team with the highest amount of expected goals should have won the game. However, this tells nothing about the difference in expected-goals values. The difference could be 0.01 xG or 5.00 xG, but this is in this very simple method not incorporated.

In order to improve on this very simple method, some distributional form can be taken into account. Dixon and Coles (1997) introduce a Poisson distribution to obtain the probabilities of match results. The Poisson distribution is a discrete probability distribution that describes the number of events (goals) within a specific time period (90 minutes). From this, Dixon and Coles (1997) give the following expression of the home team i scoring x times and the away team j scoring y times:

$$Pr(X_{i;j} = x; Y_{i;j} = y) = \quad ; \ (x;y)\frac{^{x}exp(\quad)}{x!}\frac{^{y}exp(\quad)}{y!} \tag{4.2}$$

where $X_{i;j}$ is the number of goals scored by home team i, $Y_{i;j}$ is the number of goals scored by away team j, denotes the average scoring rate of home team i, denotes the average scoring rate of away team j and $\ ; \ (x;y)$ denotes the correction for low-scoring matches.

Dixon and Coles (1997) find that the model is underestimating the probabilities of low

scoring matches. In order to adjust for this, $\tau_{\lambda,\mu}(x,y)$ is implemented in the following way:

$$\tau_{\lambda,\mu}(x,y) = \begin{cases} 1 & \text{if } x = y = 0 \\ 1 + & \text{if } x = 0, y = 1 \\ 1 + & \text{if } x = 1, y = 0 \\ 1 & \text{if } x = y = 1 \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

where controls the strength of the correction. It is possible to set an unique for every competition, but in this thesis a single is chosen. A single is chosen since it is assumed there are no significant differences between the leagues considered in this thesis. Lindstrom (2014) finds that the optimal value for is given by 0.13 and this value is also found by Sheehan (2017). Based on this previous research, in this thesis the value for is set to 0.13.

The goal of Dixon and Coles (1997) is to estimate the match outcome probabilities of future games in order to be able beat the bookmakers. The average scoring rate parameters and are therefore by Dixon and Coles (1997) based on an approximation of the attacking and defending qualities of both teams. However, since in this paper the goal is to estimate the match probabilities based on in-game statistics the expected-goals values can be used as a value of & .

By using the sum of the expected-goals values for both teams for & , the probability for every possible outcome can be calculated. Based on this, the estimated match result probabilities are calculated using:

$$Pr(\text{Home win}) = Pr(X_{i,j} > Y_{i,j}) = \sum_{\substack{x=0 \\ x>y}}^{N} \sum_{y=0}^{N} \tau_{\lambda,\mu}(x,y) \frac{\lambda^x \exp(-\lambda)}{x!} \frac{\mu^y \exp(-\mu)}{y!} \tag{4.4}$$

$$Pr(\text{Draw}) = Pr(X_{i,j} = Y_{i,j}) = \sum_{\substack{x=0 \\ x=y}}^{N} \sum_{y=0}^{N} \tau_{\lambda,\mu}(x,y) \frac{\lambda^x \exp(-\lambda)}{x!} \frac{\mu^y \exp(-\mu)}{y!} \tag{4.5}$$

$$Pr(\text{Away win}) = Pr(X_{i,j} = Y_{i,j}) = \sum_{\substack{x=0 \\ x<y}}^{N} \sum_{y=0}^{N} \tau_{\lambda,\mu}(x,y) \frac{\lambda^x \exp(-\lambda)}{x!} \frac{\mu^y \exp(-\mu)}{y!} \tag{4.6}$$

where N is the maximum number of goals for which the probability of occurring is calculated. N is set to 10 since the probability of a team scoring more than 10 goals is almost always extremely low given its expected-goals values.

Baseline 3: Betting odds

As indicated before, betting companies use the betting behavior of users to adjust the betting odds. This makes it interesting to use these outcome probabilities as a baseline model.

Over a long period of time, people have been trying to beat the bookmakers (Dixon and Coles, 1997; Langseth, 2013). As noted before in section 3.1.3, the bookmakers take an edge in

order to be sure to be profitable. The bookmaker edge can be implicitly calculated using the following equation:

$$\text{bookmaker edge} = (o_h{}^1 + o_d{}^1 + o_a{}^1) \quad 1 \tag{4.7}$$

where o_h, o_d and o_a are the odds for the home team, a draw and the away team. This edge differs per game and per bookmaker, but on average on the training set of the bookmaker odds the edge is 5.3%. From this it can be concluded that to be profitable in the long run, a better must be at least 5.3% more precise in its match predictions than the bookmaker.

From the observations stated above it can be concluded that the implied probabilities as introduced in section 3.1.3 could be a good baseline for predicting match results. This method takes the strengths of the teams and the "wisdom of the crowd" into consideration, but no in-game statistics of the match of which the match outcome probabilities are estimated.

4.2.2 Model 1: Summed expected-goals values & match statistics

In the first model, the goal is to answer the sub-question whether adding match statistics to the expected-goals values improves the estimation of outcome probabilities of football matches. The base is the same as in Baseline 2: the summed expected-goals values for both teams. Next to these summed expected-goals values, the variables described in Table 3.8 are used as explanatory variables.

Model learning

The following learning algorithms (explained in section 2.3) are chosen to estimate the outcome probabilities:

Logistic Regression

XGBoost

CatBoost

Next to these learning algorithms, it is also interesting to consider an ensemble of these learning algorithms. The ensemble used is the average of the different estimations of the learning algorithms with equal weight for each learning algorithm.

4.2.3 Model 2: Time-binned expected-goals values & match statistics

In this model, the goal is to consider the time effect of the expected-goals values. The method that is used in this model is introducing time-bins of a length of 5 minutes in which the amount of expected-goals is summed up. The time length of 5 minutes is arbitrarily chosen in order to try to have a good balance between the number of feature vectors and the number of examples per time-bin. There are different time-bins for the added time for both the first and the second

half and since the added time differs per match and per half, these time bins do not have a fixed time length.

Model learning

The biggest advantage of the time-binned approach, in comparison to a method with the raw vectors of expected-goals values, is that traditional machine learning algorithms are able to cope with the time-bins, since the algorithms observes every time-bin as a separate variable. The actual "temporal aspect" is not understood by the machine learning algorithm, but by preprocessing the data in the way described above the temporal aspect can still be exploited. Because of this, the same learning algorithms as described in Model 1 will be used to estimate the outcome probabilities:

Logistic Regression

XGBoost

Catboost

Next to the learning algorithms described above, it is interesting to consider an ensemble of the learning algorithms presented above. The ensemble is the average of the different estimations of the learning algorithms with equal weight for each learning algorithm.

4.2.4 Model 3: Vectors of expected-goals values & match statistics

Model overview

In this model, the goal is to exploit the time effect of the expected-goals values using an alternative method where the raw vectors of expected-goals values of both teams are used. The biggest challenge of using this method is the difference in the lengths of the vectors: not every team gets the same amount of chances every game. Next to this, the learning algorithm must be able to handle a multidimensional vector of values, since there is a vector for both the expected-goals values for the home team and the away team and the corresponding time in the match.

In this model the recurrent neural network (RNN), which is explained in section 2.3, is used. Normally this sort of neural networks are used for predicting stock prices, but it may also be able to memorize sequences of expected-goals values and be able to discover some temporal aspect. An LSTM layer is used in this model as a memory cell, since in section 2.3 it is explained that this is the most common memory cell in recurrent neural networks.

Next to the information that goes in the LSTM layer, also the match statistics as used in Model 1 and Model 2 are interesting to improve the estimation of the match outcome probabilities. However, since these variables are not time dependent in this format it is not beneficial to use an LSTM layer for this information. Therefore, for this information dense layers are better to use. Dense layers are hidden layers where every node is connected to every other node in the

next layer. The output of the LSTM layer can be used as input for a dense layer, which makes that both information inputs can be combined in the dense layers.

Overfitting is in neural networks a problematic issue, which is the risk of focusing too much on the training data instead of creating a general model. Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov (2014) show that by randomly dropping units and their connections from the neural network during training reduces the probability of overfitting. It is common practice to do this after every layer in the neural network.

Model 3 combines different types of layers and methods in order to get the best possible solution to this problem. In Figure 4.1 a top view of the model with the different inputs and output can be seen.

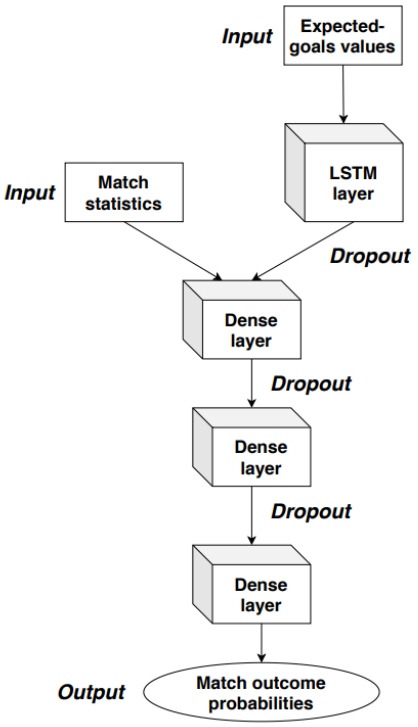


Figure 4.1: Graphical representation of Model 3

Padding

As can be seen in Figure 4.1, the expected-goals values serve as input for the LSTM layer. However, as noted before, the length of the different expected-goals vectors are not equal to each other and can hence not be used as input in this format. A method to get equal vector lengths is padding, which is often used in neural networks. There are two different sorts of padding: pre-padding and post-padding. Dwarampudi and V Subba Reddy (2019) find that pre-padding performs better than post-padding with LSTM layers, hence in this model pre-padding is used. Pre-padding takes the highest vector length of the training set and uses this as the vector length for every vector of expected-goals values. If the vector length of a vector is smaller than the highest vector length, then the rest of the vector is filled by adding zeros in

front of the original vector of expected goals vector. A simple example of pre-padding is given in Figure 4.2.

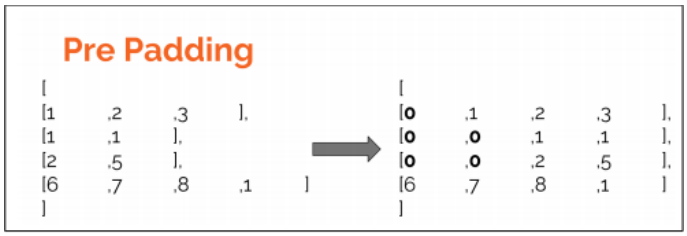


Figure 4.2: Graphical example of pre-padding. Source: Dwarampudi and V Subba Reddy (2019)

In the training set the highest number of chances for a team is 39, so the padding will be based on a maximum vector length of 39. Note that this is based on the highest amount of chances in the training set, so it might be the case that for a game in the test set there is a team in a game with more than 39 chances. This is fixed by a method called "truncating", which means that values of the vector are deleted in order to get the target vector length. In this model it is chosen to use pre-truncation, so if the vector has a length of more than 39 than the first chances will be deleted up until the moment that the vector length is 39. This means that the total amount of expected-goals values of that team would decrease, but because 39 is still a very large number of chances it is not very likely that it would change the estimated match outcome probabilities a lot.

Model con guration

For every game there are two vectors of expected-goals values: a vector of the expected-goals values of the home team and a vector of the expected-goals values of the away team. Since both vectors have to be used as input for all games in the training set, the input consists of a three-dimensional array with a size 5,221 (size of the training set) x 39 (variables) x 4 (expected-goals values of the home team and away team and the corresponding time in the match). The LSTM layer is able to directly use this three-dimensional array as input.

As explained in section 2.3, the number of neurons in a layer determines the amount of complexity a layer takes into account during the computation. The higher the amount of neurons per layer, the higher the amount of parameters that are estimated in the model. It is common practice to try different combinations of the number of neurons and observe which combinations give the best performance on the validation set. From this it is obtained that the LSTM layer consists of 32 neurons, the first dense layer of 64 neurons, the second dense layer of 32 neurons and the third and last dense layer of 16 neurons. The rectified linear unit (ReLU) activation function is used in the dense layers and in the LSTM layer, the softmax activation functions is used in the output layer. All in all, this gives that there are 10,595 trainable parameters in this recurrent neural network.

In the training of neural networks it is very difficult to choose the amount of training epochs to use. Too little epochs may result into a model which could have gotten better results and too much epochs might, despite the dropout, result into overfitting. In order to get the right amount of epochs, a method called early stopping is used. The early stopping method uses the validation set and as long as the loss of the validation set is decreasing it goes to a next epoch, up until the point that the validation loss is no longer decreasing. This enables the model to have a very high number of maximum number of epochs (in this model 500), since as long as the validation loss is decreasing (if time permits) it is beneficial for the performance of the model. It is possible that the validation in a epoch is a little higher than the last validation loss, but that the model is not yet at its optimal settings. Because of this, there is a patience parameter in the early stopping algorithm. In this model it is set to 30, so if the validation loss has not decreased in the last 30 epochs, there will not be a next epoch.

4.2.5 Tuning hyperparameters

As indicated in section 4.1.2, tuning the hyperparameters helps the model in estimating the parameters. Following the same procedure as in section 4.1.2, the optimal results of the hyperparameter tuning are given in Table 4.2.

Learning algorithm	Hyperparameter	Search space	Model 1	Model 2	Model 3
Logistic Regression					
	C	(0.001, 100)	44.1	67.3	
	Penalty	[l1, l2]	l1	l1	
XGBoost					
	Max depth	(2, 8)	3	2	
	Learning rate	(0.01, 0.20)	0.06	0.03	
	Subsample	(0.2, 1.0)	0.67	0.53	
	Min child weight	(1, 50)	12	2	
	Colsample by tree	(0.2, 1.0)	0.78	0.51	
Catboost					
	Max depth	(2, 10)	3	3	
	Learning rate	(0.01, 0.20)	0.07	0.06	
RNN					
	Batch size	(10, 50)			30
	Dropout	(0.1, 0.6)			0.5

Table 4.2: Optimal results of hyperparameter tuning of Model 1, Model 2 and Model 3

Chapter 5

Results

In this section the results of the expected-goals model and the estimation of the match outcome probabilities are presented. The models are evaluated using the different evaluation criteria introduced in section 2.4. Next to this, a feature importance analysis is performed to better understand which variables are important in the decision process of the learner.

5.1 Expected-goals model

5.1.1 Evaluation

In section 2.4 some different evaluation metrics have been presented, which are now used to evaluate the results of the different models. Before considering the evaluation metrics, it is first important to look how well calibrated the models are. As discussed before in section 2.4.4, it is very important for predictions of probabilities to be well calibrated. In Figure A.1 the calibration curves for the different models are presented. It can be seen that the models are in general very well calibrated, but that the XGBoost model slilghtly underestimates high probabilities. However, because of the small sample size of chances with such high probabilities this is not likely to be a problem.

Since the models are well calibrated, it is interesting to compare the actual performance of the different models. In Table 5.1 the results of the baseline model of the expected-goals are presented. It can be seen that the CatBoost model gives the best results in terms of these evaluation criteria. It is also interesting to see the comparison with similar models of Eggels (2016) and Van den Hoek (2019), since the baseline model outperforms these models in the evaluation criteria given in those papers. Eggels (2016) and Van den Hoek (2019) do use different data sets which makes it difficult to compare, but it gives at least an indication that the predictions of the baseline model are of good quality.

Learning algorithm	Log loss	AUC-ROC	Brier score
Logistic Regression	0.2809	0.7971	0.0803
XGBoost	0.2801	0.7983	**0.0802**
CatBoost	**0.2795**	**0.8000**	**0.0802**
Eggels (2016)	-	0.7850	-
Van den Hoek (2019)	-	0.7955	0.0818

Table 5.1: Evaluation criteria result of baseline model of expected-goals

By using the best possible baseline model, which is the CatBoost model, it is possible to compare this with Model 2, where the FIFA 19 ratings are included. The results on the same evaluation criteria are presented in Table 5.2. It can be seen that CatBoost is again the best model on the used evaluation criteria and that it is an improvement relative to the best baseline model. Note that the results of the CatBoost and the XGBoost are very close to each other. This is likely to be caused by the fact that no categorical variables are used in the model which is the part of modelling where CatBoost most distinguishes itself from XGBoost.

Learning algorithm	Log loss	AUC-ROC	Brier score
Best baseline model	0.2795	0.8000	0.0802
Logistic Regression	0.2797	0.8003	0.0800
XGBoost	0.2788	0.8021	**0.0799**
CatBoost	**0.2787**	**0.8022**	**0.0799**

Table 5.2: Evaluation criteria result of expected-goals model with FIFA 19 ratings included

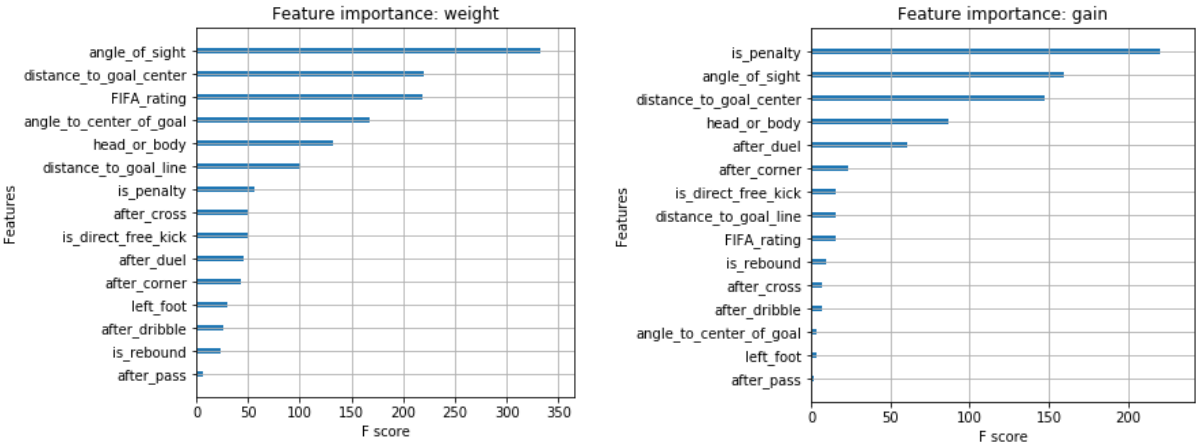
5.1.2 Feature analysis

Machine learning algorithms are often very difficult to interpret because of the complex patterns that are used. The models are often considered as a black box where the user has little knowledge about which factors make the results more accurate. However, in order to understand the problem it is useful to understand how certain results are built up.

One of the benefits of using a learning algorithm like XGBoost is that it is tree-based, which means that it is possible to visualize the importance of various features. The two metrics that are used in this thesis to show the contribution of the variables to the model are “gain” and “weight”. The gain shows the relative contribution of the variable to the model, which implies that a higher number implies it is more important for generating a prediction. The weight shows the relative number of times a variable occurs in the trees of the model. If a variable for example has a large gain but a low weight, it is likely to be a good variable to include. The variable might not influence the results of the model a lot since it does not occur a lot in the trees, but it is a very good explanatory variable for specific data points. An example of this in football is a penalty, which does not occur very often but when it occurs it has a big impact of

the likelihood of the shot resulting in a goal.

In Figure 5.1 the feature importances of the XGBoost model are presented. It can be seen that the angle of sight is a very important feature in both weight and gain, while a penalty, as explained above, has a relatively low weight but a high gain. It can be seen that the FIFA 19 rating has a high weight but a relatively low gain. This shows that the information on the FIFA 19 rating is often used in the trees, but that it does not have a big influence on the predictions. Furthermore, the distance to the center of the goal is very important in terms of weight and gain. This makes sense since every shot has a distance to the goal and in general the greater the distance to the goal the less likely it is that the shot results in a goal.



(a) Weight metric

(b) Gain metric

Figure 5.1: Feature importance metrics of XGBoost model for expected-goals

5.2 Estimation of match outcome probabilities

5.2.1 Evaluation

As discussed before in section 2.4.4, the calibration of the models is very important. Different from the calibration curve from section 5.1.1, the target variable of the estimation of the match outcome probabilities consists of three classes instead of the two classes in the expected-goals model. This makes that for every class there is a separate calibration curve. In Figure A.2 the calibration curves for the classes of a home win, a draw and an away win are presented. In general it can be seen that the models are very well calibrated, except for the higher predicted probabilities of draws. However, since the amount of predicted probabilities of draws above 40% is very small, this is not likely to be a problem.

Since all models, as well as the baseline models, are well calibrated, the results of Model 1 can be compared to the baseline models. In Table 5.3 the results of the models evaluated on the evaluation criteria of section 2.4 are presented. It can be seen that Baseline 2, which is based on Dixon and Coles (1997), outperforms the other baseline models. This is as expected since Baseline 2 uses the expected-goals values while the other baselines only use pre-game

expectations. Baseline 3 outperforms Baseline 1, which shows that the bookmakers are more accurate in estimating outcome probabilities than using the average outcomes of a football match.

When considering the results of Model 1 with the different learning algorithms, it can be seen that every learning algorithm outperforms the baseline models. The CatBoost learning algorithm produces the best results, but the ensemble of the three learning algorithms performs even better.

Learning algorithm	Log loss	AUC-ROC	Brier score
Baseline 1	0.6337	0.5000	0.6447
Baseline 2	0.5316	0.7460	0.5288
Baseline 3	0.5669	0.7033	0.5661
Logistic Regression	0.4757	0.8067	0.4713
XGBoost	0.4790	0.7986	0.4713
CatBoost	0.4747	0.8049	0.4716
Ensemble	**0.4736**	**0.8072**	**0.4698**

Table 5.3: Evaluation criteria results of Model 1

Using the best results from the baseline models and Model 1, this can be compared to the models where the temporal aspect of the expected-goals values is considered. In the calibration curves of Model 2 presented in Figure A.3, it can be seen that the models, just as in Model 1, are well calibrated. The results on the evaluation criteria of Model 2 compared to the best results of the baseline models and Model 1 are presented in Table 5.4. It can be seen that these models outperform the best baseline model, which is Baseline 2, but that Model 1 outperforms Model 2 in terms of the evaluation criteria presented.

Learning algorithm	Log loss	AUC-ROC	Brier score
Logistic Regression	0.4792	0.8007	0.4758
XGBoost	0.4812	0.7945	0.4806
CatBoost	0.4772	0.7995	0.4769
Ensemble	0.4753	0.8028	0.4726
Baseline 2	0.5316	0.7460	0.5288
Model 1	**0.4736**	**0.8072**	**0.4698**

Table 5.4: Evaluation criteria results of Model 2

The last model to consider is Model 3, which is also well calibrated as can be seen in Figure A.4. The results on the evaluation criteria of Model 3 compared to the other models are presented in Table 5.5. It can be seen that Model 3 outperforms the best baseline model and the ensemble models of Model 1 and Model 2 on the evaluation criteria.

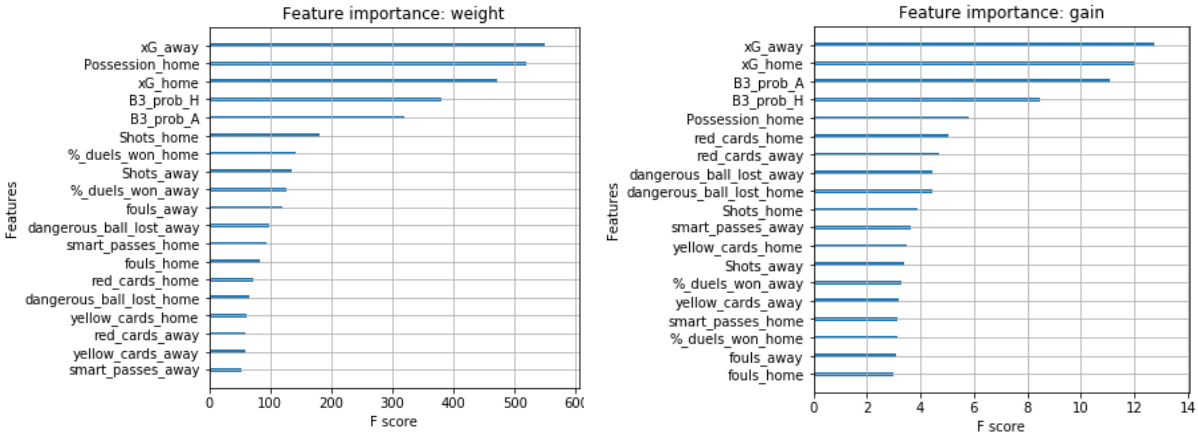
Learning algorithm	Log loss	AUC-ROC	Brier score
Model 3*	**0.4647**	**0.8154**	**0.4615**
Baseline 2	0.5316	0.7460	0.5288
Model 1	0.4736	0.8072	0.4698
Model 2	0.4753	0.8028	0.4726

* The recurrent neural network presented in Figure 4.1.

Table 5.5: Evaluation criteria results of Model 3

5.2.2 Feature analysis

As indicated in section 5.1.2, a feature analysis can improve the interpretation of machine learning models. Using the same metrics as in section 5.1.2, the results of the feature analysis of Model 1 is presented in Figure 5.2. It can be seen that the amount of summed expected-goals values are very important features in terms of weight and gain. Next to this, it can be seen that it is a more important feature than the number of shots for the home and away team, which is as expected. The number of red cards of the home and away team have a low weight but a large gain, since red cards do not occur that often but if it occurs it has a relative big influence on the estimation of the match outcome probabilities.



(a) Weight metric

(b) Gain metric

Figure 5.2: Feature importance metrics of XGBoost model for estimating match outcome probabilities

5.3 Case study: Expected league tables

5.3.1 Introduction to expected league tables

A practical use of the estimation of match outcome probabilities is the expected rankings of football leagues. For every match, the estimated match outcome probabilities can be transformed into the expected number of points of a team for this match by:

$$E[\text{Points}_{\text{Home team}}] = P[\text{Home win}] \quad 3 + P[\text{draw}] \quad 1 \tag{5.1}$$

$$E[\text{Points}_{\text{Away team}}] = P[\text{Away win}] \quad 3 + P[\text{draw}] \quad 1 \tag{5.2}$$

This formula is based on the fact that in football there are 3 points obtained for a win and 1 point for a draw. Note that the sum of expected points obtained of both teams does not always sum up to 3: there is always a probability of a draw in which in total only 2 points are awarded instead of 3.

If the number of expected points is aggregated over a period of time, a ranking based on the number of expected points can be made. In the long run, the expected points is expected to converge to the actual number of points obtained. However, in the short run this is not necessarily true and can be misleading for the actual performances of teams.

5.3.2 Case: Eredivisie 2018/2019

In this case study the Dutch Eredivisie of the season 2018/2019 is examined. The results of Model 3 are used to estimate the match outcome probabilities for the first half of the season. All the expected points and expected goals variables are summed up and can be seen in Figure B.1.

The teams are sorted by the actual ranking after 17 games, but in the column "xP_ranking" the ranking based on the expected points are shown. It is interesting to see some differences in the actual points and the expected number of points. For example, FC Groningen is in place 15 while the model expects them to be in place 9. This indicates that they might have been unlucky in the first half of the season and this is useful information to have. The management of FC Groningen might based on this not fire the manager, since based on the performances of the team they should not be that low on the league table. Next to this, for betting purposes it might be interesting to bet on FC Groningen since the bookmakers might be underestimating the team because of their low position. On the other hand, Fortuna Sittard have been lucky since they are 4 places higher than expected and got 2.15 points more than expected. This might give an indication that they are not yet safe in the fight against relegation.

In Figure B.2 the end ranking of the Eredivisie in May 2019 is presented. It can be seen that FC Groningen eventually ended up in place 9 and Fortuna Sittard ended up just above the relegation mark in place 15. It remains difficult to say whether these results come from the amount of "luck" in the first half, the better performance of the team or the addition of new players to teams, but it shows that it may be a better predictor than the current ranking.

5.3.3 League predictions

In the section 5.3.1 the intuition and the use of the expected points ranking is presented, but the question remains whether the ranking based on expected points is a better predictor than the actual ranking. In order to test this, first for all leagues[1] the expected ranking after half of the competition is played is calculated. The actual rank is used as a baseline prediction and the prediction is the ranking based on the number of expected points. This is used to predict the ranking at April 2019, where only the matches played of the second half of the competition are taken into consideration. This new ranking is used instead of the actual final ranking since the actual final ranking already contains information about the first half of the season.

As a evaluation criteria the rank-order correlation of C. Spearman (1904) is used, which uses the following formula to calculate the correlation between two rankings:

$$= 1 \quad \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 \quad 1)} \tag{5.3}$$

where is the rank-order correlation, d_i is the difference between the ranks and n is the number of cases. For all competitions this rank-order correlation is calculated and the average over the competitions is taken. The comparison between the predictions by the models and the actual new ranking is presented in Table 5.6. It can be seen that the average rank-order correlation of the ranking based on the expected-points is higher than the average rank-order correlation of the actual ranking after half of the competition is played.

League	Actual ranking	Expected points ranking
Premier League (England)	0.591	0.627
Bundesliga (Germany)	0.778	0.728
Serie A (Italy)	0.759	0.651
Ligue 1 (France)	0.750	0.726
La Liga (Spain)	0.500	0.683
Eredivisie (the Netherlands)	0.366	0.525
Average	0.624	**0.657**

Table 5.6: Rank-order correlation results of league predictions

[1] The Belgian rst division is excluded because of the di erent competition format

Chapter 6

Discussion & Conclusion

The main purpose of this thesis was to see whether using more advanced statistics improves the estimation of outcome probabilities of football matches. In this section the different sub-questions addressed in section 1.3 will be discussed, the contribution of this thesis to the literature will be stated and the limitations and possible future work will be addressed.

6.1 Research questions

The first sub-question was whether the addition of player-specific information could improve the prediction accuracy of expected-goals for individual shots. The player-specific information from FIFA 19 was added to every shot, where the choice of the FIFA 19 attribute depended on the characteristics of the shot. The results on the feature importances of the XGBoost model show that the FIFA 19 variable is often used in the decision process, but the contribution to the predictions is relatively small. However, the results of the model with the player-specific information included are better than the results of the model without the player-specific information. Next to this, the expected-goals model show better results than previous research by Eggels (2016) and Van den Hoek (2019). From this it can be concluded that the addition of player-specific information improves the prediction accuracy of expected-goals values.

The second sub-question was whether the addition of match statistics could improve the estimation of outcome probabilities of football matches. A couple of match statistics were gathered and, next to the bookmaker odds and the summed expected-goals values, used to estimate the match outcome probabilities. The results of the machine learning models in comparison to the baseline show that the estimation improves relative to the baseline models. From this it can be concluded that the addition of match statistics improves the estimation of outcome probabilities in football matches.

The third sub-question was whether considering the temporal aspect of expected-goals values could improve the estimation of outcome probabilities of football matches. In this thesis two different ways of considering the temporal aspect are presented. The first option with time bins with expected-goals values shows better performance than the baseline models, but no better

performance than the model where the summed total of the expected-goals values were used. This might be caused by the fact that the machine learning algorithms have difficulties with extracting the useful data from the time bins relative to the summed up values used in Model 1. The second option with the recurrent neural network shows better performance than the baseline models and the earlier used models. From this it can be concluded that considering the temporal aspect of expected-goals values improves the estimation of outcome probabilities of football matches.

The case study shows how the results of this thesis can be used into practice. The tables based on expected points can be very interesting for management decisions and betting companies. Next to this, it is also showed that the rankings based on the expected points are better in predicting the final ranking than the actual ranking during the winter break.

The main question was whether using more advanced statistics improves the estimation of outcome probabilities of football matches. It is shown that adding player-specific information to the expected-goals models improves the prediction accuracy of expected-goals values, that adding match statistics improves the estimation of the outcome probabilities and considering the temporal aspect of the expected-goals values improves the estimation of the outcome probabilities even further. Hence, it can be concluded that using more advanced statistics improves the estimation of outcome probabilities of football matches.

6.2 Contribution

This thesis shows that in order to estimate the outcome probabilities of a football match, it is beneficial to take more statistics into consideration than just the number of expected-goals per team. Next to this, the expected-goals values used for the estimation of outcome probabilities can be improved by adding player-specific information, since the target of the model is not to assess player performance over time but to estimate the outcome probabilities of a match as precise as possible. The exploration of the temporal aspect of the expected-goals values shows that simply adding up the expected-goals values misses a lot of information which can be exploited and improve the estimations of the outcome probabilities. The case study in this thesis shows how the results can be presented and used in the fields of management and betting.

6.3 Limitations and future work

One of the limitations of this thesis is the merging of the data sets, since for some players the matching by using the Levenshtein distance does not always give the results that are expected. This is for example a big problem for players for which the birth name and the football name is very different, so is Isco's (midfielder of Real Madrid) name officially Francisco Roman Alarcon Suarez. This is in terms of matching very hard to find and can lead to mistakes or not lead to the wanted player-specific information. Next to this, in a new season there are new players and hence the merging of the data sets is never completely finished. Furthermore, there is a problem

that there is no ground truth for the estimation of the match result probabilities, which makes the evaluation of the results more difficult. The approach in this thesis to cope with this, is by comparing the results of the models to the results of baseline models which are also already used in practice.

For further research it is interesting to base the expected-goals models on spatio-temporal tracking data instead of just event data. This extra information could improve the expected-goals models since more information is available about for example how many players are close to the ball at a shooting opportunity. Next to this, it is interesting to look how the estimation of the match result probabilities can lead to a profitable betting strategy. As shown in the case study, the ranking based on expected-points is a better predictor than the actual ranking at the winter break. It might be the case that it also outperforms the bookmakers expectations about the final ranking, which can result into a profitable betting strategy over a period of time.

Bibliography

Aberdeene, J. (2012). How to calculate soccer possession.

bet.me. (2018). Using expected goals (xg) to influence the way we bet on football. Retrieved from https://bet.me/usingexpectedgoalstobetonfootball.html

Brownlee, J. (2018). How to know if your machine learning model has good performance. Retrieved from https://machinelearningmastery.com/how-to-know-if-your-machine-learning-model-has-good-performance/

Burn-Murdoch, J. (2018). How data analysis helps football clubs make better signings. Retrieved from https://www.ft.com/content/84aa8b5e-c1a9-11e8-84cd-9e601db069b8

Caley, M. (2013). The minute-by-minute database iii: New expected goals and introducing sibot. Retrieved from https://cartilagefreecaptain.sbnation.com/2013/6/27/4463922/tottenham-hotspur-analysis-epl-expected-goals-statistics

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. (pp. 785–794). doi:10.1145/2939672.2939785

Chollet, F. (2017). *Deep learning with python* (1st). Greenwich, CT, USA: Manning Publications Co.

Cronin, B. (2019). Understanding the limitations of expected goals. Retrieved from https://www.pinnacle.com/en/betting-articles/Soccer/limitations-of-expected-goals/K4GJCLWX3VS6MWVK

Dey, A. (2016). Machine learning algorithms : A review.

Dixon, M. & Coles, S. (1997). Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *46*, 265–280. doi:10.1111/1467-9876.00065

Dwarampudi, M. & V Subba Reddy, N. (2019). Effects of padding on lstms and cnns.

Eggels, H. (2016). *Expected goals in soccer: Explaining match results using predictive analytics* (Master Thesis, Eindhoven University of Technology).

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211. doi:https://doi.org/10.1016/0364-0213(90)90002-E

Fernandez, J. & Bornn, L. (2018). Wide open spaces: A statistical technique for measuring space creation in professional soccer.

Goodman, M. (2018). The dual life of expected goals (part 1). Retrieved from https://statsbomb.com/2018/05/the-dual-life-of-expected-goals-part-1/

Green, S. (2012). Assessing the performance of premier league goalscorers.

Guardian-Sport. (2019). What are the lowest xg-scoring football matches in history? Retrieved from https://www.theguardian.com/football/2019/feb/06/what-are-the-lowest-xg-scoring-football-matches-in-history-expected-goals-the-knowledge-football

Gurpinar-Morgan, W. (2015). On single match expected goal totals. Retrieved from https://2plus2equals11.com/2015/12/16/on-single-match-expected-goal-totals/

Hamilton, H. (2017). Expected saves: An inverse of expected goal?

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput. 9* (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

Holland, M. (2018). Sports betting: Why odds change and how to take advantage.

Ijtsma, S. (2013). Forget shot numbers, let's use expected goals instead. Retrieved from http://11tegen11.net/2013/06/15/forget-shot-numbers-lets-use-expected-goals-instead/

Ijtsma, S. (2015). The best predictor for future performance is expected goals. Retrieved from http://11tegen11.net/2015/01/05/the-best-predictor-for-future-performance-is-expected-goals/

Jain, A. (2016). Complete guide to parameter tuning in xgboost with codes in python. Retrieved from https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

James, B. (1986). *The bill james historical baseball abstract*. Villard Books. Retrieved from https://books.google.nl/books?id=-JcYAAAAIAAJ

KumarSingh, B., Verma, K. & S. Thoke, A. (2015). Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. *International Journal of Computer Applications*, *116*, 11–15. doi:10.5120/20443-2793

Langseth, H. (2013). Beating the bookie: A look at statistical models for prediction of football matches. (Vol. 257, pp. 165–174). doi:10.3233/978-1-61499-330-8-165

Levenshtein, V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, *10*, 707.

Lewis, M. (2003). *Moneyball: The art of winning an unfair game*. Norton paperback. W.W. Norton. Retrieved from https://books.google.nl/books?id=RWOX%5C_2eYPcAC

Lindstrom, J. (2014). The dixon-coles model for predicting football matches in r (part 1).

Lucey, P., Bialkowski, A., Monfort, M., Carr, P. & Matthews, I. (2015). "quality vs quantity": Improved shot prediction in soccer using strategic features from spatiotemporal data. Retrieved from http://www.sloansportsconference.com/content/quality-vs-quantity-improved-shot-prediction-in-soccer-using-strategic-features-from-spatiotemporal-data/

Macdonald, B. (2012). An expected goals model for evaluating nhl teams and players. Retrieved from http://www.sloansportsconference.com/content/an-expected-goals-model-for-evaluating-nhl-teams-and-players-an-expected-goals-model-for-evaluating-nhl-teams-and-players/

Navlani, A. (2018). Understanding logistic regression in python. Retrieved from https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python

Page, D. (2015). Expected goals just don't add up - they also multiply. Retrieved from https://medium.com/@dannypage/expected-goals-just-don-t-add-up-they-also-multiply-1dfd9b52c7d0

Parkes, D. (2018). The roc curve. Retrieved from https://deparkes.co.uk/2018/02/16/the-roc-curve/

Peretz, T. (2018). Mastering the new generation of gradient boosting. Retrieved from https://towardsdatascience.com/https-medium-com-talperetz24-mastering-the-new-generation-of-gradient-boosting-db04062a7ea2

Pollard, R., Ensum, J. & Taylor, S. (2004). Estimating the probability of a shot resulting in a goal: The effects of distance, angle and space. *Int. J. Soccer Sci. 2*.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. & Gulin, A. (2017). Catboost: Unbiased boosting with categorical features. arXiv: 1706.09516 [cs.LG]

Punter2Pro. (2017). How expected goals (xg) will change the way we bet on football. Retrieved from https://punter2pro.com/expected-goals-xg-football-betting-analysis/

Qiao, F. (2019). Logistic regression model tuning with scikit-learn. Retrieved from https://towardsdatascience.com/logistic-regression-model-tuning-with-scikit-learn-part-1-425142e01af5

Rue, H. & Salvesen, O. (1997). Predicting and retrospective analysis of soccer matches in a league.

SciSports. (2016). Goal importance.

SciSports. (2018). We are proud to assist the royal belgian football association towards and during the world cup in russia. Retrieved from https://www.scisports.com/we-are-proud-to-assist-the-royal-belgian-football-association-towards-and-during-the-world-cup-in-russia/

Sheehan, D. (2017). Predicting football results with statistical modelling.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, *15*(1), 72–101. Retrieved from http://www.jstor.org/stable/1412159

Spearman, W. (2018). Beyond expected goals.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res. 15*(1), 1929–1958. Retrieved from http://dl.acm.org/citation.cfm?id=2627435.2670313

Stanley, A. (2019). Alternative premier league table after 32 games, based on expected goals. Retrieved from https://talksport.com/football/522285/alternative-premier-league-table-expected-goals-2/

Stanton, J. (2017). Expected goals: What are we learning from new metric used on match of the day? Retrieved from https://www.bbc.com/sport/football/41822455

Strumbelj, E. (2014). On determining probability forecasts from betting odds. *International Journal of Forecasting*, *30* (4), 934–943. doi:https://doi.org/10.1016/j.ijforecast.2014.02.008

Trainor, C. (2013). Chelsea's striker options. Retrieved from https://statsbomb.com/2013/08/chelseas-striker-options/

Van den Hoek, N. (2019). *Improving expected-goals models: Towards more accurate values for individual shots by considering more detailed information* (Master Thesis, Jheronimus Academy of Data Science).

Vorhies, W. (2016). Want to win competitions? pay attention to your ensembles. Retrieved from https://www.datasciencecentral.com/profiles/blogs/want-to-win-at-kaggle-pay-attention-to-your-ensembles

Appendices

Appendix A

Calibration curves

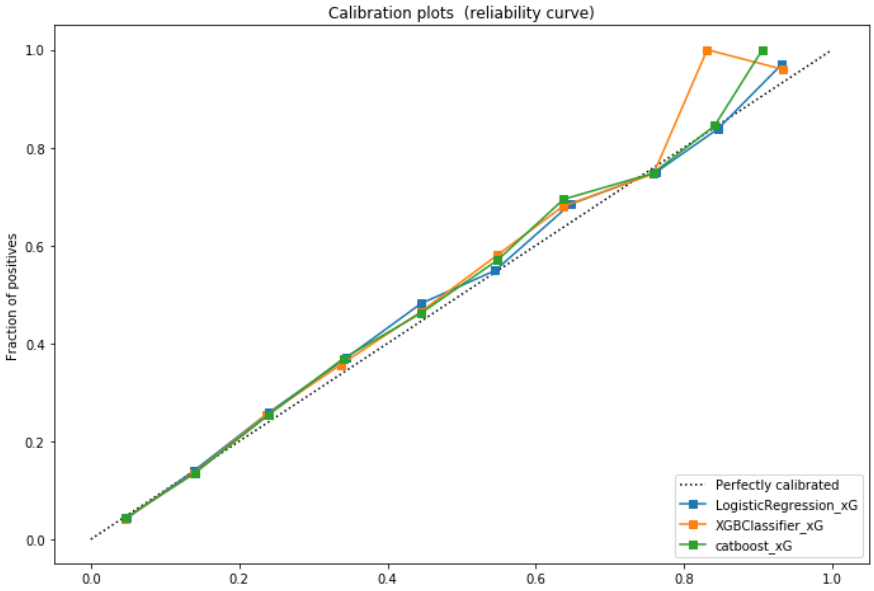
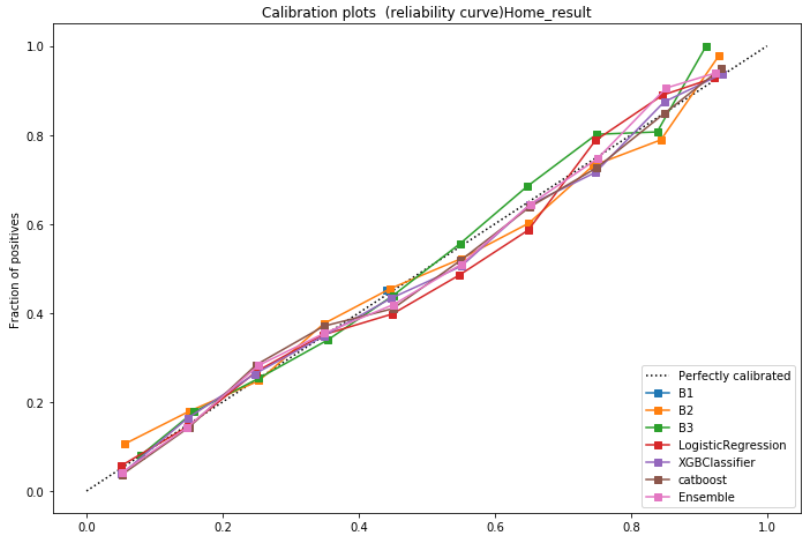
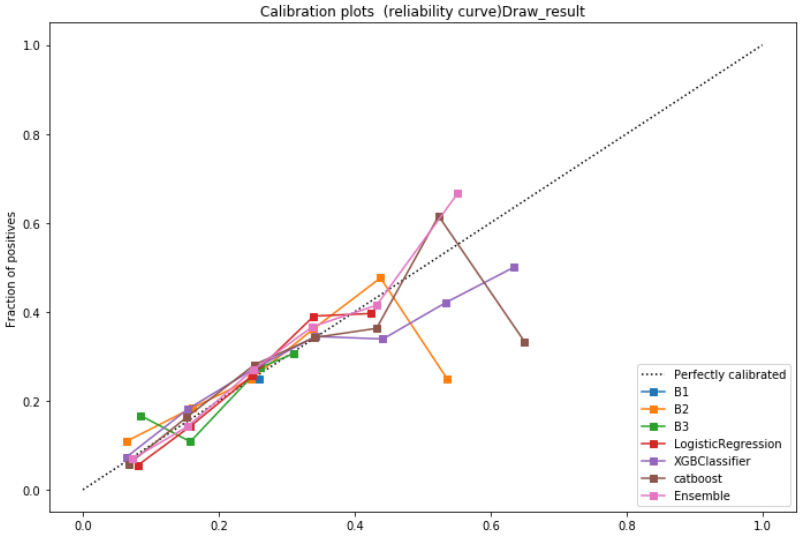


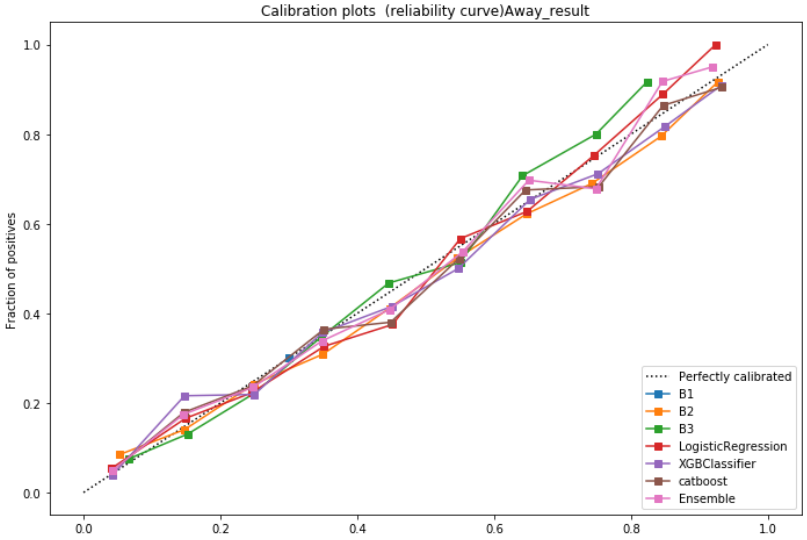
Figure A.1: Calibration curves of the expected-goals models including the FIFA 19 rating



(a) Home win

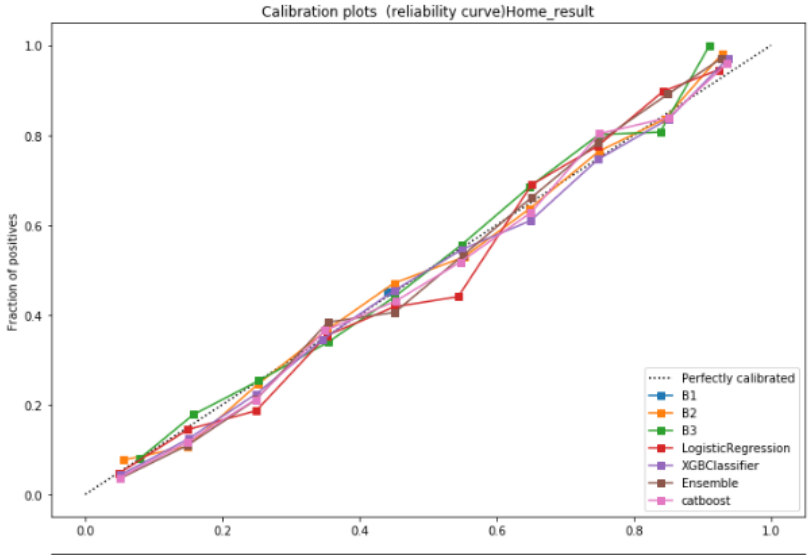


(b) Draw

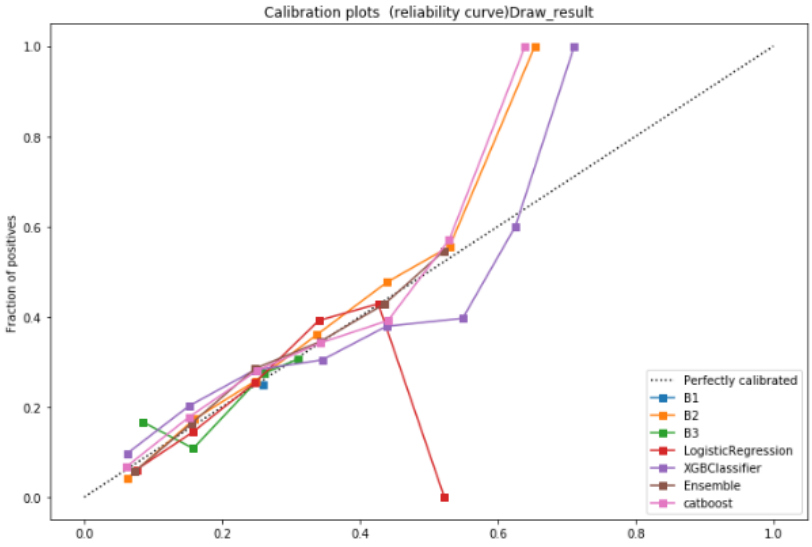


(c) Away win

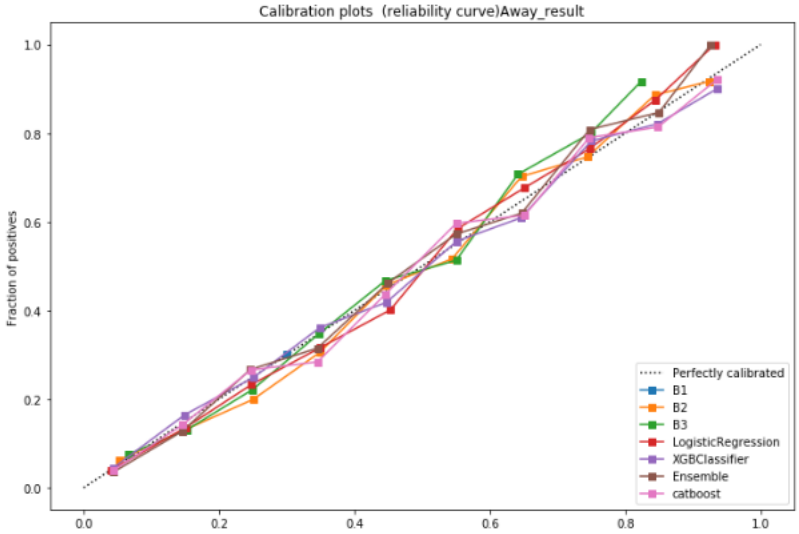
Figure A.2: Calibration curves of match outcomes of Model 1



(a) Home win

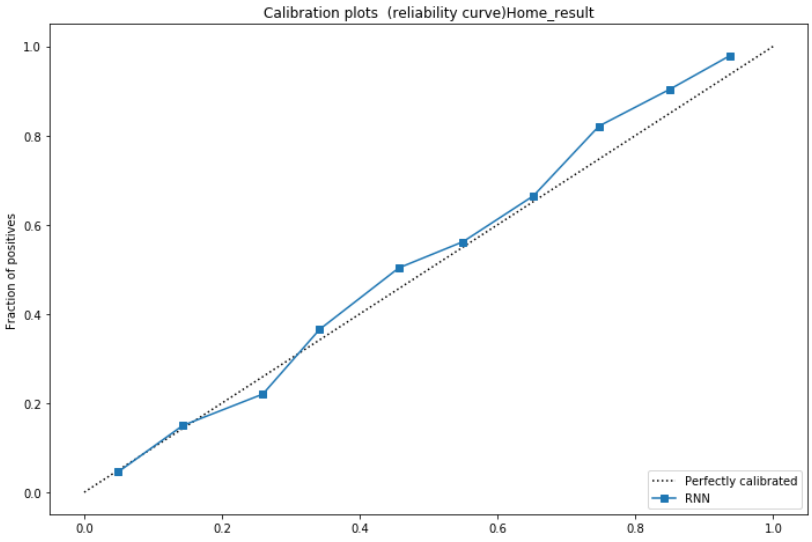


(b) Draw

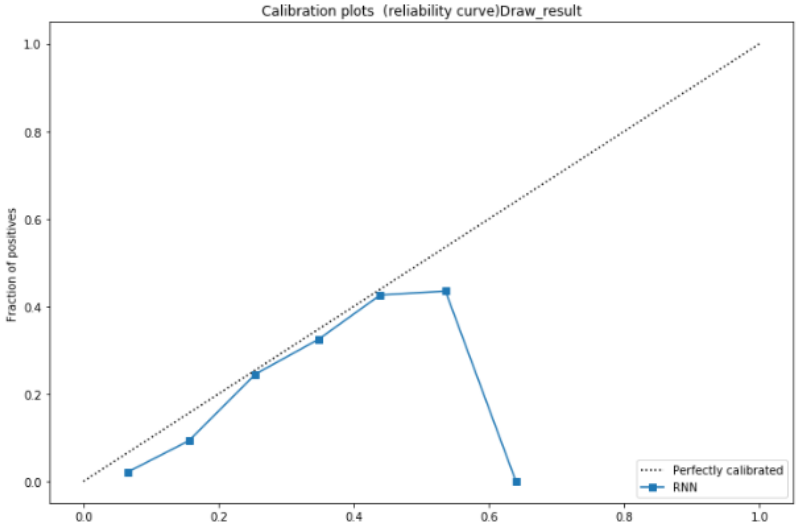


(c) Away win

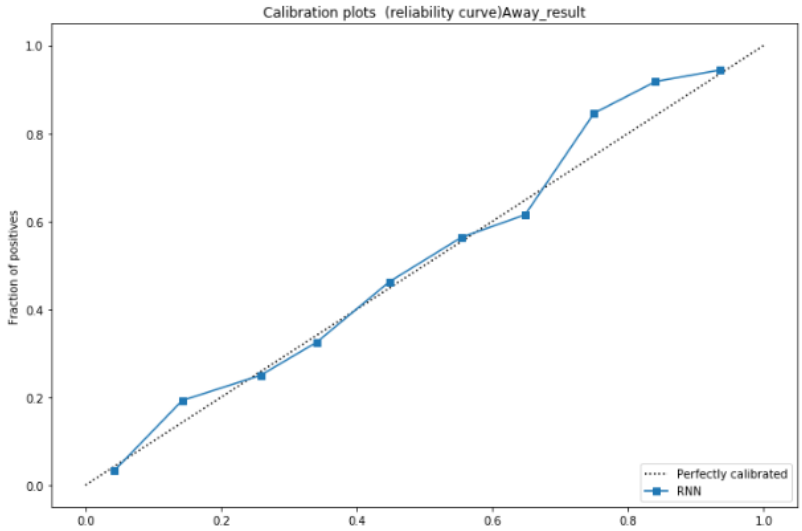
Figure A.3: Calibration curves of match outcomes of Model 2



(a) Home win



(b) Draw



(c) Away win

Figure A.4: Calibration curves of match outcomes of Model 3