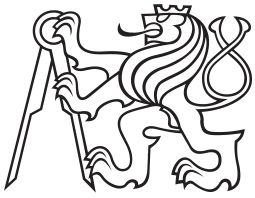
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Predicting Score-related Events in Soccer

Erik Vaknin

Supervisor: Ing. Ondřej Hubáček
Field of study: Open Informatics
Subfield: Data Science
May 2021

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Vaknin Erik** Personal ID number: **466195**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Data Science**

II. Master's thesis details

Master's thesis title in English:

Predicting Score-related Events in Soccer

Master's thesis title in Czech:

Predikování jevů přidružených fotbalovým výsledkům

Guidelines:

While predicting the outcome of a soccer match is a well-established problem in the literature, a surprisingly low amount of work has been devoted to predicting associated events such as both teams to score, under/over X goals scored, or handicapped victories. Some models in the domain are capable of forecasting the said events by computing the probability distribution over all possible scores. These models use solely the results of the past games as input features. It is desirable to collect more detailed data and examine the added value such data bring. The predictions for the score-related events can be traded on sports betting markets, therefore an evaluation in terms of profitability will be conducted.

1) Research state of the art in modeling score-related events.
2) Collect various historical data.
3) Compare different approaches to modeling the score-related events.
4) Design and implement forecasting models.
5) Evaluate the performance of your models from different perspectives.

Bibliography / sources:

1. Wheatcroft, E., 2020. A profitable model for predicting the over/under market in football. International Journal of Forecasting.
2. Wheatcroft, E., 2020. Forecasting football matches by predicting match statistics. arXiv preprint arXiv:2001.09097.
3. Hubacek, O., Gustav, S. and Zelezny, F., 2021. 40 Years of soccer match outcome modeling: an experimental review. IMA Journal of Management Mathematics (in review)

Name and workplace of master's thesis supervisor:

Ing. Ondřej Hubáček, Intelligent Data Analysis, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **12.02.2021** Deadline for master's thesis submission: **21.05.2021**

Assignment valid until: **30.09.2022**

Ing. Ondřej Hubáček Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I am grateful to Ing. Ondřej Hubáček, research assistant in Intelligent Data Analysis Research Lab, CTU Prague. I am thankful and indebted to him for sharing expertise and sincere and valuable guidance.

I must also express my gratitude to my partner and my family for providing me with support and encouragement throughout my years of study and through the process of researching and working on this thesis. Thank you.

Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 21. May 2021

Abstract

Predicting the outcome of a soccer match is a well-established problem in the literature. But too little endeavor has been devoted to forecasting associated events related to scores, such as over/under X goals scored or whether both teams score in a match. There are models that can predict such events by modeling a probability distribution over all possible scores. These models mostly take only scores of historical matches as input. In this thesis, we have gathered more detailed data and tested a hypothesis that such data can be useful in predicting score-related events. We have designed and implemented multiple models that are either estimating a probability distribution over all possible scores using Poisson distribution or predicting directly an occurrence of a specific event. To verify the hypothesis we have created a simulation for predicting and betting on the events in time. In the end, we have provided an experimental evaluation of all the models and a comparison of their performances from different perspectives. We have found out that detailed match statistics can be very useful in this problem and that using classification models is more suitable for this task than Poisson-based models. Our classification model using detailed data has achieved promising results.

Keywords: machine learning, predictive modelling, soccer, neural networks

Supervisor: Ing. Ondřej Hubáček

Abstrakt

Predikování výsledků fotbalových zápasů je v literatuře dobře zavedeným problémem. Dosud však bylo příliš málo práce věnováno predikování jevů přidružených těmto výsledkům, jako například zda bude celkový počet vstřelených gólů vyšší/nižší než X, nebo skutečnost, zda oba týmy v zápase skórují. Existují modely, které mohou tyto jevy predikovat výpočtem pravděpodobnostních rozdělení přes všechna možná výsledná skóre. Většina těchto modelů používá jako vstupní data pouze skóre historických zápasů. V této práci jsme shromáždili podrobnější data a otestovali hypotézu, že taková data mohou být užitečná při predikci jevů přidružených fotbalovým výsledkům. Navrhli a naimplementovali jsme několik modelů, které buď odhadují pravděpodobnostní rozdělení přes všechna možná skóre s užitím Poissonova rozdělení, anebo predikují přímo výskyt konkrétního jevu. K ověření hypotézy jsme vytvořili simulaci pro predikování a sázení na jevy v čase. Nakonec jsme vyhodnotili všechny modely a srovnali jejich výkony z různých perspektiv. Zjistili jsme, že detailní statistiky mohou být velmi užitečné při řešení tohoto problému, a že klasifikační modely jsou pro tuto úlohu vhodnější než modely založené na Poissonově rozdělení. Náš klasifikační model používající detailní data dosáhl slibných výsledků.

Klíčová slova: strojové učení, prediktivní modelování, fotbal, neuronové sítě

Překlad názvu: Predikování jevů přidružených fotbalovým výsledkům

Contents

Figures

Tables

Chapter 1

Introduction

In recent years the interest in forecasting sports has been growing. Predicting the outcome of a soccer match is already a well-established problem in the literature. But yet too little work has been dedicated to predicting associated events related to scores, such as over/under X goals scored or whether both teams score (BTTS). Some models predict probability distribution over all possible scores for a match. Such models are versatile and can be used for predicting score-related events by deriving the estimated probabilities. Those models mostly use only team IDs and resulting scores of historical matches as learning features. Because of the insufficient coverage of forecasting these events, we have decided to look into this problem more deeply.

In this thesis, we have collected detailed data from multiple sources. That includes more comprehensive match statistics such as shots, possession, passes, fouls, and many others. Besides this, we have been able to retrieve player ratings from the famous video game FIFA developed by the company Electronic Arts, which since the 1990s rates all the soccer players playing in the most famous leagues. Due to collected data about starting lineups of players for our dataset we could employ the ratings into our models.

After analyzing the obtained data, we have designed and implemented multiple types of models that are either Poisson-based or classification models. All of our models are based on neural network architecture. With these models, we have tested a hypothesis that additional data can be useful in predicting score-related events. For verification of the hypothesis, an environment simulating predicting, and betting on the events in time had to be developed. In this work, we have tested many various models with numerous configurations that were using different amounts of input data for making predictions.

For each fundamental type of model, we have chosen the best performing as a final model. During the concluding experiments, whose results we present in this thesis, we have evaluated the final models on a test dataset

that has not been used for evaluation of any of the models before. These comparisons are realized from multiple different perspectives. We have concluded whether more data for forecasting in this problem provides additional value. Furthermore, we have compared the perspective of using Poisson-based models against classification models for this problem. Since the models can be used for trading on the sports betting market, we have evaluated the final models on market odds in terms of profitability.

Chapter 2

Literature review

There has not been much research done on the "over/under" and "both team to score" market in soccer which we focus on. The most relevant article to our thesis is Wheatcroft 2020, where the author describes a model for predicting this market. The paper introduces a set of "Generalised Attacking Performance" ratings that are similar to the pi-ratings. The model is profitable over the course of 12 seasons, but the profit is decreasing in later years, which might be caused by eliminating inefficiencies of the market. The paper also provides evidence that measuring attacking strength in ways other than by scored goals, such as shots and corners, can be significantly more informative.

Articles devoted to models able to forecast probabilities of resulting scores are relevant to this problem as well since probabilities of score-related events can be derived from those. In Maher 1982, the author has introduced a double Poisson model and bivariate Poisson model for modeling soccer match scores. The double Poisson model assumes the scores of the teams to be independent and that the scores follow a Poisson distribution in which its parameter λ determines the scoring rate. The bivariate Poisson model adds some dependency between those distributions. According to a recent paper, Ley, Wiele, and Eetvelde 2019, the models are still very competitive.

In order to exploit inefficiencies of the football market, in Dixon and Coles 1997, the authors have modified the double Poisson model. They provide evidence that the bivariate Poisson model is unable to represent the dependency of goals of the teams for low-scoring games. Therefore they have proposed an improvement of the double Poisson model by adding a dependency of the two distributions that increases probabilities of low-scoring draws. Also, an exponential time weighting for increasing the effect of recent games was introduced in the article.

In the article Karlis and Ntzoufras 2003, the authors have pointed out that the bivariate Poisson model underestimates the probabilities of draws, and to eliminate the problem they have introduced a diagonal-inflated

bivariate Poisson model.

Then, in Karlis and Ntzoufras 2009, the authors, instead of modeling the number of goals scored by each team, have modeled the difference of the number of goals between the teams. The main advantage of this approach is eliminating the need to model the dependency between the goals of the teams. They have used a Skellam distribution for modeling the difference.

Another model, a bivariate Weibull count model, predicting score probabilities was introduced in Boshnakov, Kharrat, and McHale 2017. This model is based on the Weibull distribution and uses Frank's copula for producing a bivariate distribution of the number of goals scored. The model is claimed to outperform the Poisson-based models in predicting the match winner.

In experimental review Hubáček, Šourek, and F. Železný 2019, the authors investigate the top-performing methods in predicting score-based match outcomes, such as Poisson-based models, ranking algorithm Elo, rating system pi-ratings, and PageRank algorithm. Their findings are that the double Poisson model is not only very competitive among other statistical models, but also against more distinct approaches.

A comparison of the bivariate Poisson, Skellam, and ordered probit models was provided in Koopman and Lit 2019, where the bivariate Poisson model has achieved the best results.

Betting on the Asian handicap market has been examined in Constantinou 2020. The author has introduced the first model specifically developed for this market, which is based on hybrid Bayesian networks and rating systems.

In Eggels 2016, the author has developed a method for predicting a winner of a soccer match based on detailed statistics. Detailed match events, such as shots, passes, fouls, and more, tracking of players, and even player ratings are included. The method predicts expected goals, which is used to predict the expected match outcome.

In another study utilizing the concept of expected goals, Brechot and Flepp 2018, the authors emphasize the underestimation of the randomness of match score results by proposing a model for performance evaluation whose predictions are not heavily dependent on recent outcomes. The article claims that expected goals are much better input for predictions than match outcomes. The authors also suggest that the proposed method might be used by decision-makers of soccer clubs to avoid the fallacy of inferring the quality of performance from match outcomes.

Chapter 3

Data

3.1 Data gathering

In this thesis, we examine the value of detailed data about matches for predicting score-related events. We decided to use data including many various statistics as well as data about individual players. There are plenty of large soccer datasets, but mostly they contain only the resulting score or a few statistics. Since the data necessary for this work are not widely available, their gathering was a challenging task. Some websites are selling such data, but it is rather expensive. Other websites provide various data about sports matches, but they are intended mostly for sports fans to review the results and are not easily downloadable, let alone in a functional form. The only non-expensive way to retrieve such data was to design and implement web crawlers.

3.1.1 Data sources

We were unsuccessful in finding a data source that contains all of the information about matches we needed. Therefore we had to work with multiple of them. The process of searching for the optimal data sources was time-consuming. Many websites are providing extremely detailed statistics, but mostly only for the main leagues and for the last few seasons. To make use of wide data for reasonable predictions we need as many samples as possible and one or two seasons are not enough.

As the main source, we chose `https://www.footballcritic.com/` (CR). In addition to the basic information about matches it contains many different statistics, such as shots on/off target, possession, passes, aerials won, and others. Furthermore, the website provides line-ups with denoted positions. For this website, we implemented two spiders for retrieving data about matches, but also players data for later linking with another dataset.

The second source is `https://sofifa.com/` (SF). This website is dedicated to the soccer game series FIFA developed by Electronic Arts. In this game, all soccer players and teams have scores determining their quality. Players have furthermore scores for many different skills, such as finishing, dribbling, acceleration, jumping, and more. Those data might be useful for evaluating the approximate attacking and defending strengths of individual teams which could be used for predicting. For retrieving the data from all the downloaded HTML files we implemented XML parsers. On those websites, every day new data are occurring. The scripts for downloading and parsing the data are designed so that they can be easily modified for re-downloading all the data including the new content. The downloading and parsing process takes tens of hours.

The last source is `https://www.football-data.co.uk/` (FD). This website collects and provides soccer data in an easy-to-use form, unfortunately, they are not very detailed. From this website we only use odds for the over/under market. All of the odds we use are from only a single bookmaker (Bet365), therefore the odds and their margin should be consistent in comparison to using odds from various bookmakers.

3.2 Constructing the dataset

The approach of combining multiple data sources creates obstacles. Each source uses different IDs for matches, teams, and players, but their names differ frequently as well. In order to combine the datasets, we must link the IDs of identical entities.

We had to link players from the CR dataset with those in the SF dataset. The names of the players were very often different. For example, one player is in one dataset by his short name “Pepe” and in the other by his full name “Kléper Laveran de Lima Ferreira” in which “Pepe” is not even a substring, which would be helpful. This was unfortunately too frequent. Another problem is that there are too many names that are identical for multiple players. For combining the players we used the names in combination with dates of birth. Python library fuzzywuzzy proved to be helpful in this task. Unfortunately, even after using many heuristics, there were too many players not linked due to no similarities in names or missing and even erroneous dates of birth. Therefore a few hundred players had to be labeled manually.

To match teams from the CR dataset with those in the FD dataset we have used similarities in scores.

3.3 Exploratory data analysis

In this part of the thesis, we perform an exploratory data analysis. Such analysis is an integral part of a data science project. The goal is to get to know the dataset, explore it, find its main characteristics and other useful pieces of knowledge. Below we provide findings important for this problem.

The data features correlated with targets, in our case total amount of goals and BTTS metric, should be more probably useful in this task than those uncorrelated. For instance, we can expect the numbers of yellow and red cards not to be significantly correlated with the goals. Such features should not be included in the input for the model since they provide no additional value and on top of that, they make the model more complex, which contributes to overfitting the data.

In order to find those features, we have picked the features most directly related to the goals and the events we focus on in this work. We have also included manually created features "BTTS" and "Over 2.5". Namely

- Home/Away team scored
- Home/Away team scored in the first halftime
- Total goals scored
- BTTS
- Over 2.5

For all of the other features we have computed their correlation with the features chosen above and then the average over absolute values of all correlations for each of the features. In the table 3.1 we present the eight most and the eight least correlated features with those picked ones.

It should be pointed out that the fact that a feature A is less correlated with targets than feature B as such does not necessarily mean that feature A is less useful than feature B. A common issue in machine learning is that although the input features are correlated with the target, they are also correlated with each other and that causes problems for many models and algorithms. Also, some features can be useless by themselves but can be useful in combination with other features. Still, this approach is a great indicator and in combination with common sense can be useful. There are other feature selection methods, but due to the fact that we can only predict matches taking place in the near future, those methods become computationally too expensive.

Most correlated	Mean correlation	Least correlated	Mean correlation
A on target	0.28	H corners	0.03
H on target	0.28	H interceptions made	0.03
A total shots	0.14	A corners	0.03
H total shots	0.12	A interceptions made	0.02
H throw ins	0.11	A blocked shots	0.02
A clearances	0.11	H blocked shots	0.02
H crosses	0.10	H tackles	0.02
A aerials won	0.09	A second yellow cards	0.02

Table 3.1: The most and the least correlated features in the dataset

We have analyzed the features more deeply and decided not to include those that seemed to be the least helpful.

In this work, we work with many different leagues. This might be an important factor for making predictions. The question is whether the leagues differ significantly or whether soccer is approximately the same in the top leagues in the world. In the table 3.2 we present the prior probabilities of our target events in individual leagues. We can clearly see that there are large, maybe unexpectedly large, differences. For both "BTTS" and "Over 2.5" events the priors go from around 40% up to 60%. This means we definitely have to include at least some information about the league in the input features given to the model.

In this work, we have been able to retrieve scores for individual players in most of the matches. This might be hypothetically useful for making predictions. During conducting the data analysis, we have tested whether there is at least any interesting information contained in the data. For example, the approach of computing the correlation of mean FIFA score of the home team with the goals they have scored would not provide much value. That is because for being able to make reasonable predictions about the resulting match score we need to include information about both of the teams. In simple terms, it is suboptimal to predict the number of goals a team scores if we do not know who the team plays against. Still, we include the correlation of mean of home team players in the table below for comparison.

For just a simple analysis we measure the correlation of a difference $SF_H - SF_A$, where SF_t is the average over the FIFA scores of t team's players, to the goals scored by the home team. Also, we include the correlation for a difference $SF_H^O - SF_A^D$ where O/D denote only offensive/defensive players.

In the table 3.3 we can see a relatively large correlation, therefore there is likely useful information that might help to improve our models. The great advantage of having these features is that they are directly providing

League	BTTS (%)	Over 2.5 (%)	Total scored	Matches
Allsvenskan	55.09	54.26	2.82	1198
Argentine PD	43.67	39.33	2.26	1429
Bundesliga	57.08	58.08	2.99	1701
Championship	51.06	47.17	2.55	1982
Danish Superliga	54.82	54.11	2.79	1266
Eliteserien	56.33	55.00	2.89	1200
Eredivisie	58.68	60.37	3.12	2299
Ligue 1	48.99	47.48	2.57	2774
Premier League	49.74	51.73	2.72	2865
Primeira Liga	47.29	46.49	2.55	1366
Primera Division	49.84	49.67	2.68	2865
RFPL	44.90	42.61	2.37	1833
Scottish PL	48.89	50.79	2.66	1262
Serie A	54.60	53.59	2.79	2857
Swiss SL	59.80	59.19	3.02	816
Turkish SL	54.95	52.83	2.78	1749

Table 3.2: The prior probabilities of target by leagues

	Home team scored
$SF_H - SF_A$	0.328
$SF_H^O - SF_A^D$	0.318
SF_H	0.174

Table 3.3: Correlation table for FIFA player scores

an estimate of strength for a specific lineup in a match. Whereas for utilizing the statistics, we need to estimate the strength of a team from its historical matches while the team's strength might change significantly between matches, e.g. due to an injury of an important player. Besides that, we can see that the correlation of a mean of only one team is much less correlated with the goals as expected.

3.4 Preprocessing the input data

Some parts of the data can be forwarded to the models in their current form. The team and league IDs can be forwarded to an embedding layer. But some parts of the data need preprocessing.

3.4.1 FIFA player scores

Since the lineups are known always at least an hour or two before a match starts, the FIFA player scores can be obtained by the model before the match. Unfortunately, there are 11 scores for each team which can not be simply given as a vector to a neural network because team formations differ in individual matches.

Formation describes the way the players are approximately positioned on the pitch. A typical formation 4-4-2 denotes a positioning such that there are four defenders, four midfielders, and two forwards. Another common formation is for instance 3-4-3.

For example, we might want just a single neuron to learn a coefficient for each of the eleven positions so that it would multiply them with the scores, sum those, and output a single number representing the attacking strength of the team. This would be possible if there would be only one formation. Because that is not the case, the neuron would for example have to use a coefficient that is mostly associated with a forward for a midfielder or even a defender. Since the formations differ a lot, this would cause problems. Moreover, sometimes the player scores are missing and we want to make predictions even for the matches, in which only a few FIFA player scores are not available, since removing those matches would only further reduce our already not so large dataset.

We have solved this problem in the following way. The players of each team in each match are divided by their positions into three groups - goalkeeper alone, defensive players, and offensive players. An average of FIFA player scores SF_t^g for each group g for each team t is computed, which provides additional 2×3 input features for a model.

3.4.2 Match statistics

Processing match statistics is more complicated. Predicting the resulting scores based on the number of shots on target in the game would have very high accuracy, but it can not be done as the model receives these data after the end of a match.

Here we make an assumption that the teams that have high scores in certain statistics are more probable to maintain those high scores in the future and vice versa. For instance, if a team achieves high ball possession in the last matches, we assume that this team is likely to preserve it in the following matches. Then we can compute an average of the team's individual statistics and make it an input feature for the model. For it to be representative we need to compute the average from a reasonable amount of historical matches.

We designed the computation of the statistics features in the following way. A form $F_{t,s}$ of a team t for a specific statistic s is computed as a weighted average of the last n values $v_{s,t,i}$ of team t for statistic s. We use weighted average because recent matches are more relevant for the computation of form. As a weighting function, we use successful exponential weighting introduced in Dixon and Coles 1997.

$$F_{t,s} = \frac{\sum_{i=M_t-n}^{M} w_{t,i} v_{s,t,i}}{\sum_{i=M_t-n}^{M} w_{t,i}} \tag{3.1}$$

$$w_{t,i} = e^{-\alpha T_\Delta} \tag{3.2}$$

where α is a metaparameter and T_Δ is the number of days passed since when the match was played. For α we use a value 0.002 that was found to be well-performing in Boshnakov, Kharrat, and McHale 2017.

In our models, we work with two forms - short-term and long-term form. Therefore for each team for each statistic, two input features are provided in the input. Those two forms differ only in the number of matches n that determines how many last matches the weighted average is computed from. Since there are too many hyperparameters in the model that we could not search the optimal values for all of them, we decided to fix $n = 10$ for short-term form and $n = 60$ for long-term form. Matches with a team that does not have enough preceding matches played are not included in the training set. In the project code included, the values of statistics are stored in a circular buffer for efficiency.

3.5 Final dataset

We have covered the process of gathering, analyzing, and preprocessing the data. Below we present a description of the final dataset that was used to evaluate our models.

We have collected approximately 25000 matches. In the dataset, there are 16 different leagues that can be seen in the table 3.2. For every league, there are from 4 to 7 latest seasons contained where the last matches take place in approximately half of the 2020/2021 season. For 10 leagues for the seasons 2019/2020 and 2020/2021, we have market odds for the "over/under 2.5" set by bookmaker Bet365.

The enumeration of input features available for every match follows.

- League ID

- Home, away team IDs
- Team features. The following are contained in the feature set for each team separately.
  - short-term form - a vector of weighted averages of selected statistics (table 3.4) from the last 10 matches
  - long-term form - a vector of weighted averages of selected statistics (table 3.4) from the last 60 matches
  - SF_t^G - goalkeeper's FIFA player score
  - SF_t^D - average of FIFA player scores of defensive players
  - SF_t^O - average of FIFA player scores of offensive players

Scored	Crosses
Half-time scored	Fouls
Possession	Throw ins
Shots off target	Passes
Shots on target	Passes completed (%)
Total shots	Long balls
Corners	Touches
Pass success percentage	Aerials won
Interceptions made	Clearances

Table 3.4: Match statistics used for short-term and long-term form features

Chapter 4

Models

In this chapter, we describe the models used in this thesis. At first, we present two different classifications of them and describe the pros and cons of each type. Then we go through base models that will be used for comparison with our models. In the end, we focus in detail on the models we introduce.

Before diving into the models, we should briefly describe the way the models will be evaluated since this is not the typical approach in predictive modeling. Knowing the approximate way the models will be evaluated will help understand the models. Because our data are time-dependent, we have implemented a simulation that starts at a certain date and every day it gives to the model results of all the preceding matches and offers all matches that take place on the current day as an opportunity for predicting their outcomes. The full description of the simulation can be found in section 5.1.

The models we present in this thesis are neural networks with various inputs and they either predict parameters of statistical distributions (Poisson-based) or they predict probabilities of individual events (classification).

4.1 Categories of models by input

Most of the basic models used for making predictions in sports take only the IDs of the teams and the resulting scores as input. In this thesis, we aim to find out whether additional data in this task can be helpful. Therefore we divide the models by the data they work with.

4.1.1 TID models

In order to decide whether the additional data make a difference, we need models that do not use those data for predictions, therefore one type of model will obtain only the team IDs and the league ID as an input. For

all the historical matches the models receive the resulting scores. We refer to the models as the TID models. Such models typically try to model the strength of each team and then use it to predict for instance a winner or the expected goals scored by those teams.

4.1.2 STATS models

Those models are built around many features and are thus relatively complex. The following data are forwarded to the model for each match during the simulation. The FIFA player scores are optional.

- team IDs
- league ID
- team features for both teams
  - SF_t^G - goalkeeper's FIFA player score
  - SF_t^D - average of FIFA player scores of defensive players
  - SF_t^O - average of FIFA player scores of offensive players
  - short-term form - a vector of weighted averages of selected statistics (table 3.4) from the last 10 matches
  - long-term form - a vector of weighted averages of selected statistics (table 3.4) from the last 60 matches

4.2 Categories of models by the output

There are two types of machine learning algorithms that can be effectively used for this task - Poisson-based and classification. In this thesis, we have designed and implemented models for both of the types and in this section, we provide their description and comparison. These categories determine the last layer of the neural networks, therefore even the output and learning process.

4.2.1 Poisson-based models

In machine learning, regression is a process of finding a function that predicts values of a continuous output variable y based on input variables X. In this problem, we can use regression to estimate parameters of probability distributions of the match score outcomes. Once the model is fitted to the data, we can predict the probability of any score outcome $P(G_H = x, G_A = y)$

for a match. This is very useful since with this approach only one model is needed and such a model is versatile. With such a model we can not only predict specific match outcomes, but we can derive probabilities of related events, such as "home team wins", "draw", "away team wins" and what interests us the most in this thesis - "over/under 2.5 goals" and "both teams to score" events.

For example, we can compute the probability of the number of goals in a match being less than 2.5 by

$$P(G_H + G_A < 2.5) = \sum_{x,y \in N_0; x+y<2.5} P(G_H = x, G_A = y) \tag{4.1}$$

The disadvantage of this approach is that an assumption about the distribution of goals has to be made, which is limiting the expressivity of the model. There are many variables influencing the game and the distribution of score results. There are different targets. Most of the time the ambition for both teams is to win. Sometimes it seems unlikely for a team to win so the strategy is to play defensively and make it a draw. When the end of a season is approaching, the teams need a certain amount of points in order to finish the season with the best result achievable given the current situation, meaning that some teams do not need any additional points because they are safely in the lead and therefore they do not play aggressively anymore.

The regression of parameters of such models can be performed by optimizing the likelihood of the actual results according to the distributions, in our case the final scores.

Double Poisson model (DP)

Probably the most straightforward approach for predicting the distribution of goals is to use the double Poisson model which was introduced in Maher 1982. The model assumes the scores of both teams to be independent and that each of the scores has a Poisson distribution. Poisson distribution has a parameter λ, which is estimated by the model for each team in a match.

The assumption of the double Poisson model that the scores of both teams are independent is not probably true. Let's have a match where one team scores the first goal in a match in the second half. Both teams have to react to that and adjust their strategy. If the other team's target is to win, the coach sends in more offensive substitutes to turn the match. On the other hand, if the other teams would score, the coach might send in more defensive players to maintain the lead. Even though we believe that the goals scored by the teams are not independent, thus the assumption being incorrect, this simple and well-established model is still very competitive (Ley, Wiele, and Eetvelde 2019) and therefore we will use it for our main reference model.

The probability of a resulting match score between home and away teams being $x : y$ is defined by the model as

$$P(G_H = x, G_A = y|\lambda_H, \lambda_A) = \frac{\lambda_H^x e^{-\lambda_H}}{x!} \cdot \frac{\lambda_A^x e^{-\lambda_A}}{y!} \tag{4.2}$$

where λ_t is a mean of the underlying Poisson distribution. The λ_t can be understood as a scoring rate of team t. In the original paper, those rates were expressed by

$$\begin{aligned} \ln \lambda_H &= S_H - S_A + h \\ \ln \lambda_A &= S_A - S_H \end{aligned} \tag{4.3}$$

where S_t is a strength of a particular team and h is a home advantage. The model assigned a single strength parameter S to every team.

With the neural networks, we can either estimate those strengths and home advantage, or we can estimate the λ parameters directly. We have tested both of these approaches and the former performed significantly better, therefore we have further neglected it and we were predicting the λ parameters indirectly by estimating the strengths and using equation 4.3. The same applies to the following Poisson-based model.

Bivariate Poisson model (BP)

Even though the double Poisson model explains the data well, in fact surprisingly well taking into consideration its simplicity, it is desirable to include a more sophisticated model. We decided to use the idea of the bivariate Poisson model introduced also in Maher 1982. This model enhances the fitting by encompassing a dependency between the two Poisson distributions. For modeling the dependency we chose to use the following formula, which was originally introduced as a bivariate version of the Weibull model in Boshnakov, Kharrat, and McHale 2017.

$$\begin{aligned} P(G_H = x, G_A = y|\lambda_H, \lambda_A) = {} & C(F(x|\lambda_H), F(y|\lambda_A)) \\ & - C(F(x-1|\lambda_H), F(y|\lambda_A)) \\ & - C(F(x|\lambda_H), F(y-1|\lambda_A)) \\ & + C(F(x-1|\lambda_H), F(y-1|\lambda_A)) \end{aligned} \tag{4.4}$$

where F is cumulative distribution function and C is a copula function. In our case F can be computed as

$$F(x|\lambda_t) = \sum_{i=1}^{x} f(i;\lambda_t) = \sum_{i=1}^{x} \frac{\lambda_t^i e^{-\lambda_t}}{i!} \tag{4.5}$$

where f is Poisson probability mass function.

We decided to use Ali-Mikhail-Haq copula function (Kumar 2010) for this model. This function is defined as

$$C(u,v) = \frac{uv}{1-\kappa(1-u)(1-v)} \tag{4.6}$$

where $\kappa \in [-1,1]$ is a copula parameter. In addition to learning the λ or S parameters, this model will learn also the κ parameter. Due to the nature of our problem and repeated training during the simulation, κ will change in time.

4.2.2 Classification models

This approach is very distinct. There is no assumption about a distribution. In classification there are classes, in our case the events, we aim to predict. The training sample consists of the input features and a label, e.g. event "over 2.5 goals" being true or false. The models do not output any probabilities of concrete scores, only probabilities for each of the classes (this is not even imperative, some models do output only the estimated class instead of probabilities).

The biggest disadvantage of this approach is that there are no relationships of the classes involved in the process of making predictions even though the relationships exist. Let's consider four possible events - the resulting score being $0:0$, the score being $0:1$, the score being $1:0$, and the event of the number of goals in a match being less than 1.5. For each of the individual events, we can create a classification model predicting the probability of such an event. The problem is the possible inconsistency of those models as it is not guaranteed that

$$\sum_{x,y \in N_0; x+y<1.5} P_{x,y} \stackrel{?}{=} P_{<1.5} \tag{4.7}$$

where $P_{x,y}$ is the predicted probability of the score $x:y$ and $P_{<1.5}$ is the predicted probability of "under 1.5" by the corresponding models.

Another issue of this way of predicting is that for every type of event we need a new model which can be computationally expensive if we aim to predict multiple different events.

On the other hand, an advantage of this approach is that it is much more direct and no assumptions are made. Such models focus only on minimizing a loss function for the target we have specified which can lead to higher accuracy.

4.3 Architecture

In the previous section, we have covered two categorizations of models, each yielding two categories of models. All their combinations give us 4 fundamental types of models.

- TID + Poisson-based - Models that receive team IDs as the input and predict parameters of a statistical distribution. We refer to them as **TID-DP** if they model double Poisson distribution or **TID-BP** if they model bivariate Poisson distribution.
- TID + classification - Models that receive team IDs as the input and predict probabilities of particular events (classification). We refer to them as **TID-C** models.
- STATS + Poisson-based - Models that receive statistics and FIFA scores as input and predict parameters of a statistical distribution. We refer to them as **STATS-DP** if they model double Poisson distribution or **STATS-BP** if they model bivariate Poisson distribution.
- STATS + classification - Models that receive statistics and FIFA scores as input and predict probabilities of particular events. We refer to them as **STATS-C** models.

We were able to implement all those models as feed-forward neural networks due to their versatility. We have designed them so that each of them can be configured in many possibilities and many of those possibilities are common for all of them.

All the models have a variable number of hidden linear layers followed by a variable and optional activation function.

The difference between TID and STATS models lies in the first layer. The first layer of the TID models consists of two parallel embedding layers with a configurable dimension, one for home team ID, second for away team ID. The first layer of the STATS models consists of two parallel linear layers. Each of the linear layers processes the statistics and FIFA scores of one team. In all the models the output of the first layer is then forwarded to a sequence of linear layers.

The first layer's embedding layers or linear layers in one model can be, but do not have to be, identical (based on the configuration). Them not being identical means that the first layer learns all the coefficients separately for home and away teams. This might provide a better fit to the data if enough data is provided. We work mostly with the variant of them being identical.

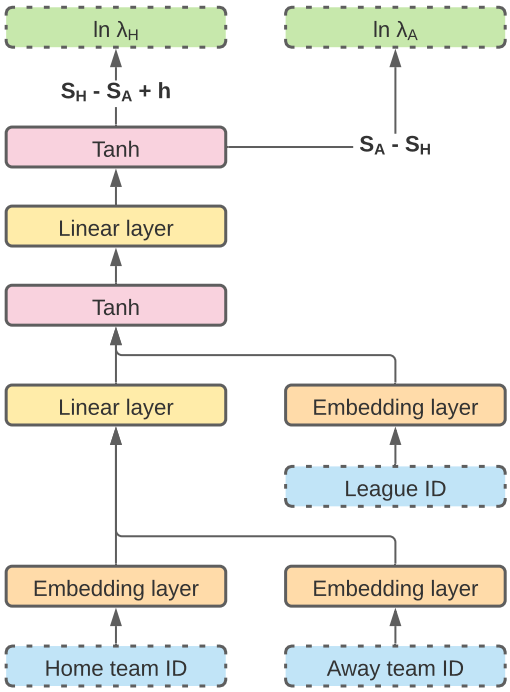


Figure 4.1: Sample TID-DP/BP model architecture

Similarly, the difference between the Poisson-based and classification models is in the last layer. The output of the sequence of hidden layers for the Poisson-based models represents the strengths S_H, S_A from which the λ parameters of the particular distribution are computed by applying transformation described by equation 4.3. On the other hand, the output of the sequence for the classification models is a vector with a dimension of the number of classes the model predicts. A softmax function is then applied to this vector, which gives the estimated probabilities for each class.

Before the last fully connected linear layer in the sequence of hidden layers, the league ID is fed to an embedding layer. The output of the embedding layer is then concatenated with the current output of the sequence of fully connected layers. This is then forwarded to the last fully connected layer.

The last experimental model we present is a combination of STATS and TID models for classification denoted by STATS+TID. It is a STATS

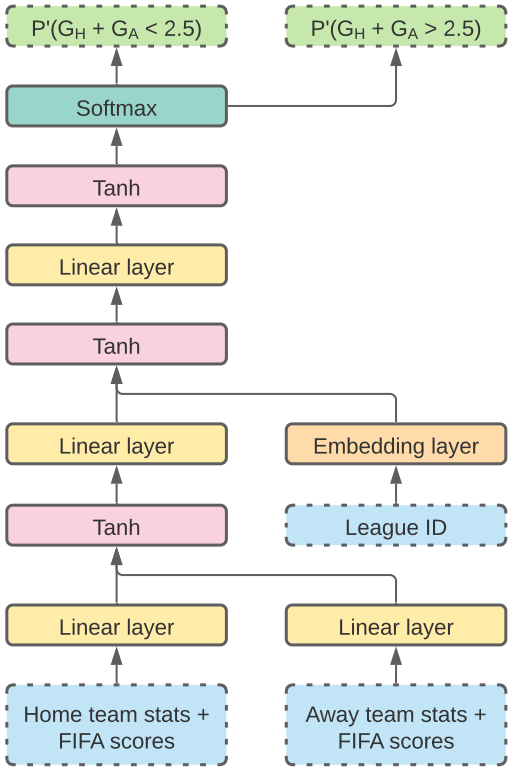


Figure 4.2: Sample STATS-C model architecture

model as we have described, only before the last linear layer in the sequence an output of a full TID model is concatenated (together with the embedded league) with the current output of the sequence. The idea is to utilize all the available information for predictions, but it may lead to an overly complex model for the amount of data we have.

A sample architectures can be seen in figures 4.1 and 4.2. The former can represent both the TID-DP or the TID-BP model as their architecture is the same since they differ only in a loss function. The last tanh activation function outputs a vector of dimension two, the estimated strengths S_H, S_A. The latter represents the STATS-C model predicting "over/under 2.5". The estimated probabilities are represented by $P'(X)$.

4.3.1 Configurations

The models have a lot of hyperparameters and there are too many configurations for us to evaluate all of them. Therefore we had to fix some of them to a certain value. Such value was mostly tested in a few evaluations and compared to an alternative while the performance was reasonably good.

The value could not be tested with all the possible configurations, therefore there might be, and probably are even better configurations. An enumeration of hyperparameters influencing performance can be found in table 4.1.

If a hyperparameter is fixed, its value is present in the table, otherwise, the hyperparameter is evaluated for multiple values. All the hyperparameters are related to all the models except for those with the * symbol, which are related only to the STATS models. "Min training samples" defines the number of samples needed for the model to begin training and making predictions. The learning rates were different for individual models based on which seemed to work well.

More important are the linear layers and activations. In this thesis, the linear layers are represented by a list of dimensions of individual layers. To illustrate, a value $[6, 4, 3]$ represents a model with 3 hidden linear layers - FC1, FC2, FC3 with dimensions $[6, 4]$, $[4, 3]$, $[3, D_O]$ respectively, where D_O is the dimension of output which is always 2 except for classification models predicting "home/draw/away" (HDA) events. The first number in the list determines a dimension D_I of the output of the model's first layer, which differs between TID and STATS models, but D_I is the same because the first layer of each of them outputs two vectors, one for each team. In this case for TID models, the dimension of the two embedding functions would be $D_I = \frac{6}{2} = 3$ for each team. Similarly, the output dimension of the two linear layers in the STATS model would be $D_I = \frac{6}{2} = 3$.

The activation functions tested in this work are sigmoid function, tanh function, and ReLU, and also using no activation function is evaluated. Only one of these options is always applied to the whole model, meaning that no model combines two different activation functions.

hyperparameter	fixed value
n matches in short-term form*	10
n matches in long-term form*	60
weighted average of statistics*	True
league embedding dimension	3
min training samples	1000
min training samples (league)	100
optimizer	Adam
same first layer	True
learning rate	0.0003-0.03
linear layers	
activation function	

Table 4.1: Hyperparameters of the models

4.4 Fitting the models

Although we have designed all the models to be neural networks, performing a parameter optimization by minimizing a loss function significantly differs between the statistical and classification models. We discuss that in this section.

A typical training of a model in machine learning is performed by splitting the dataset into two disjoint subsets (training set and validation set), then the model is fitted to the training set and evaluated on the validation set. In this specific problem, during the simulation, the model is fitted to the currently available historical data in every step. Since we do not possess too much data, giving up a part of the data for validation becomes too expensive.

Furthermore, the models present in this thesis are mostly quite complex and there are many input features. This in combination with an insufficient amount of data can easily lead to overfitting. In order to utilize the full potential of the data we designed the training of our models, which is performed in every step of the simulation, to be done in the following way:

- All the data are randomly split into two disjoint subsets - training set X_{trn} and validation set X_{val}.
- A neural network corresponding to the model and the configuration is initialized.
- The training of neural network is initiated while the number of iterations is being measured. During the training cycle, the training loss (loss measured on the X_trn) and validation loss (loss measured on the X_val) are being monitored. There are two possible ways for the training cycle to be interrupted:
  - The validation loss stops decreasing for too long. When this happens, the model is usually overfitting the data and is unable to better explain the variance in them. The cause of this is the model being too complex, e.g. due to too many input variables.
  - The decrease in training loss drops below a predefined ϵ. This usually represents the model converging to certain values of parameters and not making significant changes anymore. This may either happen when the model is too simple and being already close to the optimal setting of its parameters given the input, or when during the optimization the model hits a local minimum which is a common problem in machine learning.
- No matter which of the opportunities above interrupts the cycle, the

number of iterations for when the validation loss was the lowest is recorded.

- Another training is performed, but this time over all of the data. The stopping criterion is the recorded number of iterations for minimal validation loss in the previous training. This way we expect the model, no matter its complexity, not to overfit the data and at the same time utilize its full potential.

- The resulting neural net is stored and used for predicting the forthcoming matches. Although we have designed all the models to be neural networks, performing a parameter optimization by minimizing a loss function significantly differs between the statistical and classification models. We discuss that in this section.

4.4.1 Poisson-based models

In this work, we fit the Poisson-based models to the resulting scores of matches. It is performed by minimizing over their weighted negative log-likelihood function:

$$L = \prod_{i=1}^{N} P(G_{H,i} = x_i, G_{A,i} = y_i|\theta) \tag{4.8}$$

$$-\ln L = \sum_{i=1}^{N} w_i \cdot \ln P(G_{H,i} = x_i, G_{A,i} = y_i|\theta) \tag{4.9}$$

where θ represents the parameters of the model, w_i is a weight of a match sample and P is defined by the concrete Poisson-based model. As a weighting function, we use the same as for the weighting of statistics in the STATS model. That is the exponential time weighting introduced in Dixon and Coles 1997 as an improvement of the double Poisson model

$$w_{t,i} = e^{-\alpha T_\Delta} \tag{4.10}$$

where $\alpha = 0.002$ is a metaparameter and T_Δ is the number of days passed since when the match was played.

During training the double Poisson model outputs the current estimate of $\lambda_{H,i}, \lambda_{A,i}$ for all the input matches with resulting scores $x_i : y_i$. For such a score a negative log-likelihood has to be computed.

The bivariate Poisson model outputs also the parameter κ and the optimization is performed with respect to that as well.

4.4.2 Classification models

We fit the classification models directly to our target classes. As a loss function, we decided to use a cross-entropy loss that is defined for a sample i as

$$-\sum_{c=1}^{C} y_{i,c} \cdot \ln p_{i,c} \tag{4.11}$$

where C is the number of classes (e.g. $C = 2$ for "over/under 2.5"), $y_{i,c}$ is a binary indicator ($y_{i,c} = 1$ if c is the correct class for the sample i) and $p_{i,c}$ is the estimated probability of class c by the model.

4.4.3 Sample evolution of loss

For illustration purposes, in figures 4.3 we present the training loss during training Poisson-based (left) and classification (right) STATS models both with two inner layers. There are three losses in each figure. The training and validation losses represent the first training as described above. The "train+validation" losses represent the second training on all available data. We can see them stop at the minima of the validation loss as it was designed. The losses are so smooth most likely because each step of adjusting the model parameters is always performed over the whole training dataset. Since the validation losses are decreasing, we can see that the models are learning. Moreover, we can notice some overfitting for both the models represented by the slowly increasing validation loss, which is more visible for the classification model.

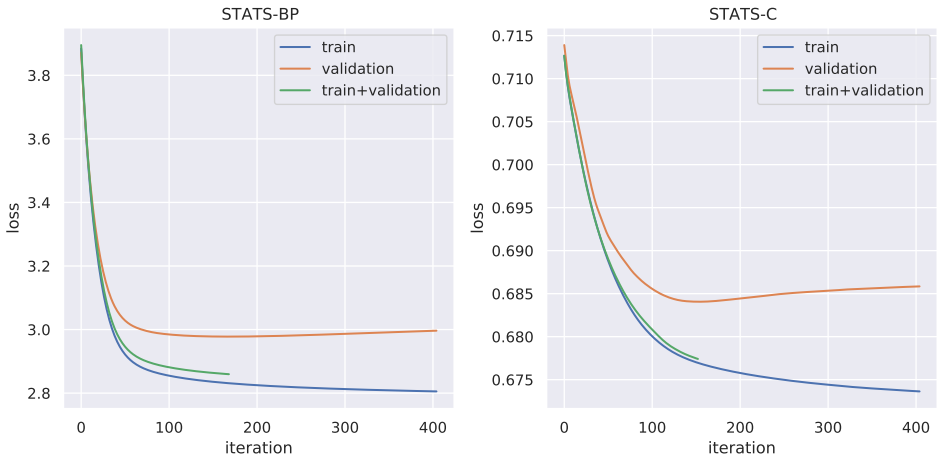


Figure 4.3: Loss of STATS models during sample training

Chapter 5

Experiments

In this chapter, we describe the experiments and their results. There were several goals. One of the goals was to examine how hard it is to predict the score-related events since it was not done thoroughly yet. Also, we aimed to create new models with the potential of providing better prediction results than the baseline models and with the possibility to be easily modified, possibly improved, and built upon. Another goal was to verify the value of additional data on top of the score results for predicting soccer events. Below we provide the description of series of experiments that examine the models from different perspectives.

5.1 Evaluation

The typical evaluation of predictive models is such that the training dataset is divided into two parts - training set and validation set. The model is fitted to the training set and the expected quality of such model is measured on the validation set. This is a very simple yet effective approach. Unfortunately due to the nature of our problem, meaning the fact that our data are time-dependent, we can not use this approach for most of the models.

In this problem for any match, we can utilize the knowledge of all the historical results happening before this match. Of course, we can not include in training any of the matches from the future. One approach would be to split the data in a specific date d into two parts - B (all matches happening before d) and A (all matches happening on or after d), then training on all the B dataset and make predictions for all the matches from A dataset. This procedure is suboptimal. For most of the matches in A, there are other matches that take place before them and the model might utilize their results for the predictions.

We have designed and implemented an environment as an entity

that simulates real-world predicting of time-dependent soccer events. This environment takes as input all of the data and the model. Based on the required data features of the model the environment builds a dataset for the session. When everything is set, the simulation starts. During the simulation, the environment starts at some specific date and for every day it performs the following:

- It gives the model all of the historical results so that the model can fit the data that include the latest results. This is very important for the double Poisson model described below since it tries to model the strengths of individual teams and the most recent results are the most relevant.
- It gives the model matches that will happen in the near future so that the model can make predictions for them.
- In the case of simulating betting on the market odds it evaluates the bets given by the model and updates its current financial balance (bankroll) based on the results.

5.2 Evaluation metrics

In this section, we describe the evaluation measures and their characteristics.

5.2.1 Accuracy

Accuracy is one of the most basic evaluation metrics for classification tasks and should be included. One has to be careful using this for evaluating the quality of a model since there is a common mistake of neglecting prior probabilities of individual classes. A 97% accuracy does not mean anything if we do not have any information about the dataset. If a majority class has a prior probability of 97%, achieving such accuracy is trivial with a so-called naive model that simply predicts always the majority class.

5.2.2 Brier score

The Brier score is a scoring rule proposed in Brier 1950 that evaluates the quality of probabilistic predictions.

Most of the examined prediction problems have only two classes and for their evaluation, we use the Brier score in this thesis. The Brier score for binary events is defined as

$$BS = \frac{1}{N}\sum_{i=1}^{N}(y_{c,i} - p_{c,i})^2 \tag{5.1}$$

where N is the number of predicted instances, c is any of the two classes since the result is the same, y_c is a binary indicator for instance i, and p_c is the estimated probability of class c for instance i by the model. This rule can be understood as a mean squared error of the forecast.

5.2.3 Ranked probability score (RPS)

The ranked probability score was introduced in Epstein 1969. This metric is used for evaluating ordinal outcomes. With using this metric the whole estimated probability distribution is taken into account. In Hubáček, Šourek, and Filip Železný 2019 and Wheatcroft 2019, the authors pointed out that the distribution does not have to correspond to the outcomes being ordinal. Regardless of that, this metric is heavily used for evaluating "home/draw/away" predictions in soccer. Predicting "HDA" events is not the main focus of this thesis, but we include it for a more proper comparison of models and therefore we include this metric as well. The RPS for one sample is defined as

$$RPS = \frac{1}{r-1}\sum_{i=1}^{C}\left(\sum_{c=1}^{i}(y_c - p_c)\right)^2 \tag{5.2}$$

where C is the number of classes, y_c is a binary indicator of the correct class and p_c is the estimated probability of class c by the model. For evaluation of a quality of a model the average RPS over all samples is computed and the lower the score, the better.

5.3 Evaluated models

5.3.1 Baseline models

As our baseline models, we have chosen the original double Poisson and bivariate Poisson models. That is due to them being simple yet still very competitive (Ley, Wiele, and Eetvelde 2019) while using only the team IDs and resulting scores as an input. With our notation, they can be denoted by TID-DP and TID-BP with no hidden layers and no activation functions.

Furthermore, we include a naive model mainly for more proper evaluation of accuracy, but for better comparison of other metrics as well.

5.3.2 Our models

In the section 4.3 we have described the four fundamental types of model:

- TID + Poisson-based
- TID + classification
- STATS + Poisson-based
- STATS + classification

For each of these types, we have created many configurations. Each of them was evaluated on all types of events - "over 2.5", "both teams to score" and "home/draw/away'. The variants of individual fundamental types differ mostly in the number and dimensions of linear layers and activation functions.

The most common layer configurations we have experimented with are [], [4], [4, 4], [4, 6, 4]. [] denotes using no hidden layers at all. From some initial experiments, it seemed that more layers would make the models overly complex.

For most of the configurations, we have tested all of the mentioned activation functions - sigmoid, tanh, ReLU and also using no activation function.

5.4 Experimental design

Due to the large number of possible configurations of our models we had to perform a hyperparameter optimization. We could not evaluate all of them so we were observing the results of some and we have fixed some of the hyperparameters to values that seemed to work reasonably well. Even though, the number of configurations in the experiments is high and therefore we have separated a part of the data to which we refer to as a test dataset. The rest of the data, that was used to train and validate all of the models, will be referred to as a training and validation (T&V) dataset.

The test dataset will be used for providing an unbiased evaluation of the final model. The final model will be chosen based on the results on the T&V dataset. It is necessary because such an extensive hyperparameter optimization can have a similar effect to model overfitting on a training dataset. The hyperparameter values, that were picked because they had the

best results on the T&V dataset, might be suboptimal due to a chance and they could perform worse on another set of samples.

For the test dataset, we have chosen the last two seasons (2019/2020 and 2020/2021), because for those we were able to retrieve market odds for the events we focus on. Unfortunately, the coronavirus crisis largely impacted the development and consistency of the seasons. Moreover, the impact differs between leagues as some of them were only paused but finished a few months later, but others stopped and never finished. Therefore there are fewer matches and also we might expect interruptions and maybe even the absence of soccer fans in the stadiums to make the matches less predictable. Moreover, there are not many matches from season 2020/2021 in the dataset because the data were retrieved only a couple of months after the beginning of the season.

We should mention that not for all of the matches predictions were made. Each type of model is restricted by some rules that define which matches the model is not supposed to make predictions for. The models using team IDs as their input features are not predicting matches in the first 6 rounds of every season because the teams change a lot over the summer due to transfers of players. The models using the long-term form of teams are not predicting matches with a team that does not have a history of 60 played matches. Since there are new teams incoming every season from lower leagues that do not have any history, it filters a lot of the matches from the original amount. In the evaluation, we take into consideration only the matches predicted by all of the models tested. In the T&V dataset, it constitutes approximately 3500 matches, in the test dataset, it is almost 3000.

As an approach for the hyperparameter optimization on the T&V dataset, we have chosen a simple grid search. We have manually specified a subset of hyperparameter space that was exhaustively searched through. The hyperparameter spaces were not identical for different types of models due to their different characteristics of them and the fact that some of the hyperparameters are not present in other models. We examine the results of the evaluation on the T&V dataset from different perspectives, such as accuracy, Brier score, or ranked probability score. Then we choose the final models based on the results and this model is tested on the test dataset.

5.4.1 Betting against bookmaker

Because we were able to retrieve the odds for the final seasons, we have evaluated the final model also by betting on the market based on the predictions. In this thesis we work with decimal odds represented by a value $o_X \in \mathbb{R}, o_X > 1$ where X is the event we can bet on. For instance $o_{O2.5} = 2.2$ means that if we bet 1 unit on "Over 2.5" and we win, we receive 2.2 units

back, which makes a profit of 1.2 units.

We can easily derive a probability of event happening that corresponds the odds as $p_o(X) = \frac{1}{o_X}$. If for an estimated probability of the event X by our model $p_m(X)$ it holds that $p_m(X) > p_o(X)$, the expected value of this bet for us is positive, therefore we shall bet.

For the evaluation of the models on the market odds we use three different strategies - "unit-loss", "unit-win" and "unit-impact" (Barge-Gil and Garcia-Hiernaux 2020). While using any of these strategies, we bet b units on every odds for which $p_m(X) > p_o(X)$ while the amount b is determined by the strategy. The "unit-loss" strategy is trivial, it always bets 1 unit. While using the "unit-win" strategy, we bet

$$b = \frac{c_w}{o_X - 1} \tag{5.3}$$

where c_w is a constant, the potential profit of every bet we make. The "unit-impact" instead of holding constant the potential loss or win, it holds constant the difference between them, which is achieved by betting

$$= c_i \frac{o_X - 1}{o_X} \tag{5.4}$$

where c_i is the constant difference.

Furthermore from the odds, we have derived estimated probabilities of individual events so that we could evaluate all the metrics for the bookmaker as well for comparison with our models. Because the real market odds are not fair it holds that

$$\sum_{X \in \Omega} p_o(X) > 1 \tag{5.5}$$

where Ω is a set of all possible outcomes in the classical sense. As an estimate of the bookmaker's predicted probabilities, we have normalized $p_o(X)$ so that its sum over Ω is 1.

$$p_o^*(X) = \frac{p_o(X)}{\sum_{X' \in \Omega} p_o(X')} \tag{5.6}$$

Chapter 6

Results and discussion

In this chapter, we present the results of the experiments. Since there are many different types of models, multiple events, and two datasets, there are many outcomes and we highlight the most interesting insights. In the table 6.1 we can see the resulting values for all the metrics. The table is divided into three sections, each one for one type of events predicted - "BTTS", "O2.5" and "HDA". The left three columns containing numbers represent the results on the T&V dataset and the right ones represent the test dataset. The abbreviations ACC, Brier, RPS, xEnt stand for the metrics accuracy, Brier score, Rank Probability Score, and cross-entropy respectively. Furthermore, in the O2.5 section, an evaluation of the bookmaker's predictions is included. For easier readability, we have separated baseline models, Poisson-based models, classification models, and bookmaker by a thin line. A bold value represents the best result amongst our models for the metric in that type of event. The final configurations of the models can be found in table 6.2.

6.1 Baseline models

In the table, we can see the results of the naive model and the baseline models. We should mention that since "BTTS" is a binary classification problem and the accuracy for baseline models is below 50%, we can paradoxically swap the predictions and achieve accuracy of $1 - A$, where A is the original accuracy. But even though, the baseline models perform significantly worse on "BTTS" in all metrics than the naive model which is not very surprising. The DP model uses only a single value for each team for estimating the distributions underlying the resulting scores, whereas BP adds also a κ parameter representing some dependency between the distributions. Those models are very simple and indirect predictions of "BTTS" seem to be inappropriate.

Predicting "O2.5" was a little bit better, but the models only slightly outperformed the naive model on the test dataset. This might mean that the

models were learning some variation of the data, but were not able to predict the probabilities correctly enough.

On the contrary, the results of predicting HDA are much better for the baseline models. One of the reasons for evaluating the performance on the "HDA" was to be sure that the models are correctly implemented.

In order to explain the results, one can imagine the estimated probability distribution of score for a match as a matrix

$$\begin{bmatrix} P_{0,0} & P_{0,1} & \dots \\ P_{1,0} & P_{1,1} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{6.1}$$

where P_{G_H,G_A} is the estimated probability of resulting score $G_H : G_A$. The most probably correct explanation for the mentioned results is the size and shape of the parts of the matrix representing the particular events. It is very hard to predict all the probabilities very precisely, but it can be easier to model some parts of the matrix than others. We compute such a matrix from only up to three parameters and overestimation and underestimation of some probabilities are inevitable.

These models use mainly the strengths of both teams for constructing such matrix, and the results for events "home", "draw" and "away" share an important common characteristic. For instance, all the score results corresponding to the home team winning share the characteristics that the home team scored more goals, which is intuitively related to the strengths. Whereas there is no such characteristic for "BTTS" or "O2.5", that could be easily connected to strengths.

Surprisingly, the DP model always performed slightly better than the BP model. It might be due to the chosen copula function or the method of modeling its parameter κ and other approaches could be more suitable.

6.2 Poisson-based models

In this section, we describe the results of the TID-DP, TID-BP, STATS-DP, and STATS-BP models. When comparing the baseline models with our Poisson-based models, we can see that generalizing the idea of DP and BP into simple neural networks proved to be a significant improvement in predicting "BTTS" and "O2.5". Considering "BTTS", the models had troubles outperforming the naive model, but they are no longer noticeably worse.

In predicting "O2.5" the improvement is even more visible. The models clearly outperform both the naive and the baseline models. We can

BTTS						
	T&V			Test		
Model	ACC	Brier	xEnt	ACC	Brier	xEnt
Naive	53.10	0.2490	0.6912	53.44	0.2488	0.6908
DP	48.97	0.2525	0.6982	48.15	0.2522	0.6976
BP	48.68	0.2521	0.6975	49.36	0.2516	0.6963
TID-DP	53.31	0.2484	0.6899	53.89	0.2481	0.6893
TID-BP	53.54	**0.2476**	**0.6883**	52.61	0.2488	0.6908
STATS-DP	**55.00**	0.2484	0.6900	52.65	0.2491	0.6914
STATS-BP	53.17	0.2487	0.6906	52.99	0.2495	0.6921
TID-C	53.22	0.2485	0.6902	53.23	0.2489	0.6910
STATS-C	53.63	**0.2476**	0.6884	**55.41**	**0.2463**	**0.6857**
STATS+TID-C	53.36	0.2505	0.6945	53.96	0.2476	0.6884
O2.5						
	T&V			Test		
Model	ACC	Brier	xEnt	ACC	Brier	xEnt
Naive	53.62	0.2487	0.6905	52.92	0.2491	0.6914
DP	52.76	0.2465	0.6858	53.58	0.2476	0.6884
BP	52.67	0.2468	0.6865	53.27	0.2479	0.6890
TID-DP	55.09	0.2441	0.6811	56.55	0.2434	0.6798
TID-BP	56.60	0.2432	0.6796	55.72	0.2450	0.6833
STATS-DP	55.37	0.2453	0.6837	54.31	0.2468	0.6868
STATS-BP	55.64	0.2443	0.6814	55.14	0.2465	0.6860
TID-C	57.20	0.2430	0.6790	57.14	0.2447	0.6825
STATS-C	**58.20**	**0.2421**	**0.6772**	**58.01**	**0.2418**	**0.6767**
STATS+TID-C	55.19	0.2462	0.6857	57.14	0.2423	0.6776
Bookmaker				59.42	0.2391	0.6709
HDA						
	T&V			Test		
Model	ACC	RPS	xEnt	ACC	RPS	xEnt
Naive	45.26	0.2288	1.0656	41.34	0.2319	1.0799
DP	53.73	0.1967	0.9705	50.50	0.2094	1.0163
BP	53.48	0.1970	0.9719	50.40	0.2096	1.0165
TID-DP	**54.64**	0.1966	**0.9685**	49.91	0.2084	1.0123
TID-BP	53.22	0.2029	0.9880	48.15	0.2135	1.0277
STATS-DP	53.45	0.1990	0.9765	50.78	0.2054	1.0046
STATS-BP	**54.64**	**0.1965**	0.9693	**51.19**	**0.2048**	**1.0027**

Table 6.1: Experimental results of evaluated models on both T&V and test datasets

Model	Layers	Activation	FIFA score
TID-DP	[4]	None	-
TID-BP	[4]	Tanh	-
STATS-DP	[4]	Tanh	True
STATS-BP	[4, 6, 4]	None	True
TID-C (O2.5)	[4, 4]	Tanh	-
TID-C (BTTS)	[4, 4]	Tanh	-
STATS-C (O2.5)	[4, 6, 4]	None	False
STATS-C (BTTS)	[4, 6, 4]	Tanh	False
STATS+TID-C (O2.5)	[4, 4]	Tanh	True
STATS+TID-C (BTTS)	[4, 4]	Tanh	True

Table 6.2: Final configurations of individual types of models

say that the models were able to learn some variance of the data. There is a mostly moderate decrease in the quality of metrics between the T&V and test datasets which was expected given the number of configurations we have been choosing from. So far, we can see that TID models are more successful than STATS models. A reason for that might be that the input data are too wide and the model is too complex for predicting the strengths of the teams.

On the other hand, the improvement is not so visible in predicting "HDA". This is also the only type of event for which the STATS Poisson-based models achieve better results than TID.

In the "HDA" results we can also notice the largest decrease in performance between the T&V and test datasets - 3-5 percentage points difference for all the models. The accuracy of the naive model gives us the ratio of matches in which the home team has won since "home" is the majority class. That means that in the last two seasons there was a significant decrease in home teams winnings. These last two seasons were severely impacted by the coronavirus crisis and soccer fans were mostly banned from attending the matches. This could have reduced the effect of home advantage since most of the fans in a stadium were usually supporting the home team. This might be the reason for the decrease in quality of the models in the test dataset, therefore it does not necessarily mean the performance is poor.

6.3 Classification models

In this section, we describe the results of the TID-C, STATS-C, and STATS+TID-C models. It should be mentioned that classification models denoted by the same name predicting the different types of events are in-

dependent. For instance, the TID-C model predicting "BTTS" and TID-C model predicting "O2.5" are different models. On the other hand for each type of Poisson-based model, only one representative was chosen for all three types of events. For this reason, we do not include classification models predicting "HDA", they would be mutually independent with those predicting score-related events and thus less relevant for this work.

It is noticeable that the classification models perform generally better than Poisson-based models, which suggests that the classification approach is better for predicting these types of events despite its disadvantages.

Considering "BTTS", results of TID-C are very similar to the naive model's and it does not really perform better than TID Poisson-based models. The STATS-C on the other hand clearly outperformed all of the other models. Even though its accuracy was not the best on the T&V dataset, it showed to have potential with the results on the test dataset.

In predicting "O2.5", the STATS-C model stands out. It has consistently performed the best in all the metrics. Also, the other classification models have achieved better results than the rest of the models.

By combining the STATS and TID models, the resulting STATS+TID-C models have probably become too complex and the outcomes have worsened in comparison to the STATS-C models. There might be better ways of integrating those two types of models which is a potential for future work.

Furthermore, we can see that the decrease in quality of the classification models on the test dataset is negligible, sometimes the performance even increases.

Given the results of the STATS-C models, we can say that additional data can be very helpful in predicting score-related events, which is one of the main findings of this thesis. Furthermore, there is a large potential for improving the input data by adding or removing some features, or processing them in a different way.

6.4 Other insights

One can notice that none of the models were able to perform as well as the bookmaker did, which was not our primary goal. It is generally very hard to do that, but it shows that there is great potential for improving the models we have presented.

The "BTTS" event proved to be very difficult to predict. Baseline models were not even able to learn the majority class. Also, most of our models were not able to outperform the naive model. On the other hand,

even though our models were not as successful as the bookmaker in predicting "O2.5", the models have significantly outperformed both the naive and the baseline models and demonstrated that they have great potential.

Another interesting finding is that the best predictor for good results of predicting the score-related events on the test dataset proved to be the Brier score. The models that performed the best on the test dataset had always the lowest Brier score on the T&V dataset, and the pattern is visible for other models as well. The cross-entropy seems to be strongly correlated with the Brier score and proved to be useful as well. On the other hand, the accuracy was not a good predictor; there are models whose accuracy has dropped or even increased significantly between the datasets.

In the table 6.2 we present the final configurations chosen for each type of model based on the best results on the T&V dataset. For the TID models, it was counter-productive to employ more than two inner layers. Except for the STATS-DP, STATS models have proved to work best with at least two inner layers.

Some models worked better with the tanh activation function, for other models it was best to not include any activation function. We are aware that the resulting function making predictions becomes linear, but due to great performance, we have included it in this thesis. On the contrary, the sigmoid and ReLU have never outperformed the before mentioned options. The decrease in the quality of the models was significant.

Also for some models, it was beneficial not to use the FIFA player scores for making predictions. This might be due to the STATS models being already very complex and adding additional features while using a not so large dataset could be hurtful. There might be better ways of including such data in our models. For instance, the data might not have to be included in the first layer's input but could be added in a later stage of the computation. This is another potential for future work. Other interesting results might be provided by designing a model using only FIFA player scores as input data.

6.5 Profitability on the betting market

In this part of the experiments, we provide the results of evaluating the models by betting against the bookmaker on the "over/under 2.5" market. For comparing the models in terms of profitability we chose the "unit-win" strategy because of the best results, but the differences were minor and we will discuss that as well.

The results can be seen in the figure 6.1. The figure is divided into two charts, one depicting the running absolute profit of the Poisson-based

models, the second one represents the classification models. The naive and the baseline models are represented by dotted lines. In the charts, a 'day' represents a day in which at least one betting opportunity takes place.

It should be mentioned that it is generally very hard to make a profit by betting against a bookmaker in the long-term for two reasons. First is that the bookmaker's predictions are already very good, which we can see for example in the experimental results on our test dataset. The second reason is that the bookmaker's odds are unfair. For the purpose of this thesis, there is no need to understand how exactly the unfairness work. One can imagine it as a percentage fee for each bet, for instance, 4% of the bet amount. Let's have a model that is able to give better predictions than the bookmaker so that given fair odds, the model would generate an average profit of 1% per bet. Unfortunately given that we have to pay a fee of 4%, we lose an average of 3% on every bet.

The first thing to notice in the charts is that the baseline models generate similar losses to the naive model, approximately 115-135 units, which is not a large difference in this scale.

In the upper chart, we can see that none of the Poisson-based models was able to be significantly less loss-making than the baseline models. The smallest loss of 107 units was generated by the TID-BP model, whereas a Poisson-based model whose performance was the best on the test dataset, the TID-DP has finished with the largest loss amongst them. But once again, those differences are insignificant and a change could have played a role in this as well.

In the bottom chart, it can be seen that the STATS+TID-C model's performance is poor as well. On the contrary, the other classification models achieve far better results. TID-C finishes with a loss of only 79 units, which seems to be a great result in comparison with other TID models. But by far the best result was achieved by the STATS-C which finished the incomplete two seasons being profitable, +15 units to be specific, which is a very surprising result given the unfairness of the odds and the fact that on the test dataset the bookmaker has far better performance in the evaluated metrics.

There are several thoughts on how it is possible. The first one is simply a chance. The more models we evaluate, the larger is the chance that a profitable model occurs. It would be interesting to evaluate this model on a larger dataset, but acquiring more of such detailed data is problematic. One option is to wait several years, gather the new data using the code in this project, and re-evaluate the models. Another reason for this result being possible, as was described in Hubáček and Šír 2020, is the disadvantage of a "market maker". "Market maker" creates a large set of odds and our model as a "market taker" can bet only on those odds that it is confident enough to

do so. The third reason is that the model has learned to be better than the bookmaker on different matches. Nonetheless, the results of the STATS-C model are very promising.

We further provide a comparison of the models and the betting strategies in terms of "Return on Investment" (ROI). ROI is a metric measuring the profitability of an investment, more specifically the amount of return relative to the investment. In this case, we calculate ROI for a model as

$$ROI = \frac{P}{T} \tag{6.2}$$

where P is the absolute profit and T is the total sum of resources the model has bet.

This gives us another perspective on the comparison of the performance of the models. In the figure 6.2 we can see that most models achieve ROI of approximately from -7 to -8%. An interesting result is that the BP model, which is the worst-performing in terms of absolute profit, has actually better ROI than the TID-DP model. It is because even though the TID-DP has bet fewer resources, its loss was almost as large as the one of the BP model.

In the figure we can also see that the differences between the strategies were minor, only some models were noticeable more successful using the unit-win strategy. For a more proper comparison of the strategies, a significantly larger dataset would be needed.

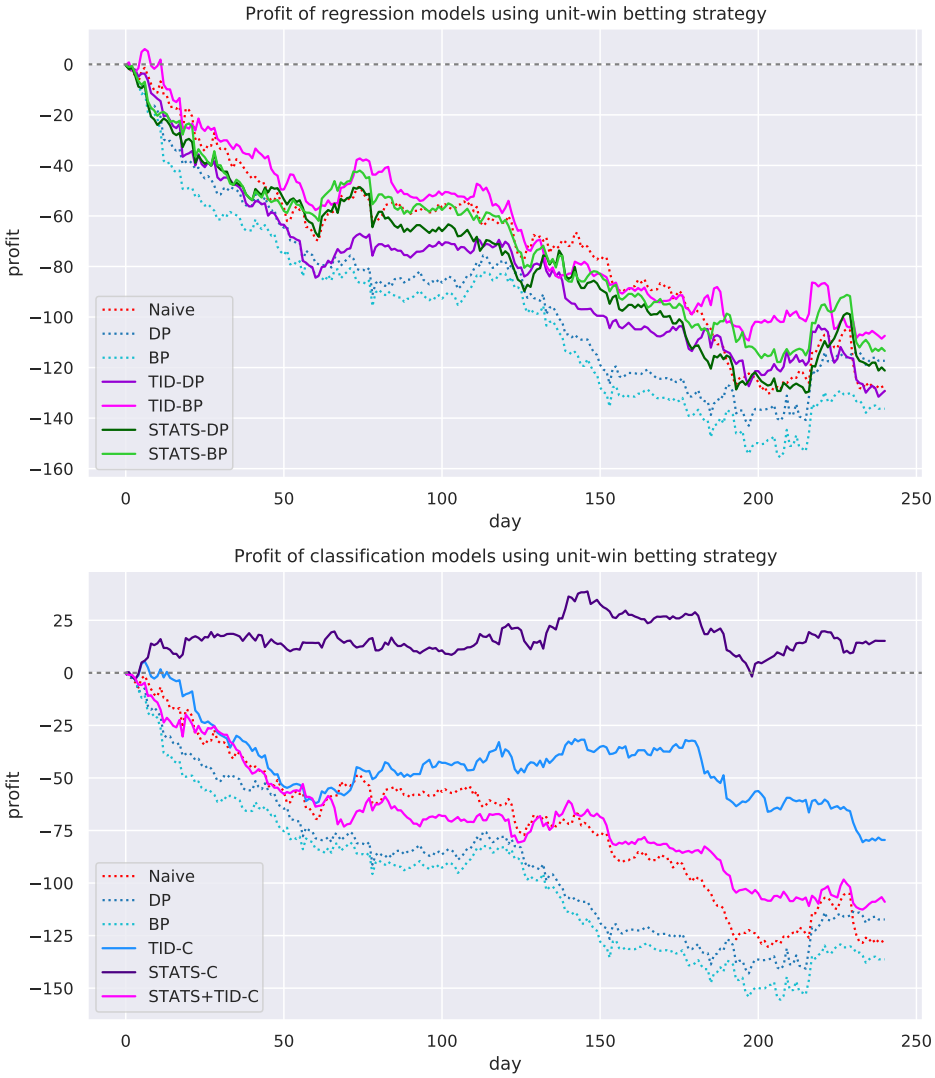


Figure 6.1: Profit of the evaluated models on the test dataset

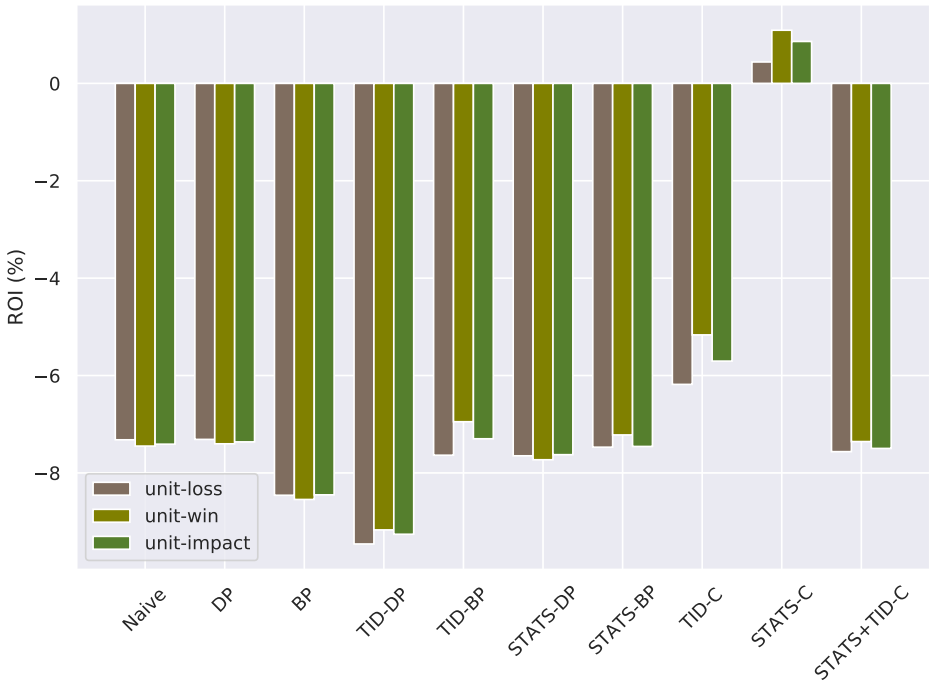


Figure 6.2: Comparison of ROI between the models and the strategies

Chapter 7

Conclusion

One of the goals of this thesis was to compare different approaches to modeling score-related events in soccer. As baseline models, we have chosen the original double Poisson (DP) and bivariate Poisson (BP) models whose competitiveness was shown in Ley, Wiele, and Eetvelde 2019. Our models were either Poisson-based or classification models. The Poisson-based models are a generalization of the DP and BP models. In this work, we have found out that even though the original DP and BP models are good for predicting "HDA" events, their performance in predicting "BTTS" and "O2.5" was very poor. Although our Poisson-based models did significantly improve the baseline models on "BTTS" and "O2.5" events, they have still performed worse than the classification models, which proved to be the best approach out of those evaluated. This is probably due to its advantage of estimating directly the probabilities of the individual events.

Another goal was to test the hypothesis that detailed data can be useful in forecasting score-related events. This was achieved by dividing all the types of models into two categories - TID and STATS models. TID models were taking as input for every match only the team IDs, date, and resulting scores, whereas the STATS models were receiving detailed statistics for every team optionally with FIFA player scores. On the test dataset, the STATS models have always significantly outperformed the TID models, therefore we believe that additional data provide a large potential for modeling score-related events.

We have also evaluated the models by betting on "over/under 2.5 goals" market odds. As expected, mainly due to unfair odds, most of the models were not profitable while Poisson-based models performed generally worse than classification models. Surprisingly STATS classification model happened to be profitable over the course of two incomplete seasons. There might be multiple reasons for this happening and more data is needed to properly evaluate the model's potential, but the results are very promising.

7.1 Future work

Based on the results in this thesis, many future work opportunities arise. For many of the steps in this work we have chosen only a small subset of possible approaches for testing, but the others could have worked better.

Retrieving more data samples for this problem would be a hard task, but the preprocessing of the data could be done differently. For instance, variances of individual statistics might be included along with the averages. Also instead of using league ID as a feature, league statistics such as goal average per match could be better utilizable by the model.

The models themselves are very modular. Because of the large number of possible architectures and configurations, it is more than likely that some would improve the current performance significantly. For instance, some data such as FIFA scores might be employed separately from the match statistics or even in later stages of computations. Also, some regularization techniques or different loss functions might be used. Our regression models were based on the Poisson distribution, but other approaches might be more suitable, such as a bivariate Weibull count model introduced in Boshnakov, Kharrat, and McHale 2017.

For further validation of our findings, an evaluation of our models on the upcoming seasons could be done.

Bibliography

[1] Andrés Barge-Gil and Alfredo Garcia-Hiernaux. "Staking in Sports Betting Under Unknown Probabilities: Practical Guide for Profitable Bettors". In: *Journal of Sports Economics* 21.6 (2020), pp. 593–609.

[2] Georgi Boshnakov, Tarak Kharrat, and Ian G McHale. "A bivariate Weibull count model for forecasting association football scores". In: *International Journal of Forecasting* 33.2 (2017), pp. 458–466.

[3] Marc Brechot and Raphael Flepp. *Dealing with Randomness in Match Outcomes: How to Rethink Performance Evaluation and Decision-making in European Club Football*. en. SSRN Scholarly Paper ID 3122219. Rochester, NY: Social Science Research Network, Sept. 2018.

[4] Glenn W Brier. "Verification of forecasts expressed in terms of probability". In: *Monthly weather review* 78.1 (1950), pp. 1–3.

[5] Anthony Constantinou. "Asian Handicap football betting with Rating-based Hybrid Bayesian Networks". In: *arXiv:2003.09384 [cs, stat]* (Mar. 2020). arXiv: 2003.09384.

[6] Mark J Dixon and Stuart G Coles. "Modelling association football scores and inefficiencies in the football betting market". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 46.2 (1997), pp. 265–280.

[7] Hph Harm Eggels. "Expected goals in soccer:explaining match results using predictive analytics". In: 2016.

[8] Edward S Epstein. "A scoring system for probability forecasts of ranked categories". In: *Journal of Applied Meteorology (1962-1982)* 8.6 (1969), pp. 985–987.

[9] Ondřej Hubáček and Gustav Šír. "Beating the market with a bad predictive model". In: *arXiv preprint arXiv:2010.12508* (2020).

[10] Ondřej Hubáček, Gustav Šourek, and F. Železný. *Score-based soccer match outcome modeling – an experimental review*. en. 2019.

[11] Ondřej Hubáček, Gustav Šourek, and Filip Železný. “Learning to predict soccer results from relational data with gradient boosted trees”. en. In: *Machine Learning* 108.1 (Jan. 2019), pp. 29–47.

[12] Dimitris Karlis and Ioannis Ntzoufras. “Analysis of sports data by using bivariate Poisson models”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 52.3 (2003), pp. 381–393.

[13] Dimitris Karlis and Ioannis Ntzoufras. “Bayesian modelling of football outcomes: using the Skellam’s distribution for the goal difference”. In: *IMA Journal of Management Mathematics* 20.2 (2009), pp. 133–145.

[14] Siem Jan Koopman and Rutger Lit. “Forecasting football match results in national league competitions using score-driven time series models”. In: *International Journal of Forecasting* 35.2 (2019), pp. 797–809.

[15] Pranesh Kumar. “Probability distributions and estimation of Ali-Mikhail-Haq copula”. In: *Applied Mathematical Sciences* 4.14 (2010), pp. 657–666.

[16] Christophe Ley, Tom Van de Wiele, and Hans Van Eetvelde. “Ranking soccer teams on the basis of their current strength: A comparison of maximum likelihood approaches”. In: *Statistical Modelling* 19.1 (2019), pp. 55–73.

[17] Michael J Maher. “Modelling association football scores”. In: *Statistica Neerlandica* 36.3 (1982), pp. 109–118.

[18] Edward Wheatcroft. “A profitable model for predicting the over/under market in football”. en. In: *International Journal of Forecasting* 36.3 (July 2020), pp. 916–932.

[19] Edward Wheatcroft. “Evaluating probabilistic forecasts of football matches: The case against the Ranked Probability Score”. In: *arXiv preprint arXiv:1908.08980* (2019).

Appendix A

List of attachments

- Source code for the work done in this thesis. We declare that all the source code was created by us.