Introduction to Data Science

Project report

Group C14

Members: Hans Matthias Andreas, Mathias Are

1st semester of 2020/2021

Section 1: Business understanding

Problem

In this project we attempt to address the following problem: are soccer match results predictable using machine learning (from now on: ML) algorithms on data about player individual quality and skill ratings and team synergy characteristics given in the European Soccer database in Kaggle.
We also aim to determine which of these approaches is more effective in terms of prediction accuracy.

Background

Our idea of this project came from our personal interest towards soccer and our curiosity of ML algorithms in betting scenarios and predicting the results of sport events. Our primary goal of this project is for us to get better acquainted with machine learning and data science in general. Creating a viable model for real world betting would be an added benefit to our interests.

Success criteria

We count the project successful once we have trained and optimized two ML models that consistently perform better than a coin toss, fixed the methods we plan to use in order to compare the two learners and finally produce replicable results of the comparison along with the source code in our Github page.

Available resources:

To achieve our goals we plan to use the aforementioned Kaggle database as our dataset for training and validating the models, python 3 and various libraries like numpy, pandas, scikit as our coding tools. We are currently hoping to be able to produce the results with our own personal laptops, but we are also considering using Colab or Google Cloud Computation when we deem our hardware unable to train sufficient models.

Requirements, assumptions, constraints

Results of the project need to be presented on december 17th. Thus, the models and the comparison should be finished a few days prior so that we have ample time to analyze the results of our project and to create a poster for the presentation.

Risks and contingencies

Our main concerns about the project are currently that the given dataset is not suitable for training these models, because of lack of data or lack of correlations between the predictable value and the attributes we use for prediction.

Also the current situation with Covid-19 might affect our ability to regularly contribute and advance our project, thus increasing the time cost of certain tasks which could lead to the project not being completely finished at the required deadline.

Terminology

Player and team attributes - Numerical values indicating the player's or team's proficiencies in specific aspects of the game. Data is obtained from the FIFA video game series.

Player and team ratings - Numerical values indicating the player's or team's overall proficiency that is calculated from their respective attributes. Data is obtained from the FIFA video game series.

Costs and benefits

We expect the time cost of this project to be around 70-80 hours in total divided to 35-40 hours per person.

The benefits would be our own personal development and some interesting results as a contribution to the Kaggle dataset thread.

Data mining goals:

1)Develop 2 models with different approaches described above.
2)Determine the most accurate learning algorithms for this problem.
3)Find correlations between player attributes and/or team attributes against the match result.
4)Create a detailed comparison between the models and elaborate on their viability and effectiveness.
5)Compare the more accurate model against betting results.

Data-mining success criteria: Built models predict match outcomes better than a coin toss would.

Section 2: Data understanding

Gathering data

Model A - using player attributes and ratings to predict match results

To train this model there are three main requirements: Firstly we need the results for each match and the teams that participated. Secondly, there needs to be a way to link the players that were on the field with each match or at least with the teams that were playing for each match. Final requirement is to have some kind of ordinal or numerical attributes that describe each player's ability to play football and a way to link these attributes to each individual player that played.

In my initial attempts to use the data I could load the sqllite file into the project, create working queries by using a separate sql query file and also by using python. Also by investigating contents of the tables and column names, it seems that the requirements stated above are indeed met.

For Model A we plan to use the tables : "Match", "Player" and "Player_attributes" and in case we are unable to link tables "Match" and "Player" without the "Team" table, then we include that as well.

Here is a table that describes which columns we plan to use for each database table:

Table name	**Relevant columns for model A**
Match	home_team_goal, away_team_goal - to determine winner [home_player_X1-home_playerX11] - to link players to matches [away_player_Y1-away_playerY11] - to link players to matches
Player	id, player_api_id- to link player to his attributes
Player Attributes	player_api_id - to link attributes to player Specific attribute columns are determined later when we investigate/determine correlations.

Model B - using Team attributes and ratings to predict match results

On the surface, this model is similar to model A, first we need the match results and the teams that played in them from individual matches. Then we need to link the match results to the corresponding teams. Finally, we need ordinal or numerical attributes to describe the various strengths of each of the teams.

For model B we plan to use tables : “Match”, “Team”, “Team_attributes”. In the case where we are not satisfied with this model only using those three tables, we might make use of the “Player_attributes” table as well.

Here is a table that describes which columns we plan to use for each database table:

Table name	**Relevant columns for model B**
Match	home_team_goal, away_team_goal - to determine winner [home_team_api_id] - to link teams to matches [away_team_api_id] - to link teams to matches
Team	id, team_api_id- to link team to it’s attributes
Team Attributes	team_api_id - to link attributes to player Specific attribute columns are determined later when we investigate/determine correlations.

Describing Data

Data description report

Model A

As mentioned before, the data necessary for this model can be found from tables Match, Player and Player Attributes of the Kaggle European Soccer database. The data is accessible via SQL queries and Python sqllite3 library and is also convertible to CSV format. In general all the columns needed for this data are numeric: there are assigned ID-s for each match and player and also almost all of the player attributes are numeric or in few cases categorical.

Model B

Like in model A, the necessary data can be found in the Kaggle European Soccer database and is accessible using the aforementioned methods. In addition to the Match table, this model uses the Team and Team Attributes tables. Each match and team has their corresponding ID and majority of the features in the Team Attributes table has numeric values.

Data exploration report

Model A

To begin with the table Match, it has 25979 rows of data and 115 columns. The first 10 rows are id-s for league, country, fifa-api and most important information like the final score result and date the match was played. Columns 11-76 are for player ID-s that were on the starting 11 or on the substitute bench. Then there are a few general statistics like shots on target, corner count, cross count and possession percentage, and in the end of the table there are betting odds of different sites. In terms of distribution the matches are from 11 leagues divided quite evenly: 1700-3400 games per league.As we tried to find unreliabilities with the table we found that there are quite a lot of null values in some of the columns, but for the relevant columns there were not too many. For each player ID in the table there were about 2000 null values and fortunately they all overlapped almost perfectly, which means most of the data is usable in terms of associating each player with the specific match.

For the Player table it was all quite simple and clear. There are 11060 rows and only 7 columns in the table: 3 different IDs (probably only the regular id or player_api_id is needed for the model), name and birthday in text values and height and weight as integers. There are no empty fields in this table, which means it is much easier to use it in the latter stages of data processing.

Arguably the most important table for model A, the player_attributes table has 42 columns, three ID values as the other tables discussed so far and 39 columns for describing each individual player's attributes and rating. 36 of the player features columns are numeric values and describe the player's ability in a scale of 1-100. The true distribution is usually around 30-99 where the mean is mostly at 65-68. As we further analyzed the player attributes table we found surprisingly low null value count in the whole table: there is a set of 30 attributes that only have 836 null values and about 8 columns have close to 3000 null values, (in both cases the null values overlap on the same row as in the Match table) which is still low when compared to the total of 183978 rows in the table.

Model B

From the Match table, the important columns are team_api_id's and team_goal's which we use to get match results between teams. In the Team Attributes table which has 1458 rows and 25 columns, most of the features there are both categorical and numerical values which describe how often certain scenarios like passing occur or how aggressive the attackers are, etc. The Team table just contains information like team names and their shortened names in addition to their respective ID's and contains 299 rows.

Data quality report

In conclusion to the sections of data description and data exploration the Kaggle European Soccer Database is indeed sufficient for training these models in our judgement. Although there are still further steps of data processing needed, where we get rid of null values, condense the relevant data and use the ID values to join the tables that each model uses.

Section 3: Project plan

Task	**Description**	**Time cost(h) per person**	**Category**
Import data	Import data from Kaggle.	1	data preparation
Create queries	Create queries to access data in different tables	2.5	data preparation
Clean data	Remove nan values, determine if data is balanced, balance if necessary.	2.5	data preparation
Construct necessary data	Create csv files that include only the data required for the models in a format that is usable for training the models.	4	data preparation
Distribute data into training/test/val.	Determine the sizes of train/test/val sets and separate the data in random fashion. Use cross-val. if needed.	2	data preparation
Build model A	Build model from player data	3	modeling
Build model B	Build model from team data	3	modeling
Optimize models	Choose the most effective algorithm and optimize hyperparameters.	6	modeling
Validate models	Use the validation set to determine the final accuracy/auc score of models.	2	modeling

Create comparisons	Create methods to compare the models accuracy, speed, add roc curve, chart performance related to dataset size etc.	4	Evaluation
Assess comparisons	Determine if the results of the comparisons are applicable in real life scenarios.	1.5	Evaluation
Derive conclusions	Derive conclusions and write about it	1	Evaluation
Format the results and code	Make the code easily usable and understandable for others.	5.5	Deployment
Publish	Create a poster where the results are visualised and easy to read and publish it	4	Deployment