UNIVERSITY OF TARTU

Field of mathematics and statistics

Institute of computer science

Informatics curriculum

Rasmus Lellep, Jürgen Leppsalu

Entropy of Football Statistics

Report 1

Tartu 2019

Task 1: Github repository

Our project can be found and monitored at the GitHub. An invite to collaborate has been sent out to all of the instructors. In case any mishaps a link of the repository is provided down below

GitHub repository: https://github.com/leppsalujyrgen/Entropy-of-Football-Statistics

Task 2: Business understanding

A sports betting company wants to take on live betting but they do not have an algorithm for predicting the odds yet. In order to build a live predictor, a base of offline predictor should be built. If the algorithm predicts the winning team/draw correctly at least 85% of the games, it is considered a valid base and can go on to be developed to a more specific predictor.

The algorithm code will be written in python 3 and the work will be done by 2 people. The data consists of players' statistics and the scores in every game of a football league for a couple of years. In order to have a good predictor, it has to find the most relevant features in a game that do not relate directly to the score by training on past games. Estimated completion is in 10 business days. The data could become a problem since there may not be enough to train the algorithm to the required accuracy. Should this be the reason for low accuracy, it is a possibility to find additional data to train with (not preferred since probably has different gathered statistics and could take a lot of extra work) Also, adding support for predicting with fewer columns of statistics could delay the completion since the existing testing games' data could have fewer columns.

Ideally, the betting company gets a new base predictor for their upcoming live betting feature and since the project's cost is practically non-existent, there is nothing to lose.

In order to get the winner prediction to best possible accuracy, it would be best if the predicted teams' scores were the same or close to the real score. It means that the algorithm does not predict the winner/draw right away but rather predicts both teams' score and then compares them.

While the predicted winner/draw is a choice out of 3 and therefore a lot easier to perfect than a combination of two teams' goal scores, the goal is to predict the correct score of every game at least in 70% of the instances. It might seem that 70% is quite a low prediction accuracy. Actually, there are a number of different possible outcomes which could have almost equal probabilities so the 70% might actually be too high, for example, provided neither team scores more than 3 goals, there are already 16 possible scores of which several could have very similar probabilities.

Task 3: Data understanding

Gathering data

For the predictor to predict correctly we need to have lots of data. One of the best free datasets about English first division football is a Fantasy Premier League repository on GItHub called Fantasy-Premier-League, that has every players' every game performance described with an expansive list of attributes. Now, one might notice the word "Fantasy" appearing, which might lead some to think that the data is fictional, however, this is not the case. This repository gathers data from Fantasy Premier League API which is hosted and maintained by the Premier League and the only difference between Fantasy and non-Fantasy dataset would be extra columns like current_cost, transfers_in and transfers_out which are purely related to the Fantasy Premier League game. Other statistics like passes_attempted and open_play_crosses are real-life statistics recorded by the Premier League Association itself.

There are three main selection criteria we will be using to determine whether an attribute should be in our dataset.

1. An attribute should directly reflect real-life events

2. An attribute should not be in 1-to-1 correspondence with the goal count. For example "assists" attribute is the prime example of an attribute we can not use because every assist corresponds to a goal.

3. An attribute should at least potentially be affecting the result of the game. Attributes like "round" and "date" for example do not satisfy this criterion.

Describing data

The data we have of player performances is in CSV file format. Each file has 500 rows. one row for each player that made an appearance this game week. Each file also has 55 columns out of which 24 match the criteria for our dataset. There are 38 game weeks in total in each Premier League season and so there is 38 of these files in each season. Currently, we have the data for 3 seasons: the 2016/17 season. the 2017/18 season and the 2018/19 season. The ongoing 2019/2020 season can not be used as the data for each game week has changed format to be more general (less columns).

Exploring data

The data files have the following columns:

- name (String) - full name of the player,
- assists (Integer) - number of assists a player made in one game,
- attempted_passes (Integer) - number of passes player made,
- big_chances_created (Integer) - number of chances created where the player's teammate had a clear opportunity to score (chances where a player has only the goalkeeper to score past or low range efforts that have a clear path to goal),
- big_chances_missed (Integer) - number of chances missed where the player had a clear opportunity to score (chances where a player has only the goalkeeper to score past or low range efforts that have a clear path to goal),
- bonus (Integer) - Fantasy Premier League bonus points for this round's performance,
- bps (Integer) - Bonus Points System or Fantasy Premier League's points system that determines which player should be awarded bonus points for their contribution this round.
- clean_sheets (Binary) - Determines whether the player's team conceded goals or not.
- clearenced_blocks_interceptions (Integer) - the amount of clearances (times where player kicks the ball away from the goal he is defending), blocks (times where player stops the opposition from shooting towards goal by blocking it with his body) and interceptions (times where a player intercepts a pass from one opponent to the other)
- completed_passes (Integer) - the amount of attempted passes that reached player's teammate
- creativity - a metric that illustrates how many much threat did the player's passing and dribbling create
- dribbles (Integer) - the amount of times a player beats/surpasses an opponent while maintaining possession.
- ea_index (0) - Always zero, no intended purpose specified anywhere

- errors_leading_to_goal (Integer) - The number of times possession was lost by the player that directly led to the opponents scoring
- errors_leading_to_goal_attempt (Integer) -
- fixture (Integer) - identifier for each game
- fouls (Integer) - amount of unlawful actions during the match that the referee penalized.
- goals_conceded (Integer) - the amount of goals that the player's team allowed the opponent to score.
- goals_scored (Float) - the amount of goals that the player scored
- ict_index - Influence creativity and threat index is a combined metric that is calculated based on a player's performance.
- id - unique identifier for every player
- influence (Float) - metric that describes how much influence a player had on the team's performance.
- key_passes (Integer) - number of passes that led to a shot on target
- kickoff_time (Date) - datetime for the game's kickoff that the player took part in
- kickoff_time_formatted (String) - datetime of the game that the player played in more readable format
- minutes (Integer) - the number of minutes the player played this fixture
- offside (Integer) - the amount of times the player was determined to be offside by the match officials
- open_play_crosses (Integer) - the amount of crosses attempted by a player that does not include set-pieces like corners and free-kicks
- own_goals (Integer) - goals scored into your own goal
- penalties_conceded (Integer) - number of penalties that the player managed to give away to the opponents
- penalties_missed (Integer) - number of penalties missed by the player

- penalties_saved (Integer) - number of penalties saved by the player
- recoveries (Integer) - an event in which a player gains possession after control of the ball has been lost by the opposition
- red_cards (Binary) - determines whether the match officials chose to dismiss the player for unlawful actions
- yellow_cards (0-2) - determines how many warnings did the player receive during a match (players are dismissed after two warnings /two yellow cards)
- round (Integer) - round identifier for each game
- saves (Integer) - number of saves the player made (only applies for goalkeepers)
- selected (Long) - Fantasy Football statistic that shows how many people selected this player for their Fantasy Premier League team
- tackled (Integer) - number of times the player was tackled when he was in the possession of the ball.
- tackles (Integer) - number of times the player successfully tackled an opponent who had possession of the ball
- target_missed (Integer) - number of times the players shot failed to hit the target or goal
- team_a_score (Integer) - number of away team goals
- team_h_score (Integer) - number of home team goals
- threat (Float) - fantasy football metric that shows how
- total_points (Integer) - Fantasy Football points that are distributed for each player each game week based on performances
- transfers_in - Fantasy Football metric that shows how many players decided to choose this player for their fantasy football team

- transfers out (Integer) - Fantasy Football metric that shows how many players decided to replace this player from their fantasy football team
- transfers_balance - transfers_in minus transfers_out
- value - the cost of the player in the fantasy premier league market (fluctuates based on performances)
- was_home (Boolean) - determines whether the player played for the home or away team
- winning_goals (Binary) - determines whether the player scored a game-winning goal

Initial inspection shows no signs of data quality problems. However, column value ea_index seem to have no meaning as it is always zero. Other columns seem to be fine.

Verifying data quality

We are going to be using a GitHub repository (https://github.com/vaastav/Fantasy-Premier-League) that gets its statistics from the Fantasy Premier League API (https://fantasy.premierleague.com/api/bootstrap-static/) and the API is for the Premier League's Fantasy Football website (https://fantasy.premierleague.com/) which is hosted and maintained by the Premier League Association. As the league we are focusing on is the Premier League, this is the most reliable source for statistics since it is the most well documented and watched football league in the world.

Task 4: Planning the project

Tasks:

1. Choosing the needed data files and cleaning them, contributions - 5 hours per person
2. Merging selected files' data, contributions - 4 hours per person

3. Exploring the data’s correlations, contributions - 7 hours per person
4. Try and development an algorithm for predicting the end result, improving the algorithm if needed, contributions - 5 hours per person
5. Report the findings and accuracy of the algorithm, conclusion, contributions - 6 hours per person

Merging in this case means connecting players' data (features) to game result data (labels).

Exploring the correlations is especially important here to get specific indicators for predicting each team's score. For example a team might have a much higher win possibility when their star player is having a good day than when any other player is performing.

3 hours of extra work time is reserved for possible delays.