**Ranking System Prediction Based on English Premier League**

**INFO-6205 Program Structure & Algorithm Final Project**

**Sichen Zhao, Xianling Zeng**



**Background and Statement**

Since EPL have announced that suspended all match in response to the coronavirus pandemic, this season has a high probability that cannot complete as normal. Under this scenario, our team plans to predict the final rank of this season in EPL, to predict the champion, UEFA Champions League team, and the relegation team.

**Aim and Expected Result of Project**

Our desired output should contain three parts:

First is the , where is the probability that (home team) would beat (away team) if they met in a head to head.

The second part is our simulation result for remain matches in this season. There are total of 91 uncompleted matches in this season. As we know the win rate when two teams met, we will simulate the remain match and get the result.

The third part is the final ranking prediction of this season based on simulation result. We will calculate the uncompleted match result and add point for each team. Finally, the table should similar to the real-world rank table. Each column is Team name, win, draw, lose and final points, respectively.

**Datasets**

**Most recent 3-years match history**: our team consist of two people, so we decided to use most recent 3-years match history as our data. However, there are some pros and cons when we used it as our source.

Pros: Most recent data have more correlation current season’s rank. It is easy to explain that existing match result in this season will most influence the final rank of this season. A ten-years ago match history has few influences on the prediction. Therefore, only using 3-years match history will make the prediction more accurate.

Cons: The data only has 3-years match history, technically two teams will only have at most 6 matches with each other (3 home and 3 away). The prediction will be very inaccurate if one team lost all home match or all away match. Take Man City and Liverpool as example, in recent two years, there are only 2 matches that is Man City as home and Liverpool as away because they didn’t match in this season due to coronavirus. Since Liverpool lost these 2 matches, as the result, the algorithm finally calculate = 93%, which means Man City has 93% of probability win the match in home! Based on the realistic, both Many City and Liverpool and top-level team and the Man City’s win rate should not too high. This cons ultimately influenced our result and also forced our team to do some improvements. We will be mentioned it in the improvement section.

**Remain matches in 2019-2020 Season:** For each season, every team should have 38 matches. Currently most of teams already finished 29 matches. There are total of 91 uncomplete matches in EPL. We create a csv file for those matches, and application will simulate uncompleted matches result, then add them to the current rank to generate the final rank.

**Current Rank Table:** 2019-2020 Season already finished 29 matches for each team. Therefore, we need the current rank of EPL. According to the data, the first place is Liverpool with 82 points, the second place is Man City with 57 points and so on.

**Implementation**

**Method Selection:** At the first, we have two options for normal distribution simulation:

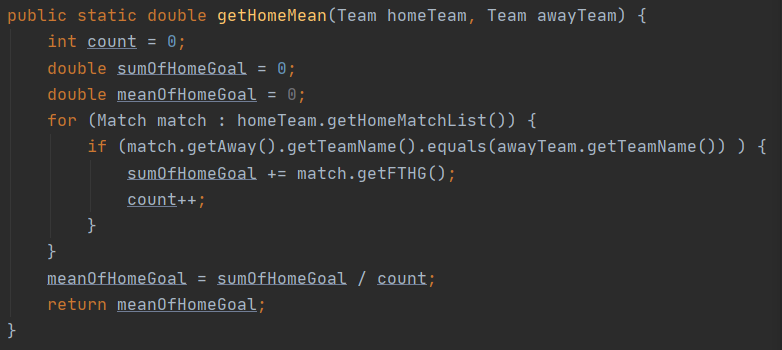
1. Use the number of goals of the team (FTHG, FTAG) to fit the normal distribution

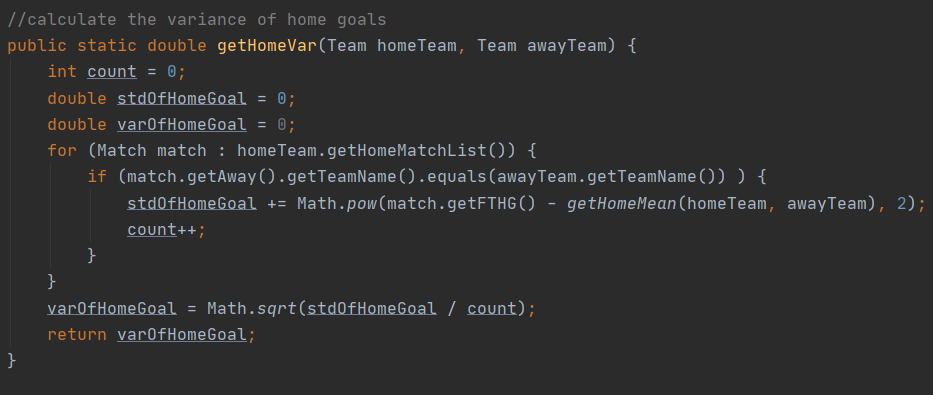
2. Use the goal difference (GD) between two teams to apply the normal distribution

During the process of many practices, we found that Method 2 has certain flaws. Due to the type of the result simulated by the java normal distribution model is floating, and eventually we need to round up, so when we simulate a goals difference around 0 (such as 0.2, 0.3, 0.4), we will round it to 0. But this will destroy the probability of draw.

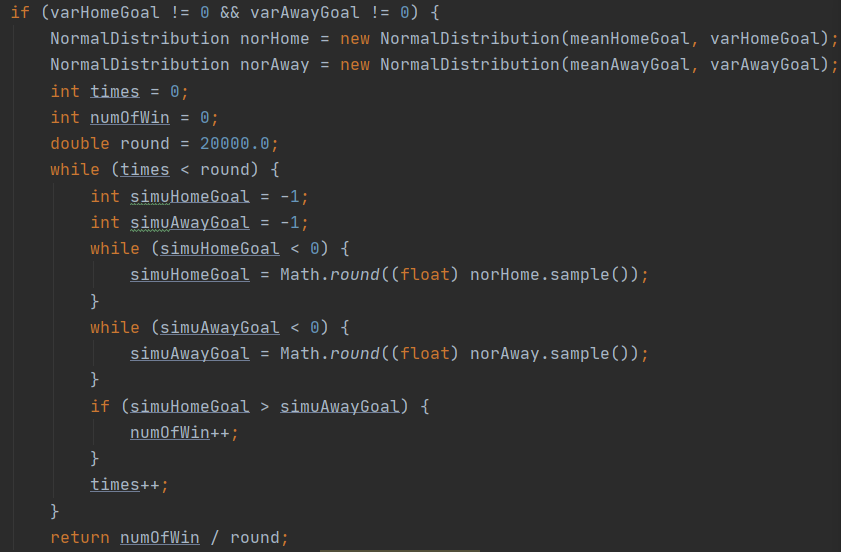
So Use the number of goals of the team (FTHG, FTAG) to fit the normal distribution is a better choice.

**Modeling:** The match score would follow a normal distribution. Therefore, we need to calculate the mean goals and standard deviation in each match. We consider home advantage would be a significant factor which affect the match result as well as it happens in real world, so we calculate the home mean goal and away mean goal separately.

Mean:

Standard Deviation

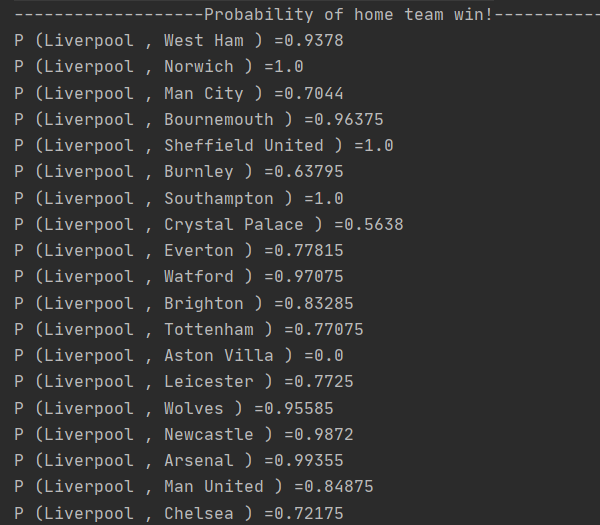
As we get mean goal and standard deviation of each team, we could develop the normal distribution for each team. However, before putting it into normal distribution, we have to check if the standard deviation is not equal to 0. For instance, if a team keeps same score during entire season, such as 0:1 in all matches, it won’t have a standard deviation. Even if it seldom happens but we still need to eliminate this scenario in advance. Once this scenario happened, we will use its own mean goal instead of using its standard deviation. To ensure the accurate of prediction, we run it for 20000 times and get the average number.



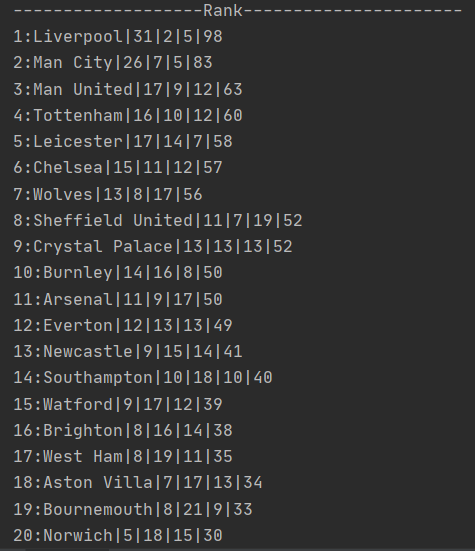
**Match Result Simulation:** By importing the existing library called: Apache Commons Math 3.6.1, we could easily calculate expected goal of each team for uncompleted match. Input mean and standard deviation, the library will apply into normal distribution and get the result. We will use its own mean goal when standard deviation equals 0. When home goal greater than away goal, home win.

**Result Generation:** As we get the uncompleted match result, we will add points for each team based on football rules, add 3 for win, 1 for draw and 0 for lose. Add with the existing rank, we will generate our predicting rank.

**Output:**

, left team means the team is home, and right one is away team.

**Uncompleted Match Result Prediction**

**Final Rank**

Based on output, our team predicts the champion is Liverpool. Four teams which are qualified to attend UEFA Champions League are Liverpool, Man City, Man United and Tottenham Spurs. Three relegation teams are Aston Villa, Bournemouth and Norwich.

**Improvement**

During the project, we found there are some defects could be improved.

The first one is limited data. 3 years data is a relatively low volume to predict. However, use match history from very long time ago would also make prediction inaccurate. It sounds like a paradox, and we think this problem will be solved if two team could have more matches in one season. Another potential method is expanding 3 years to 5 years but adjusting the weight average for each season. The most recent has more weight and earliest season has less weight.