INFO 6205

EPL Ranking System Report

Spring 2020

Section: 03

Team member: Chao Ma, Yunwen Sun, Lu Liu

Due Date: April 17, 2020

Introduction

The Premier League is one of the most popular sports leagues in the world. It broadcast in 212 territories to 643 million homes. Also, it has about 4.7 million people potential TV audience all over the world. In this project, we developed a ranking system which is able to predict the result of competition and evaluate the ranking of each season in English Premier League. In order to implement our goals, we use Poisson Distribution and Random in the project. For the implementation part, using two maps to save home teams and away teams data (five competition seasons from 2015 -- 2020). Then calculate the scoring rate for each team, and find the probability score of two teams. Finally, according to the predicted situations ranking 20 teams.

Problem Statement

In this project, we designed a flexible predict model for EPL. Because of COVID-19, the remaining matches of Premier League in the 2019 -- 2020 season have been postponed. Our model focus on solving the following two problems:

- Complete the simulation of the 2019 -- 2020 season. Our ranking system needs the full season data, so it is necessary to simulate the results of remaining matches.
- Predict the game results and ranking teams.

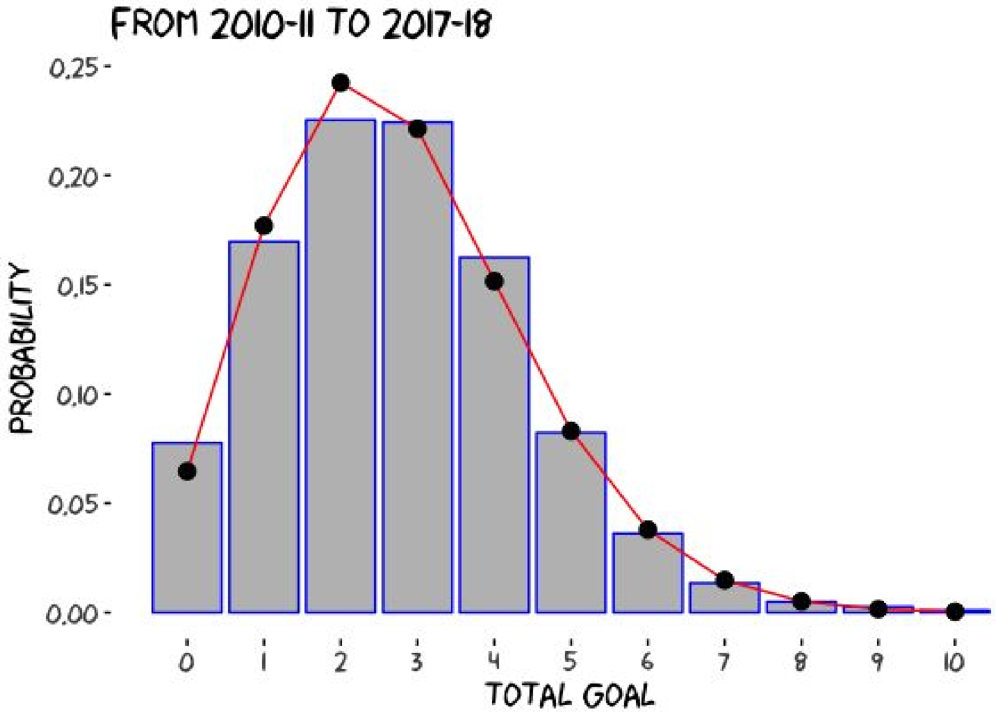
Methodology

Probability Distribution

A probability distribution is a mathematical function that provides the probabilities of occurrence of different possible outcomes in an experiment. In this ranking system, we assume each game is an independent event, which means the result of the current round of team A will not influence its next game.

We use the probabilities of goal/game of the two teams to simulate the result. For example, if the last 100 game result of team A is 0 goal(30 games), 1 goal(50 games), 2 goals(15 games) and 2+ goals(5 games), then the goal probability of team A will be 0 goal(30%), 1 goal(50%), 2 goals(15%) and 2+ goals(5%). Therefore, we can get a discrete probability distribution of goal/game for team A. A similar example bar chart is shown below.

In this project, we use the dataset of the last 5 seasons to calculate the goal probability. The performance of a team at home may differ a lot from away games. So the goal probability of home games and away games are separated in the system and used accordingly in the prediction.



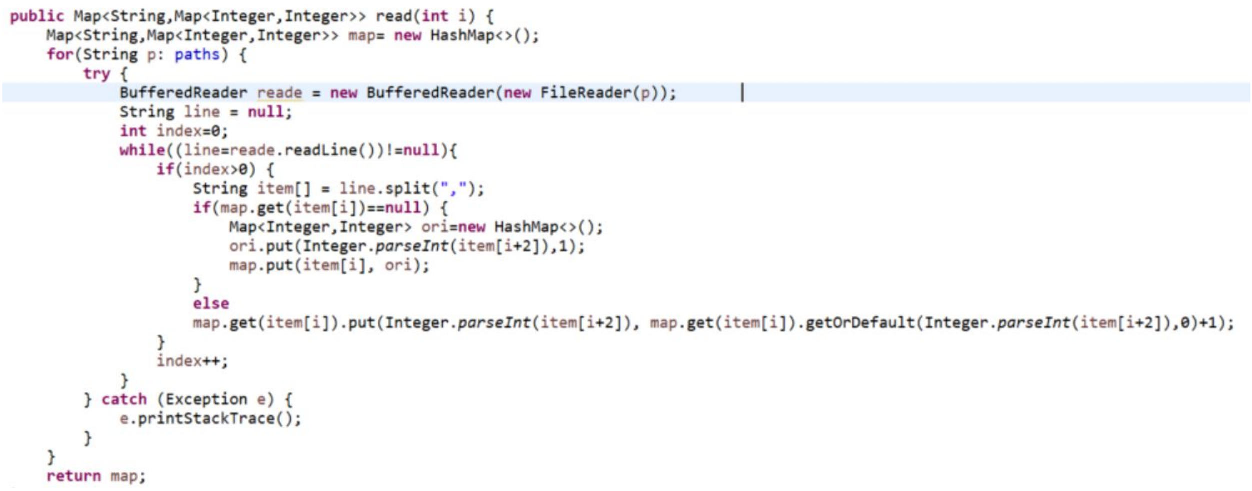
Generate Random Result

The result of a game is not fixed and that is the beauty of soccer. Even the underdog has the chance to win the game. Therefore, the way we use to generate a game result is based on a random generator instead of the most possible result. When predicting, a number N from 1 to 100 will be generated randomly. Assume that team A has 50% as the probability of winning, 30% to lose and 20% to draw. Then if N is between 1 and 50, team A wins. If N is between 51 and 80, team A loses. And if N is larger than 80, the result will be a draw.

Implementation

Collect and Process Data of Previous Seasons

Get the data in the past 5 seasons and save them into two maps as hmap and amap. For each team and its home games, select the goal of the team as key, the number of the goal appears as value to create a map. Put all of the teams and their maps into *hmap*. Similarly, generate the maps of teams running away games, and put them into *amap*. So that we can know each team, the goals they have achieved in past seasons and the number of times they achieved each goal which is saved by home game and away game.

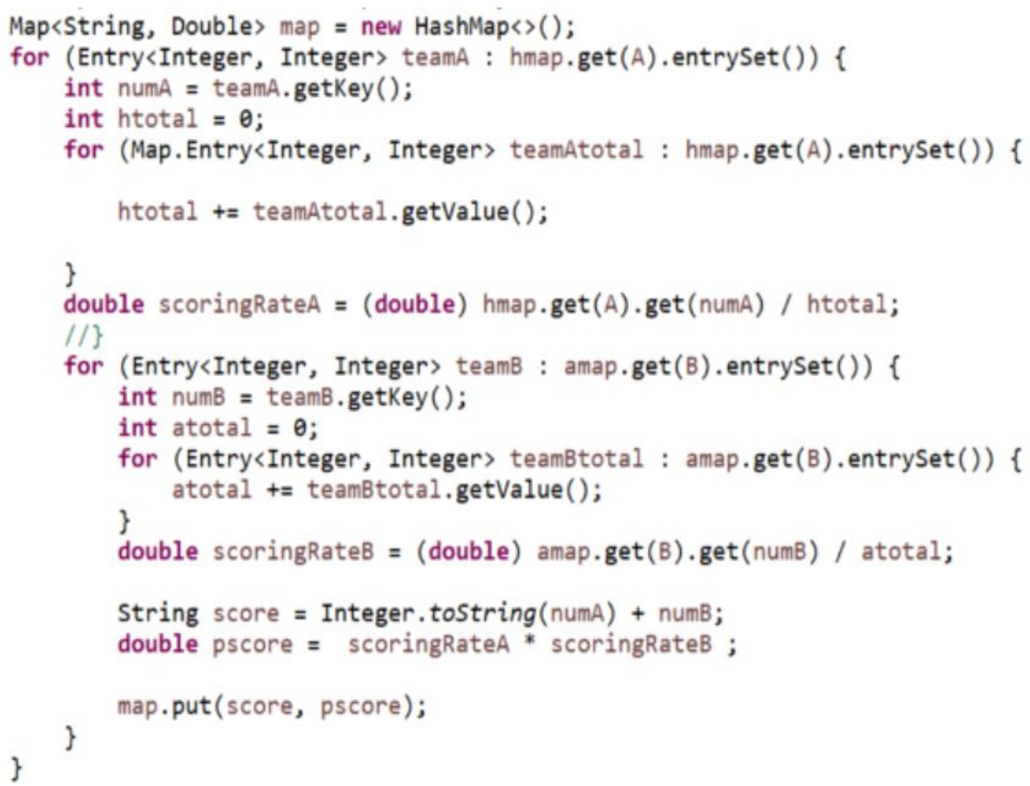


```
public Map<String,Map<Integer,Integer>> read(int i) {
    Map<String,Map<Integer,Integer>> map= new HashMap<>();
    for(String p: paths) {
        try {
            BufferedReader reade = new BufferedReader(new FileReader(p));
            String line = null;
            int index=0;
            while((line=reade.readLine())!=null){
                if(index>0) {
                    String item[] = line.split(",");
                    if(map.get(item[i])==null) {
                        Map<Integer,Integer> ori=new HashMap<>();
                        ori.put(Integer.parseInt(item[i+2]),1);
                        map.put(item[i], ori);
                    }
                    else
                    map.get(item[i]).put(Integer.parseInt(item[i+2]), map.get(item[i]).getOrDefault(Integer.parseInt(item[i+2]),0)+1);
                }
                index++;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return map;
```

Predict the Result of Two Teams

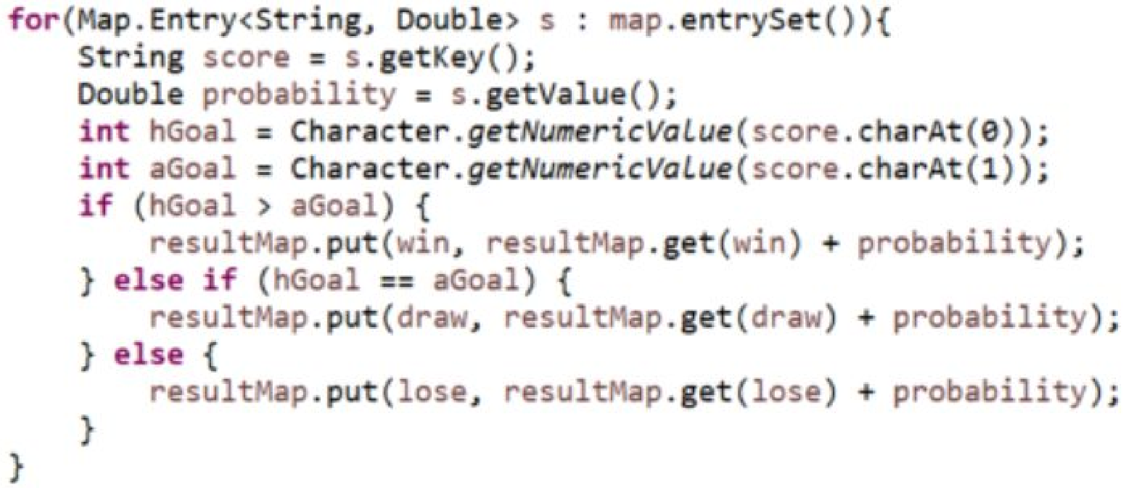
Given a game including the home team and away team, we can predict the result by following steps:

1. Calculate the goals and their probability that the home team may get. Goals can be read in the hmap, probabilities result from the frequency that the goal shows in all its home games. Use the same way to calculate the goals and their probability that the away team may get.
2. Predict the score of the game. List all the permutations of scores that may appear in the game and calculate each probability by multiplying the goals probability of both sides. Based on the dataset, we assume that the score will be less than 10.

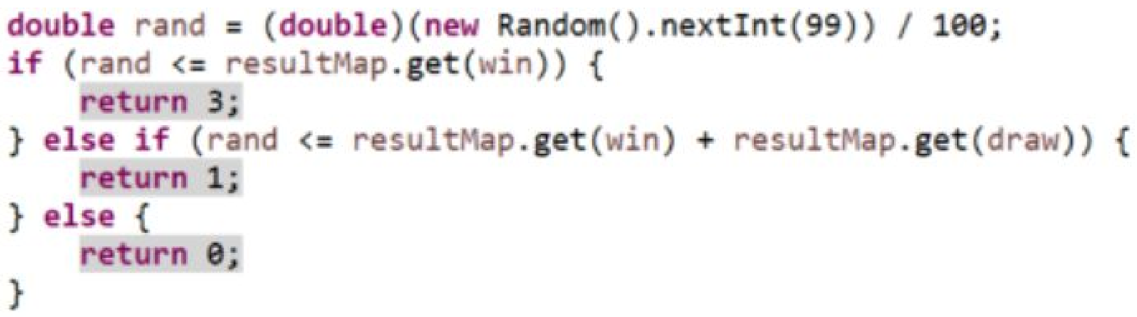


```
Map<String, Double> map = new HashMap<>();
for (Entry<Integer, Integer> teamA : hmap.get(A).entrySet()) {
    int numA = teamA.getKey();
    int htotal = 0;
    for (Map.Entry<Integer, Integer> teamAtotal : hmap.get(A).entrySet()) {

        htotal += teamAtotal.getValue();

    }
    double scoringRateA = (double) hmap.get(A).get(numA) / htotal;
    //}
    for (Entry<Integer, Integer> teamB : amap.get(B).entrySet()) {
        int numB = teamB.getKey();
        int atotal = 0;
        for (Entry<Integer, Integer> teamBtotal : amap.get(B).entrySet()) {
            atotal += teamBtotal.getValue();
        }
        double scoringRateB = (double) amap.get(B).get(numB) / atotal;

        String score = Integer.toString(numA) + numB;
        double pscore =  scoringRateA * scoringRateB ;

        map.put(score, pscore);
    }
}
```

3. Divide all of the score probabilities into “win” “draw” “lose” and attach them with points 3, 1, 0.



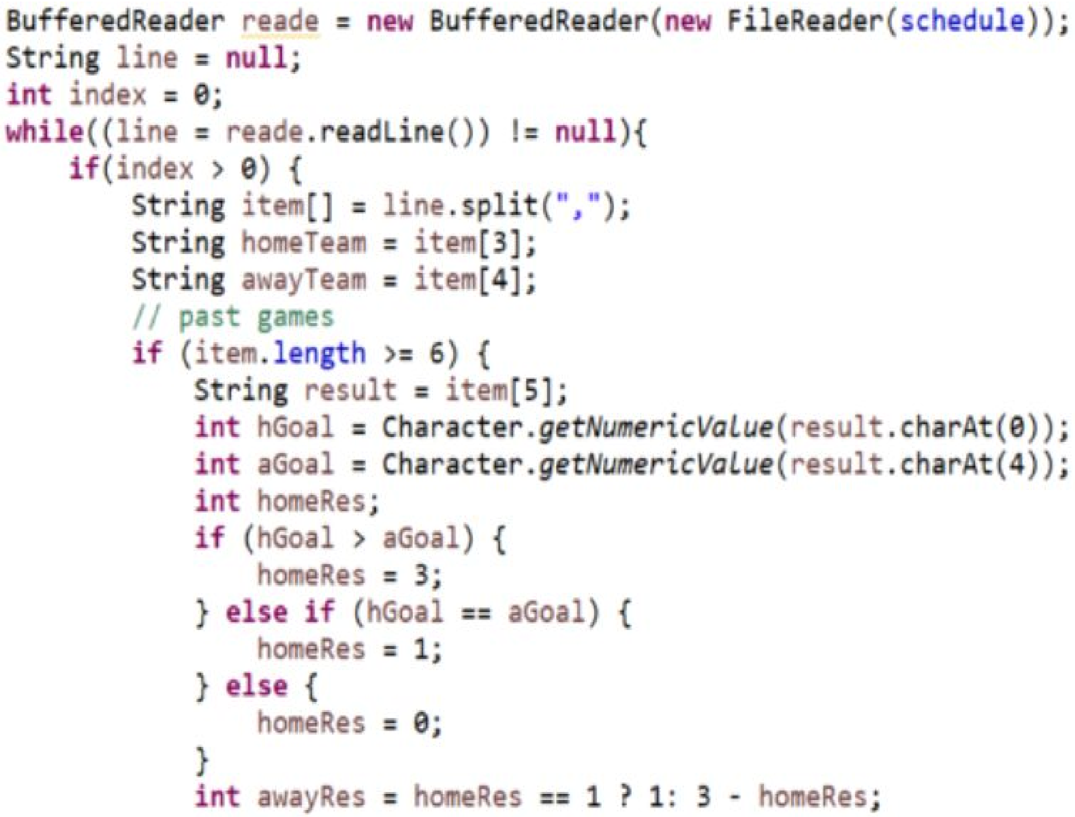
```
for(Map.Entry<String, Double> s : map.entrySet()){
    String score = s.getKey();
    Double probability = s.getValue();
    int hGoal = Character.getNumericValue(score.charAt(0));
    int aGoal = Character.getNumericValue(score.charAt(1));
    if (hGoal > aGoal) {
        resultMap.put(win, resultMap.get(win) + probability);
    } else if (hGoal == aGoal) {
        resultMap.put(draw, resultMap.get(draw) + probability);
    } else {
        resultMap.put(lose, resultMap.get(lose) + probability);
    }
}
```



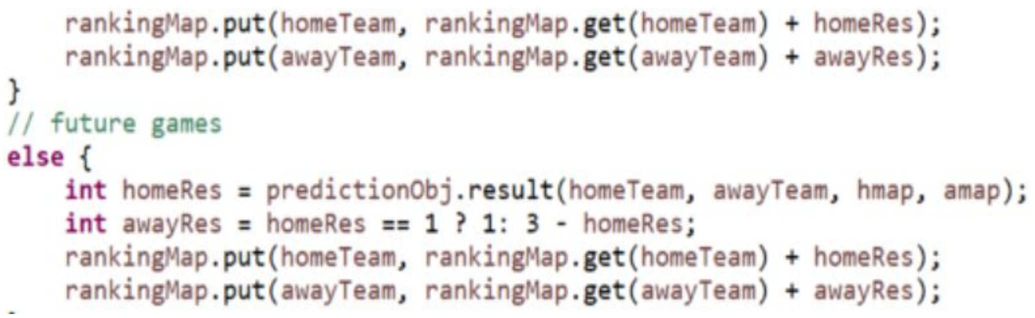
```java
double rand = (double)(new Random().nextInt(99)) / 100;
if (rand <= resultMap.get(win)) {
    return 3;
} else if (rand <= resultMap.get(win) + resultMap.get(draw)) {
    return 1;
} else {
    return 0;
}
```

Predict the 2019-2020 season

We find the game schedule of the current season. Read the already held games and transfer the score result into team points. As for the still-not-held ones, use the predict method mentioned above to predict the results. Then calculate points each team saved and sort them to get the final ranking result of this season.



```java
BufferedReader reade = new BufferedReader(new FileReader(schedule));
String line = null;
int index = 0;
while((line = reade.readLine()) != null){
    if(index > 0) {
        String item[] = line.split(",");
        String homeTeam = item[3];
        String awayTeam = item[4];
        // past games
        if (item.length >= 6) {
            String result = item[5];
            int hGoal = Character.getNumericValue(result.charAt(0));
            int aGoal = Character.getNumericValue(result.charAt(4));
            int homeRes;
            if (hGoal > aGoal) {
                homeRes = 3;
            } else if (hGoal == aGoal) {
                homeRes = 1;
            } else {
                homeRes = 0;
            }
            int awayRes = homeRes == 1 ? 1: 3 - homeRes;
```

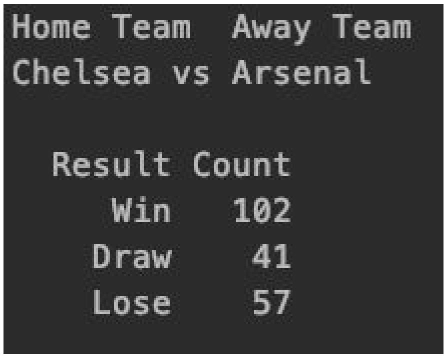


```
    rankingMap.put(homeTeam, rankingMap.get(homeTeam) + homeRes);
    rankingMap.put(awayTeam, rankingMap.get(awayTeam) + awayRes);
}
// future games
else {
    int homeRes = predictionObj.result(homeTeam, awayTeam, hmap, amap);
    int awayRes = homeRes == 1 ? 1: 3 - homeRes;
    rankingMap.put(homeTeam, rankingMap.get(homeTeam) + homeRes);
    rankingMap.put(awayTeam, rankingMap.get(awayTeam) + awayRes);
```

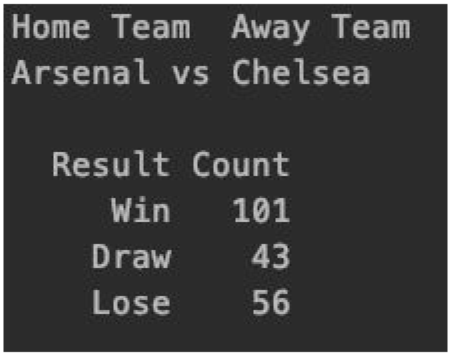
Outputs

Game Simulation

Based on the probability distribution of the game result, we simulated 400 games between Chelsea and Arsenal as shown below. As we all know, Chelsea and Arsenal are both good teams. Their games may be hard to predict, but from the simulation results, it infers that whoever is the home team will have a better chance to win the game.



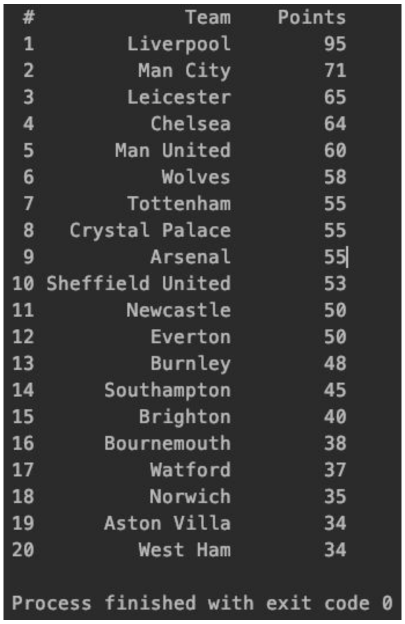
```
Home Team  Away Team
Chelsea vs Arsenal

  Result Count
     Win   102
    Draw    41
    Lose    57
```



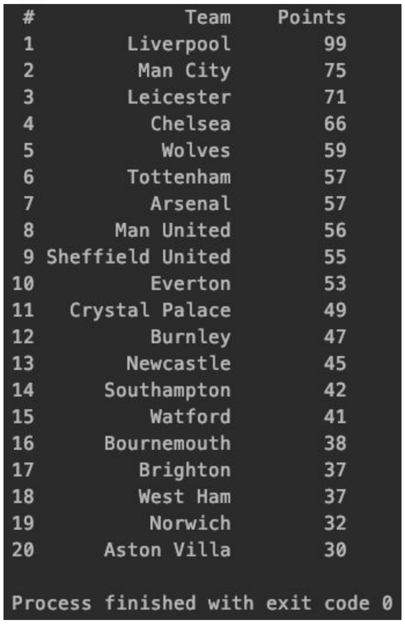
```
Home Team  Away Team
Arsenal vs Chelsea

  Result Count
     Win   101
    Draw    43
    Lose    56
```

EPL Ranking Simulation

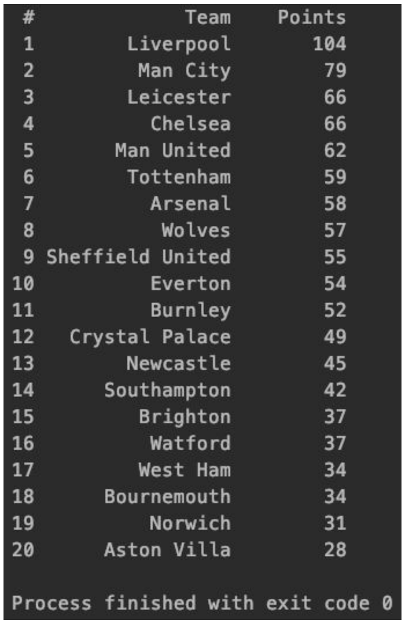
Based on the prediction result of each game, we simulated all the rest games of the 2019/2020 season. Three example simulation results are shown below.



```
#            Team    Points
1       Liverpool        95
2        Man City        71
3       Leicester        65
4         Chelsea        64
5      Man United        60
6          Wolves        58
7       Tottenham        55
8  Crystal Palace        55
9         Arsenal        55
10 Sheffield United      53
11      Newcastle        50
12        Everton        50
13        Burnley        48
14    Southampton        45
15       Brighton        40
16    Bournemouth        38
17        Watford        37
18        Norwich        35
19    Aston Villa        34
20       West Ham        34

Process finished with exit code 0
```



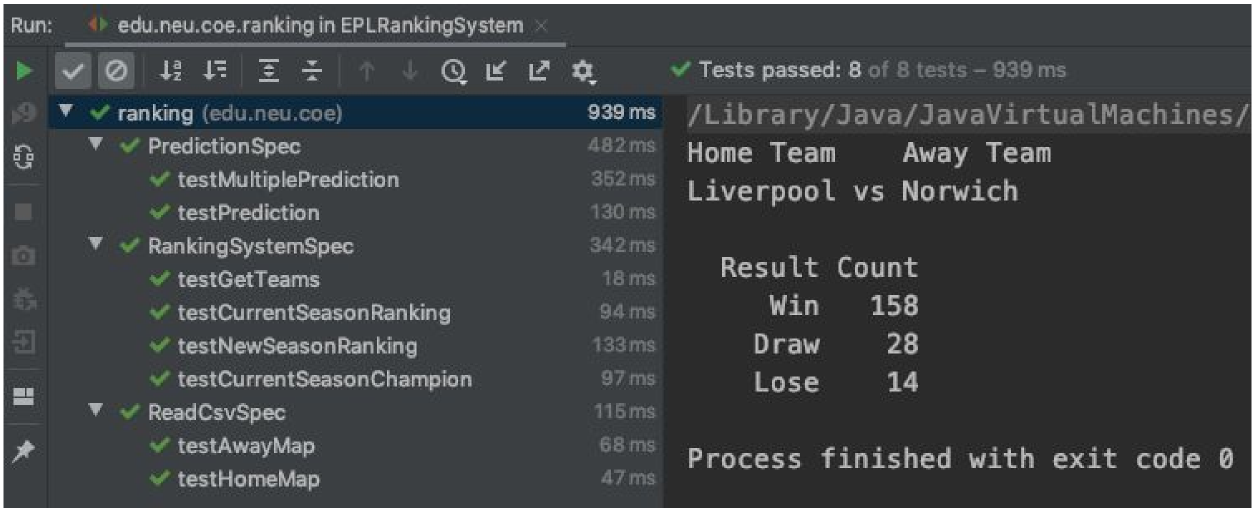
```
#            Team    Points
1       Liverpool        99
2        Man City        75
3       Leicester        71
4         Chelsea        66
5          Wolves        59
6       Tottenham        57
7         Arsenal        57
8      Man United        56
9 Sheffield United       55
10        Everton        53
11 Crystal Palace        49
12        Burnley        47
13      Newcastle        45
14    Southampton        42
15        Watford        41
16    Bournemouth        38
17       Brighton        37
18       West Ham        37
19        Norwich        32
20    Aston Villa        30

Process finished with exit code 0
```



```
#            Team    Points
1       Liverpool       104
2        Man City        79
3       Leicester        66
4         Chelsea        66
5      Man United        62
6       Tottenham        59
7         Arsenal        58
8          Wolves        57
9 Sheffield United       55
10        Everton        54
11        Burnley        52
12 Crystal Palace        49
13      Newcastle        45
14    Southampton        42
15       Brighton        37
16        Watford        37
17       West Ham        34
18    Bournemouth        34
19        Norwich        31
20    Aston Villa        28

Process finished with exit code 0
```

Unit Test

We have 8 test cases for different classes. The screenshot is shown below.



Tests passed: 8 of 8 tests – 939 ms

```
ranking (edu.neu.coe)              939 ms
  PredictionSpec                   482 ms
    testMultiplePrediction         352 ms
    testPrediction                 130 ms
  RankingSystemSpec                342 ms
    testGetTeams                    18 ms
    testCurrentSeasonRanking        94 ms
    testNewSeasonRanking           133 ms
    testCurrentSeasonChampion       97 ms
  ReadCsvSpec                      115 ms
    testAwayMap                     68 ms
    testHomeMap                     47 ms
```

```
/Library/Java/JavaVirtualMachines/
Home Team    Away Team
Liverpool vs Norwich

  Result Count
     Win   158
    Draw    28
    Lose    14

Process finished with exit code 0
```

References

1. The Beautiful Game: Predicting the Premier League with a random model

   https://towardsdatascience.com/o-jogo-bonito-predicting-the-premier-league-with-a-random-model-1b02fa3a7e5a

2. England Football Results Betting Odds

   http://www.football-data.co.uk/englandm.php

3. English Premier League 2019/20 fixture and results

   https://fixturedownload.com/results/epl-2019

4. Probability distribution-Wikipedia

   https://en.wikipedia.org/wiki/Probability_distribution

5. Premier League

   https://en.wikipedia.org/wiki/Premier_League