INFO 6205 - RANKING SYSTEM

By Meda Saikanth Reddy

Abstract

A **sports ranking system** is a system that analyzes the results of sports competitions to provide rankings for each team or player. Common systems include polls of expert voters, crowdsourcing, non-expert voters, betting markets, and computer systems. Ratings, or **power ratings**, are numerical representations of competitive strength, often directly comparable so that the game outcome between any two teams can be predicted. Rankings, or **power rankings**, can be directly provided (e.g., by asking people to rank teams), or can be derived by sorting each team's ratings and assigning an ordinal rank to each team, so that the highest rated team earns the #1 rank. Rating systems provide an alternative to traditional sports standings which are based on win-loss-tie ratios.

The **Premier League**, often referred to as the **English Premier League** or the **EPL** outside England, is the top level of the English primeur league. Contested by 20 clubs, it operates on a system of promotion and relegation with the (EFL). Seasons run from August to May with each team playing 38 matches (playing all 19 other teams both home and away).

Problem Statement

In this paper we are going to predict the final standing of teams in EPL 2019-2020. Due to Covid19 many matches have postponed so using the current standings and matches data I am going to predict the remaining game outcomes and make a final standings table of teams. Currently 216 matches had been played by various teams. So, from the remaining 92 matches I am going to predict the outcomes through which I will find the final Standings.

Implementation

First, we need to fetch data of matches that had been played in this season. I managed to get that data from http://www.football-data.co.uk. The data for the remaining matches has been scraped from website https://www.foxsports.com/soccer/schedule.

Created a class to fetch data of current season and preprocessed it to get data in format needed.

```
Class: public class EplData

Methods: private void getDataFromCsv(String filename)
          protected List<GameOutcome> getRemainingGamesData()
         protected List<GameOutcome> getDataset()
```

```
Test Class: public class EplDataTest
```

In Order to represent GameData for particular match I have created a class called GameOutcome which consists of all the data of a particular match. That is about the teams played and goals by each team. Winner of the match etc.

```
Class: public class GameOutcome

Methods: public String getHomeTeam()
         public String getAwayTeam()
         public int getGoalDifference()
```

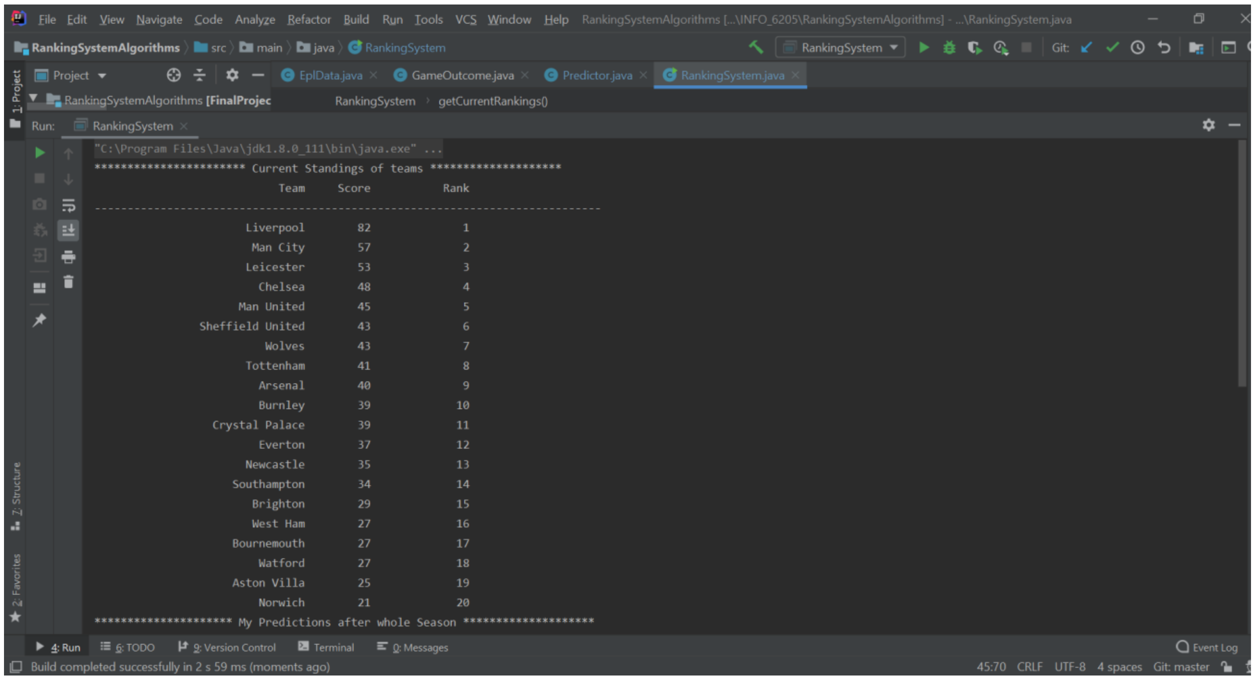
```
Test Class: public class GameOutcomeTest
```

In order to find the current standings, keep track of team and respective points team had gained till now.

For this hashmap would be the right data structure. In order to sort the hashmap with values we can use a comparator.

```
Class : public class RankingSystem

Method: public static Map<String, Integer> sortByValue(Map<String, Integer> hm)
```



```
"C:\Program Files\Java\jdk1.8.0_111\bin\java.exe" ...
********************** Current Standings of teams *******************
                         Team     Score          Rank
---------------------------------------------------------------------------
                    Liverpool      82              1
                     Man City      57              2
                    Leicester      53              3
                      Chelsea      48              4
                   Man United      45              5
             Sheffield United      43              6
                       Wolves      43              7
                    Tottenham      41              8
                      Arsenal      40              9
                      Burnley      39             10
               Crystal Palace      39             11
                      Everton      37             12
                    Newcastle      35             13
                  Southampton      34             14
                     Brighton      29             15
                     West Ham      27             16
                  Bournemouth      27             17
                      Watford      27             18
                  Aston Villa      25             19
                      Norwich      21             20
******************** My Predictions after whole Season *******************
```

Build completed successfully in 2 s 59 ms (moments ago)

Now that we have current standing and remaining matches data. We need to build a model for prediction.

In order to do this, I have created a class Predictor. Initially I have built a simple model where we look at the previous matches and gave certain weightage to wins, losses and goal differences.

Model 1

Rating of team A when playing with team B =

0.3 * wins of team A with other teams in season

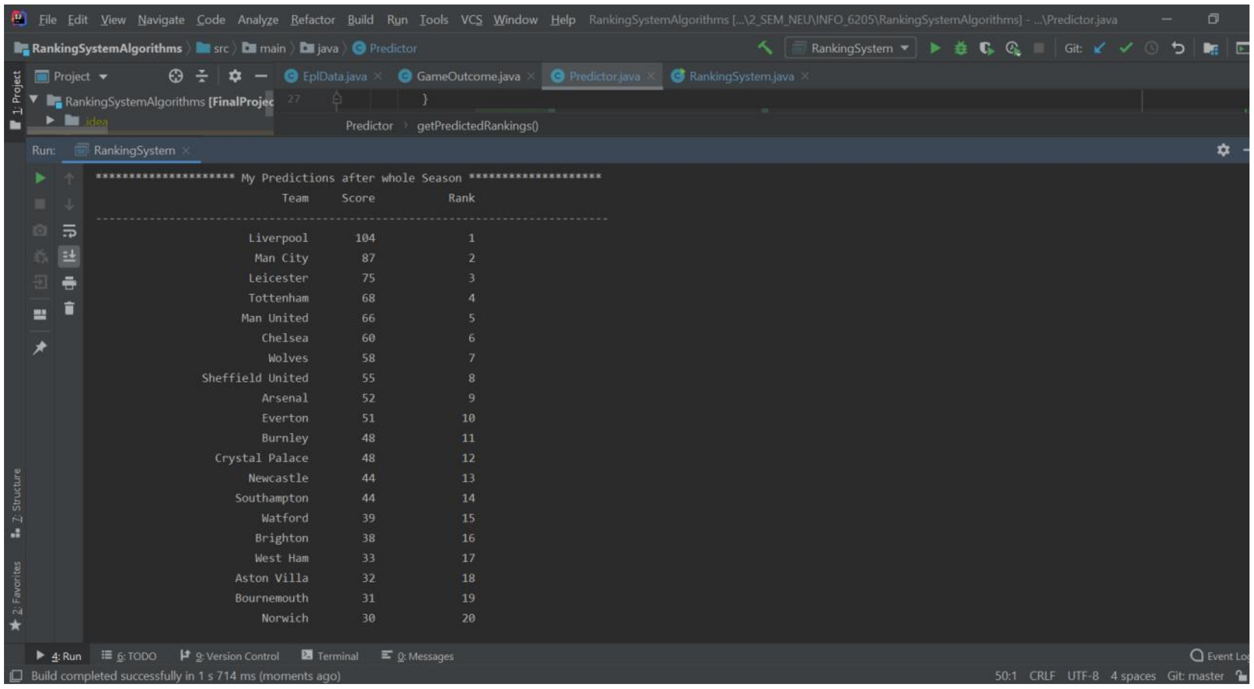
+ 0.3 * goal differences average it made during matches in season

+ 0.4 * wins of team A with team B in season

Implemented in

```
Class: public class Predictor
Method: private String predictGame(String hometeam, String awayteam)
```

Results using this prediction model



```
******************** My Predictions after whole Season ********************
                    Team     Score          Rank
-----------------------------------------------------------------------------
               Liverpool      104             1
                Man City       87             2
               Leicester       75             3
               Tottenham       68             4
              Man United       66             5
                 Chelsea       60             6
                  Wolves       58             7
        Sheffield United       55             8
                 Arsenal       52             9
                 Everton       51            10
                 Burnley       48            11
          Crystal Palace       48            12
               Newcastle       44            13
             Southampton       44            14
                 Watford       39            15
                Brighton       38            16
                West Ham       33            17
             Aston Villa       32            18
             Bournemouth       31            19
                 Norwich       30            20
```

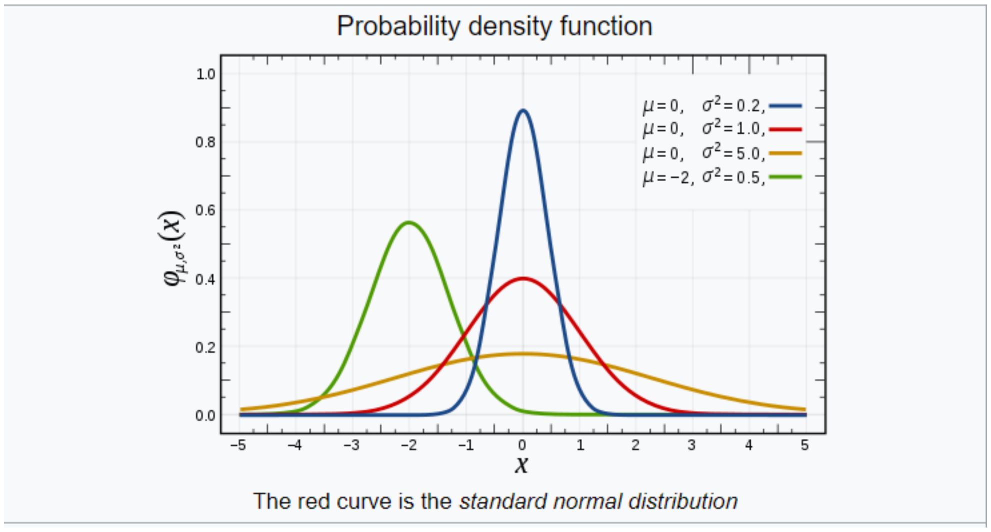
Build completed successfully in 1 s 714 ms (moments ago)

Model 2

In this model we use probability distribution functions for every team in order to do our predictions.

Normal Distribution Curves for goal differences

This is normal distribution curve where x-axis is the goal differences. Similarly, we can draw these curves for each team and by using previous data. In order to have a normal distribution curve we need mean and standard deviation. So, we look at the data of previous matches goal difference and find the mean and standard deviation for that team.



$$y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\begin{aligned} \mu &= \text{Mean} \\ \sigma &= \text{Standard Deviation} \\ \pi &\approx 3.14159\cdots \\ e &\approx 2.71828\cdots \end{aligned}$$

Implementation

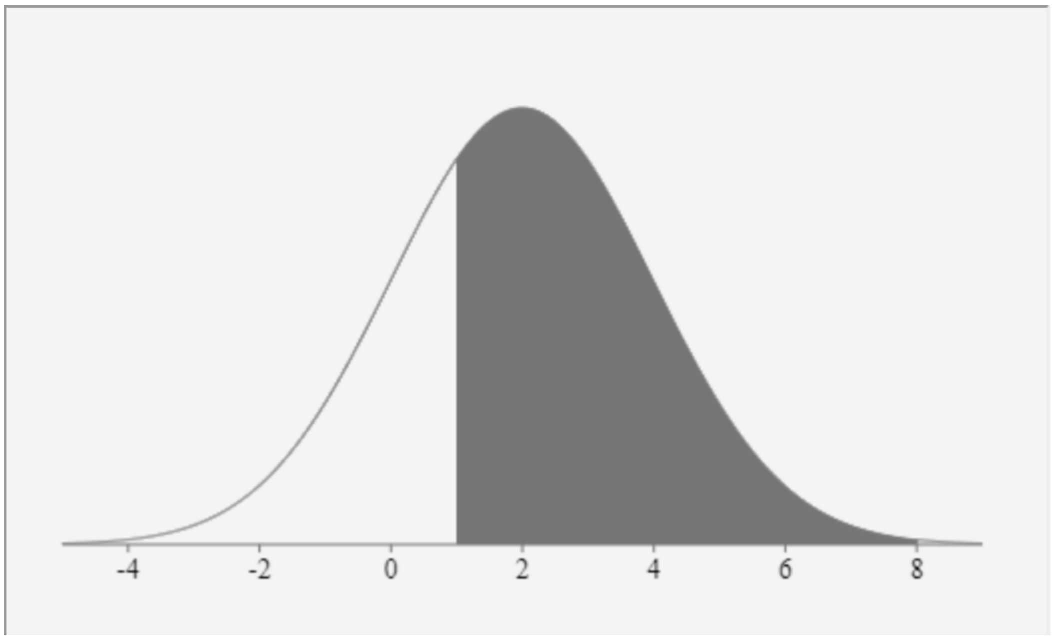
```
Class: public class Predictor

Method: private List<Double> normalDistributionPdf(String team)
```

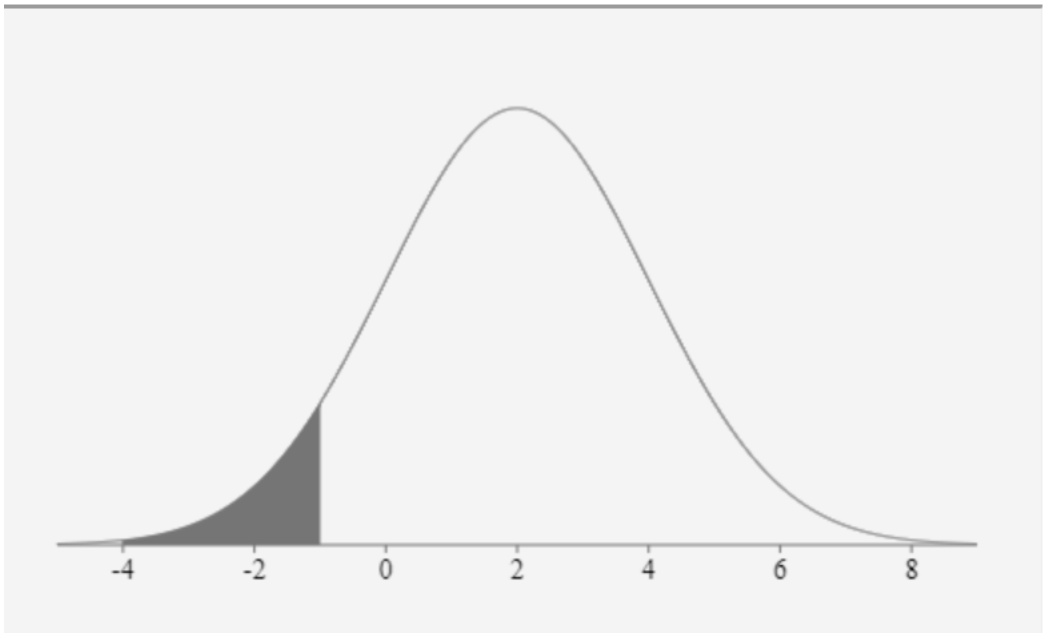
This method returns ArrayList<Double> where at index 0 we have mean and at index 1 we have standard deviation.

In order to find the probability that a team wins a match we can use this pdf (probability distribution function) That is area below the pdf where x-axis is greater than 1.

Below is the graph that is showing the probability that a team will win the match.



Below is a graph (pdf) of a team with goal differences in x axis. It shows probability that a team can loss a match



In order to find the area below the graph I have implemented a fuction

```
Class: public class Predictor {
Method: private double getAreaUnderNormalCurve(double z1, double z2, double mean,
double std)
```

In this model when team A and team B are playing a match. We get the probability distribution function for both teams using past data. From this we calculate the probability of each team winning. That is the are below the graph. If probability of team A winning is higher then we will declare that team A wins the match.

Implementation

```
Class: public class Predictor
Method: private String pdfPrediction(String hometeam, String awayteam)
```

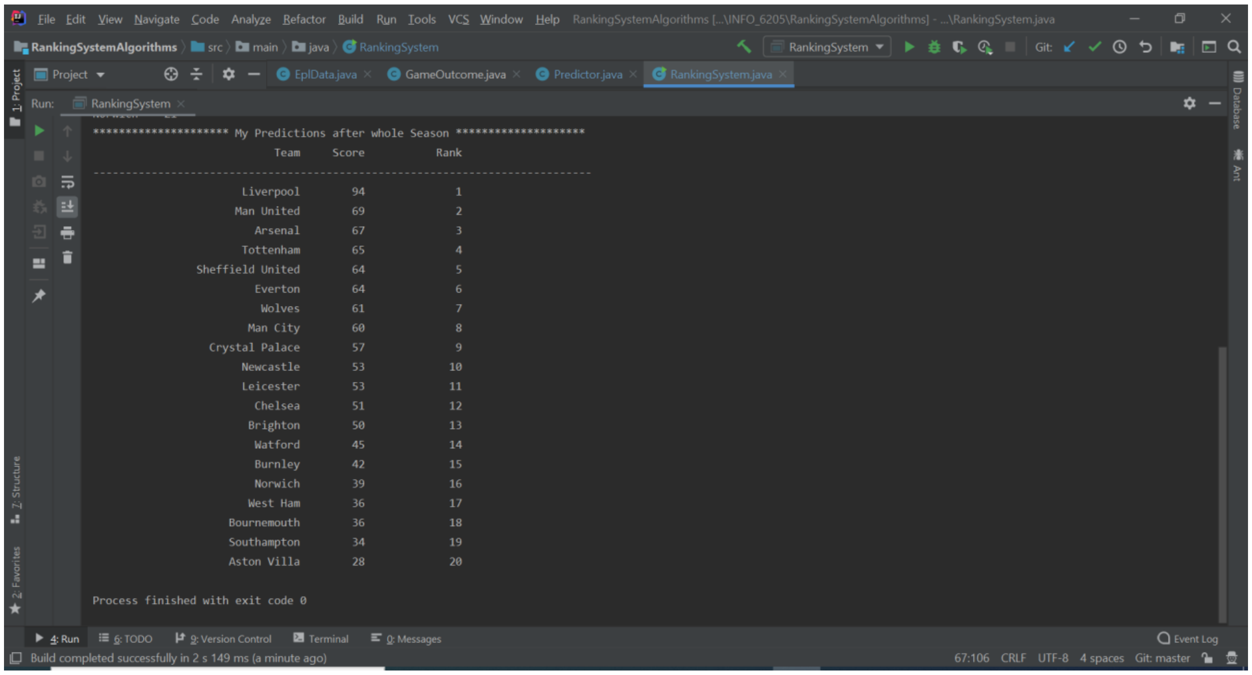
Now that we have model for predictor, we can iterate through all the remaining matches and predict each match and update the rankings simultaneously.

Implementation

```
Class: public class Predictor
Method: public Map<String, Integer> getPredictedRankings()
```

Conclusion

Thus, using two models built in this paper I predicted the final standing of teams for EPL season 2019-2020 with all the data we have currently in this season. Which testes with past seasons gave accuracy of 70%.



```
******************** My Predictions after whole Season ********************
                      Team     Score           Rank
---------------------------------------------------------------------------
                 Liverpool        94              1
                Man United        69              2
                   Arsenal        67              3
                 Tottenham        65              4
          Sheffield United        64              5
                   Everton        64              6
                    Wolves        61              7
                  Man City        60              8
            Crystal Palace        57              9
                 Newcastle        53             10
                 Leicester        53             11
                   Chelsea        51             12
                  Brighton        50             13
                   Watford        45             14
                   Burnley        42             15
                   Norwich        39             16
                  West Ham        36             17
               Bournemouth        36             18
               Southampton        34             19
               Aston Villa        28             20

Process finished with exit code 0
```

Build completed successfully in 2 s 149 ms (a minute ago)

References

https://stackoverflow.com/questions/27096743/how-to-calculate-area-under-a-normal-distribution-in-java

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