FINAL PROJECT - RANKING SYSTEM FOR EPL

Project Report

Yanhong Chen

April 12, 202

1. Objective Description

The EPL (England Premier League) 2019/2020 season has been suspended after March 11, 2020, Due to the spread of COVID-19. I implemented this project to predict the final ranking of EPL by constructing a PDF (probability distribution function) using previous data and predict the team's winning probability through the existed data as well.

3. Solution

My idea is to calculate the expected value of goals different for each two pair teams based on the previous matches, and then adjusting the expected value by adding more data which is completed matches. These results will come up with a prediction table of how many goals each team will win each other team in the next games, and then we will use this result to calculate the average goals different for each other team and this will be our ranking.

In order to predict the final ranking of this season, I using a random function to choose a reasonable goal difference (Greater than -6 and less than 6) based on the normal distribution of the probability of possibility, the variance and mean value of the normal distribution are taken form the result of previous calculation. by cumulating matches and adjusting means and variance, In the end we got the team ranking for this season.

3. Java Class Design

Main.Class	Call the calculation method and provide a list of functions
Calculator.Class	Implement all calculation and prediction logic
CSVReader.Class	Read CSV file content and create Match object
Match.Class	Used to store match attributes

4. Implementation Details

4.1 Normal distribution model

I created a HashMap call teamsWithNormalDist to store the key-value pair of match name and normal distribution attribute which are mean and variance. Once I enter a new match, system will generate a game name and search it at this HashMap.

If it is the first game between those two teams, I using the goals different as the mean value of the normal distribution of goals different probability of the next match, Obviously, the mean value will have highest probability on next match. After that, we generate the variance, as we know the goals difference of football between -3 ~ 3 are very common, so I assign those kinds of goals difference with wide variance 4 because those two teams have similar levels, there will be a lot of uncertainty in the next game, but when the goals different less than -3 or greater than 3, This means that the level of the two teams is very different, and it is likely that there will be a huge spread in their scores in the next game as well. so I reduce the variance to 1.

```java
if (!teamsWithNormalDist.containsKey(game)){
    // Whenever the gap between two teams greater than six we decreasing it to 6 manually
    // because this situation is too rare to repeat
    if( absMean <= 3 ){
        // when the goals different are 0, 1, 2, 3, I assume the variance to be 4, the standard deviation to be 2
        teamsWithNormalDist.put(game, new Double[]{Double.valueOf(mean), 4.0 });
    } else {
        // when the goals different are 4, 5, 6, I assume the variance to be 1, the standard deviation to be 1
        teamsWithNormalDist.put(game, new Double[]{Double.valueOf(mean), 1.0 });
    }
}
```

If those two teams have played before, the system will retrieve the mean and variance, and the average mean value will be the new value of the new normal distribution. Then we use the same theory to calculate the variance of this new match, the new variance will be the sum of those new and previous variances. Finally we update the values of HashMap.

```java
        // If the two teams have already played
        Double[] NDList = teamsWithNormalDist.get(game);
        Double previousMean = NDList[0];
        Double previousVariance = NDList[1];
        Double newVariance;
        Double newMean = (mean + previousMean)/2;
        if( absMean <= 3 ){
            newVariance = previousVariance + 4.0;
        } else {
            newVariance = previousVariance + 1.0;
        }
        teamsWithNormalDist.replace(game, new Double[]{ newMean, newVariance });
```

Once we enter the suspended matches (match with team names but no goals), Using random.nextGaussian() function with means and variance retrieved from the HashMap (If it's unretrievable, which means those means did not meet before, I assign their goals with 0 : 0) to generate a random score difference (The probability of their generation is in accordance with the normal distribution we calculated before). After rounding this value it will be using as parameter to generate new mean and variance in order to update the HashMap.

```java
    // prediction for suspended matches.
if(homeTeamGoals == -1 && awayTeamGoals == -1){
    Random random = new Random();
    if (teamsWithNormalDist.containsKey(game)){
        Double[] NDList = teamsWithNormalDist.get(game);
        Double previousMean = NDList[0];
        Double previousVariance = NDList[1];

        int predictScoreDiff = 0;
        do{
            // using random.nextGaussian() to generate random score based on Gaussian distribution.
            double predictScoreDiffDouble = Math.sqrt(previousVariance) * random.nextGaussian() + previousMean;
            predictScoreDiff = (int)Math.round(predictScoreDiffDouble);
        }while(predictScoreDiff > 6 || predictScoreDiff < -6);

        if(predictScoreDiff <= 0){
            homeTeamGoals = 0;
            awayTeamGoals = 0 - predictScoreDiff;
        }else{
            homeTeamGoals = predictScoreDiff;
            awayTeamGoals = 0;
        }
    } else{
        homeTeamGoals = 0;
        awayTeamGoals = 0;
    }
    System.out.println("Prediction for match of " + match.getMatchDate() +
            ": HomeTeam: " + match.getHomeTeam() +
            " Goals: " + homeTeamGoals +
            " AwayTeam: " + match.getAwayTeam() +
            " Goals: " + awayTeamGoals );
};
```

4.2 Ranking

I using HashMap call RankMap to store the ranking of EPL teams. when we calculate all matches after all steps I mentioned above. we already have a prediction table of how many goals each team will win each other team in the next games. By calculating the mean of each row of the table. We can calculate the average goal difference for one team to win other teams. After that, the value which is sorted by size will be the final ranking.

```java
public static void calculateRank() {
    for (String team : teamList) {
        int count = 0;
        double totalMean = 0;
        for (String key : teamsWithNormalDist.keySet()) {
            if (key.contains(team) && key.indexOf(team) == 0) { // if team name in the first position of key
                count++;
                totalMean += teamsWithNormalDist.get(key)[0];
            } else if (key.contains(team) && key.indexOf(team) != 0) { // if team name in the second position of key
                count++;
                totalMean += (-teamsWithNormalDist.get(key)[0]);
            }
        }
        double averageMean = totalMean/count;
        RankMap.put(team, averageMean);
    }
    Map<String, Double> sortedRankMap = sortByValue(RankMap);
    int rank = 1;
    for (Map.Entry<String, Double> entry : sortedRankMap.entrySet()) {
        System.out.println("Rank:"+ rank + "  Team: " + entry.getKey());
        rank++;
        // System.out.println("Team = " + entry.getKey() + ", AverageMean = " + entry.getValue());
    }
}
```

4.2 Prediction using PDF and CDF

When we making prediction of upcoming matches, we can build a PDF(Probability Density Function) using mean and variance from previous steps. By substituting the parameters into the formula in the image below, we can get the corresponding probability after entering the desired prediction goal difference. (Since PDF can only calculate the probability of a certain point, I use CDF while calculating the probability. When the user wants to calculate the probability of a goal difference of 1, it is to calculate the total probability of the point difference between 0.5 and 1.5).

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

When we calculating the winning probability , the CDF (Cumulative Distribution Function) provides us the probability that the score difference is higher than a certain value. when we type in 0 as mean to this formula, which means the two teams will tie the next game, It will calculate the sum probability of negative infinity to zero. We subtract this value from 1 to get the probability that this team will win at the next game.

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

```java
// Normal distribution pdf (probability distribution function ) with expectation
public static void calculatePDF(String homeTeam, String awayTeam, Integer expectedGoal){
    try{
        String keyHA = homeTeam + " --VS-- " + awayTeam;
        String keyAH = awayTeam + " --VS-- " + homeTeam;
        double mean;
        double variance;
        if (teamsWithNormalDist.containsKey(keyHA)){
            mean = teamsWithNormalDist.get(keyHA)[0];
            variance = teamsWithNormalDist.get(keyHA)[1];
        } else if (teamsWithNormalDist.containsKey(keyAH)){
            mean = -teamsWithNormalDist.get(keyAH)[0];
            variance = teamsWithNormalDist.get(keyAH)[1];
        } else {
            System.out.println( "oops, Unable to find previous match between these two teams, please re-enter.");
            return;
        };

        double winningPercentage = 1 - calculateCDF( expectedGoal: 0, mean, variance);
        double Prediction = calculateCDF( expectedGoal: expectedGoal + 0.5, mean, variance) -
                calculateCDF( expectedGoal: expectedGoal - 0.5, mean, variance);

        System.out.println( "Team " + homeTeam + " has " + String.format("%.2f", winningPercentage * 100) +
                "% chances of winning Team " + awayTeam + " at next match."+ "\n" +
                "The chance of team " + homeTeam + " winning team " + awayTeam +
                " by " + expectedGoal + " goals is "+ String.format("%.2f", Prediction * 100) +"%");

    }catch(Exception e){
        System.out.println( "oops, Unable to find previous match between these two teams, please re-enter.");
    }
}
```

```java
// Normal distribution CDF (Cumulative distribution function ) with mean and variance
public static Double calculateCDF(double expectedGoal, double mean, double variance){
    double variable = (expectedGoal - mean) / Math.sqrt(variance);
    if (variable < -8.0) return 0.0;
    if (variable >  8.0) return 1.0;
    double sum = 0.0, term = variable;
    for (int i = 3; sum + term != sum; i += 2) {
        sum  = sum + term;
        term = term *  Math.pow((expectedGoal - mean), 2) / (i * variance);
    }
    double CDF = 0.5 + sum * Math.exp( - Math.pow((expectedGoal - mean),2)/(2 * variance)) / Math.sqrt(2 * Math.PI);
    return CDF;
}
```

5. Evidence of Running

5.1 Goals predictions for suspended matches.

```
Prediction for match of 2020-05-02: HomeTeam: Crystal Palace Goals: 2 AwayTeam: Manchester Utd Goals: 0
Prediction for match of 2020-05-02: HomeTeam: Chelsea Goals: 0 AwayTeam: Norwich City Goals: 0
Prediction for match of 2020-05-02: HomeTeam: Everton Goals: 3 AwayTeam: Aston Villa Goals: 0
Prediction for match of 2020-05-02: HomeTeam: Arsenal Goals: 2 AwayTeam: Liverpool Goals: 0
Prediction for match of 2020-05-02: HomeTeam: Southampton Goals: 0 AwayTeam: Brighton Goals: 0
Prediction for match of 2020-05-02: HomeTeam: Leicester City Goals: 0 AwayTeam: Sheffield Utd Goals: 5
Prediction for match of 2020-05-09: HomeTeam: Tottenham Goals: 1 AwayTeam: Leicester City Goals: 0
Prediction for match of 2020-05-09: HomeTeam: Brighton Goals: 2 AwayTeam: Newcastle Utd Goals: 0
Prediction for match of 2020-05-09: HomeTeam: Wolves Goals: 0 AwayTeam: Crystal Palace Goals: 3
Prediction for match of 2020-05-09: HomeTeam: Liverpool Goals: 0 AwayTeam: Chelsea Goals: 2
Prediction for match of 2020-05-09: HomeTeam: Sheffield Utd Goals: 0 AwayTeam: Everton Goals: 1
Prediction for match of 2020-05-09: HomeTeam: Manchester Utd Goals: 1 AwayTeam: West Ham Goals: 0
Prediction for match of 2020-05-09: HomeTeam: Bournemouth Goals: 0 AwayTeam: Southampton Goals: 1
Prediction for match of 2020-05-09: HomeTeam: Watford Goals: 5 AwayTeam: Manchester City Goals: 0
Prediction for match of 2020-05-09: HomeTeam: Aston Villa Goals: 0 AwayTeam: Arsenal Goals: 2
Prediction for match of 2020-05-09: HomeTeam: Norwich City Goals: 6 AwayTeam: Burnley Goals: 0
Prediction for match of 2020-05-17: HomeTeam: Chelsea Goals: 0 AwayTeam: Wolves Goals: 4
Prediction for match of 2020-05-17: HomeTeam: Manchester City Goals: 3 AwayTeam: Norwich City Goals: 0
Prediction for match of 2020-05-17: HomeTeam: Newcastle Utd Goals: 0 AwayTeam: Liverpool Goals: 0
Prediction for match of 2020-05-17: HomeTeam: Leicester City Goals: 0 AwayTeam: Manchester Utd Goals: 2
Prediction for match of 2020-05-17: HomeTeam: Everton Goals: 0 AwayTeam: Bournemouth Goals: 0
Prediction for match of 2020-05-17: HomeTeam: West Ham Goals: 0 AwayTeam: Aston Villa Goals: 1
Prediction for match of 2020-05-17: HomeTeam: Crystal Palace Goals: 4 AwayTeam: Tottenham Goals: 0
Prediction for match of 2020-05-17: HomeTeam: Arsenal Goals: 0 AwayTeam: Watford Goals: 0
Prediction for match of 2020-05-17: HomeTeam: Southampton Goals: 0 AwayTeam: Sheffield Utd Goals: 1
Prediction for match of 2020-05-17: HomeTeam: Burnley Goals: 1 AwayTeam: Brighton Goals: 0
```

5.2 Current ranking

```
Press 1 to view current ranking.
Press 2 to type in a new match.
Press 3 to make predictions.
Press 4 to exit.
1
Rank:1  Team: Liverpool
Rank:2  Team: Manchester City
Rank:3  Team: Leicester City
Rank:4  Team: Chelsea
Rank:5  Team: Manchester Utd
Rank:6  Team: Wolves
Rank:7  Team: Tottenham
Rank:8  Team: Arsenal
Rank:9  Team: Sheffield Utd
Rank:10  Team: Burnley
Rank:11  Team: Southampton
Rank:12  Team: West Ham
Rank:13  Team: Everton
Rank:14  Team: Bournemouth
Rank:15  Team: Newcastle Utd
Rank:16  Team: Aston Villa
Rank:17  Team: Watford
Rank:18  Team: Brighton
Rank:19  Team: Crystal Palace
Rank:20  Team: Norwich City
```

5.2 Adding new matches and adjusting ranking.

```
Press 1 to view current ranking.
Press 2 to type in a new match.
Press 3 to make predictions.
Press 4 to exit.
2
Please enter the name of home team
Manchester City
Please enter the score of home team
10
Please enter the name of away team
Liverpool
Please enter the score of away team
0
Rank:1  Team: Manchester City
Rank:2  Team: Liverpool
Rank:3  Team: Leicester City
Rank:4  Team: Manchester Utd
Rank:5  Team: Chelsea
Rank:6  Team: Sheffield Utd
Rank:7  Team: Wolves
Rank:8  Team: Tottenham
Rank:9  Team: Crystal Palace
Rank:10  Team: Arsenal
Rank:11  Team: Burnley
Rank:12  Team: Southampton
Rank:13  Team: West Ham
Rank:14  Team: Brighton
Rank:15  Team: Everton
Rank:16  Team: Watford
Rank:17  Team: Newcastle Utd
Rank:18  Team: Bournemouth
Rank:19  Team: Aston Villa
Rank:20  Team: Norwich City
```

5.2 Calculating winning probability.

```
Press 1 to view current ranking.
Press 2 to type in a new match.
Press 3 to make predictions.
Press 4 to exit.

3
Please enter the name of home team
Liverpool
Please enter the name of away team
Manchester City
Please enter the score difference you wish to predict
4
Team Liverpool has 81.16% chances of winning Team Manchester City at next match.
The chance of team Liverpool winning team Manchester City by 4 goals is 12.21%
```