

# Project Report On EPL Ranking System

**Course: INFO 6205**

**Team Members:**

**Aayush Jain (001401616)**

[https://github.com/aayushjain92/INFO6205\\_Final\\_Project](https://github.com/aayushjain92/INFO6205_Final_Project)



# Contents

Introduction .....	3
What is Ranking System? .....	3
Aim of the Project .....	3
Datasets used in the Project .....	4
Implementation .....	5
Data Structures.....	6
Features: .....	7
Output.....	7
References .....	9

# Introduction

## What is Ranking System?

Ranking system is an algorithm designed to evaluate the expression  $P(X_i, X_j)$  where  $X_i, X_j$  are elements from a set of competing elements  $X$  and  $P(X_i, X_j)$  is the probability that  $X_i$  would beat  $X_j$  if they met in a head to head matchup at neutral territory.

The application could be used to predict the rankings of the teams in a tournament and to get the likelihood of a team winning.

## Aim of the Project

The 2019–20 Premier League is the 28th season of the Premier League, the top English professional football league, since its establishment in 1992. The League was started on 9 August 2019 but since March, the season has been affected by the 2020 coronavirus pandemic and the remaining matches have been postponed.

There are 20 clubs **in the Premier League**. During the course of a season (from August to May) each club **plays** the others **twice** (a **double** round-robin system), once at their home stadium and once at that of their opponents', for 38 games. Teams receive three points for a win and one point for a draw.

The Ranking System is an application designed to predict the outcome of the remaining matches that have been postponed and to predict the ranks of the clubs in this season.

# Datasets used in the Project

The data has been published on the following website. It contains the entire history of EPL (English Premiere League) since 2000.

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

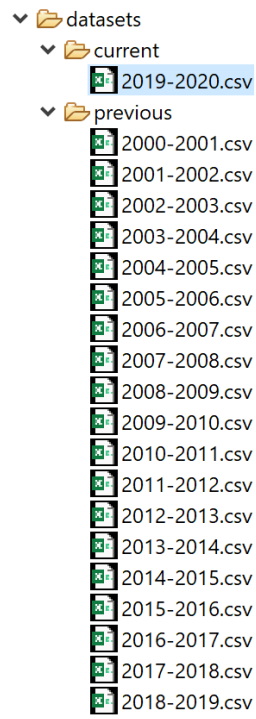


Fig: Sample screenshot of the data file

	Div	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG
1	E0	9/8/2019	Liverpool	Norwich	4	1	H	4	0
2	E0	10/8/2019	West Ham	Man City	0	5	A	0	1
3	E0	10/8/2019	Bournemouth	Sheffield United	1	1	D	0	0
4	E0	10/8/2019	Burnley	Southampton	3	0	H	0	0
5	E0	10/8/2019	Crystal Palace	Everton	0	0	D	0	0
6	E0	10/8/2019	Watford	Brighton	0	3	A	0	1
7	E0	10/8/2019	Tottenham	Aston Villa	3	1	H	0	1
8	E0	11/8/2019	Leicester	Wolves	0	0	D	0	0
9	E0	11/8/2019	Newcastle	Arsenal	0	1	A	0	0
10	E0	11/8/2019	Man United	Chelsea	4	0	H	1	0
11	E0	17/08/2019	Arsenal	Burnley	2	1	H	1	1
12	E0	17/08/2019	Aston Villa	Bournemouth	1	2	A	0	2

The historical data and the data of the current season has been separated into previous and current directories.

Following is the description of the columns used:

Div = League Division

Date = Match Date (dd/mm/yy)

Time = Time of match kick off

HomeTeam = Home Team

AwayTeam = Away Team

FTHG and HG = Full Time Home Team Goals

FTAG and AG = Full Time Away Team Goals

# Implementation

Let's first get the overview of the code design:

Classes:

- `com.epl.model`
  - `League.java`
    - The business logic is using League and Team Model to store and process the data
    - League contains three member variables i.e. a team directory which has 20 participating teams this season; `matchesPlayed` variable which contains all the info about the matches that have already been played; `remainingMatches` variable which contains all the info about the matches in the history because these matches have been postponed and are going to be predicted on the historical data
  - `Team.java`
    - It contains the team related information and the most important one is Score.
- `com.epl.prediction`
  - `Ranking System.java`
    - It contains the main function
    - It also handles the printing functions to the console
  - `ScoreCalculator.java`
    - It contains the business logic and the merging of the Gaussian distributions to reach the most probable prediction
    - It uses the historical data to create a probability density function
- `com.epl.services`
  - `LoadCSVData.java`
    - It loads the data from the CSV files in the model

```

com.epl.model
├── League.java
│   ├── League
│   │   ├── matchesPlayed
│   │   ├── remainingMatches
│   │   ├── teamDirectory
│   │   ├── League()
│   │   ├── addTeam(String) : void
│   │   ├── getMatchesPlayed() : Map<String, Map<String, List>>
│   │   ├── getRemainingMatches() : Map<String, Map<String, List>>
│   │   ├── getTeamDirectory() : Map<String, Team>
│   │   ├── initializeCurrentSeason(List<String>) : void
│   │   ├── initializePreviousSeason(List<String>) : void
│   │   └── populateRemainingMatches(String[], String[]) : void
│   └── Team.java
│       ├── Team
│       │   ├── matches
│       │   ├── name
│       │   ├── score
│       │   ├── Team(String)
│       │   ├── compareTo(Team) : int
│       │   ├── getMatches() : Map<String, List>
│       │   ├── getName() : String
│       │   ├── getScore() : int
│       │   └── setScore(int) : void

```

```

com.epl.prediction
├── RankingSystem.java
│   ├── RankingSystem
│   │   ├── currentSeasonDataFiles
│   │   ├── previousSeasonDataFiles
│   │   ├── listFilesForFolder(String, File) : List<String>
│   │   ├── main(String[]) : void
│   │   ├── printProbabilityTable(League) : void
│   │   └── printRankingPrediction(List<Team>) : void
│   └── ScoreCalculator.java
│       ├── ScoreCalculator
│       │   ├── calculateProbability(League) : void
│       │   └── updateTeamScores(League) : void
└── com.epl.services
    ├── LoadCSVData.java
    │   ├── LoadCSVData
    │   │   ├── initializeCurrent(League, String) : void
    │   │   └── initializePrevious(League, String) : void

```

## Data Structures

Java.util.Map Collection has been heavily used to enable faster lookup of the data. The first “key” points to the home team and the next “key” points to the away team, making sure to reach the linked list of goal differences in O(1) time.

For Example:

**remainingMatches.get(“Arsenal”).get(“Liverpool”)** will provide a list of goal differences between the two teams considering all the matches in the past since 2000.

```

public class League {

    private Map<String, Team> teamDirectory;

    private Map<String, Map<String, List>> matchesPlayed;

    private Map<String, Map<String, List>> remainingMatches;

```

## Features:

Home team advantage has been considered and accounted separately:

```
public void calculateProbability(League league2019_20) {
    // TODO Auto-generated method stub
    Map<String, Map<String, List>> remainingMatches = league2019_20.getRemainingMatches();

    for (String homeTeam : remainingMatches.keySet()) {
        Map<String, List> rivals = remainingMatches.get(homeTeam);
        for (String awayTeam : rivals.keySet()) {
            int winFreq = 0, loseFreq = 0, drawFreq = 0;
            List<Integer> list = rivals.get(awayTeam);

            // calculating the frequency with the same team of winning, losing and drawing a
            // game
            for (Integer goalDifference : list) {
                if (goalDifference > 0) {
                    ++winFreq;
                } else if (goalDifference == 0) {
                    ++drawFreq;
                } else if (goalDifference < 0) {
                    ++loseFreq;
                }
            }
        }
    }
```

## Output

Aston Villa	Liverpool	[ Home Winning Prob: 0.0625, Home Losing Prob: 0.6875, Draw Probability: 0.25]
Aston Villa	Sheffield United	[ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
Aston Villa	Norwich	[ Home Winning Prob: 0.8, Home Losing Prob: 0, Draw Probability: 0.2]
Aston Villa	West Ham	[ Home Winning Prob: 0.538462, Home Losing Prob: 0.153846, Draw Probability: 0.307692]
Aston Villa	Newcastle	[ Home Winning Prob: 0.333333, Home Losing Prob: 0.266667, Draw Probability: 0.4]
Aston Villa	Leicester	[ Home Winning Prob: 0.6, Home Losing Prob: 0.2, Draw Probability: 0.2]
Aston Villa	Burnley	[ Home Winning Prob: 0.5, Home Losing Prob: 0.5, Draw Probability: 0]
Aston Villa	Tottenham	[ Home Winning Prob: 0.25, Home Losing Prob: 0.4375, Draw Probability: 0.3125]
Aston Villa	Man United	[ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
Aston Villa	Bournemouth	[ Home Winning Prob: 0, Home Losing Prob: 1, Draw Probability: 0]
Aston Villa	Crystal Palace	[ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
Aston Villa	Southampton	[ Home Winning Prob: 0.333333, Home Losing Prob: 0.333333, Draw Probability: 0.333333]
Aston Villa	Watford	[ Home Winning Prob: 0.5, Home Losing Prob: 0.5, Draw Probability: 0]
Aston Villa	Wolves	[ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
Aston Villa	Arsenal	[ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
Aston Villa	Chelsea	[ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
Aston Villa	Everton	[ Home Winning Prob: 0.375, Home Losing Prob: 0.25, Draw Probability: 0.375]
Aston Villa	Man City	[ Home Winning Prob: 0.266667, Home Losing Prob: 0.4, Draw Probability: 0.333333]

The above output table shows the probabilities of Aston Villa (Home Team) winning, losing and drawing a match against different teams.

```

Console Problems Debug Shell Search
<terminated> RankingSystem (2) [Java Application] C:\Program Files\Java\jdk1.8.0_111\bin\javaw.exe (Apr 17, 2020, 10:56:18 PM)
West Ham Liverpool [ Home Winning Prob: 0.25, Home Losing Prob: 0.5625, Draw Probability: 0.1875]
West Ham Brighton [ Home Winning Prob: 0, Home Losing Prob: 0.5, Draw Probability: 0.5]
West Ham Aston Villa [ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
West Ham Sheffield United [ Home Winning Prob: 1, Home Losing Prob: 0, Draw Probability: 0]
West Ham Norwich [ Home Winning Prob: 0.666667, Home Losing Prob: 0, Draw Probability: 0.333333]
West Ham Newcastle [ Home Winning Prob: 0.428571, Home Losing Prob: 0.357143, Draw Probability: 0.214286]
West Ham Leicester [ Home Winning Prob: 0.285714, Home Losing Prob: 0.428571, Draw Probability: 0.285714]
West Ham Burnley [ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
West Ham Tottenham [ Home Winning Prob: 0.375, Home Losing Prob: 0.5, Draw Probability: 0.125]
West Ham Man United [ Home Winning Prob: 0.25, Home Losing Prob: 0.4375, Draw Probability: 0.3125]
West Ham Bournemouth [ Home Winning Prob: 0.25, Home Losing Prob: 0.5, Draw Probability: 0.25]
West Ham Crystal Palace [ Home Winning Prob: 0.333333, Home Losing Prob: 0.333333, Draw Probability: 0.333333]
West Ham Watford [ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
West Ham Southampton [ Home Winning Prob: 0.7, Home Losing Prob: 0.3, Draw Probability: 0]
West Ham Wolves [ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
West Ham Arsenal [ Home Winning Prob: 0.125, Home Losing Prob: 0.5, Draw Probability: 0.375]
West Ham Chelsea [ Home Winning Prob: 0.44812, Home Losing Prob: 0.284211, Draw Probability: 0.267669]
West Ham Everton [ Home Winning Prob: 0.1875, Home Losing Prob: 0.5625, Draw Probability: 0.25]
West Ham Man City [ Home Winning Prob: 0.266667, Home Losing Prob: 0.466667, Draw Probability: 0.266667]

```

The above output table shows the probabilities of West Ham (Home Team) winning, losing and drawing a match against different teams.

# ===== RANKING PREDICTION =====

CLUB	SCORE
-----	
Liverpool	100
Arsenal	85
Man United	84
Tottenham	81
Chelsea	78
Man City	75
Everton	72
Newcastle	59
Sheffield United	52
Southampton	51
Wolves	51
Leicester	50
Crystal Palace	49
Aston Villa	48
West Ham	47
Burnley	47
Norwich	46
Brighton	44
Bournemouth	44
Watford	42

The above output table displays the predicted ranking of the clubs participated this season. Liverpool ended up securing first rank whereas Brighton, Bournemouth, and Wartford are at the last positions.



## References

[https://en.wikipedia.org/wiki/Premier\\_League](https://en.wikipedia.org/wiki/Premier_League)