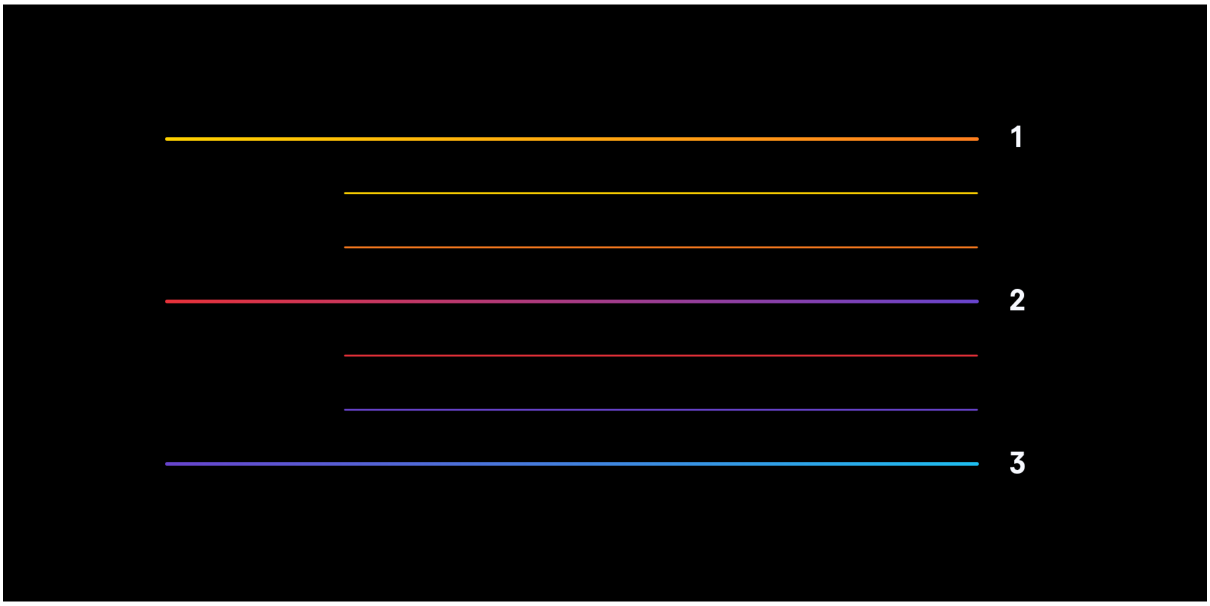


INFO 6205 Ranking System Project

Yuanxin Zhang 001405356
Yongjun Ou 001023025

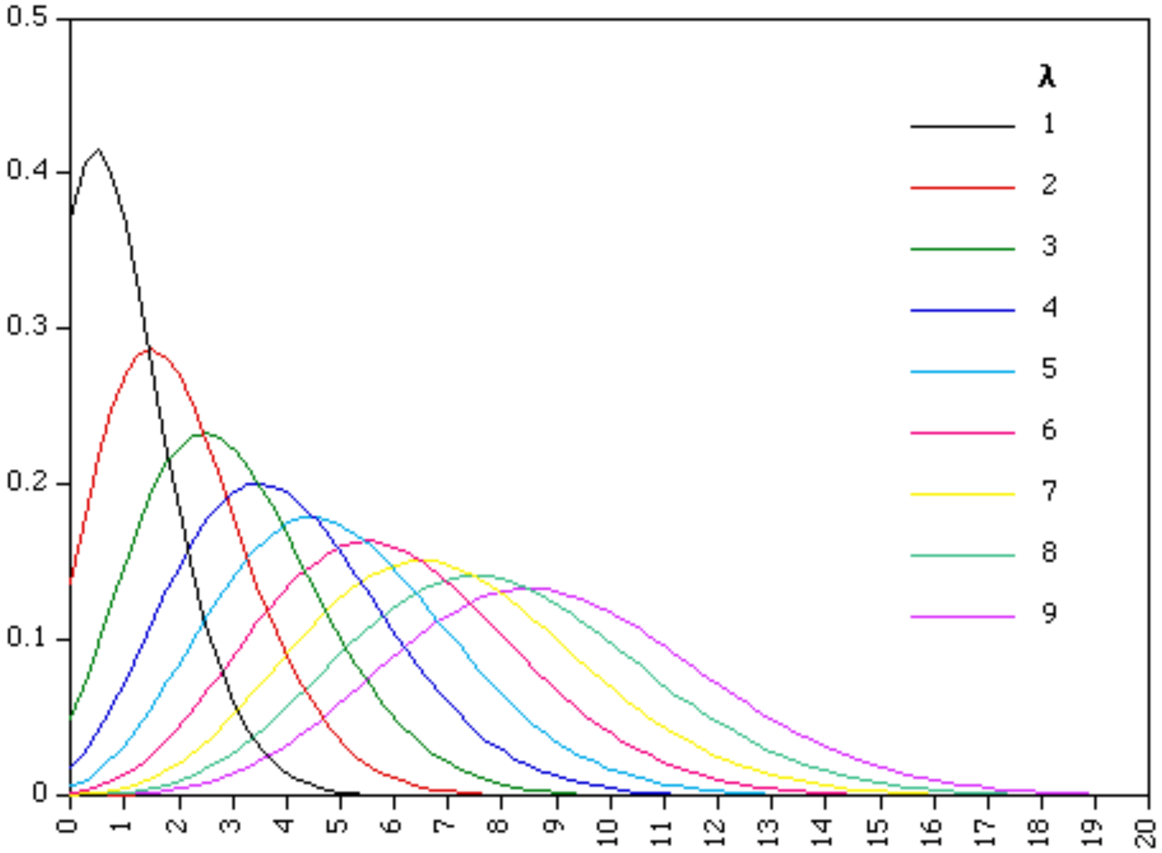


Figure1: Poisson Distribution

Introduction

What is Ranking System?

In statistics, ranking is the data transformation in which numerical or ordinal values are replaced by their rank when the data are sorted[1]. For example, we have the numerical data 27.5, 10.9, 17.8, 21.9, the ranks of these data items would be 1, 4, 3 and 2 (from large to small) respectively. In this project, we will based on the English Premier League, use our own method and proper program structure to rebuild a new analysis mechanism to rank all the teams. Finally, we will show the team ranking result in tables.

What is Poisson Distribution?

The Poisson distribution, named after French mathematician Simeon Denis Poisson, is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event[2]. We will use it and the *Cumulative Distribution Function* to calculate the theoretical calculation distribution of the number of goals, instead of *Possibility Density Function of Normal Distribution*. We will discuss it more detailed in the following Implementation section.

About the Premier League

The Premier League, often referred to as the English Premier League or the EPL outside England, is the top level of the English football league system. Contested by 20 clubs, it operates on a system of promotion and relegation with the English Football League (EFL). Seasons run from August to May with each team playing 38 matches (playing all 19 other teams both home and away)[3]. Here are the origin terms that we use in our project:

Relevant Terms

- FTHG: The total number of goals scored by the home team during the match at full time.
- FTAG: The total number of goals scored by the away team during the match at full time.
- FTR: The full time result, denoted as 'H' for home team win, 'A' for away team win, or 'D' for draw

2019–2020 Premier League

The 2019–20 Premier League is the 28th season of the Premier League. Different from the past, it is currently halted until at least 30 April 2020, following a decision on 13 March 2020 by the Premier League to suspend the league after a number of players and other club staff became ill due to the coronavirus pandemic[4].

Problem Statement

Designed a Ranking System for the EPL

- Get the EPL season data from official website, and convert it to usable data
- Build Flexible, Dynamic Data Model from source file for the following rank step.
- Simulate reliable data for predicting the probability to each team pairs for the result.
- Generate final ranking table with meaningful and logical ranking method.
- Generate table for each team showing their winning probability to other teams.

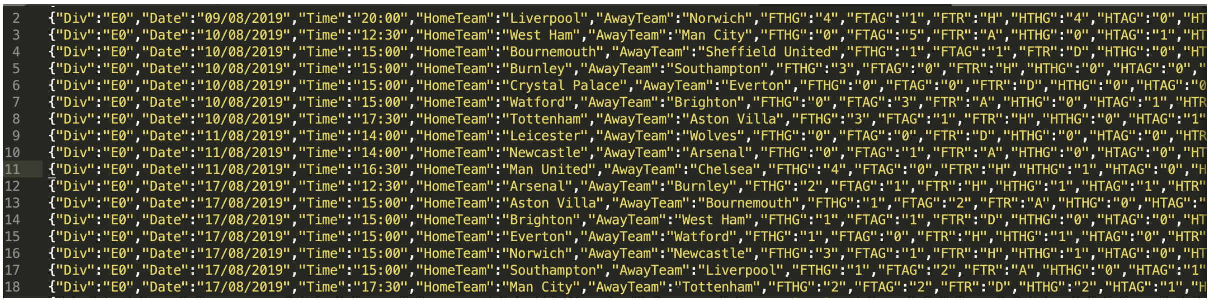
Project Description

This project is mainly about developing a ranking system which is able to evaluate the following expression where x_i, x_j are elements from a set of competing elements *X: P(x_i, x_j)* where *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.

We take teams as an object to complete the rank table. First, we counted the results of all matches and ranked team according to the number of wins. Then, we counted the goal difference of each team, and then sort them from large to small. Additionally, this project also includes the evaluation mentioned above, which is about pair comparison. We depends on

Datasets used in the Project

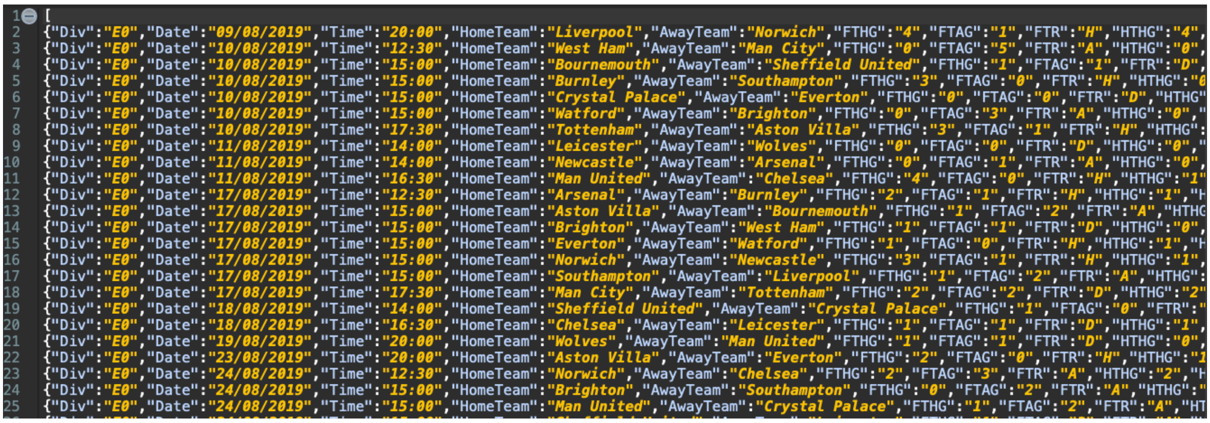
Json Data Set: season-1920.json (from www.football-data.co.uk/englandm.php)



```
{"Div":"E0","Date":"09/08/2019","Time":"20:00","HomeTeam":"Liverpool","AwayTeam":"Norwich","FTHG":"4","FTAG":"1","FTR":"H","HTHG":"4","HTAG":"0","HT
{"Div":"E0","Date":"10/08/2019","Time":"12:30","HomeTeam":"West Ham","AwayTeam":"Man City","FTHG":"0","FTAG":"5","FTR":"A","HTHG":"0","HTAG":"1","HT
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Bournemouth","AwayTeam":"Sheffield United","FTHG":"1","FTAG":"1","FTR":"D","HTHG":"0","HT
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Burnley","AwayTeam":"Southampton","FTHG":"3","FTAG":"0","FTR":"H","HTHG":"0","HTAG":"0","
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Crystal Palace","AwayTeam":"Everton","FTHG":"0","FTAG":"0","FTR":"D","HTHG":"0","HTAG":"0
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Watford","AwayTeam":"Brighton","FTHG":"0","FTAG":"3","FTR":"A","HTHG":"0","HTAG":"1","HTR
{"Div":"E0","Date":"10/08/2019","Time":"17:30","HomeTeam":"Tottenham","AwayTeam":"Aston Villa","FTHG":"3","FTAG":"1","FTR":"H","HTHG":"0","HTAG":"1"
{"Div":"E0","Date":"11/08/2019","Time":"14:00","HomeTeam":"Leicester","AwayTeam":"Wolves","FTHG":"0","FTAG":"0","FTR":"D","HTHG":"0","HTAG":"0","HTR
{"Div":"E0","Date":"11/08/2019","Time":"14:00","HomeTeam":"Newcastle","AwayTeam":"Arsenal","FTHG":"0","FTAG":"1","FTR":"A","HTHG":"0","HTAG":"0","HT
{"Div":"E0","Date":"11/08/2019","Time":"16:30","HomeTeam":"Man United","AwayTeam":"Chelsea","FTHG":"4","FTAG":"0","FTR":"H","HTHG":"1","HTAG":"0","H
{"Div":"E0","Date":"17/08/2019","Time":"12:30","HomeTeam":"Arsenal","AwayTeam":"Burnley","FTHG":"2","FTAG":"1","FTR":"H","HTHG":"1","HTAG":"1","HTR"
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Aston Villa","AwayTeam":"Bournemouth","FTHG":"1","FTAG":"2","FTR":"A","HTHG":"0","HTAG":"
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Brighton","AwayTeam":"West Ham","FTHG":"1","FTAG":"1","FTR":"D","HTHG":"0","HTAG":"0","HT
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Everton","AwayTeam":"Watford","FTHG":"1","FTAG":"0","FTR":"H","HTHG":"1","HTAG":"0","HTR"
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Norwich","AwayTeam":"Newcastle","FTHG":"3","FTAG":"1","FTR":"H","HTHG":"1","HTAG":"0","HT
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Southampton","AwayTeam":"Liverpool","FTHG":"1","FTAG":"2","FTR":"A","HTHG":"0","HTAG":"1"
{"Div":"E0","Date":"17/08/2019","Time":"17:30","HomeTeam":"Man City","AwayTeam":"Tottenham","FTHG":"2","FTAG":"2","FTR":"D","HTHG":"2","HTAG":"1","H
```

Figure2: Part of season-1920.jason file screenshot

In order to test our system’s correctness, we also use other season’s data like season-1819.json:



```
[
{"Div":"E0","Date":"09/08/2019","Time":"20:00","HomeTeam":"Liverpool","AwayTeam":"Norwich","FTHG":"4","FTAG":"1","FTR":"H","HTHG":"4",
{"Div":"E0","Date":"10/08/2019","Time":"12:30","HomeTeam":"West Ham","AwayTeam":"Man City","FTHG":"0","FTAG":"5","FTR":"A","HTHG":"0",
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Bournemouth","AwayTeam":"Sheffield United","FTHG":"1","FTAG":"1","FTR":"D",
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Burnley","AwayTeam":"Southampton","FTHG":"3","FTAG":"0","FTR":"H","HTHG":"0
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Crystal Palace","AwayTeam":"Everton","FTHG":"0","FTAG":"0","FTR":"D","HTHG"
{"Div":"E0","Date":"10/08/2019","Time":"15:00","HomeTeam":"Watford","AwayTeam":"Brighton","FTHG":"0","FTAG":"3","FTR":"A","HTHG":"0","
{"Div":"E0","Date":"10/08/2019","Time":"17:30","HomeTeam":"Tottenham","AwayTeam":"Aston Villa","FTHG":"3","FTAG":"1","FTR":"H","HTHG":
{"Div":"E0","Date":"11/08/2019","Time":"14:00","HomeTeam":"Leicester","AwayTeam":"Wolves","FTHG":"0","FTAG":"0","FTR":"D","HTHG":"0","
{"Div":"E0","Date":"11/08/2019","Time":"14:00","HomeTeam":"Newcastle","AwayTeam":"Arsenal","FTHG":"0","FTAG":"1","FTR":"A","HTHG":"0",
{"Div":"E0","Date":"11/08/2019","Time":"16:30","HomeTeam":"Man United","AwayTeam":"Chelsea","FTHG":"4","FTAG":"0","FTR":"H","HTHG":"1"
{"Div":"E0","Date":"17/08/2019","Time":"12:30","HomeTeam":"Arsenal","AwayTeam":"Burnley","FTHG":"2","FTAG":"1","FTR":"H","HTHG":"1","H
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Aston Villa","AwayTeam":"Bournemouth","FTHG":"1","FTAG":"2","FTR":"A","HTHG
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Brighton","AwayTeam":"West Ham","FTHG":"1","FTAG":"1","FTR":"D","HTHG":"0",
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Everton","AwayTeam":"Watford","FTHG":"1","FTAG":"0","FTR":"H","HTHG":"1","H
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Norwich","AwayTeam":"Newcastle","FTHG":"3","FTAG":"1","FTR":"H","HTHG":"1",
{"Div":"E0","Date":"17/08/2019","Time":"15:00","HomeTeam":"Southampton","AwayTeam":"Liverpool","FTHG":"1","FTAG":"2","FTR":"A","HTHG":
{"Div":"E0","Date":"17/08/2019","Time":"17:30","HomeTeam":"Man City","AwayTeam":"Tottenham","FTHG":"2","FTAG":"2","FTR":"D","HTHG":"2"
{"Div":"E0","Date":"18/08/2019","Time":"14:00","HomeTeam":"Sheffield United","AwayTeam":"Crystal Palace","FTHG":"1","FTAG":"0","FTR":"
{"Div":"E0","Date":"18/08/2019","Time":"16:30","HomeTeam":"Chelsea","AwayTeam":"Leicester","FTHG":"1","FTAG":"1","FTR":"D","HTHG":"1",
{"Div":"E0","Date":"19/08/2019","Time":"20:00","HomeTeam":"Wolves","AwayTeam":"Man United","FTHG":"1","FTAG":"1","FTR":"D","HTHG":"0",
{"Div":"E0","Date":"23/08/2019","Time":"20:00","HomeTeam":"Aston Villa","AwayTeam":"Everton","FTHG":"2","FTAG":"0","FTR":"H","HTHG":"1
{"Div":"E0","Date":"24/08/2019","Time":"12:30","HomeTeam":"Norwich","AwayTeam":"Chelsea","FTHG":"2","FTAG":"3","FTR":"A","HTHG":"2","H
{"Div":"E0","Date":"24/08/2019","Time":"15:00","HomeTeam":"Brighton","AwayTeam":"Southampton","FTHG":"0","FTAG":"2","FTR":"A","HTHG":"
{"Div":"E0","Date":"24/08/2019","Time":"15:00","HomeTeam":"Man United","AwayTeam":"Crystal Palace","FTHG":"1","FTAG":"2","FTR":"A","HT
```

Figure2: Part of season-1819.jason file screenshot

Implementation

A. Build Data Model

Here is the outline of project in Eclipse IDE:

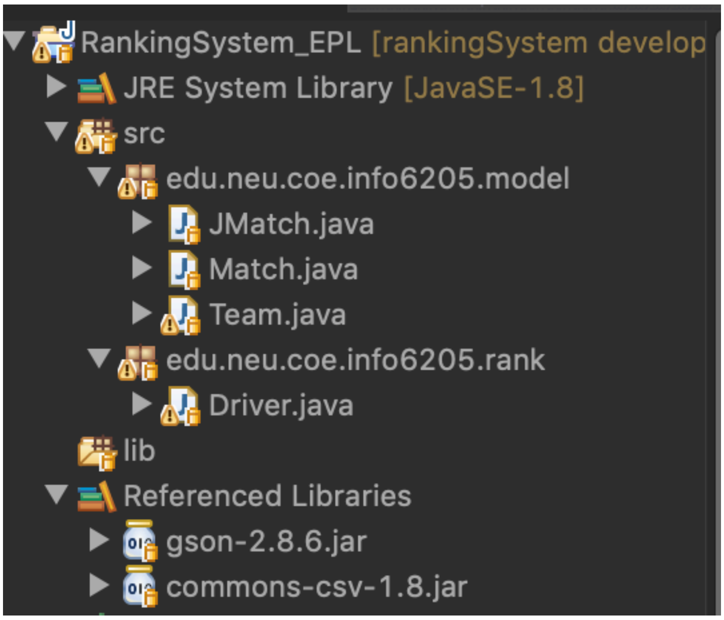
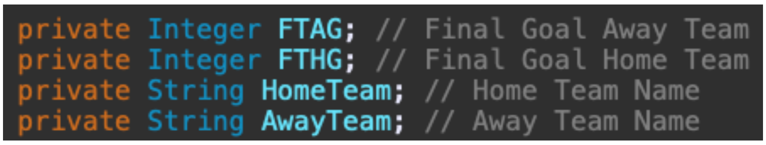

RankingSystem_EPL [rankingSystem develop
JRE System Library [JavaSE-1.8]
src
edu.neu.coe.info6205.model
JMatch.java
Match.java
Team.java
edu.neu.coe.info6205.rank
Driver.java
lib
Referenced Libraries
gson-2.8.6.jar
commons-csv-1.8.jar

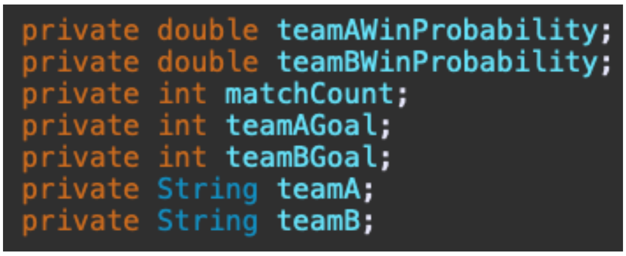
Figure4: Project outline screenshot

As known from the screenshot, we has two package, one is for the data model, the other is in charge of drive the program and do the rank. In model section, *JMatch.java* is use to input the origin data from json file, Match.java is the match object we exactly used in data process, Team.java is the team object, which includes various attributes like the following pictures show.



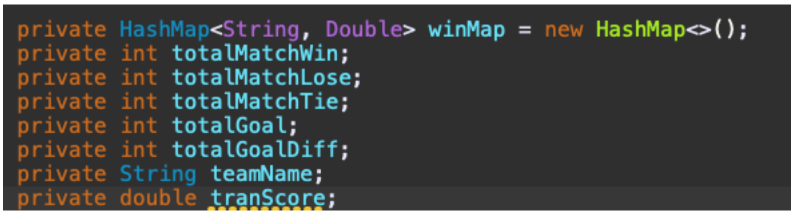
```java
private Integer FTAG; // Final Goal Away Team
private Integer FTHG; // Final Goal Home Team
private String HomeTeam; // Home Team Name
private String AwayTeam; // Away Team Name
```

Figure5: JMatch's attributes screenshot



```java
private double teamAWinProbability;
private double teamBWinProbability;
private int matchCount;
private int teamAGoal;
private int teamBGoal;
private String teamA;
private String teamB;
```

Figure6: Match's attributes screenshot



```java
private HashMap<String, Double> winMap = new HashMap<>();
private int totalMatchWin;
private int totalMatchLose;
private int totalMatchTie;
private int totalGoal;
private int totalGoalDiff;
private String teamName;
private double tranScore;
```

Figure7: Match's attributes screenshot

B. *Poisson distribution*:

a. If the value of discrete variable X can be 0,1,2….and the probability of each value is

b. $$P(X=k)=\frac{\lambda^k e^{-\lambda}}{k!}, k=0,1,2,\ldots$$ ($\lambda > 0$), we call X obey the Poisson Distribution with variable λ . λ is the expectation.

C. We have known that the goal of football game match Poisson Distribution. Then we using the Poisson distribution model and the average number of goals per game, calculate the theoretical calculation distribution of the number of goals.

D. As we have calculated the number of goals through the previous process. And we know the exact distribution of the event that we are going to simulate. We can then do the Monte Carlo method. Randomly generate gaming goals as well as result following Poisson Distribution of every λ , and calculating the probability of win, tie and lose.

Solution Flowchart

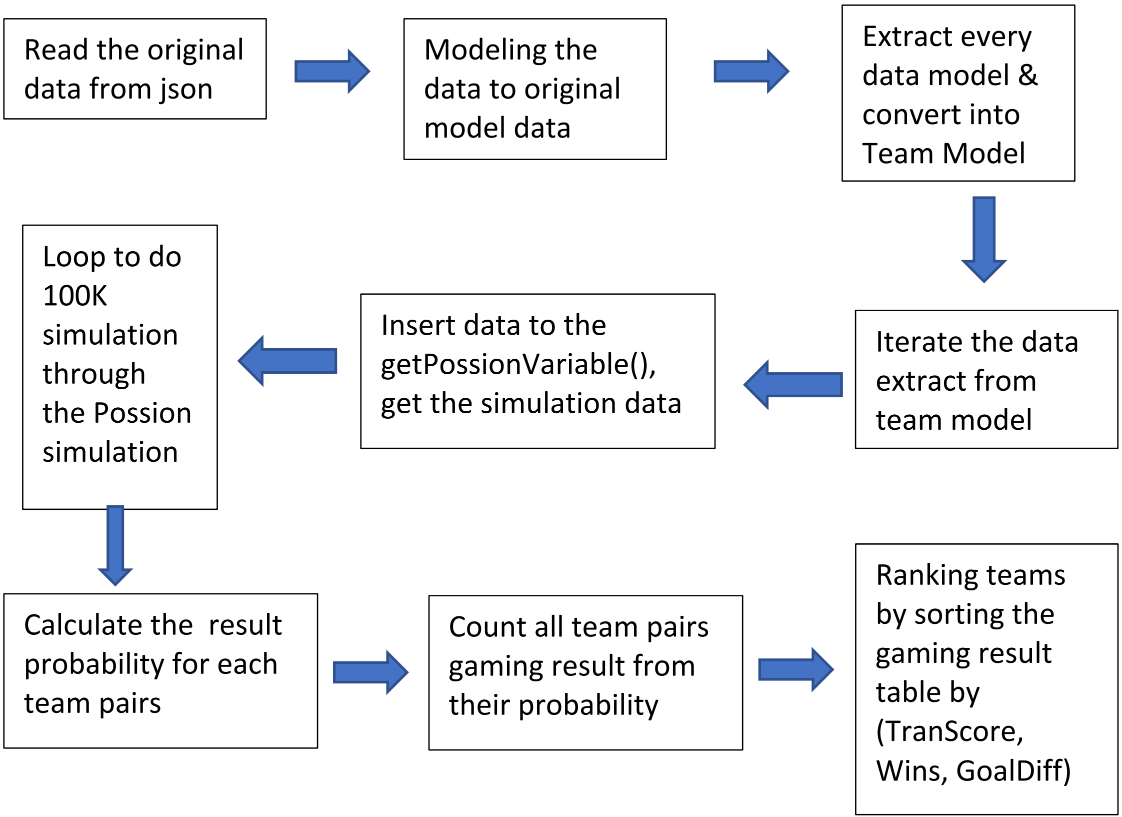
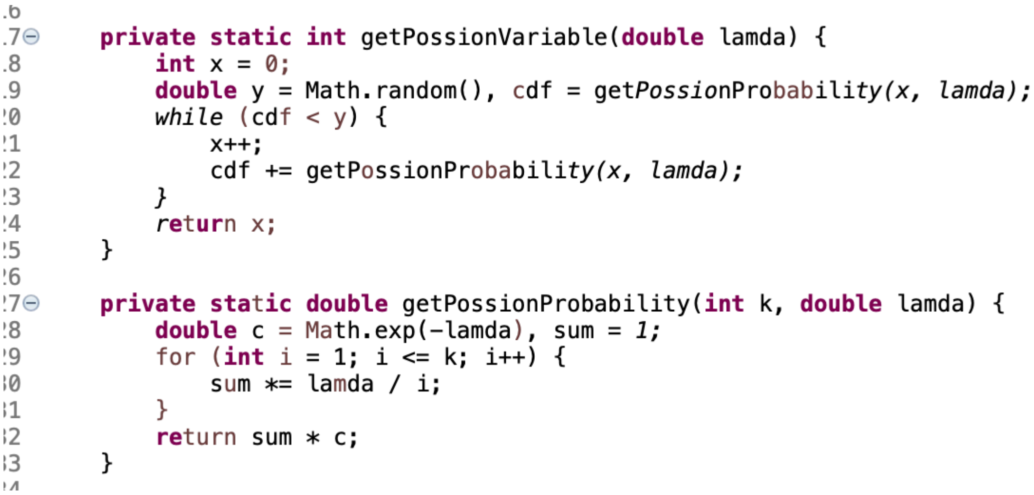

Read the original data from json
Modeling the data to original model data
Extract every data model & convert into Team Model
Iterate the data extract from team model
Insert data to the getPossionVariable(), get the simulation data
Loop to do 100K simulation through the Possion simulation
Calculate the result probability for each team pairs
Count all team pairs gaming result from their probability
Ranking teams by sorting the gaming result table by (TranScore, Wins, GoalDiff)

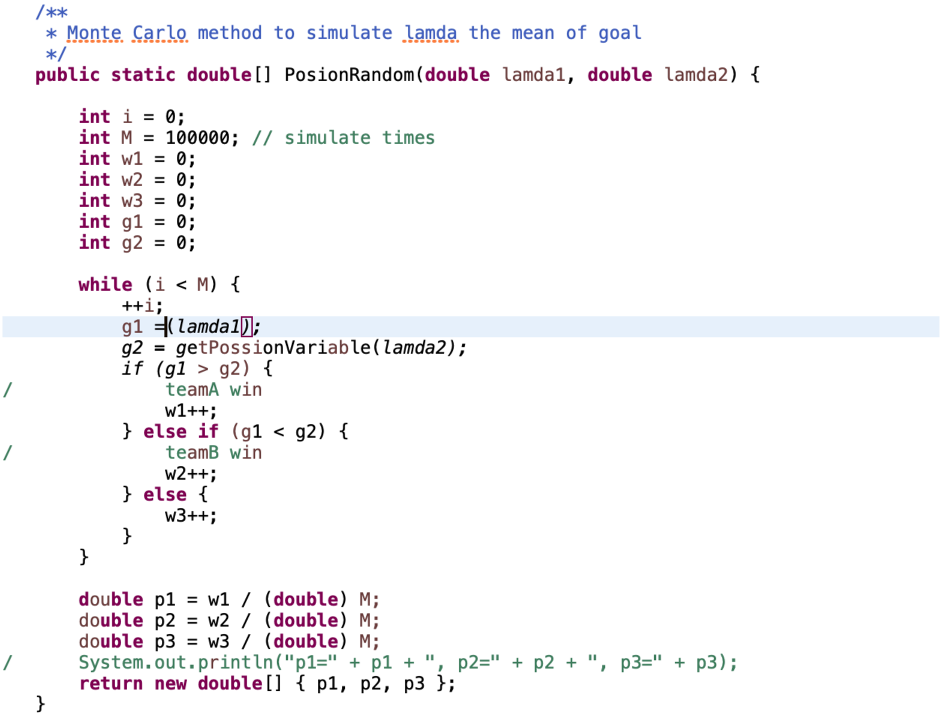
Figure8: Implementation step in Flowchart

Implementation of get Poisson Variable from Lamda



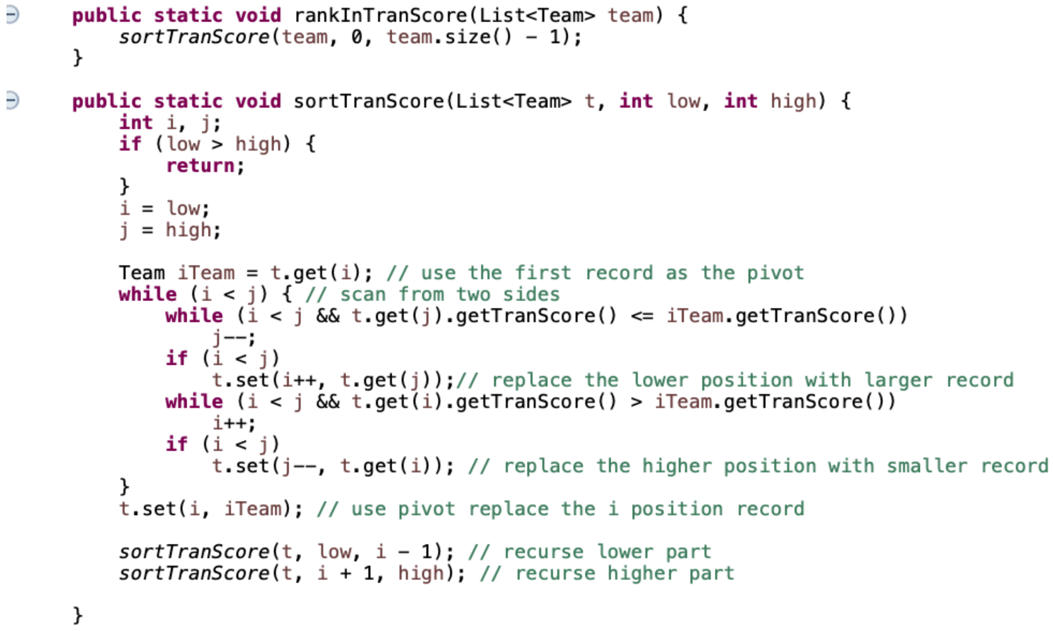
```
    private static int getPossionVariable(double lamda) {
        int x = 0;
        double y = Math.random(), cdf = getPossionProbability(x, lamda);
        while (cdf < y) {
            x++;
            cdf += getPossionProbability(x, lamda);
        }
        return x;
    }

    private static double getPossionProbability(int k, double lamda) {
        double c = Math.exp(-lamda), sum = 1;
        for (int i = 1; i <= k; i++) {
            sum *= lamda / i;
        }
        return sum * c;
    }
```

Implementation of Monte Carlo method

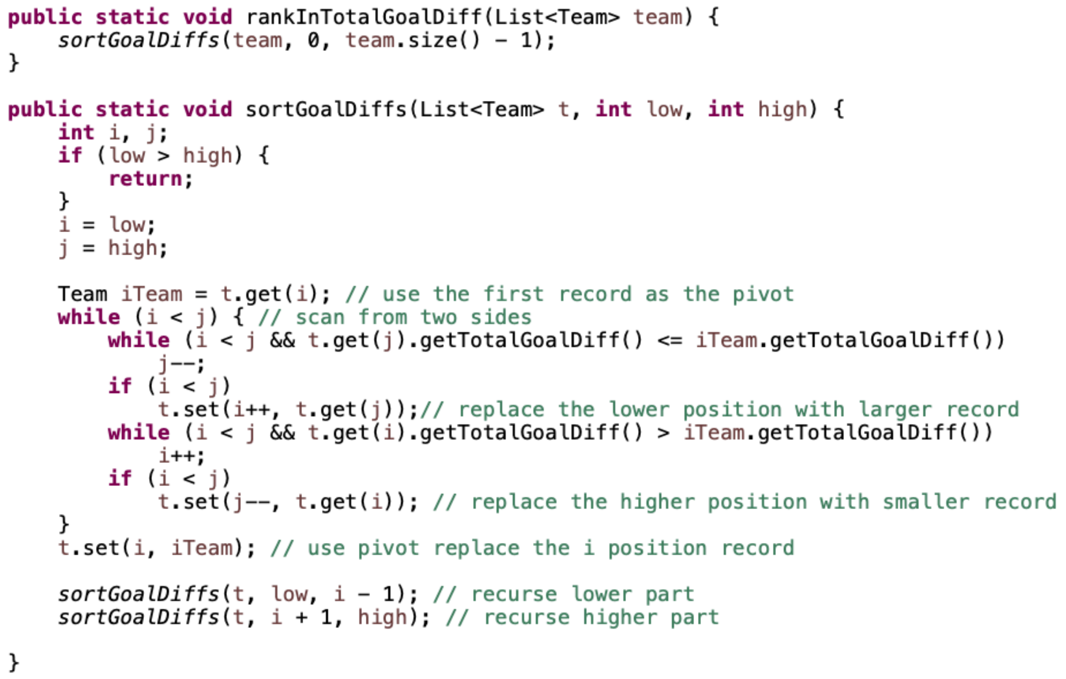


```
/**
 * Monte Carlo method to simulate lamda the mean of goal
 */
public static double[] PosionRandom(double lamda1, double lamda2) {

    int i = 0;
    int M = 100000; // simulate times
    int w1 = 0;
    int w2 = 0;
    int w3 = 0;
    int g1 = 0;
    int g2 = 0;

    while (i < M) {
        ++i;
        g1 =(lamda1);
        g2 = getPossionVariable(lamda2);
        if (g1 > g2) {
/               teamA win
            w1++;
        } else if (g1 < g2) {
/               teamB win
            w2++;
        } else {
            w3++;
        }
    }

    double p1 = w1 / (double) M;
    double p2 = w2 / (double) M;
    double p3 = w3 / (double) M;
/   System.out.println("p1=" + p1 + ", p2=" + p2 + ", p3=" + p3);
    return new double[] { p1, p2, p3 };
}
```

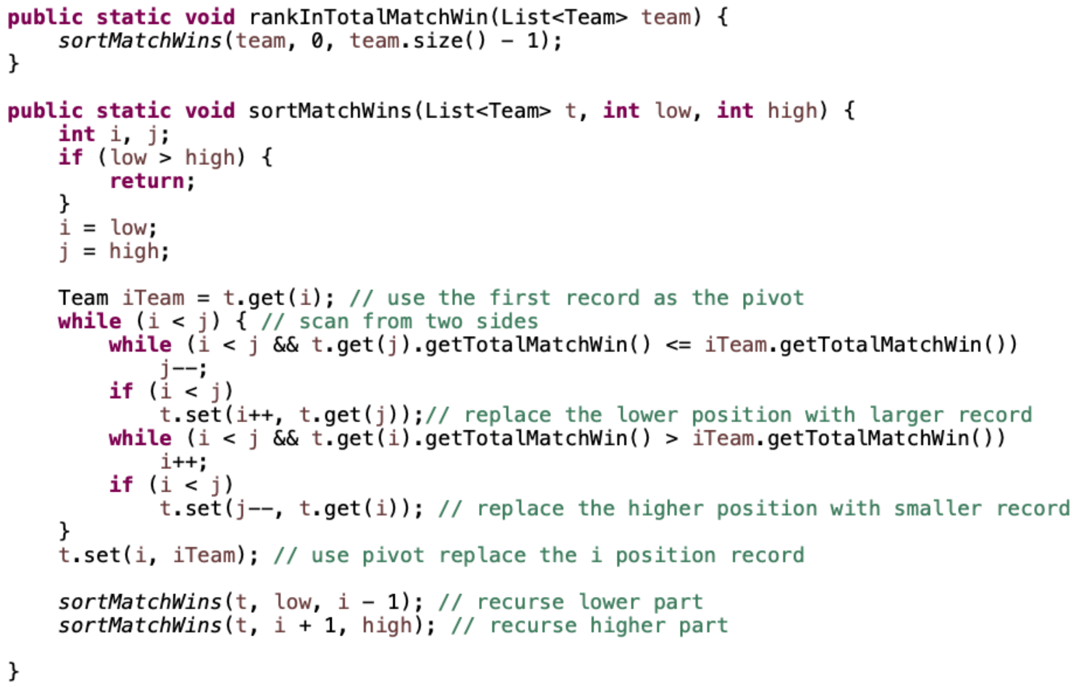
Implementation of Sorting



```
public static void rankInTranScore(List<Team> team) {
    sortTranScore(team, 0, team.size() - 1);
}

public static void sortTranScore(List<Team> t, int low, int high) {
    int i, j;
    if (low > high) {
        return;
    }
    i = low;
    j = high;

    Team iTeam = t.get(i); // use the first record as the pivot
    while (i < j) { // scan from two sides
        while (i < j && t.get(j).getTranScore() <= iTeam.getTranScore())
            j--;
        if (i < j)
            t.set(i++, t.get(j));// replace the lower position with larger record
        while (i < j && t.get(i).getTranScore() > iTeam.getTranScore())
            i++;
        if (i < j)
            t.set(j--, t.get(i)); // replace the higher position with smaller record
    }
    t.set(i, iTeam); // use pivot replace the i position record

    sortTranScore(t, low, i - 1); // recurse lower part
    sortTranScore(t, i + 1, high); // recurse higher part

}
```



```
public static void rankInTotalGoalDiff(List<Team> team) {
    sortGoalDiffs(team, 0, team.size() - 1);
}

public static void sortGoalDiffs(List<Team> t, int low, int high) {
    int i, j;
    if (low > high) {
        return;
    }
    i = low;
    j = high;

    Team iTeam = t.get(i); // use the first record as the pivot
    while (i < j) { // scan from two sides
        while (i < j && t.get(j).getTotalGoalDiff() <= iTeam.getTotalGoalDiff())
            j--;
        if (i < j)
            t.set(i++, t.get(j));// replace the lower position with larger record
        while (i < j && t.get(i).getTotalGoalDiff() > iTeam.getTotalGoalDiff())
            i++;
        if (i < j)
            t.set(j--, t.get(i)); // replace the higher position with smaller record
    }
    t.set(i, iTeam); // use pivot replace the i position record

    sortGoalDiffs(t, low, i - 1); // recurse lower part
    sortGoalDiffs(t, i + 1, high); // recurse higher part

}
```

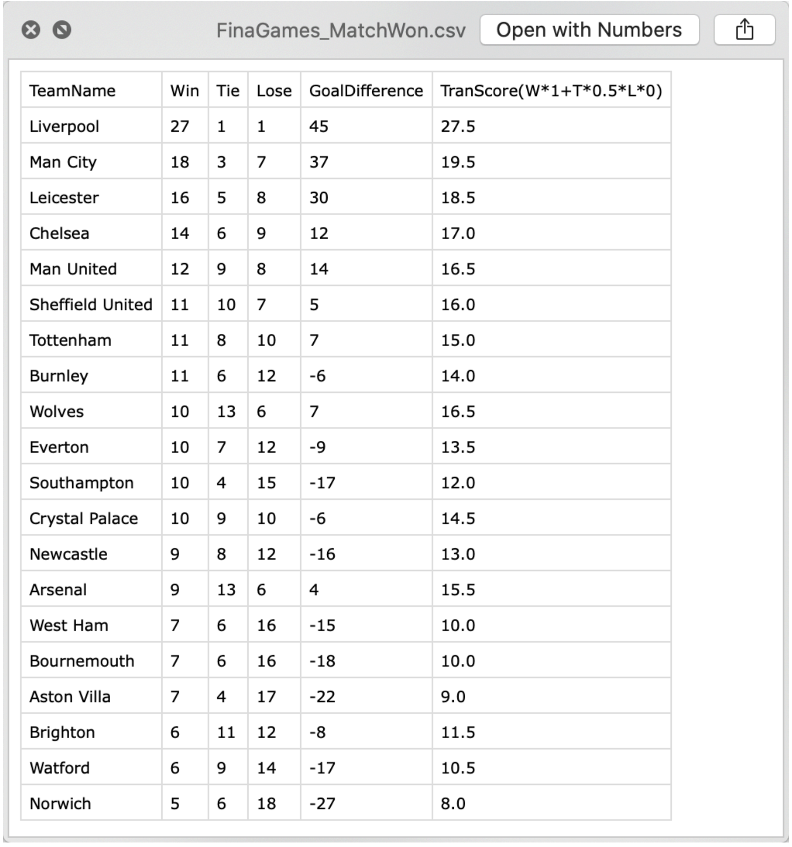


```
public static void rankInTotalMatchWin(List<Team> team) {
    sortMatchWins(team, 0, team.size() - 1);
}

public static void sortMatchWins(List<Team> t, int low, int high) {
    int i, j;
    if (low > high) {
        return;
    }
    i = low;
    j = high;

    Team iTeam = t.get(i); // use the first record as the pivot
    while (i < j) { // scan from two sides
        while (i < j && t.get(j).getTotalMatchWin() <= iTeam.getTotalMatchWin())
            j--;
        if (i < j)
            t.set(i++, t.get(j));// replace the lower position with larger record
        while (i < j && t.get(i).getTotalMatchWin() > iTeam.getTotalMatchWin())
            i++;
        if (i < j)
            t.set(j--, t.get(i)); // replace the higher position with smaller record
    }
    t.set(i, iTeam); // use pivot replace the i position record

    sortMatchWins(t, low, i - 1); // recurse lower part
    sortMatchWins(t, i + 1, high); // recurse higher part

}
```

Result & Analysis

Here are the result based on season 19-20 dataset, we divided it into 3 final table according to different rank method

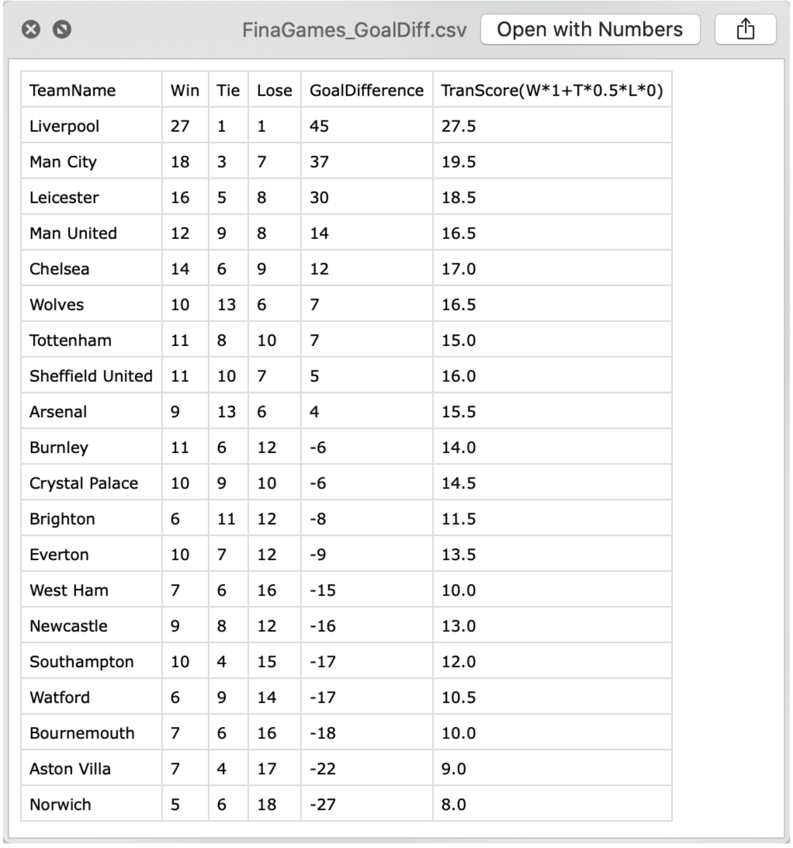
I. Ranking By Match wins



TeamName	Win	Tie	Lose	GoalDifference	TranScore(W*1+T*0.5*L*0)
Liverpool	27	1	1	45	27.5
Man City	18	3	7	37	19.5
Leicester	16	5	8	30	18.5
Chelsea	14	6	9	12	17.0
Man United	12	9	8	14	16.5
Sheffield United	11	10	7	5	16.0
Tottenham	11	8	10	7	15.0
Burnley	11	6	12	-6	14.0
Wolves	10	13	6	7	16.5
Everton	10	7	12	-9	13.5
Southampton	10	4	15	-17	12.0
Crystal Palace	10	9	10	-6	14.5
Newcastle	9	8	12	-16	13.0
Arsenal	9	13	6	4	15.5
West Ham	7	6	16	-15	10.0
Bournemouth	7	6	16	-18	10.0
Aston Villa	7	4	17	-22	9.0
Brighton	6	11	12	-8	11.5
Watford	6	9	14	-17	10.5
Norwich	5	6	18	-27	8.0

Figure9: Rank Table in Match Wins

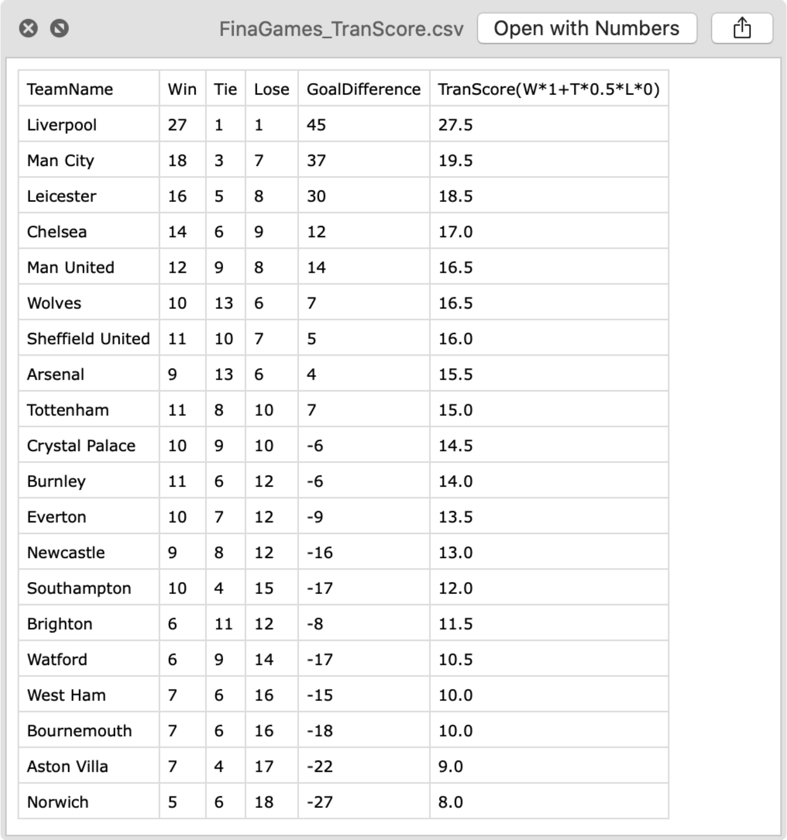
II. Ranking By Game Goal difference



TeamName	Win	Tie	Lose	GoalDifference	TranScore(W*1+T*0.5*L*0)
Liverpool	27	1	1	45	27.5
Man City	18	3	7	37	19.5
Leicester	16	5	8	30	18.5
Man United	12	9	8	14	16.5
Chelsea	14	6	9	12	17.0
Wolves	10	13	6	7	16.5
Tottenham	11	8	10	7	15.0
Sheffield United	11	10	7	5	16.0
Arsenal	9	13	6	4	15.5
Burnley	11	6	12	-6	14.0
Crystal Palace	10	9	10	-6	14.5
Brighton	6	11	12	-8	11.5
Everton	10	7	12	-9	13.5
West Ham	7	6	16	-15	10.0
Newcastle	9	8	12	-16	13.0
Southampton	10	4	15	-17	12.0
Watford	6	9	14	-17	10.5
Bournemouth	7	6	16	-18	10.0
Aston Villa	7	4	17	-22	9.0
Norwich	5	6	18	-27	8.0

Figure10: Rank Table in Goal Difference

III. Ranking By TranScore (Add weight to match: 1 for Wins, 0.5 for Ties, 0 for Loses)

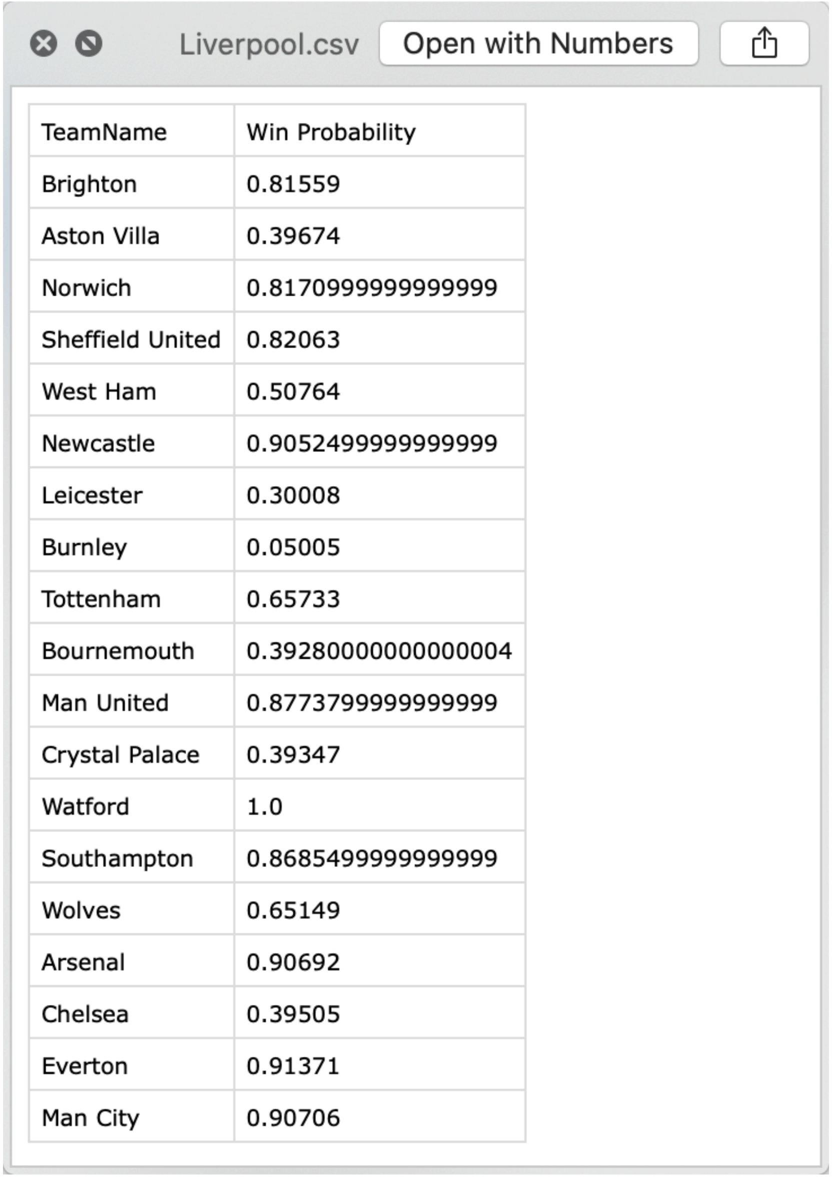


TeamName	Win	Tie	Lose	GoalDifference	TranScore(W*1+T*0.5*L*0)
Liverpool	27	1	1	45	27.5
Man City	18	3	7	37	19.5
Leicester	16	5	8	30	18.5
Chelsea	14	6	9	12	17.0
Man United	12	9	8	14	16.5
Wolves	10	13	6	7	16.5
Sheffield United	11	10	7	5	16.0
Arsenal	9	13	6	4	15.5
Tottenham	11	8	10	7	15.0
Crystal Palace	10	9	10	-6	14.5
Burnley	11	6	12	-6	14.0
Everton	10	7	12	-9	13.5
Newcastle	9	8	12	-16	13.0
Southampton	10	4	15	-17	12.0
Brighton	6	11	12	-8	11.5
Watford	6	9	14	-17	10.5
West Ham	7	6	16	-15	10.0
Bournemouth	7	6	16	-18	10.0
Aston Villa	7	4	17	-22	9.0
Norwich	5	6	18	-27	8.0

Figure10: Rank Table in TranScore

The following table shows the win possibility for one each team, against to the other 19 teams:

Team table (Ex: Liverpool vs. Other Teams)



TeamName	Win Probability
Brighton	0.81559
Aston Villa	0.39674
Norwich	0.8170999999999999
Sheffield United	0.82063
West Ham	0.50764
Newcastle	0.9052499999999999
Leicester	0.30008
Burnley	0.05005
Tottenham	0.65733
Bournemouth	0.39280000000000004
Man United	0.8773799999999999
Crystal Palace	0.39347
Watford	1.0
Southampton	0.8685499999999999
Wolves	0.65149
Arsenal	0.90692
Chelsea	0.39505
Everton	0.91371
Man City	0.90706

Figure11: Rank Table in Win Probability

1. Ranking tables sorting by Number of game wins, number of Goal difference, Transfer Score. (Transfer Score = 1*wins + 0.5*tie + 0* loss)
2. Tables showing for each team. The probability result of gaming with other teams are shown.
3. By analyzing the data table below, EPL season 2019-2020, the team Liverpool ranks #1 on the 3 kinds of final ranking table. But it will lost some games with high probability. The team Norwich lies on the bottom of the ranking tables.
4. When the predicted probability of occurrence is 0 or 1, it does not mean that the event will or must not occur, but it means that in the 100k times of simulated fitting data, its probability of occurrence is extremely small and infinitely close to 0.

5. Monte Carlo method(simulation)---Because the problem we solve (EPL) can be transformed into a certain random distribution of the number of features (Possion Distribution). Random sampling method is used to estimate the probability of the occurrence of random events, or the sampled digital characteristics to estimate the digital characteristics of random variables and use them as the solution to the problem.

Reference

[1] https://en.wikipedia.org/wiki/Ranking
[2] https://en.wikipedia.org/wiki/Poisson_distribution
[3] https://www.premierleague.com/
[4] https://en.wikipedia.org/wiki/2019%E2%80%9320_Premier_League
[5] https://en.wikipedia.org/wiki/Monte_Carlo_method