

Ranking System Project Report

Course: INFO 6205.

Project Name: NBA Ranking System.

Team Leader: Jin Li.

Team Member: 1.

Professor: Prof. Robin Hillyard.

Introduction

What is Ranking System?

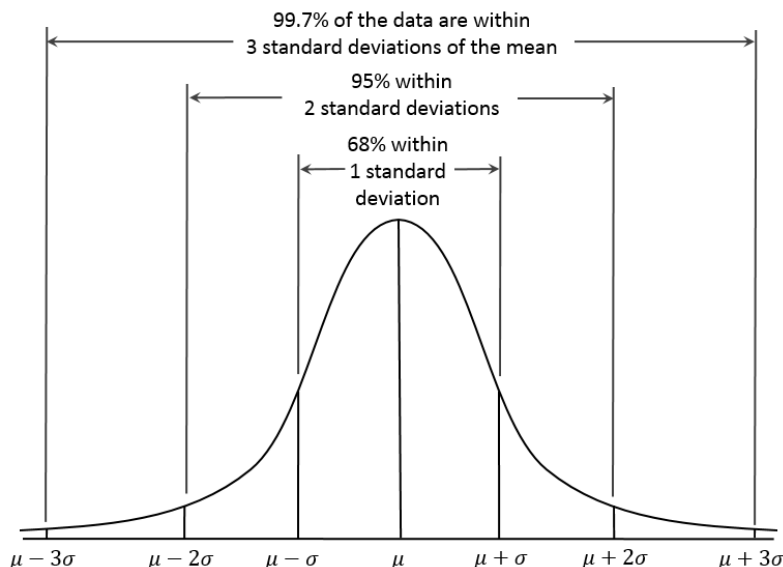
A ranking is a relationship between a set of items such that, for any two items, the first is either 'ranked higher than', 'ranked lower than' or 'ranked equal to' the second. ^[1] Ranking System is a system that can analyze a list of items and give each of them a ranking value according to certain criteria.

Ranking system exists in lots of areas. In a school, students are in a ranking system and sorting by their grades. In a company, employees are in a ranking system and sorting by their performance. It exists widely in sports. Every teams and athletes want to increase their rank value and win the championship.

In this project, I created an NBA Ranking System. I collected data for every game in this year and last year. These data were used for calculating the win probability of every team if they meet with another team in a head to head matchup. The value of win probability is a probability density function. And I used Normal Distribution as the format.

Probability Density Function:

The Probability Density Function is a function that whose value at any given sample (or point) in the sample space can be interpreted as providing a relative likelihood that the value of the random variable would equal that sample. ^[2]



The general form is:

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

In this formula, μ is the mean of the distribution, σ is the standard deviation of the distribution. When $\mu = 0$, and $\sigma = 1$, it is called the standard normal distribution.

If X and Y is a random variable of Probability Density Function, then:

- $U = X + Y \sim N(\mu_X + \mu_Y, \sigma^2_X + \sigma^2_Y)$
- $V = X - Y \sim N(\mu_X - \mu_Y, \sigma^2_X + \sigma^2_Y)$

I used this property to build my matchup code.

Aim of the Project:

Designed the NBA Ranking System:

- Create a matchup function and Calculate the win probability that X will beat Y if they meet with each other in a head to head matchup.
- Sum the probability of each team that beats rest of other teams and set each of them a rank value.
- Build a ranking table to list all teams by their rank value.
- Design UI to show the Ranking System.
- Test the result.

Datasets used in the project:

In this project, datasets are 2018-2019 and 2019-2020 NBA season matchup summary from the NBA website.

- A part of matchup data in October 2018:

October Schedule Share & more ▼

Date	Start (ET)	Visitor/Neutral	PTS	Home/Neutral	PTS	Attend.	Notes
Tue, Oct 16, 2018	8:00p	Philadelphia 76ers	87	Boston Celtics	105	Box Score	18,624
Tue, Oct 16, 2018	10:30p	Oklahoma City Thunder	100	Golden State Warriors	108	Box Score	19,596
Wed, Oct 17, 2018	7:00p	Milwaukee Bucks	113	Charlotte Hornets	112	Box Score	17,889
Wed, Oct 17, 2018	7:00p	Brooklyn Nets	100	Detroit Pistons	103	Box Score	20,332
Wed, Oct 17, 2018	7:00p	Memphis Grizzlies	83	Indiana Pacers	111	Box Score	17,923
Wed, Oct 17, 2018	7:00p	Miami Heat	101	Orlando Magic	104	Box Score	19,191
Wed, Oct 17, 2018	7:30p	Atlanta Hawks	107	New York Knicks	126	Box Score	18,249
Wed, Oct 17, 2018	7:30p	Cleveland Cavaliers	104	Toronto Raptors	116	Box Score	19,915
Wed, Oct 17, 2018	8:00p	New Orleans Pelicans	131	Houston Rockets	112	Box Score	18,055
Wed, Oct 17, 2018	8:30p	Minnesota Timberwolves	108	San Antonio Spurs	112	Box Score	18,354
Wed, Oct 17, 2018	10:00p	Utah Jazz	123	Sacramento Kings	117	Box Score	17,583
Wed, Oct 17, 2018	10:30p	Denver Nuggets	107	Los Angeles Clippers	98	Box Score	19,068
Wed, Oct 17, 2018	10:30p	Dallas Mavericks	100	Phoenix Suns	121	Box Score	18,055
Thu, Oct 18, 2018	8:00p	Chicago Bulls	108	Philadelphia 76ers	127	Box Score	20,302
Thu, Oct 18, 2018	8:00p	Miami Heat	113	Washington Wizards	112	Box Score	20,409
Thu, Oct 18, 2018	10:30p	Los Angeles Lakers	119	Portland Trail Blazers	128	Box Score	19,996
Fri, Oct 19, 2018	7:00p	Charlotte Hornets	120	Orlando Magic	88	Box Score	17,668
Fri, Oct 19, 2018	7:30p	New York Knicks	105	Brooklyn Nets	107	Box Score	17,732
Fri, Oct 19, 2018	8:00p	Atlanta Hawks	117	Memphis Grizzlies	131	Box Score	17,019
Fri, Oct 19, 2018	8:00p	Cleveland Cavaliers	123	Minnesota Timberwolves	131	Box Score	18,978
Fri, Oct 19, 2018	8:00p	Sacramento Kings	129	New Orleans Pelicans	149	Box Score	18,337
Fri, Oct 19, 2018	8:00p	Boston Celtics	101	Toronto Raptors	113	Box Score	19,800
Fri, Oct 19, 2018	8:30p	Indiana Pacers	101	Milwaukee Bucks	118	Box Score	17,341
Fri, Oct 19, 2018	10:30p	Oklahoma City Thunder	92	Los Angeles Clippers	108	Box Score	14,816
Fri, Oct 19, 2018	10:30p	Golden State Warriors	124	Utah Jazz	123	Box Score	18,306
Sat, Oct 20, 2018	7:00p	Brooklyn Nets	112	Indiana Pacers	132	Box Score	17,007
Sat, Oct 20, 2018	7:00p	Toronto Raptors	117	Washington Wizards	113	Box Score	16,185
Sat, Oct 20, 2018	7:30p	Boston Celtics	103	New York Knicks	101	Box Score	19,427
Sat, Oct 20, 2018	7:30p	Orlando Magic	115	Philadelphia 76ers	116	Box Score	20,300
Sat, Oct 20, 2018	8:00p	Detroit Pistons	118	Chicago Bulls	116	Box Score	21,289

- All of matchup data are saved in a CSV file named “data.csv”. In this project, I read and used data from this file.

	Date	Time	Visitor	Home	Visitor PTS	Home PTS	Attend.	Notes
1	Tue Oct 22	2019,8:00p	New Orleans Pelicans	122	Toronto Raptors	130	Box Score	OT,20787,
2	Tue Oct 22	2019,10:30p	Los Angeles Lakers	102	Los Angeles Clippers	112	Box Score	,19068,
3	Wed Oct 23	2019,7:00p	Chicago Bulls	125	Charlotte Hornets	126	Box Score	,15424,
4	Wed Oct 23	2019,7:00p	Detroit Pistons	119	Indiana Pacers	110	Box Score	,17923,
5	Wed Oct 23	2019,7:00p	Cleveland Cavaliers	85	Orlando Magic	94	Box Score	,18846,
6	Wed Oct 23	2019,7:30p	Minnesota Timberwolves	127	Brooklyn Nets	126	Box Score	OT,17732,
7	Wed Oct 23	2019,7:30p	Memphis Grizzlies	101	Miami Heat	120	Box Score	,19600,
8	Wed Oct 23	2019,7:30p	Boston Celtics	93	Philadelphia 76ers	107	Box Score	,20422,
9	Wed Oct 23	2019,8:30p	Washington Wizards	100	Dallas Mavericks	108	Box Score	,19816,
10	Wed Oct 23	2019,8:30p	New York Knicks	111	San Antonio Spurs	120	Box Score	,18354,
11	Wed Oct 23	2019,9:00p	Oklahoma City Thunder	95	Utah Jazz	100	Box Score	,18306,
12	Wed Oct 23	2019,10:00p	Sacramento Kings	95	Phoenix Suns	124	Box Score	,18055,
13	Wed Oct 23	2019,10:00p	Denver Nuggets	108	Portland Trail Blazers	100	Box Score	,19991,
14	Thu Oct 24	2019,7:00p	Atlanta Hawks	117	Detroit Pistons	100	Box Score	,20338,
15	Thu Oct 24	2019,8:00p	Milwaukee Bucks	117	Houston Rockets	111	Box Score	,18055,
16	Thu Oct 24	2019,10:30p	Los Angeles Clippers	141	Golden State Warriors	122	Box Score	,18064,
17	Fri Oct 25	2019,7:00p	Toronto Raptors	106	Boston Celtics	112	Box Score	,18624,
18	Fri Oct 25	2019,7:00p	Minnesota Timberwolves	121	Charlotte Hornets	99	Box Score	,14879,
19	Fri Oct 25	2019,7:30p	New York Knicks	109	Brooklyn Nets	113	Box Score	,17732,
20	Fri Oct 25	2019,8:00p	Chicago Bulls	110	Memphis Grizzlies	102	Box Score	,17794,
21	Fri Oct 25	2019,8:00p	Dallas Mavericks	123	New Orleans Pelicans	116	Box Score	,17027,
22	Fri Oct 25	2019,8:00p	Washington Wizards	97	Oklahoma City Thunder	85	Box Score	,18203,
23	Fri Oct 25	2019,9:00p	Phoenix Suns	107	Denver Nuggets	108	Box Score	OT,19557,
24	Fri Oct 25	2019,10:00p	Portland Trail Blazers	122	Sacramento Kings	112	Box Score	,17583,
25	Fri Oct 25	2019,10:30p	Utah Jazz	86	Los Angeles Lakers	95	Box Score	,18997,
26	Sat Oct 26	2019,5:00p	Miami Heat	131	Milwaukee Bucks	126	Box Score	OT,17467,
27	Sat Oct 26	2019,7:00p	Philadelphia 76ers	117	Detroit Pistons	111	Box Score	,16207,
28	Sat Oct 26	2019,7:30p	Orlando Magic	99	Atlanta Hawks	103	Box Score	,17078,
29	Sat Oct 26	2019,7:30p	Boston Celtics	118	New York Knicks	95	Box Score	,19812,
30	Sat Oct 26	2019,8:00p	Toronto Raptors	108	Chicago Bulls	84	Box Score	,21498,
31	Sat Oct 26	2019,8:00p	Indiana Pacers	99	Cleveland Cavaliers	110	Box Score	,19432,

Project Description:

Designed an NBA Ranking System using OOD and knowledge I learned from the course.

1. Business Part

- Download the matchup datasets from the NBA website and save as CSV files.
- Created a comparable class named Team to represent the team in NBA leagues. Added the Commons Math3 libraries and Imported Normal Distribution as its field. It had update distribution function that could update the μ and σ when a new matchup data being input. It could be compared by the rank value.
- Created a class named Matchup. It had a static function named probabilityWinner that could calculate the win probability of team one when team one meet with team two. It was implemented by using cumulativeProbability function from the NormalDistribution class.
- Created a class named Ranking for ranking all of teams. The list of teams was gotten from the “team.csv” file by using readTeamFromCsv() function. And it has a readFromCsv() function to read data from “data.csv” file.

2. Interface Part

- Used Netbeans IDE for designing UI.
- Created a class named MainJFrame class as the main class.
- Imported Java Swing package. Using JFrame, JPanel, JSplitPanel, JScrollPanel, JTable, JLabel and JButton for building the system interface.
- Implemented the Action Listener function in these classes.

Implementation:

- Team class implementation:

```

Team.java
65 //update the distribution if input a matchup result.
66 public void updateDistribution(String line) {
67     String[] fields = line.split(",");
68     if (name.equals(fields[2])) {
69         if (Integer.parseInt(fields[3]) > Integer.parseInt(fields[5])) {
70             win++;
71             addDataValue(1.0);
72         } else if (Integer.parseInt(fields[3]) < Integer.parseInt(fields[5])) {
73             addDataValue(0.0);
74         } else {
75             lose--;
76             addDataValue(-1.0);
77         }
78     } else {
79         if (Integer.parseInt(fields[3]) < Integer.parseInt(fields[5])) {
80             win++;
81             addDataValue(1.0);
82         } else if (fields[3].equals("0")) {
83             addDataValue(0.0);
84         } else {
85             lose--;
86             addDataValue(-1.0);
87         }
88     }
89 }
90 }
91
92 //update the mean and deviation of the distribution.
93 public void addDataValue(double x) {
94     dataNum++;
95     s = s + 1.0 * (dataNum - 1) / dataNum * (x - mean) * (x - mean);
96     mean = mean + (x - mean) / dataNum;
97     dev = Math.sqrt(s / (dataNum - 1));
98 }
99

```

- Matchup class implementation

```

Matchup.java
2
3 import java.util.List;
4
5
6 /**
7  * @author jinli
8  */
9 public class Matchup {
10
11
12 // calculate the win probability of team one when it meets with team two.
13 public static double probabilityWinner(Team one, Team two) {
14     double totalMean = one.mean - two.mean;
15     double totalDev = Math.sqrt((one.dev) * (one.dev) + (two.dev) * (two.dev));
16     NormalDistribution nd = new NormalDistribution(totalMean, totalDev);
17
18     return 1.0 - nd.cumulativeProbability(0.0);
19 }
20
21 public static double probabilityWinner(String one, String two, Ranking ranking) {
22     Team first = new Team();
23     Team second = new Team();
24     List<Team> teams = ranking.getTeams();
25     for (Team t : teams) {
26         String name = t.getName();
27         if (name.equals(one)) {
28             first = t;
29         }
30         else if (name.equals(two)) {
31             second = t;
32         }
33     }
34     return probabilityWinner(first, second);
35 }
36 }
37

```

```

100
101 // calculate the total win probability between this team and rest of other team.
102 public void calRank(List<Team> teams) {
103     for (Team t : teams) {
104         if (t.name.equals(this.name)) {
105             continue;
106         }
107         rank += Matchup.probabilityWinner(this, t);
108     }
109 }
110
111 //update the normaDistribution field.
112 public void updateNomalDis() {
113     this.normalDistribution = new NormalDistribution(mean, dev);
114 }
115
116
117
118 //compare teams by their rank.
119 @Override
120 public int compareTo(Team o) {
121     return Double.compare(this.rank, o.rank);
122 }
123
124 }
125 }
126

```

- Ranking class implementation:

```

Ranking.java
1 package rankingSystem;
2
3 import java.io.BufferedReader;
4
5
6 public class Ranking {
7
8     private ArrayList<Team> teams; // list of teams in NBA
9
10    public Ranking() {
11        teams = new ArrayList<>();
12        readTeamFromCsv(); //read team.
13        readFromCsv(); //read matchup data
14        calculateTeamRank(); //calculate team win probability
15        Collections.sort(teams, Collections.reverseOrder()); //sort list by team win probabili
16        updateRankValue(); //update the rank value
17    }
18
19    public ArrayList<Team> getTeams() {
20        return teams;
21    }
22
23    public void setTeams(ArrayList<Team> teams) {
24        this.teams = teams;
25    }
26
27    //read teams names from "team.csv"
28    public void readTeamFromCsv() {
29        String data = "team.csv";
30        try (BufferedReader br = new BufferedReader(new FileReader(data))) {
31            String line = null;
32            while ((line = br.readLine()) != null) {
33                Team newTeam = new Team(line);
34                teams.add(newTeam);
35            }
36        } catch (IOException ex) {
37
38        }
39    }
40
41 }

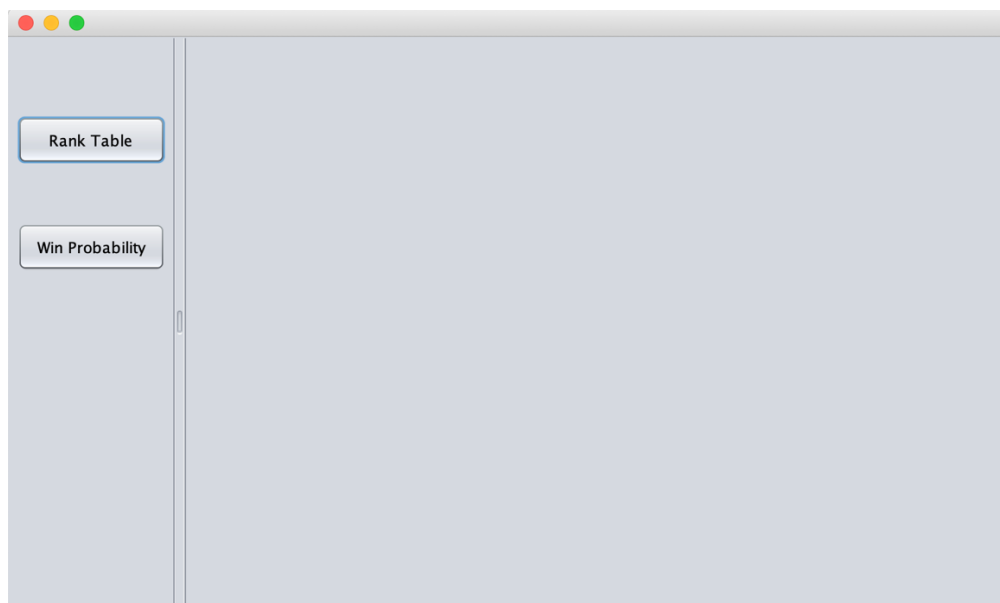
```

```
Ranking.java
40     ex.printStackTrace();
41 }
42 }
43
44 //read matchup data from "data.csv" or "19-data.csv".
45 public void readFromCsv() {
46     String data = "19-20data.csv";
47     try (BufferedReader br = new BufferedReader(new FileReader(data))) {
48         String line = null;
49         while ((line = br.readLine()) != null) {
50             String[] fields = line.split(",");
51             for (Team t : teams) {
52                 String teamName = t.getName();
53                 if (teamName.equals(fields[2]) || teamName.equals(fields[4])) {
54                     t.updateDistribution(line);
55                 }
56             }
57         }
58         for (Team t : teams) {
59             t.updateNomalDis();
60         }
61     } catch (IOException ex) {
62         ex.printStackTrace();
63     }
64 }
65
66 //calculate the team win probability.
67 private void calculateTeamRank() {
68     for (Team t : teams) {
69         t.calRank(teams);
70     }
71 }
72 }
73
74 //update the team rank value.
```

Output:

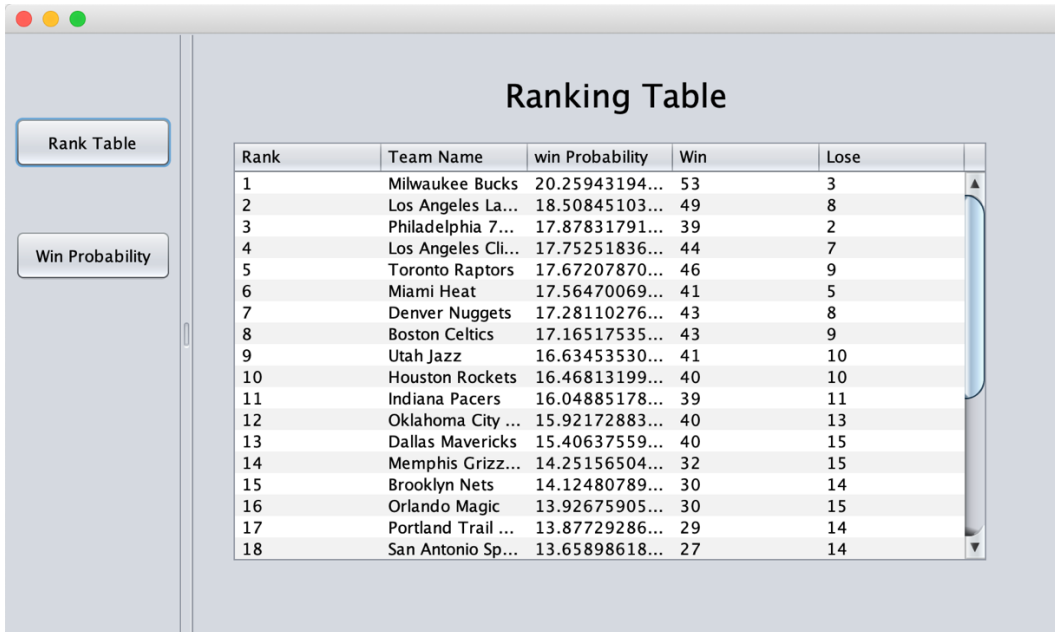
1.Home Page

When running the mainJFrame class file. The Ranking System application will show up. There are two buttons on the left side.



2. Rank Table

When click the Rank Table button, the ranking table will show up on the right side. It shows 30 teams sorting by their total win probability. It also shows the number of win and lose of each team.



Rank	Team Name	win Probability	Win	Lose
1	Milwaukee Bucks	20.25943194...	53	3
2	Los Angeles La...	18.50845103...	49	8
3	Philadelphia 7...	17.87831791...	39	2
4	Los Angeles Cli...	17.75251836...	44	7
5	Toronto Raptors	17.67207870...	46	9
6	Miami Heat	17.56470069...	41	5
7	Denver Nuggets	17.28110276...	43	8
8	Boston Celtics	17.16517535...	43	9
9	Utah Jazz	16.63453530...	41	10
10	Houston Rockets	16.46813199...	40	10
11	Indiana Pacers	16.04885178...	39	11
12	Oklahoma City ...	15.92172883...	40	13
13	Dallas Mavericks	15.40637559...	40	15
14	Memphis Grizz...	14.25156504...	32	15
15	Brooklyn Nets	14.12480789...	30	14
16	Orlando Magic	13.92675905...	30	15
17	Portland Trail ...	13.87729286...	29	14
18	San Antonio Sp...	13.65898618...	27	14

3. Win Probability

When click the Win Probability button, we can get the win probability of team one and team two.



Win Probability

Team1 Team2

Milwaukee Bucks VS Milwaukee Bucks OK

Team1 : Win Probability

Team2 : Win Probability

We can select different team one and team two. And click the OK button, the win probability of team 1 and team 2 will show up.



Conclusion and Future Implementations:

- From this project, I have learned to build a ranking system for a list of items. I also learned the basic concept of Probability Density Function and how to update the mean and deviation of the Normal Distribution.
- By designing UI, I have learned to use Swing class and design frames and forms using Netbeans IDE.
- In the future, I can analyze more matchup data. Also, I can add more function like get the win probability of two athletes or coaches in this system. In addition, the UI can be more attractive.

Reference:

1. Ranking. (n.d.). Retrieved from <http://www.merriam-webster.com/dictionary/ranking>
2. Grinstead, C. M., & Snell, J. L. (2012). Introduction to probability. American Mathematical Soc..