INFO 6205 Ranking Project Report

Letian Zheng 001058610

Zhonghao Gu 001083361

图片包含 游戏机, 食物, 杯子

描述已自动生成

Introduction

What is this project for?

Since the Premier League has been postponed due to the COVID-19, it is a huge challenge for people to imagine the result of the remaining matches and the final rank.

This project will use a mathematical method to help people to foresee the remaining season, find the winner of this season and teams maybe relegated.

In this project, we will use **Normal Distribution** to simulate the unprocessed matches. Normal Distribution is a type of continuous probability distribution for a real-valued random variable. The general form of its probability density function is The parameter is the mean or expectation of the distribution; and is its standard deviation. The general form of its probability density functions is:



Normal distributions are important in [statistics](https://en.wikipedia.org/wiki/Statistics) and are often used in the [natural](https://en.wikipedia.org/wiki/Natural_science) and [social sciences](https://en.wikipedia.org/wiki/Social_science) to represent real-valued [random variables](https://en.wikipedia.org/wiki/Random_variable) whose distributions are not known. Their importance is partly due to the [central limit theorem](https://en.wikipedia.org/wiki/Central_limit_theorem). It states that, under some conditions, the average of many samples (observations) of a [random variable](https://en.wikipedia.org/wiki/Random_variable) with finite [mean](https://en.wikipedia.org/wiki/Mean) and [variance](https://en.wikipedia.org/wiki/Variance) is itself a random variable whose distribution [converges](https://en.wikipedia.org/wiki/Convergence_in_distribution) to a normal distribution as the number of samples increases. Therefore, physical quantities that are expected to be the sum of many independent processes (such as [measurement errors](https://en.wikipedia.org/wiki/Measurement_error)) often have distributions that are nearly normal.

Moreover, Gaussian distributions have some unique properties that are valuable in analytic studies. For instance, any linear combination of a fixed collection of normal deviates is a normal deviate. Many results and methods (such as [propagation of uncertainty](https://en.wikipedia.org/wiki/Propagation_of_uncertainty) and [least squares](https://en.wikipedia.org/wiki/Least_squares) parameter fitting) can be derived analytically in explicit form when the relevant variables are normally distributed.

A general Normal Distributions looks like below:

图片包含 游戏机, 灯光

描述已自动生成

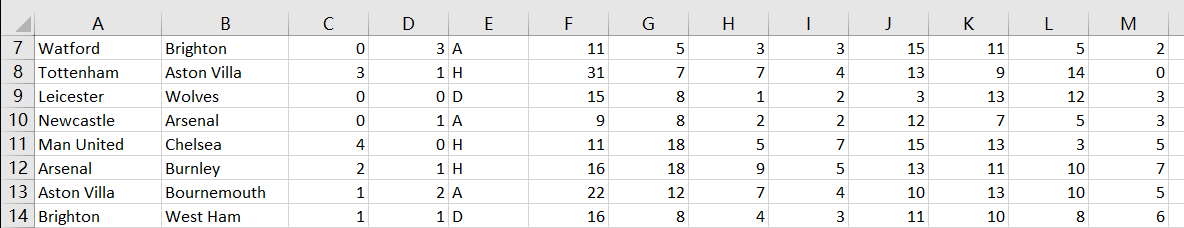
As we can see from the chart, those curves are average to the mid point, which can give us a possible result in our project.

However, football match cannot be easily predicted by one function. There are lots of other unpredictable related affairs to consider such as sudden injury, off-site factors or even board of directors change. Thus, this project is not quite comprehensive in some fields.

Data used in this project:

English Premier League 2019-2020 seasons result.(Please check attached file E0.CSV)

Below is a part of the data.

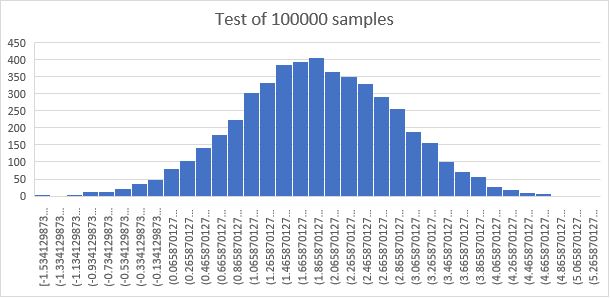


The reason we used only one season to foresee the rest matches’ result of this season is because we believe the former data does not influence anything to now. Every seasons, lots of player transfer to other team, some of them are so key to one team that may result a totally different season process.

Project Description

Use Normal Distributions to foresee the rank of the whole season and the remaining matches.

We will import the data, and use our algorithm to calculate those data, we use the goal difference between two teams to calculate the remaining matches’ result and goal difference. Here is the test result when the history goal difference is 2, and samples contain 100000 data:



As we can see, the test result do match Normal Distribution.

Finally we will export a csv file with the stimulation result and final rank.

Data Structure used in this project:

手机屏幕的截图

描述已自动生成

We create a Team class to store each team’s information. “String TeamName”represent the name of the team, “int totalGoalDifference” represent the sum of all matches’s goal difference . “int totalpoints” represent the rank points of the team. If a team win the math it will add 3 points, lose will get 0 points,else will add 1 point. We use a HashMap to store the game result(Goal Difference) with other teams.

And we can can use the hashmap to get a team’s average goal difference compete with other teams. And through this to calculate the Normal Distribution.

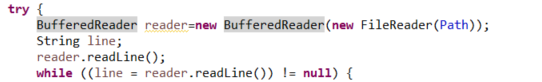
电脑屏幕的截图

描述已自动生成

We also use a TeamList class to help us manage the teams and help us rank the team after simulation. We override the sort method in Collections class to rank the teams. The rank based on the total points of each team, if two or more team’s total points are same, we will consider the team with more total Goal Difference will get high ranking.

Project Achieve

We use BufferedReader function to import the origin data.



The below data is the data we calculate from the imported current season statistics(29 rounds).

Rank Team Points Goal Difference

1. Liverpool 82 45

2. Man City 57 37

3. Leicester 53 30

4. Chelsea 48 12

5. Man United 45 14

6. Wolves 43 7

7. Sheffield United 43 5

8. Tottenham 41 7

9. Arsenal 40 4

10. Burnley 39 -6

11. Crystal Palace 39 -6

12. Everton 37 -9

13. Newcastle 35 -16

14. Southampton 34 -17

15. Brighton 29 -8

16. West Ham 27 -15

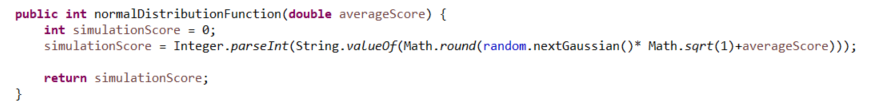
17. Watford 27 -17

18. Bournemouth 27 -18

19. Aston Villa 25 -22

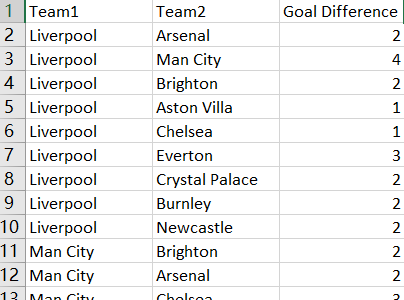
20. Norwich 21 -27

Then, we use Normal Distributions function



Let average goal difference be the mean and 1 be the variance, to simulate the remaining matches.

This is part of the matches simulation result:



(We will only show the goal difference since we use goal difference to draw the Noraml Distribution chart)

After simulation, we use below algorithm to Re-rank all teams:



And we will finally get the whole season’s rank:

1. Liverpool 109 68

2. Leicester 75 46

3. Man City 74 50

4. Chelsea 64 18

5. Sheffield United 58 7

6. Man United 56 12

7. Tottenham 56 10

8. Crystal Palace 54 -9

9. Newcastle 52 -16

10. Arsenal 51 -5

11. Wolves 50 1

12. Burnley 49 -12

13. Southampton 48 -14

14. Everton 43 -18

15. West Ham 42 -10

16. Bournemouth 38 -19

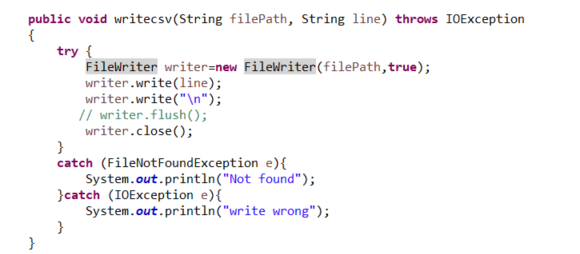
17. Brighton 37 -15

18. Aston Villa 34 -24

19. Watford 32 -33

20. Norwich 27 -37

Finally, we use FileWriter function to export the result CSV.



Following Two files will be generated after the program finished. One is the result of all the matches we simulated and another one is the final rank after simulation.

手机屏幕的截图

描述已自动生成