Ranking System

—— English Premier League

Group Member:

001349439 Yifan Chen, 001087410 Jingyu Wang, 001300548 Yiyu Chen

1. Problem Statement

Develop a ranking system which is able to evaluate the following expression where xi, xj are elements from a set of competing elements X: P(xi, xj). where P(xi, xj) is the probability that xi would beat xj if they met in a head to head matchup at neutral territory.

A by-product of this expression would be the ability to build a table of elements ordered such that if an element xi appears above another element xj, then you may infer that P(xi, xj) > 0. Note that there may be a set of undecidable cycles (like rock, paper, scissors) in which case you will have to mark those elements as equal in your table.

It is desirable also that the value of P is not just a single number, but a probability density function (pdf), in other words there should perhaps be some sort of bounds on the P, maybe a uniform pdf between two values. As you get more data, the bounds will become narrower (i.e. your precision will improve).

The input to your system will be a set of prior encounters with a result. These results can be win-loss or they can be scores, as in for example the English Premier League (EPL). If appropriate, you may also consider home team advantage if you feel that it matters.

2. Problem Analysis

Many aspects of football can influence the outcome of the forecast, and some of them can give the bottom-scoring teams in the standings a great opportunity to beat the top-scoring teams in the standings.

a. The recent situation of the team

The winning team tends to be in high spirits and has an advantage in the next game. Usually, when a team is in the doldrums, problems arise in the locker room and the problems persist.

b. Recent records between two teams in historical battles

Teams like Arsenal have a poor record against teams like Liverpool, Chelsea and Manchester United. Analyze previous games between these teams to see if there is a pattern.

c. Home and away form

Knowing each team's home and away form is the defining factor of football, with some teams not winning all season and some having 12 unbeaten games at home. Statistically, the average team is much better at home than away. Home-field advantage is an important factor in football matches.

d. Time difference, distance, plateau and other factors.

In the World Cup and other international competitions, these factors often need special consideration, but in English Premier League, we do not need to consider these factors.

3. Mathematical Model Building

We combine mechanism analysis methods and test ingesting methods for the Premier League to establish a mathematical model, first of all, according to the understanding of the reality of football matches to analyze its causality, to find out the law reflecting the internal mechanism, need to have practical significance, according to the factors affecting the football match, we can give each team a ranking value, in the teams category of attributes for rank, the initial value of 100, according to the results of 288 games before the 2020-2021 season, Calculate the rank value of the 20 teams at the end of the current tournament. The rules are as follows:

A:B(A is home team)	**Win**	**Tie**	**Lose**
A.points>B.points	(+5,-7)	(-3,+2)	(-7,+4)
A.points=B.points	(+5,-5)	(-1,+1)	(-7,+7)

A.points<B.points	(+7,-5)	(+0,-0)	(-4,+7)

Remarks: points are current points for the team(win+3,tie+1,lose do nothing).

We see the two-team game as a "black box" system, because the strength of the two teams can not be directly judged, so the internal mechanism can not be directly sought. Therefore, we can data the data we consider before, such as the source of data, such as rank, points, the number of average goal each teams get accordingly etc. , and use statistical analysis methods on this basis, according to the predetermined guidelines in a certain type of model to choose the best fit with the data model.

Then we also calculate the possibility of wins and losses between Team xi and xj, which is P(xi,xj) when xi is a home team and xj is an away team. Considering the factor that the average team is much better at home than away, P(xi,xj) is different from (xj,xi). Also we give different weights depending on the year of the game. Because as time goes on, the team's performance changes back, so the more recent the game, the more representative the team's ability. This is described in detail in the following methods. Our group calls it a lottery pool.

4. Algorithm design and selection

According to what we build for the analysis. We choose to use the BP (back propagation) algorithm which is a kind of supervised learning algorithm. Because when we give some input learning sample, use reverse propagation algorithm to adjust the power and deviation of the network repeatedly, so that the output array and the expected array as close as possible, when the error square of the network output layer is less than the specified error when the training is completed, save the network's weight and deviation.

First, use the data we collect and calculate to do the initialization, randomly given each connection right w, v and the threshold θi, rt.

Then the output of the cells of the hidden layer and output layer is calculated by the given input and output mode bj=f(wijai-θj) ct=f(vjtbj-rt).

bj is the actual output of the first neuron of the hidden layer, ct is the actual output of the t neuron in the output layer, wij is the connection right from the input layer to the hidden layer, and vjt is the connection right from the hidden layer to the output layer.

Lastly the BP algorithm selects the next input mode to return to the previous step and repeat training until the network set output error to reach the required end of training.In this case, the traditional BP algorithm helps us to turn a set of sample input/output problems into a nonlinear optimization problem.

We use java's framework Encog to realize machine learning . In Encog, the creation of neural networks mainly uses the basic Network and BasicLayer classes, and of course other classes such as activation functions.

First, Import related packages using maven.

```
<dependencies>
 <dependency>
     <groupId>org.encog</groupId>
     <artifactId>encog-core</artifactId>
     <version>3.4</version>
 </dependency>
</dependencies>
```

Constructing a neural network: It's especially easy to build a neural network with Encog, first by first you need a New BasicNetwork object and then to add layers to the object. The first parameter of the BasicLayer constructor specifies the activation function, the second parameter specifies the number of neurons by bias neurons, the third parameter specifies the number of neurons, if only one number is passed, means that only the number of neurons is specified, and the default activation function is Sigmoid, with biased neurons.

```
BasicNetwork  nw=new BasicNetwork();
nw.addLayer(new BasicLayer(null,true,6));
nw.addLayer(new BasicLayer(new ActivationSigmoid(),true,2));
nw.addLayer(new BasicLayer(new ActivationSigmoid(),false,2));
nw.getStructure().finalizeStructure();
nw.reset();
```

Because paranoid neurons affect the next layer, and the activation function affects the data of the previous layer. So generally the first layer (input layer) does not need to activate the function, and the last layer (output layer) does not need to bias neurons. The call finalizeStructure indicates that the build has been completed and that there is no need to add layers to it, and reset is the random initialization of the connection weights between the layers.

Processing data:

Ready to enter data and the data we expect to output

Construct an MLDataSet object training set, and we pass our training data to it.

```
MLDataSet ts=new BasicMLDataSet(input,ideal);
```

Input layer:

```
public static double [][]input= new double[6748][6];
```

Output layer:

```
public static double[][] output =new double[6748][2];
```

We assume that [0,6] is a normal point in a match. So we divided the goals by 6 so that the output will be normalized.

```
public static double[][] output1 =new double[6748][2];
```

Hidden layers: In the previous discussion, when neural networks contained the input layer and output, sometimes the output layer and the input layer were the same, but most were usually two separate layers, and there might be other layers between the input layer and the output layer, which is called the hidden layer, and the hidden layer is between the input layer and the output layer. Hidden layers can also take on more complex structures.

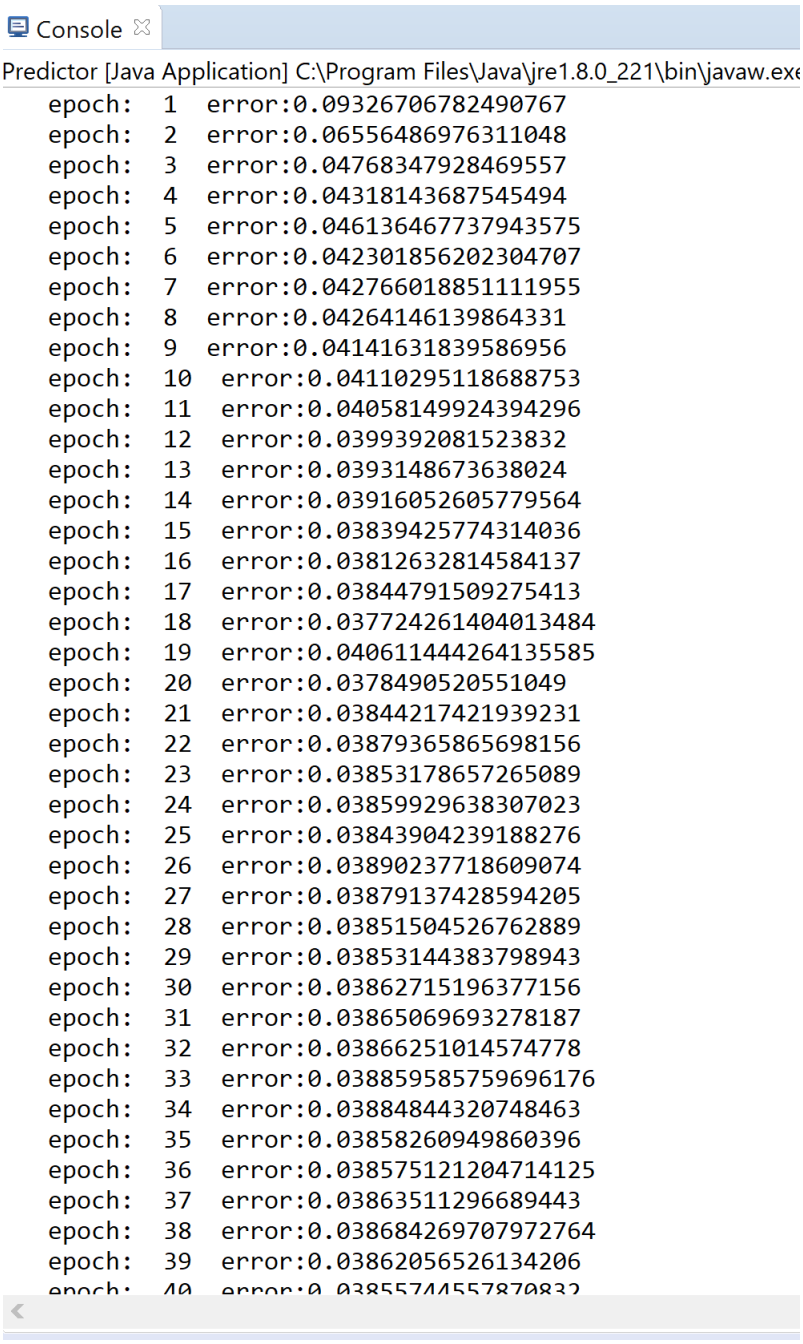
Using Elastic Propagation (RPROP) training: We train our network in this program using Elastic Communication Training (RPROP), which is the best training algorithm supported by Encog, which of course offers other training techniques and can solve certain problems with certain training techniques. The code for rPROP training is as follows:

```
MLTrain t=new ResilientPropagation (nw,ts);

int epoch=1;
do {
    t.iteration();

    epoch++;
}while(t.getError()>0.2);
```

The above code passes through multiple iterations, and when iterate to make the neural network error rate below 1, the neural network training is over. Here we can run our code to see the results of the training:



The error rate is high during the first iteration, and then decreases as the number of iterations increases. Each run turns out to be different, but as long as the error rate is decreasing, our neural network is built successfully.

Above is one method we use. (Totally using BP algorithm to predict)

The other method is combining BP algorithm and Lottery pool algorithm.

a. data structure

Creating an array of Team types and store all team information in the array. Team [] teams;

Creating an array of Match types, reading all match information for 20 years than stored in this array. Match [] matches;

(BP) Creating a Two-dimensional array to store the information as an input for training as a neural network error reverse-propagation algorithm. Int [][] input;

Creating a Two-dimensional array as an output to store the result of a neural network error reverse-propagation algorithm. Int [][] output;

(Calculate P(xi,xj)) Creating a hashmap store for the number of the home and away field as the key value, the value is a hashmap that stores all the scores of this team, and key value of inner hashmap is the score, and the value is the number of times the score appears. (Factors like home and away , game time determine the weight of the number of occurrences)

HashMap <String, HashMap<String,Integer>() map;

5. Algorithmic indicators

a. Correctness: We will represent the correctness of the two algorithms in the test (title 7)

b. Efficiency and storage demand:

This is a simple three-tier BP neural network, the number of neurons per layer is n1, n2, n3. Two matrix operations are performed, and the two matrix multiplications (actually vectors and matrix multiplications) are performed n1 n2 and n2 n3 etc alculations, respectively. Since the number of junctions (n1 and n3) of the input layer and the final output layer is determined, it can be considered a constant, and the hidden layer n2 in the middle can be set by itself.

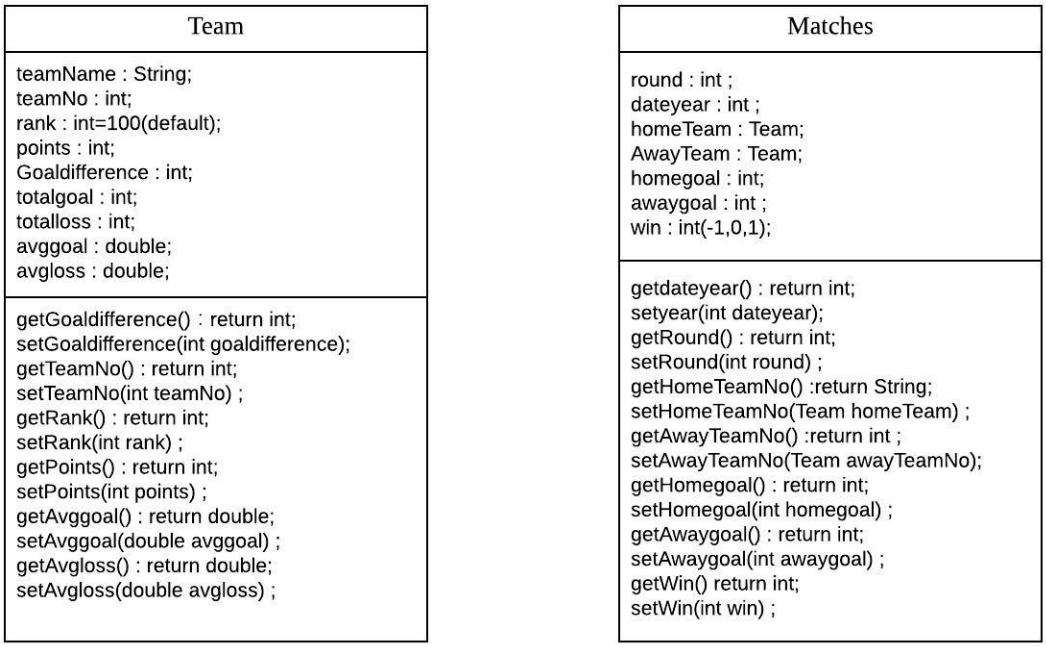
Therefore, the time complexity of calculating a feed-forward for a sample should be O (n1*n2+n2*n3). The time complexity and feed-forward calculation slot in reverse propagation are the same, what here are a total of m training samples,

each of which is trained only once, then the time complexity of training a neural network should be O (92*n^2).(92 is a constant number which can be ignored).

The time of using HashMap to find record is O(1). And predicting matches needs to iterate the outer hashmap and iterate the inner hashmap. Therefore , the time of predicting is O(½ n^2).

6. Algorithm implementation

Classes:



Team:

Field Name	Description
teamName	The current team's name
teamNo	The current team's No
rank	The number of current team's ability
points	The points of current team

Goldifference	The number of current team's goal difference
totalgoal	The number of total goals of current team
totalloss	The number of total conceded goals of current team
avggoal	The number of average goals of current team
avgloss	The number of average conceded goals of current team
played;	The number of matches have played by current team

Match:

Field Name	Description
round	The round of the match
homeTeamNo	The number of the home team
AwayTeamNo	The number of the away team
homegoal	The number of the home team's goal
awaygoal	The number of the away team's goal
win	The result of the match. 1 means home team win, 0 means draw, -1 means away team win

Method1:

Field Name	Description
int numTeams	Set to 20, final. The number of total teams from 2001-2020. The first 20 is the team of 2019-2020
Team[] teams	The array that has to be filled up with all of the team names and stats.
Team[] trainteams	The array that has to be filled up with all of the team names and stats for BP training.
int numMatchs	Set to 380, final. The number of total matches of 2019-2020
Matches[] matches	The array that has to be filled up with all of the matches and stats of 2019-2020.
double [][]input	The array that has to be filled up with converted input information for BP training.

double [][]output	The array that has to be filled up with converted output information for BP training.
Int success	The number of correct prediction.
double[][] outputcompressed	Normalize the output.
int[][] points	The array that has to be filled up the ranking of 2018-2019 after the prediction
HashMap<String,HashMap<String,Integer>>	The outer hashmap key is String which contains hometeam number and away team number and the outer hashmap value is a hashmap. And the inner hashmap key is String contains score and value is Number of occurrences multiply weight.

Method Name	Description
public static void readteams()	The method to read the team information form the csv files and put it into the teams array.
public static void readmatch()	The method to read the team information form the csv files and put it into the matches hashmap.
public static HashMap<String,HashMap<String,Integer>> calculatewithweight (Matches1[] matches)	The method to read the remaining matches from the csv files and put the round, home team and away team. Record the record between two teams. If the date of record before 2015, the weight is 2. If the date of record between 2016 and 2010 , the weight is 4. The other weight is 6. Also score should be recorded reversely in reversed team number at the same time the weight should be divided by 2 respectively.
HashMap<String,HashMap<String,Integer>> calculatewithbp(Matches1[] matches)	Combine the method above and BP algorithm when there are no record between two teams

public static ArrayList<Integer> predict1(int a,int b,double[][] input,double[][] ideal)	The method to predict matches through BP. The main factor is the abilities(rank), points and goals difference of the home team and the away team.
public static void renew(Matches2 m)	The method to renew the team information after reading a match's result.
public static void BP(double [][] data , double [][] target	The method to train a network to predict the matches.
static void result(HashMap<String,HashMap<String,Integer>> res)	Read the array which stores the remaining matches. Take out the number of the home team and the away team. Get a hashmap which contains all records of matches between the two teams before. And give a random number to decide what score they would have this time.

Method2:

Field Name	Description
int numTeams	Set to 40, final. The number of total teams from 2001-2020. The first 20 is the team of 2019-2020
Team2[] teams	The array that has to be filled up with all of the team names and stats.
Team2[] trainteams	The array that has to be filled up with all of the team names and stats for BP training.
int numMatchs	Set to 380, final. The number of total matches of 2019-2020
Matches2[] matches	The array that has to be filled up with all of the matches and stats of 2019-2020.
Matches2[] trainmatches	The array that has to be filled up with all of the matches and stats for BP training.
double [][]input	The array that has to be filled up with converted input information for BP training.
double [][]output	The array that has to be filled up with converted output information for BP training..
Int success	The number of correct prediction.

int[] stats	The 220 stats, 11 per team, are inputted into this array, as they read into a file.
int[][] points	The array that has to be filled up the ranking of 2018-2019 after the prediction

Method Name	Description
public static void read()	The method to read the team information form the csv files and put it into the teams array.
public static void trainread()	The method to read the team information form the csv files and put it into the trainteams array.
public static void predictmatch()	The method to read the remaining matches from the csv files and put the round, home team and away team into the matches array. And use the BP prediction or elo prediction to predict a result of the match then put it into the array. At last, renew the ranking of the teams.
public static void print() throws IOException	The method to get the total inputs for BP training and get a CSV file
public static ArrayList<Integer> predict1(int a,int b,double[][] input,double[][] ideal)	The method to predict matches through BP. The main factor is the abilities, average goals and average conceded goals of the home team and the away team.
public static int predict(int a, int b)	The method to predict the matches through elo prediction. The main factor is the team's current ranking.
public static void readmatch())	The method to read the match information form the csv files and put it into the matches array.
public static void trainmatch()	The method to read the match information form the csv files and put it into the trainmatches array.

public static void renew(Matches2 m)	The method to renew the ranking after reading a match's result.
public static void renew1(Matches2 m)	The method to renew the team information after reading a match's result.
public static void BP(double [][] data , double [][] target	The method to train a network to predict the matches.

7. Algorithm improvements

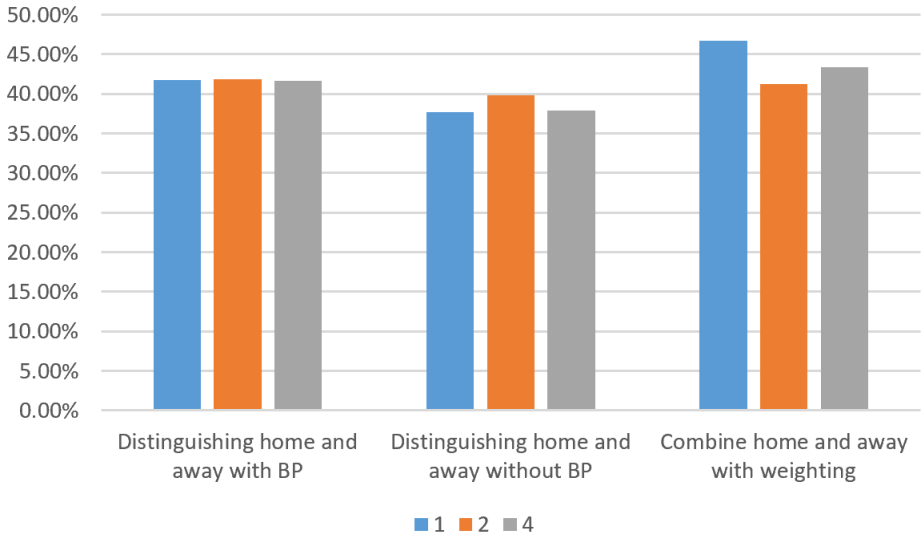
Override the sorting for teams according to their points. When points are the same , using the Goal difference.

8.Testing

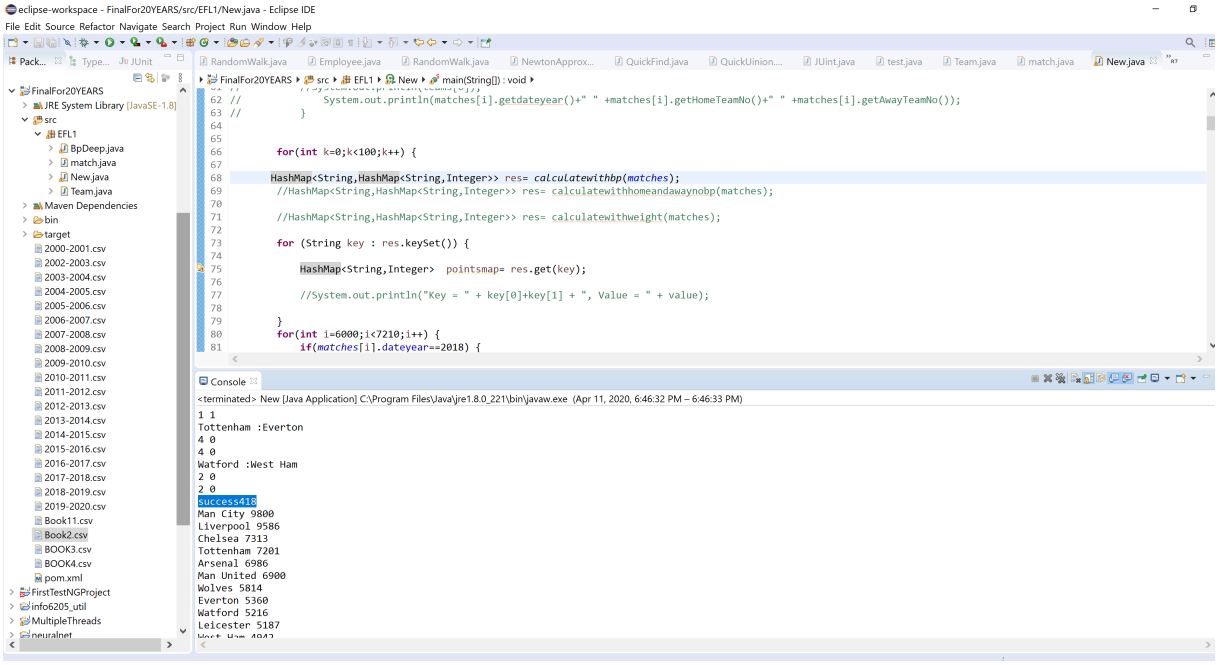
We predict the last 10, 20 and 40 matches in 2018-2019 premier league and calculate the accuracy.

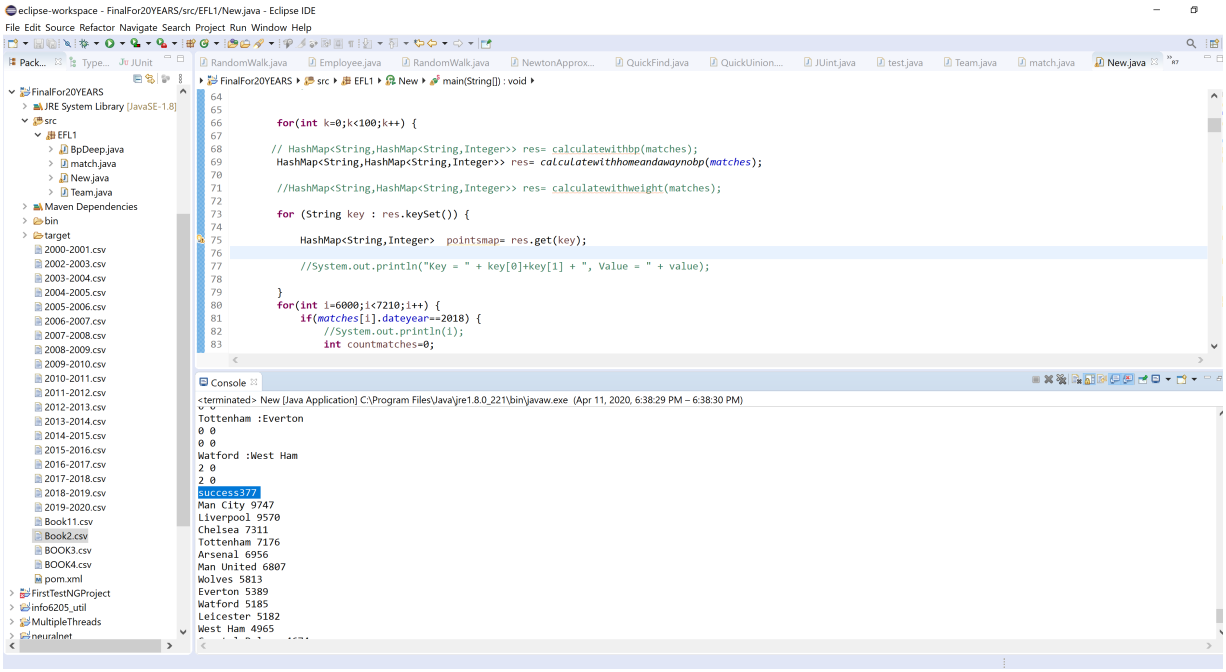
Method_A (combing lottery pool and BP algorithm) :

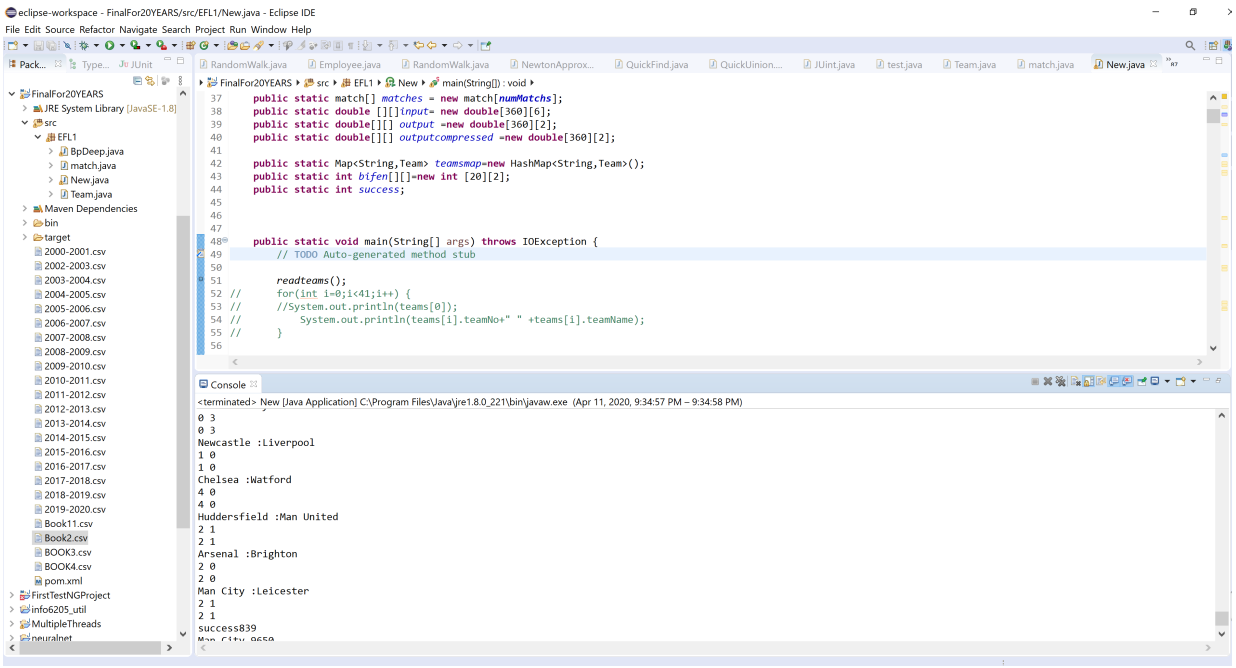
Predicted rounds	1	2	4
Distinguishing home and away with BP	41.80%	41.90%	41.70%
Distinguishing home and away without BP	37.70%	39.80%	37.90%
Combine home and away with weighting	46.70%	41.20%	43.40%

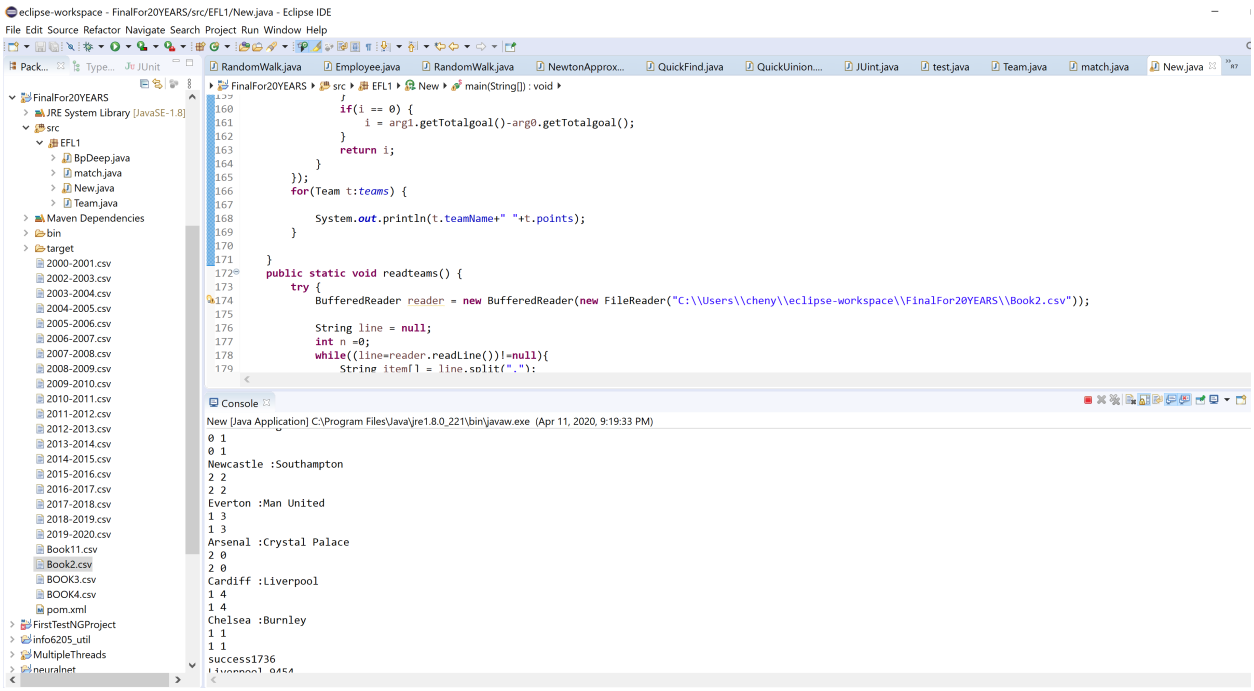


Some short cuts:






eclipse-workspace - FinalFor20YEARS/src/EFL1/New.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
FinalFor20YEARS
JRE System Library [JavaSE-1.8]
src
EFL1
BpDeep.java
match.java
New.java
Team.java
Maven Dependencies
bin
target
2000-2001.csv
2002-2003.csv
2003-2004.csv
2004-2005.csv
2005-2006.csv
2006-2007.csv
2007-2008.csv
2008-2009.csv
2009-2010.csv
2010-2011.csv
2011-2012.csv
2012-2013.csv
2013-2014.csv
2014-2015.csv
2015-2016.csv
2016-2017.csv
2017-2018.csv
2018-2019.csv
2019-2020.csv
Book11.csv
Book2.csv
BOOK3.csv
BOOK4.csv
pom.xml
FirstTestNGProject
info6205_util
MultipleThreads
neuralnet
RandomWalk.java
Employee.java
RandomWalk.java
NewtonApprox...
QuickFind.java
QuickUnion....
JUint.java
test.java
Team.java
match.java
New.java
FinalFor20YEARS ▸ src ▸ EFL1 ▸ New ▸ main(String[]) : void ▸
37 public static match[] matches = new match[numMatchs];
38 public static double [][]input= new double[360][6];
39 public static double[][] output =new double[360][2];
40 public static double[][] outputcompressed =new double[360][2];
41
42 public static Map<String,Team> teamsmap=new HashMap<String,Team>();
43 public static int bifen[][]=new int [20][2];
44 public static int success;
45
46
47
48 public static void main(String[] args) throws IOException {
49 // TODO Auto-generated method stub
50
51 readteams();
52 // for(int i=0;i<41;i++) {
53 // //System.out.println(teams[0]);
54 // System.out.println(teams[i].teamNo+" " +teams[i].teamName);
55 // }
56
Console
<terminated> New [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (Apr 11, 2020, 9:34:57 PM – 9:34:58 PM)
0 3
0 3
Newcastle :Liverpool
1 0
1 0
Chelsea :Watford
4 0
4 0
Huddersfield :Man United
2 1
2 1
Arsenal :Brighton
2 0
2 0
Man City :Leicester
2 1
2 1
success839


eclipse-workspace - FinalFor20YEARS/src/EFL1/New.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
FinalFor20YEARS
JRE System Library [JavaSE-1.8]
src
EFL1
BpDeep.java
match.java
New.java
Team.java
Maven Dependencies
bin
target
2000-2001.csv
2002-2003.csv
2003-2004.csv
2004-2005.csv
2005-2006.csv
2006-2007.csv
2007-2008.csv
2008-2009.csv
2009-2010.csv
2010-2011.csv
2011-2012.csv
2012-2013.csv
2013-2014.csv
2014-2015.csv
2015-2016.csv
2016-2017.csv
2017-2018.csv
2018-2019.csv
2019-2020.csv
Book11.csv
Book2.csv
BOOK3.csv
BOOK4.csv
pom.xml
FirstTestNGProject
info6205_util
MultipleThreads
neuralnet
RandomWalk.java
Employee.java
RandomWalk.java
NewtonApprox...
QuickFind.java
QuickUnion....
JUint.java
test.java
Team.java
match.java
New.java
FinalFor20YEARS ▸ src ▸ EFL1 ▸ New ▸ main(String[]) : void ▸
160 if(i == 0) {
161 i = arg1.getTotalgoal()-arg0.getTotalgoal();
162 }
163 return i;
164 }
165 });
166 for(Team t:teams) {
167
168 System.out.println(t.teamName+" "+t.points);
169 }
170
171 }
172 public static void readteams() {
173 try {
174 BufferedReader reader = new BufferedReader(new FileReader("C:\\Users\\cheny\\eclipse-workspace\\FinalFor20YEARS\\Book2.csv"));
175
176 String line = null;
177 int n =0;
178 while((line=reader.readLine())!=null){
179 String item[] = line.split(",");
Console
New [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (Apr 11, 2020, 9:19:33 PM)
0 1
0 1
Newcastle :Southampton
2 2
2 2
Everton :Man United
1 3
1 3
Arsenal :Crystal Palace
2 0
2 0
Cardiff :Liverpool
1 4
1 4
Chelsea :Burnley
1 1
1 1
success1736

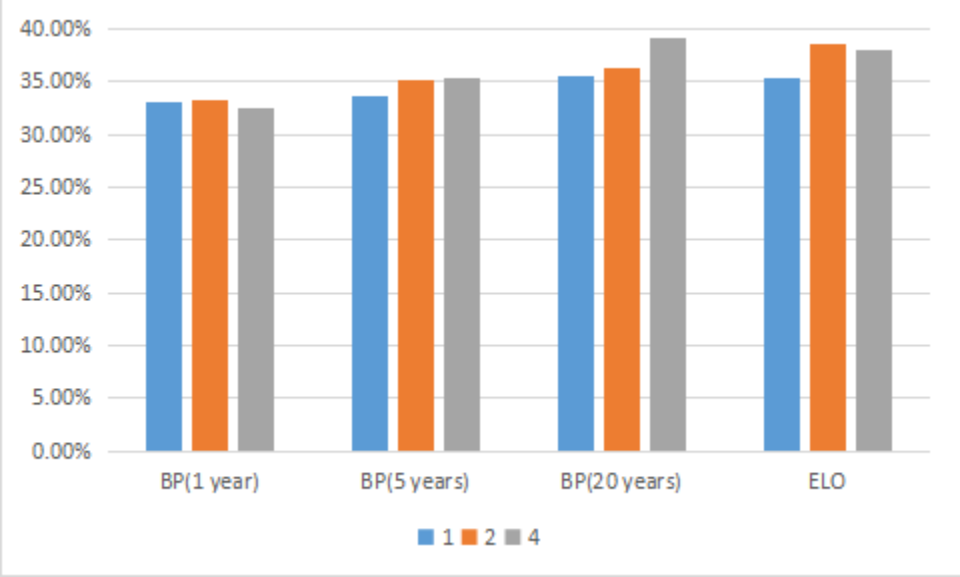
Conclusion:

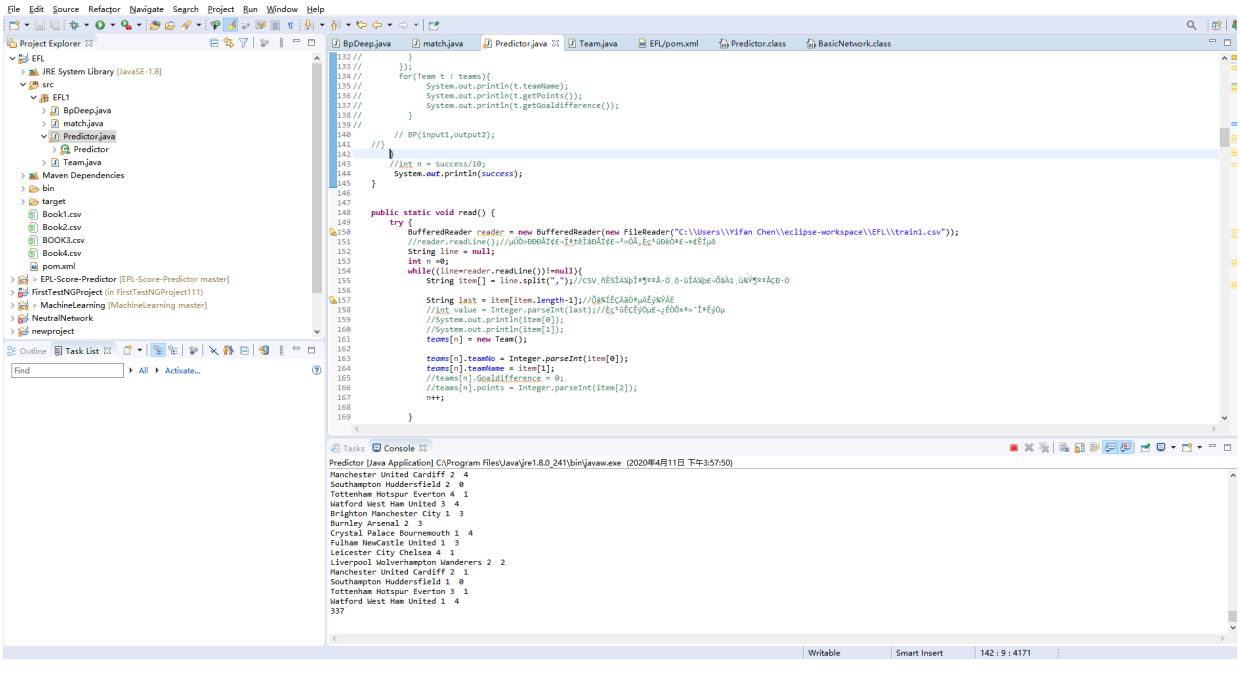
1. **The more games you record, the fewer games you predict, the more accurate.**
2. **Combination makes the algorithms more stable**

Method_B (Using BP algorithm):

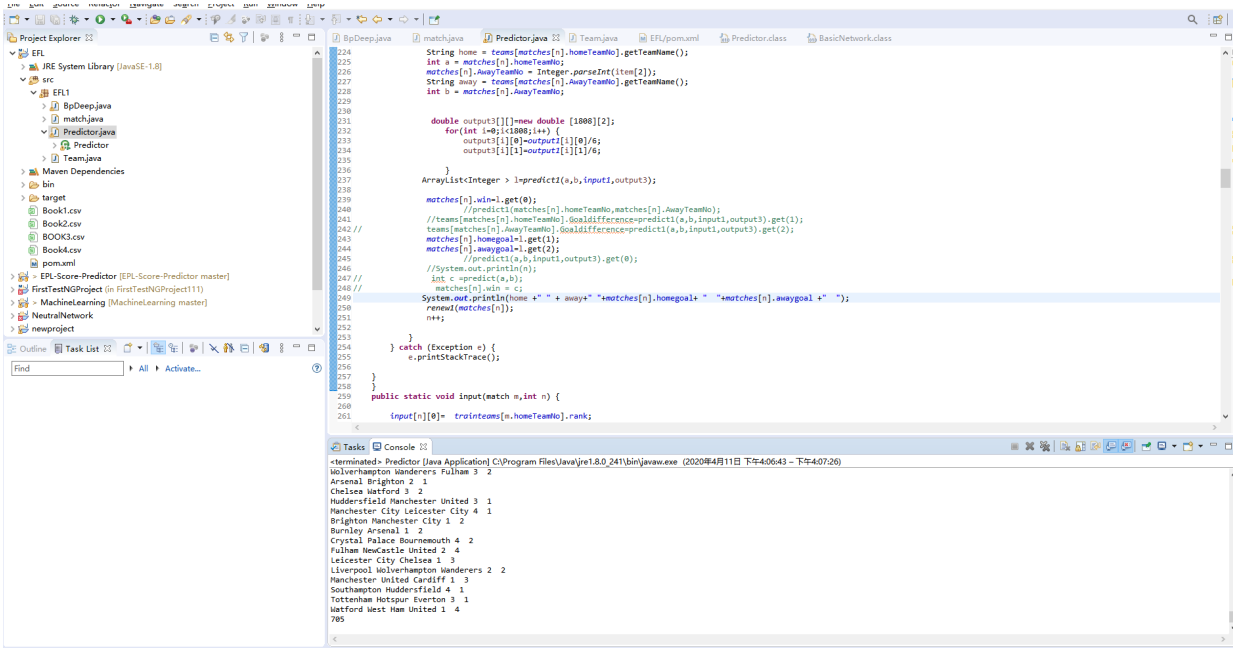
Test Result of BP (which learning 1 year ,5 years and 20 years accordingly) and ELO

Round(count backwards)	1	2	4
BP(1 year)	33.10%	33.20%	32.55%
BP(5 years)	33.70%	35.25%	35.30%
BP(20 years)	35.50%	36.37%	39.10%
ELO	35.30%	38.50%	38.05%

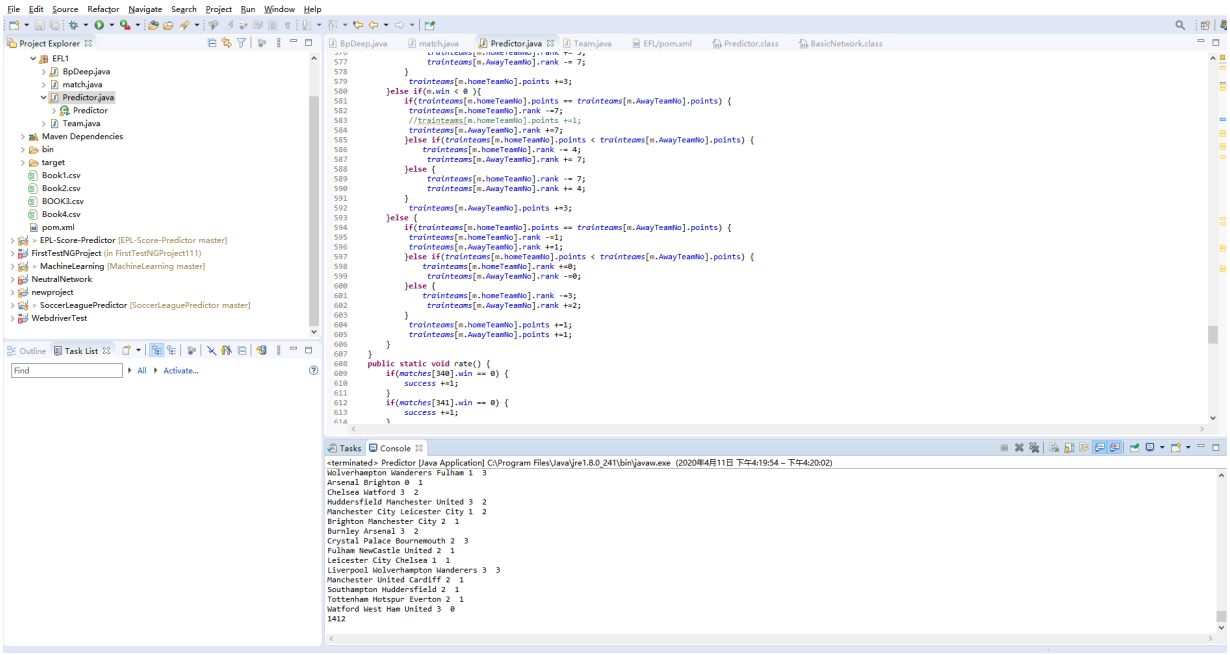




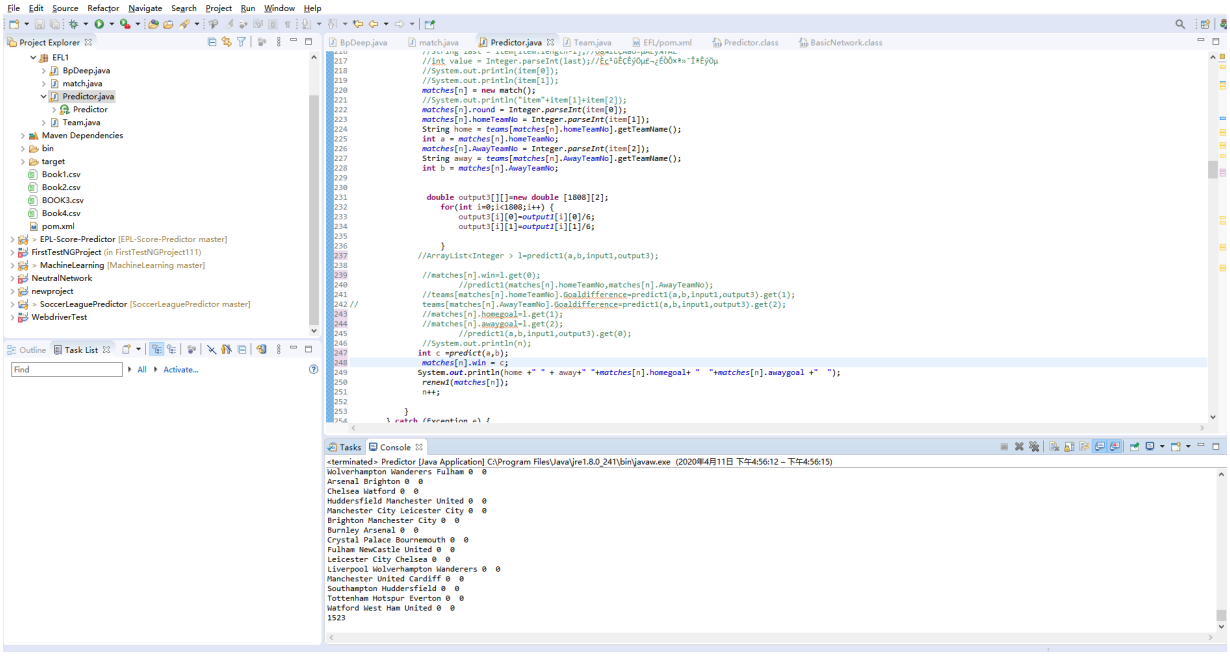
```
Predictor [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe  (2020年4月11日 下午3:57:50)
Manchester United Cardiff 2  4
Southampton Huddersfield 2  0
Tottenham Hotspur Everton 4  1
Watford West Ham United 3  4
Brighton Manchester City 1  3
Burnley Arsenal 2  3
Crystal Palace Bournemouth 1  4
Fulham NewCastle United 1  3
Leicester City Chelsea 4  1
Liverpool Wolverhampton Wanderers 2  2
Manchester United Cardiff 2  1
Southampton Huddersfield 1  0
Tottenham Hotspur Everton 3  1
Watford West Ham United 1  4
337
```



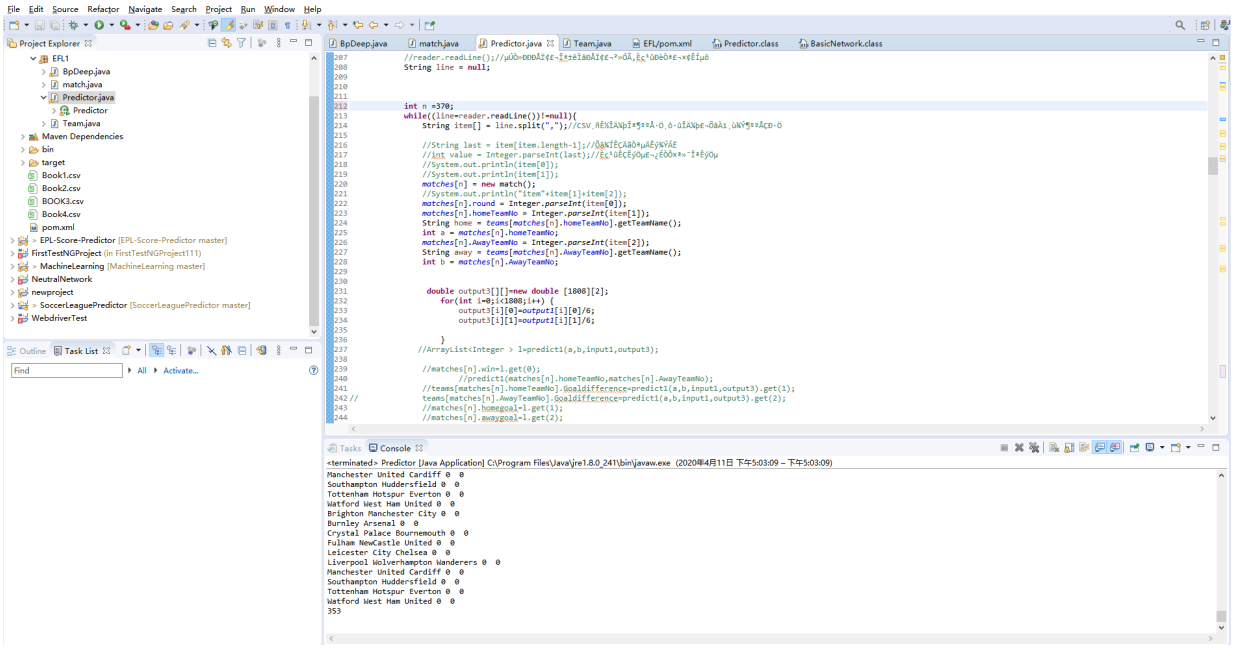
```
<terminated> Predictor [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe  (2020年4月11日 下午4:06:43 – 下午4:07:26)
Wolverhampton Wanderers Fulham 3  2
Arsenal Brighton 2  1
Chelsea Watford 3  2
Huddersfield Manchester United 3  1
Manchester City Leicester City 4  1
Brighton Manchester City 1  2
Burnley Arsenal 1  2
Crystal Palace Bournemouth 4  2
Fulham NewCastle United 2  4
Leicester City Chelsea 1  3
Liverpool Wolverhampton Wanderers 2  2
Manchester United Cardiff 1  3
Southampton Huddersfield 4  1
Tottenham Hotspur Everton 3  1
Watford West Ham United 1  4
705
```



```
Wolverhampton Wanderers Fulham 1  3
Arsenal Brighton 0  1
Chelsea Watford 3  2
Huddersfield Manchester United 3  2
Manchester City Leicester City 1  2
Brighton Manchester City 2  1
Burnley Arsenal 3  2
Crystal Palace Bournemouth 2  3
Fulham NewCastle United 2  1
Leicester City Chelsea 1  1
Liverpool Wolverhampton Wanderers 3  3
Manchester United Cardiff 2  1
Southampton Huddersfield 2  1
Tottenham Hotspur Everton 2  1
Watford West Ham United 3  0
1412
```



```
Wolverhampton Wanderers Fulham 0  0
Arsenal Brighton 0  0
Chelsea Watford 0  0
Huddersfield Manchester United 0  0
Manchester City Leicester City 0  0
Brighton Manchester City 0  0
Burnley Arsenal 0  0
Crystal Palace Bournemouth 0  0
Fulham NewCastle United 0  0
Leicester City Chelsea 0  0
Liverpool Wolverhampton Wanderers 0  0
Manchester United Cardiff 0  0
Southampton Huddersfield 0  0
Tottenham Hotspur Everton 0  0
Watford West Ham United 0  0
1523
```

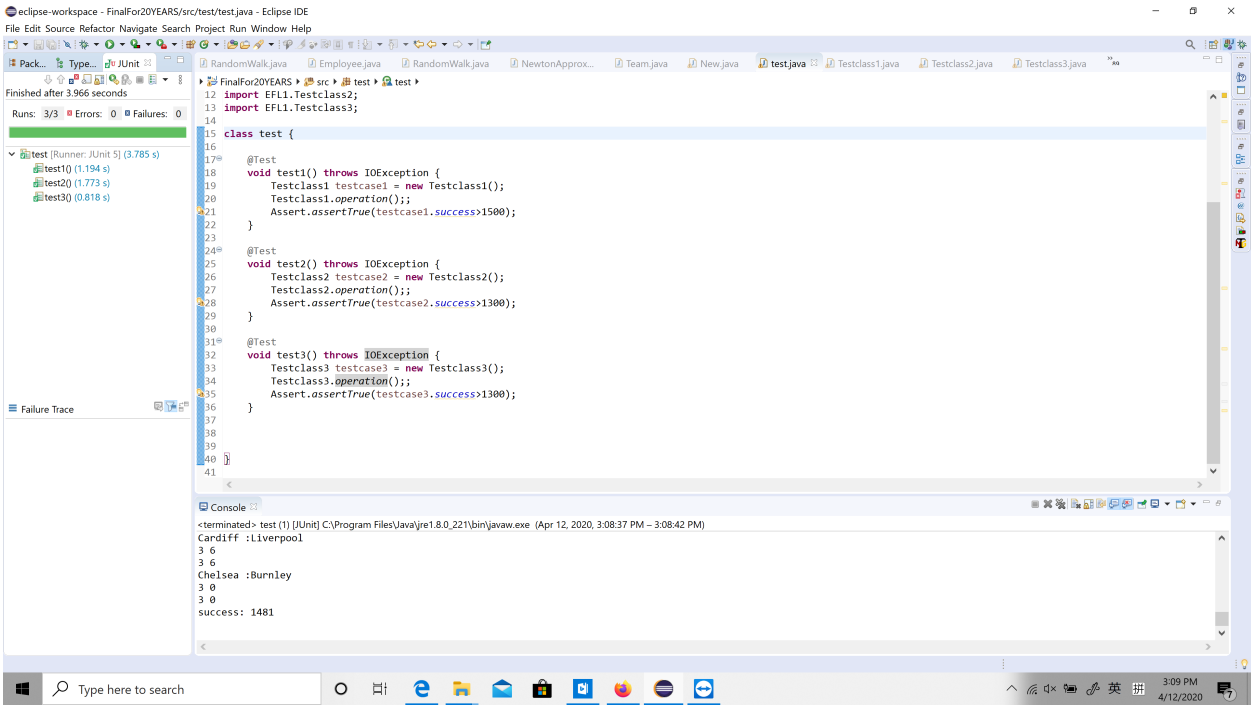


Conclusion:

1. **The larger the BP training sample size, the higher the prediction accuracy.**
2. **The more prediction fields, the higher the prediction accuracy. The final accuracy rate is around 40%. (It may be because the more games played, the lower the chance of upset.)**
3. **The prediction of Premier League 2019-2020 is as the picture below, Liverpool will be the champion, Man City will be the second, and Liverpool, Leicester City, Chelsea and Man United will attend the UEFA Champions League.**

Bournemouth, Aston Villa and Norwich will downgrade

Test result :

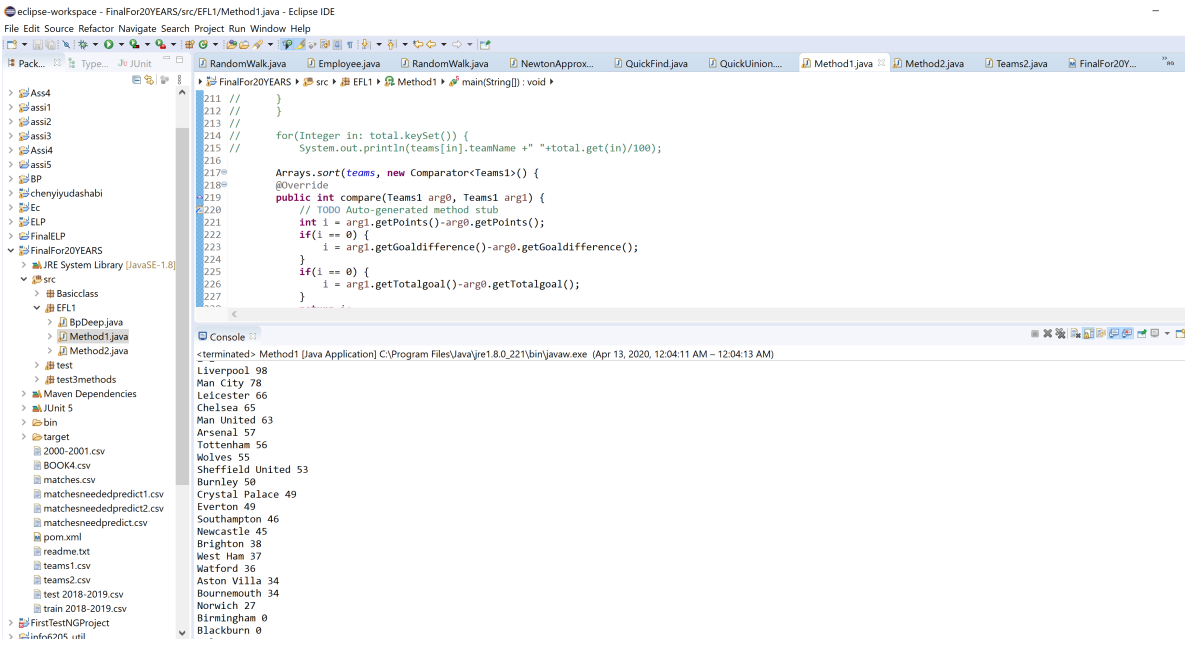


9.Results

Method1:

```
Aston Villa :Sheffield United
3 0
Man City :Arsenal
1 3
Bournemouth :Crystal Palace
2 2
Aston Villa :Chelsea
0 4
Brighton :Arsenal
2 1
Man City :Burnley
1 1
Newcastle :Sheffield United
0 1
Norwich :Southampton
1 1
Watford :Leicester
2 1
Tottenham :Man United
1 2
West Ham :Wolves
0 1
Everton :Liverpool
1 3
Tottenham :West Ham
2 0
Burnley :Watford
1 2
Chelsea :Man City
1 1
Liverpool :Crystal Palace
1 3
Newcastle :Aston Villa
1 1
Norwich :Everton
1 1
```

Liverpool	98
Man City	78
Leicester	66
Chelsea	65
Man United	63
Arsenal	57
Tottenham	56
Wolves	55
Sheffield United	53
Burnley	50
Crystal Palace	49
Everton	49
Southampton	46
Newcastle	45
Brighton	38
West Ham	37
Watford	36
Aston Villa	34
Bournemouth	34
Norwich	27



Method2:

Liverpool	99
Manchester City	73
Leicester City	66
Chelsea	60
Wolverhampton Wanderers	59
Manchester United	58
Sheffield United	56
Arsenal	53
Tottenham Hotspur	53
Crystal Palace	49
Burnley	48
Everton	48
NewCastle United	46
Southampton	45
Brighton	41
West Ham United	38
Watford	37
Bournemouth	37
Aston Villa	36
Norwich City	30