Figure 1 - Manchester City Celebrates 2019 EPL Victory [1]

English Premier League Ranking

https://github.com/renbeynolds/info6205_final_project

INFO 6205

Northeastern University

Benjamin Reynolds

NUID 001409591

[1] https://www.nytimes.com/2019/05/12/sports/manchester-city-premier-league-title.html

Table of Contents

Introduction

The **Premier League** began in 1992 and is the most elite league in English football and the most watched sports league in the world with billions of television viewers annually. Each season pits 20 of the world's best football teams against one another two times (one game at each team's home arena) for a total of 38 matches. The league standings are determined by a point system with each match providing the team with an opportunity to increase their total by 3 in the case of a win or 1 in the event of a tie[2].

The nature of the EPL ranking system and scoring method for football matches lends well to simulation using **Poisson Distribution**. The Poisson Distribution probability mass function (pmf) gives the probability of *k* events occuring in a fixed amount of time when the average frequency or rate of occurrence is known. The probability *P* is given by the following equation where *k* is the number of events and λ is the average rate [3].

$$P(k \text{ events in interval}) = e^{-\lambda}\frac{\lambda^k}{k!}$$

This equation can be applied to find the probability of a particular football team scoring *k* goals in a given match and therefore can be used to predict match outcomes and overall league rankings. This approach will be used in this project to project the results of the 2019 - 2020 EPL season which was interrupted prior to its completion due to the COVID-19 pandemic.

Problem Statement

The primary objective of this project is to develop a ranking system capable of evaluating the expression *P(Xi, Xj)* where *Xi* and *Xj* are competing teams and *P(Xi, Xj)* is the probability that *Xi* would beat *Xj* in a head to head matchup. The probability should be given as a density function of resultant match scores instead of a single number. Through repeated application of this expression to simulate a complete EPL season a ranking of teams can be derived. A secondary objective of the project is to apply this strategy in simulating the completion of the 2019 - 2020 EPL season.

[2] https://www.bbc.co.uk/newsround/40891247
[3] https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459

Implementation

Statistics

This application uses the total number of goals scored in a match by each the home and away team as input parameters for building a statistical model. The first step in building the model is to read past match data from CSV files[4] to compute the following values both for each team individually as well as for the entire league as a whole:

a) Average goals scored in home matches
b) Average goals conceded in home matches
c) Average goals scored in away matches
d) Average goals conceded in away matches

Relative strength ratings for each team are then calculated by normalizing the team's values against those for the entire league:

e) Home attacking strength = a(self) / a(league)
f) Home defensive strength = b(self) / b(league)
g) Away attacking strength = c(self) / c(league)
h) Away defensive strength = d(self) / d(league)

These values are then used in calculating the λ rate parameter for use in the Poisson Distribution probability mass function:

i) λ home = e * h * a(league)
j) λ away = g * f * c(league)

These values indicate how many goals each team is expected on average to score in a full 90 minute football match, factoring in their own scoring ability based on whether they are home or away as well as the opposing team's defensive strength. To simulate the outcome of a single match, the Poisson PMF is applied for each team with *k* values ranging from 0 - 10 since the likelihood of any team scoring more than 10 goals in a single match is incredibly low. The probability of a resultant match score of *k1* - *k2* is then given by *P(k1 home) * P(k2 away)*. This yields 121 discrete match outcomes and their probability.

Calculating the probability of a home win for example then becomes a trivial matter of summing the probability of all match outcomes where *k1* > *k2*. Likewise the probability of an away victory

[4] http://www.football-data.co.uk/englandm.php

is the sum of probabilities where *k2 > k1* and the probability of a tie is the sum of probabilities where *k1 = k2*. The last piece in generating a ranking of teams is to compute the points expected by each team over the course of an entire season. Points are assigned to a team for each simulated match using the following since the premier league gives 3 points for a win and 1 for a tie:

- Home team points = 3 * *P(home win)* + *P(tie)*
- Away team points = 3 * *P(away win)* + *P(tie)*

Programming

This application is built as a Java Command Line Interface (CLI) using picocli [5]. The main application class references three subcommand classes:

- Rank: simulate entire season and ranks teams based on expected point values
- Compete: simulate single match and display probability of each possible outcome
- Table: simulate entire season and display most probable scoreline of each match

Each of these subcommand classes extends from a BaseCommand class which handles parsing CSV files using the DataLoader class and creating a Simulator instance. The Simulator performs most of the business logic for the application with assistance from the models and Poisson class for calculation. The three main data structures used in the application are HashSet (for unordered data not requiring lookup), HashMap (for unordered data which needs to be gettable based on a key), and TreeMap (for data which needs to be both ordered and gettable).

When the application starts, CSV files are parsed to create MatchData objects which are stored in a HashSet. These historic records have no need for sorting or keyd lookup since they are only used by a single loop in the Simulator to construct team and league models based on the goals scored in the match.

As the TeamModel objects are built by the simulator they are put into a TreeMap using the team name as the key. This keeps them sorted by name for easier generation of a sorted league table output while also allowing the models to be retrieved by team name during match simulation.

Lastly, two-dimensional HashMaps are used for storing match information which needs to be retrievable but not sorted. The first level key is the home team name and the second key is the away team name. This structure is used in two different places. The first is when simulating a partial season. When loading the CSV containing the matches of the incomplete season, the data for a given match is stored keyd by home and away team names allowing the Simulator to

[5] https://picocli.info/

later lookup if that match has been completed or not. The second is when the Simulator is running a full season simulation. Each MatchModel which it generates is similarly stored by home team name and away team name.

Results

Compete

The **compete** command performs a single simulated match between two teams based on models of those teams generated from past match data. The application accepts a range of past seasons to consider when generating the team models. Figure 2 below shows a surface plot of match outcome probabilities for a Manchester City (Home) vs Liverpool (Away) match using data from the 2014 - 2018 seasons. The data for this plot was produced using the following command:

```
java -jar build/libs/eplranking.jar \
    compete \
    -d ./datasets \
    -f 2014 \
    -l 2018 \
    -h "Man City" \
    -a "Liverpool" \
    -t
```

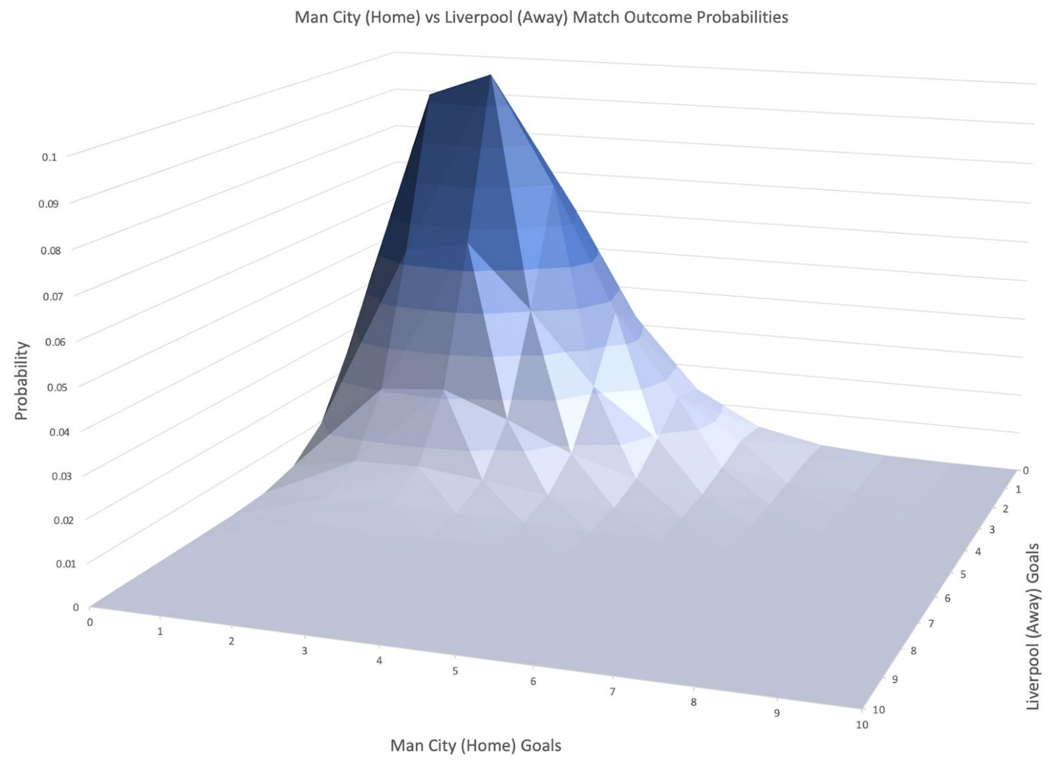


Figure 2 - Manchester City vs Liverpool Surface Plot

The summary option produces the following output for the match which aligns with the data depicted in Figure 2:

```
Home: Man City        Away: Liverpool

Probability of home win: 0.5971
Probability of tie     : 0.2040
Probability of away win: 0.1989

Most likely outcome:
Man City beats Liverpool with a score of 2 - 1 (probability 0.0980)
```

Interestingly, when the home and away roles are reversed the expected outcome changes, thus demonstrating the importance of home team advantage in football. Giving Liverpool the home team advantage in this instance allows them to hold Manchester City to a tie instead of conceding a 2 - 1 loss. This result also demonstrates the point that the most probable scoreline may not correspond with the most likely victor. In this particular case, the most likely final score is a 1 - 1 draw but the most likely result overall is still a Manchester victory:

```
Home: Liverpool       Away: Man City

Probability of home win: 0.3580
Probability of tie     : 0.2541
Probability of away win: 0.3879

Most likely outcome:
Liverpool ties Man City with a score of 1 - 1 (probability 0.1200)
```

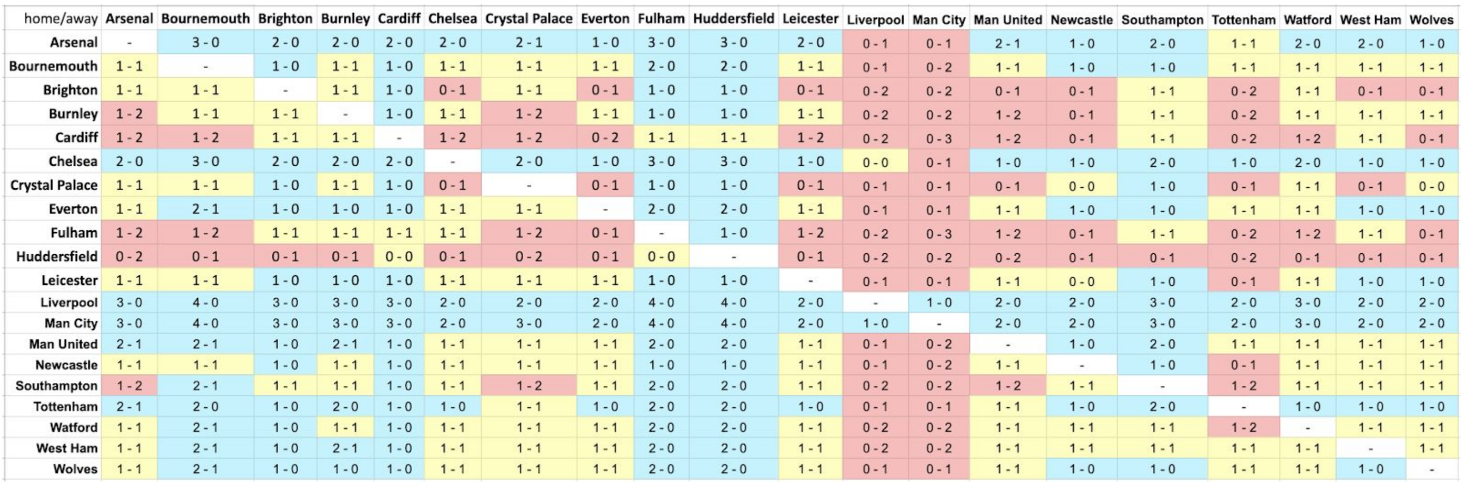
Table

The **table** command performs a double round robin tournament of all teams present in the input data sets and generates a league table of the most probable outcomes. In an actual EPL season, 20 teams each play one another twice for a total of 38 matches unlike this command which can consider more than 20 teams if present. For the simulation presented here, the input data was restricted to only the 2018 - 2019 season and the application was executed using this command:

```
java -jar build/libs/eplranking.jar \
    table \
    -d ./datasets \
    -f 2018 \
    -l 2018 > output.csv
```

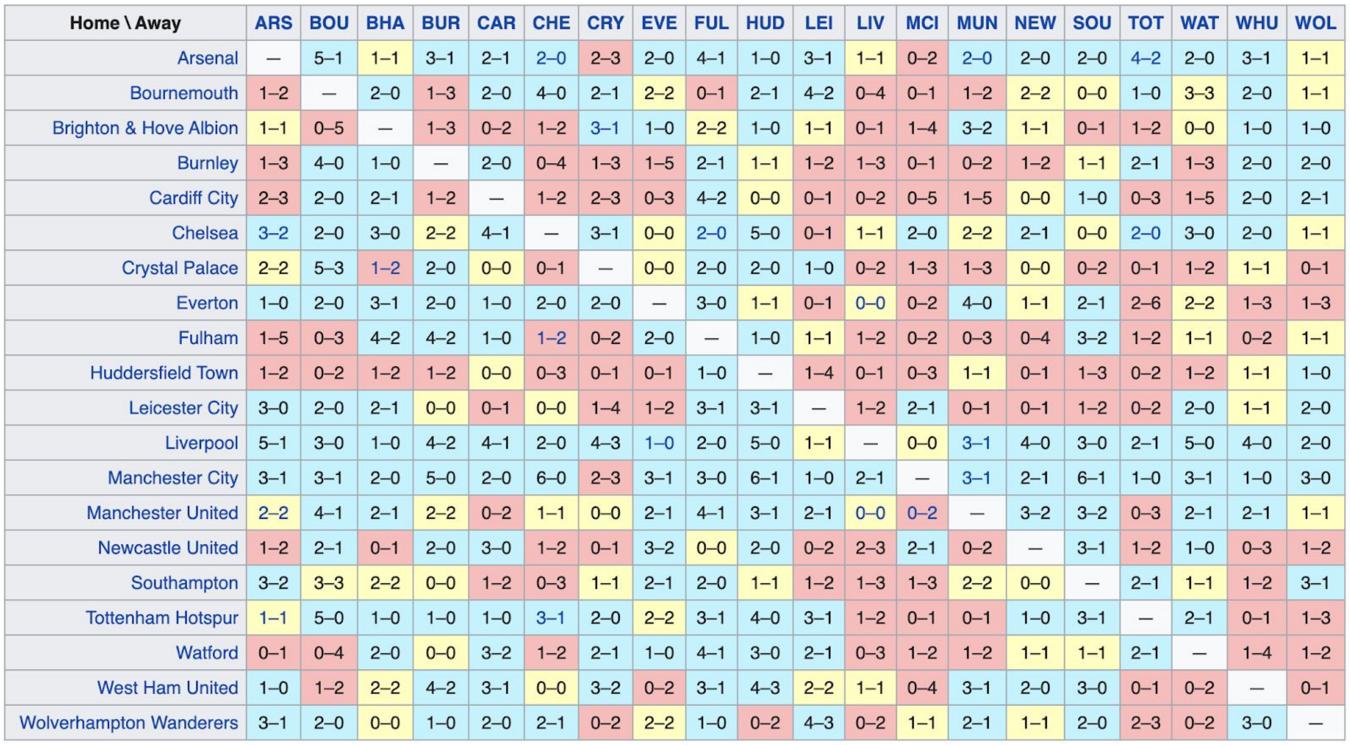
Figure 3 on the following page shows the output CSV data colorized to highlight home wins (blue), ties (yellow), and away wins (red). Figure 4 is the actual league table for the 2018 - 2019 EPL season. Comparing the two provides some interesting insights. One observation which can be made is that the simulation favors lower scoring games compared to real life. Probability dictates that lower scores are more likely when considering all matches overall in accordance with the law of large numbers [6]. Another observation is that the simulation results are much less one sided than reality. This is likely due in part due to factors outside the scope of this model such as a team losing heart after falling a couple goals behind. The model also inherently reduces the skill gap between teams since all teams are normalized against the league average thus making all teams appear more closely matched. Despite these inaccuracies, however, the simulation still manages a fair approximation of the actual season as shown on the following page:

[6] https://en.wikipedia.org/wiki/Law_of_large_numbers



home/away	Arsenal	Bournemouth	Brighton	Burnley	Cardiff	Chelsea	Crystal Palace	Everton	Fulham	Huddersfield	Leicester	Liverpool	Man City	Man United	Newcastle	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	-	3 - 0	2 - 0	2 - 0	2 - 0	2 - 0	2 - 1	1 - 0	3 - 0	3 - 0	2 - 0	0 - 1	0 - 1	2 - 1	1 - 0	2 - 0	1 - 1	2 - 0	2 - 0	1 - 0
Bournemouth	1 - 1	-	1 - 0	1 - 1	1 - 0	1 - 1	1 - 1	1 - 1	2 - 0	2 - 0	1 - 1	0 - 1	0 - 2	1 - 1	1 - 0	1 - 0	1 - 1	1 - 1	1 - 1	1 - 1
Brighton	1 - 1	1 - 1	-	1 - 1	1 - 0	0 - 1	1 - 1	0 - 1	1 - 0	1 - 0	0 - 1	0 - 2	0 - 2	0 - 1	0 - 1	1 - 1	0 - 2	1 - 1	0 - 1	0 - 1
Burnley	1 - 2	1 - 1	1 - 1	-	1 - 0	1 - 1	1 - 2	1 - 1	1 - 0	1 - 0	1 - 1	0 - 2	0 - 2	1 - 2	0 - 1	1 - 1	0 - 2	1 - 1	1 - 1	1 - 1
Cardiff	1 - 2	1 - 2	1 - 1	1 - 1	-	1 - 2	1 - 2	0 - 2	1 - 1	1 - 1	1 - 2	0 - 2	0 - 3	1 - 2	0 - 1	1 - 1	0 - 2	1 - 2	1 - 1	0 - 1
Chelsea	2 - 0	3 - 0	2 - 0	2 - 0	2 - 0	-	2 - 0	1 - 0	3 - 0	3 - 0	1 - 0	0 - 0	0 - 1	1 - 0	1 - 0	2 - 0	1 - 0	2 - 0	1 - 0	1 - 0
Crystal Palace	1 - 1	1 - 1	1 - 0	1 - 1	1 - 0	0 - 1	-	0 - 1	1 - 0	1 - 0	0 - 1	0 - 1	0 - 1	0 - 1	0 - 0	1 - 0	0 - 1	1 - 1	0 - 1	0 - 0
Everton	1 - 1	2 - 1	1 - 0	1 - 0	1 - 0	1 - 1	1 - 1	-	2 - 0	2 - 0	1 - 1	0 - 1	0 - 1	1 - 1	1 - 0	1 - 0	1 - 1	1 - 1	1 - 0	1 - 0
Fulham	1 - 2	1 - 2	1 - 1	1 - 1	1 - 1	1 - 1	1 - 2	0 - 1	-	1 - 0	1 - 2	0 - 2	0 - 3	1 - 2	0 - 1	1 - 1	0 - 2	1 - 2	1 - 1	0 - 1
Huddersfield	0 - 2	0 - 1	0 - 1	0 - 1	0 - 0	0 - 1	0 - 2	0 - 1	0 - 0	-	0 - 1	0 - 2	0 - 2	0 - 2	0 - 1	0 - 1	0 - 2	0 - 1	0 - 1	0 - 1
Leicester	1 - 1	1 - 1	1 - 0	1 - 0	1 - 0	1 - 1	1 - 1	1 - 1	1 - 0	1 - 0	-	0 - 1	0 - 1	1 - 1	0 - 0	1 - 0	0 - 1	1 - 1	1 - 0	1 - 0
Liverpool	3 - 0	4 - 0	3 - 0	3 - 0	3 - 0	2 - 0	2 - 0	2 - 0	4 - 0	4 - 0	2 - 0	-	1 - 0	2 - 0	2 - 0	3 - 0	2 - 0	3 - 0	2 - 0	2 - 0
Man City	3 - 0	4 - 0	3 - 0	3 - 0	3 - 0	2 - 0	3 - 0	2 - 0	4 - 0	4 - 0	2 - 0	1 - 0	-	2 - 0	2 - 0	3 - 0	2 - 0	3 - 0	2 - 0	2 - 0
Man United	2 - 1	2 - 1	1 - 0	2 - 1	1 - 0	1 - 1	1 - 1	1 - 1	2 - 0	2 - 0	1 - 1	0 - 1	0 - 2	-	1 - 0	2 - 0	1 - 1	1 - 1	1 - 1	1 - 1
Newcastle	1 - 1	1 - 1	1 - 0	1 - 1	1 - 0	1 - 1	1 - 1	1 - 1	1 - 0	1 - 0	1 - 1	0 - 1	0 - 2	1 - 1	-	1 - 0	0 - 1	1 - 1	1 - 1	1 - 1
Southampton	1 - 2	2 - 1	1 - 1	1 - 1	1 - 0	1 - 1	1 - 2	1 - 1	2 - 0	2 - 0	1 - 1	0 - 2	0 - 2	1 - 2	1 - 1	-	1 - 2	1 - 1	1 - 1	1 - 1
Tottenham	2 - 1	2 - 0	1 - 0	2 - 0	1 - 0	1 - 0	1 - 1	1 - 0	2 - 0	2 - 0	1 - 0	0 - 1	0 - 1	1 - 1	1 - 0	2 - 0	-	1 - 0	1 - 0	1 - 0
Watford	1 - 1	2 - 1	1 - 0	1 - 1	1 - 0	1 - 1	1 - 1	1 - 1	2 - 0	2 - 0	1 - 1	0 - 2	0 - 2	1 - 1	1 - 1	1 - 1	1 - 2	-	1 - 1	1 - 1
West Ham	1 - 1	2 - 1	1 - 0	2 - 1	1 - 0	1 - 1	1 - 1	1 - 1	2 - 0	2 - 0	1 - 1	0 - 2	0 - 2	1 - 1	1 - 1	1 - 1	1 - 1	1 - 1	-	1 - 1
Wolves	1 - 1	2 - 1	1 - 0	1 - 0	1 - 0	1 - 1	1 - 1	1 - 1	2 - 0	2 - 0	1 - 1	0 - 1	0 - 1	1 - 1	1 - 0	1 - 0	1 - 1	1 - 1	1 - 0	-

Figure 3 - Simulated 2018 - 2019 Season League Table



Home \ Away	ARS	BOU	BHA	BUR	CAR	CHE	CRY	EVE	FUL	HUD	LEI	LIV	MCI	MUN	NEW	SOU	TOT	WAT	WHU	WOL
Arsenal	—	5–1	1–1	3–1	2–1	2–0	2–3	2–0	4–1	1–0	3–1	1–1	0–2	2–0	2–0	2–0	4–2	2–0	3–1	1–1
Bournemouth	1–2	—	2–0	1–3	2–0	4–0	2–1	2–2	0–1	2–1	4–2	0–4	0–1	1–2	2–2	0–0	1–0	3–3	2–0	1–1
Brighton & Hove Albion	1–1	0–5	—	1–3	0–2	1–2	3–1	1–0	2–2	1–0	1–1	0–1	1–4	3–2	1–1	0–1	1–2	0–0	1–0	1–0
Burnley	1–3	4–0	1–0	—	2–0	0–4	1–3	1–5	2–1	1–1	1–2	1–3	0–1	0–2	1–2	1–1	2–1	1–3	2–0	2–0
Cardiff City	2–3	2–0	2–1	1–2	—	1–2	2–3	0–3	4–2	0–0	0–1	0–2	0–5	1–5	0–0	1–0	0–3	1–5	2–0	2–1
Chelsea	3–2	2–0	3–0	2–2	4–1	—	3–1	0–0	2–0	5–0	0–1	1–1	2–0	2–2	2–1	0–0	2–0	3–0	2–0	1–1
Crystal Palace	2–2	5–3	1–2	2–0	0–0	0–1	—	0–0	2–0	2–0	1–0	0–2	1–3	1–3	0–0	0–2	0–1	1–2	1–1	0–1
Everton	1–0	2–0	3–1	2–0	1–0	2–0	2–0	—	3–0	1–1	0–1	0–0	0–2	4–0	1–1	2–1	2–6	2–2	1–3	1–3
Fulham	1–5	0–3	4–2	4–2	1–0	1–2	0–2	2–0	—	1–0	1–1	1–2	0–2	0–3	0–4	3–2	1–2	1–1	0–2	1–1
Huddersfield Town	1–2	0–2	1–2	1–2	0–0	0–3	0–1	0–1	1–0	—	1–4	0–1	0–3	1–1	0–1	1–3	0–2	1–2	1–1	1–0
Leicester City	3–0	2–0	2–1	0–0	0–1	0–0	1–4	1–2	3–1	3–1	—	1–2	2–1	0–1	0–1	1–2	0–2	2–0	1–1	2–0
Liverpool	5–1	3–0	1–0	4–2	4–1	2–0	4–3	1–0	2–0	5–0	1–1	—	0–0	3–1	4–0	3–0	2–1	5–0	4–0	2–0
Manchester City	3–1	3–1	2–0	5–0	2–0	6–0	2–3	3–1	3–0	6–1	1–0	2–1	—	3–1	2–1	6–1	1–0	3–1	1–0	3–0
Manchester United	2–2	4–1	2–1	2–2	0–2	1–1	0–0	2–1	4–1	3–1	2–1	0–0	0–2	—	3–2	3–2	0–3	2–1	2–1	1–1
Newcastle United	1–2	2–1	0–1	2–0	3–0	1–2	0–1	3–2	0–0	2–0	0–2	2–3	2–1	0–2	—	3–1	1–2	1–0	0–3	1–2
Southampton	3–2	3–3	2–2	0–0	1–2	0–3	1–1	2–1	2–0	1–1	1–2	1–3	1–3	2–2	0–0	—	2–1	1–1	1–2	3–1
Tottenham Hotspur	1–1	5–0	1–0	1–0	1–0	3–1	2–0	2–2	3–1	4–0	3–1	1–2	0–1	0–1	1–0	3–1	—	2–1	0–1	1–3
Watford	0–1	0–4	2–0	0–0	3–2	1–2	2–1	1–0	4–1	3–0	2–1	0–3	1–2	1–2	1–1	1–1	2–1	—	1–4	1–2
West Ham United	1–0	1–2	2–2	4–2	3–1	0–0	3–2	0–2	3–1	4–3	2–2	1–1	0–4	3–1	2–0	3–0	0–1	0–2	—	0–1
Wolverhampton Wanderers	3–1	2–0	0–0	1–0	2–0	2–1	0–2	2–2	1–0	0–2	4–3	0–2	1–1	2–1	1–1	2–0	2–3	0–2	3–0	—

Figure 4 - Actual 2018 - 2019 Season League Table[7]

[7] https://en.wikipedia.org/wiki/2018%E2%80%9319_Premier_League

Rank

Similar to the table command, the **rank** command simulates an entire double round robin tournament using all teams present in the input data. Instead of producing a league table, however, the rank command assigns each team points based on the likelihood of a victory or tie in each of their matches. The table given in Figure 5 below represents a ranking of all teams based on data from the 2010 - 2018.

Rank	Team Name	Expected Points
1	Man City	157.99
2	Chelsea	140.89
3	Man United	140.65
4	Liverpool	139.24
5	Arsenal	137.96
6	Tottenham	134.66
7	Everton	113.7
8	Leicester	108.28
9	Southampton	103.21
10	West Ham	94.97
11	Crystal Palace	93.79
12	Newcastle	91.82
13	Swansea	91.72
14	Stoke	89.53
15	West Brom	89.47
16	Bournemouth	89.36
17	Bolton	87.14
18	Watford	86.76
19	Blackpool	84.65
20	Blackburn	84.62
21	Sunderland	84.35
22	Burnley	83.4
23	Fulham	82.58
24	Wolves	82.55
25	Wigan	80.18
26	Birmingham	79.8
27	Norwich	79.23
28	Aston Villa	76.54
29	Brighton	75.61
30	Hull	74.26
31	QPR	74.04
32	Reading	73.7
33	Middlesbrough	68.08
34	Cardiff	60.3
35	Huddersfield	53.11

Figure 5 - Simulated Rankings 2010 - 2018

The table to the left was generated using the following command:

```
java -jar build/libs/eplranking.jar \
    rank \-d ./datasets \
    -f 2010 \
    -l 2018
```

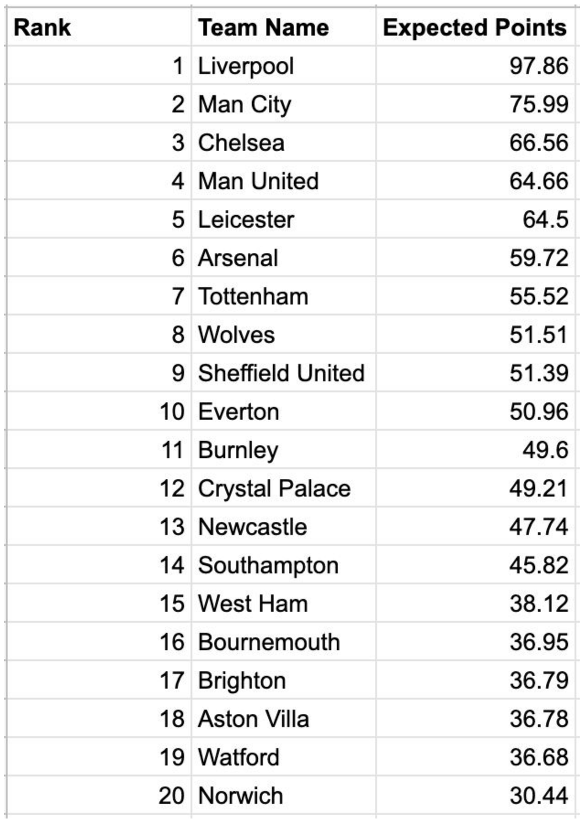
As shown, this application predicts that in a double round robin tournament of the 35 teams listed, Manchester City would be the overwhelming victor with Chelsea and Manchester United being very close to one another in second and third. This aligns well with reality since Manchester City won four of the nine seasons used as input data and finished second in the remaining five [8].

[8] https://en.wikipedia.org/wiki/List_of_Premier_League_seasons

Completing the 2019 - 2020 Season

The 2019 - 2020 EPL season was interrupted in March of 2020 due to the COVID-19 pandemic and safety concerns. This application is capable of applying the same simulation techniques described above to simulate the completion of an in-progress season. When run in this mode, the program first loads the actual data for those matches which have already occured. It then performs a full season (double round robin) simulation substituting actual match data in place of simulation data when available. Figure 6 below shows the expected rankings at the completion of the 2019 - 2020 season based on data from the 2000 - 2018 seasons. The table was generated using the following command:

```
java -jar build/libs/eplranking.jar \
    rank \
    -d ./datasets \
    -f 2000 \
    -l 2018 \
    -p ./datasets/2019-2020-partial.csv
```

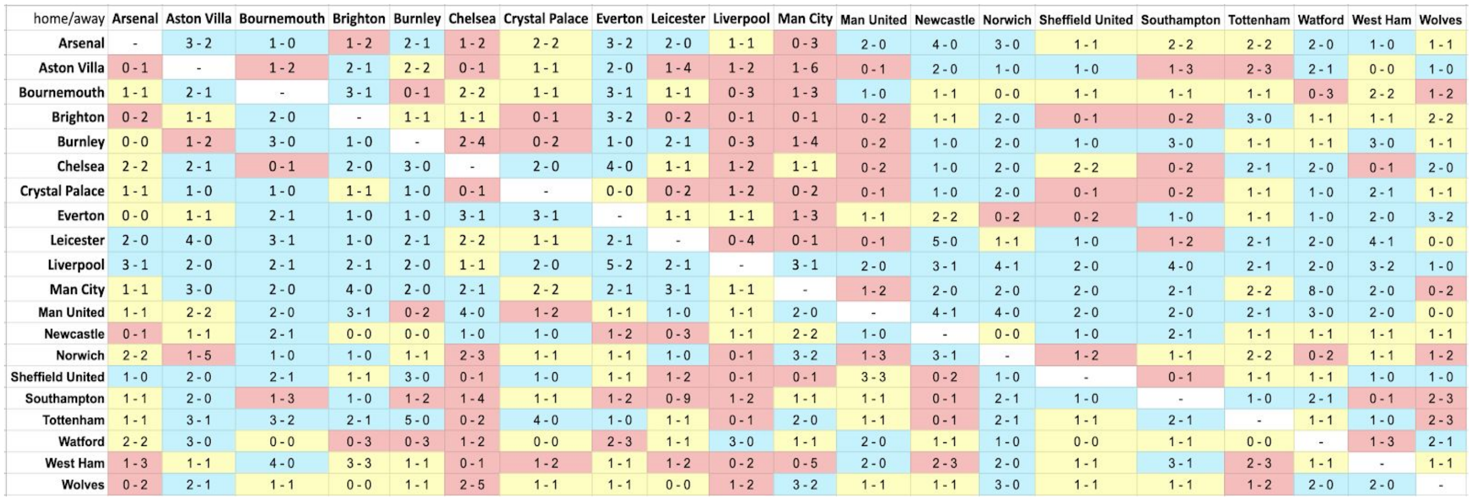


Rank	Team Name	Expected Points
1	Liverpool	97.86
2	Man City	75.99
3	Chelsea	66.56
4	Man United	64.66
5	Leicester	64.5
6	Arsenal	59.72
7	Tottenham	55.52
8	Wolves	51.51
9	Sheffield United	51.39
10	Everton	50.96
11	Burnley	49.6
12	Crystal Palace	49.21
13	Newcastle	47.74
14	Southampton	45.82
15	West Ham	38.12
16	Bournemouth	36.95
17	Brighton	36.79
18	Aston Villa	36.78
19	Watford	36.68
20	Norwich	30.44

Figure 6 - 2019 - 2020 Final Rankings

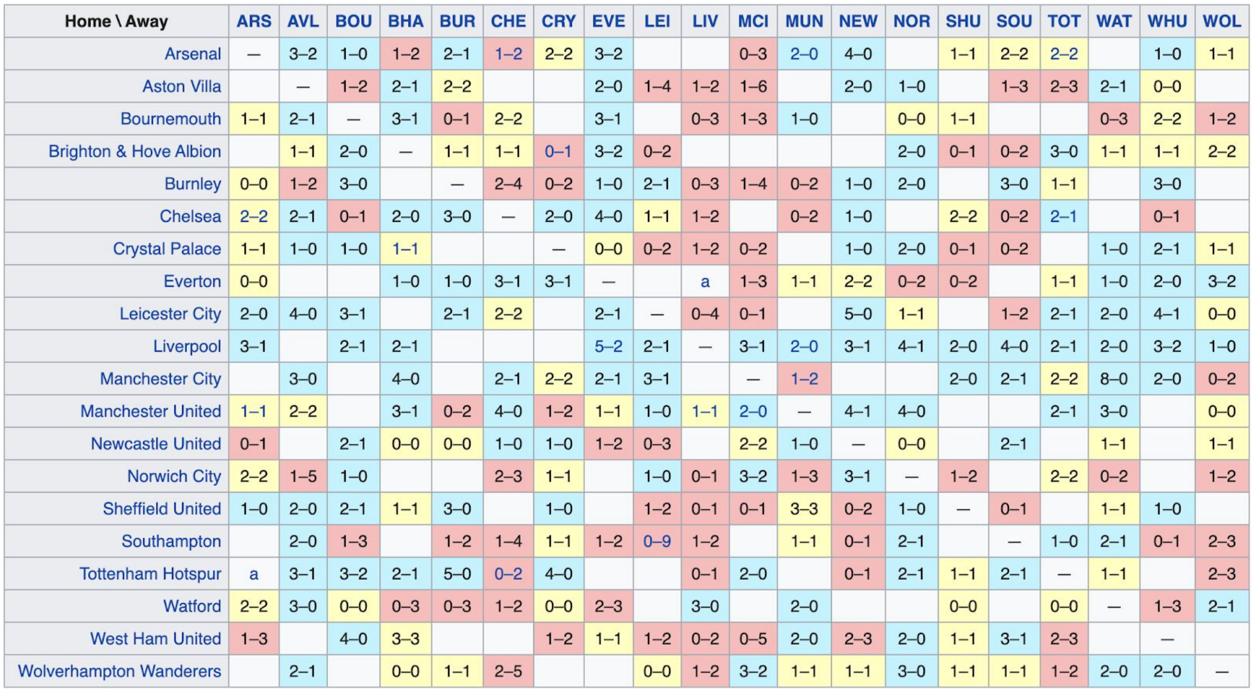
As shown in the table, the season is expected to close with Liverpool on top. The league tables on the next page provide additional insight into this outcome. Figure 7 shows the completed league table generated by the simulation and Figure 8 shows the actual league table including those matches which were completed prior to the season being paused. The simulated table was generated with the following command:

```
java -jar
build/libs/eplranking.jar \
    rank \
    -d ./datasets \
    -f 2000 \
    -l 2018 \
    -p
./datasets/2019-2020-partial.csv
```



home/away	Arsenal	Aston Villa	Bournemouth	Brighton	Burnley	Chelsea	Crystal Palace	Everton	Leicester	Liverpool	Man City	Man United	Newcastle	Norwich	Sheffield United	Southampton	Tottenham	Watford	West Ham	Wolves
Arsenal	-	3 - 2	1 - 0	1 - 2	2 - 1	1 - 2	2 - 2	3 - 2	2 - 0	1 - 1	0 - 3	2 - 0	4 - 0	3 - 0	1 - 1	2 - 2	2 - 2	2 - 0	1 - 0	1 - 1
Aston Villa	0 - 1	-	1 - 2	2 - 1	2 - 2	0 - 1	1 - 1	2 - 0	1 - 4	1 - 2	1 - 6	0 - 1	2 - 0	1 - 0	1 - 0	1 - 3	2 - 3	2 - 1	0 - 0	1 - 0
Bournemouth	1 - 1	2 - 1	-	3 - 1	0 - 1	2 - 2	1 - 1	3 - 1	1 - 1	0 - 3	1 - 3	1 - 0	1 - 1	0 - 0	1 - 1	1 - 1	1 - 1	0 - 3	2 - 2	1 - 2
Brighton	0 - 2	1 - 1	2 - 0	-	1 - 1	1 - 1	0 - 1	3 - 2	0 - 2	0 - 1	0 - 1	0 - 2	1 - 1	2 - 0	0 - 1	0 - 2	3 - 0	1 - 1	1 - 1	2 - 2
Burnley	0 - 0	1 - 2	3 - 0	1 - 0	-	2 - 4	0 - 2	1 - 0	2 - 1	0 - 3	1 - 4	0 - 2	1 - 0	2 - 0	1 - 0	3 - 0	1 - 1	1 - 1	3 - 0	1 - 1
Chelsea	2 - 2	2 - 1	0 - 1	2 - 0	3 - 0	-	2 - 0	4 - 0	1 - 1	1 - 2	1 - 1	0 - 2	1 - 0	2 - 0	2 - 2	0 - 2	2 - 1	2 - 0	0 - 1	2 - 0
Crystal Palace	1 - 1	1 - 0	1 - 0	1 - 1	1 - 0	0 - 1	-	0 - 0	0 - 2	1 - 2	0 - 2	0 - 1	1 - 0	2 - 0	0 - 1	0 - 2	1 - 1	1 - 0	2 - 1	1 - 1
Everton	0 - 0	1 - 1	2 - 1	1 - 0	1 - 0	3 - 1	3 - 1	-	1 - 1	1 - 1	1 - 3	1 - 1	2 - 2	0 - 2	0 - 2	1 - 0	1 - 1	1 - 0	2 - 0	3 - 2
Leicester	2 - 0	4 - 0	3 - 1	1 - 0	2 - 1	2 - 2	1 - 1	2 - 1	-	0 - 4	0 - 1	0 - 1	5 - 0	1 - 1	1 - 0	1 - 2	2 - 1	2 - 0	4 - 1	0 - 0
Liverpool	3 - 1	2 - 0	2 - 1	2 - 1	2 - 0	1 - 1	2 - 0	5 - 2	2 - 1	-	3 - 1	2 - 0	3 - 1	4 - 1	2 - 0	4 - 0	2 - 1	2 - 0	3 - 2	1 - 0
Man City	1 - 1	3 - 0	2 - 0	4 - 0	2 - 0	2 - 1	2 - 2	2 - 1	3 - 1	1 - 1	-	1 - 2	2 - 0	2 - 0	2 - 0	2 - 1	2 - 2	8 - 0	2 - 0	0 - 2
Man United	1 - 1	2 - 2	2 - 0	3 - 1	0 - 2	4 - 0	1 - 2	1 - 1	1 - 0	1 - 1	2 - 0	-	4 - 1	4 - 0	2 - 0	2 - 0	2 - 1	3 - 0	2 - 0	0 - 0
Newcastle	0 - 1	1 - 1	2 - 1	0 - 0	0 - 0	1 - 0	1 - 0	1 - 2	0 - 3	1 - 1	2 - 2	1 - 0	-	0 - 0	1 - 0	2 - 1	1 - 1	1 - 1	1 - 1	1 - 1
Norwich	2 - 2	1 - 5	1 - 0	1 - 0	1 - 1	2 - 3	1 - 1	1 - 1	1 - 0	0 - 1	3 - 2	1 - 3	3 - 1	-	1 - 2	1 - 1	2 - 2	0 - 2	1 - 1	1 - 2
Sheffield United	1 - 0	2 - 0	2 - 1	1 - 1	3 - 0	0 - 1	1 - 0	1 - 1	1 - 2	0 - 1	0 - 1	3 - 3	0 - 2	1 - 0	-	0 - 1	1 - 1	1 - 1	1 - 0	1 - 0
Southampton	1 - 1	2 - 0	1 - 3	1 - 0	1 - 2	1 - 4	1 - 1	1 - 2	0 - 9	1 - 2	1 - 1	1 - 1	0 - 1	2 - 1	1 - 0	-	1 - 0	2 - 1	0 - 1	2 - 3
Tottenham	1 - 1	3 - 1	3 - 2	2 - 1	5 - 0	0 - 2	4 - 0	1 - 0	1 - 1	0 - 1	2 - 0	1 - 1	0 - 1	2 - 1	1 - 1	2 - 1	-	1 - 1	1 - 0	2 - 3
Watford	2 - 2	3 - 0	0 - 0	0 - 3	0 - 3	1 - 2	0 - 0	2 - 3	1 - 1	3 - 0	1 - 1	2 - 0	1 - 1	1 - 0	0 - 0	1 - 1	0 - 0	-	1 - 3	2 - 1
West Ham	1 - 3	1 - 1	4 - 0	3 - 3	1 - 1	0 - 1	1 - 2	1 - 1	1 - 2	0 - 2	0 - 5	2 - 0	2 - 3	2 - 0	1 - 1	3 - 1	2 - 3	1 - 1	-	1 - 1
Wolves	0 - 2	2 - 1	1 - 1	0 - 0	1 - 1	2 - 5	1 - 1	1 - 1	0 - 0	1 - 2	3 - 2	1 - 1	1 - 1	3 - 0	1 - 1	1 - 1	1 - 2	2 - 0	2 - 0	-

Figure 7 - League Table with Unfinished Matches Simulated



Home \ Away	ARS	AVL	BOU	BHA	BUR	CHE	CRY	EVE	LEI	LIV	MCI	MUN	NEW	NOR	SHU	SOU	TOT	WAT	WHU	WOL
Arsenal	—	3–2	1–0	1–2	2–1	1–2	2–2	3–2			0–3	2–0	4–0		1–1	2–2	2–2		1–0	1–1
Aston Villa		—	1–2	2–1	2–2			2–0	1–4	1–2	1–6		2–0	1–0		1–3	2–3	2–1	0–0	
Bournemouth	1–1	2–1	—	3–1	0–1	2–2		3–1		0–3	1–3	1–0		0–0	1–1			0–3	2–2	1–2
Brighton & Hove Albion		1–1	2–0	—	1–1	1–1	0–1	3–2	0–2					2–0	0–1	0–2	3–0	1–1	1–1	2–2
Burnley	0–0	1–2	3–0		—	2–4	0–2	1–0	2–1	0–3	1–4	0–2	1–0	2–0		3–0	1–1		3–0	
Chelsea	2–2	2–1	0–1	2–0	3–0	—	2–0	4–0	1–1	1–2		0–2	1–0		2–2	0–2	2–1		0–1	
Crystal Palace	1–1	1–0	1–0	1–1			—	0–0	0–2	1–2	0–2		1–0	2–0	0–1	0–2		1–0	2–1	1–1
Everton	0–0			1–0	1–0	3–1	3–1	—		a	1–3	1–1	2–2	0–2	0–2		1–1	1–0	2–0	3–2
Leicester City	2–0	4–0	3–1		2–1	2–2		2–1	—	0–4	0–1		5–0	1–1		1–2	2–1	2–0	4–1	0–0
Liverpool	3–1		2–1	2–1				5–2	2–1	—	3–1	2–0	3–1	4–1	2–0	4–0	2–1	2–0	3–2	1–0
Manchester City		3–0		4–0		2–1	2–2	2–1	3–1		—	1–2			2–0	2–1	2–2	8–0	2–0	0–2
Manchester United	1–1	2–2		3–1	0–2	4–0	1–2	1–1	1–0	1–1	2–0	—	4–1	4–0			2–1	3–0		0–0
Newcastle United	0–1		2–1	0–0	0–0	1–0	1–0	1–2	0–3		2–2	1–0	—	0–0		2–1		1–1		1–1
Norwich City	2–2	1–5	1–0			2–3	1–1		1–0	0–1	3–2	1–3	3–1	—	1–2		2–2	0–2		1–2
Sheffield United	1–0	2–0	2–1	1–1	3–0		1–0		1–2	0–1	0–1	3–3	0–2	1–0	—	0–1		1–1	1–0	
Southampton		2–0	1–3		1–2	1–4	1–1	1–2	0–9	1–2		1–1	0–1	2–1		—	1–0	2–1	0–1	2–3
Tottenham Hotspur	a	3–1	3–2	2–1	5–0	0–2	4–0			0–1	2–0		0–1	2–1	1–1	2–1	—	1–1		2–3
Watford	2–2	3–0	0–0	0–3	0–3	1–2	0–0	2–3		3–0		2–0			0–0		0–0	—	1–3	2–1
West Ham United	1–3		4–0	3–3			1–2	1–1	1–2	0–2	0–5	2–0	2–3	2–0	1–1	3–1	2–3		—	
Wolverhampton Wanderers		2–1		0–0	1–1	2–5			0–0	1–2	3–2	1–1	1–1	3–0	1–1	1–1	1–2	2–0	2–0	—

Figure 8 - Actual League Table with incomplete Matches

Comparing the two tables above shows that all completed matches in Figure 8 are reflected with their actual scores in Figure 7. This highlights the fact that the simulator used actual match data for completed matches and simulated the remaining matches.

Summary

The results generated by this project indicate that the use of Poisson probability mass function is a suitable means of predicting football matches. In all cases given above, the simulator produced results in alignment with real-world data and expectations. The program also proved flexible and full featured, allowing for varying types of simulation and outputs making cross-examination and verification possible. Lastly, the Java based implementation utilizes the most applicable features of the language to produce clean, extensible, and efficient code.