

Soccer Match Outcome Prediction

Gaurav Derasaria, Sharad Ghule, Aniket Shenoy, and Prathamesh Jangam

School of Informatics, Computing and Engineering, Indiana University
Bloomington, USA, 47401

ABSTRACT

Soccer is a sport widely played and followed throughout Europe. We take data of 380 matches of English Premier League for the season of 2014 across 20 teams and try to build a multinomial logistic regression model that would predict the outcome of any match which can be win, loss or a draw. We use team attributes related to the play styles of both teams, select relevant features through exploratory data analysis, use feature engineering to build new features and try different models that would give us the highest accuracy. We are able to predict the outcome with 59.65% accuracy using a *Proportional Odds Logistic Regression* model which is higher than the accuracy with which bookies who take bets on the matches can predict.

Introduction

Soccer is one of the most followed sports in the world. Especially in Europe and South American countries, more than a game, soccer is a religion. Among the many European leagues, the English Premier League (EPL) is the most popular. In fact, the EPL is the most-watched sports league in the world, thereby generating millions in revenue. Betting on the outcome of matches is legal in Europe and is a billion-dollar industry.

Goals:

The goal of this project is to predict the outcome of a match based on records of previous performances. We want to explore the potential of exploratory data analysis to maximize the odds of winning a bet.

Data Description:

The dataset comprised of an SQLite database with tables of matches, players and team attributes. Below is a summary of the dataset:

- +25,000 matches
- +10,000 players
- 11 European Countries
- Seasons 2008 to 2016
- Players and Teams' attributes sourced from EA Sports' FIFA video game series

Below are the snapshots of the raw data before preprocessing:

Figure 1: Matches Table

id	country_id	league_id	season	stage	date	match_api_id	home_team_api_id	away_team_api_id	home_team_goal	away_team_goal
1	1	1	2008/2009	1	2008-08-17 00:00:00	492473	9987	9993	1	1
2	1	1	2008/2009	1	2008-08-16 00:00:00	492474	10000	9994	0	0
3	1	1	2008/2009	1	2008-08-16 00:00:00	492475	9984	8635	0	3
4	1	1	2008/2009	1	2008-08-17 00:00:00	492476	9991	9998	5	0
5	1	1	2008/2009	1	2008-08-16 00:00:00	492477	7947	9985	1	3
6	1	1	2008/2009	1	2008-09-24 00:00:00	492478	8203	8342	1	1

Figure 2: Team Attributes

id	team_fifa_api_id	team_api_id	date	buildUpPlaySpeed	buildUpPlaySpeedClass	buildUpPlayDribbling	buildUpPlayDribblingClass	buildUpPlayPassing	buildUpPlayPassingClass
1	434	9930	2010-02-22 00:00:00	60	Balanced		Little	50	Mixed
2	434	9930	2014-09-19 00:00:00	52	Balanced	48	Normal	56	Mixed
3	434	9930	2015-09-10 00:00:00	47	Balanced	41	Normal	54	Mixed
4	77	8485	2010-02-22 00:00:00	70	Fast		Little	70	Long
5	77	8485	2011-02-22 00:00:00	47	Balanced		Little	52	Mixed
6	77	8485	2012-02-22 00:00:00	58	Balanced		Little	62	Mixed

Data preprocessing:

We are working on a subset of the data that consists of the Matches table joined with the Team Attributes Table and we are only considering the `_English Premier League_`. Below is a snapshot of the data we are working with:

- 380 observations: Since 20 teams, each playing 2 matches against every other team, we have records of 380 matches.
- 18 numerical variables (on a scale of 100) represent one attribute for each team: Since there would have been duplication of attributes, 1 for each team, we decided to take a ratio of these attributes.

- 9 variables represent a ratio of an attribute for home vs. away team: After taking ratios of corresponding attributes, we ended up having 9 attributes of form (home_team_attribute/away_team_attribute)
- 1 multinomial predicted variable (win/loss/draw): Outcome of a match depends on the number of goals score by each team. We converted the number of goals by each team in 2 separate columns to a single column.
- Examples of attributes: buildUpPlaySpeed_ratio, buildUpPlayDribbling_ratio, defenceAggression_ratio, defenceTeamWidth_ratio

Below is a summary of the final dataset:

```

id_x      season      home_team_api_id away_team_api_id result  buildUpPlaySpeed_ratio buildUpPlayDribbling_ratio
Min. :4009   Length:380      Min. : 8191   Min. : 8191   0:172   Min. :0.5694   Min. :0.4615
1st Qu.:4104   Class :character 1st Qu.: 8470   1st Qu.: 8470   1: 93   1st Qu.:0.8571   1st Qu.:0.8953
Median :4198   Mode :character  Median : 8663   Median : 8663   2:115   Median :1.0000   Median :1.0000
Mean :4198      Mean : 9146      Mean : 9146      Mean :1.0238   Mean :1.0313
3rd Qu.:4293      3rd Qu.:10045    3rd Qu.:10045    3rd Qu.:1.1667   3rd Qu.:1.1170
Max. :4388      Max. :10261      Max. :10261      Max. :1.7561   Max. :2.1667

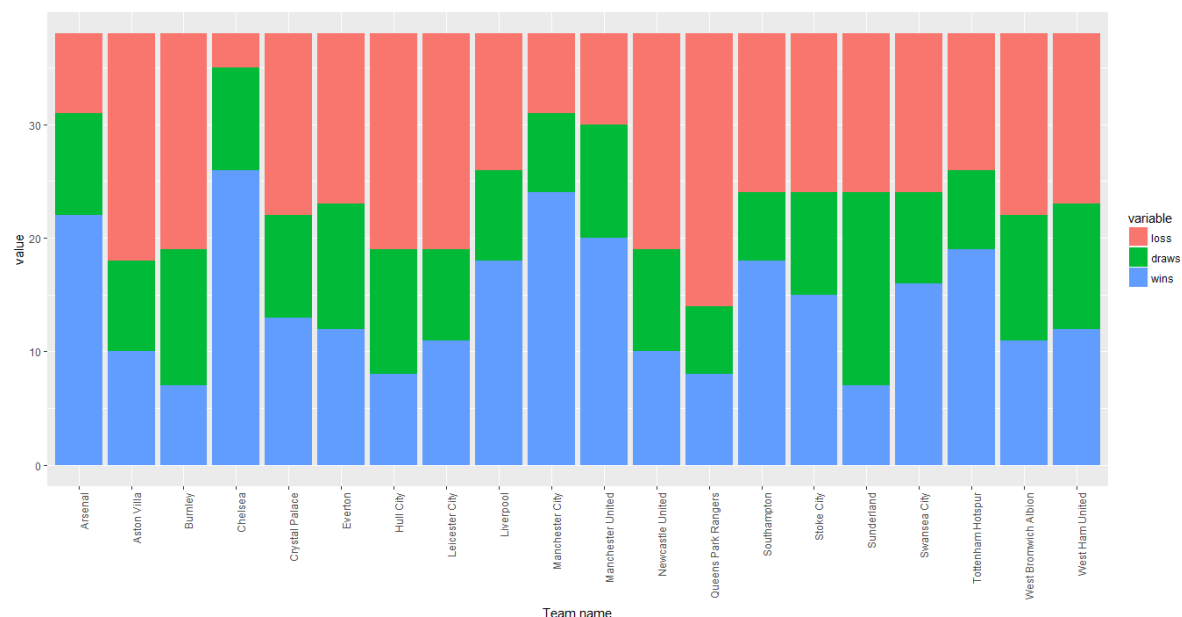
buildUpPlayPassing_ratio chanceCreationPassing_ratio chanceCreationCrossing_ratio chanceCreationShooting_ratio defencePressure_ratio
Min. :0.3514   Min. :0.3889   Min. :0.5694   Min. :0.4638   Min. :0.4839
1st Qu.:0.7258   1st Qu.:0.7640   1st Qu.:0.8644   1st Qu.:0.8276   1st Qu.:0.8039
Median :1.0000   Median :1.0000   Median :1.0000   Median :1.0000   Median :1.0000
Mean :1.0998   Mean :1.0719   Mean :1.0201   Mean :1.0489   Mean :1.0389
3rd Qu.:1.3778   3rd Qu.:1.3089   3rd Qu.:1.1569   3rd Qu.:1.2083   3rd Qu.:1.2439
Max. :2.8462   Max. :2.5714   Max. :1.7561   Max. :2.1562   Max. :2.0667

defenceAggression_ratio defenceTeamWidth_ratio home_team_name away_team_name
Min. :0.5312   Min. :0.6613   Length:380   Length:380
1st Qu.:0.8206   1st Qu.:0.8913   Class :character   Class :character
Median :1.0000   Median :1.0000   Mode :character   Mode :character
Mean :1.0393   Mean :1.0136
3rd Qu.:1.2186   3rd Qu.:1.1220
Max. :1.8824   Max. :1.5122
>

```

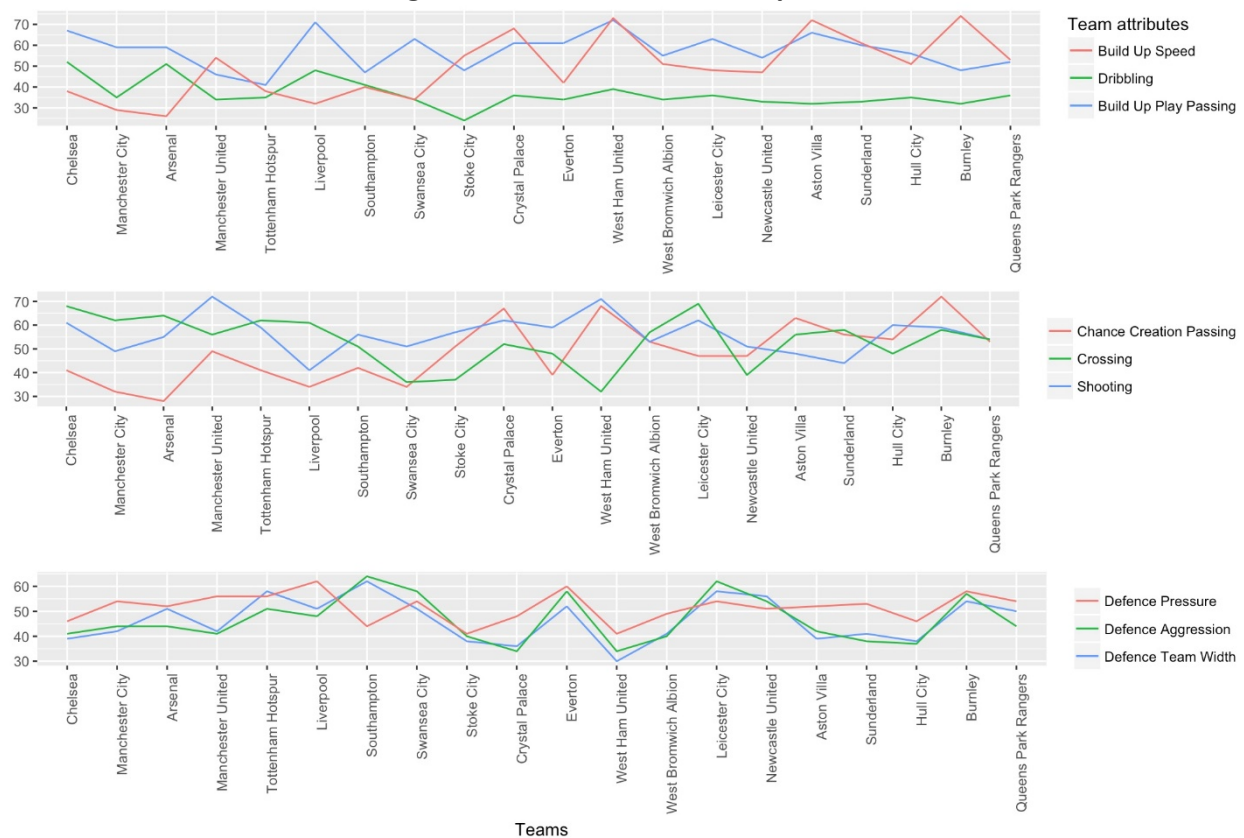
Exploring the dataset:

Figure 3. Wins/losses/draws



To get acquainted with the data, we try to plot the wins/draws/losses of all teams for the season. We see that every team plays 38 matches and there are 20 teams participating in the league. Thus, every team plays two matches with every other team in the league. After transforming the data so that it represents results for each team, we observe that the win/draw to loss ratio for team Chelsea seems to be the highest. We also know from our data that Chelsea won the league in that season. The team with the lowest wins/draws Queen Park Rangers have seemed to lose the season. Later on, we explore our data to see how the attributes of every team matter towards the result of a season.

Figure 4. Team Attribute Comparison

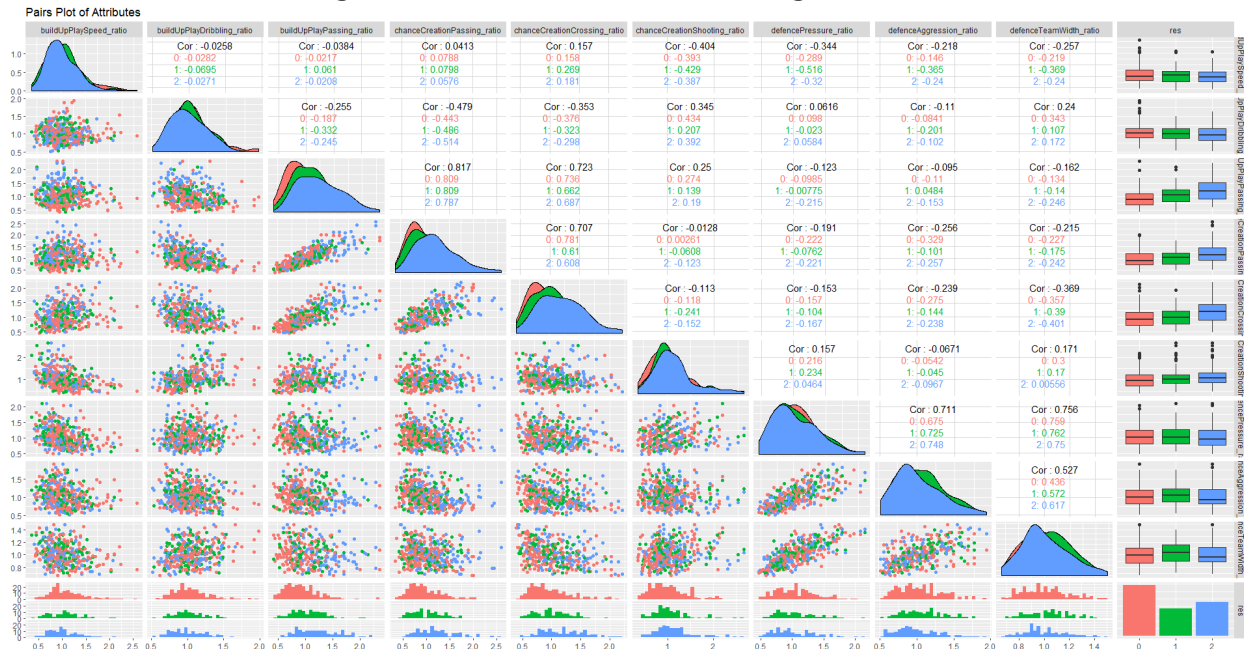


As we can see from above graph, the x-axis represents the teams in the league sorted on the total number of points in a descending order. The number of points of any team are equal to their wins multiplied by three plus the number of their draws. It means that Chelsea score the highest points and Queen Park Rangers scoring the lowest points.

We faceted our plot to show three variables in each graph. We can argue that the winning team does not have higher attributes than the losing team or any other team. Majority attributes of teams do not actually seem to follow a pattern across the teams. On the other hand, we can also see that some attributes do follow a pattern. For example, we do see that dribbling of a team that has won the season seems to be higher and decreases as the points of teams go on

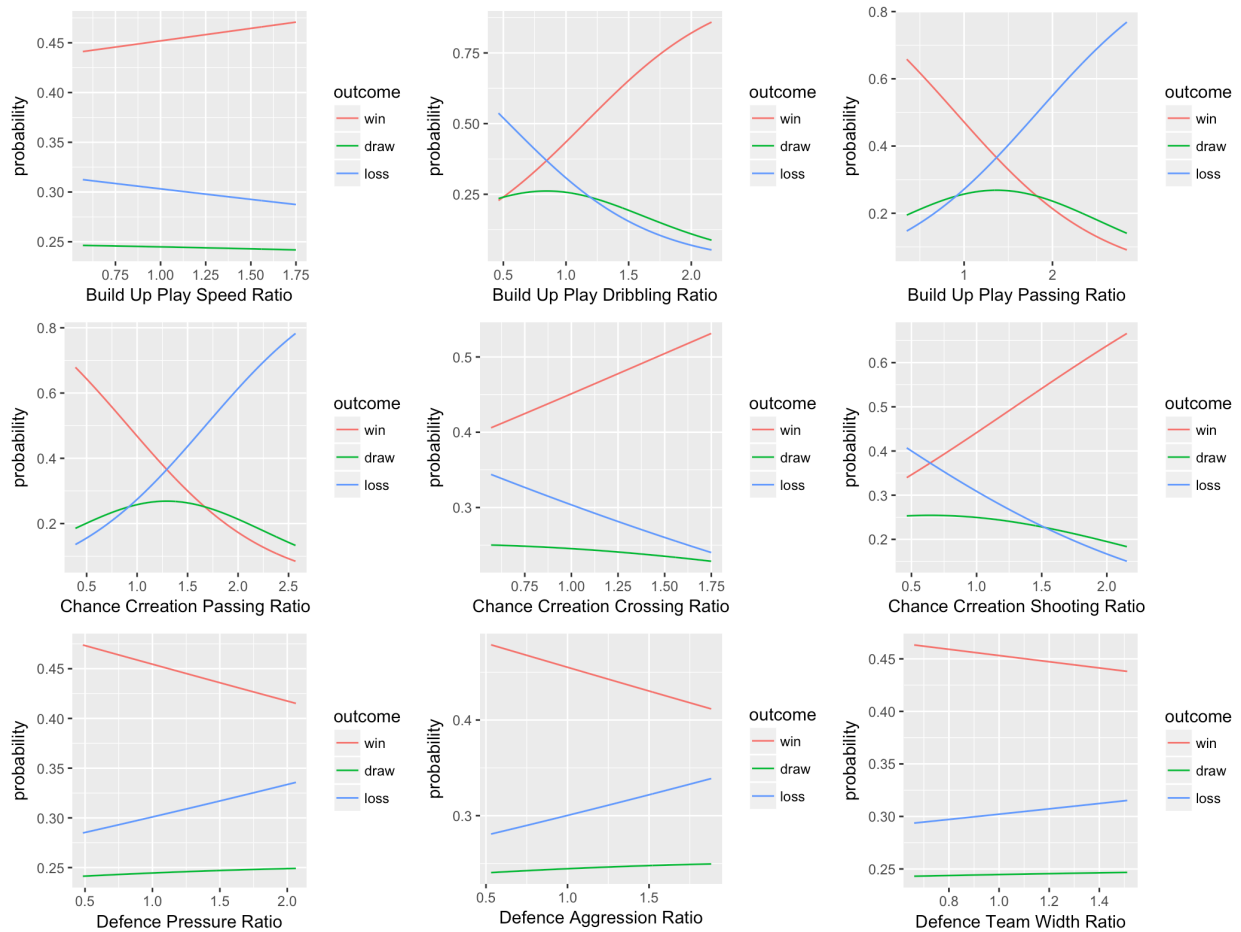
decreasing. Some attributes also actually seem to have an inverse relation. We can see that the chance creation passing seems to increase as the points scored by a team decreases. Although the attributes do not seem to follow a pattern, we will still try to fit various models and continue our analysis to see if we can predict if a team wins, based on the ratio of its attributes to the attributes of the opposite team. In the next graph, we see how the ratios of attributes of different teams correlate with each other and the different classes they belong to.

Figure 5: Pairwise Correlation Among Attributes



We plotted the pairwise correlation between the ratios of attributes. The ratio of any attribute represents $\text{Home_Team_Attribute} / \text{Away_Team_Attribute}$ and the prediction Win or Loss represents if the home team wins or home team losses respectively. We plot this graph to see which of our predictor variables have a high correlation. We can see that Build Up Play Passing ratio and the Chance Creation Passing ratio seem to be highly correlated. We can also see that Defense Team Width and Defense Pressure are highly correlated. Chance Creation Crossing and Defense Team Width also seem to be highly correlated with other few attributes. We can thus see if dropping these variables can increase the accuracy of our model. We can also infer from the graph that no two of our variables together separate our result classes. We now try to perform a *Proportional Odds Logistic Regression* on our data and try to predict the outcome of the match. We first try to fit a *polr* model on our attributes individually and see how individual attributes matter towards the predictions.

Figure 6: Attribute Selection



We decided to see how the probability of every outcome varies with increase in each attribute. The x-axis represents a range of that attribute (from the minimum to maximum value of the attribute) and the y axis represents the probability of an outcome. We can see that 5 out of 9 variables do not actually contribute to the outcome at all. Some attributes always give the highest probability of winning. Thus, we can definitely say that we cannot use individual attributes to predict the outcome but we can also see that the graph of 'Build Up Play Passing' and 'Chance Crration Passing' seems to be quite similar. We can see in both attributes that as the attributes increase, the probability of the team winning decreases. Thus, we can conclude that dropping either variable might not affect our results much. In attributes, chance creation shooting and build play dribbling as the ratio increases, the probability of winning increases too.

Methods

POLR:

Since our response is an ordered categorical variable, we went ahead with ordered categorical regression using proportional odds logistic regression. The data was split into train and test sets with an 80:20 ratio.

1. Model 1

As a preliminary model, we fit a polr model using all team attributes as dependent variables.

```
Call:
polr(formula = factor(result) ~ buildupPlaySpeed_ratio + buildupPlayDribbling_ratio +
      buildupPlayPassing_ratio + chanceCreationPassing_ratio +
      chanceCreationCrossing_ratio + chanceCreationShooting_ratio +
      defencePressure_ratio + defenceAggression_ratio + defenceTeamwidth_ratio,
      data = m5.train)

Coefficients:
      buildupPlaySpeed_ratio      buildupPlayDribbling_ratio      buildupPlayPassing_ratio      chanceCreationPassing_ratio
              2.6967071              -2.1357843              -0.9516667              2.6574128
chanceCreationCrossing_ratio      chanceCreationShooting_ratio      defencePressure_ratio      defenceAggression_ratio
              0.1330835              -0.6296298              4.8194513              -2.7112242
      defenceTeamwidth_ratio
              -0.2206415

Intercepts:
      0|1      1|2
3.584093 4.823948

Residual Deviance: 526.1071
AIC: 548.1071
```

According to Occam's razor, a simpler model is a better model. Thus, we further analyse if we can reduce the complexity of the model by dropping some variables.

2. Model 2

In the pairwise correlation plot in Figure 3, it can be seen that there is a high correlation between chanceCreationPassing_ratio and chanceCreationCrossing_ratio. Thus, we wanted to see if there would be a significant impact on the model when we drop any of these variables. In this model, we drop chanceCreationPassing_ratio.

```
Call:
polr(formula = factor(result) ~ buildupPlaySpeed_ratio + buildupPlayDribbling_ratio +
      buildupPlayPassing_ratio + chanceCreationCrossing_ratio +
      chanceCreationShooting_ratio + defencePressure_ratio + defenceAggression_ratio +
      defenceTeamwidth_ratio, data = m5.train)

Coefficients:
      buildupPlaySpeed_ratio      buildupPlayDribbling_ratio      buildupPlayPassing_ratio      chanceCreationCrossing_ratio
              2.45490440              -1.84401913              1.29024966              0.07907531
chanceCreationShooting_ratio      defencePressure_ratio      defenceAggression_ratio      defenceTeamwidth_ratio
              -0.48852067              4.89892865              -2.79591060              -0.62316682

Intercepts:
      0|1      1|2
2.930679 4.163629

Residual Deviance: 528.1576
AIC: 548.1576
```

3. Model 3

It can also be seen in Figure 3 that there is a high correlation between defenceTeamPressure_ratio and defenceTeamWidth_ratio. In Model 3, we drop defenceTeamWidth_ratio and chanceCreationCrossing_ratio.

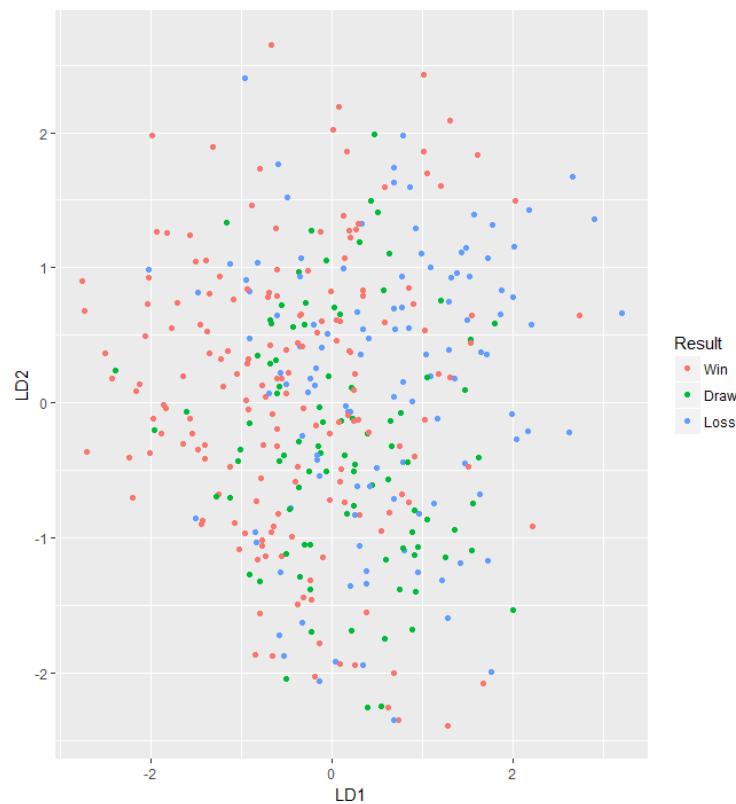
```
Call:
polr(formula = factor(result) ~ buildupPlayspeed_ratio + buildupPlayDribbling_ratio +
      buildupPlayPassing_ratio + chanceCreationPassing_ratio +
      chanceCreationShooting_ratio + defencePressure_ratio + defenceAggression_ratio,
      data = m5.train)

Coefficients:
      buildupPlayspeed_ratio      buildupPlayDribbling_ratio      buildupPlayPassing_ratio      chanceCreationPassing_ratio
      2.5468408          -2.0357646          -1.0158987          2.7566533
chanceCreationShooting_ratio      defencePressure_ratio      defenceAggression_ratio
      -0.6905324          4.5800576          -2.5772568

Intercepts:
      0|1      1|2
3.482558 4.722224

Residual Deviance: 526.1882
AIC: 544.1882
```

LDA:



We performed a Linear Discriminant Analysis to find any linear combination between our predictor variables. The first plot describes the two generated discriminants. As we can see from

the graph both our discriminants do no separate our classes at all. We still proceeded to fit the model and check the accuracy of our predictions. We split our data 80:20 for train and test respectively. We ran this 100 times and generated mean error and mean accuracy. We also tried to fit a Mixture Discriminant Analysis using the same method.

Results

1. Model 1

	0	2
0	42	8
1	16	8
2	14	26

Accuracy: 59.649%

2. Model 2

	0	2
0	41	9
1	15	9
2	17	23

Accuracy: 56.14%

3. Model 3

	0	2
0	42	8
1	16	8
2	14	26

Accuracy: 59.649%

It can be seen that Models 1 and 3 give us the same results and Model 2 gives us a lower accuracy. Also, the AIC seems to be more for Model 2. This shows that the variable `chanceCreationPassing_ratio` is an important dependent variable whereas `chanceCreationCrossing_ratio` and `defenceTeamWidth_ratio` do not seem to contribute much. Thus, we go ahead with Model 3 as it is the least complex and still gives us the highest accuracy.

Conclusion and Discussion

When it comes to match outcome predictions, Bookies get it right about 53% of the times. As of now, our final model is slightly better than the benchmark. There is a bias towards giving a win or loss over a draw and we believe that it is difficult to remove this bias through a logistic regression model.

As mentioned earlier, we are currently only dealing with the English Premier League. So in our future work, we would like to expand our analysis across various other European leagues. We would also like to carry out time series analysis across seasons to see whether there is any trend fit or other time component involved that contributes to variation in match outcomes. Another thing we would like to do is consider individual player attributes as a team's playing squad can change from match to match and across seasons. To conclude, once we have a robust model, we would like to obtain a dataset which is sampled more frequently that covers real time statistics of matches (for example, weekly instead of yearly).

References

Dataset: <https://www.kaggle.com/hugomathien/soccer>