



# Top 4 Rank Prediction & Analysis

Hyunil Yoo

# Overview

1. Why I chose this topic
2. How I'm going to approach this topic
3. Analysis
4. What are the limits and future work

# Why I chose this topic?

I'm a huge soccer and Chelsea FC fan, and I was wondering about what are the crucial factors that teams must have to win the league. Is it number passes, pass accuracy, shot on target, and so on.

What factors should a team to focus on?

It will be interesting to know what factors are actually affecting matches and use that insight to improve team's performances.

In addition, I'm curious if I can predict the top 4 teams and rank according to how teams do until December which is a half mark in season.

# Approach

1. Getting data from the official EPL website
  - a. Web scraping
2. Exploring dataset
  - a. Data Wrangling
  - b. EDA
  - c. Selecting features
3. Test ML models
4. Predict the Top 4 rank
5. Conclusion/ Limitation

# Getting data from official website by web scraping

## General table data 2010 to 2017

Tables

First Team

PL2

U18











Filter by Competition  
Premier League

Filter by Season  
2017/18


Filter by Matchweek  
All Matchweeks

Filter by Home or Away  
All Matches

Premier League

More	Position	Club	Played	Won	Drawn	Lost	GF	GA
▼	1 ●	 Manchester City	29	25	3	1	83	20
▼	2 ●	 Manchester United	29	19	5	5	56	22
▼	3 ●	 Liverpool	29	17	9	3	67	32
▼	4 ●	 Tottenham Hotspur	29	17	7	5	55	24
▼	5 ●	 Chelsea	29	16	5	8	50	26
▼	6 ●	 Arsenal	29	13	6	10	52	41
▼	7 ●	 Burnley	29	10	10	9	24	26
▼	8 ●	 Leicester City	29	9	10	10	41	42
▼	9 ▲	 Watford	29	10	6	13	39	47
▼	10 ▲	 Brighton and Hove Albion	29	8	10	11	28	38





## Club's statistics data



# Chelsea

Stamford Bridge

Official Website: [www.chelseafc.com](http://www.chelseafc.com)

Overview

Squad

Fixtures

Results

Stats

Tickets

Stadium

Season History

Filter by Season  
 2017/18

Reset Filters

Matches Played	Wins	Losses	Goals	Goals Conceded	Clean Sheets
29	16	8	50	26	14

### Attack

Goals	50
Goals Per Match	1.72
Shots	458
Shots On Target	175
Shooting Accuracy %	38%
Penalties Scored	3
Big Chances Created	46

### Team Play

Passes	16,132
Passes Per Match	556.28
Pass Accuracy %	84%
Crosses	565
Cross Accuracy %	20%

### Defence

Clean Sheets	14
Goals Conceded	26
Goals Conceded Per Match	0.90
Saves	59
Tackles	501
Tackle Success %	67%
Blocked Shots	126



# Example of table data

General table data 2010 to 2017

	club_name	drawn	goal	goal_against	lost	points	position	won
0	Manchester City	3	83	20	1	78	1	25
1	Manchester United	5	58	23	5	65	2	20
2	Liverpool	9	68	34	4	60	3	17
3	Tottenham Hotspur	7	55	24	5	58	4	17
4	Chelsea	5	52	27	8	56	5	17
5	Arsenal	6	52	41	10	45	6	13
6	Burnley	10	27	26	9	43	7	11
7	Leicester City	10	45	43	10	40	8	10
8	Everton	7	35	49	13	37	9	10
9	Watford	6	39	47	13	36	10	10
10	Brighton and Hove Albion	10	28	40	12	34	11	8
11	Bournemouth	9	34	44	12	33	12	8
12	Newcastle United	8	30	40	14	32	13	8
13	Swansea City	7	25	42	15	31	14	8

# Example of club's statistics data

1	chelsea											
	aerial_battles	big_chance_created	clearance	club_name	cross	cross_accuracy	goal_conceded_per_match	goal_per_match	interceptions	pass_accuracy	pass_per_game	shooting_acc
0	2,682	48	839	Chelsea	688	19%	0.87	2.24	510	84%	529.61	
1	3,075	66	1,027	Chelsea	682	22%	0.84	1.92	376	83%	533.37	
2	3,055	50	1,141	Chelsea	809	25%	0.71	1.87	380	83%	480.68	
3	2,469	56	981	Chelsea	863	20%	1.03	1.97	504	83%	484.87	
4	2,551	86	762	Chelsea	995	25%	0.87	1.82	616	84%	506.21	

# Reference on club's statistics data

## Soccer Jargons

**Aerial Battle:** The number of winning the balls in the air.

**Big Chance Created:** The number of chances that are directly related to score.

**Clearance:** The number of clearances in the defensive situations.

**Cross:** The number of crosses that are executed.

**Cross Accuracy:** The accuracy of cross that delivers the ball to the own team.

**Goal Conceded Per Match:** Average conceding goals per match.

**Goal Per Match:** Average scoring goals per match.

**Interception:** The number of intercepts.

**Pass Accuracy:** The accuracy of pass that delivers the ball to the own team.

**Pass Per Game:** The number of passes that are executed.

**Shooting Accuracy:** The accuracy of shooting that shots on goal.

**Shot On Target:** The number of shootings on goal.

**Tackle Success:** The accuracy of takle that successfully steals the ball from opponents.



# Exploring dataset - Data Wrangling

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 160 entries, 0 to 159  
Data columns (total 8 columns):  
club_name      160 non-null object  
drawn          160 non-null int64  
goal           160 non-null int64  
goal_against   160 non-null int64  
lost           160 non-null int64  
points         160 non-null int64  
position       160 non-null int64  
won            160 non-null int64  
dtypes: int64(7), object(1)  
memory usage: 10.1+ KB
```

## Table's data types

- 'goal' and 'goal\_against' need to be converted to integer data type
- 'position' need to be converted to categorical data type

- Since the purpose of this analysis is which factors should a team to focus on to getting into top 4 on the table, 'win', 'drawn', 'lost', and 'point' will not be included in this analysis.

# Exploring dataset - Data Wrangling

	aerial_battles	big_chance_created	clearance	club_name	cross	cross_accuracy	goal_conceded_per_match	goal_per_match
0	2,682	48	839	Chelsea	688	19%	0.87	2.24
1	3,075	66	1,027	Chelsea	682	22%	0.84	1.92
2	3,055	50	1,141	Chelsea	809	25%	0.71	1.87
3	2,469	56	981	Chelsea	863	20%	1.03	1.97
4	2,551	86	762	Chelsea	995	25%	0.87	1.82

interceptions	pass_accuracy	pass_per_game	shooting_accuracy	shot_on_target	tackle_success
510	84%	529.61	35%	204	71%
376	83%	533.37	37%	210	81%
380	83%	480.68	33%	229	77%
504	83%	484.87	34%	212	76%
616	84%	506.21	33%	244	74%

- Unnecessary comma and percentage sign need to be removed before convert the data type to integer

# Exploring dataset - Data Wrangling

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 14 columns):
aerial_battles          5 non-null object
big_chance_created      5 non-null int64
clearance              5 non-null object
club_name              5 non-null object
cross                  5 non-null int64
cross_accuracy          5 non-null object
goal_conceded_per_match 5 non-null float64
goal_per_match          5 non-null float64
interceptions          5 non-null int64
pass_accuracy          5 non-null object
pass_per_game          5 non-null float64
shooting_accuracy       5 non-null object
shot_on_target         5 non-null int64
tackle_success         5 non-null object
dtypes: float64(3), int64(4), object(7)
memory usage: 640.0+ bytes
```

## Club's statistics data types

- Every features need to be converted to integer



# Exploring dataset - Data Wrangling

- The values need to be in average to compare in ‘per match’ unit.

## Example of outputs

club_name	goal_per_match	goal_conceded_per_match	shooting_accuracy	shot_on_target	pass_accuracy	pass_per_game	cross	cross_accuracy	interceptions	aerial_battles	big_chance_created	clearance	tackle_success
Chelsea	2.24	0.87	0.35	5.368421	0.84	529.61	18.105263	0.19	13.421053	70.578947	1.263158	22.078947	0.71
Chelsea	1.92	0.84	0.37	5.526316	0.83	533.37	17.947368	0.22	9.894737	80.921053	1.736842	27.026316	0.81
Chelsea	1.87	0.71	0.33	6.026316	0.83	480.68	21.289474	0.25	10.000000	80.394737	1.315789	30.026316	0.77
Chelsea	1.97	1.03	0.34	5.578947	0.83	484.87	22.710526	0.20	13.263158	64.973684	1.473684	25.815789	0.76
Chelsea	1.82	0.87	0.33	6.421053	0.84	506.21	26.184211	0.25	16.210526	67.131579	2.263158	20.052632	0.74

# Exploring dataset - Data Wrangling

- Combine 'table' data frame with 'club statistics' data frame for further analysis.

## Example of outputs

club_name	goal_per_match	goal_conceded_per_match	shooting_accuracy	shot_on_target	pass_accuracy	pass_per_game	cross	cross_accuracy	interceptions	...	big_chance_created	clearance	tackle_success	drawn	goal
Chelsea	2.24	0.87	0.35	5.368421	0.84	529.61	18.105263	0.19	13.421053	...	1.263158	22.078947	0.71	3	85
Chelsea	1.92	0.84	0.37	5.526316	0.83	533.37	17.947368	0.22	9.894737	...	1.736842	27.026316	0.81	9	73
Chelsea	1.87	0.71	0.33	6.026316	0.83	480.68	21.289474	0.25	10.000000	...	1.315789	30.026316	0.77	7	71
Chelsea	1.97	1.03	0.34	5.578947	0.83	484.87	22.710526	0.20	13.263158	...	1.473684	25.815789	0.76	9	75
Chelsea	1.82	0.87	0.33	6.421053	0.84	506.21	26.184211	0.25	16.210526	...	2.263158	20.052632	0.74	8	69

Continues

goal_against	lost	points	position	won
33	5	93	1	30
32	3	87	1	26
27	6	82	3	25
39	7	75	3	22
33	9	71	2	21



# Exploring dataset - Data Wrangling

According to 'table' data,  
here are the list of teams and the years who were on Top 4 in past 8 years

**Manchester City:** 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010

**Manchester United:** 2017, 2014, 2012, 2011, 2010

**Arsenal:** 2015, 2014, 2013, 2012, 2011, 2010

**Chelsea:** 2016, 2014, 2013, 2012, 2010

**Liverpool:** 2017, 2016, 2013

**Tottenham Hotspur:** 2017, 2016, 2015, 2011

**Leicester City:** 2015

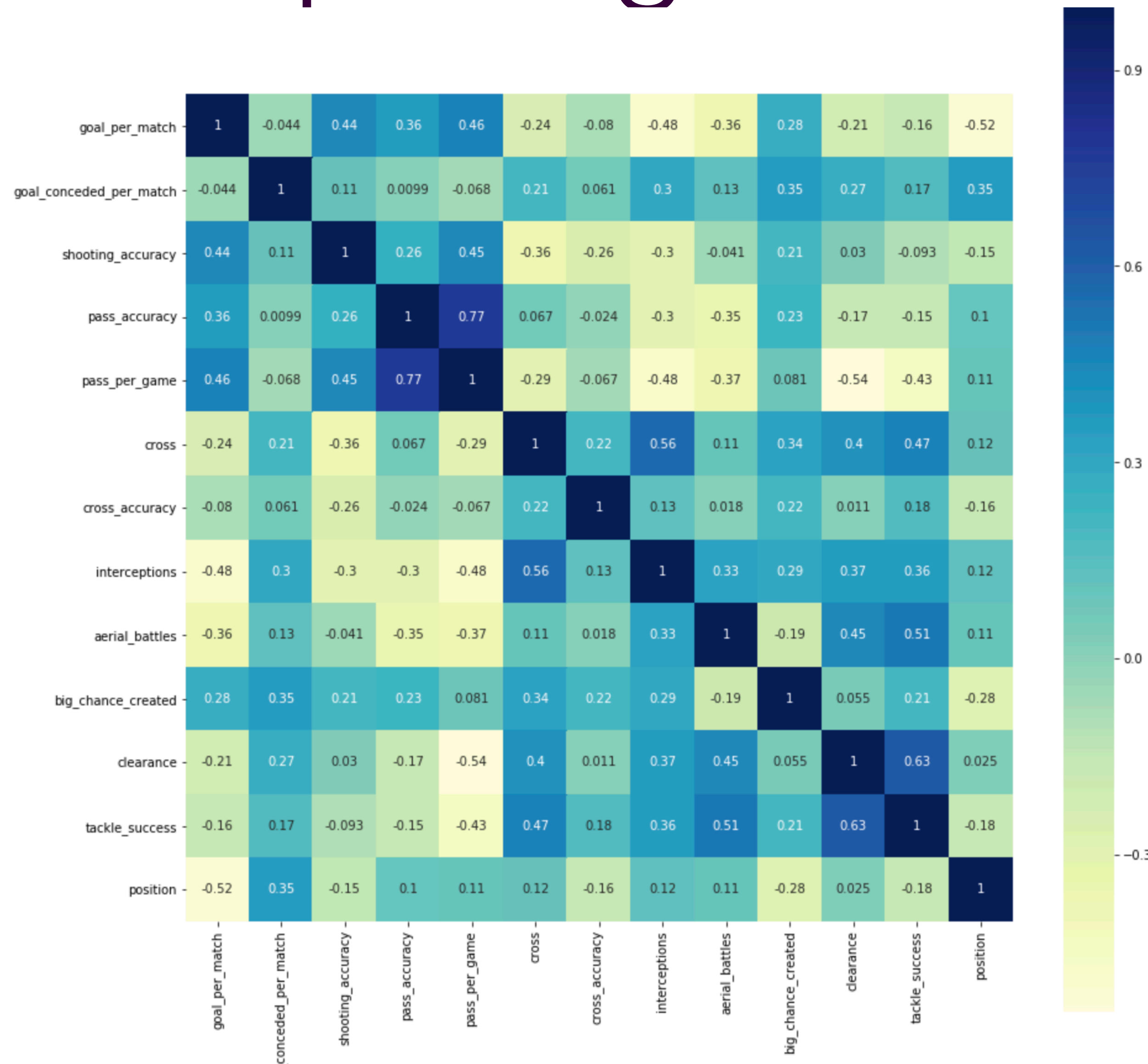
These clubs' data will be used for further analysis since my interest is in predicting Top 4

# Exploring dataset - Data Wrangling

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 13 columns):
goal_per_match          32 non-null float64
goal_conceded_per_match 32 non-null float64
shooting_accuracy       32 non-null float64
pass_accuracy           32 non-null float64
pass_per_game           32 non-null float64
cross                   32 non-null float64
cross_accuracy          32 non-null float64
interceptions           32 non-null float64
aerial_battles           32 non-null float64
big_chance_created      32 non-null float64
clearance               32 non-null float64
tackle_success          32 non-null float64
position                32 non-null object
dtypes: float64(12), object(1)
memory usage: 3.3+ KB
```

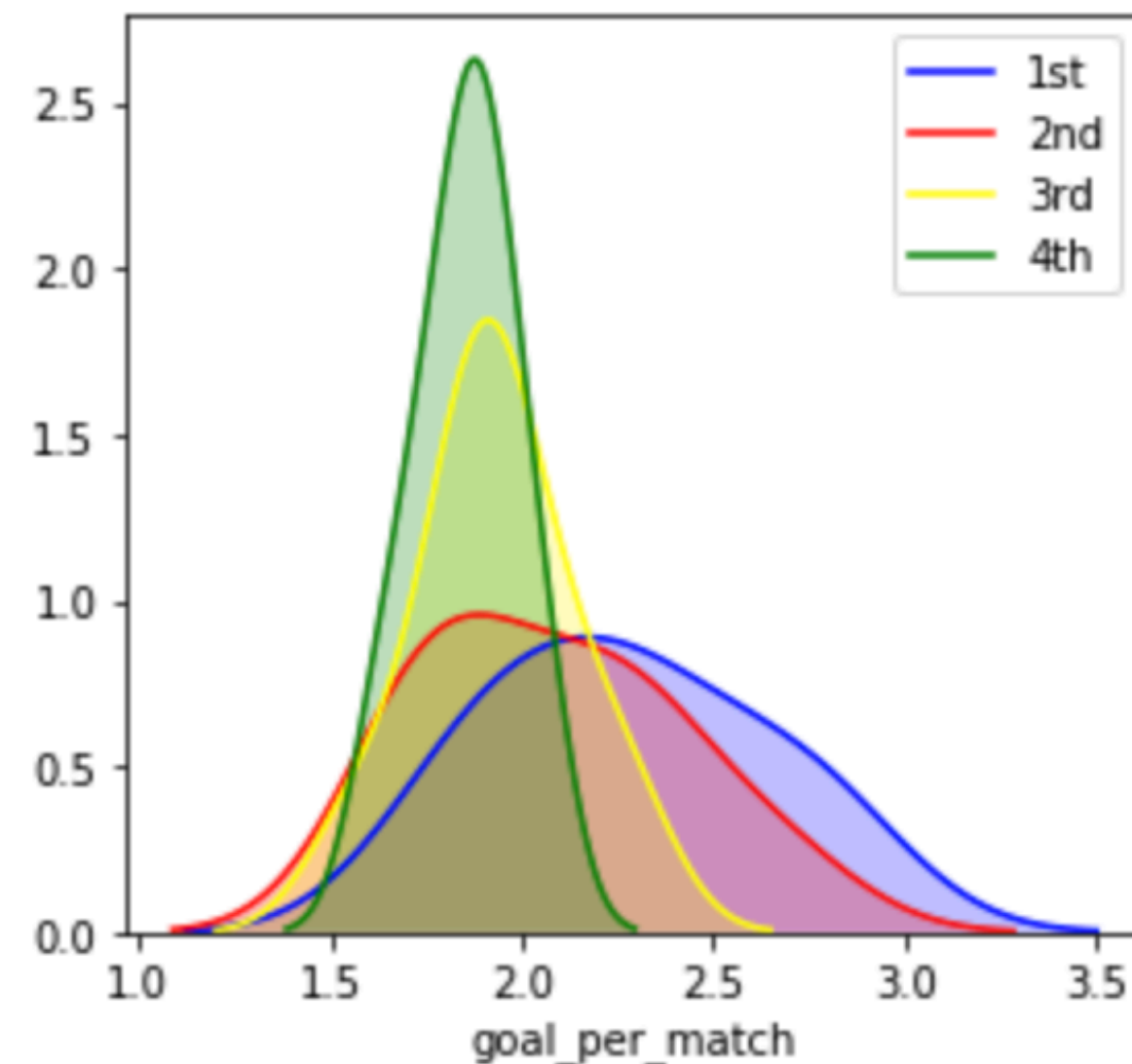
- Combining the data that need in this analysis, result in getting 32 rows of dataset which is very low volume of data. Therefore, overfitting is most likely to occur when performing machine learning models.

# Exploring dataset - EDA

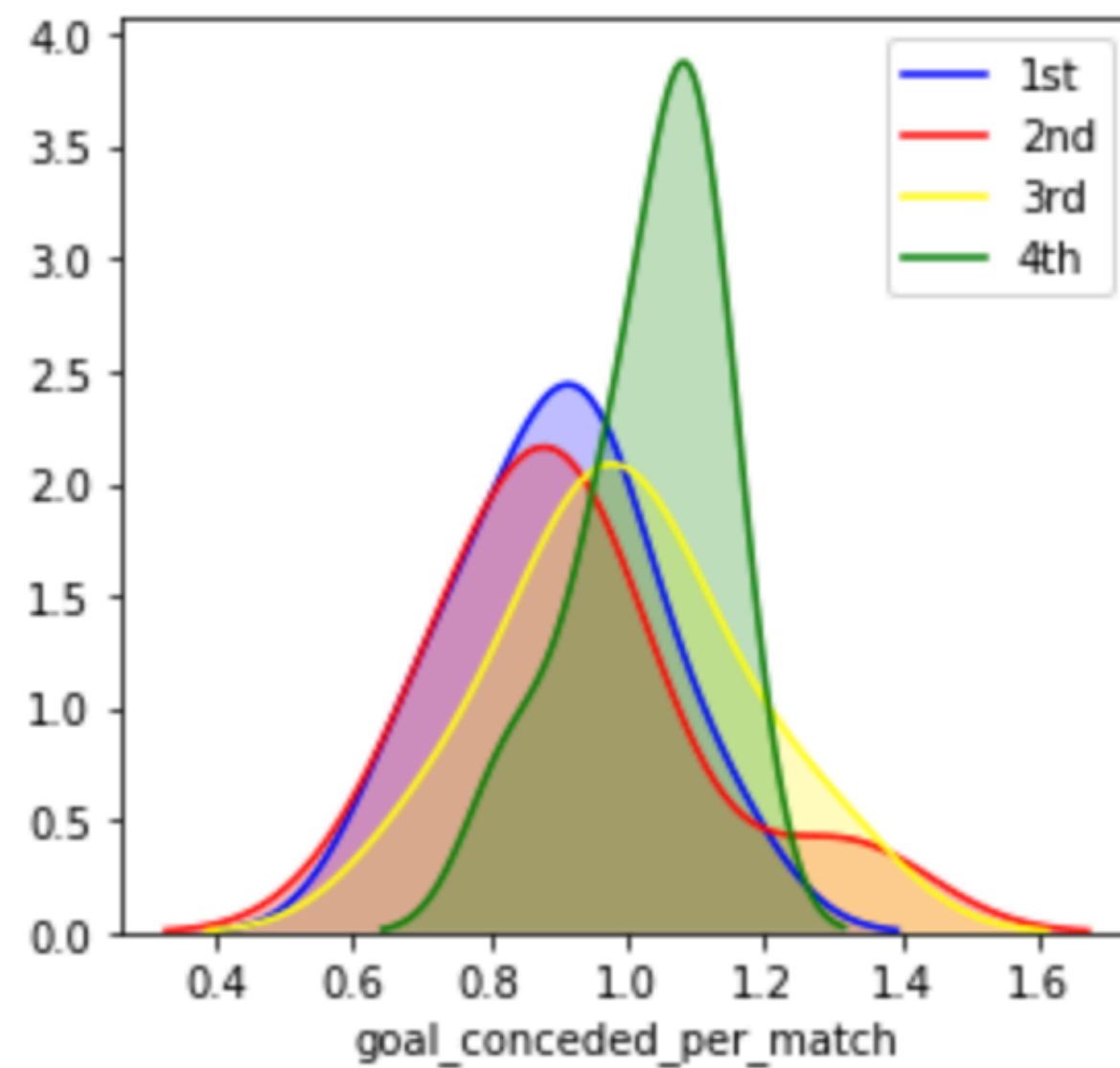


- Insight from heatmap: it looks like '**goal\_per\_match**', '**goal\_per\_conceded\_per\_match**', and '**big\_chance\_created**' has a correlation with position.

# Exploring dataset - EDA



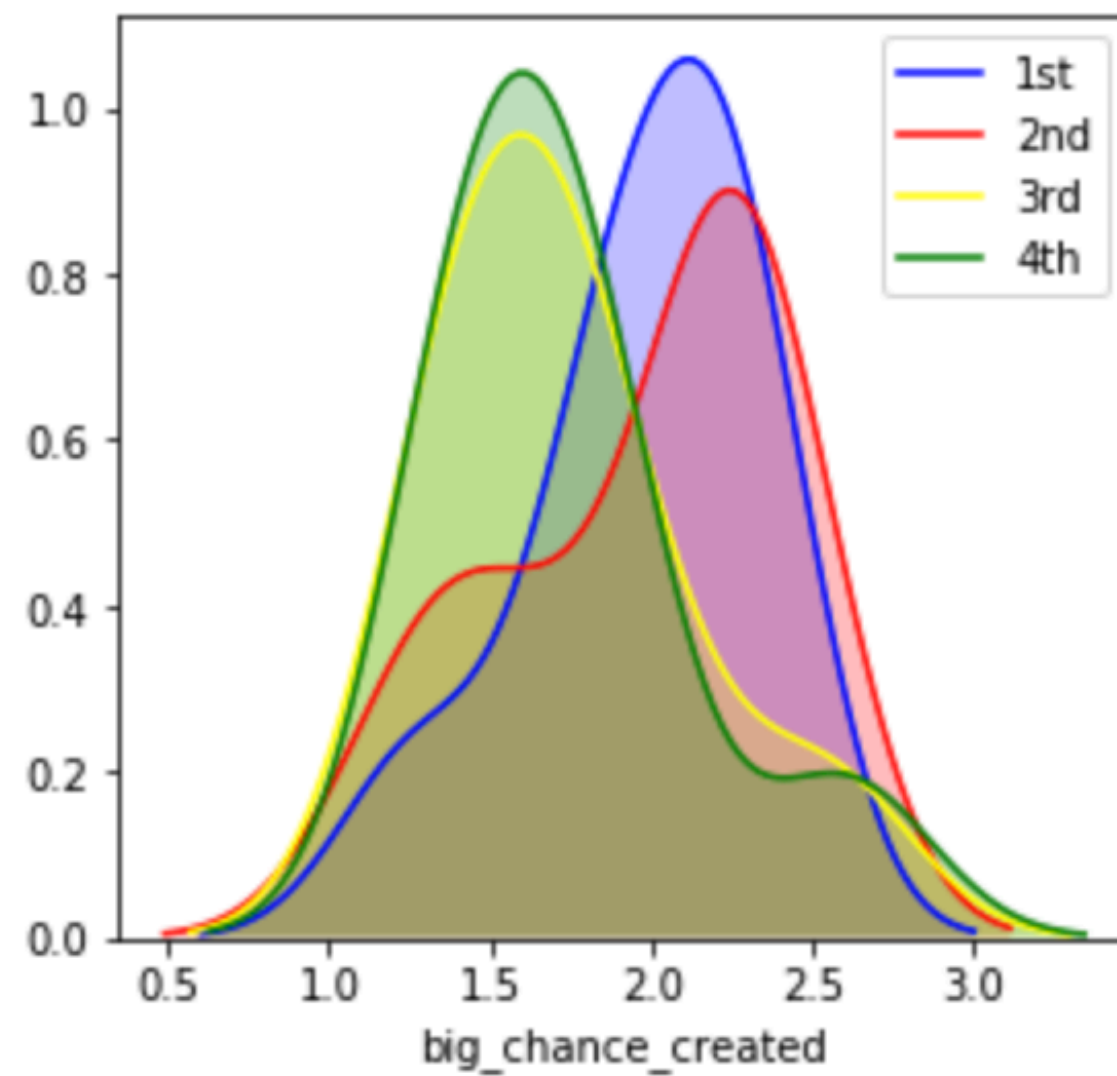
- Goal per match: 1st and 2nd has dispersive range of goal per match, but 3rd and 4th has a lot of goals between 1.5 to 2.5.



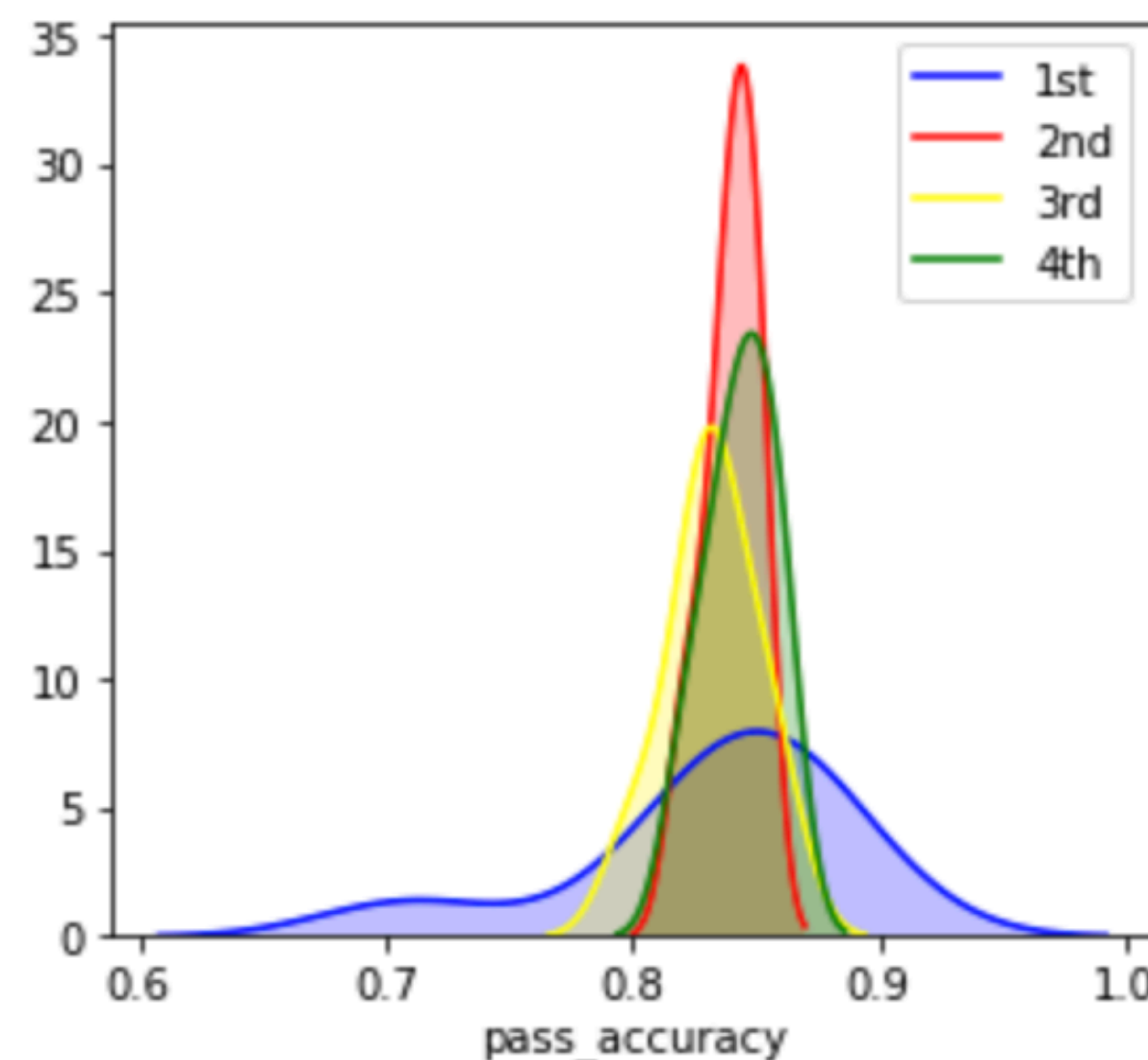
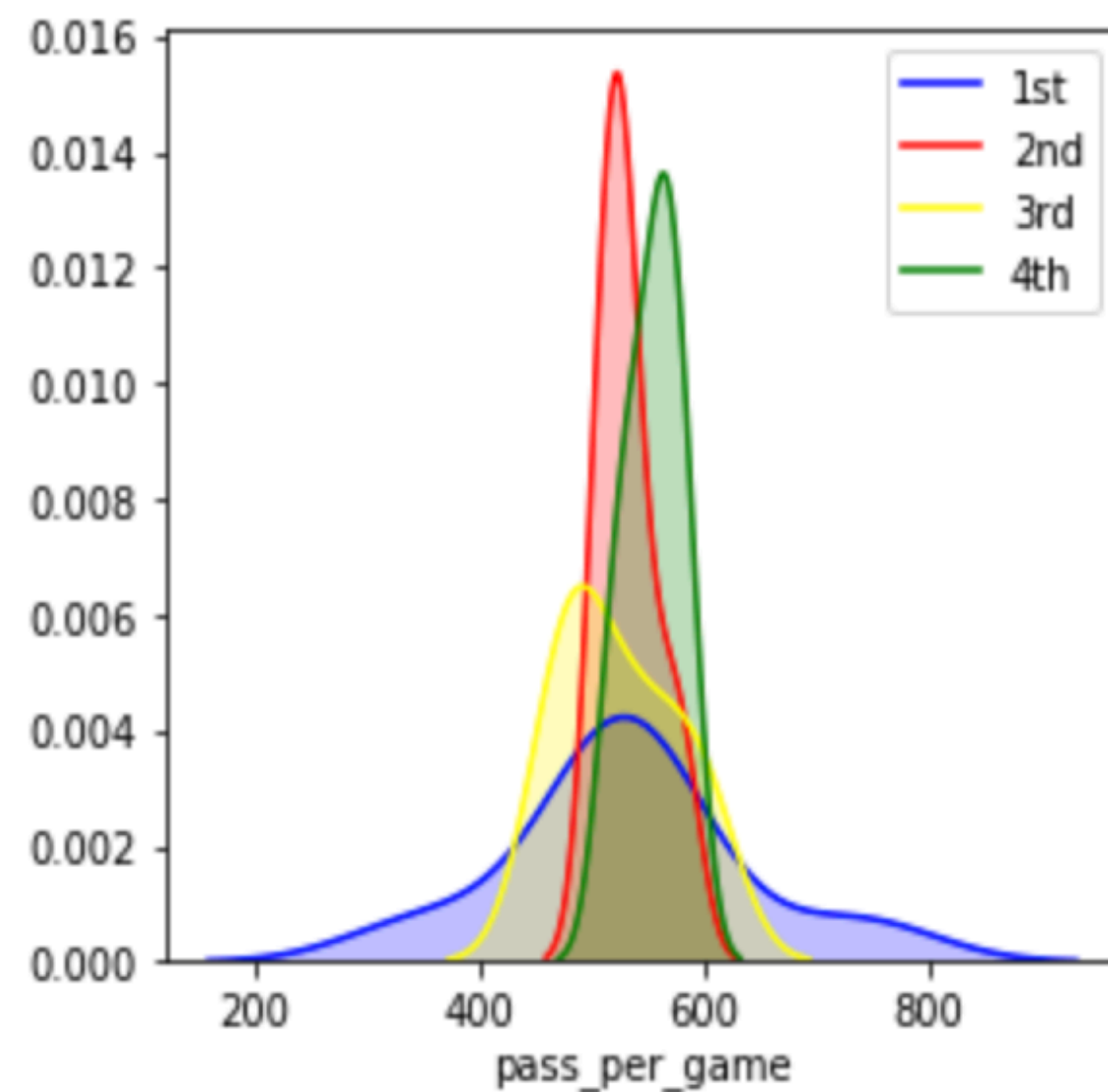
- Goal conceded per match: 4th teams has much more goal conceded than other ranks



# Exploring dataset - EDA



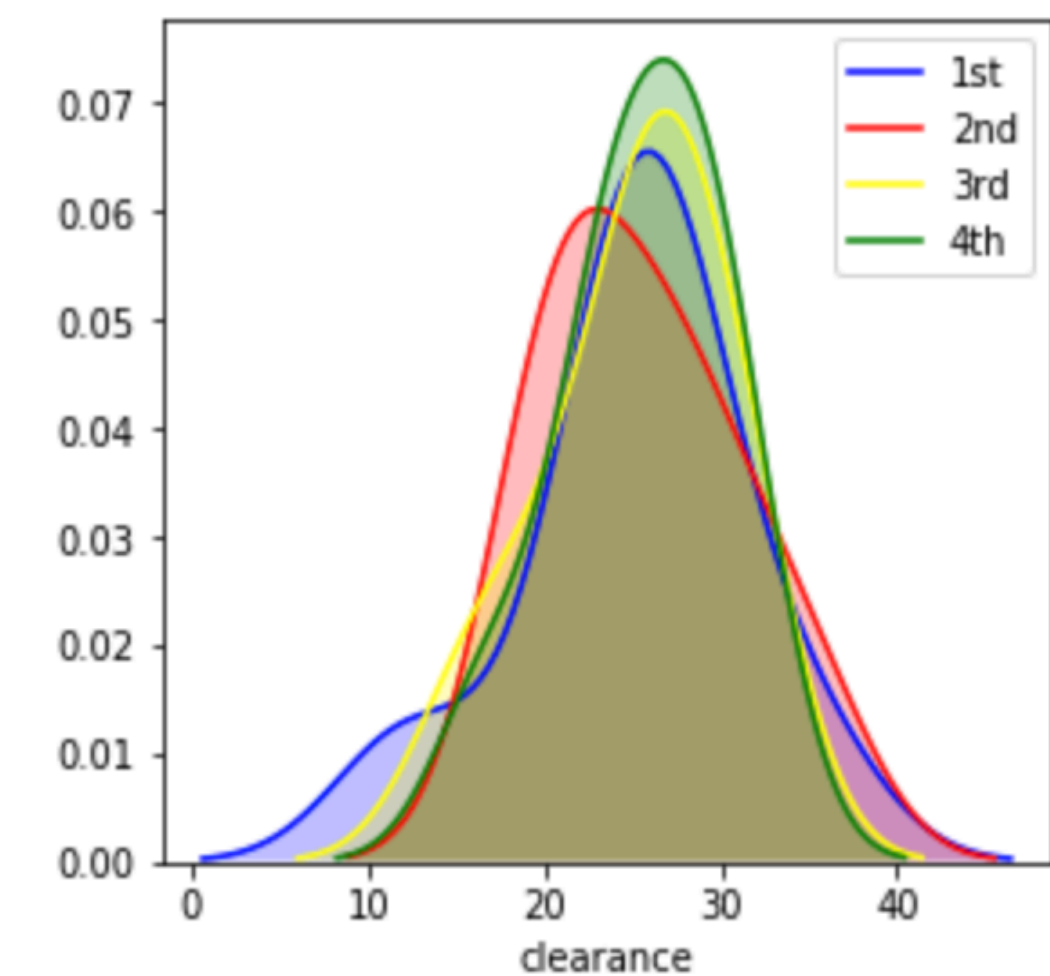
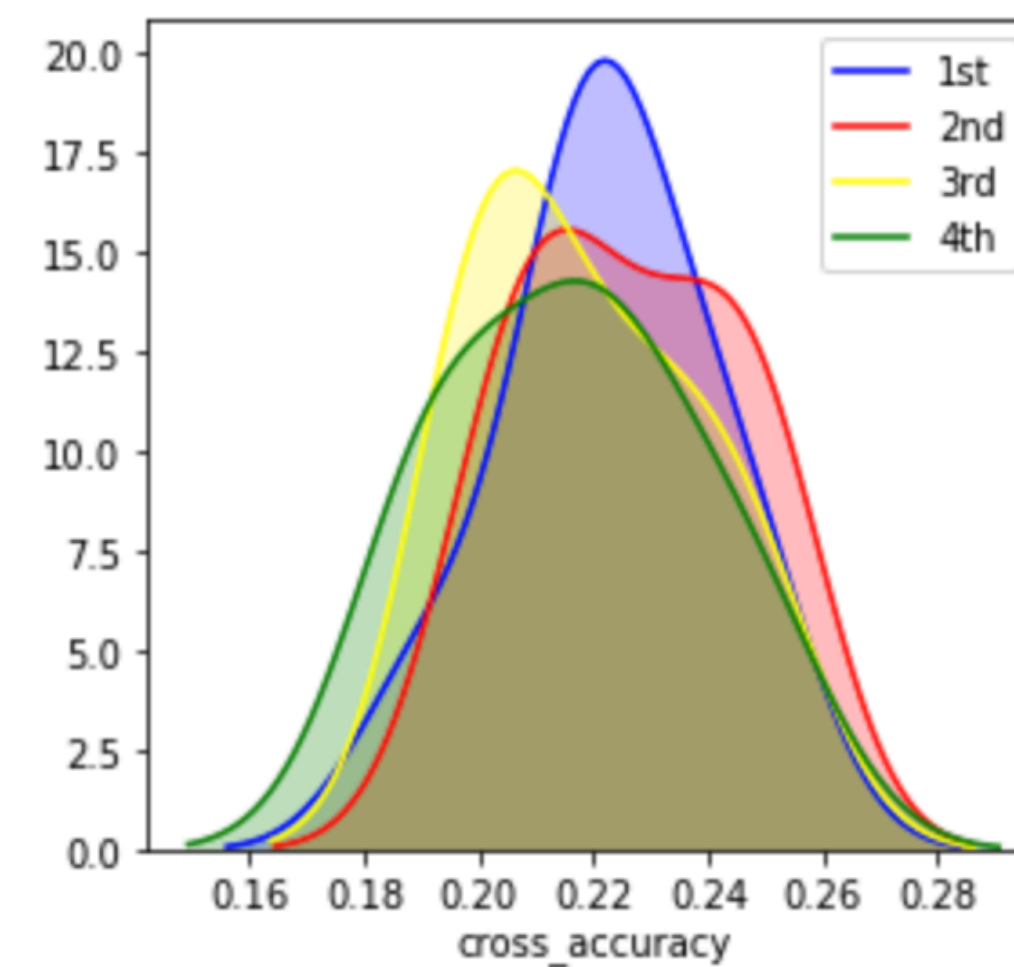
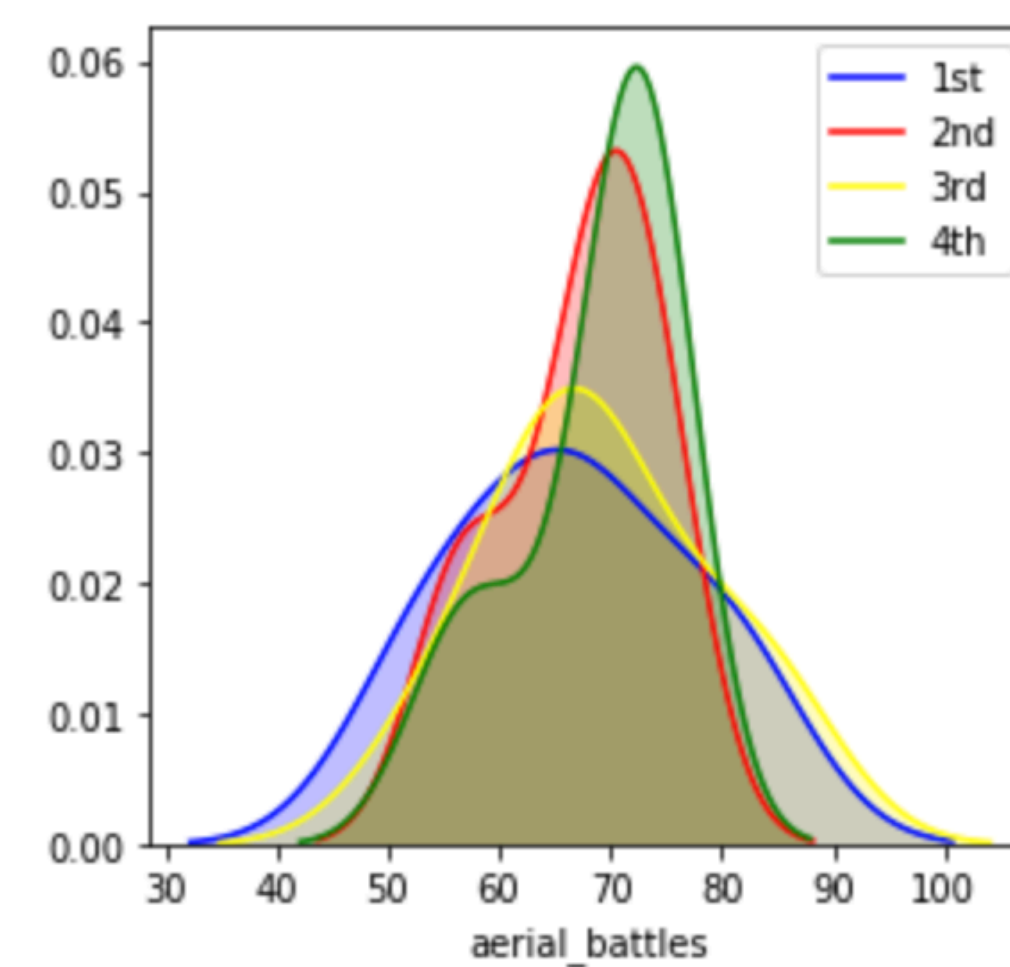
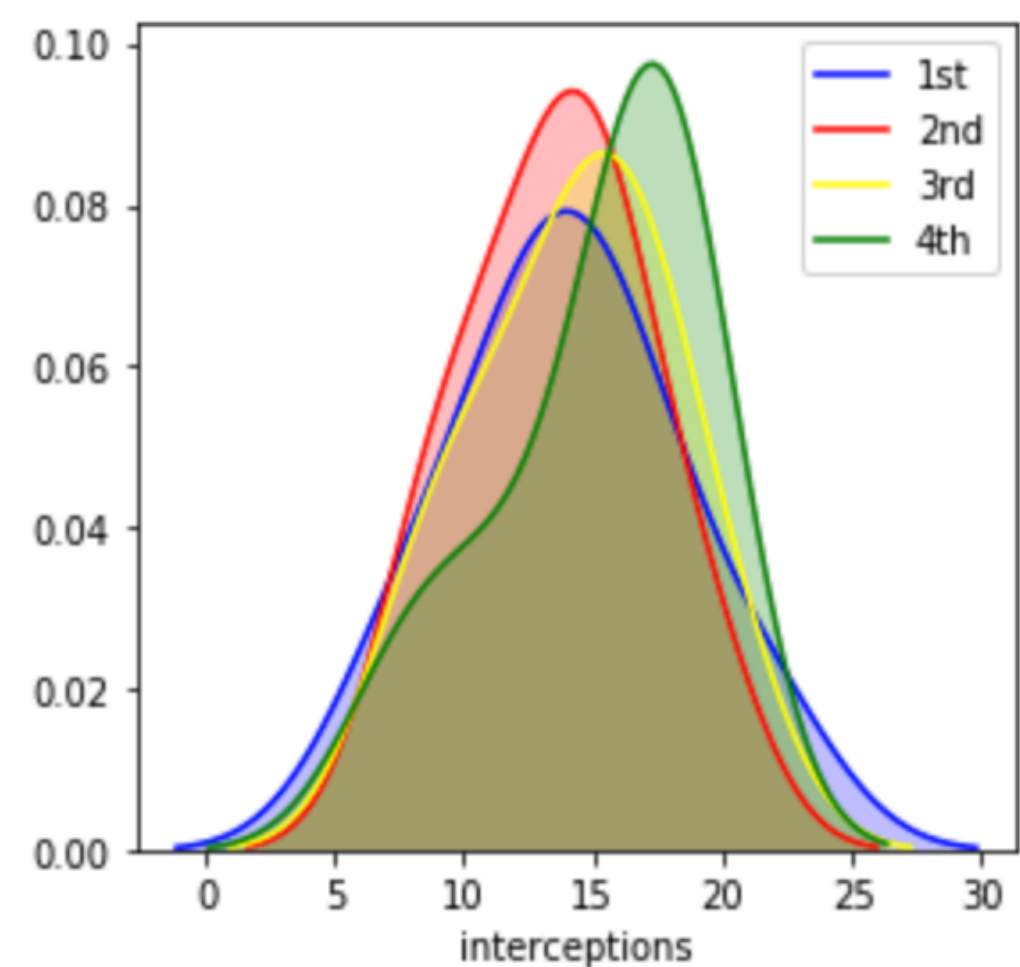
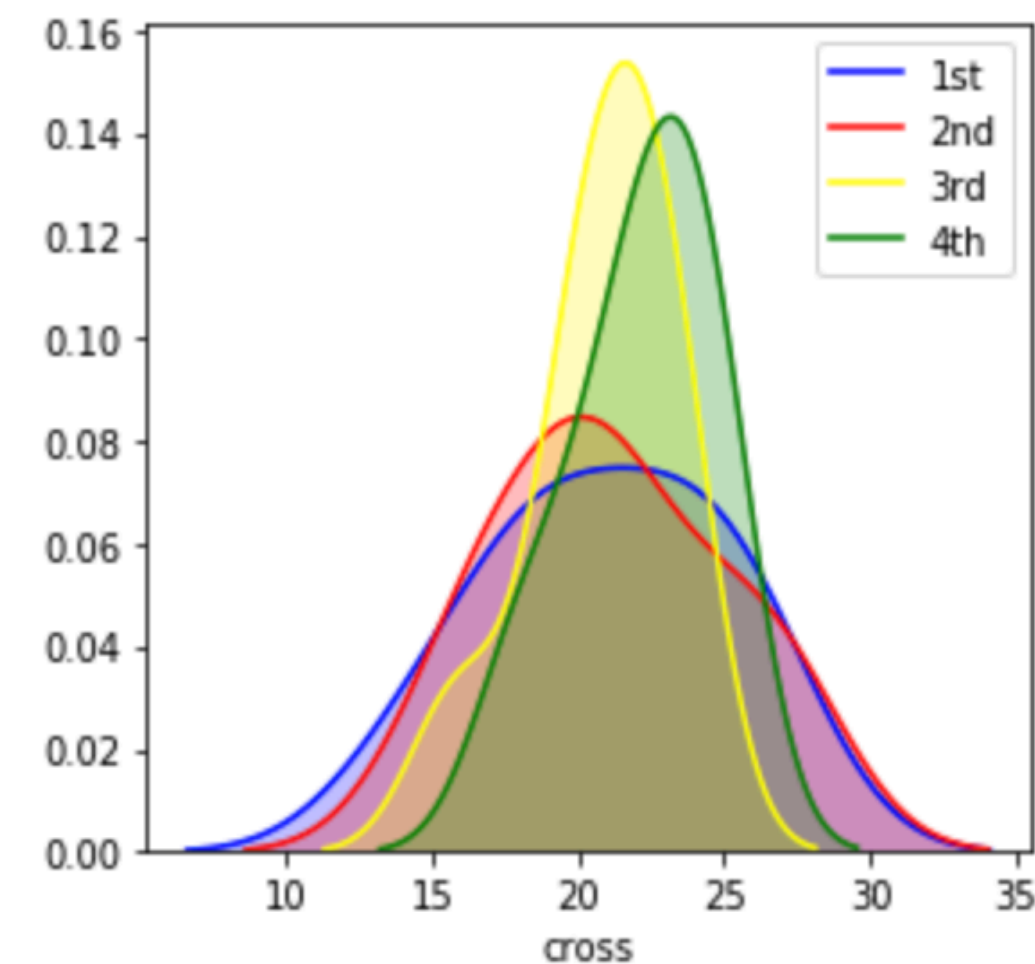
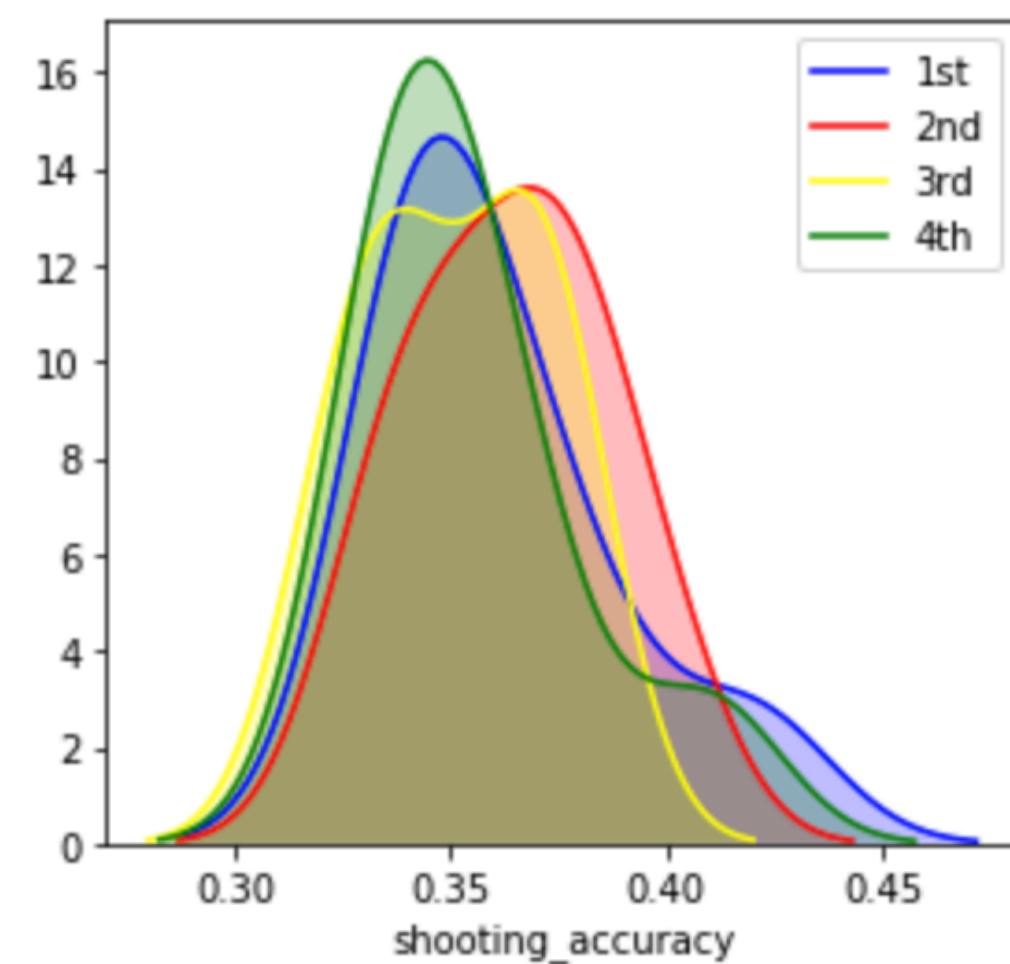
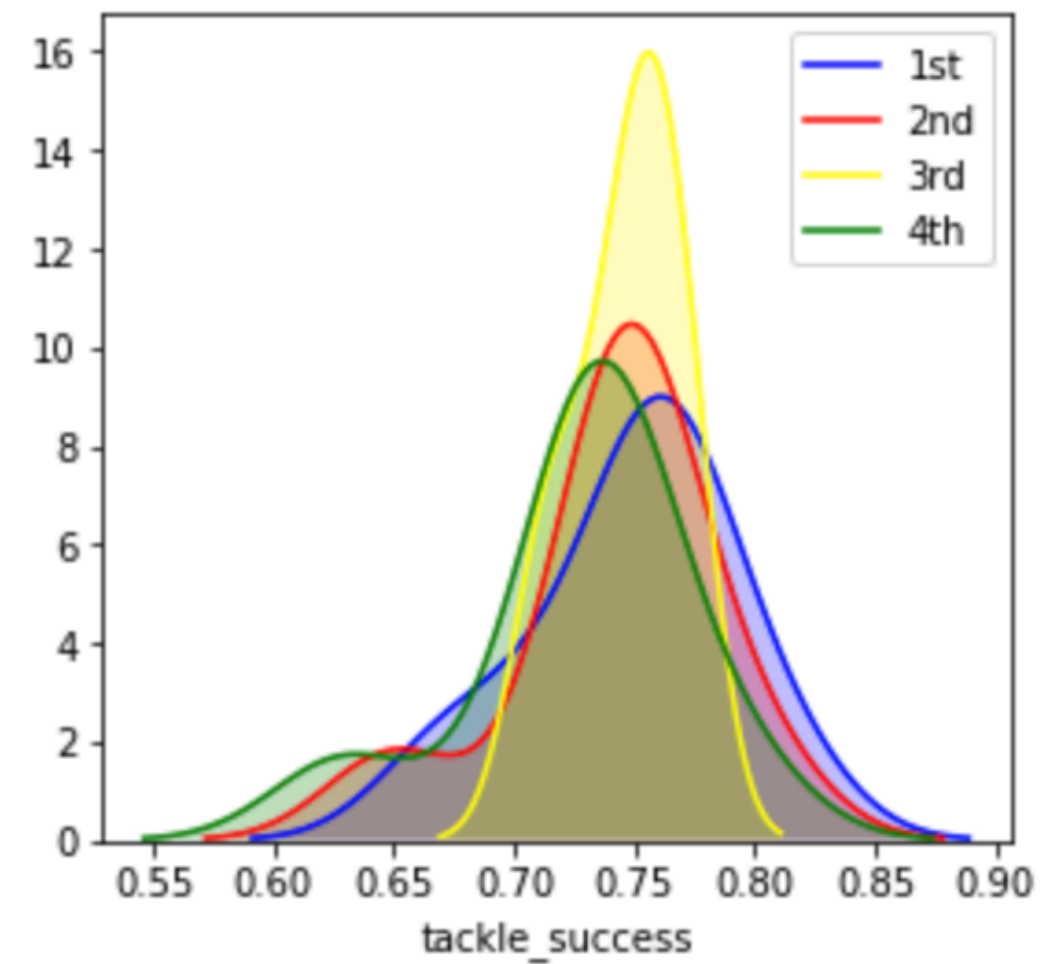
- Big chance created: There is clear signal that more big chance created is better for team's rank.



- Pass per game and Pass accuracy: 1st teams usually have dispersive range of the features than other ranks



# Exploring dataset - EDA



- I do not see much information on these plots.

# Exploring dataset - Selecting features

- Let's see if my visual analysis is relevantly correct by using backward elimination and feature selection model from scikit-learn(KBest)

## Feature selection based on Logistics Regression model

```
[ True  True False  True False False False False  True False  True]
[1  1  4  1  8  2  5  3  7  1  6  1]
Index(['goal_per_match', 'goal_conceded_per_match', 'shooting_accuracy',
      'pass_accuracy', 'pass_per_game', 'cross', 'cross_accuracy',
      'interceptions', 'aerial_battles', 'big_chance_created', 'clearance',
      'tackle_success'],
      dtype='object')
```

**Top 5 features: goal\_per\_match, goal\_conceded\_per\_match, pass\_accuracy, big\_chance\_created, tackle\_success**

# Exploring dataset - Selecting features

## Feature selection based on Xgboost model

```
[ True  True False  True  True False False False False  True False False]
[1 1 6 1 1 7 4 8 5 1 3 2]
Index(['goal_per_match', 'goal_conceded_per_match', 'shooting_accuracy',
      'pass_accuracy', 'pass_per_game', 'cross', 'cross_accuracy',
      'interceptions', 'aerial_battles', 'big_chance_created', 'clearance',
      'tackle_success'],
      dtype='object')
```

**Top 5 features: goal\_per\_match, goal\_conceded\_per\_match, pass\_accuracy, pass\_per\_game, big\_chance\_created**



# Exploring dataset - Selecting features

## Feature selection based on Random Forest model

```
[ True  True False False  True False False False  True  True False False]
[1 1 8 6 1 4 2 7 1 1 5 3]
Index(['goal_per_match', 'goal_conceded_per_match', 'shooting_accuracy',
      'pass_accuracy', 'pass_per_game', 'cross', 'cross_accuracy',
      'interceptions', 'aerial_battles', 'big_chance_created', 'clearance',
      'tackle_success'],
      dtype='object')
```

**Top 5 features: goal\_per\_match, goal\_conceded\_per\_match, pass\_per\_game, aerial\_battle, big\_chance\_created**

# Exploring dataset - Selecting features

## Feature selection based on SVM and KNN

- SVM and KNN does not provide logic to rank the feature; therefore, we cannot implement this method. For SVM and KNN, feature will be selected according to EDA

**Top 5 features: goal\_per\_match, goal\_conceded\_per\_match, pass\_accuracy  
pass\_per\_game, big\_chance\_created**



# Test ML models

- Testing few models to see which model works the best on this problem
    - However, according to analysis from previous, the dataset is too small; therefore, overfitting will likely to occur on every models
1. Multinomial Logistic Regression
  2. Support Vector Machine
  3. XG boost
  4. Knn
  5. Random Forest

# Test ML models

## Training data output

```
Logistic Regression: 0.42857142857142855  
SVM: 0.42857142857142855  
Xgboost: 1.0  
KNN: 0.47619047619047616  
Random Forest: 1.0
```

## Test data output

```
Logistic Regression: 0.18181818181818182  
SVM: 0.2727272727272727  
Xgboost: 0.36363636363636365  
KNN: 0.45454545454545453  
Random Forest: 0.18181818181818182
```

- The outputs clearly show that models are overfitting except KNN. KNN is also highly likely to overfit the data, but it might just got lucky or it may work on this analysis.

# Predict the Top 4 rank

- Getting a half of 2018 data by the same web scraping method and conduct the same technique to clean the data to perform a prediction.
- Perform predictions with fitted models in a previous step.

## Prediction Outcome

	club_name	position	logist	svm	xgboost	knn	random_forest
0	Manchester City	0	2	2	2	2	2
1	Manchester United	1	2	2	3	2	3
2	Liverpool	3	2	2	3	2	3
3	Tottenham Hotspur	2	1	2	1	2	1
4	Chelsea	4	2	2	3	2	3
7	Leicester City	8	2	2	2	2	2

- Based on the results, it is clear that every model is overfitting and giving a poor prediction (so KNN got lucky on training and test data set to validate overfitting).

# Conclusion

- **Finalizing the analysis**

There are clearly some differences in features among rank 1 to 4.

Higher rank teams tend to have:

- a. Higher goal\_per\_match and less goal\_conceded\_per\_match, which means a team that strongly focus on attacking or defending tactics is less chance to get into higher rank on the table.
- b. Higher big\_chance\_created, pass\_accuracy and pass\_per\_game. This could mean that higher rank teams generally good at passing and execute more passes throughout a game, and this mean it could lead a team to have more chance to score goals.

- **Conclusion**

As the analysis above, there are features that favor higher rank teams. According the analysis, training on passing tactics and skills will help a team the most out of all features. Therefore, if a team wants to achieve higher rank on the table, working on passing tactics and skills are recommended. On the other hand, predicting the Top 4 rank is not applicable with the low volume dataset. Every machine learning models are overfitting and giving a poor outcome.

# Limitation

- **Limitation**

- a. The volume of dataset was too low to fit machine learning models.
- b. There was not enough resource to get the data like the ones I web scraped.
- c. Mentality is a huge part of sports, but it is hard to quantify it and could not find any resource to combine with the dataset.

- **For future analysis**

- a. Find a way to replace or get more data to have a large volume of dataset.
- b. Apply Bayesian Inference on this analysis. Since it was hard to predict the actual ranks, Bayesian Inference calculates how much a team more likely to be a certain rank.