European Football Data Analysis

Pham Trung Hieu

1 Overview

1.1 Dataset

This task uses the soccer dataset on kaggle. All the details can be found on this link: european-football-database. I provide a litte description here. The dataset is about match's result collection from some european leagues in the range from season 2005/2006 to 2020/2021. There're 2 tables on the .sqlite database. divisions table stores the information of the leagues:

division	name	country
B1	Division 1A	Belgium
D1	Bundesliga	Deutschland
D2	2. Bundesliga	Deutschland
E0	Premier League	England
E1	EFL Championship	England
E2	EFL League One	England
E3	EFL League Two	England
EC	National League	England
F1	Ligue 1	France
F2	Ligue 2	France
G1	Superleague	Greece
I1	Seria A	Italy
I2	Seria B	Italy
N1	Eredivisie	Netherlands
P1	Liga NOS	Portugal
SC0	Scottish Premiership	Scotland
SC1	Scottish Championship	Scotland
SC2	Scottish League One	Scotland
SP1	LaLiga	Spain
SP2	LaLiga 2	Spain
T1	Süper Lig	Turkey

Figure 1: division table's content.

matchs table stores all the games in the given time range with some statistics:

Div	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	season
B1	2020-08-08	Club Brugge	Charleroi	0	1	A	2021
B1	2020-08-08	Antwerp	Mouscron	1	1	D	2021
B1	2020-08-08	Standard	Cercle Brugge	1	0	H	2021
B1	2020-08-09	St Truiden	Gent	2	1	H	2021
B1	2020-08-09	Waregem	Genk	1	2	A	2021
B1	2020-08-09	Mechelen	Anderlecht	2	2	D	2021
B1	2020-08-09	Kortrijk	Waasland-Beveren	1	3	A	2021
B1	2020-08-10	Oud-Heverlee Leuven	Eupen	1	1	D	2021
B1	2020-08-10	Oostende	Beerschot VA	1	2	A	2021
B1	2020-08-14	Mouscron	Mechelen	0	1	A	2021
B1	2020-08-15	Genk	Oud-Heverlee Leuven	1	1	D	2021
B1	2020-08-15	Charleroi	Oostende	1	0	H	2021
B1	2020-08-15	Gent	Kortrijk	1	2	A	2021
B1	2020-08-16	Cercle Brugge	Antwerp	2	1	H	2021
B1	2020-08-16	Beerschot VA	Waregem	3	1	H	2021
B1	2020-08-16	Eupen	Club Brugge	0	4	A	2021
B1	2020-08-16	Anderlecht	St Truiden	3	1	H	2021
B1	2020-08-17	Waasland-Beveren	Standard	1	2	A	2021
B1	2020-08-21	Kortrijk	Eupen	0	0	D	2021
B1	2020-08-22	Waregem	Waasland-Beveren	4	1	H	2021
B1	2020-08-22	Mechelen	Cercle Brugge	2	3	A	2021
B1	2020-08-22	Oud-Heverlee Leuven	Charleroi	1	3	A	2021

Figure 2: Top rows of matchs table.

1.2 Brief content

Here I recap the works below. Data gathering is straightforward. At first, there're nothing to do for data cleaning, no duplicated, no null value, no unused record. I use a subset only including the chosen leagues. There's a little problem getting in the way when a small table is created. It's needed an easy cleaning step to tackle. Some functions are defined for special query purposes related to dataset. It can be seasonal result or pairwise statistics of 2 given teams,... After that, analysing steps begin with the explorations based on not original cleaned dataset, but retrieved tables that are outputs of the pre-defined functions. Ad-hoc metrics arise and are considered for building the more interesting statistics. They, in turn, are used for answering a few questions about data. To sum up, the reporting stage is conducted using Power BI's visualization tools.

1.3 Code organization

I divide the task into files and store all of them in the repo. Data storage files consist of original .sqlite file gathered directly from kaggle dataset, and some .csv files that are exported from and imported to other coding files. A .pbix file serves the reporting phase will be uploaded soon. All .ipynb files is compiled in Google Colab and need little changes for available in Jupiter, Kaggle Notebook,...

2 Data cleaning

I just comment from now on for explaining the messy lines of code. This section and the next section are coded in the first file of the work. `european_soccer_processing.ipynb` casts .sqlite file as data source. Perhaps due to the way dataset is collected, the format is very clear and clean as we can see from the two above figures.

2.1 Data checking

We use only subset of 6 most common leagues in this work, include EPL, LaLiga, Serie A, Bundesliga, Ligue 1 and Eredivise. The correctness of score columns can be compared to real information from other public sources, and I don't dive into it here. It means we assume these values are true by default. About other columns, an error, wrong date value, wrong team's name, wrong division's name, may occurs by many reasons. I built one simple way to validate them all: a league's information table. Moreover, the use of the table is not only for checking original data, but also for aiding many functions afterward.

2.2 The information table

The table comprise information of the number of matches and the number of teams in each division and each season. There're at least 2 ways to return season's cardinality. The first one is to calculate from the number of matches. Given n teams in a season, there're $\frac{n}{2}$ matches per round (n must be even). Each team play against all the other teams two time in season, one in home round and one in away round, so there're $2(n-1)$ rounds per season. We have the formula for m matches in the season

$$m = 2(n-1).\frac{n}{2} = n(n-1). \tag{1}$$

Hence $n = \lceil \sqrt{m} \rceil$.

```
dup = matches[['Div', 'season']].duplicated()
div_season = matches[['Div', 'season']][~dup]
for d in divisions['division']:
    print('{:<5}'.format(d), end=' | ')
    seasons = set(div_season.query('Div == @d')['season'])
    print(seasons) # all leagues have been recorded in the given period from 2006 to 2021 season
```

```
D1    | {2016, 2017, 2018, 2019, 2020, 2021, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015}
E0    | {2016, 2017, 2018, 2019, 2020, 2021, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015}
F1    | {2016, 2017, 2018, 2019, 2020, 2021, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015}
I1    | {2016, 2017, 2018, 2019, 2020, 2021, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015}
N1    | {2016, 2017, 2018, 2019, 2020, 2021, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015}
SP1   | {2016, 2017, 2018, 2019, 2020, 2021, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015}
```

Figure 3: All divisions have been recorded in the same periods.

The frame has 96 rows, equal to 6 divisions through 16 seasons. But a problem arises. We know that all divisions have 20 teams joining, except the cases of 18 of Bundesliga and Eredivise. However, the number of teams of Ligue 1 and Eredivise isn't unique.

Div	season	No of Matches	No of Teams
D1	2006	306	18.0
	2007	306	18.0
	2008	306	18.0
	2009	306	18.0
	2010	306	18.0
...	...	...	...
SP1	2017	380	20.0
	2018	380	20.0
	2019	380	20.0
	2020	380	20.0
	2021	380	20.0

96 rows × 2 columns

Figure 4: The information table.

```
div_season_match_team.groupby('Div')['No of Teams'].nunique() # abnormal detection

Div
D1     1
E0     1
F1     2
I1     1
N1     2
SP1    1
Name: No of Teams, dtype: int64
```

Figure 5: The abnormal's detection.

We need to use second simple way to retrive the number of teams using collection of all team's names in a league. Compare two theoretically right result, we have image here Only inequal row values are shown. Now, we can easily explain these unexpected numbers as follow. Due to covid quarantine in season 2020, French and Dutch division limited the total number of rounds, so the equation (1) is wrong. But the second way also return false number where 27 Dutch teams joined in one season. This's because data itself when some team's names have trailing spaces. Luckily, the names are still true. Cleaning step is needed and the information table is updated too.

```
# clean the name columns
matches['HomeTeam'] = matches['HomeTeam'].apply(lambda x: x.strip())
matches['AwayTeam'] = matches['AwayTeam'].apply(lambda x: x.strip())
# correcting stats table
div_season_match_team.loc['F1'].loc[2020, 'No of Teams'] = 20.0
div_season_match_team.loc['N1'].loc[2020, 'No of Teams'] = 18.0
```

The table is now correct and our data is ready to be used to some extent.

3 Data manipulating and queries

Dataset is now standardized, but not in the right format to work with. Let's consider a query, for instance, that requires to show final result of any season from users. The data can't easily produce the answer in the original form itself. So I defined some available complex results based on the raw data:

```
matches_in_round('E0', 2007, 2)
# matches_in_round('E0', 2006, 3.1)
```

	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR
108737	2006-08-22	Watford	West Ham	1.0	1.0	D
108736	2006-08-22	Tottenham	Sheffield United	2.0	0.0	H
108738	2006-08-23	Aston Villa	Reading	2.0	1.0	H
108739	2006-08-23	Blackburn	Everton	1.0	1.0	D
108740	2006-08-23	Charlton	Man United	0.0	3.0	A
108741	2006-08-23	Fulham	Bolton	1.0	1.0	D
108742	2006-08-23	Man City	Portsmouth	0.0	0.0	D
108743	2006-08-23	Middlesbrough	Chelsea	2.0	1.0	H
108750	2006-08-26	Wigan	Reading	1.0	0.0	H
108749	2006-08-26	Watford	Man United	1.0	2.0	A

Figure 6: The results of all matches in the given round.

```
league_result_after_round('E0', 2006, 3)
# league_result_after_roundd('SP1', 2007, 2)
```

	Played	Won	Drawn	Lost	GF	GA	GD	Points	Rank
Team									
Chelsea	4	4.0	0.0	0.0	8.0	0.0	8.0	12.0	1
Man City	4	3.0	1.0	0.0	6.0	3.0	3.0	10.0	2
Tottenham	4	2.0	1.0	1.0	4.0	2.0	2.0	7.0	3
Arsenal	3	2.0	0.0	1.0	6.0	2.0	4.0	6.0	4
Man United	2	2.0	0.0	0.0	3.0	0.0	3.0	6.0	5
Charlton	2	2.0	0.0	0.0	4.0	1.0	3.0	6.0	6
Aston Villa	4	1.0	2.0	1.0	4.0	4.0	0.0	5.0	7
West Ham	2	1.0	1.0	0.0	3.0	1.0	2.0	4.0	8
Middlesbrough	3	1.0	1.0	1.0	3.0	2.0	1.0	4.0	9
Liverpool	2	1.0	1.0	0.0	1.0	0.0	1.0	4.0	10
Bolton	3	1.0	1.0	1.0	4.0	3.0	1.0	4.0	11
Blackburn	4	1.0	1.0	2.0	3.0	5.0	-2.0	4.0	12
West Brom	3	1.0	1.0	1.0	2.0	5.0	-3.0	4.0	13
Everton	2	1.0	0.0	1.0	1.0	2.0	-1.0	3.0	14
Portsmouth	4	0.0	1.0	3.0	3.0	7.0	-4.0	1.0	15
Newcastle	3	0.0	1.0	2.0	0.0	4.0	-4.0	1.0	16
Fulham	3	0.0	1.0	2.0	2.0	6.0	-4.0	1.0	17
Birmingham	3	0.0	1.0	2.0	1.0	5.0	-4.0	1.0	18
Wigan	2	0.0	0.0	2.0	0.0	2.0	-2.0	0.0	19
Sunderland	3	0.0	0.0	3.0	2.0	6.0	-4.0	0.0	20

Figure 7: League's result after arbitary given timepoint.

```
league_result_season('F1', 2020)
```

Team	Played	Won	Drawn	Lost	GF	GA	GD	Points	Rank
Paris SG	27	22.0	2.0	3.0	75.0	24.0	51.0	68.0	1
Marseille	28	16.0	8.0	4.0	41.0	29.0	12.0	56.0	2
Rennes	28	15.0	5.0	8.0	38.0	24.0	14.0	50.0	3
Lille	28	15.0	4.0	9.0	35.0	27.0	8.0	49.0	4
Reims	28	10.0	11.0	7.0	26.0	21.0	5.0	41.0	5
Nice	28	11.0	8.0	9.0	41.0	38.0	3.0	41.0	6
Lyon	28	11.0	7.0	10.0	42.0	27.0	15.0	40.0	7
Montpellier	28	11.0	7.0	10.0	35.0	34.0	1.0	40.0	8
Monaco	28	11.0	7.0	10.0	44.0	44.0	0.0	40.0	9
Angers	28	11.0	6.0	11.0	28.0	33.0	-5.0	39.0	10
Strasbourg	27	11.0	5.0	11.0	32.0	32.0	0.0	38.0	11
Bordeaux	28	9.0	10.0	9.0	40.0	34.0	6.0	37.0	12
Nantes	28	11.0	4.0	13.0	28.0	31.0	-3.0	37.0	13
Brest	28	8.0	10.0	10.0	34.0	37.0	-3.0	34.0	14
Metz	28	8.0	10.0	10.0	27.0	35.0	-8.0	34.0	15
Dijon	28	7.0	9.0	12.0	27.0	37.0	-10.0	30.0	16
St Etienne	28	8.0	6.0	14.0	29.0	45.0	-16.0	30.0	17
Nimes	28	7.0	6.0	15.0	29.0	44.0	-15.0	27.0	18
Amiens	28	4.0	11.0	13.0	31.0	50.0	-19.0	23.0	19
Toulouse	28	3.0	4.0	21.0	22.0	58.0	-36.0	13.0	20

Figure 8: 2019/2020 Ligue 1's final result with limited round.

```
team_performance_compare_plot(['Barcelona', 'Real Madrid', 'Sevilla'], 2010)
```

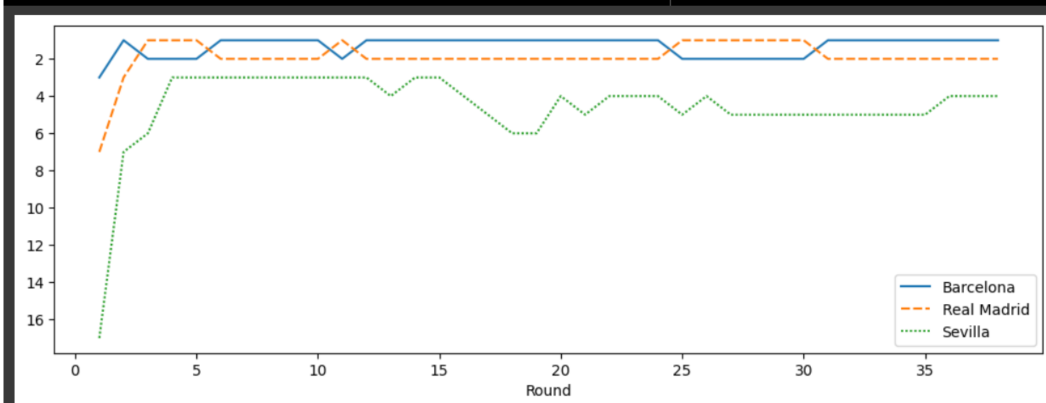

Figure 9: Performances of the choosing teams in 2009/2010 LaLiga.

```
head_to_head('Inter', 'Milan')
```

	Inter	stats	Milan
0	32.0	Matches	32.0
1	18.0	Won	8.0
2	6.0	Drawn	6.0
3	51.0	Goals	36.0
4	11.0	Clean Sheets	8.0

Figure 10: Head-to-head stats between Inter and AC depending on entire dataset (from 2006 to 2021 season).

All functions for these mini-tasks are the main content of `european_processing.py` file. For later consistent usage, I packaged definition of functions into `functions_collection.py` module. The results of every round are extracted in csv format too. Note that the data exported will be used for conveninent analyzing afterward, so they don't comply with any data normalization criterias.

4 Measures

Here we try to enlighten the hidden features of each league just by the prepared data. I used one straightforward measure and built two new measures on my own for this purpose.

4.1 The first measure

The first measure is the new position of the last champion. It easily defines by the final ranking in the current season of the very last champion of each league. Because every season has equal importance, I chose average as a statistics. Ideally, the higher the measure, the harsher the league. In that scenario, the number one team not only can't defend its title, but also end the season at far from the previous position.

$$\text{pre\_cur\_rank}(r) = \frac{\sum_{i=2}^{n} \text{cur\_rank}_i - r}{n}, \tag{2}$$

with n is the number of seasons we consider, argument r is the last rank, cur_rank_i is the new rank in ith season.

4.2 The second measure

The second measure is the variation of top teams through a season. The metric records upper teams in every round on season and return the changing cardinality of these sets. For instance, with the parameter of 4, we take 4 leading clubs in 10th round of 2008 LaLiga. One of them losed the 11th game, and fall down to 5th position. The measure have value 1 at 11th round of 2008 LaLiga because there's one club excluded from the old leading set in this round.

$$\begin{aligned} \text{top}(t, r) &= \{\text{team} \mid \text{ team's rank} \leq t \text{ in } r\text{th round}\}, \\ \text{upper\_change}(t, r) &= |\text{top}(t, r) - \text{top}(t, r-1)|, \end{aligned} \tag{3}$$

where t is argument of top positions and r is the round's index.

Values of all rounds will be aggregated to figure out the statistics of the season. Because the rounds are not in the same level, I used weighted average here. The top 4 clubs in each division will be the representatives of the national league that appear in UEFA Champions League next year. It's the honour of any big team and they compete every seasons for these ranks. The latter rounds must be weighted more to align with the effort of the teams at the ending moment of season. Here, I chose, for illustrative purpose, the value of 5 for the last 5 rounds and 3 for the next 5 rounds. All the other rounds, except 10 last rounds, will be put the weight of 1.

$$\text{weighted\_average}(x) = \frac{\sum_{i=1}^{n} w_i x_i}{n}, \tag{4}$$

where

$$\begin{aligned} w_i &= 1 \text{ for } i = \overline{1, n-11}, \\ w_i &= 3 \text{ for } i = \overline{n-10, n-6}, \\ w_i &= 5 \text{ for } i = \overline{n-5, n}, \end{aligned}$$

and n is the length of x.

Expectedly, the bigger the measure, the more varied the upper-rank set. It's mean the race is more serious. Obviously, other top parameter and weights can be chosen as long as they're likely.

4.3 The third measure

The third measure is the point's difference. I used the points of two consecutive teams after a round to measure the pace of the race. In each timepoint, we calculate the squared root of mean squared of differences. Again, weighted average is applied to all roots of the season. The latter the match, the more valuable the points. One point only, that equivalent to a tied match, make big difference in the result table. I chose the value of 0.2 for the last 5 rounds and 0.5 for the next 5 rounds. All the other rounds, except 10 last rounds, will ve put the weight of 1.

$$\text{diff\_point}(x) = \sqrt{\frac{\sum_{i=2}^{n} (x_i - x_{i-1})^2}{n-1}}, \tag{5}$$

where x is a sorted list, and n is the length of x.

$$\text{weighted\_average}(x) = \frac{\sum_{i=1}^{n} w_i x_i}{n}, \tag{6}$$

where

$$w_i = 1 \text{ for } i = \overline{1, n-11},$$
$$w_i = 0.5 \text{ for } i = \overline{n-10, n-6},$$
$$w_i = 0.2 \text{ for } i = \overline{n-5, n}.$$

The weighted_average functions will be applied to the list of diff_point results of all rounds in the season.

Logically, the smaller the measure, the more aggressive the season. It happens when the deviations among teams are very little. I also suggest another parameter of location in the table. The points of the top teams are fundamentally different from the bottom's. The squared root just takes the mean and may cause the bias when all proximities are considered in the same level. Functions are build for upper locations and lower locations. Of course, other locative paramaters and weights can be applied as long as they're reasonable.

5 Analyzing

In this section, I show the done analyses using the basic methods altogether the defined measure in the previous section.

5.1 Winrate

As I mentioned earlier, the original data aren't well-formatted to work with. But, we can easily get the meaningful information from them: winrate calculated from result columns.

H	45.941008
A	28.744438
D	25.314553

(a)

FTR	A	D	H
Div			
D1	30.085784	25.102124	44.812092
E0	29.539474	24.457237	46.003289
F1	27.228634	28.014718	44.756648
I1	28.305921	26.200658	45.493421
N1	29.261717	23.102447	47.635836
SP1	28.388158	24.555921	47.055921

(b)

FTR	A	D	H
season			
2006	27.485929	26.454034	46.060038
2007	26.266417	27.016886	46.716698
2008	27.298311	26.407129	46.294559
2009	27.532833	25.187617	47.279550
2010	26.454034	25.234522	48.311445
2011	26.641651	25.844278	47.514071
2012	27.157598	25.703565	47.138837
2013	28.658537	25.891182	45.450281
2014	29.362101	23.921201	46.716698
2015	29.362101	25.656660	44.981238
2016	29.924953	25.656660	44.418386
2017	28.611632	23.170732	48.217636
2018	30.347092	24.390244	45.262664
2019	29.502814	24.953096	45.544090
2020	31.016863	23.965253	45.017885
2021	34.474672	25.469043	40.056285

(c)

Figure 11: Home winrate filtered by (a) no filter, (b) division's filter, and (c) season's filter, respectively.

Because the result of a match only takes three cases, home win, away win or draw, home winrate tables also show away winrate values.

5.2 Goal scores

Next, we examine the factors that can strongly affect the final ranks. We know that the more win games a team have, the more points they earn and consequent the higher rank. But the goal score is another aspect. A club can win a match by just one goal in difference, and can lose a game by unlimited goal conceded. Based on this fact, I dived in the relation between the goal score rank and the true position at the board.

The relation is quantitied by the deviation of 2 corresponding ranks. For example, if the top 3rd club of the season achieve top 5th goal commited score, we have the deviation of 2. There're 3 goal scores are applied, GF (goal for) score, GA (goal against) score, and GD (goal difference) score.

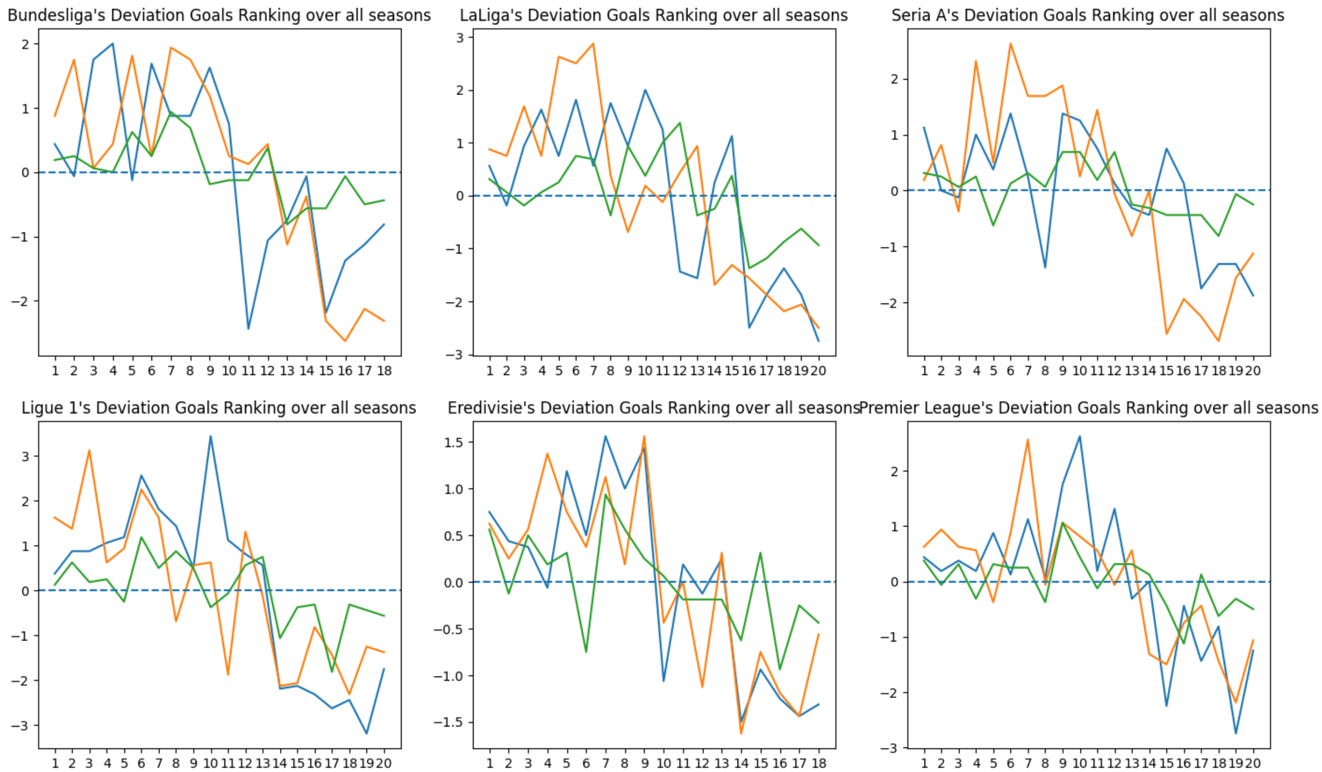

Figure 12: The deviations of 3 goal score ranks to real rank of 6 leagues, plot against time axis.

GA score is used in reverse order, because a strong team require excelent defenders that show small GA score by means of data-driven. We can easily note that GD score is the fittest score for estimating the team's rank. Mean statistics also show the similarity.

	D1	E0	F1	I1	N1	SP1
GF_rank	1.111111	0.92500	1.66250	0.8500	0.854167	1.35625
GA_rank	1.208333	0.91875	1.40625	1.3375	0.791667	1.40000
GD_rank	0.375000	0.38750	0.55625	0.3625	0.409722	0.61875

Figure 13: Averages are calculated over the time of each division.

In all leagues, the pattern is same: GD score is smallest. What can we deduce from this? The answer is the big team, whose position is relatively good, has not only more goals but also less goal against score. The balance between attacking and defending abilities becomes significant. The signed contracts with shocked values for the defensive positions of clubs are wise and reasonable decisions.

5.3 The first measure

Now, we begin to exploit the measures built before. Let's me recall that the first measure is about the down rank of the last champions. Here I expanded the function to include information of top 4 teams of the previous season.

Figure 14: Rank's change of number one clubs over time of all leagues.

There's significantly high values of red line in EPL, the most interesting division in the world. Let's show some numbers.

	D1	E0	F1	I1	N1	SP1
0	2.1875	3.5000	2.3125	1.3125	1.9375	1.8125
1	5.2500	3.1250	4.2500	3.3750	2.6875	2.0000
2	5.5625	2.6875	5.1875	4.2500	4.1875	2.6875
3	5.2500	3.7500	7.8125	6.1250	4.1875	7.1250

Figure 15: Average of rank's change over all time.

In the first row, the extreme value of EPL (3.5) is partly due to the fail of Leicester City when they created the fairy tale of sport's history by winning the previous season. For the remaining rows, it turns out that EPL has smallest values. It's mean, except the champion, all other top teams maintained their performances better over consecutive seasons. Use this result, we can state that EPL is the fiercest division by means of the fall of the last champion.

But in case of the other upper posisions, EPL is the most stable. So the chance of taking part in the continential cups is wider for all teams of the other national leagues. We can clearly look at the crazy plot of Ligue 1.

Take a look at LaLiga's numbers. The first 3 titles are also small compared to real values. However, the 4th position is too far from 4 (7.125). This's perhaps because there're only three steady big teams, Barca, Real, Atletico and the other clubs can only compete for another seat of top four.

5.4 The second measure

The second measure quantities top members across the season.

	D1	E0	F1	I1	N1	SP1
2006	0.090909	0.486486	0.972973	0.594595	0.696970	0.648649
2007	0.363636	0.540541	1.108108	0.648649	0.272727	0.621622
2008	1.484848	0.648649	0.540541	0.594595	1.242424	0.567568
2009	0.727273	0.540541	1.189189	0.702703	0.666667	0.513514
2010	0.666667	0.621622	1.054054	0.540541	0.454545	0.540541
2011	0.696970	0.243243	0.837838	0.783784	0.787879	0.216216
2012	0.545455	0.459459	0.783784	0.810811	0.909091	0.432432
2013	0.575758	0.594595	1.081081	0.351351	0.363636	0.729730
2014	0.484848	0.405405	0.243243	0.216216	0.939394	0.108108
2015	0.757576	0.324324	0.675676	0.864865	0.666667	0.351351
2016	1.090909	0.324324	1.135135	0.486486	0.818182	0.270270
2017	0.606061	0.324324	0.405405	0.702703	0.333333	0.459459
2018	1.121212	0.405405	0.243243	0.378378	0.606061	0.216216
2019	0.787879	0.918919	0.513514	0.486486	0.303030	1.000000
2020	0.969697	0.378378	1.259259	0.405405	0.760000	0.702703
2021	0.515152	0.918919	0.486486	0.972973	0.515152	0.486486

Figure 16: The average values of variations of every season with the parameter of 4.

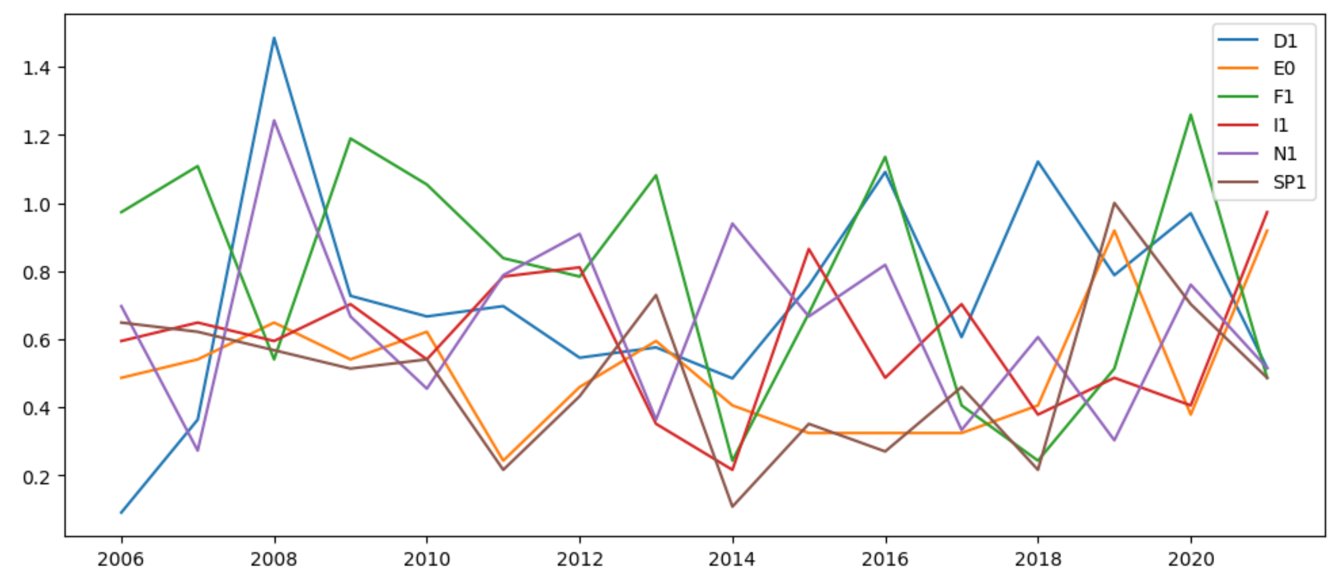

Figure 17: The plot of above stats table.

The orange line of EPL is more stable and lower than the other leagues. Let's apply again with the parameter of 1, that means we consider only first leading team.

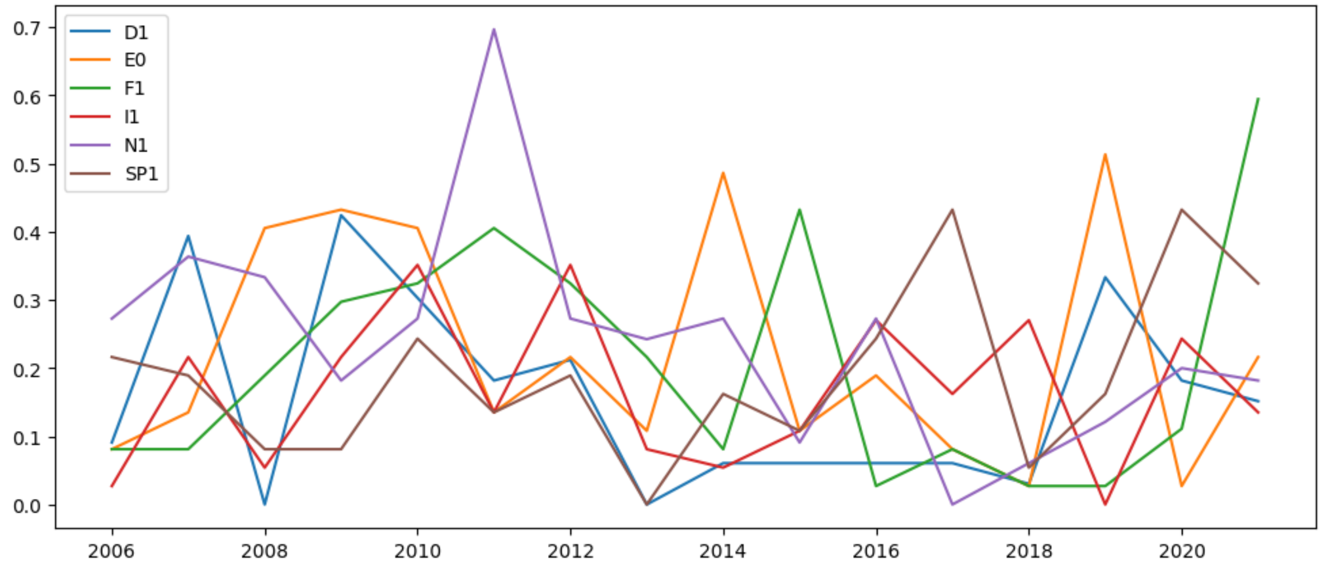

Figure 18: The plot of variations with the parameter of 1.

Here're the statistics of the two above plots.

```
top_4_variation.mean()

D1      0.717803
E0      0.508446
F1      0.783096
I1      0.596284
N1      0.645985
SP1     0.491554
dtype: float64
```

(a)

```
top_1_variation.mean()

D1      0.159091
E0      0.222973
F1      0.206269
I1      0.167230
N1      0.239773
SP1     0.190878
dtype: float64
```

(b)

Figure 19: The average of measures over the time of each division.

There're the interesting properties revealed. Consider the highest position, i.e. parameter equals 1, Eredivisie has the biggest value (0.239773), and EPL is just the second one (0.222973). In the case of top 4 highest, Ligue 1 is the biggest one (0.783096) with significantly far way others value. EPL is the second smallest (0.508446) and higher just a little bit compared to LaLiga (0.491554).

According to this information, we shown the features that the competition for the champion title in EPL is relative hight, but Ligue 1 is most aggressive by means of the disorder for other titles. Put together with the previous conclusion, especially the French's plot in Figure 14, we have the stable patterns of EPL and Ligue 1. In EPL, the battle for top 4 isn't serious compared to other leagues, but the potential champion title varied most. Ligue 1 is the fiercest division for the continental cup's positions. This trend is consistent not only round-by-round using the second measure, but also year-by-year using the first measure.

5.5 The third measure

The third measure is position's proximity through the season.

	D1	E0	F1	I1	N1	SP1
2006	1.945844	2.685264	2.085829	2.226633	2.337157	1.681333
2007	1.313331	2.038758	2.264038	2.352884	1.965447	1.543392
2008	1.490289	2.067931	2.002616	1.778579	1.631678	2.006432
2009	1.334350	1.635367	1.601222	1.682381	1.731939	2.066617
2010	1.579030	1.840013	1.910904	1.739811	2.143992	2.308519
2011	1.905978	1.452540	2.068804	1.577322	1.843483	2.181718
2012	1.540829	1.983686	1.506987	1.646185	1.708331	2.386639
2013	2.480542	1.985673	1.497831	1.687498	1.689790	2.320060
2014	2.466952	1.679445	1.926601	2.206814	1.285342	2.339920
2015	1.888713	1.760569	1.448442	2.118268	2.067890	1.882064
2016	2.355953	1.689547	3.232557	1.715121	2.198306	1.832216
2017	1.950804	1.916337	1.903244	2.048305	1.996890	1.830590
2018	2.418850	2.454647	2.524797	2.624096	1.947032	2.255043
2019	1.773383	2.275985	2.577371	2.579192	2.247457	1.692149
2020	1.582613	2.764342	1.392353	1.982186	1.445788	1.627398
2021	1.852419	2.054652	1.807276	1.890429	1.919183	1.684110

Figure 20: Collection table of values aggregated by every rounds.

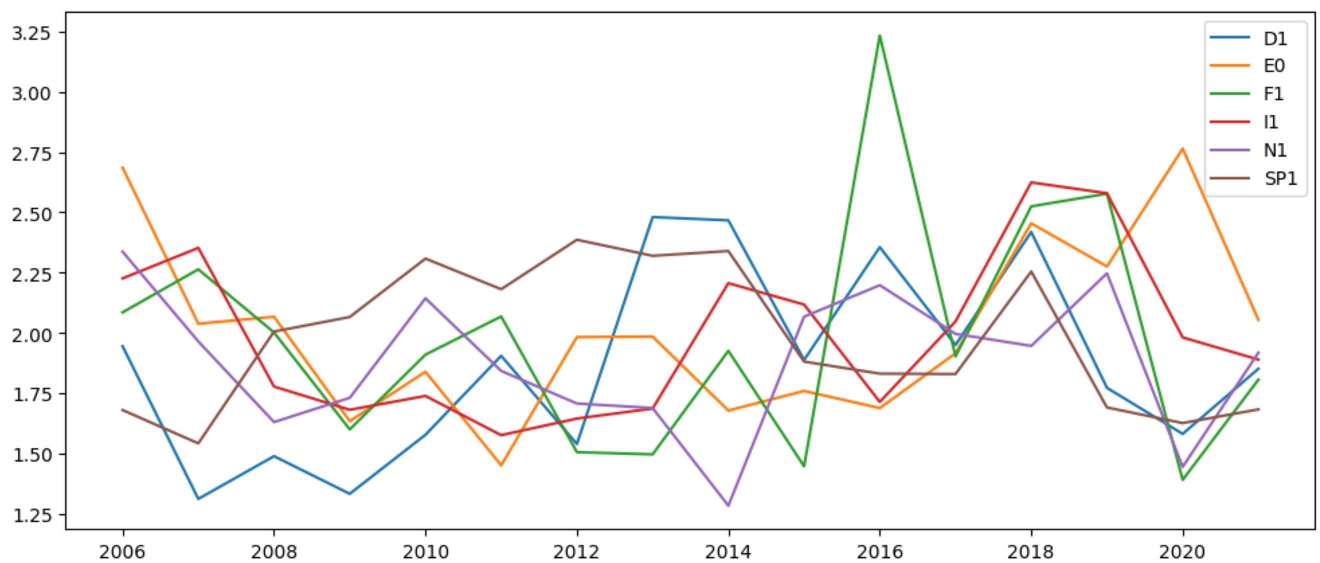

Figure 21: The plot of the above table.

Choose the other options for locative parameter, we have

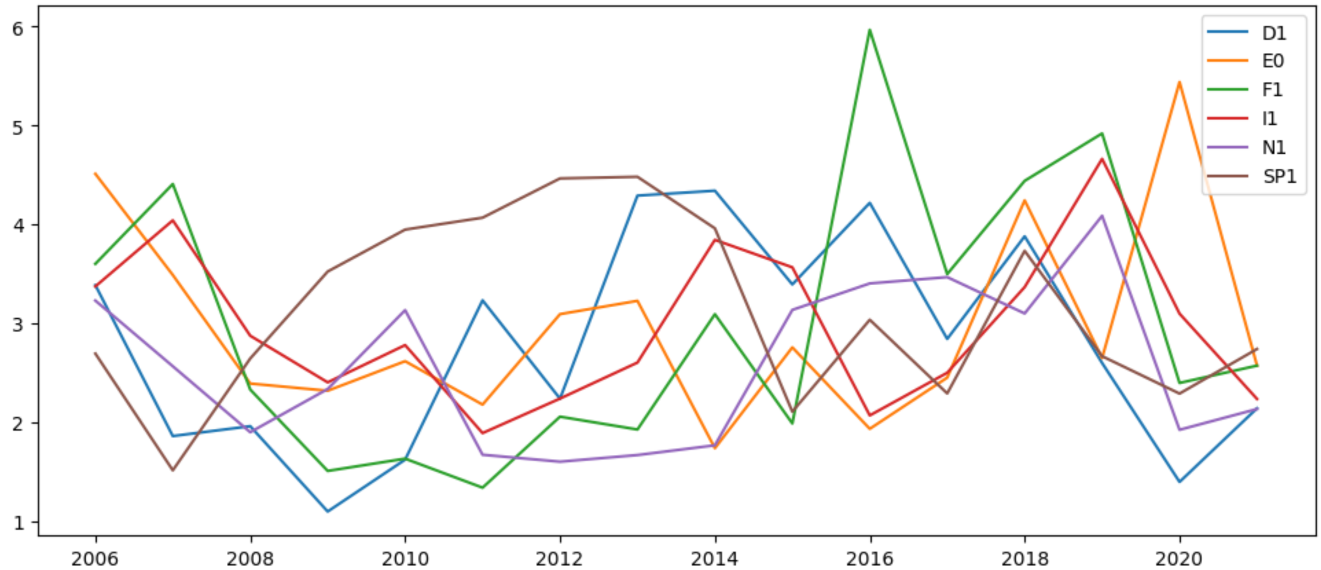

Figure 22: The plotting result of applying for just top 5 teams.

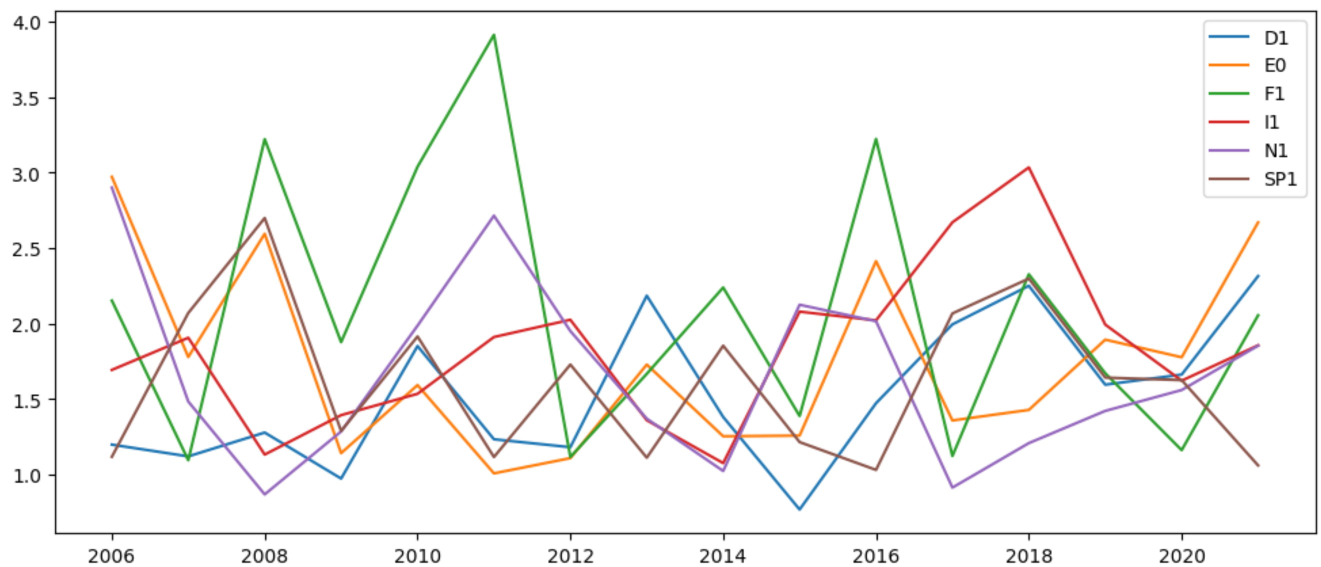

Figure 23: The plotting result of applying for just bottom 5 teams.

Here's the comparative sheet for the three plots using mean as aggregation.

	total	top_local	bot_local
D1	1.867492	2.780491	1.529530
E0	2.017797	2.975066	1.749250
F1	1.984430	2.979220	2.079851
I1	1.990982	2.971093	1.832442
N1	1.884982	2.569425	1.668260
SP1	1.977387	3.134674	1.615693

We see that:

- There's the same pattern with different scales between total plot and top_local plot.
- For 5 of 6 leagues, except Ligue 1, total locative values is smaller than top locative values and higher than bot locative values.
- The smallest values belong to Bundesliga and Eredivisie.

What do theses mean?

Firstly, the total locative measure is highly affected by the top locative measure. We can estimate that the proximities of all teams will be high when the top teams' high and vice versa.

Secondly, my early argument is true that there's significant gap between best teams and bottom teams. In general, the measure with top optional parameter has higher value than the bottom one. This mean the race for final titles is less aggresive than the competition for evading relegation zone, when the teams don't want to fall down into the lower league in the ordered system.

Thirdly, the lowest of German and Dutch national leagues is explainable. Because these divisions have just 18 competitors, so the total number of matches is not so high. The less the match's amount, the less the points are distributed. Consequently, the proximity that is decided by the diffences of points is stricter. Equally, we compare the other divisions

	total	top_local	bot_local
E0	2.017797	2.975066	1.749250
F1	1.984430	2.979220	2.079851
I1	1.990982	2.971093	1.832442
SP1	1.977387	3.134674	1.615693

Figure 24: The comparision of 20-club divisions.

The lowest belongs to LaLiga.

6 Summary

Using the variety of metrics, I recap here the notable conclusions about data:

- Home winrate is high with the value approximate 45% by many filter contexts.
- GD score is the reasonable feature to assess the club. A strong team has strongly not only offensive but also deffensive abilities.
- The race for champion title is unpredictable in EPL, when the other titles of top four is stable.
- In the other hand, Ligue 1 has massive disorder of top four teams, every team tries to reach the UEFA Champions League.
- LaLiga is the harsh environment by the mean of point's proximity, but the top three positions are almost decided.
- In all the leagues, the battle for defending position of the upcoming season of the bottom teams is more attractive than the competition at top positions.

All these conclusions is achieved by using dataset we have in hand and the built measures. By applying other methods, or just by changing parameter's values, we can get deeper insights hopefully. This work is still being developed.