From your current rank and evolution, get to know how well it will end for you

Personal project

Carl-Erik Gauthier

Motivation

- 2019-2020 season from Ligue-1 and Ligue-2 were permanently interrupted
  - Position after 27 legs were frozen
  - loss of opportunity for teams to reach their objective in the 11 remaining games
- Is soccer ranking predictable ?
  - is it true for the main championships

Objectives

DS :

- Analytics : how is the point evolution depending on the final rank ? with regards to its historical performance
- Build a ML to predict the final ranking

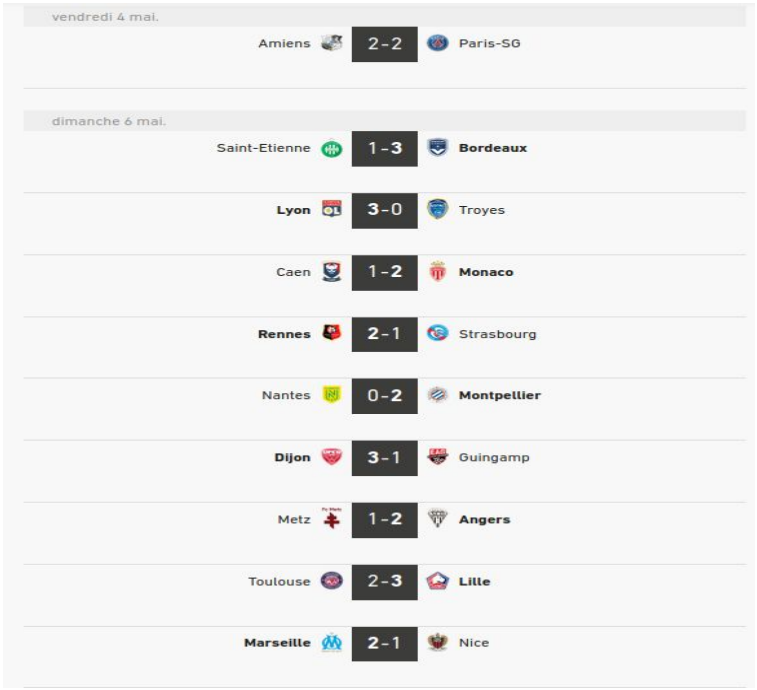
skills

- storytelling on personal project
- build a data science from data collection to app building

Data

- It has been scrapped on *L'Équipe* website
- It covers all season from 2004-2005 to 2018-2019 + the 2019-2020 for application
- Championships are :
  - Ligue-1 : FR top league
  - Ligue-2 : FR 2nd level
  - Bundesliga : DE top league
  - Premier-League : ENG top league
  - Serie-A : IT top league
  - La Liga : ESP top league

Data collection



⇒

	country	season	leg	team	play	goals_scored	opponent	goals_conceded	nb_points
0	France	2017-2018	36	Amiens	Home	2	Paris-SG	2	1
1	France	2017-2018	36	Paris-SG	Away	2	Amiens	2	1
2	France	2017-2018	36	Saint-Étienne	Home	1	Bordeaux	3	0
3	France	2017-2018	36	Bordeaux	Away	3	Saint-Étienne	1	3
4	France	2017-2018	36	Lyon	Home	3	Troyes	0	3
5	France	2017-2018	36	Troyes	Away	0	Lyon	3	0
6	France	2017-2018	36	Caen	Home	1	Monaco	2	0
7	France	2017-2018	36	Monaco	Away	2	Caen	1	3
8	France	2017-2018	36	Rennes	Home	2	Strasbourg	1	3
9	France	2017-2018	36	Strasbourg	Away	1	Rennes	2	0
10	France	2017-2018	36	Nantes	Home	0	Montpellier	2	0
11	France	2017-2018	36	Montpellier	Away	2	Nantes	0	3
12	France	2017-2018	36	Dijon	Home	3	Guingamp	1	3
13	France	2017-2018	36	Guingamp	Away	1	Dijon	3	0
14	France	2017-2018	36	Metz	Home	1	Angers	2	0
15	France	2017-2018	36	Angers	Away	2	Metz	1	3
16	France	2017-2018	36	Toulouse	Home	2	Lille	3	0
17	France	2017-2018	36	Lille	Away	3	Toulouse	2	3
18	France	2017-2018	36	Marseille	Home	2	Nice	1	3
19	France	2017-2018	36	Nice	Away	1	Marseille	2	0

Basic EDA

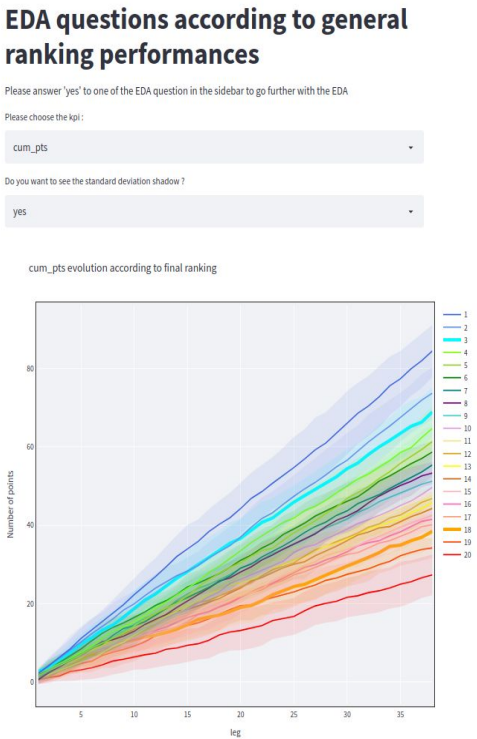
Championship	Nb. of participants (played min 1 season)	Nb. of teams having played all seasons	Nb. of distinct champions	Maximum number of titles won by 1 team
Ligue-1 (FR)	41	9	7	6 (PSG)
La Liga (ESP)	41	7	3	10 (FC Barcelona)
Serie-A (IT)	38	6	3	10 (Juventus Turin)
Bundesliga (GER)	34	6	4	11 (Bayern Munich)
Premier League (ENG)	39	7	4	5 (Chelsea, Man. Utd)
Ligue-2 (FR)	48	0	12	3 (Metz)

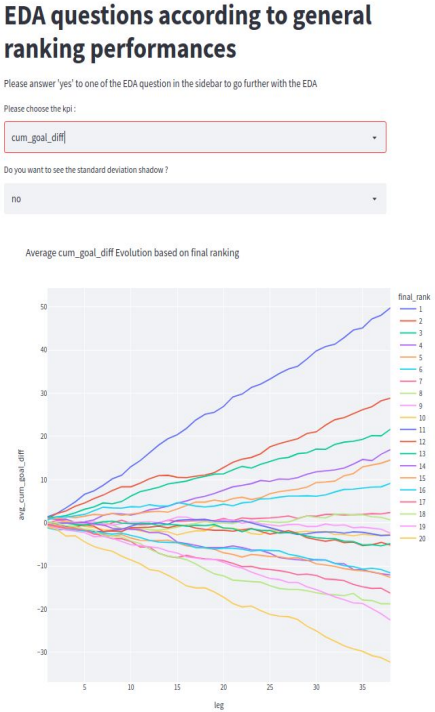
Note that Man. City won the Premier League 4 times out of the 15 considered seasons.

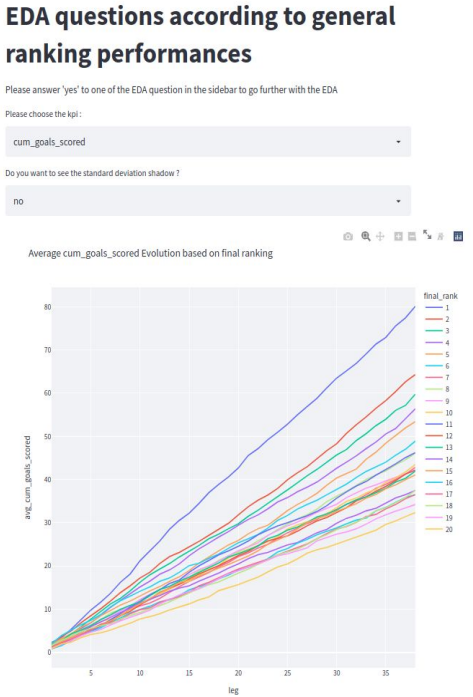
App Content

- EDA : based on the 15 ‘normal’ season
- EDA on 2019-2020 season : for seasons that are not over
- Prediction :
  - train a model
  - exploit an existing model

EDA - historical data

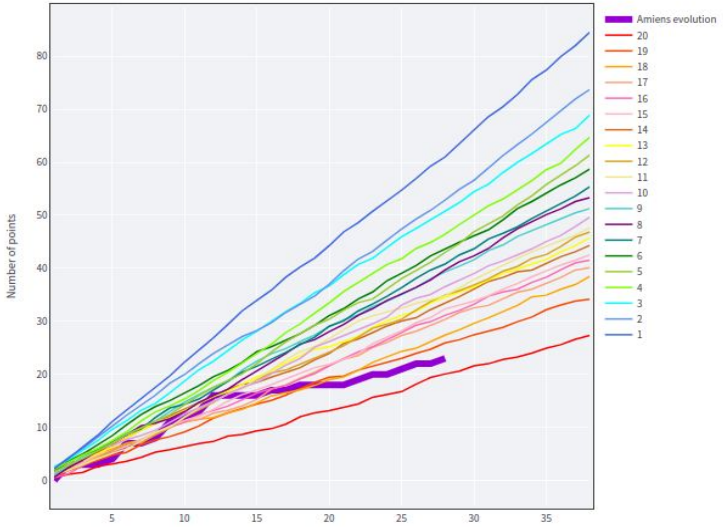


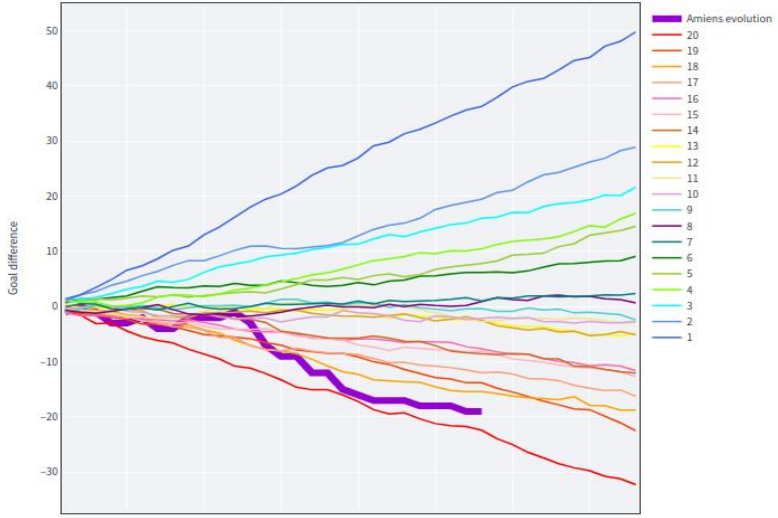




Plots are about Ligue-1. Left is the cumulative points evolution. Center is the goal difference and right is the number of goals scored

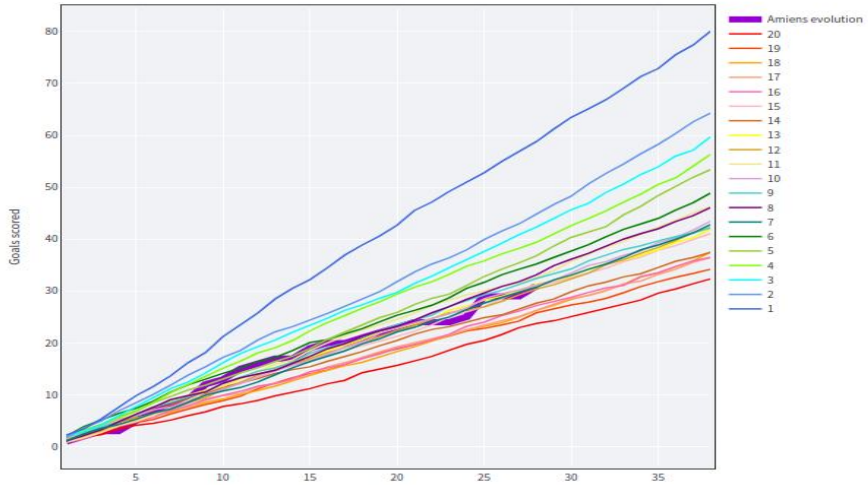
EDA about the 2019-2020 season





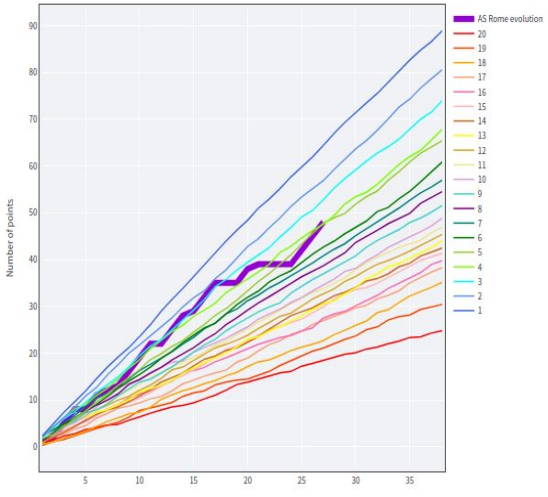
Amiens' performance. Left is the nb of points and right the goal difference

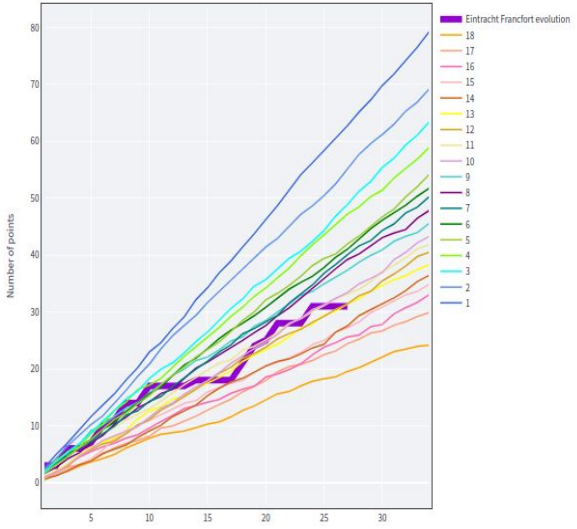
EDA about the 2019-2020 season

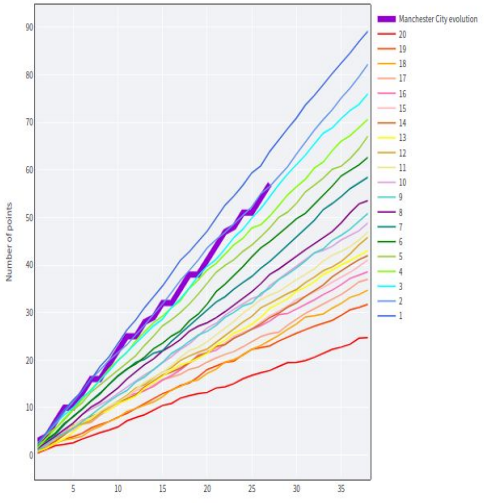


Amiens' performance when it comes to the number of goals scored

Some other examples about points evolution





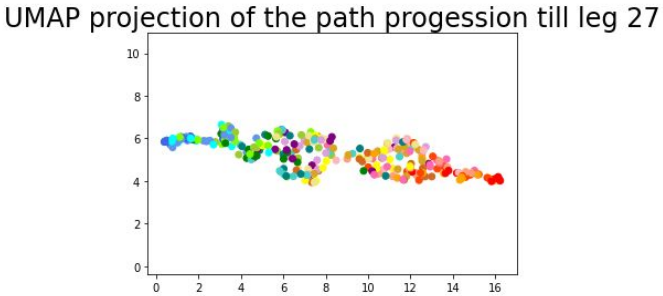


Left : AS Roma; center Eintracht Frankfurt and right Man. City

The different models

4 type of models have been implemented:

- ➔ **Naive** : compute the average the nb of points per game since the start of the season and use it to predict the final nb. of point
- ➔ **Regression** : target is the average nb of points to be won per game until the end of the championship ⇒ ranking based final number of points won
- ➔ **Ranker** : XGBoost Ranker algorithm ⇒ specially designed for ranking task
- ➔ **Classification** : uses teams' cumulative point evolution during the season



UMAP 2D embedding from the cumulative point evolution per team-season pairs in Ligue-1. Colors are related to the final rank (e.g. blue is rank 1 and red is rank 20)

Metric

The *Normalized Discounted Cumulative Gain* metric has been used

$$NDCG(\sigma) = DCG(\sigma) / DCG(\pi)$$

$$DCG(\sigma) = \sum_{k=1}^{nb.\,teams} \frac{g_{\sigma^{-1}(k)}}{log_2(1+k)}$$

where π is the optimal ranking and g the gain function. WLOG π = Id

In our case, the function was chosen to be

```python
def _get_rank_scoring(nb_teams: int):
    max_bonus = 250 * nb_teams
    return {1 + i // 2 if i % 2 == 0 else nb_teams - i // 2: max_bonus - i * 250 for i in range(nb_teams)}
```

⇒

```
{1: 5000,
 2: 4500,
 3: 4000,
 4: 3500,
 5: 3000,
 6: 2500,
 7: 2000,
 8: 1500,
 9: 1000,
 10: 500,
 11: 250,
 12: 750,
 13: 1250,
 14: 1750,
 15: 2250,
 16: 2750,
 17: 3250,
 18: 3750,
 19: 4250,
 20: 4750}
```

Prediction : Ligue-1

leg	team	rank	cum_pts	cum_goals_scored	cum_goal_diff
27	Paris-SG	1	68.0	75.0	51.0
27	Marseille	2	55.0	39.0	12.0
27	Rennes	3	47.0	33.0	9.0
27	Lille	4	46.0	34.0	7.0
27	Lyon	5	40.0	42.0	16.0
27	Montpellier	6	40.0	35.0	6.0
27	Monaco	7	40.0	43.0	1.0
27	Reims	8	38.0	25.0	4.0
27	Nice	9	38.0	39.0	2.0
27	Strasbourg	10	38.0	32.0	0.0
27	Nantes	11	37.0	28.0	-1.0
27	Bordeaux	12	36.0	39.0	6.0
27	Angers	13	36.0	26.0	-7.0
27	Brest	14	34.0	34.0	-2.0
27	Metz	15	31.0	25.0	-9.0
27	Saint-Étienne	16	29.0	28.0	-16.0
27	Dijon	17	27.0	25.0	-11.0
27	Nîmes	18	27.0	28.0	-14.0
27	Amiens	19	22.0	29.0	-19.0
27	Toulouse	20	13.0	21.0	-35.0

⇒

RANK	NAIVE	REGRESSION TYPE	CLASSIFICATION TYPE	RANKING TYPE
1	Paris-SG	Paris-SG	Paris-SG	Paris-SG
2	Marseille	Marseille	Marseille	Marseille
3	Rennes	Rennes	Rennes	Rennes
4	Lille	Lille	Lille	Lille
5	Monaco	Lyon	Montpellier	Nice
6	Montpellier	Montpellier	Bordeaux	Montpellier
7	Lyon	Monaco	Nice	Lyon
8	Strasbourg	Nice	Strasbourg	Nantes
9	Nice	Reims	Angers	Monaco
10	Reims	Strasbourg	Nantes	Bordeaux
11	Nantes	Nantes	Reims	Strasbourg
12	Bordeaux	Bordeaux	Lyon	Reims
13	Angers	Angers	Monaco	Saint-Étienne
14	Brest	Brest	Brest	Angers
15	Metz	Metz	Saint-Étienne	Brest
16	Saint-Étienne	Saint-Étienne	Metz	Nîmes
17	Dijon	Dijon	Nîmes	Dijon
18	Nîmes	Nîmes	Dijon	Metz
19	Amiens	Amiens	Amiens	Amiens
20	Toulouse	Toulouse	Toulouse	Toulouse

Prediction Serie-A

leg	team	rank	cum_pts	cum_goals_scored	cum_goal_diff
27	Juventus Turin	1	66	52	28
27	Lazio Rome	2	62	62	36
27	Inter Milan	3	58	54	26
27	Atalanta Bergame	4	54	77	40
27	AS Rome	5	48	53	17
27	Naples	6	42	43	7
27	Parme	7	39	37	4
27	AC Milan	8	39	32	-3
27	Hellas Vérone	9	38	31	2
27	Cagliari	10	35	43	1
27	Bologne	11	34	38	-6
27	Sassuolo	12	33	45	-1
27	Fiorentina	13	31	33	-4
27	Torino	14	31	30	-16
27	Udinese	15	28	21	-17
27	Sampdoria Gênes	16	26	30	-18
27	Genoa	17	25	32	-19
27	Lecce	18	25	35	-25
27	SPAL	19	18	20	-25
27	Brescia	20	17	23	-27

⇒

RANK	NAIVE	REGRESSION TYPE	CLASSIFICATION TYPE	RANKING TYPE	True result
1	Juventus Turin	Juventus Turin	Juventus Turin	Juventus Turin	**Juventus Turin (83, +17)**
2	Lazio Rome	Lazio Rome	Inter Milan	Inter Milan	**Inter Milan (82, +24)**
3	Inter Milan	Inter Milan	Lazio Rome	Lazio Rome	**Atalanta Bergame (78, +24)**
4	Atalanta Bergame	Atalanta Bergame	AS Rome	Atalanta Bergame	**Lazio Rome (78, +16)**
5	AS Rome	AS Rome	Atalanta Bergame	AS Rome	**AS Rome (70, +22)**
6	Naples	Naples	Naples	Bologne	**AC Milan (66, +27)**
7	Parme	Parme	Hellas Vérone	Parme	**Naples (62, +20)**
8	AC Milan	Hellas Vérone	Parme	Hellas Vérone	**Sassuolo (51, +18)**
9	Hellas Vérone	AC Milan	AC Milan	Naples	**Hellas Vérone (49, +11)**
10	Cagliari	Cagliari	Bologne	Cagliari	**Fiorentina (49, +18)**
11	Bologne	Bologne	Cagliari	AC Milan	**Parme (49, +10)**
12	Sassuolo	Sassuolo	Sassuolo	Sassuolo	**Bologne (47, +13)**
13	Torino	Fiorentina	Udinese	Torino	**Udinese (45, +17)**
14	Fiorentina	Torino	Fiorentina	Fiorentina	**Cagliari (45, +10)**
15	Udinese	Udinese	Torino	Udinese	**Sampdoria Gênes (42, +16)**
16	Sampdoria Gênes	Sampdoria Gênes	Genoa	Genoa	**Torino (40, +9)**
17	Genoa	Genoa	Sampdoria Gênes	Brescia	**Genoa (39, +14)**
18	Lecce	Lecce	Lecce	Lecce	**Lecce (35, +10)**
19	SPAL	SPAL	Brescia	Sampdoria Gênes	**Brescia (25, +8)**
20	Brescia	Brescia	SPAL	SPAL	**SPAL (20, +2)**

Conclusion

- ➔ Predictions not as good as expected
  - some teams ended very close to each other ⇒ very difficult to predict given that a win gives 3 pts and a tie 1 pt
  - better so for the interest of the sport
- ➔ Could practise some skills (data visualization, storytelling, scrapping, etc.)

Let’s go… to the app

1. Clone the Github repository
2. Create and activate a virtual environment
3. Within the environment install the requirements
4. Move to the *soccer_season_prediction* directory

   and run ***streamlit run soccer_front_app.py*** in your terminal to launch the app

5. … Play with the app :)