Prédiction des résultats de match de football

Mo LI, Zeyuan Wu

ENSAE Paris

Notre base des données

- Les données sur l'information de base des joueurs .
- Les données sur l'information de base des équipes
- Les données sur les capacités des joueurs
- Les données sur des statistiques techniques (les coins, les pénalités, les possessions des équipes…)
- …

Les facteurs utiles pour prédire le résultat d'un match de football

- Les capacités des joueurs des 2 équipes

  NB: 40 si aucune information dans la base des données

- Les états récents des 2 équipes
- Les record des confrontations passés…

```
Entrée [15]: fifaData.min()

Out[15]: home_player_1_overall_rating          48.0
         home_player_2_overall_rating          42.0
         home_player_3_overall_rating          45.0
         home_player_4_overall_rating          42.0
         home_player_5_overall_rating          42.0
         home_player_6_overall_rating          45.0
         home_player_7_overall_rating          44.0
         home_player_8_overall_rating          41.0
         home_player_9_overall_rating          40.0
         home_player_10_overall_rating         46.0
         home_player_11_overall_rating         40.0
         away_player_1_overall_rating          49.0
         away_player_2_overall_rating          42.0
         away_player_3_overall_rating          42.0
         away_player_4_overall_rating          42.0
         away_player_5_overall_rating          42.0
         away_player_6_overall_rating          40.0
         away_player_7_overall_rating          44.0
         away_player_8_overall_rating          42.0
         away_player_9_overall_rating          42.0
         away_player_10_overall_rating         40.0
         away_player_11_overall_rating         43.0
         match_api_id                      486350.0
         dtype: float64
```

Variables d'intérêt pour faire des pédictions

- Les formes des deux équipes des 5 matchs plus récent (mesurés par différence de buts
- Résultats des 5 derniers matches entre les deux équipes
- Le « League » de match

Lecture des données

In [13]:

```
# Chargements des données
database = './database.sqlite'
conn = sqlite3.connect(database)

playerData = pd.read_sql("SELECT * FROM Player;", conn)

playerStatsData = pd.read_sql("SELECT * FROM Player_Attributes;", conn)

teamData = pd.read_sql("SELECT * FROM Team;", conn)

matchData = pd.read_sql("SELECT * FROM Match;", conn)

```

playerData

In [212]:
```
playerData.head()
```

Out[212]:

	id	player_api_id	player_name	player_fifa_api_id	birthday	height	weight
0	1	505942	Aaron Appindangoye	218353	1992-02-29 00:00:00	182.88	187
1	2	155782	Aaron Cresswell	189615	1989-12-15 00:00:00	170.18	146
2	3	162549	Aaron Doran	186170	1991-05-13 00:00:00	170.18	163
3	4	30572	Aaron Galindo	140161	1982-05-08 00:00:00	182.88	198
4	5	23780	Aaron Hughes	17725	1979-11-08 00:00:00	182.88	154

teamData

In [214]:
```
teamData.head()
```

Out[214]:

	id	team_api_id	team_fifa_api_id	team_long_name	team_short_name
0	1	9987	673.0	KRC Genk	GEN
1	2	9993	675.0	Beerschot AC	BAC
2	3	10000	15005.0	SV Zulte-Waregem	ZUL
3	4	9994	2007.0	Sporting Lokeren	LOK
4	5	9984	1750.0	KSV Cercle Brugge	CEB

playerStatsData

In [213]:
```
playerStatsData.head()
```

Out[213]:

	id	player_fifa_api_id	player_api_id	date	overall_rating	potential	preferred_foot	attacking_work_rate	defensive_work_rate	crossing	...	vision	penalties
0	1	218353	505942	2016-02-18 00:00:00	67.0	71.0	right	medium	medium	49.0	...	54.0	48.0
1	2	218353	505942	2015-11-19 00:00:00	67.0	71.0	right	medium	medium	49.0	...	54.0	48.0
2	3	218353	505942	2015-09-21 00:00:00	62.0	66.0	right	medium	medium	49.0	...	54.0	48.0
3	4	218353	505942	2015-03-20 00:00:00	61.0	65.0	right	medium	medium	48.0	...	53.0	47.0
4	5	218353	505942	2007-02-22 00:00:00	61.0	65.0	right	medium	medium	48.0	...	53.0	47.0

matchData

In [15]:
```
matchData.head()
```

Out[15]:

	id	country_id	league_id	season	stage	date	match_api_id	home_team_api_id	away_team_api_id	home_team_goal	...	SJA	VCH	VCD	VCA
145	146	1	1	2008/2009	24	2009-02-27 00:00:00	493017	8203	9987	2	...	2.30	2.65	3.25	2.35
153	154	1	1	2008/2009	25	2009-03-08 00:00:00	493025	9984	8342	1	...	2.25	2.65	3.20	2.35
155	156	1	1	2008/2009	25	2009-03-07 00:00:00	493027	8635	10000	2	...	8.50	1.30	4.35	8.00
162	163	1	1	2008/2009	26	2009-03-13 00:00:00	493034	8203	8635	2	...	1.73	4.35	3.30	1.75
168	169	1	1	2008/2009	26	2009-03-14 00:00:00	493040	10000	9999	0	...	5.00	1.65	3.50	4.50

Obtenir des variables pour faire des pédictions

def get_match_result(match):

```
In [219]: 1  get_match_result(matchData.iloc[1])

             match_api_id   label
          0      493025.0  Defeat

Out[219]: match_api_id    493025
          label           Defeat
          Name: 0, dtype: object
```

Obtenir des variables pour faire des pédictions

def get_fifa_stats(

```
In [205]: 1 get_fifa_stats(matchData.iloc[1],playerStatsData)

Out[205]: home_player_1_overall_rating         64.0
          home_player_2_overall_rating         64.0
          home_player_3_overall_rating         63.0
          home_player_4_overall_rating         62.0
          home_player_5_overall_rating         62.0
          home_player_6_overall_rating         72.0
          home_player_7_overall_rating         68.0
          home_player_8_overall_rating         67.0
          home_player_9_overall_rating         69.0
          home_player_10_overall_rating        68.0
          home_player_11_overall_rating        65.0
          away_player_1_overall_rating         73.0
          away_player_2_overall_rating         67.0
          away_player_3_overall_rating         66.0
          away_player_4_overall_rating         67.0
          away_player_5_overall_rating         66.0
          away_player_6_overall_rating         70.0
          away_player_7_overall_rating         69.0
          away_player_8_overall_rating         68.0
          away_player_9_overall_rating         67.0
          away_player_10_overall_rating        73.0
          away_player_11_overall_rating        68.0
          match_api_id                    493025.0
          Name: 0, dtype: float64
```

Obtenir des variables pour faire des pédictions

In [146]:
```
get_last_matches(matchData, '20200101',8650)
```

Out[146]:

	id	country_id	league_id	season	stage	date	match_api_id	home_team_api_id	away_team_api_id	home_team_goal	...
4707	4708	1729	1729	2015/2016	38	2016-05-15 00:00:00	1987606	8659	8650	1	...
4622	4623	1729	1729	2015/2016	30	2016-05-11 00:00:00	1989004	8650	8455	1	...
4692	4693	1729	1729	2015/2016	37	2016-05-08 00:00:00	1989074	8650	9817	2	...
4685	4686	1729	1729	2015/2016	36	2016-05-01 00:00:00	1989067	10003	8650	3	...
4672	4673	1729	1729	2015/2016	35	2016-04-23 00:00:00	1989054	8650	10261	2	...
4579	4580	1729	1729	2015/2016	27	2016-04-20 00:00:00	1988971	8650	8668	4	...

Obtenir des variables pour faire des pédictions

In [155]:
```
get_last_facing(matchData,'20200101',10260,8650,11)
```

Out[155]:

	id	country_id	league_id	season	stage	date	match_api_id	home_team_api_id	away_team_api_id	home_team_goal	..
4531	4532	1729	1729	2015/2016	22	2016-01-17 00:00:00	1988923	8650	10260	0	..
4722	4723	1729	1729	2015/2016	5	2015-09-12 00:00:00	1988754	10260	8650	3	..
4240	4241	1729	1729	2014/2015	30	2015-03-22 00:00:00	1724274	8650	10260	1	..
4084	4085	1729	1729	2014/2015	16	2014-12-14 00:00:00	1724138	10260	8650	3	..
3863	3864	1729	1729	2013/2014	30	2014-03-16 00:00:00	1475020	10260	8650	0	..
3852	3853	1729	1729	2013/2014	3	2013-09-01 00:00:00	1474162	8650	10260	1	..
3392	3393	1729	1729	2012/2013	22	2013-01-13 00:00:00	1229333	10260	8650	2	..
3579	3580	1729	1729	2012/2013	5	2012-09-23 00:00:00	1228368	8650	10260	1	..

Obtenir des variables pour faire des pédictions

def get_goals(matches, team):

```
In [164]:  1  get_goals(matchData1,10260)
Out[164]:  572
```

def get_goals_conceided(matches, team):

```
In [167]:  1  get_goals_conceided(matchData, 10260)
Out[167]:  278
```

Obtenir des variables pour faire des pédictions

def get_wins(matches, team):

```
In [169]:  1  get_wins(matchData,8650)
Out[169]:  147
```

Obtenir des variables pour faire des pédictions

def get_match_features(match, matches, x = 5):

```
In [217]:  1  get_match_features(matchData.loc[4531],matchData)

Out[217]: match_api_id                  1988923.0
          league_id                        1729.0
          home_team_goals_difference         -1.0
          away_team_goals_difference         -2.0
          games_won_home_team                 4.0
          games_won_away_team                 2.0
          games_against_won                   2.0
          games_against_lost                  6.0
          Name: 0, dtype: float64
```

Obtenir des variables pour faire des pédictions

In [258]:
```
create_feables(matchData,fifaData)
```

Out[258]:

	match_api_id	home_team_goals_difference	away_team_goals_difference	games_won_home_team	games_won_away_team	games_against_won	games_against
0	493017.0	0.0	0.0	0.0	0.0	0.0	
1	493025.0	0.0	0.0	0.0	0.0	0.0	
2	493027.0	0.0	0.0	0.0	0.0	0.0	
3	493034.0	1.0	2.0	1.0	1.0	0.0	
4	493040.0	-2.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	
95	665753.0	1.0	-10.0	3.0	2.0	0.0	
96	665757.0	-8.0	6.0	2.0	5.0	0.0	
97	665334.0	9.0	0.0	6.0	1.0	0.0	
98	665335.0	-3.0	6.0	1.0	5.0	0.0	
99	665337.0	-6.0	-3.0	1.0	0.0	0.0	

Features

In [254]:

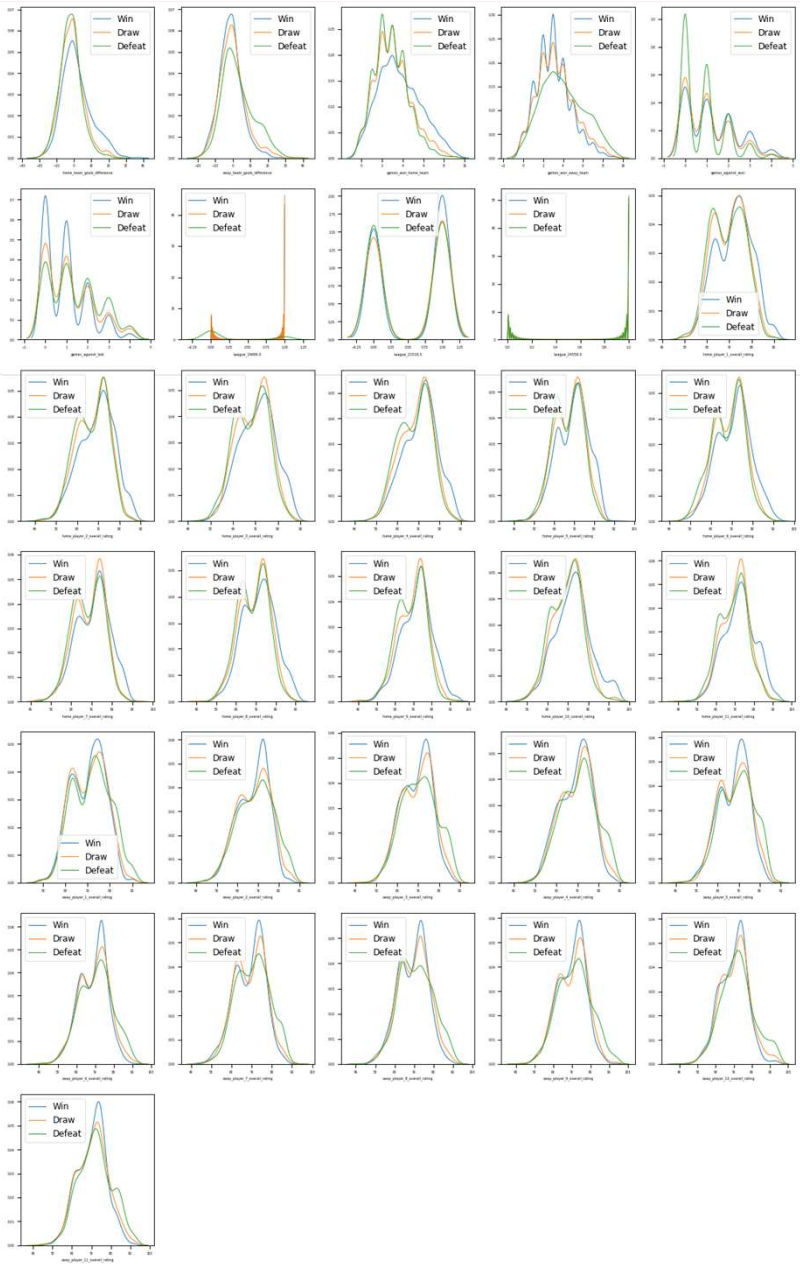
```
for x in features.columns:
    print(x)
```

```
home_team_goals_difference
away_team_goals_difference
games_won_home_team
games_won_away_team
games_against_won
games_against_lost
League_1.0
League_1729.0
League_4769.0
League_7809.0
League_10257.0
League_13274.0
League_15722.0
League_17642.0
League_19694.0
League_21518.0
League_24558.0
home_player_1_overall_rating
home_player_2_overall_rating
home_player_3_overall_rating
home_player_4_overall_rating
home_player_5_overall_rating
home_player_6_overall_rating
home_player_7_overall_rating
home_player_8_overall_rating
home_player_9_overall_rating
home_player_10_overall_rating
home_player_11_overall_rating
away_player_1_overall_rating
away_player_2_overall_rating
```

Label

```
In [255]: labels

Out[255]: 0        Win
          1     Defeat
          2        Win
          3        Win
          4       Draw
                 ...
          95       Win
          96       Win
          97       Win
          98      Draw
          99      Draw
          Name: label, Length: 100, dtype: object
```



- En bleu: l'équipe gagné
- En vert : l'équipe perdu
- En orange : les deux équipes à égalité

- Les équipes gagné sont les capabilités des joueurs plutôt « à droite »
- Les formes des équipes sont aussi signifiants

Les modèles

- On a trop de variable – considère de faire une ACP

- Comparer les effets des algorithmes suivants :

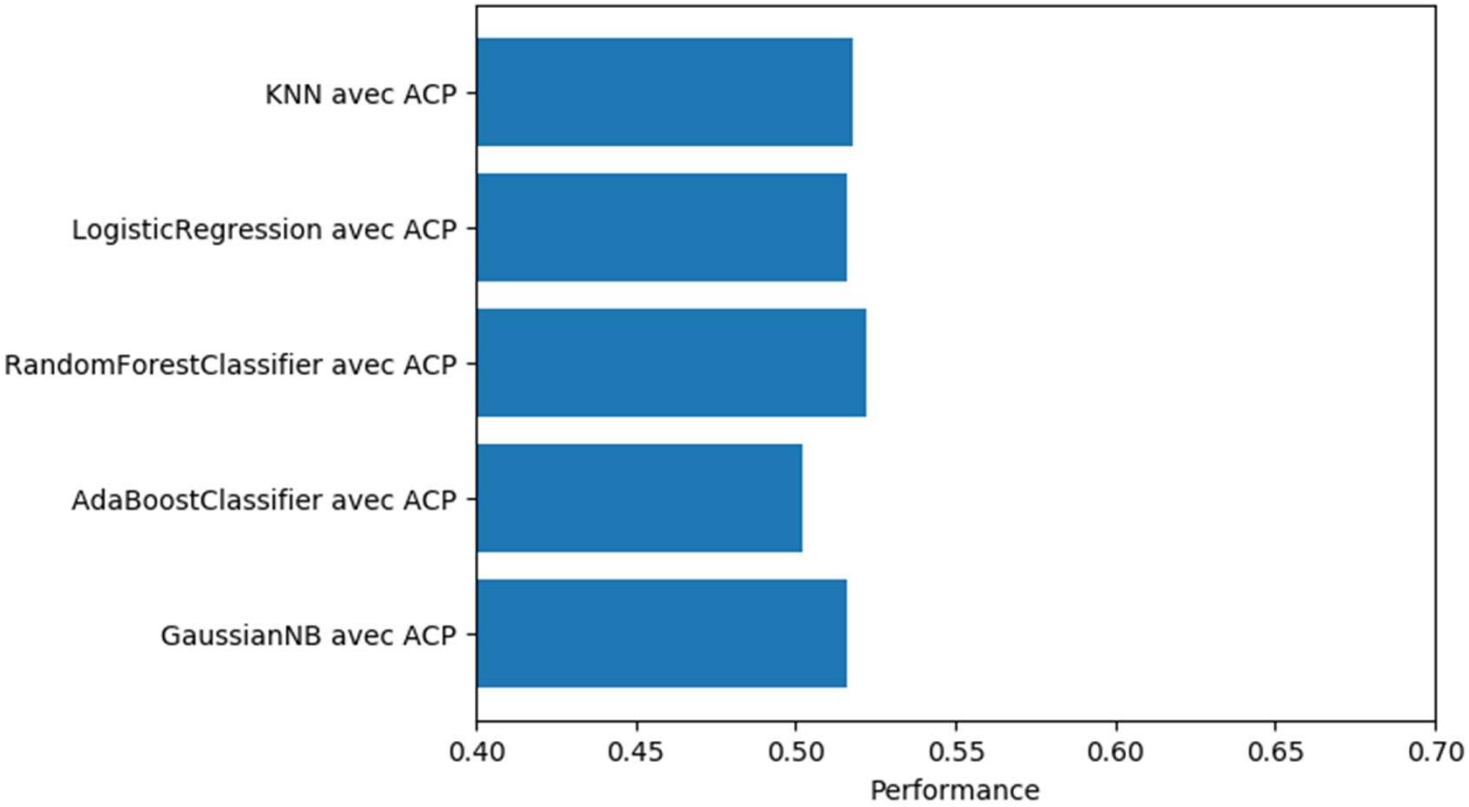
  KNN, Logistique régression, Forêt aléatoire, AdaBoost, GaussianNB

Les modèles

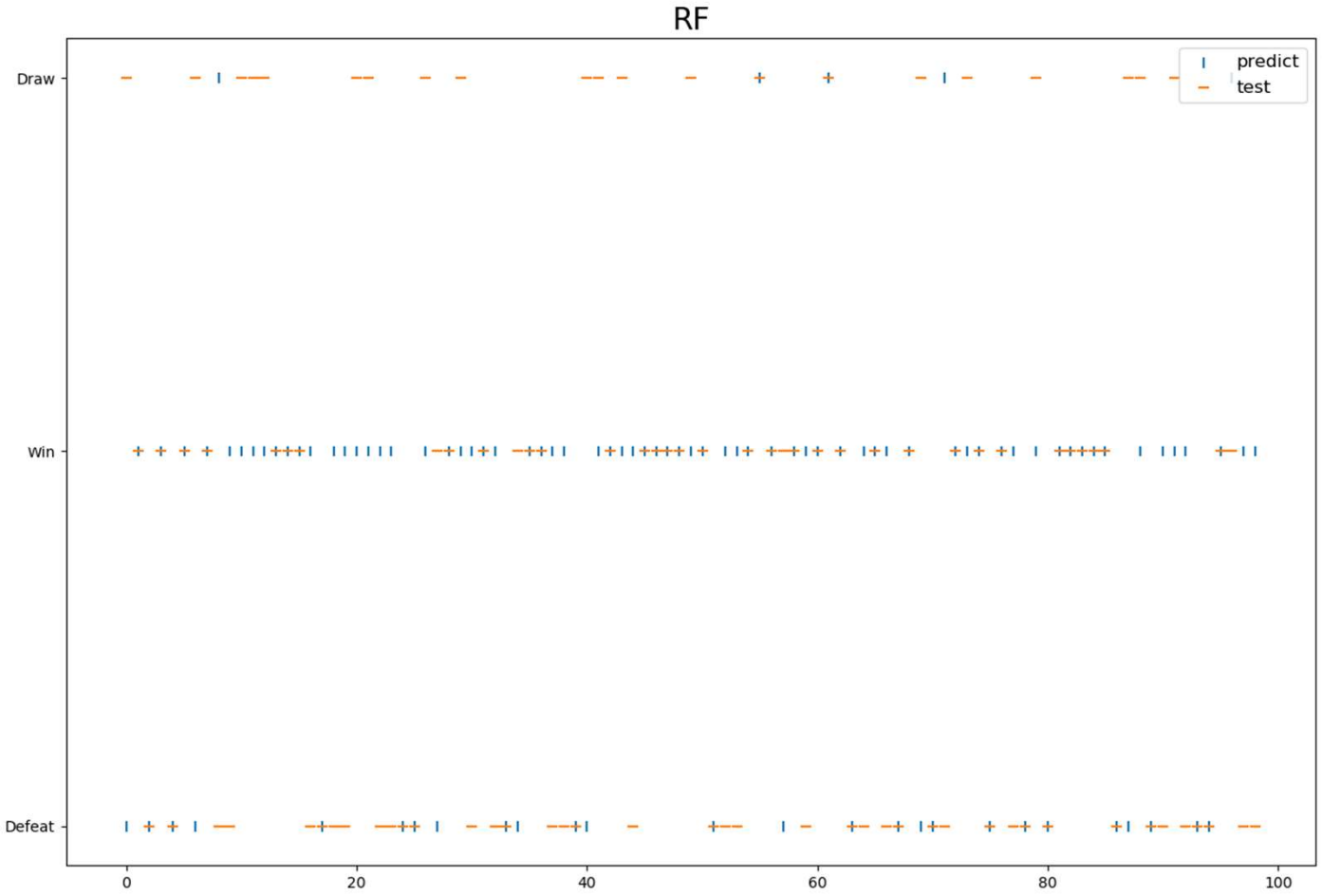
Entrée [66]:
```
n_jobs = 1
clfsFB, dimensionReductionsFB, trainScores, testScores = find_best_classifier(clfs, dimensionReductions, scorer,
                                                                              X_train, y_train, X_calibrate,
                                                                              y_calibrate, X_test, y_test,
                                                                              cv_sets, parameters, n_jobs)
```

```
L'apprentisage de KNeighborsClassifier avec PCA...
L'apprentisage de KNeighborsClassifier est fini
Le score de CalibratedClassifierCV pour base d'apprentissage: 0.5686.
Le score de CalibratedClassifierCV pour base de test: 0.5180.
L'apprentisage de LogisticRegression avec PCA...
L'apprentisage de LogisticRegression est fini
Le score de CalibratedClassifierCV pour base d'apprentissage: 0.5368.
Le score de CalibratedClassifierCV pour base de test: 0.5160.
L'apprentisage de RandomForestClassifier avec PCA...
L'apprentisage de RandomForestClassifier est fini
Le score de CalibratedClassifierCV pour base d'apprentissage: 0.9996.
Le score de CalibratedClassifierCV pour base de test: 0.5220.
L'apprentisage de AdaBoostClassifier avec PCA...
L'apprentisage de AdaBoostClassifier est fini
Le score de CalibratedClassifierCV pour base d'apprentissage: 0.5500.
Le score de CalibratedClassifierCV pour base de test: 0.5020.
L'apprentisage de GaussianNB avec PCA...
L'apprentisage de GaussianNB est fini
Le score de CalibratedClassifierCV pour base d'apprentissage: 0.5400.
Le score de CalibratedClassifierCV pour base de test: 0.5160.
```

Les modèles


KNN avec ACP
LogisticRegression avec ACP
RandomForestClassifier avec ACP
AdaBoostClassifier avec ACP
GaussianNB avec ACP
0.40
0.45
0.50
0.55
0.60
0.65
0.70
Performance

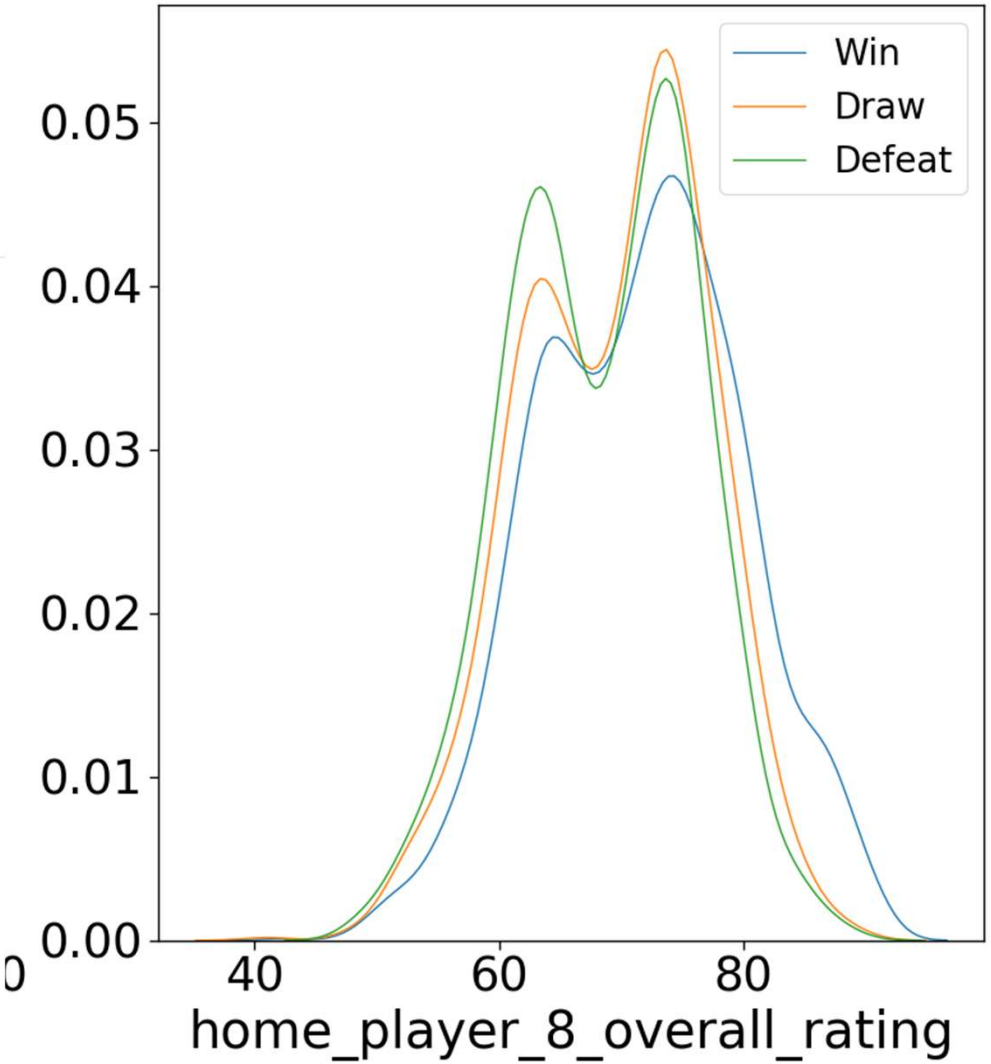
Les modèles



Conclusion & amélioration

- 1. Utiliser des données plus fiables pour représenter les capacités des joueurs.
- 2. Ajouter d'autres variables de régression
- 3. Essayer d'autres modèles

Merci


Win
Draw
Defeat
0.05
0.04
0.03
0.02
0.01
0.00
40
60
80
home_player_8_overall_rating