المدرسة الوطنية للعلوم التطبيقية - فاس
Ecole Nationale des Sciences Appliquées de Fès
جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

2ème Année Master Architectures et Applications des Systèmes d'Information

Prédiction du vainqueur de la Coupe du Monde de Football 2022

Réalisé par :

BOUAFIA Youssef

EL HADAD Amina

Responsable :

Pr. FRIKH Bouchra

Année Universitaire : 2023 /2024

Extraire et nettoyer des données historiques afin de leur appliquer un modèle statistique.

La coupe du monde de football 2022 a débuté dimanche
Sans se concentrer sur le côté sportif (forme et méforme des joueurs et équipes),
Emparons-nous du sujet pour en faire un projet DATA complet en 3 étapes:

1) Extraire les données historiques qui vont nous permettre de répondre à l'objectif
2) Nettoyer les données récoltées
3) Construire plusieurs modèles prédictifs et choisir le meilleur.

Nom	Modifié le	Type	Taille
donnees	02/01/2023 15:25	Dossier de fichiers	
1.scrapping_tableaux_phase_de_groupe	02/01/2023 15:24	JetBrains DataSpell	11 Ko
2.scrapping_match(Qata2022_worldcup)	02/01/2023 15:18	JetBrains DataSpell	4 Ko
3.cleaning_worldcup_match	02/01/2023 15:24	JetBrains DataSpell	5 Ko
4.cleaning_Qatatr2022_match	02/01/2023 15:25	JetBrains DataSpell	10 Ko
5.predection_Qatar2022	02/01/2023 14:06	JetBrains DataSpell	71 Ko

Figure 1- Dossier qui contient tout le projet

Étape 1 - Récolter des données historiques des précédentes Coupes du Monde de Football

utiliser Python et BeautifulSoup pour extraire les données de toutes les coupes du monde jouées jusqu'à présent (1930-2018) et le tableau de la coupe du monde 2022.

Donc une première étape classique de Web Scraping !

Faire le web scraping des datasets

```
In 8
import pandas as pd
import pickle
from string import ascii_uppercase as alphabet

In 9
all_tables = pd.read_html('https://en.wikipedia.org/wiki/2022_FIFA_World_Cup')

dict_table = {}
for letter, i in zip(alphabet, range(9,61,7)): # A=11, B=18, ...
    df = all_tables[i]
    df.rename(columns={df.columns[1]:'Team'}, inplace=True)
    df.pop('Qualification')
    dict_table[f'Group {letter}'] = df

In 10
dict_table.keys()

Out 10
dict_keys(['Group A', 'Group B', 'Group C', 'Group D', 'Group E', 'Group F', 'Group G', 'Group H'])

In 11
dict_table['Group F']

Out 11
4 rows  4 rows × 10 columns  CSV
```

	Pos	Team	Pld	W	D	L	GF	GA	GD	Pts
0	1	Morocco	3	2	1	0	4	1	+3	7
1	2	Croatia	3	1	2	0	4	1	+3	5
2	3	Belgium	3	1	1	1	1	2	-1	4
3	4	Canada	3	0	0	3	2	7	-5	0

```
In 12
with open('dict_table', 'wb') as output:
    pickle.dump(dict_table, output)
```

Figure 2- Scrapping des groupes stage

In 26

```python
import pandas as pd
from bs4 import BeautifulSoup
import requests
```

In 27

```python
def get_matches(y):
    site = f'https://en.wikipedia.org/wiki/{y}_FIFA_World_Cup'
    req = requests.get(site)
    content = req.text
    soup = BeautifulSoup(content, 'lxml')
    matches = soup.find_all('div', class_='footballbox')
    home = []
    score = []
    away = []
    for match in matches:
        home.append(match.find('th', class_='fhome').get_text())
        score.append(match.find('th', class_='fscore').get_text())
        away.append(match.find('th', class_='faway').get_text())

    dict_foot= {'home': home, 'score': score, 'away': away}
    football = pd.DataFrame(dict_foot)
    football['year'] = y
    return football
```

Figure 3-Fonction de Scrapping

```python
years = [ 1930,1934,1950,1954,1958,1962,1966,1970,1974,1978,1982,1986,1990,1994,1998,2002,2006, 2010, 2014, 2018]
```

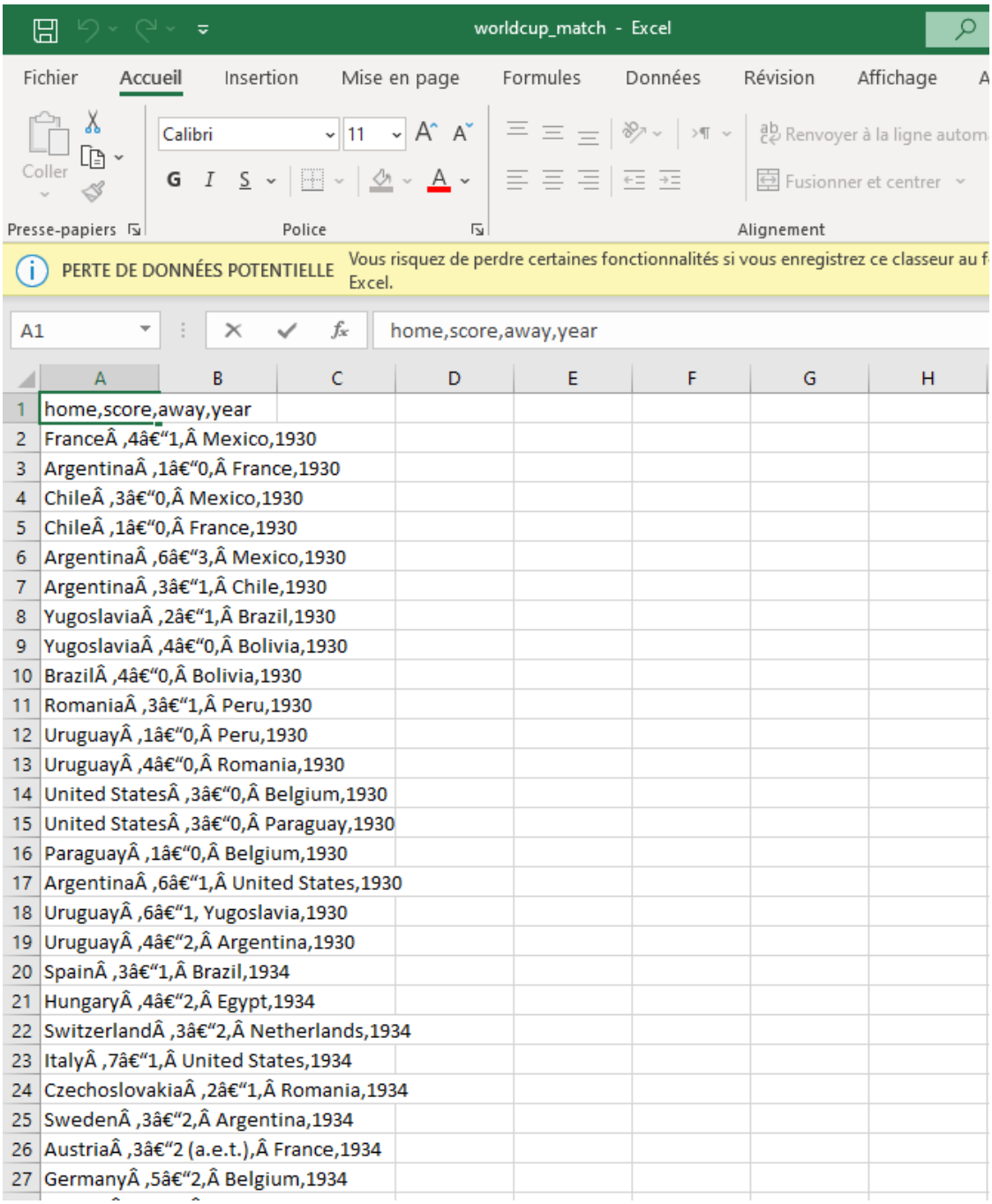
```python
# les anciennes match
```

```python
worldcup_match = [get_matches(year) for year in years]
worldcup_match = pd.concat(worldcup_match, ignore_index=True)
worldcup_match.to_csv("donnees/worldcup_match.csv", index=False)
```

```python
#les matches de coupe du monde 2022
```

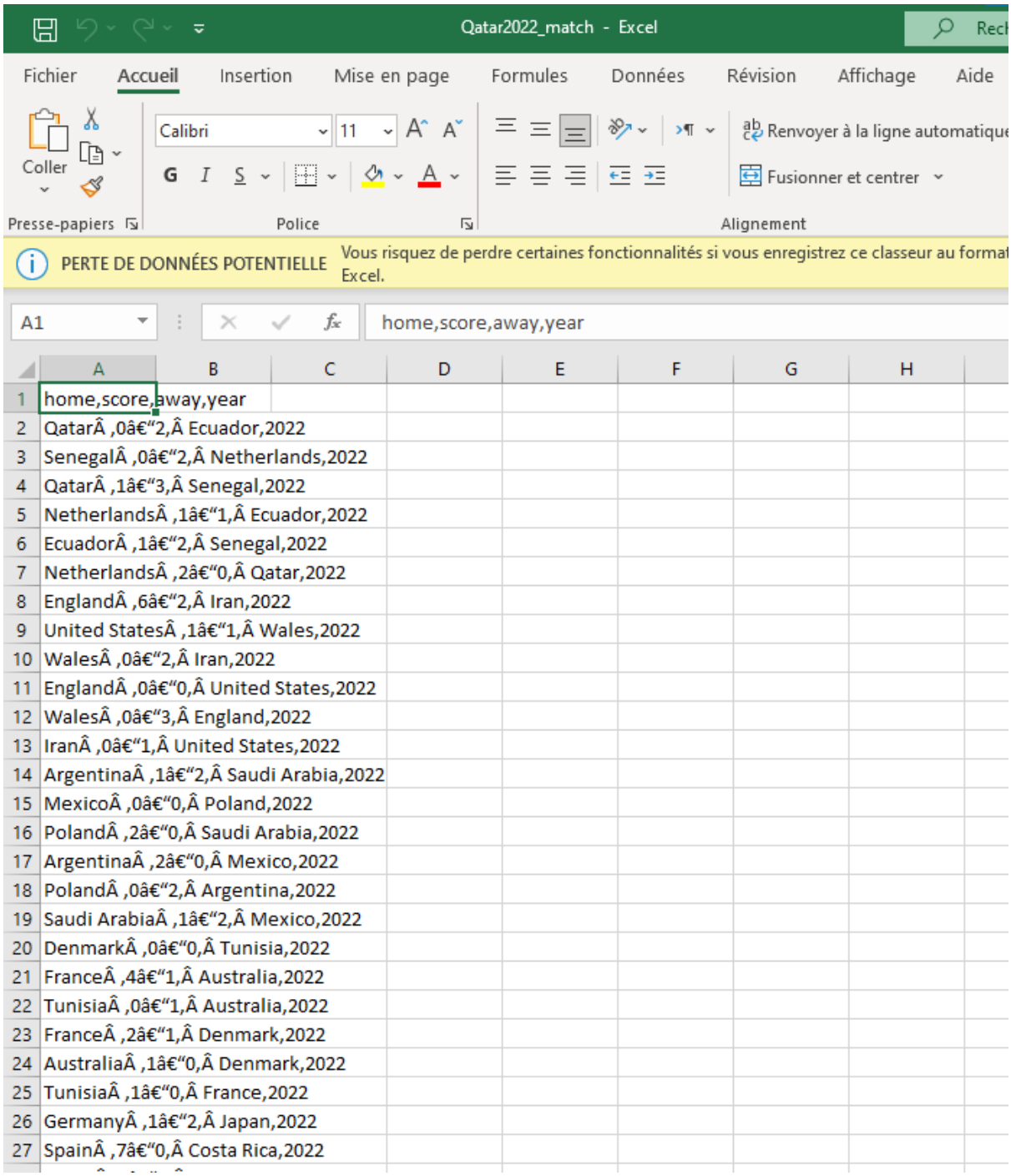
```python
Qatar2022 = get_matches(2022)
Qatar2022.to_csv('donnees/Qatar2022_match.csv', index=False)
```

Figure 4-Scapping des anciens matchs et des match du Qatar 2022



	A
1	home,score,away,year
2	FranceÂ ,4â€“1,Â Mexico,1930
3	ArgentinaÂ ,1â€“0,Â France,1930
4	ChileÂ ,3â€“0,Â Mexico,1930
5	ChileÂ ,1â€“0,Â France,1930
6	ArgentinaÂ ,6â€“3,Â Mexico,1930
7	ArgentinaÂ ,3â€“1,Â Chile,1930
8	YugoslaviaÂ ,2â€“1,Â Brazil,1930
9	YugoslaviaÂ ,4â€“0,Â Bolivia,1930
10	BrazilÂ ,4â€“0,Â Bolivia,1930
11	RomaniaÂ ,3â€“1,Â Peru,1930
12	UruguayÂ ,1â€“0,Â Peru,1930
13	UruguayÂ ,4â€“0,Â Romania,1930
14	United StatesÂ ,3â€“0,Â Belgium,1930
15	United StatesÂ ,3â€“0,Â Paraguay,1930
16	ParaguayÂ ,1â€“0,Â Belgium,1930
17	ArgentinaÂ ,6â€“1,Â United States,1930
18	UruguayÂ ,6â€“1, Yugoslavia,1930
19	UruguayÂ ,4â€“2,Â Argentina,1930
20	SpainÂ ,3â€“1,Â Brazil,1934
21	HungaryÂ ,4â€“2,Â Egypt,1934
22	SwitzerlandÂ ,3â€“2,Â Netherlands,1934
23	ItalyÂ ,7â€“1,Â United States,1934
24	CzechoslovakiaÂ ,2â€“1,Â Romania,1934
25	SwedenÂ ,3â€“2,Â Argentina,1934
26	AustriaÂ ,3â€“2 (a.e.t.),Â France,1934
27	GermanyÂ ,5â€“2,Â Belgium,1934

Figure 5- Le fichier CSV qui contient les anciens matchs



	A
1	home,score,away,year
2	QatarÂ ,0â€“2,Â Ecuador,2022
3	SenegalÂ ,0â€“2,Â Netherlands,2022
4	QatarÂ ,1â€“3,Â Senegal,2022
5	NetherlandsÂ ,1â€“1,Â Ecuador,2022
6	EcuadorÂ ,1â€“2,Â Senegal,2022
7	NetherlandsÂ ,2â€“0,Â Qatar,2022
8	EnglandÂ ,6â€“2,Â Iran,2022
9	United StatesÂ ,1â€“1,Â Wales,2022
10	WalesÂ ,0â€“2,Â Iran,2022
11	EnglandÂ ,0â€“0,Â United States,2022
12	WalesÂ ,0â€“3,Â England,2022
13	IranÂ ,0â€“1,Â United States,2022
14	ArgentinaÂ ,1â€“2,Â Saudi Arabia,2022
15	MexicoÂ ,0â€“0,Â Poland,2022
16	PolandÂ ,2â€“0,Â Saudi Arabia,2022
17	ArgentinaÂ ,2â€“0,Â Mexico,2022
18	PolandÂ ,0â€“2,Â Argentina,2022
19	Saudi ArabiaÂ ,1â€“2,Â Mexico,2022
20	DenmarkÂ ,0â€“0,Â Tunisia,2022
21	FranceÂ ,4â€“1,Â Australia,2022
22	TunisiaÂ ,0â€“1,Â Australia,2022
23	FranceÂ ,2â€“1,Â Denmark,2022
24	AustraliaÂ ,1â€“0,Â Denmark,2022
25	TunisiaÂ ,1â€“0,Â France,2022
26	GermanyÂ ,1â€“2,Â Japan,2022
27	SpainÂ ,7â€“0,Â Costa Rica,2022

Figure 6- Le fichier CSV qui contient les matchs du Qatar 2022

Étape 2 - Nettoyer les données des matchs et de leurs résultats

Les Data Scientists passent 60 % de leur temps à nettoyer et organiser les données.
Cela montre l'importance du nettoyage et du traitement des données dans un projet de Data Science.

utiliser les fonctions principales de Pandas pour nettoyer et transformer un ensemble de données qui sera utilisé dans la 3ème partie du projet Data.

→ 2 étapes en 1 : (Data Cleaning + Data Wangling)

In 12
```python
import pandas as pd
```

In 13
```python
match = pd.read_csv('donnees/Qatar2022_match.csv')
```

In 14
```python
match['score'] = match['score'].str.replace('[^\d-]', '', regex=True)
match['score']='-'
match['home'] = match['home'].str.strip() # clean blank spaces: Yugoslavia twice
match['away'] = match['away'].str.strip()

match
```

Out 14

11 rows 64 rows × 4 columns CSV

	home	score	away	year
0	Qatar	-	Ecuador	2022
1	Senegal	-	Netherlands	2022
2	Qatar	-	Senegal	2022
3	Netherlands	-	Ecuador	2022
4	Ecuador	-	Senegal	2022
...	...	...	...	...
59	England	-	France	2022
60	Argentina	-	Croatia	2022
61	France	-	Morocco	2022

In 15
```python
match.to_csv('donnees/Qatar2022_match_Clean.csv',index=False)
```

Figure 7- Cleanning des matchs du Qatar 2022

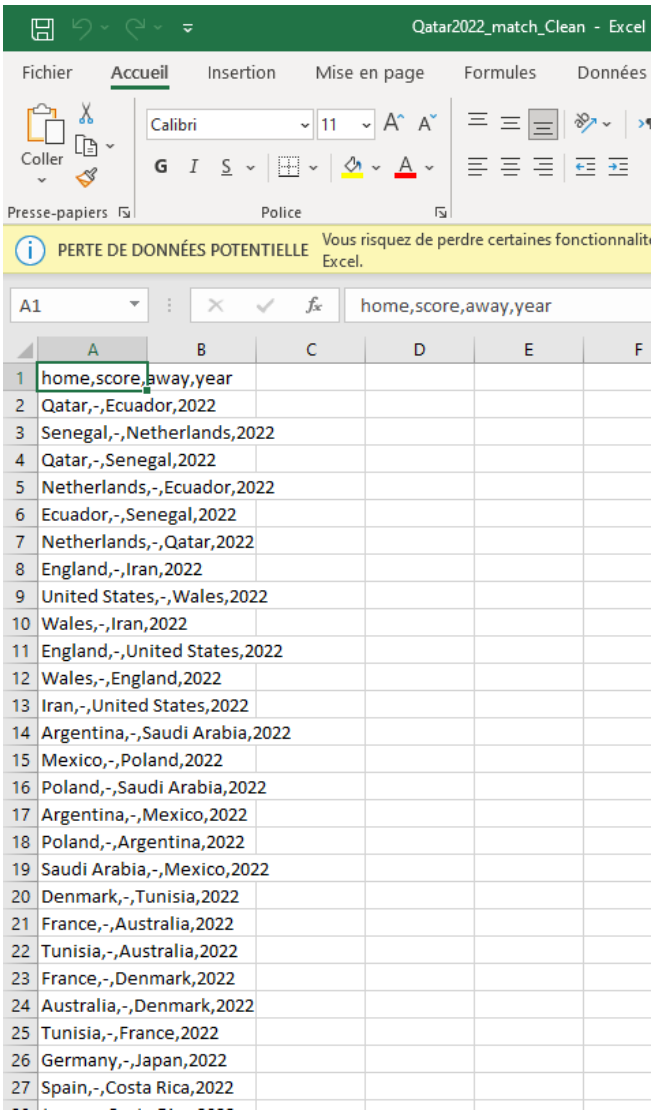


Figure 8-Affichage des données des matchs du Qatar après le nettoyage

```
In 1
import pandas as pd

In 2
match_coupdumonde = pd.read_csv('donnees/worldcup_match.csv')

In 3
match_coupdumonde.drop_duplicates(inplace=True)
match_coupdumonde.sort_values('year', inplace=True)

In 4
# nettoyer les colonnes
match_coupdumonde['score'] = match_coupdumonde['score'].str.replace('[^\d-]', '', regex=True)
match_coupdumonde['home'] = match_coupdumonde['home'].str.strip()
match_coupdumonde['away'] = match_coupdumonde['away'].str.strip()

# creation de deux nouveau colonnes et la supression de la colone score
match_coupdumonde[['HomeGoals', 'AwayGoals']] = match_coupdumonde['score'].str.split('-', expand=True)
match_coupdumonde.drop('score', axis=1, inplace=True)

# renommer et changer les types du colonnes
match_coupdumonde.rename(columns={'home': 'HomeTeam', 'away': 'AwayTeam',
                                  'year':'Year'}, inplace=True)
match_coupdumonde = match_coupdumonde.astype({'HomeGoals': int, 'AwayGoals':int, 'Year': int})

# creation de le colonne "totalgoals"
match_coupdumonde['TotalGoals'] = match_coupdumonde['HomeGoals'] + match_coupdumonde['AwayGoals']

In 10
match_coupdumonde.to_csv('donnees/worldcup_match_Clean.csv',index=False)
```

Figure 9- Cleaning des anciens matchs

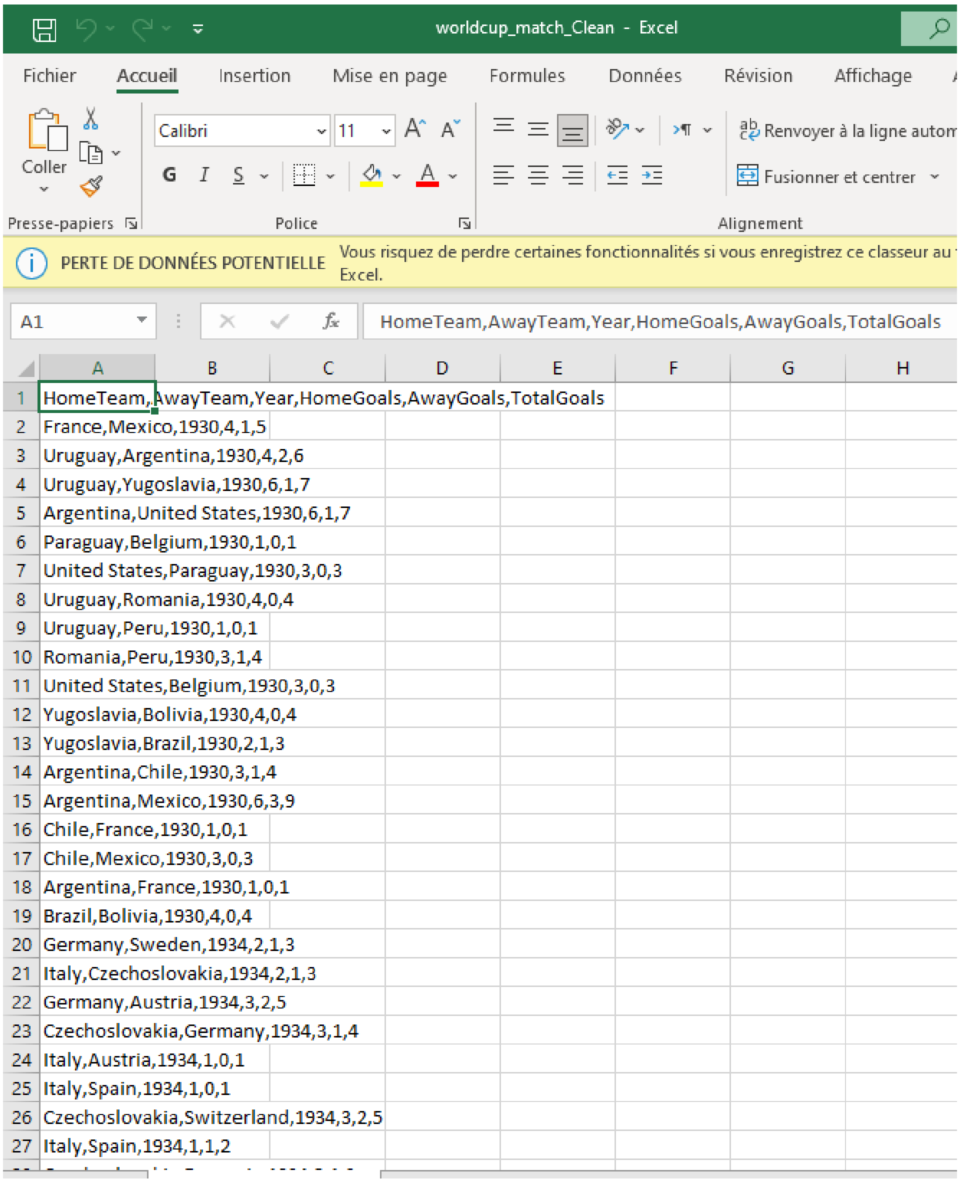


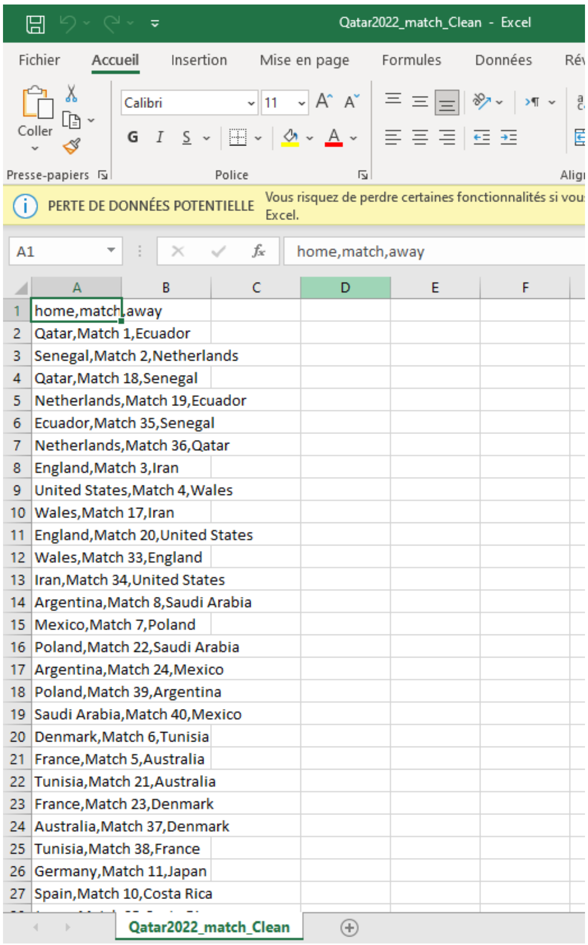
Figure 10- Affichage des données des anciens matchs après le nettoyage

Étape 3 - Prédire le résultat de chaque match jusqu’au vainqueur

Beaucoup de gens parlent du football comme d’un jeu imprévisible parce qu’un match de football comporte différents facteurs (arbitrage, blessures, cartons rouges, chance…) qui peuvent changer le score final.

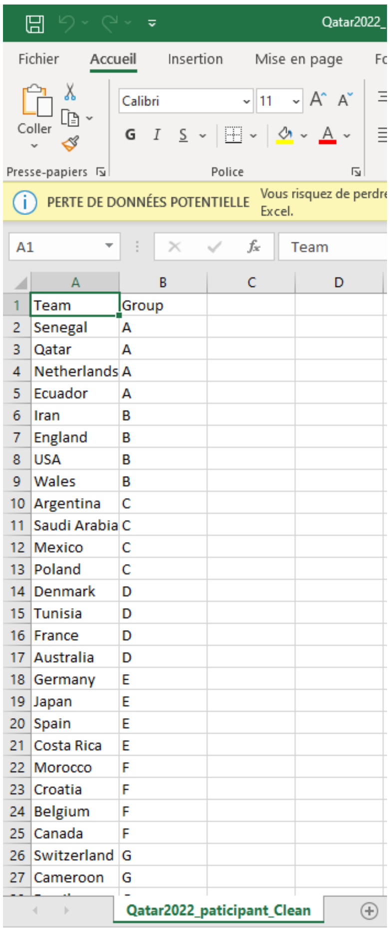
Il est difficile de prédire le score final ou le vainqueur d’un match, mais ce n’est pas le cas lorsqu’il s’agit de prédire le vainqueur d’une compétition.

Voilà les 4 fichiers dont on a besoin : les classements fifa, les équipes participants, les matchs internationaux et les matchs du Qatar 2022.



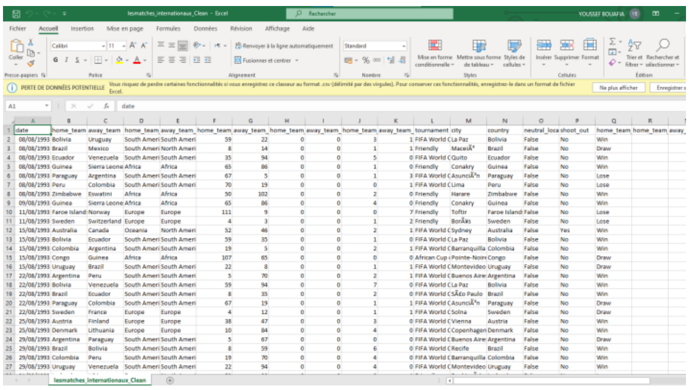
	A
1	home,match,away
2	Qatar,Match 1,Ecuador
3	Senegal,Match 2,Netherlands
4	Qatar,Match 18,Senegal
5	Netherlands,Match 19,Ecuador
6	Ecuador,Match 35,Senegal
7	Netherlands,Match 36,Qatar
8	England,Match 3,Iran
9	United States,Match 4,Wales
10	Wales,Match 17,Iran
11	England,Match 20,United States
12	Wales,Match 33,England
13	Iran,Match 34,United States
14	Argentina,Match 8,Saudi Arabia
15	Mexico,Match 7,Poland
16	Poland,Match 22,Saudi Arabia
17	Argentina,Match 24,Mexico
18	Poland,Match 39,Argentina
19	Saudi Arabia,Match 40,Mexico
20	Denmark,Match 6,Tunisia
21	France,Match 5,Australia
22	Tunisia,Match 21,Australia
23	France,Match 23,Denmark
24	Australia,Match 37,Denmark
25	Tunisia,Match 38,France
26	Germany,Match 11,Japan
27	Spain,Match 10,Costa Rica

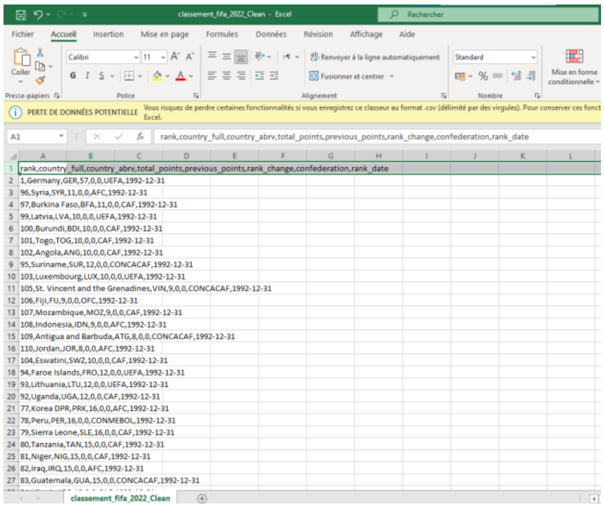
Qatar2022_match_Clean



	A	B
1	Team	Group
2	Senegal	A
3	Qatar	A
4	Netherlands	A
5	Ecuador	A
6	Iran	B
7	England	B
8	USA	B
9	Wales	B
10	Argentina	C
11	Saudi Arabia	C
12	Mexico	C
13	Poland	C
14	Denmark	D
15	Tunisia	D
16	France	D
17	Australia	D
18	Germany	E
19	Japan	E
20	Spain	E
21	Costa Rica	E
22	Morocco	F
23	Croatia	F
24	Belgium	F
25	Canada	F
26	Switzerland	G
27	Cameroon	G

Qatar2022_paticipant_Clean





```
In 125
import pandas as pd
import pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

In 126
match=pd.read_csv("data 2/Qatar2022_match_Clean.csv")
equipe = pd.read_csv("data 2/Qatar2022_paticipant_Clean.csv", sep=";")
classement = pd.read_csv("data 2/classement_fifa_2022_Clean.csv")
data_match = pd.read_csv("data 2/lesmatches_internationaux_Clean.csv", sep=";")
dict_table = pickle.load(open('data 2/dict_table','rb'))
```

Figure 11-Les bibliothèques importées

```
In 134
data_match_df['rank_difference'] = data_match_df['home_team_fifa_rank'] - data_match_df['away_team_fifa_rank']
data_match_df['average_rank'] = (data_match_df['home_team_fifa_rank'] + data_match_df['away_team_fifa_rank'])/2
data_match_df['point_difference'] = data_match_df['home_team_total_fifa_points'] - data_match_df['away_team_total_fifa_points']
data_match_df['is_stake'] = data_match_df['tournament'] != 'Friendly'
data_match_df['is_worldcup'] = 'FIFA World Cup' in data_match_df['tournament']
data_match_df['score_difference'] = data_match_df['home_team_score'] - data_match_df['away_team_score']
data_match_df['is_won'] = data_match_df['score_difference'] > 0 # Take draw as lost

In 135
A= data_match_df.loc[:,['average_rank', 'rank_difference', 'point_difference', 'is_stake', 'is_worldcup']]
B=data_match_df['is_won']
A_train, A_test, B_train, B_test = train_test_split(A, B, test_size=0.2, random_state=42)
```

Figure 12- Données divisées sur un nouveau colonne

```
In 142
# Support Vector Machines

svc = SVC()
svc.fit(A_train, B_train) #pour les variables d'entrainemenet
svm_pred = svc.predict(A_test)
accurrancy= round(svc.score(A_test, B_test) , 2)
accurrancy

Out 142
0.68

In 143
# Random Forest

random_forest = RandomForestClassifier(n_estimators=50)
random_forest.fit(A_train, B_train)  #pour les variables d'entrainemenet
Y_pred = random_forest.predict(A_test)
random_forest.score(A_train, B_train)
accurrancy = round(random_forest.score(A_test, B_test), 2)
accurrancy

Out 143
0.64
```

Figure 13- Les Accuracy des deux modèles

```
In 144
classement_df = classement_df.loc[(classement_df['rank_date'] == classement_df['rank_date'].max()) &
 (classement_df['country_full'].isin(equipe['Team'].unique()))]

classement_coup_du_monde = classement_df.set_index(['country_full'])
```

```
In 147
country_win_prob = {}
for country in equipe['Team'].unique():
    country_win_prob[country] = list()

country_win_prob
```

```
Out 147
  'Morocco': [],
  'Croatia': [],
  'Belgium': [],
  'Canada': [],
  'Switzerland': [],
  'Cameroon': [],
  'Brazil': [],
  'Serbia': [],
  'Uruguay': [],
  'South Korea': [],
  'Portugal': [],
  'Ghana': []}
```

Figure 14- Les équipes participants

```
def predict_points(home, away):

    # creer et initailiser les lignes et les column porte `average_rank', 'rank_difference', 'point_difference', 'is_stake',
     'is_worldcup
        ligne = pd.DataFrame(np.array([[np.nan, np.nan, np.nan, True, True]]), columns=A_train.columns)

        home_rank = classement_coup_du_monde.loc[home, 'rank']
        home_points = classement_coup_du_monde.loc[home, 'total_points']
        away_rank = classement_coup_du_monde.loc[away, 'rank']
        away_points = classement_coup_du_monde.loc[away, 'total_points']

        ligne['average_rank'] = (home_rank + away_rank) / 2
        ligne['rank_difference'] = home_rank - away_rank
        ligne['point_difference'] = home_points - away_points

        modele = svc.predict(ligne)

        awaypoints=0
        homepoints=0

        if modele == False:
            awaypoints = 3

        elif modele== True:
            homepoints= 3

        return homepoints,awaypoints
```

Figure 15- Fonction avec SVC pour la prédiction des points des groupes Stage

diviser les phases de la competition

```
In 149
phase_de_groupes = match[:48].copy()
huitième_de_finale = match[48:56].copy()
quarts_de_finale = match[56:60].copy()
demi_finale= match[60:62].copy()
final = match[62:].copy()
```

Figure 16- Les différentes phases de la compétition

```python
for group in dict_table:
    teams_in_group = dict_table[group]['Team'].values
    match_group = phase_de_groupes[phase_de_groupes['home'].isin(teams_in_group)]
    for index, row in match_group.iterrows():
        home, away = row['home'], row['away']
        if home=='United States'or away=='United States':
            points_home=0
            points_away=0
        else:
            points_home, points_away = predict_points(home, away)
        dict_table[group].loc[dict_table[group]['Team'] == home, 'Pts'] += points_home
        dict_table[group].loc[dict_table[group]['Team'] == away, 'Pts'] += points_away

    dict_table[group] = dict_table[group].sort_values('Pts', ascending=False).reset_index()
    dict_table[group] = dict_table[group][['Team', 'Pts']]
```

```python
dict_table
```

```
{'Group A':           Team  Pts
 0  Netherlands    9
 1      Senegal    3
 2    Qatar (H)    0
 3      Ecuador    0,
```

Figure 17- Prédiction des groupes Stage

```python
group_winner={}
for group in dict_table:
    group_winner[group] = dict_table[group].loc[0, 'Team'],dict_table[group].loc[1, 'Team']

group_winner
```

```
{'Group A': ('Netherlands', 'Senegal'),
 'Group B': ('England', 'Iran'),
 'Group C': ('Argentina', 'Mexico'),
 'Group D': ('France', 'Denmark'),
 'Group E': ('Spain', 'Germany'),
 'Group F': ('Belgium', 'Croatia'),
 'Group G': ('Brazil', 'Switzerland'),
 'Group H': ('Uruguay', 'Portugal')}
```

Figure 18- Affichage des gagnants du groupe Stage

```python
for group in dict_table:
    group_winner = dict_table[group].loc[0, 'Team']
    runners_up = dict_table[group].loc[1, 'Team']
    huitième_de_finale.replace({f'Winners {group}':group_winner,
                                f'Runners-up {group}':runners_up}, inplace=True)

huitième_de_finale['winner'] = '?'
huitième_de_finale
```

8 rows × 4 columns

	home	match	away	winner
48	Netherlands	Match 49	Iran	?
49	Argentina	Match 50	Denmark	?
50	France	Match 52	Mexico	?
51	England	Match 51	Senegal	?
52	Spain	Match 53	Croatia	?
53	Brazil	Match 54	Portugal	?
54	Belgium	Match 55	Germany	?
55	Uruguay	Match 56	Switzerland	?

Figure 19- Affichage des matchs du Huitième-final

```python
In 156
def predict_victory(home, away):

    # creer et initailiser les lignes et les column porte `average_rank', 'rank_difference', 'point_difference', 'is_stake',
     'is_worldcup
    ligne = pd.DataFrame(np.array([[np.nan, np.nan, np.nan, True, True]]), columns=A_train.columns)

    home_rank = classement_coup_du_monde.loc[home, 'rank']
    home_points = classement_coup_du_monde.loc[home, 'total_points']
    away_rank = classement_coup_du_monde.loc[away, 'rank']
    away_points = classement_coup_du_monde.loc[away, 'total_points']

    ligne['average_rank'] = (home_rank + away_rank) / 2
    ligne['rank_difference'] = home_rank - away_rank
    ligne['point_difference'] = home_points - away_points

    modele = random_forest.predict(ligne)

    if modele == False:
        victory = 'away'

    elif modele== True:
        victory = 'home'

    return  victory
```

Figure 20- Fonction de prédiction des phases éliminatoires

```python
def get_winner(match):
    for index, row in match.iterrows():
        home, away = row['home'], row['away']
        winn = predict_victory(home, away)
        if winn=='home':
            winner = home
        else:
            winner = away
        match.loc[index, 'winner'] = winner
    return match
```

Figure 21- Fonction qui implémente les gagnants des matchs à l'aide de la fonction de prediction predict_victory

```
In 158   get_winner(huitième_de_finale)
```

Out 158 — 8 rows × 4 columns

	home	match	away	winner
48	Netherlands	Match 49	Iran	Netherlands
49	Argentina	Match 50	Denmark	Argentina
50	France	Match 52	Mexico	Mexico
51	England	Match 51	Senegal	England
52	Spain	Match 53	Croatia	Spain
53	Brazil	Match 54	Portugal	Portugal
54	Belgium	Match 55	Germany	Belgium
55	Uruguay	Match 56	Switzerland	Switzerland

Figure 22- Affichage des gangnats du huitième-final

```
In 159
def update_table(round_1,round_2):
    for index, row in round_1.iterrows():
        winner = round_1.loc[index, 'winner']
        match = round_1.loc[index, 'match']
        round_2.replace({f'Winners {match}':winner}, inplace=True)
    round_2['winner'] = '?'
    return round_2
```

Figure 23-Afficher les équipes des matchs de la phase suivante

```
update_table(huitième_de_finale, quarts_de_finale)
```

4 rows × 4 columns

	home	match	away	winner
56	Spain	Match 58	Portugal	?
57	Netherlands	Match 57	Argentina	?
58	Belgium	Match 60	Switzerland	?
59	England	Match 59	Mexico	?

```
get_winner(quarts_de_finale)
```

4 rows × 4 columns

	home	match	away	winner
56	Spain	Match 58	Portugal	Portugal
57	Netherlands	Match 57	Argentina	Argentina
58	Belgium	Match 60	Switzerland	Belgium
59	England	Match 59	Mexico	England

Figure 24- Affichage des gagnants du quart-final

```
update_table(quarts_de_finale, demi_finale)
```

2 rows × 4 columns

	home	match	away	winner
60	Argentina	Match 61	Portugal	?
61	England	Match 62	Belgium	?

```
get_winner(demi_finale)
```

2 rows × 4 columns

	home	match	away	winner
60	Argentina	Match 61	Portugal	Portugal
61	England	Match 62	Belgium	Belgium

Figure 25- Affichage des gagnants du demi-final

Out 164

1 rows × 4 columns

	home	match	away	winner
62	Portugal	Match 64	Belgium	?

In 165

```
get_winner(final)
```

Out 165

1 rows × 4 columns

	home	match	away	winner
62	Portugal	Match 64	Belgium	Belgium

Figure 26- Affichage du vainqueur de la coupe du monde 2022