Group Project Description

Introduction to Computer Programming 2023/2024

Group Delegate ID: E00678

1 Dataset Description

For this group project, you will be working the European Soccer Database. The dataset contains some data about several European soccer leagues: players, teams, matches, bets, and so on.

For the sake of data handling, the dataset is split into 7 different `.csv` files, namely files whose data are organized in a table-like fashion where values are separated by comma[1].

In the following, we describe what each `.csv` file contains.

1.1 Description of `Country.csv`

`Country.csv` contains information about the considered European countries. Each row of this file corresponds to a different country. A snapshot of this file (coloured by columns for visualization purposes) is given in Figure 1:

[1] More info about CSV files can be found at `https://en.wikipedia.org/wiki/Comma-separated_values`

Figure 1: Snapshot of `Country`.

As you can see from the snapshot, each country is identified by the following two fields:

1. `id`: a unique numerical identifier for each country
2. `name`: the name of the country itself.

1.2 Description of `League.csv`

This file includes data about the considered leagues in each country. A snapshot of this file (coloured by columns for visualization purposes) is given in Figure 2:

Figure 2: Snapshot of `League`.

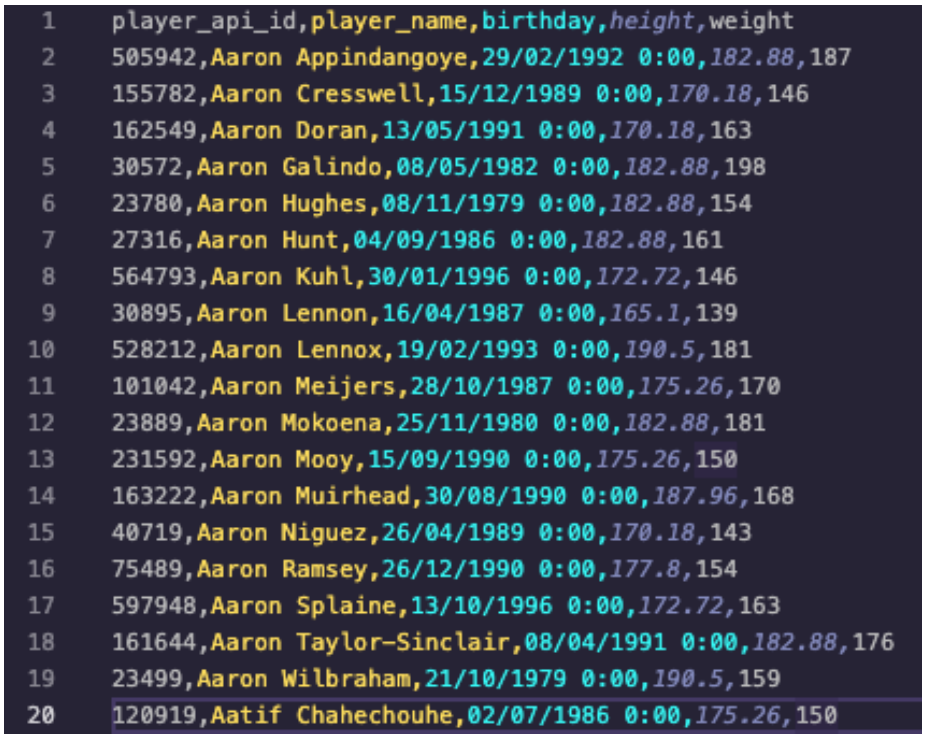
As you can see from the snapshot, each row (i.e., league) is identified by the following three fields:

1. `id`: a unique numerical identifier for each league
2. `country_id`: the identifier of the country that league belongs to
3. `name`: the name of the league

Note: For the sake of simplicity, for each country, we consider only its major league (e.g., the Serie A in Italy or the Premier League in England). For this reason, each country has only one league and the two IDs are the same.

1.3 Description of `Player.csv`

This file includes data about the players. A snapshot of this file (coloured by columns for visualization purposes) is given in Figure 3:



```
player_api_id,player_name,birthday,height,weight
505942,Aaron Appindangoye,29/02/1992 0:00,182.88,187
155782,Aaron Cresswell,15/12/1989 0:00,170.18,146
162549,Aaron Doran,13/05/1991 0:00,170.18,163
30572,Aaron Galindo,08/05/1982 0:00,182.88,198
23780,Aaron Hughes,08/11/1979 0:00,182.88,154
27316,Aaron Hunt,04/09/1986 0:00,182.88,161
564793,Aaron Kuhl,30/01/1996 0:00,172.72,146
30895,Aaron Lennon,16/04/1987 0:00,165.1,139
528212,Aaron Lennox,19/02/1993 0:00,190.5,181
101042,Aaron Meijers,28/10/1987 0:00,175.26,170
23889,Aaron Mokoena,25/11/1980 0:00,182.88,181
231592,Aaron Mooy,15/09/1990 0:00,175.26,150
163222,Aaron Muirhead,30/08/1990 0:00,187.96,168
40719,Aaron Niguez,26/04/1989 0:00,170.18,143
75489,Aaron Ramsey,26/12/1990 0:00,177.8,154
597948,Aaron Splaine,13/10/1996 0:00,172.72,163
161644,Aaron Taylor-Sinclair,08/04/1991 0:00,182.88,176
23499,Aaron Wilbraham,21/10/1979 0:00,190.5,159
120919,Aatif Chahechouhe,02/07/1986 0:00,175.26,150
```

Figure 3: Snapshot of `Player`.

As you can see from the snapshot, each row (i.e., player) is identified by the following five fields:

1. `player_api_id`: a unique numerical identifier for each player
2. `player_name`: its name
3. `birthday`: its birthday
4. `height`: its height
5. `weight`: its name

1.4 Description of `Team.csv`

This file includes data about the teams. A snapshot of this file (coloured by columns for visualization purposes) is given in Figure 4:

```
team_api_id,team_long_name,team_short_name
9987,KRC Genk,GEN
9993,Beerschot AC,BAC
10000,SV Zulte-Waregem,ZUL
9994,Sporting Lokeren,LOK
9984,KSV Cercle Brugge,CEB
8635,RSC Anderlecht,AND
9991,KAA Gent,GEN
9998,RAEC Mons,MON
7947,FCV Dender EH,DEN
9985,Standard de Liège,STL
8203,KV Mechelen,MEC
8342,Club Brugge KV,CLB
9999,KSV Roeselare,ROS
8571,KV Kortrijk,KOR
4049,Tubize,TUB
9996,Royal Excel Mouscron,MOU
10001,KVC Westerlo,WES
9986,Sporting Charleroi,CHA
9997,Sint-Truidense VV,STT
9989,Lierse SK,LIE
```

Figure 4: Snapshot of `Team`.

As you can see from the snapshot, each row (i.e., team) is identified by the following three fields:

1. `team_api_id`: a unique numerical identifier for each team
2. `team_long_name`: the name of the team (extended, full version)
3. `team_short_name`: the abbreviation of the team name

1.5 Description of `Match.csv`

This is the core file that contains information about each football match. Each row (i.e., match) is identified by the following 68 fields:

1. `country_id` and `league_id`: the identifiers for the country (hence, the league)

2. `season`, `stage` and `date`: the season (i.e., year) and stage for the match along with the exact date in which the match took place

3. `match_api_id`: a unique numerical identifier for the match

4. `home_team_api_id` and `away_team_api_id`: the unique numerical identifiers of the home team and the away team, respectively

5. `home_team_goal` and `away_team_goal`: the number of goals scored by the home team and the away team, respectively

6. `home_player_1` to `home_player_11`: the IDs of the players that played for the home team (initial squad only, substitutions not accounted for the sake of simplicity). Some values might be missing.

7. `away_player_1` to `away_player_11`: the IDs of the players that played for the away team (initial squad only, substitutions not accounted for the sake of simplicity). Some values might be missing.

8. `goal`: a string encoding a list-of-dictionaries that contains information about the goals scored during the match. Each dictionary contains the following keys: `elapsed`, `player1` and `team` that indicate the timestamp of the goal, the player ID who scored it and its team ID, respectively. All corresponding values are given as strings.[2]

9. `foulcommit`: a string encoding a list-of-dictionaries that contains information about the fouls committed during the match. It has the very same structure as `goal`

10. `card`: a string encoding a list-of-dictionaries that contains information about the cards given during the match. Each dictionary contains the following keys: `elapsed`, `player1`, `team` and `card_type` that indicate the timestamp of the card, the player ID who got it, its team ID and the card type (yellow `"y"`, second yellow `"y2"`, straight red `"r"`). All corresponding values are given as strings.[3]

11. `cross`: a string encoding a list-of-dictionaries that contains information about the crosses done during the match. It has the very same structure as `goal`

[2] If there is no such list-of-dictionaries, it means that the information was not available. Similarly, if the list-of-dictionaries contains just a dictionary with `"goal"` as key and `None` as value, then no goals were scored during the match.

[3] If there is no such list-of-dictionaries, it means that the information was not available. Similarly, if the list-of-dictionaries contains just a dictionary with `"card"` as key and `None` as value, then no cards were issued during the match.

12. `corner`: a string encoding a list-of-dictionaries that contains information about the corner kicks taken during the match. It has the very same structure as `goal`

13. `possession`: a string encoding a list-of-dictionaries that contains information about the ball possession between the two teams during the match. Each dictionary contains the following keys: `elapsed`, `awaypos`, `homepos` that indicate the percentage of ball possession for the away team (`awaypos`) and the home team (`homepos`) as measured after `elapsed` minutes. Some values might be missing.

14. the last 30 attributes indicate the betting odds for 10 different betting providers. For each provider we have the winning bet for the home team, the winning bet for the away team and the betting odds for a draw. For example, `B365H` indicates the betting odds for the home team as provided by the betting provider `B365`. Similarly, `B365A` and `B365D` indicate the betting odds for the away team and for a draw, respectively. Some values might be missing.

1.6 Description of `PlayerAttributes.csv`

This dataset includes statistical data about the players measured at different timestamps. Each row (i.e., player) is identified by the following 40 fields:

1. `player_api_id`: the ID of the player

2. `date`: the date in which the statistics were measured

3. `overall_rating` to `gk_reflexes`: the statistics of the player

1.7 Description of `TeamAttributes.csv`

Similarly to `PlayerAttributes.csv`, this dataset includes statistical data about the teams measured at different timestamps. Each row (i.e., team) is identified by the following 23 fields:

1. `team_api_id`: the ID of the team

2. `date`: the date in which the statistics were measured

3. `Speed` to `defenceDefenderLineClass`: the statistics of the team

2 Query #1

Write a Python script that calculates the teams that (within the same season) remained unbeaten at home matches. The resulting pickle file must contain a list with the name(s) of the teams. If a given team remains unbeaten for several different reasons, then it must appears as many times in the output list. More on pickle files later.

3 Query #2

Write a Python script that calculates the team that (within the same season) had the highest number of different goalkeepers in the initial squad. The resulting pickle file must contain a tuple with the name of the team and the season.

4 Query #3

Write a Python script that calculates the two teams that faced each other the most (counting both home and away matches). The resulting pickle file must contain a list with the name of the two teams.

5 Final Remarks

5.1 Downloading the Dataset

On the LUISS Learn homepage of the course, you will find a folder called "Group Project Dataset". Inside this folder, you can download all files.

5.2 Dataset Loading

You can load the dataset into Python thanks to the standard input/output functions on files, that is, by `open()`-ing the files and then running all known file methods such as `read()`, `readline()` and `readlines()`. Alternatively, you can use the `csv` module included in the Standard Python Library[4]. Once imported, you are free to organize those files into the Python data structure that you find most appropriate (e.g., list-of-lists, list-of-tuples, dictionaries, and so on).

As regards the list-of-dictionaries encoded as strings (see the `Match.csv` file), you can easily convert this string into a proper list of dictionaries by means of the `json.loads(s)` function that takes as input a string `s` that encodes either a dictionary or a list-of-dictionaries and returns the proper data structure for further manipulation in Python. Before using the `json.loads()` function, remember to `import` the `json` module (included in the Standard Python Library[5]).

5.3 Constraints

In principle you are free to choose your own approach to solve the three different queries. The only constraint follows: you are not allowed to use any third-party library. You can only use libraries and modules belonging to the Python Standard Library.

[4]Documentation available at `https://docs.python.org/3.11/library/csv.html`.
[5]Documentation available at `https://docs.python.org/3.11/library/json.html`.

5.4 On the Project Submission

The deadline for submitting the project is 2 days before the exam date. Details will be given on LUISS Learn in due time.

Submissions must include:

1. three Python source files: `query1.py`, `query2.py` and `query3.py` containing the source code for the three tasks in Sections 2, 3 and 4, respectively. Each Python file must start with the dataset loading and end with the call to the *pickle* file to be saved with the results;
2. three *pickle* files: `query1.pkl`, `query2.pkl`, `query3.pkl` containing the output of the three tasks in Sections 2, 3 and 4, respectively.

The submission must be done by sending the above material to

`amartino@luiss.it`

The subject of the e-mail must read as

`[Intro to CP 2023/2024] Group <ID> Project Submission`

where `<ID>` must be replaced with the student ID of the Group Delegate.

Keep in mind the following:

- each group is limited to one submission and any further submissions will automatically be ignored, so make sure that your submission contains the files that you indeed plan to submit;
- submissions must be done by the respective Group Delegates: any submission coming from different team members will automatically be ignored;
- the deadline is strict: any late submissions will automatically be postponed to the next trial.

If you are stuck and you feel difficulties in going on with your project, please send me an e-mail.