PYTHON AND R LAB PROJECT

ANIME DATABASE

During the first part of our Python and R lab course we have been asked to collect data from an API, to order them in a data frame and to convert it to a csv file. Out of all the possibilities we had we selected the MyAnimeList API since we like the topic and it gave us the impression of being a lot interesting.

We began by importing the necessary libraries for retrieving data from the Jikan.moe website, manipulating the json file, creating the data frame and converting it to a csv file.

In order to do so we initialized a list l2 and created an empty data frame with pandas, subsequently we created a list of values ranging from 0 to 3000 and we performed a for loop in that range for retrieving all the animes info one by one according to their IDs and append them to the empty list l2. Unfortunately, the IDs in the API missed some values, so in order to handle the value error we came up with a smart trick consisting in an if condition. If the response of the requests is = ‘404’ the code keeps executing and passes to the next value of the list completely ignoring the bad response of the API. We also included a line of code of code from the time library which is time.sleep() whose purpose is that of applying a delay that, according to the documentation of the API of Myanimelist, should be of 4 seconds.

Later we defined a function called create\_dataframe() that iterates through the elements of the list l2 (which basically are all json files) and retrieves that data needed and then appends them to the empty data frame. We decided to keep only the most important features, or the most useful for an analysis, that are reported in the table below. Since we got some KeyErrors due to missing values in some rows, we decided to handle those errors with try and except assigning to every KeyError a Null value called None.

At the end of this process, we noticed that among the 2814 rows retrieved, 2626 were valid while the other had None in each field. We checked the percentage of missing values for each variable, and noticed that only “aired\_to” has a big number of missing value, but this is normal since many anime have not finished airing or, as can be seen in the column “type”, many of them are movies or OAVs. The last function we had to implement is clean(), which drops the 188 invalid rows (identified as those with no “anime\_name” attribute) and looks for and drops duplicates rows.

Finally, we converted everything in a csv file called “anime project.csv” that we uploaded on Github. In the folder you can find the complete code in the notebook “Dataset\_creation”, while the other notebook was our first attempt with several differences. We also added a not executed python code, since we noticed that the notebook is not heavy and might have difficulties in loading in github.

|  |  |
| --- | --- |
| FEATURES | DESCRIPTION |
| anime\_name | Name of the anime |
| anime\_id | Id on the Myanimelist website |
| type | Typology of the anime (TV,Special, movie,OAV,…) |
| episodes | Number of episodes  (1 for OAVs and Movies) |
| score | Score assigned by the Myanimelist website |
| rank | Position in the Myanimelist ranking |
| popularity | How much popular is anime among the users (Numbers represent the position in a ranking of popularity) |
| scored\_by | Number of users who voted for that anime |
| likes | Number of likes from the users |
| aired\_from | The date in which the anime started airing |
| aired\_to | The date in which the anime finished airing |
| duration | Mean of the duration of an episode |
| Storyline | Plot of the anime |