System Configuration Elements

1. TMDB API Key –

For you to execute these codes, you will need to create your own account on TMDB and generate an API Key using which the code can extract the data from the repository.

Link - <https://www.themoviedb.org/documentation/api>

1. MongoDB as a NoSQL Database –

As the API of TMDB provides data in JSON Format, you will need to have a NoSQL Database to store this data. In this case, we have used MongoDB as our NoSQL solution. You will also need to create a user and password in MongoDB. Command line that you can use to create the user given in the code is below –

db.createUser( { user: "dapgrpl",

pwd: "dapgrpl",

roles: [ { role: "root", db: "admin" } ]

} )

1. PostgresSQL as an RDBMS solution –

Once a dataframe is created in the code, this is stored as a table in an RDBMS. We have chosen PostgresSQL as our RDBMS solution. Here as well, you will need a user to be created in postgresSQL. You can use the below command in postgres to create the user used in the code –

CREATE USER dapgrpl WITH PASSWORD 'dapgrpl' CREATEDB;

1. Google Chrome Browser –

As the visualizations created open in an html file, you will need a web browser to see and interact with these visualizations. We tried with default windows browsers like Internet Explorer and Edge and the interactions in the html files were misbehaving. Hence, we would urge you to install Chrome browser to see these visualizations without any hiccups.

Note – We haven’t tried with Safari browser and hence cannot infer on how the visuals would behave there.

1. Python –

As the code is written in python, you will need to download and install the latest version of python along with the below given packages in it using the “pip install” method or any other method you prefer.

|  |  |
| --- | --- |
| **Name** | **Purpose in our code** |
| requests | To extract data from the TMDB API |
| pymongo | For connecting and creating MongoDB Database, collections and its underlying documents |
| pandas | To create a dataframe and insert data into the PostgresSQL database |
| psycopg2 | To connect to the PostgresSQL Database |
| numpy | To perform transformations on the data |
| plotly | To create visualizations of the data |
| statistics | To do mathematical calculations using inbuilt formulas |
| json | To output the api data into a json file |
| nltk | To perform text analytics on the review data |
| afinn | To analyze and assign sentiment scores to reviews |
| scattertext | To create interactive visualization for authors and words in the review data |
| re |
| io |
| pprint |
| scipy |
| spacy |
| os |
| pkgutil |
| urllib |
| IPython |
| sklearn | To get eigen value matrix for review analysis |

1. Spyder or any other Python IDE for executing the code –

It will help to have any kind of Python IDE installed on your machine to have a better look at the code and to execute it easily in the python console.

1. Final checks –

Please verify and change the IP Address mentioned in the code to the IP Address of the machine on which you have installed MongoDB and PostgresSQL along with the port numbers if you have changed while installing these softwares. Also, if you are creating a user with a different user ID and password, be sure to change the same as well in the code before proceeding with the execution.

1. While Executing –

Execute the TV\_FULL and MOVIE\_FULL codes before executing the Movie\_Review and TV\_Review codes as the review codes are using some data from the descriptive data for movies and tv shows.