1st Capstone; Data Wrangling Steps.

Title: **“Exploring Netflix’s movie recommendation system”**

A preprocessed version of the Netflix Prize dataset can be downloaded from Kaggle, by using the following link: <https://www.kaggle.com/netflix-inc/netflix-prize-data/data>.

For the initial data wrangling steps, I used the “numpy”, “pandas”, “glob” and “csv” python libraries.

The data comes in the form of four .txt files, each with the following format:

Movie X:

Customer ID\_A, rating, date of rating

Customer ID\_B, rating, date of rating

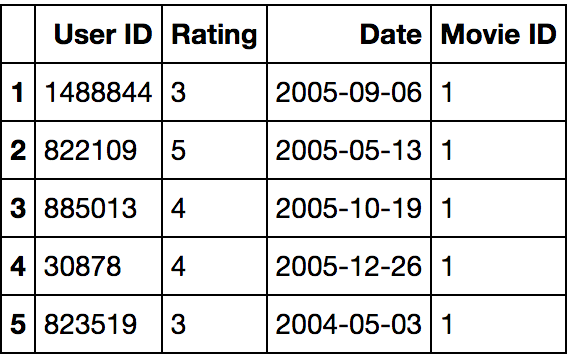
....

Movie Y:

Customer ID\_C, rating, date of rating

Customer ID\_D, rating, date of rating

My first step was to compile the 4 txt file into a single dataframe, where each row corresponds to 1 movie rating by one specific user. The columns of the resulting data frame are: “User ID”, ‘Rating”, “Date”, “Movie ID”. The following image shows how this dataframe looks like:

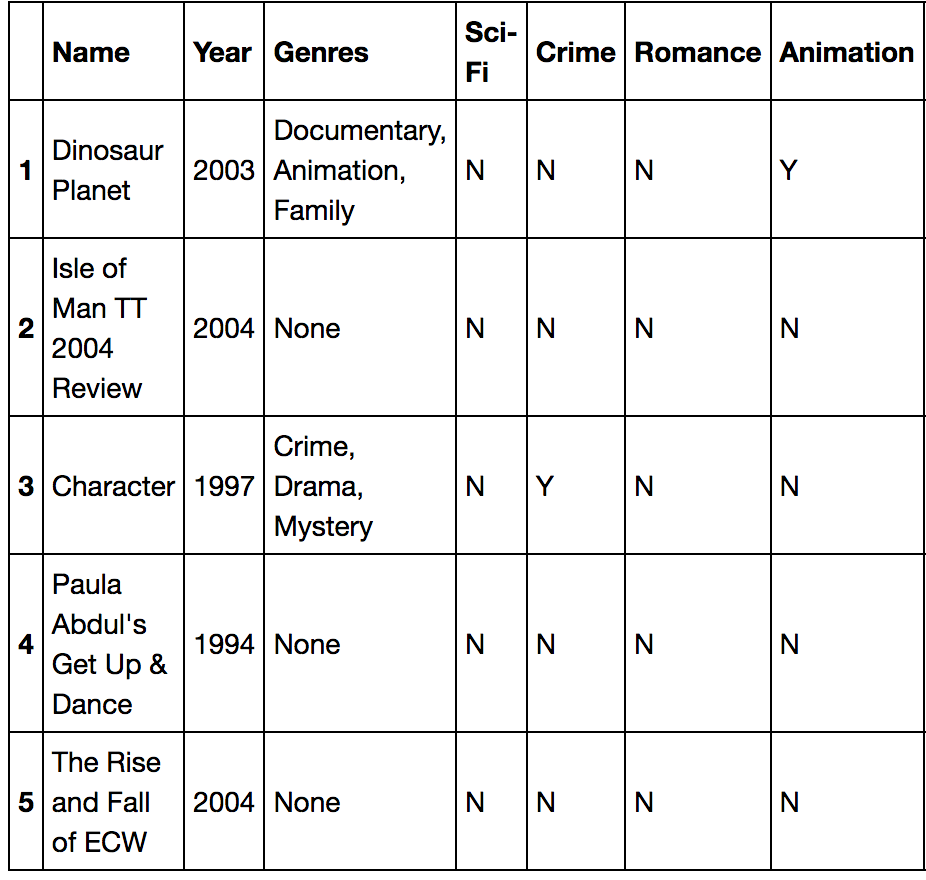


All of the steps up to this point are included in the “data\_processing.ipynb” notebook.

In addition, I attempted to acquire the genre to which each movie belongs. For this, I used the “omdb” python package, which allows performing queries to the “Open Movie Database” (<http://www.omdbapi.com/>). Unfortunately, this database does not contain information for all of the movies in the Netflix Prize data set. Regardless, I was able to retrieve the genre information for ~72 % of the all the movies (17770).

The steps required to perform this query are contained in the “movie\_omdb\_request.ipynb” notebook. Of note, the omdb api has a limit to the number of queries in a certain window of time, thus I had to re-start this script a few times.

The genre information retrieved from the omdb api came in the form of a single string, containing the respective genres. The “data\_processing\_2.ipynb” has the steps I used in order to extract/split this information. For subsequent data analysis, I opted to open the list of movie names (‘movie\_titles.csv’) and add one column for each available genre. For this columns, ‘Y’ means that the respective movie belongs to that genre, and ‘N’ that it doesn’t. The following image shows what the resulting data frame looks like:



In “data\_processing\_2.ipynb”, I also modified my original ratings data frame, so that includes the year in which he movie was release, as well a single string (as obtained from the omdb api) with the respective genres. The following image shows what the resulting dataframe looks like:

