EDA PROJECT
Netflix Titles Plot Popularity Detailed Analysis

study made by: Jorge García Navarro

1. Introduction

As part of the bootcamp's schedule at The Bridge Digital Talent, this EDA project was planned in order to put into practice all the knowledge acquired up until that stage.
The chosen topic was to analyze Netflix Titles Plots aiming to find the most popular words from each genre, creating a path to a future Machine Learning project.

1.1.Hypothesis

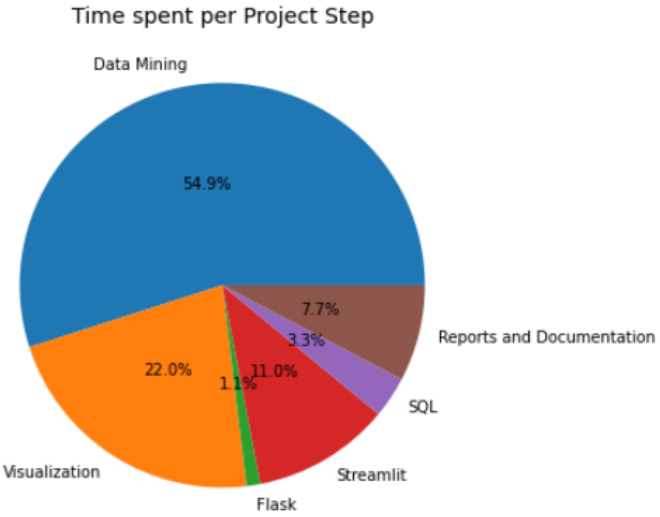
Without any previous knowledge about this certain topic, a vague hypothesis was made in order to establish an objective:

The genre Drama has the highest mean rating and its most popular words are: murder and vengeance.

It was later proven that this hypothesis was completely offset.

2. Project Steps

In this section there will be a detailed walkthrough of each phase which took place during the project's lifetime.



2.1.Data Mining

All data mining functionality is contained in the mining_data_tb.py file.
Classes: OmdbCleaner, NetCleaner, WordCleaner.

2.1.1.Data Sources

The data was extracted from three separate sites:

- Netflix user ratings Dataset: Netflix Prize data | Kaggle
- IMDb official site Dataset: (title.basics.tsv.gz) IMDb
- OMDb API: OMDb API - The Open Movie Database

Netflix user ratings Dataset

The data was divided into four 'combined.txt' files, each with a csv style format with a '\t' separator. Also it was separated by sections according to its own 'netflix_id'. Many transformations were made in order to convert each of them into a proper csv file and combine them into one single Dataframe.
Furthermore, it was grouped by its 'netflix_id' calculated by one side the total votes and its mean rating. It was later joined with the 'netflix_movies_titles.csv' file in order to add the movie titles associated with their 'netflix_id'.
The result of this was a Dataframe with the following columns: netflix_id, title, number_of_votes and netflix_rating.

IMDb official site Dataset

A simple dataset containing both movie title name and its associated imdb ID. Combined with the Netflix dataset it was possible to acquire the imdb ID of most of the Netflix titles. Many titles were lost due to missmatching titles in both datasets.

OMDb API

The source of the main data for analysis, this API contains all the title plots from IMDb.
An API key was obtained, but the only way to obtain the plot data was by making a title by title API request. There was a limit of a thousand free requests per day, so this process was the longest and most limiting.
In the end it was possible to obtain every Netflix title plot with an imdb ID, resulting in a ~8000 row Base Dataframe.
Had there been more time to develop this project, other routes would have been taken in order to acquire more data.

2.1.2.Cleaning the data

Once the BASE Dataframe was obtained, the development of new programs in order to clean the data started.
It was noted that the majority of titles had more than one Genre associated with them. In order to enable a genre by genre plot analysis, a method was made to 'expand' the Base Dataframe, unfolding every title by all its genres. This resulted in the duplication of many titles with the difference that each one of them had only one Genre associated.
Once the EXPANSE Dataframe was obtained, new methods were developed in order to extract all the words in every plot for each genre and generate a Dataframe with statistics of every word. Due to the large amount of words and many of them giving redundant or unuseful information, only nouns were extracted by the help of the nltk python library.
A Dataframe was returned and saved for every genre, containing nouns with a threshold of percentage occurrence of 5%. As part of the statistics, besides the percentage occurrence a mean rating was calculated (mean rating of every title containing a certain word in a select genre), a number of titles where the word appeared and a total count of word occurrence.

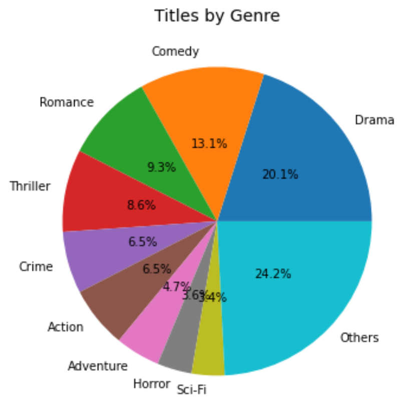
The joining of every Dataframe resulted in the WORD_STATS Dataframe, containing every noun word for every genre over the stated threshold.
An option was included to exclude from each Dataframe during visualization the most common words from all genres. A new percentage occurrence was calculated where words with a 30% threshold from each top 5 words were extracted. These words were: “new”, “young” and “life”. It was left as an option in order not to subjectivize the data study.
Cleaning the data was by far the most challenging, constantly optimizing the processing methods in order to reduce the computing time.

2.2. Visualization

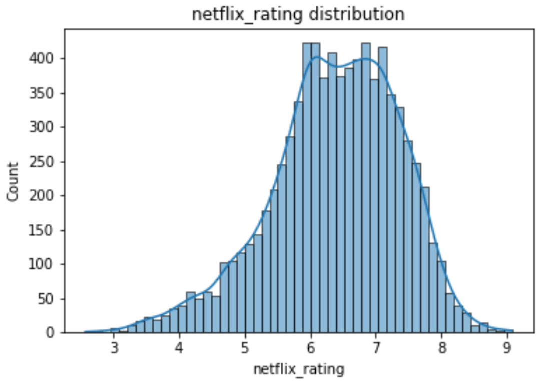
All data mining functionality is contained in the visualization_tb.py file.
Classes: Visualizer.

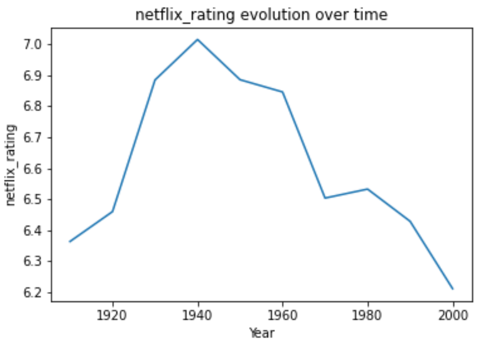
2.2.1.Base Dataframe and Expanse Dataframe

A Title distribution pie chart was made, which contributed to the understanding of each genre’s analysis fidelity.



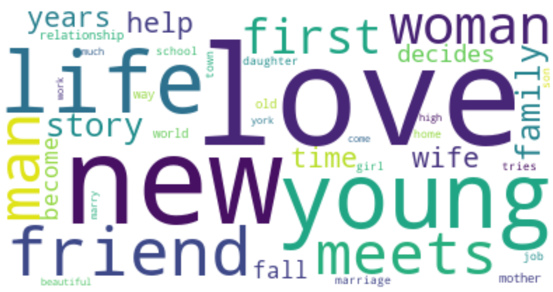
A density plot for each rating(Netflix, IMDb, Metascore) and a line plot for each rating based on the title’s decade. Options were included to select all data or a certain genre.



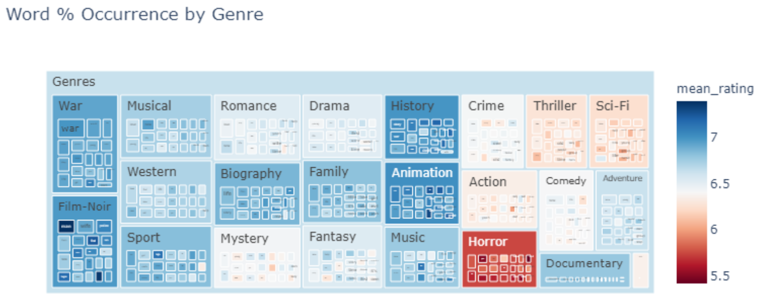


2.2.2.Word Stats Dataframe

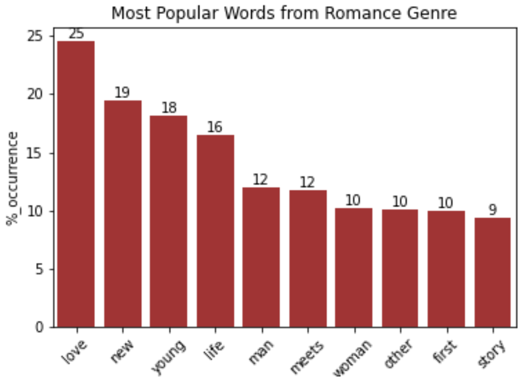
For a basic understanding of each genre’s word popularity, a word cloud was made with the use of the wordcloud python library.



For further information, an interactive Treemap containing all genres was made.



Additionally, a basic bar chart was made for each genre for a more specific observation.



2.2.3 Histograms and Heatmaps

In order to fulfill certain requirements, Histograms and Heatmaps were plotted. The heatmaps did not provide any useful information for this particular study.

2.3. Flask and Streamlit

2.3.1.Flask

A simple Flask API was developed, containing an endpoint which returned a json of the BASE Dataframe. It required a specific token_id as a ‘get’ parameter and argparse was included for command line execution.
All its functionality is contained in the server.py and apis_tb.py files.

2.3.2.Streamlit

Streamlit is a python library destined to represent in a simple and dynamic dashboard the Analyzed Data. A section for each type of visualization was made, including a request to the previously designed Flask API endpoint to retrieve the json data.
All its functionality is contained in the app.py and dashboard_tb.py file.
Classes: StreamFuncs.

3. MySQL Server

As another requirement, it was ordered to insert the BASE Dataframe into a MySQL Server. A method was made in order to automatically create and insert said Dataframe into the given server.
All its functionality is contained in the sql_tb.py file.
Classes: MySQL.

4.Conclusions

Part I

a. Was it possible to demonstrate the hypothesis? Why?
After many data transformations on the Base Dataset, it was possible to find an answer to the stated hypothesis. The analyzed data showed not only that the best rated genre was Film-Noir instead of Drama, but neither 'murder' nor 'vengeance' were in the top 20.

b. What can you conclude about your data study?
Based on the analyzed data it can be stated that the genre 'Film-Noir' has the highest rating and the most common words by %_occurrence are: 'new', 'young', 'life'.

c. What would you change if you need to do another EDA project?
The data mining should be more specific. There was not a clear goal at the beginning and valuable time was wasted on this particular task. Furthermore, the results would be more reliable if more data was gathered.

d. What did you learn doing this project?
To specify as much as possible the objective of the project and what is needed to reach that purpose.

Part II

a. Are there outliers or some rare data?
Based on the nature of the study on the collected data, there are no outliers to be considered.

b. What are the columns that have more repeated values?
Every row on the used Datasets is unique, so there should not be any repeated values.
There is an issue pending for future optimization were some similar titles returned the same IMDb ID when merged (eg. The Godfather parts II and III).