Netflix Data Analysis Report

First part

I. Description of the raw data file.

Columns in the unprocessed Netflix data is as follows:

1. “show_id” – Netflix uses this indexed column to identify movies and TV shows in the database in a sequential manner.
2. “type” – The database's video category, featuring movies and television shows as the most popular choices.
3. “title” – The titles of movies and television shows are listed in this column.
4. “director” – Director/s of movies/tv-shows a listed in this column
5. “cast” – The actors in the movies and shows
6. “country” – In terms of country, the origins of the movies and shows.
7. “date_added” – The date the movie/show was added to the Netflix database.
8. “release_year” – The year the movie/show was released.
9. “rating” – The rating of a movie/show.
10. “duration” – The number of seasons in a television show and the length of a movie.
11. “listed_in” – A movie could be a documentary, a comedy, or both in the movie/tv show category.
12. “description” – A short narrative of a movie/tv show.

By default, the other columns have object datatypes, except the "release year" column, which has an int64 data type.

II. Description of the computational steps

1. Preprocessing:

This step can be found in the “preprocessing.py” file. The file contains a **class** with the name “PreProcessData” with three arguments, “filename”, “index_column”, and “null_fill”.

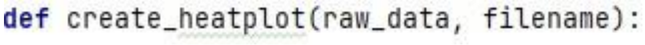
The preprocessing step starts by loading the data file as a “Pandas” dataframe, then continues by removing any duplication within the rows of the data.

The next stage deals with empty or null values within the data. The private method “_handle_null” within the **class** is responsible for handling this task. For the Netflix data, **null** values are replaced with a unique category, thus, “No item specified”. This value is later identified at the analysis stage. Generally, the method “_handle_null” is designed to either replace null entries with a unique value or delete rows containing null entries.

Finally, the cleaned data is written into a file stored as “netflix_titles_cleaned.csv”.

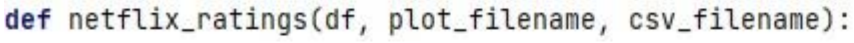
2. Analysis:

The **functions** and **classes** for the analysis stage can be found in a file called “analysis.py”. Most of the **functions** in this file require parameters like the “dataframe”, file naming “plot_filename, csv_filename” etc. All plots are handled by the “`_graph_plots`” function. Below is a description of the functions and classes created.



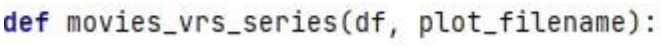
```
def create_heatplot(raw_data, filename):
```

The first step of the analysis stage with respect to the “workflow” has to do with a heat map plot of the distribution of **null values** within the data. “Seaborn” and “Matplotlib” python libraries were used for this plot and subsequent ones.



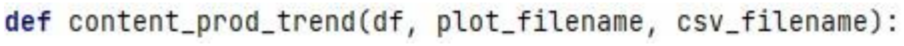
```
def netflix_ratings(df, plot_filename, csv_filename):
```

The goal of this function is to display the frequency of Netflix's ratings. In this function, data is extracted from the main data using the “**groupby(['rating'])**” command. The individual unique rating values are then counted and summed. This then creates a sub-dataframe with columns of “ratings” and “counts”. The function then proceeds to save the results as a table and a graph in a .csv and .pdf file, respectively.



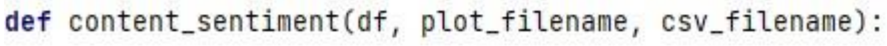
```
def movies_vrs_series(df, plot_filename):
```

This function compares the overall number of movies with the total number of television shows. The method slices the “type” column from the data then creates a pie chart of the different video types (Movies or TV shows). The matplotlib library was directly used for this task.



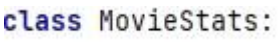
```
def content_prod_trend(df, plot_filename, csv_filename):
```

The goal here is to study the number of movies and tv shows released in a specified period of time. The function starts by slicing the “released_year” and “type” columns from the data. These columns are then renamed, grouped, and filtered to consider active years in the movie industry (thus, released years not less than 2010). A line plot was further produced with the final results along with a .csv file.



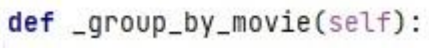
```
def content_sentiment(df, plot_filename, csv_filename):
```

In this function, we try to perform a **sentiment analysis** on each movie or tv show taking into account their **description** content. We start by considering the “release_year” and “description” columns. A loop is then constructed to run through each row of the selected columns. Within the loop, each description is handled and processed by a method “`TextBlob(descry)`” from a python library called **textblob,** this then returns an int64 data type. The returned value is then later used to show whether a description or statement is either positive or negative. The results are then plotted and the table saved in a .csv file.



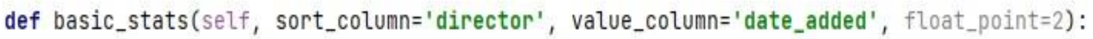
```
class MovieStats:
```

This class generally performs basic statistical operations like mean, standard deviation, and median on given columns in the data. It also creates distribution graphs of selected columns. Some of the major methods in this class are as follows.



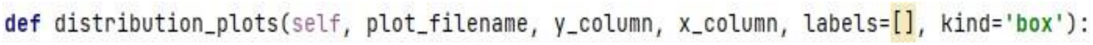
```
def _group_by_movie(self):
```

This method sorts the data and extracts rows with "Movies" as the type value. The “duration” column is then formatted and converted to an integer data type.



```
def basic_stats(self, sort_column='director', value_column='date_added', float_point=2):
```

This method contains lines of codes that groups data with reference to the method’s input arguments (“sort_column” and “value_column”). The second part of the method performs basic statistical operations like mean, median, and standard deviation on the given input parameters. The last part of the method saves the final results into a .csv file.



```
def distribution_plots(self, plot_filename, y_column, x_column, labels=[], kind='box'):
```

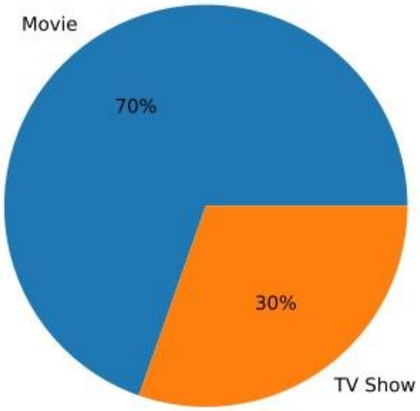
Here we perform a distribution plot of some selected columns from the input arguments “y_column” and “x_column”. The “kind” argument accepts parameters like “violin”, “box”, etc.

Second part

I. Main questions and answers

1. What is the number or percentage of movies against television shows in the Netflix database?

   Ans: The answer to this question can be found in the pie chart below. This shows the percentage of movies and tv shows. From the chart, 70% of the content is made up of movies while the other 30% covers the tv show.



2. In terms of the number of movies and tv shows released, which year was the most prolific for the film industry?

   Ans. From the plot and table shown in files “content_prod_trend.pdf” and “content_prod_trend.csv”, respectively, we notice that in 2017 the movie industry released close to 767 movies.

3. Which countries have the longest history of filmmaking?

   Ans: The answer to this question can be found in the distribution plot in the file “dist_plot_of_movie_released-yr_vrs_countries.pdf”. The ranking is United States, India, then the United Kingdom.

4. Is the average duration of an Indian movie higher than that of the United States?

   Ans: The plot in the file “dist_plot_of_movie_durations_vrs_countries.pdf” gives the information required to answer this question. And the answer is yes, thus, the average duration of an Indian movie is around 140 minutes and that for the United States is about 87 minutes.

5. What are the various Netflix rating systems?

   Ans: Answer is shown in the files “netflix_content_ratings.pdf” and “netflix_content_ratings.csv”

6. What is the average duration of a movie in which an actor like A.R. Rahman has appeared?

   Ans: The file “mean-std-median of cast-duration columns” has the statistics for this question. The mean column shows the average duration of each actor’s appearance in a movie. That for A.R Rahman is 87 minutes.

7. The significance of a narrative in a movie is determined by the power of its comment on the human condition. Are most narratives in movies and tv shows over the years dominated by positive or negative storylines?

   Ans: From the plot in the file “content_sentiment.pdf” we notice that most movies released over the years have positive narratives as compared to negative. This data was analyzed using movie descriptions.