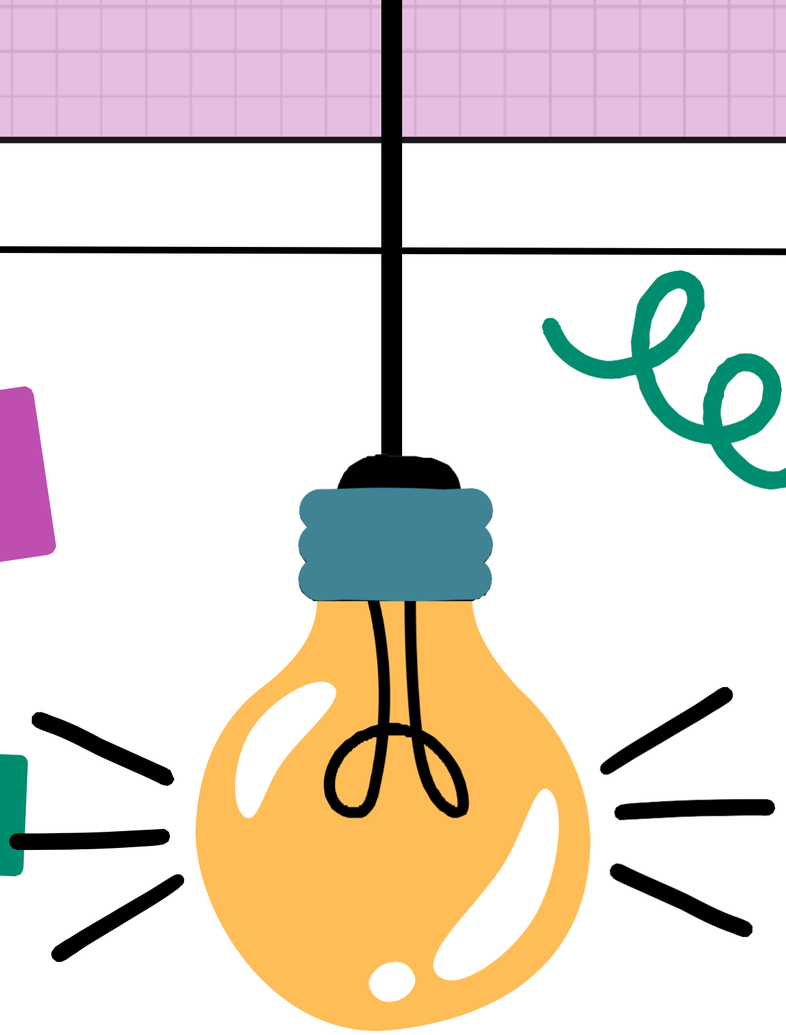
DATA ANALYZING OF AMAZON PRIME MOVIES AND TV SHOWS



Presented By :
Shaik Neelofar - AP21110010047

CONTENTS

INTRODUCTION

This data analysis project is focused on exploring the vast content of Amazon Prime Movies and TV Shows.

ABSTRACT

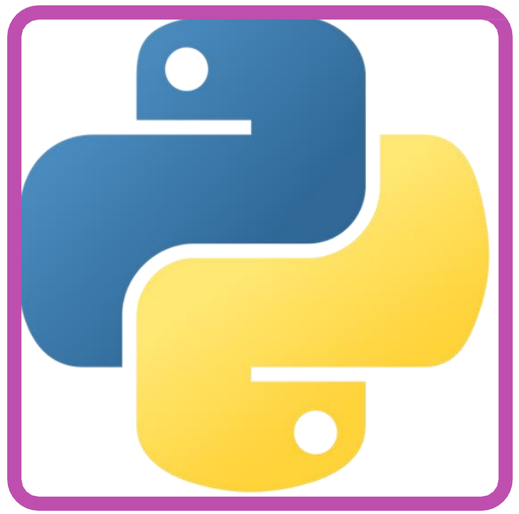
- The project utilizes a comprehensive dataset sourced from kaggle.com, comprising two CSV files: titles.csv and credits.csv.

- The primary objectives of this project include an in-depth analysis of various content aspects, such as genres, ratings, user preferences, and temporal trends.

- The project will provide a comprehensive report depicting key insights and findings of digital entertainment.

SYSTEM REQUIREMENTS

OS: Windows 7 and above



Python (version 3.10)

Jupyter Notebook

LIBRARIES USED

PANDAS:

- Used for data manipulation and analysis.
- Offers powerful data structures like DataFrames for handling structured data.
- Provides functionalities for filtering, grouping, and transforming data efficiently.

NUMPY:

- Used for numerical computations and array operations.

MATPLOTLIB:

- Used for creating basic static visualizations like line plots, bar charts, and histograms.

SEABORN:

- Built on top of Matplotlib, it enhances visualizations with minimal code.
- Provides attractive statistical graphics like scatter plots, box plots, and heatmaps.

Warnings:

- Used to suppress unnecessary warnings during the analysis and visualization process.
- Prevents warning messages from cluttering the output.

LIBRARIES USED

Plotly Express and Graph Objects:

- Used for creating interactive and visually appealing visualizations.

Missingno:

- Used for visualizing and understanding missing data patterns.
- Helps to identify missing values in the dataset and potential data quality issues.

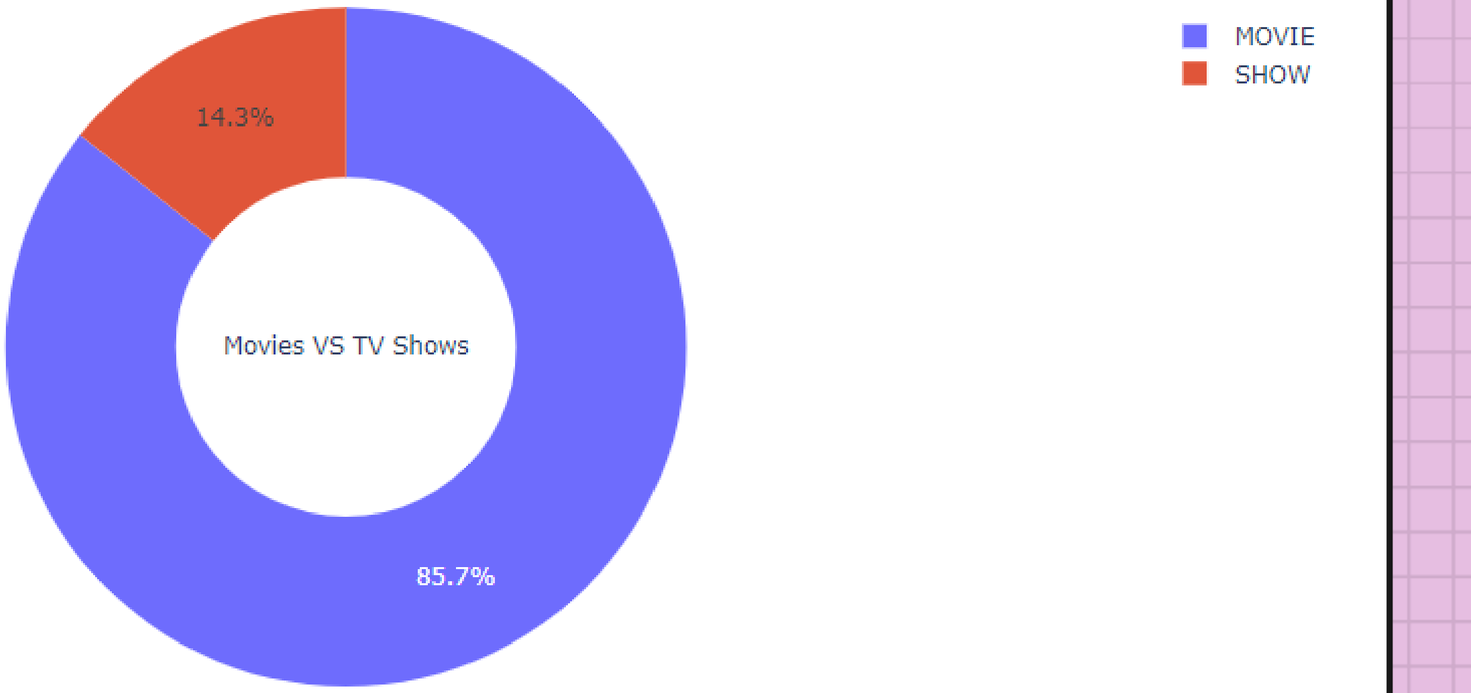
TextBlob:

- Used for text processing and sentiment analysis.
- Enables sentiment polarity detection (positive, negative, neutral) in textual data.

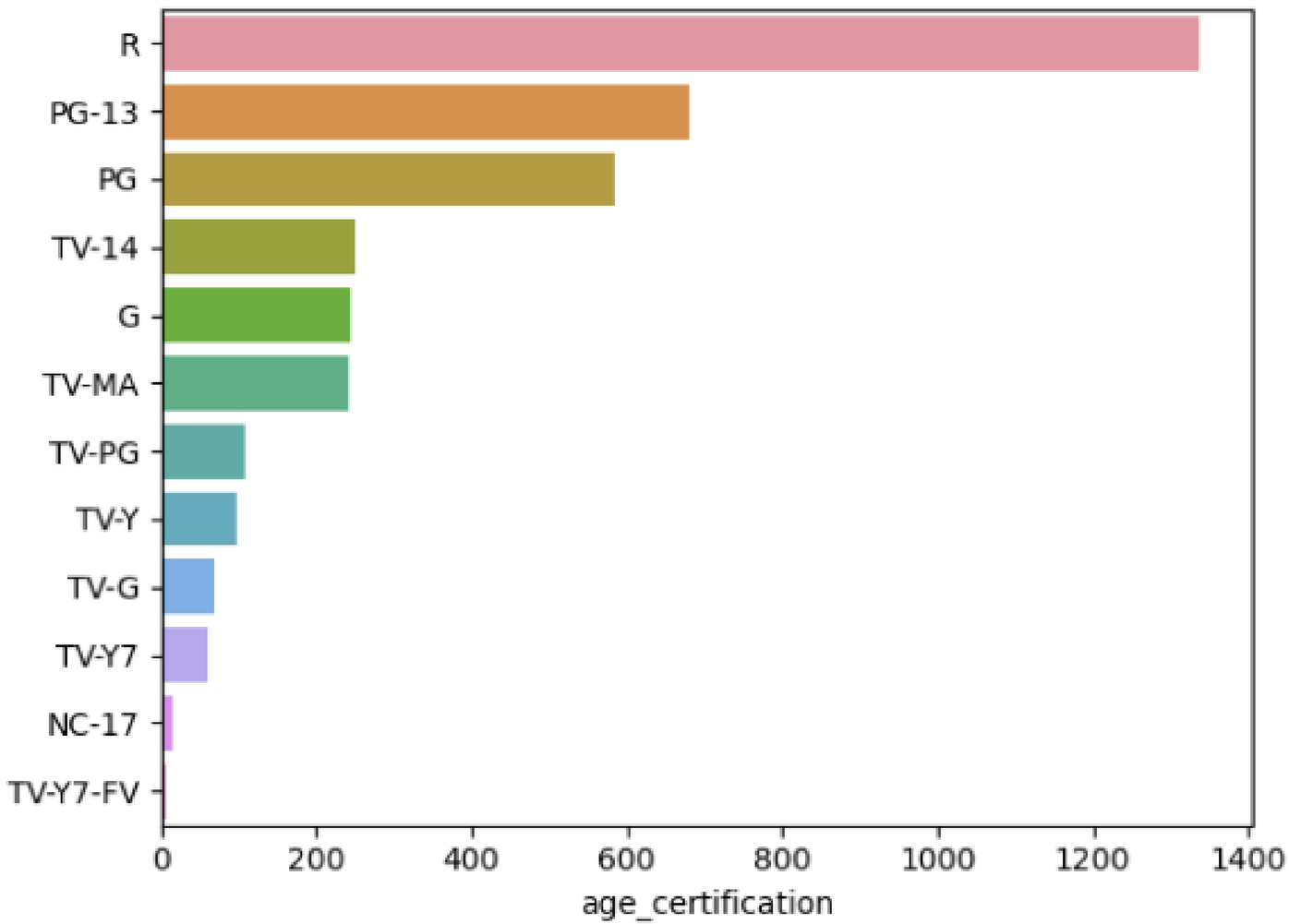
WordCloud:

- Used for creating word clouds, a popular and engaging visualization for text data.
- Presents word frequency in a graphical format, with larger words indicating higher frequency.

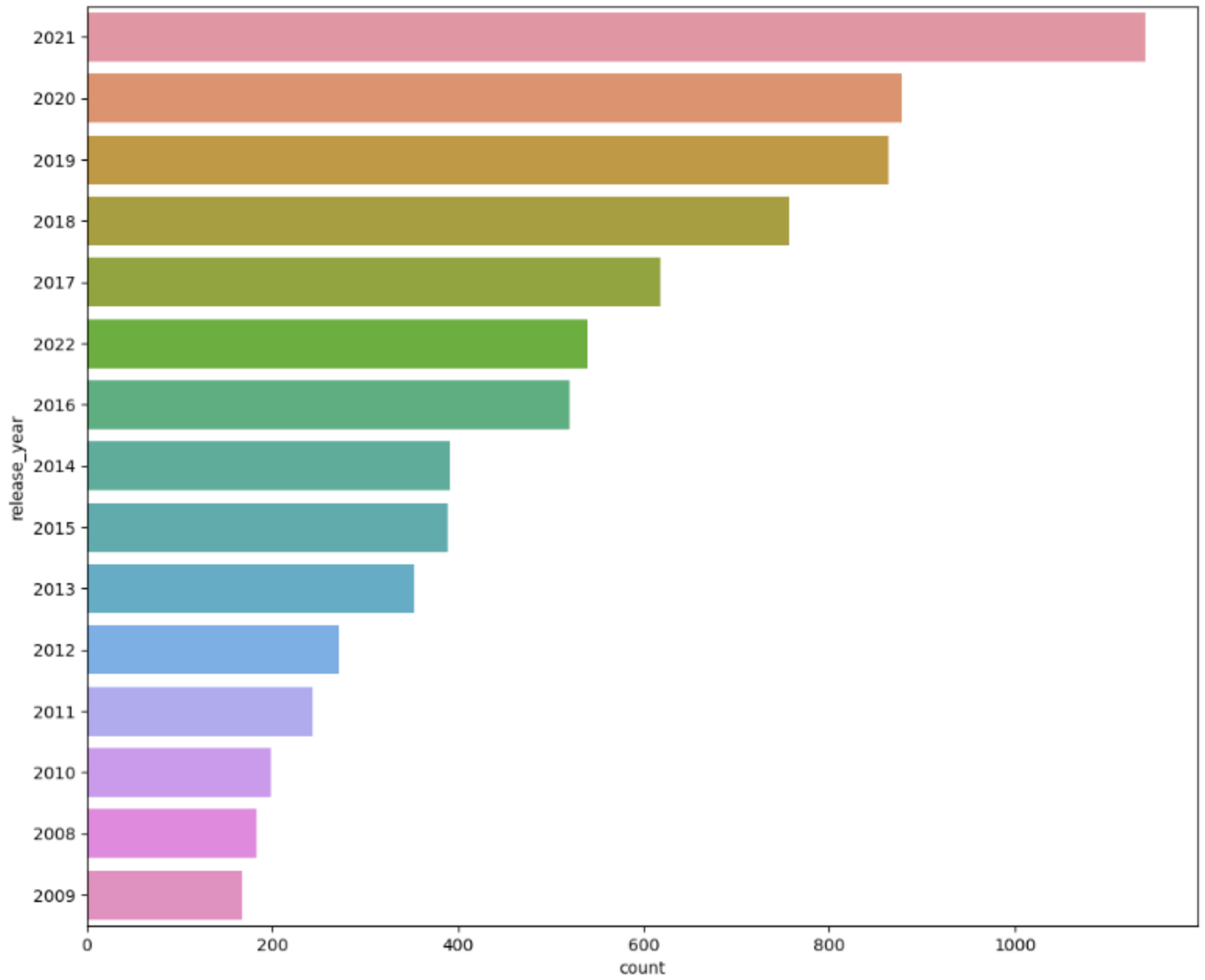
MOVIES VS TV SHOWS



AGE CERTIFICATION



TOP 15 YEAR WISE RELEASES



Top 10 actors

```
: 1 # Top 10 actors
  2 df2[df2["role"] == "ACTOR"]["name"].value_counts().head(10)
```

```
: George 'Gabby' Hayes    53
  Roy Rogers              48
  Bess Flowers            45
  Prakash Raj             41
  Nassar                  40
  Earl Dwire              36
  Herman Hack             35
  Eric Roberts            34
  George Morrell          34
  John Wayne              33
  Name: name, dtype: int64
```

Top 10 directors

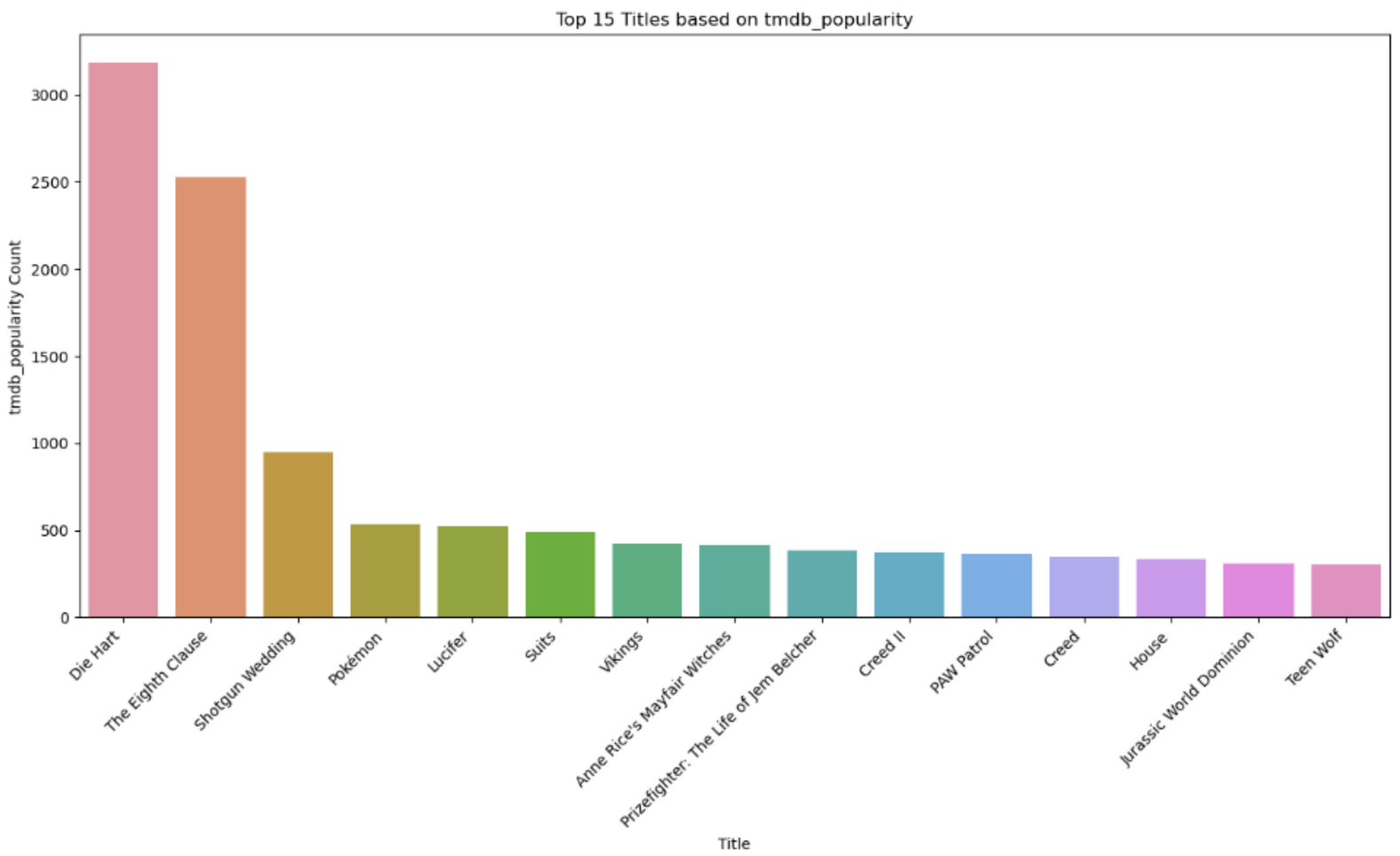
```
1 #Top 10 directors
2 df2[df2["role"] == "DIRECTOR"]["name"].value_counts().head(10)
```

```
Joseph Kane             45
Sam Newfield            38
Jay Chapman             33
Brian Volk-Weiss        27
Harry L. Fraser         20
William Nigh            20
Lesley Selander         19
Manny Rodriguez         19
Robert N. Bradbury      18
Robert F. Hill          13
Name: name, dtype: int64
```

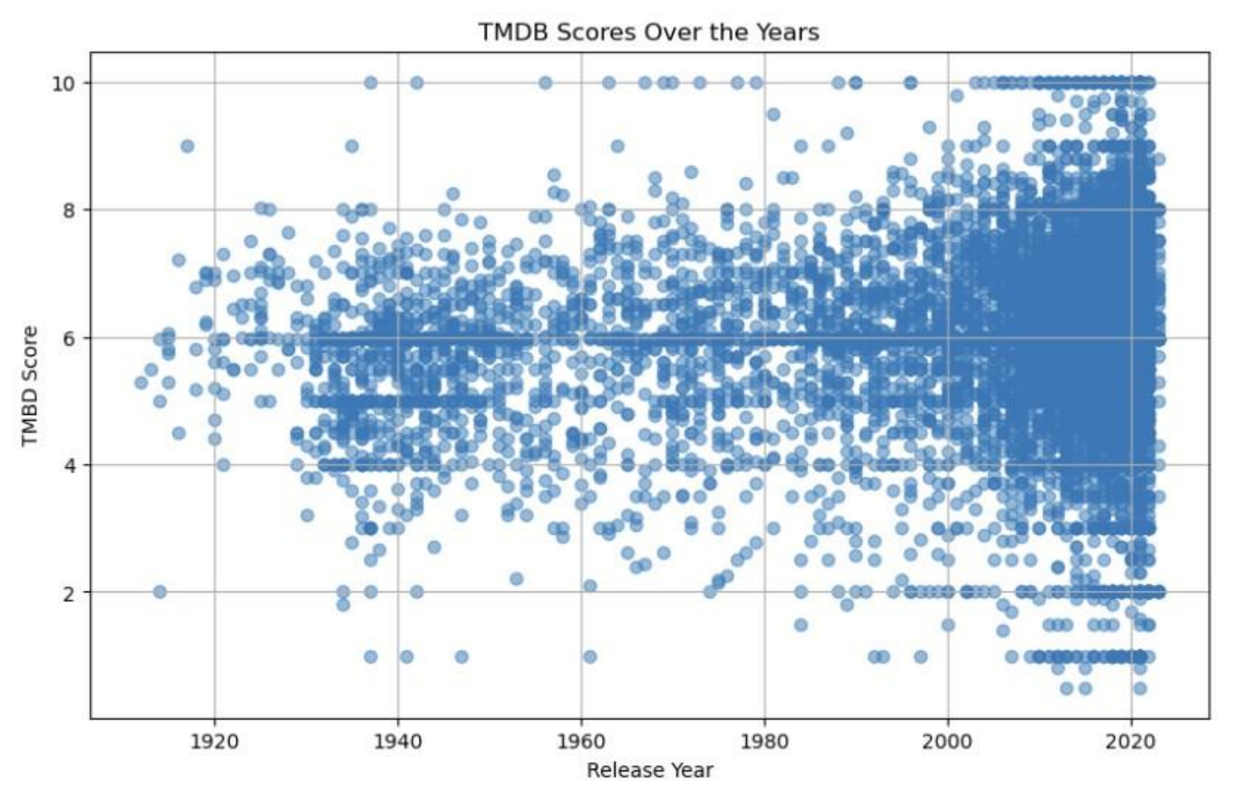
TOP 5 GENRES

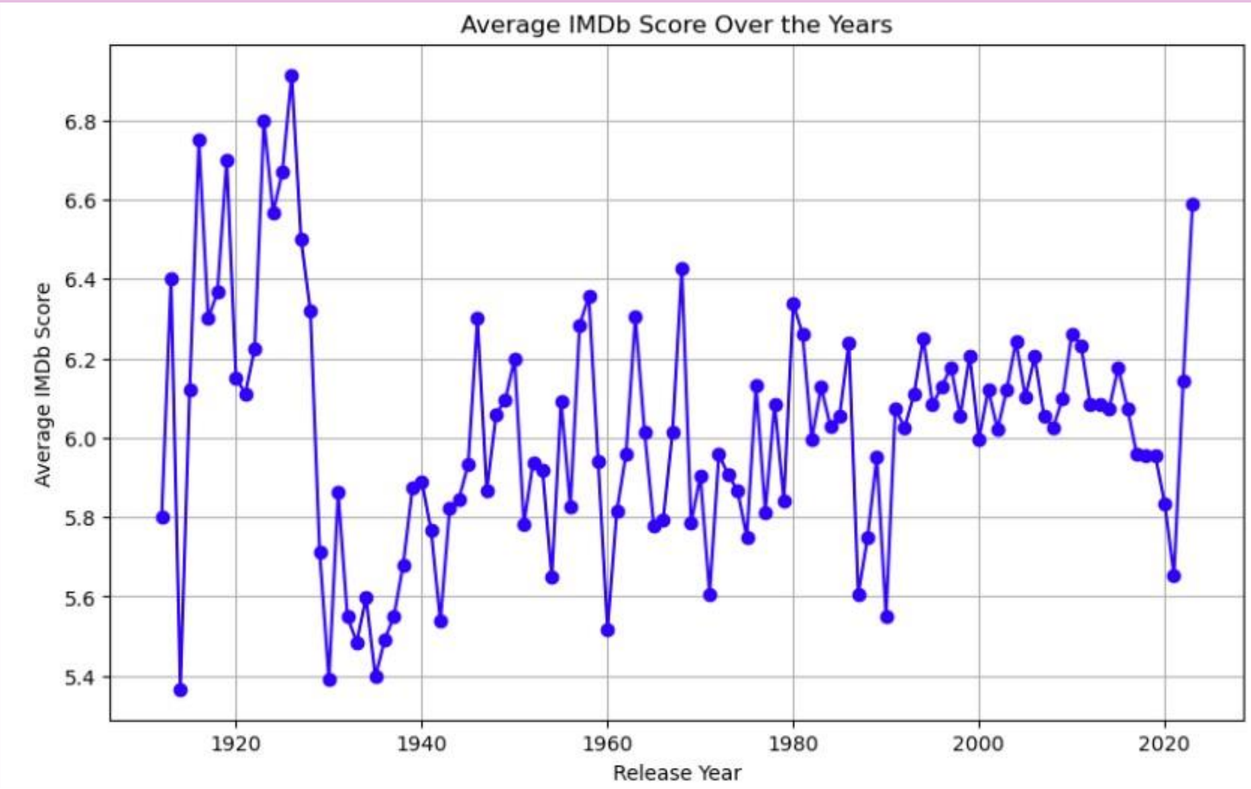
```python
# Top 5 genres
df1["genres"].value_counts().head(5)
```

```
['drama']                985
['comedy']               805
['documentation']        581
['drama', 'romance']     227
['horror']               200
Name: genres, dtype: int64
```


Top 15 Titles based on tmdb_popularity
tmdb_popularity Count
3000
2500
2000
1500
1000
500
0
Die Hart
The Eighth Clause
Shotgun Wedding
Pokémon
Lucifer
Suits
Vikings
Anne Rice's Mayfair Witches
Prizefighter: The Life of Jem Belcher
Creed II
PAW Patrol
Creed
House
Jurassic World Dominion
Teen Wolf
Title

TMDb and IMDb Scores

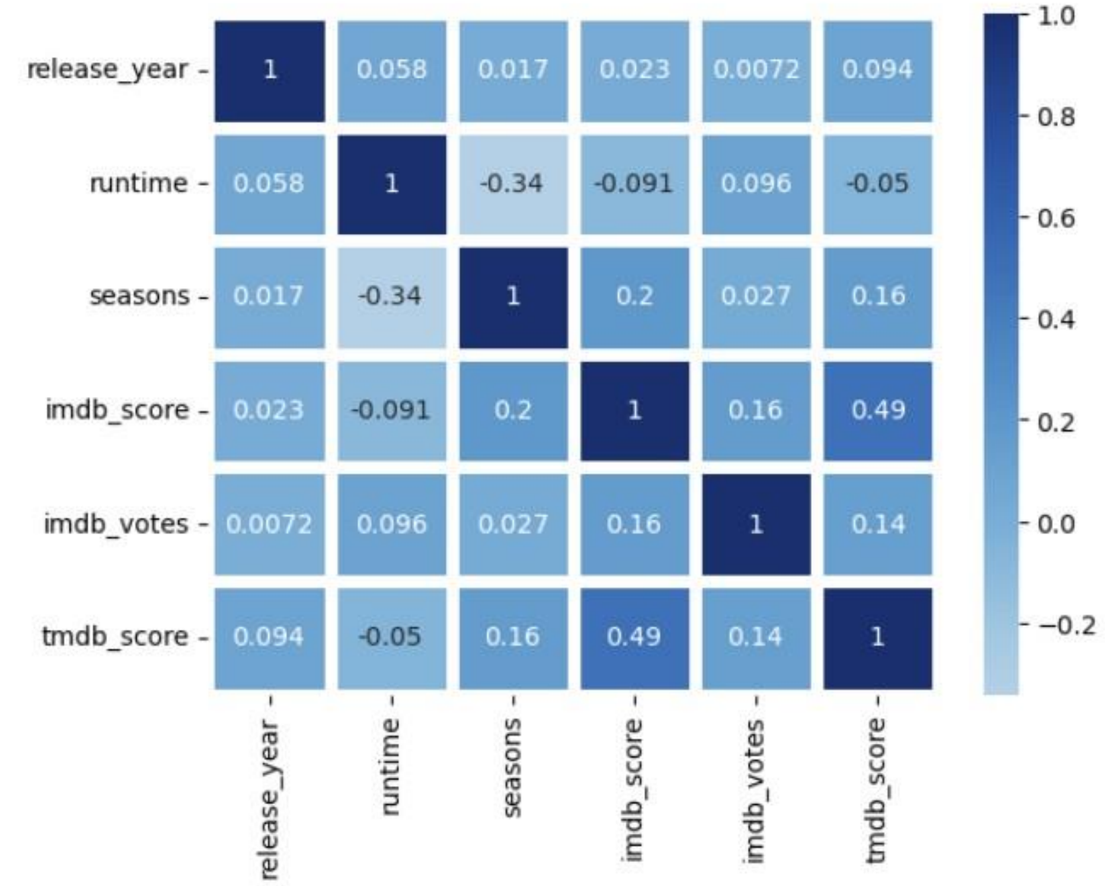




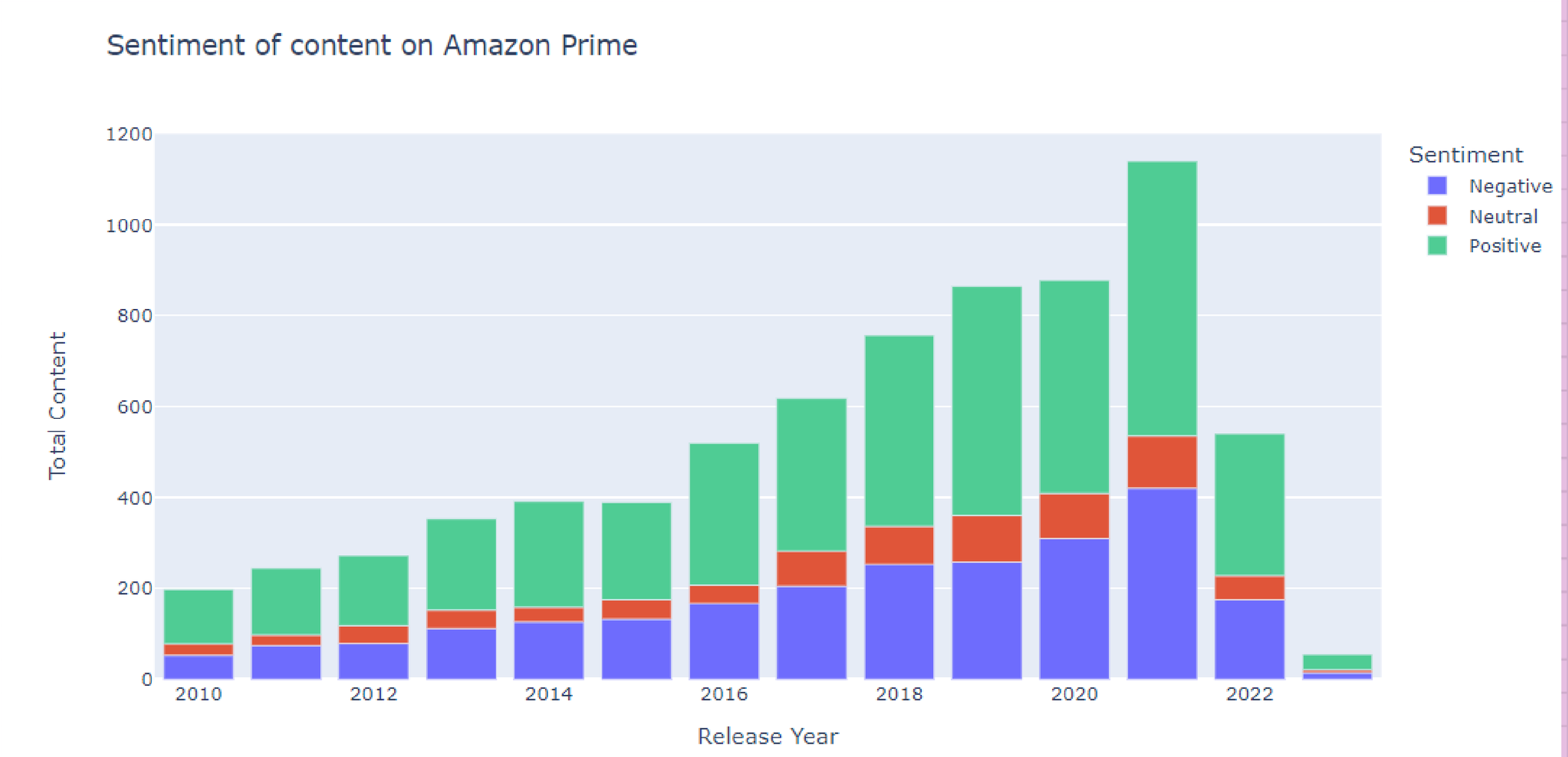
HEATMAP

```
1 sns.heatmap(df1.corr(),center=0,annot=True,linewidth=5,cmap='Blues')
```

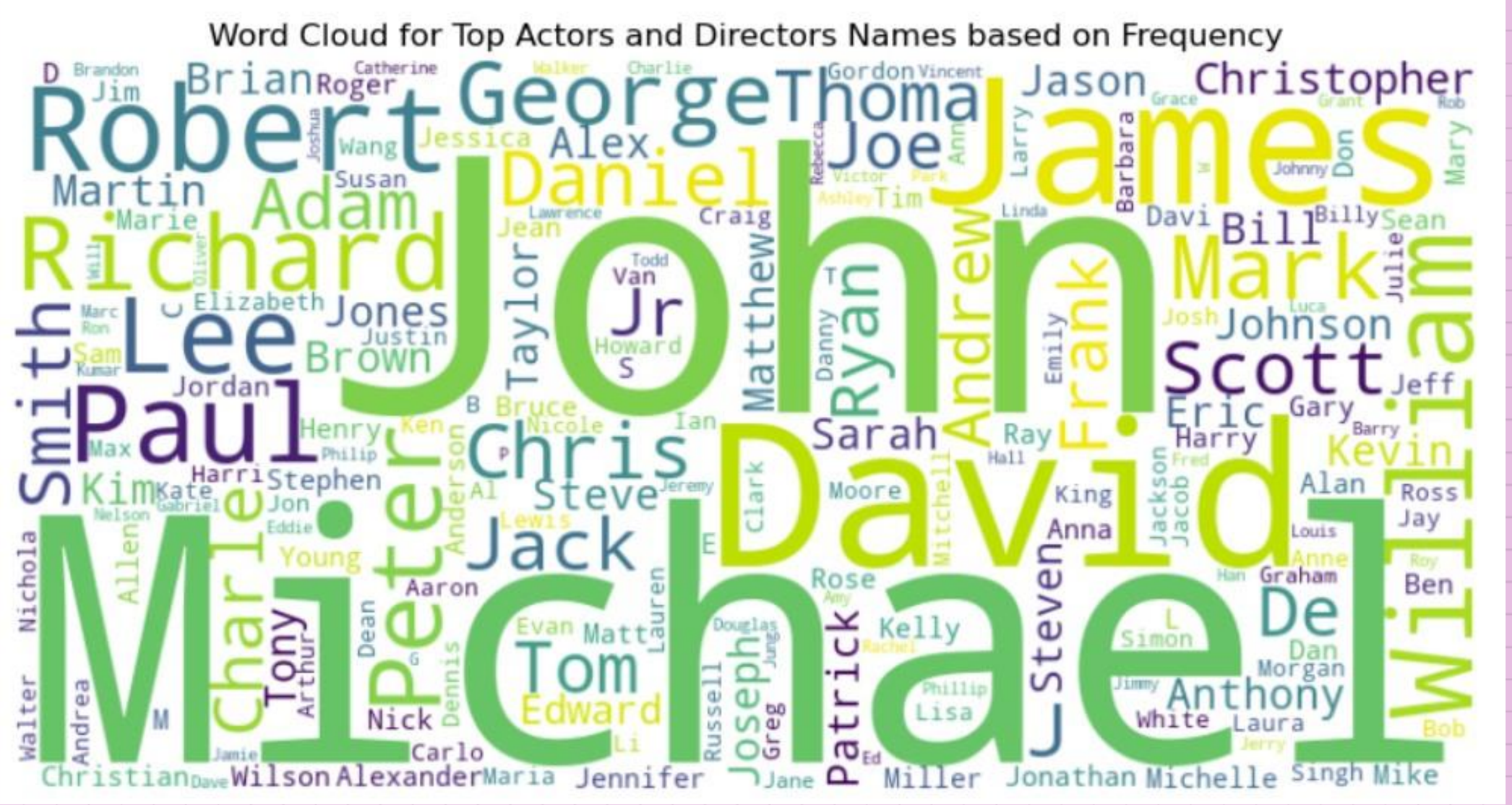
<AxesSubplot:>



SENTIMENT ANALYSIS



WORDCLOUD FOR TEXT VISUALIZATION



LONGEST AND SHORTEST MOVIES

```
#shortest Movie
shortest_movie=movies_df.loc[(movies_df['runtime']==np.min(movies_df.runtime))]
shortest_movie
```

	id	title	type	description	release_year	age_certification	runtime	genres	production_countries	seasons	imdb_id	imdb_score	imdb_vot
5966	tm414289	Two Wrongs	MOVIE	A short film about revenge and gangs in the st...	2016	Not Available	4	['action', 'thriller', 'drama']	['US']	0.0	Not Available	5.970558	8973.2317

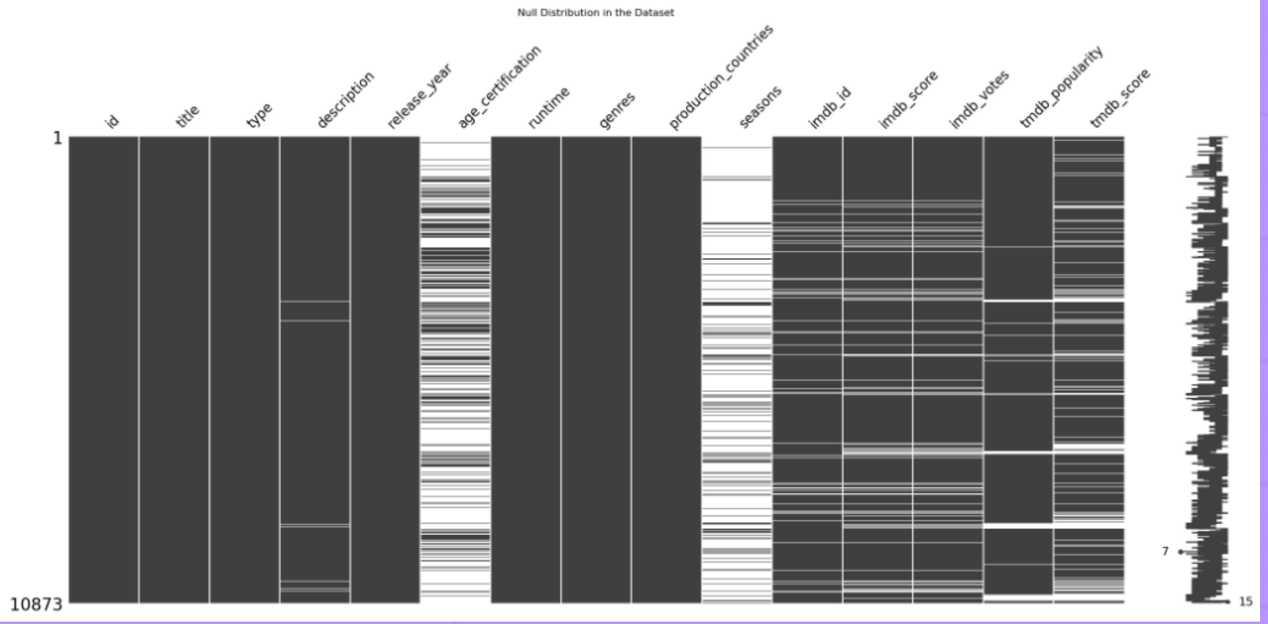
```
#Longest Movie
longest_movie=movies_df.loc[(movies_df['runtime']==np.max(movies_df.runtime))]
longest_movie
```

	id	title	type	description	release_year	age_certification	runtime	genres	production_countries	seasons	imdb_id	imdb_score	imdb_vot
1734	tm127342	Once Bitten	MOVIE	Mark wants to lose his virginity, but his girl...	1985	PG-13	940	['comedy', 'horror', 'drama', 'fantasy']	['US']	0.0	tt0089730	5.5	1538

Before handling null values

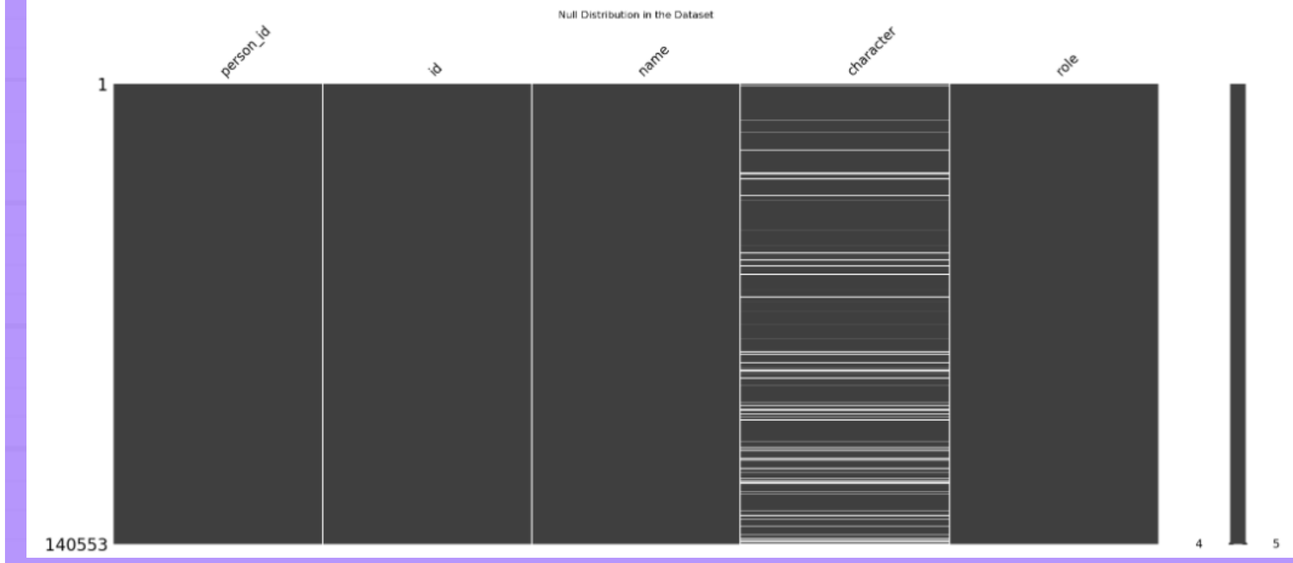
Inspecting Missing Values in the Dataset of Titles

```
1 msno.matrix(df1)
2 plt.title('Null Distribution in the Dataset')
3 plt.show()
```



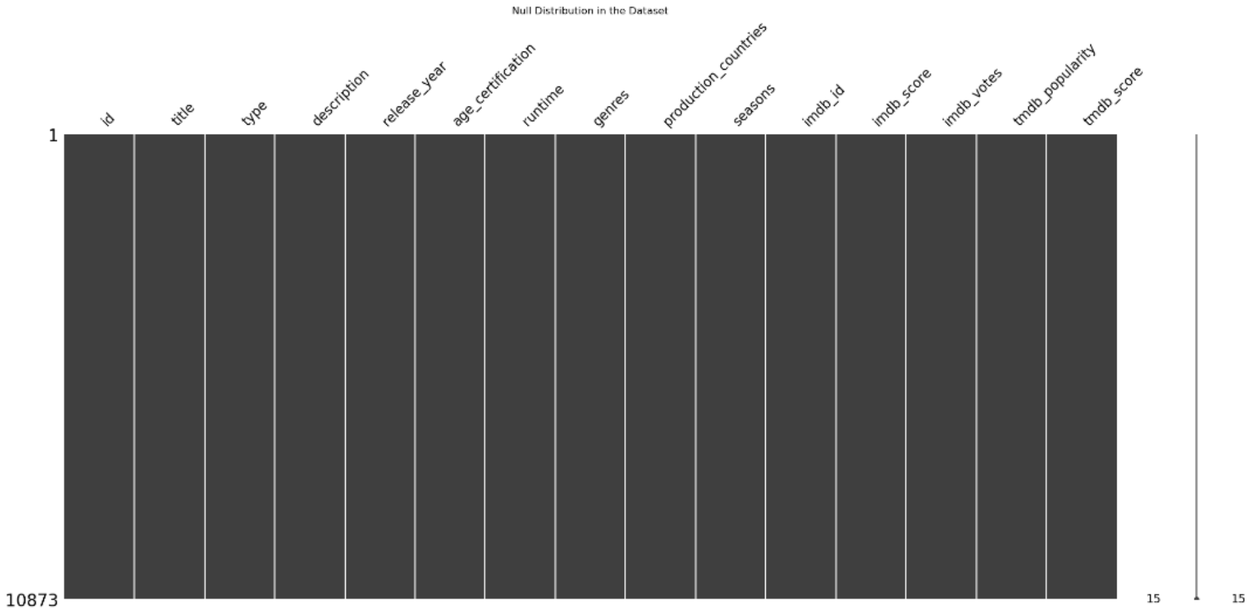
Inspecting Missing Values in the Dataset of Credits

```
1 msno.matrix(df2)
2 plt.title('Null Distribution in the Dataset')
3 plt.show()
```

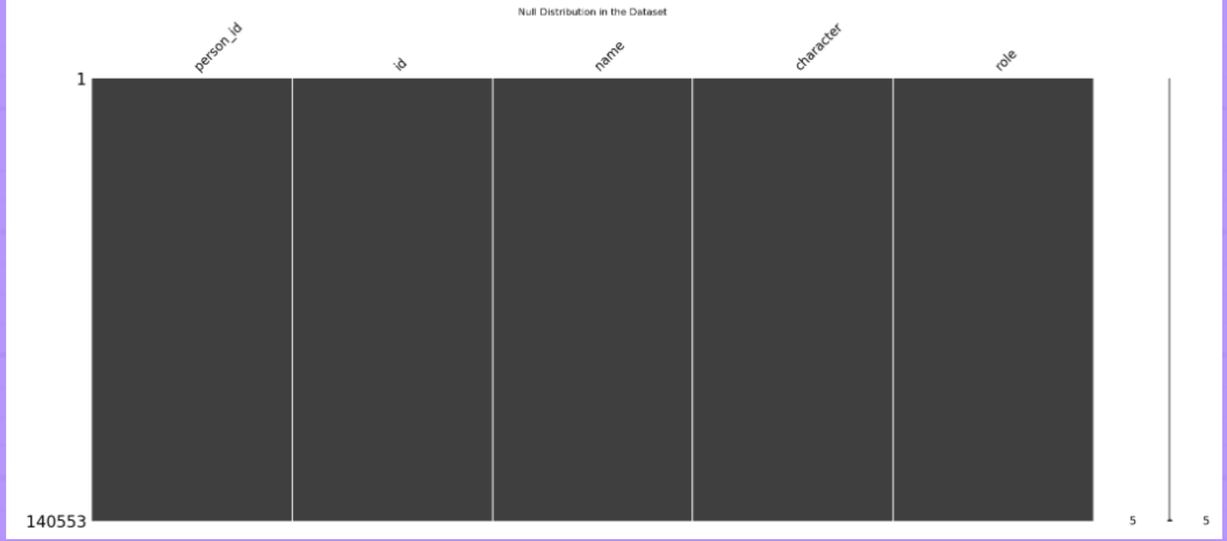


After handling null values

```
msno.matrix(df1)
plt.title('Null Distribution in the Dataset')
plt.show()
```




```
msno.matrix(df2)
plt.title('Null Distribution in the Dataset')
plt.show()
```



CONCLUSIONS

As we conclude this data analysis project, our report offers a unique perspective on the content of Amazon Prime Movies and TV Shows. By Data Analysis using Python, we provide content creators and producers with actionable insights to optimize content curation, enhance user experience, and stay ahead in the fiercely competitive digital entertainment field.