Project Report

Objective:

- Gain insights into Netflix's content distribution, focusing on movies and TV shows.
- Understand the trends in Netflix's content library, including factors such as ratings, genres, international distribution, and the platform's emphasis on TV shows versus movies.
- Creating a movie recommendation algorithm which uses the TF-IDF matrix and cosine similarity to find titles that are textually like a given input title.
- Identify directors or actors known for producing or starring in content with specific ratings.

Dataset:

```
]: netflix_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7787 entries, 0 to 7786
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       7787 non-null   object
 1   type          7787 non-null   object
 2   title         7787 non-null   object
 3   director      5398 non-null   object
 4   cast          7069 non-null   object
 5   country       7280 non-null   object
 6   date_added    7777 non-null   object
 7   release_year  7787 non-null   int64 
 8   rating        7780 non-null   object
 9   duration      7787 non-null   object
 10  listed_in     7787 non-null   object
 11  description   7787 non-null   object
```

1.a Dataset Profile

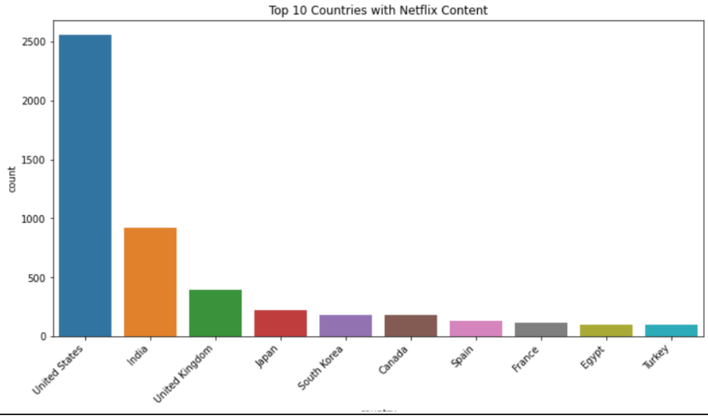
Overview:

The project extensively employed exploratory data analysis (EDA) using Python, utilizing libraries such as pandas, matplotlib, seaborn, and scikit-learn. The Netflix dataset was meticulously explored, revealing its structural nuances. Key analytical tasks included:

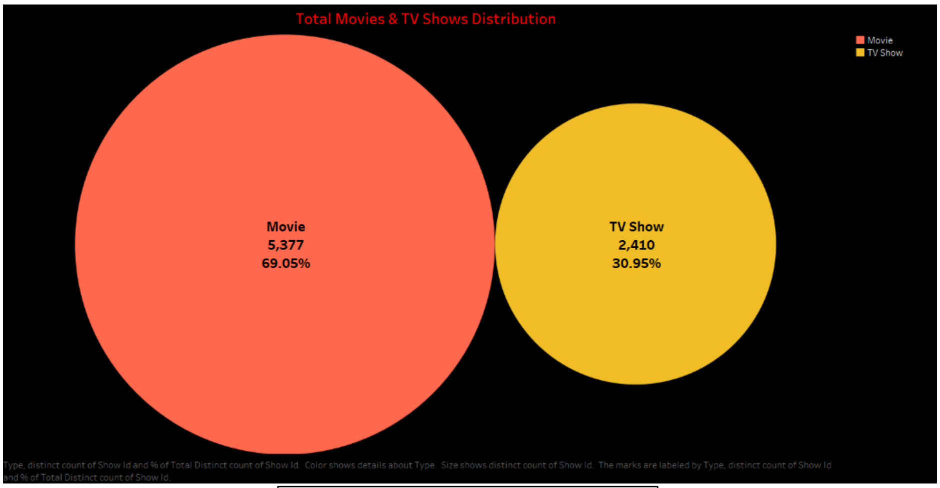
1. **Content Distribution by Country:**

   - Explored top countries driving the content on Netflix based on different content ratings, providing insights into the distribution of highly rated content on the platform.

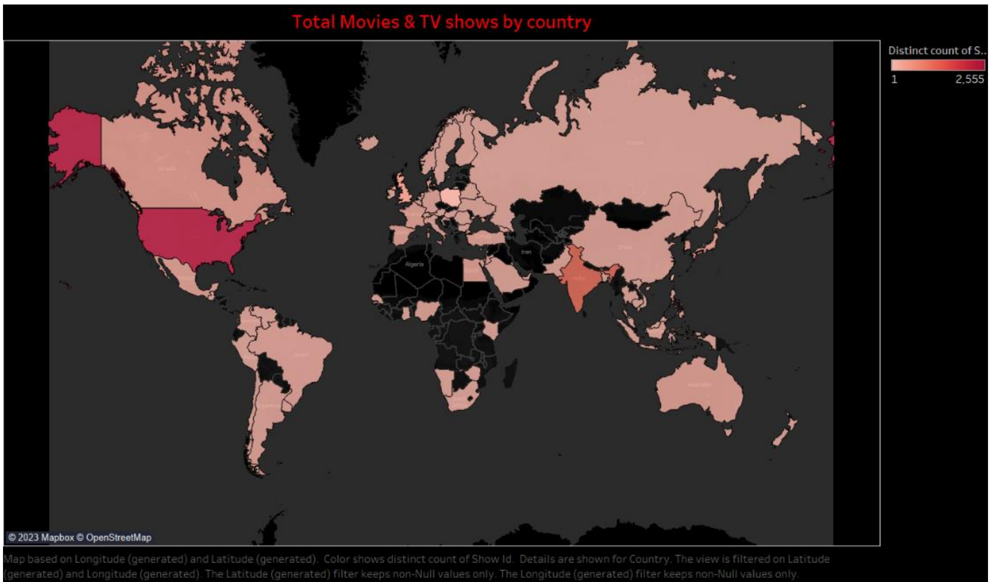
```python
# Visualize the distribution of content by country
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.countplot(x='country', data=netflix_data, order=netflix_data['country'].value_counts().index[:10]
plt.title('Top 10 Countries with Netflix Content')
plt.xticks(rotation=45, ha='right')
plt.show()
```



1.b Major countries leading the content across world in Netflix

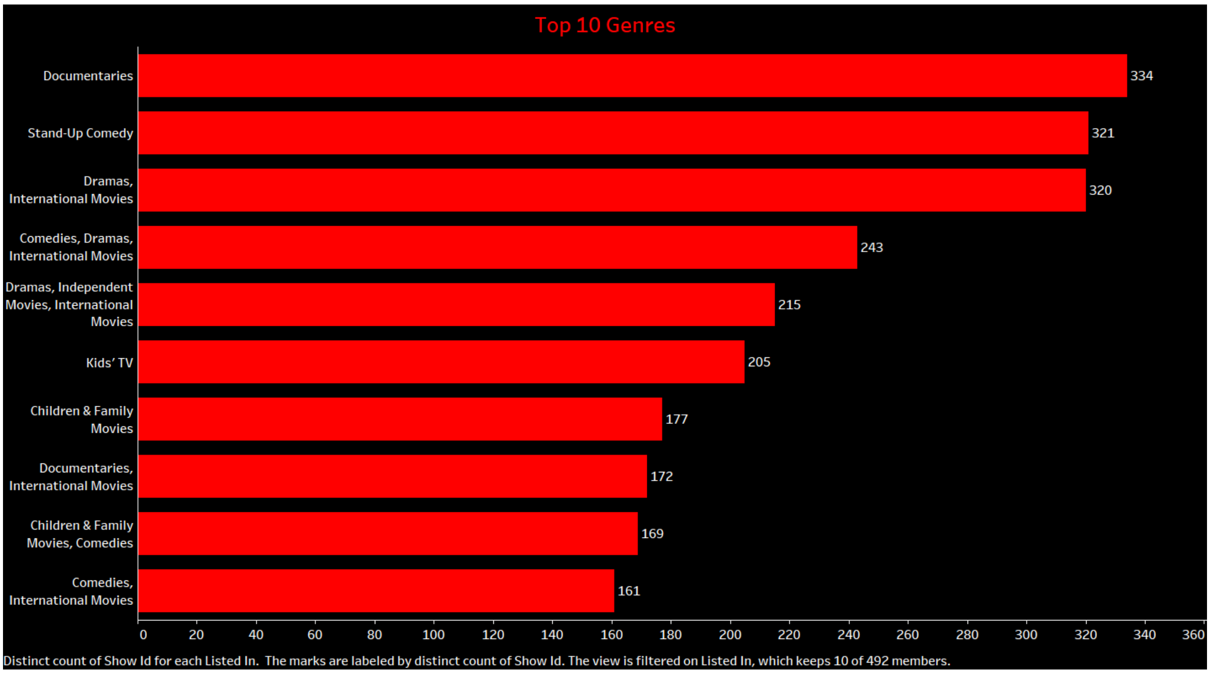


1.c Content Distribution



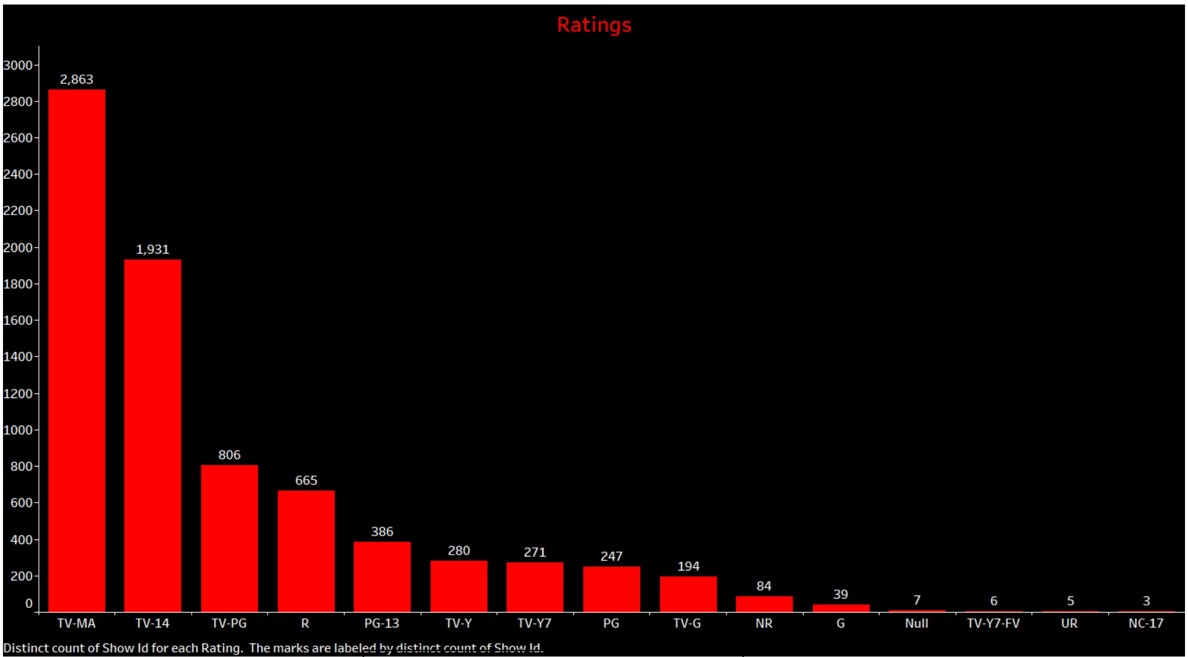
1.d Geographical Distribution

2. **Top Genres among content:**
    - Documentaries, Stand Up comedies and Dramas, International movies are the most popular genres.



1.e Top Genres among content

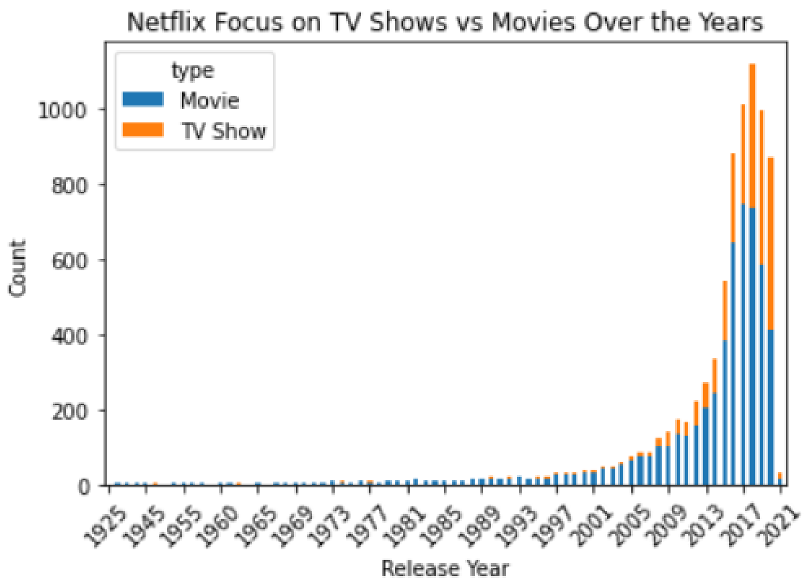
3. **Most popular ratings**:
    - TV-MA, TV-14, TV-PG are the most popular ratings for content which emphasising more on mature, teen, and intense storylines which need parental guidance.



1.f Most popular ratings

4. **Decadal Analysis of Ratings**:
    - Transformed release years into decades and created a stacked bar chart illustrating the count of top ratings for each decade, unveiling the evolution of top-rated content over different time periods.
    - It was clear that Netflix is investing more on TV shows rather than movies in recent decades.

```python
# Task 3: Is Netflix increasingly focusing on TV rather than movies in recent years

# Extract relevant columns for this task
focus_data = netflix_data[['type', 'release_year']]

# Count the number of TV shows and movies released each year
focus_count = focus_data.groupby(['release_year', 'type']).size().unstack().fillna(0)
# Set a larger figure size
plt.figure(figsize=(12, 8))

# Plot the results with adjusted x-axis interval
ax = focus_count.plot(kind='bar', stacked=True)
plt.title('Netflix Focus on TV Shows vs Movies Over the Years')
plt.xlabel('Release Year')
plt.ylabel('Count')

# Adjust x-axis interval further
plt.xticks(range(0, len(focus_count.index), 4), focus_count.index[::4], rotation=45)

plt.show()
```



1.g Netflix Content types among decades

5. **Director or Actor Preferences:**
- Identified directors and actors known for producing or starring in content with specific ratings, aiming to uncover potential bias in viewership.

```python
# Function to get unique directors for a specific rating
def get_directors_by_rating(df, rating):
    filtered_data = df[(df['rating'] == rating) & (df['director'].notna())]
    directors = filtered_data['director'].str.split(',').explode().str.strip().unique()
    return directors

# Function to get unique actors for a specific rating
def get_actors_by_rating(df, rating):
    filtered_data = df[(df['rating'] == rating) & (df['cast'].notna())]
    actors = filtered_data['cast'].str.split(',').explode().str.strip().unique()
    return actors

# Example: Get unique directors and actors for TV-MA rated content
tv_ma_directors = get_directors_by_rating(df, 'TV-MA')
tv_ma_actors = get_actors_by_rating(df, 'TV-MA')

# Display the results
print(f"Unique Directors for TV-MA: {tv_ma_directors}")
print(f"Unique Actors for TV-MA: {tv_ma_actors}")
```

```python
Actors_df = pd.DataFrame(tv_ma_actors)
Actors_df.head(10)
```

	0
0	João Miguel
1	Bianca Comparato
2	Michel Gomes
3	Rodolfo Valente
4	Vaneza Oliveira
5	Rafael Lozano
6	Viviane Porto
7	Mel Fronckowiak
8	Sergio Mamberti
9	Zezé Motta

1.h Most popular actor among TV-MA rated movies

```python
Directors_df = pd.DataFrame(tv_ma_directors)
Directors_df.head(10)
```

	0
0	Jorge Michel Grau
1	Serdar Akar
2	Yasir Al Yasiri
3	Vikram Bhatt
4	Zak Hilditch
5	Diego Enrique Osorno
6	Nottapon Boonprakob
7	Cho Il
8	Cristina Jacob
9	Frank Ariza

1.i Most popular directors among TV-MA rated

6. **Movie Recommendation System using TF-IDF and Cosine Similarity**
   - Introduced a movie recommendation system to enhance user engagement and content discoverability. The recommendation system is based on advanced natural language processing techniques, specifically TF-IDF (Term Frequency-Inverse Document Frequency), and cosine similarity.

Techniques Used:

TF-IDF (Term Frequency-Inverse Document Frequency):

- TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents (corpus). It considers the frequency of a term in a document (Term Frequency) and the rarity of the term across the entire corpus (Inverse Document Frequency).
- The TF-IDF matrix is created using the TfidfVectorizer from scikit-learn, which converts a collection of raw documents to a matrix of TF-IDF features.

Cosine Similarity:

- Cosine similarity is a metric used to measure how similar two documents are. It calculates the cosine of the angle between two vectors, representing the documents, in a multidimensional space.
- In our recommendation system, we compute the cosine similarity between all movies based on their TF-IDF vectors. This similarity score is then used to identify movies that are most like a given input.

Results:

- The recommendation system successfully provides meaningful and relevant movie suggestions based on the textual features of the titles. When a user inputs a specific movie title, the system identifies similar movies from the dataset. The results are displayed as a list of recommended titles.
- These recommendations leverage the textual information (title and description) of the movies to suggest content that is semantically similar, providing users with a personalized and engaging viewing experience.
- This recommendation system enhances the overall user experience on the Netflix platform by offering tailored suggestions, increasing user satisfaction, and encouraging continued engagement with the diverse content library.

Example:

- Consider the example where the user is interested in recommendations for the movie with the title containing '3%'. The system processes the TF-IDF vectors and computes the cosine similarity scores. The top 5 movies with the highest similarity scores, excluding the input movie, are then presented as recommendations.

  Output:
  Recommendations for '3%':
  1. Elite Squad: The Enemy Within
  2. The Mechanism
  3. City of God
  4. O Mecanismo
  5. The Constant Gardener

```python
In [21]: # Task 2: Identifying similar content by matching text-based features
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel

# Combine relevant text-based features
netflix_data['text_features'] = netflix_data['title'] + ' ' + netflix_data['description']

# Create a TF-IDF matrix
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(netflix_data['text_features'].fillna(''))

# Compute the cosine similarity
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)

# Function to get recommendations based on similarity
def get_recommendations(title):
    idx = netflix_data.index[netflix_data['title'] == title].tolist()[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:6]
    movie_indices = [i[0] for i in sim_scores]

    # Adjust indices by subtracting 1
    movie_indices = [idx + 1 for idx in movie_indices]

    return netflix_data['title'].iloc[movie_indices]
```

Output:

```
Recommendations for '3%':
2173     Fire in the Blood
5822           Stolen Away
6526            The Killer
6908          THE STRANGER
3635              Lifeline
Name: title, dtype: object
```

1.J Recommended movies for 3% by alogorithm

Conclusions

1. Key Findings:
    - The top-rated movies analysis highlighted the prevalence of TV-MA ratings, followed by TV-14 and TV-PG.
    - Decadal analysis showcased shifts in the prominence of specific ratings over different decades.
    - Content distribution by country identified the United States, India, and the United Kingdom as leading contributors to Netflix's diverse content library.
    - The content similarity analysis successfully provided recommendations, improving user experience and content discovery.
    - The focus analysis indicated a notable increase in TV show production compared to movies in recent years.
2. Strategic Content Adaptation:
    - Netflix has strategically evolved its content library, making significant adjustments over the years to cater to changing viewer preferences and market dynamics.
3. Genre and Rating Focus:
    - The platform's content strategy involves a keen focus on specific genres and ratings, reflecting a targeted approach to meet diverse viewer demands and preferences.
4. Format Preference:
    - The investigation reveals a noteworthy shift in focus towards TV shows in recent years, indicating a strategic emphasis on episodic content over traditional movies.
5. Enhanced User Engagement:
    - The recommendations generated through content similarity analysis contribute to an enriched user experience, fostering higher engagement by aligning content suggestions with viewer preferences.
6. Actionable Insights:
    - The insights gained from this analysis provide valuable guidance for content creators, platform managers, and decision-makers within Netflix, offering actionable strategies to further optimize content offerings and viewer satisfaction.