Capstone Project – 4

NETFLIX MOVIES AND TV SHOWS CLUSTERING

By - Puneet Suthar

What are we talking about ?

Content :

- Introduction to Netflix
- Problem Statement
- General overview of the dataset
- Feature engineering and data cleaning.
- Exploratory Data Analysis
- Clustering
- Conclusion

Introduction

Netflix operates as a streaming service, offering an extensive library of movies and TV shows accessible online at any time. The company derives its revenue from users who make monthly payments to utilize the platform, but customers can cancel their subscriptions without restrictions. Consequently, Netflix must ensure that users remain engaged with the platform and not lose their interest. To achieve this, recommendation systems play a crucial role by providing users with valuable suggestions to enhance their viewing experience.z

Problem Statement

In this project, you are required to do :-

1. Exploratory Data Analysis
2. Understanding what type content is available in different countries
3. Is Netflix has increasingly focusing on TV rather than movies in recent years.
4. Clustering similar content by matching text-based features

How our dataset Look like ?

General overview of the dataset

12 Columns

&

7787 Rows

General overview of the dataset

12 Columns

1. **Show id** : Unique identifier of the record in the dataset
2. **Type** : Whether it is a TV show or movie
3. **Title** : Title of the show or movie
4. **Director** : Director of the TV show or movie
5. **Cast** : The cast of the movie or TV show
6. **Country** : The list of the country in which a show/ movie is released or watched
7. **Date added** : The date on which the content was onboarded on the Netflix platform
8. **Release year** : Year of the release of the show/ movie
9. **Rating** : The rating informs about the suitability of the content for a specific age group
10. **Duration** : Duration is specified in terms of minutes for movies and in terms of the number of seasons in the case of TV shows
11. **Listed in** : This columns species the category/ genre of the content
12. **Description** : A short summary about the storyline of the content

Feature Engineering and data cleaning

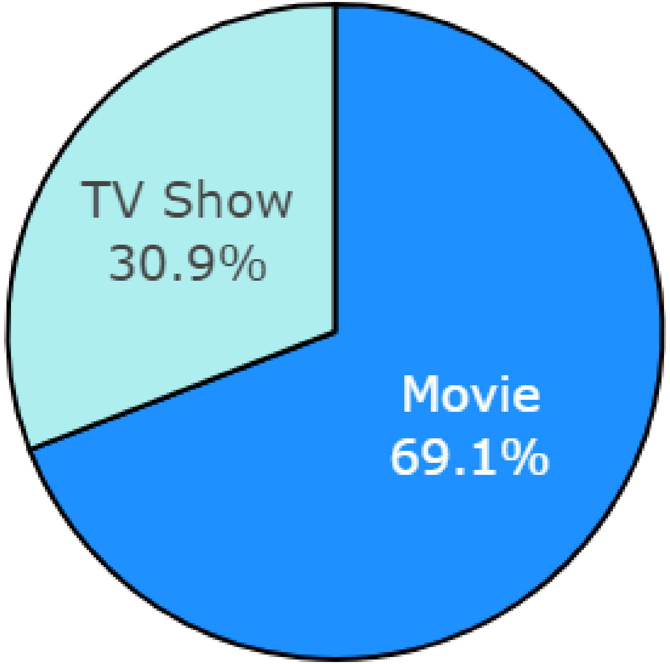
Feature Engineering and data cleaning

1. We have 2389 null values in director column. We have almost 30% null values in this column. so replacing null with unknown.
2. We have 718 null values in cast column. and it can be replaced with 'unknown'.
3. We have 507 null values in country column. Replacing nulls with 'mode' value that is USA.
4. Also we have 10 null values in date_added column. we have few rows of date_added so we can 'drop' these rows.
5. As rating column has 0.08% null values , so replacing nulls with most frequent TV-MA rating.
6. We left with 7777 Rows.
7. We created new colums named day_added, month_added, year_added.
8. Adding new features based the list of above converted lists:-
   1. number_of_director
   2. number_of_cast
   3. number_of_countries
   4. number_of_genres

EDA

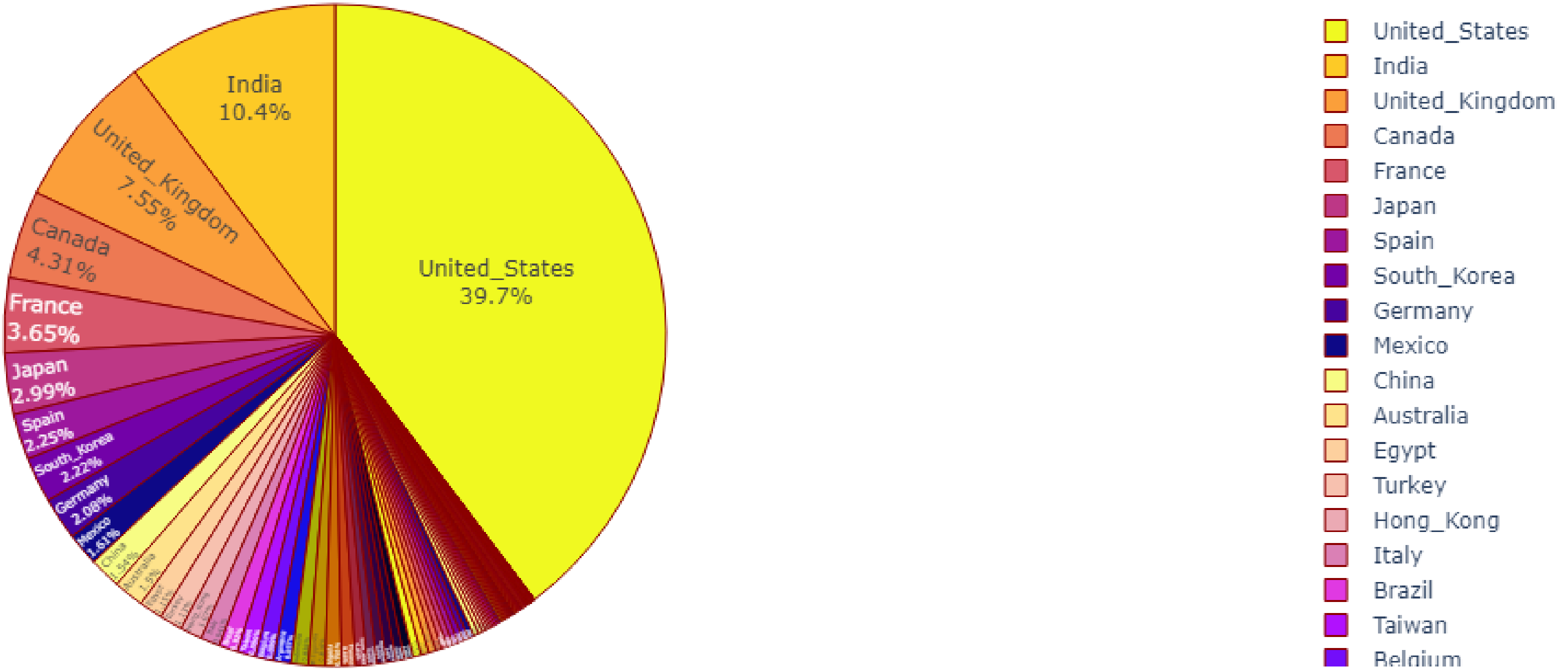
(Exploratory Data Analysis)

Type of content



The major portion is acquired by movies

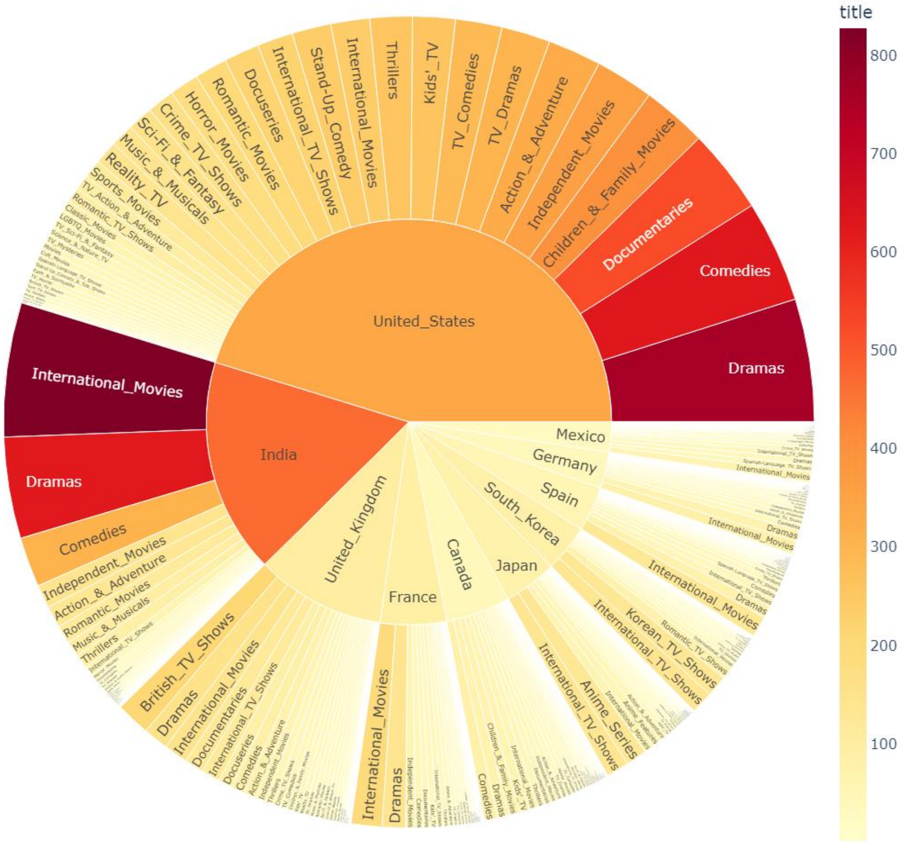
Countries


India
10.4%
United_Kingdom
7.55%
Canada
4.31%
France
3.65%
Japan
2.99%
Spain
2.25%
South_Korea
2.22%
Germany
2.08%
Mexico
1.61%
China
Australia
United_States
39.7%
United_States
India
United_Kingdom
Canada
France
Japan
Spain
South_Korea
Germany
Mexico
China
Australia
Egypt
Turkey
Hong_Kong
Italy
Brazil
Taiwan
Belgium

Genre

EDA
(Exploratory Data Analysis)

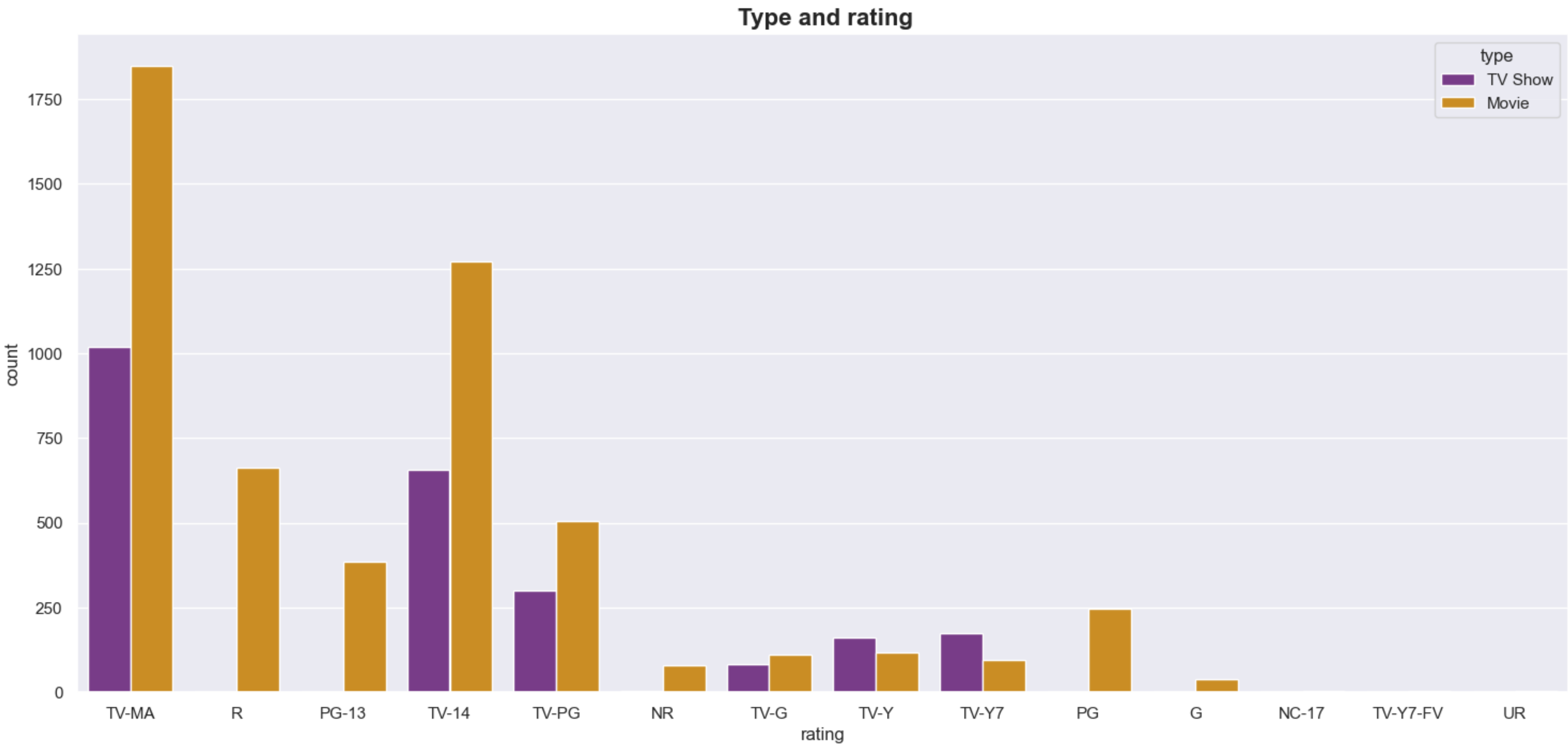
Number of Titles by Country and Listed In Category



Most of the countries have international movies or tv shows as highest count genre.

Type and rating

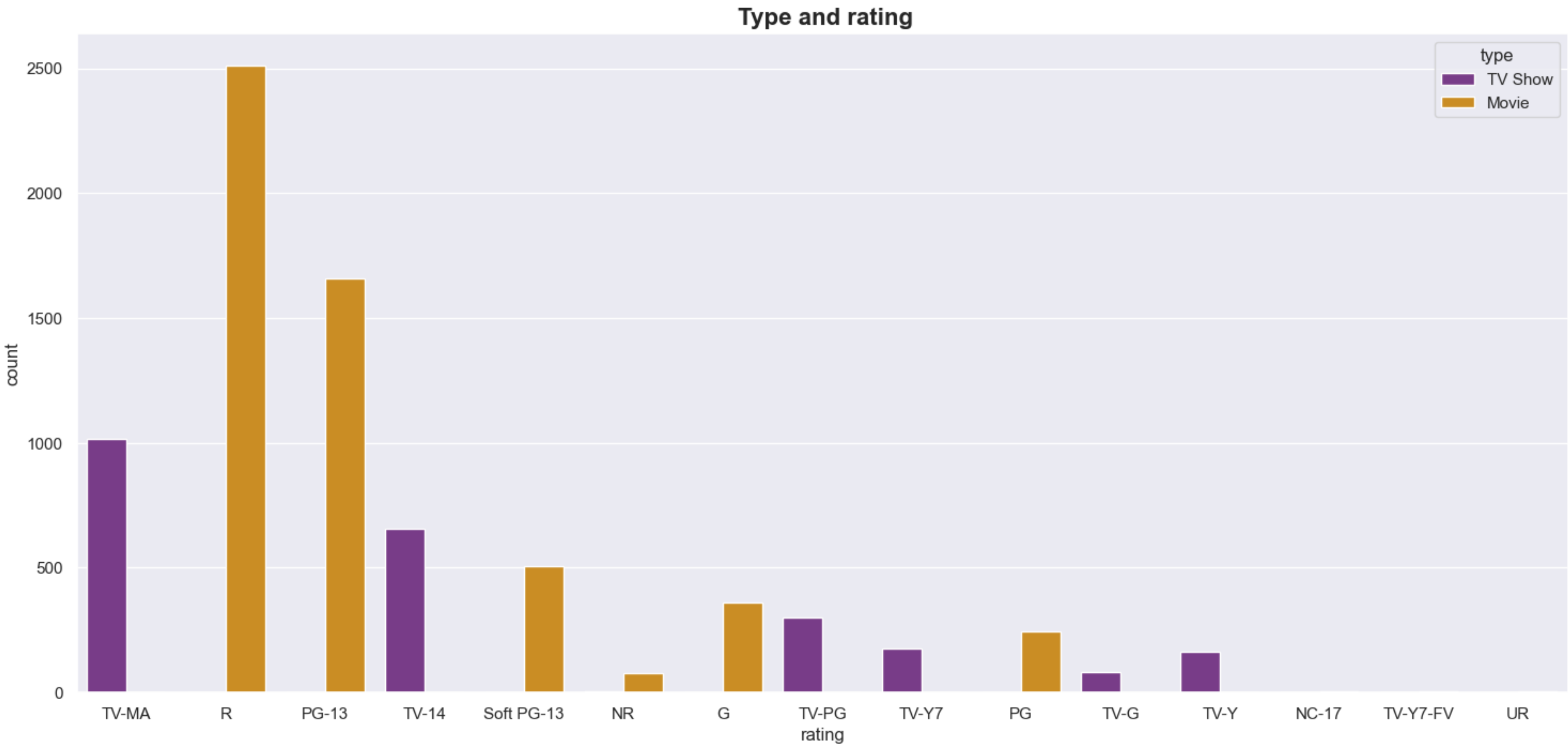
EDA
(Exploratory Data Analysis)



Type and rating fixed

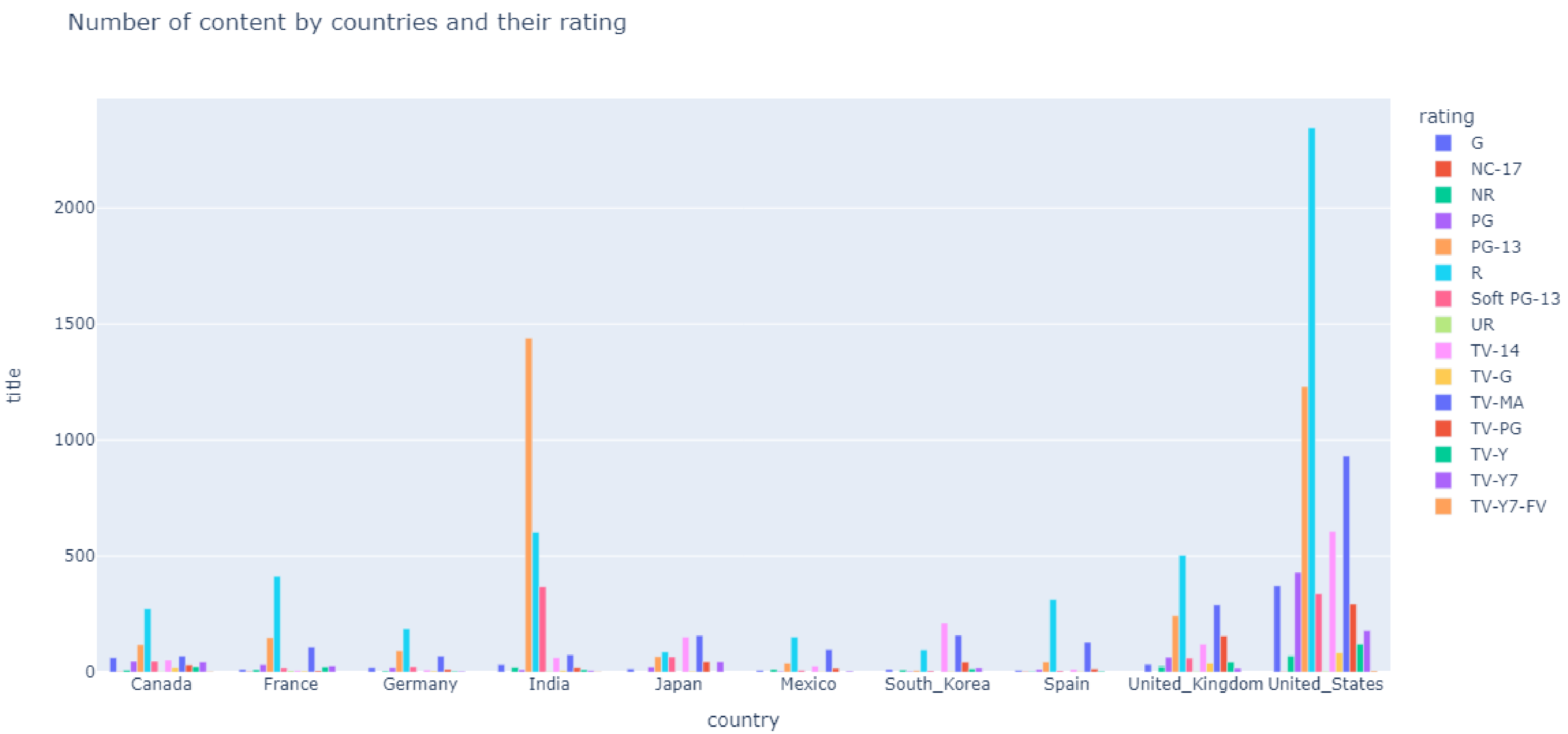
EDA
(Exploratory Data Analysis)

AI

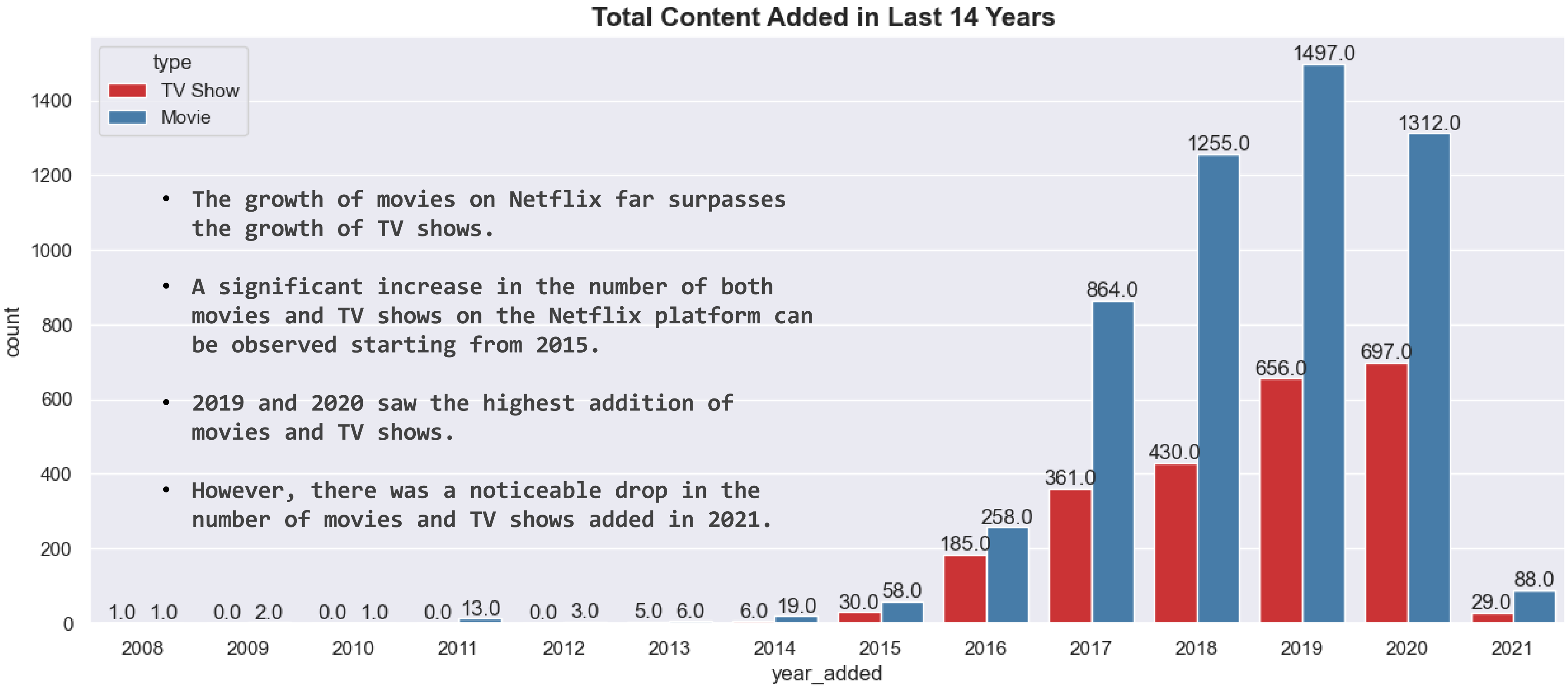


Countries

EDA
(Exploratory Data Analysis)



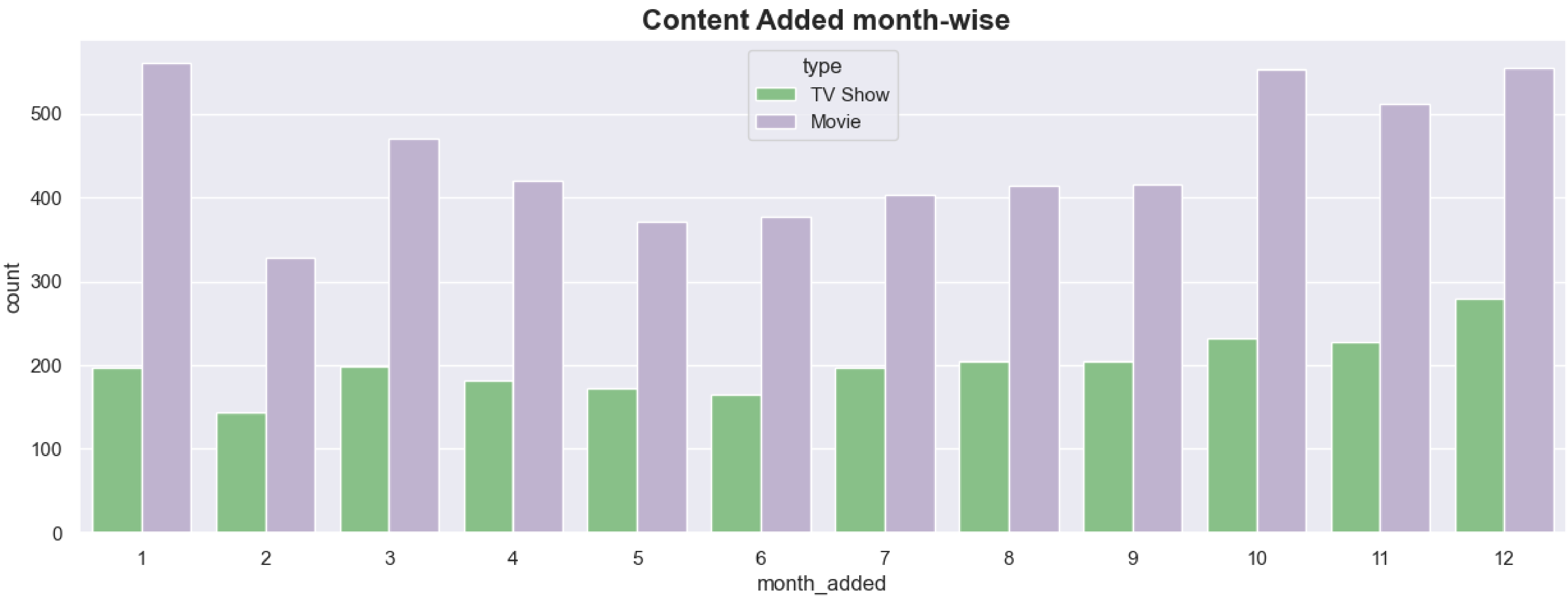
Year



Total Content Added in Last 14 Years

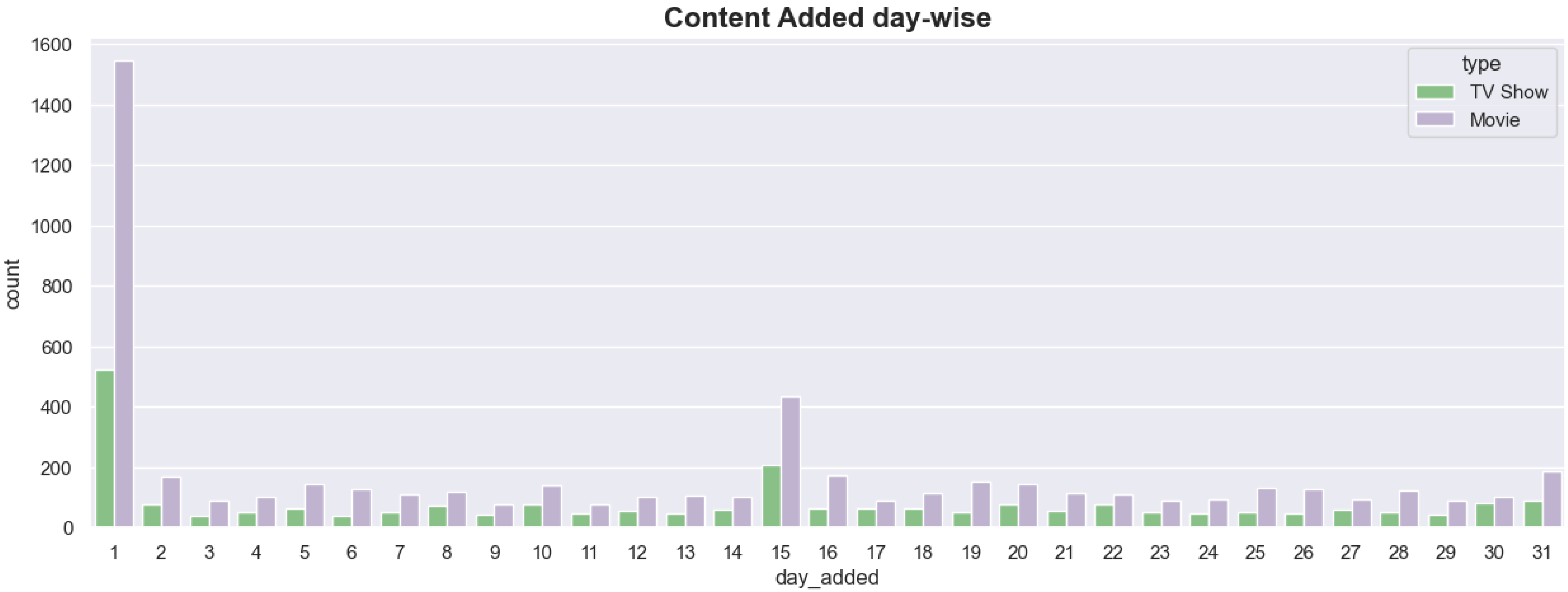
- The growth of movies on Netflix far surpasses the growth of TV shows.
- A significant increase in the number of both movies and TV shows on the Netflix platform can be observed starting from 2015.
- 2019 and 2020 saw the highest addition of movies and TV shows.
- However, there was a noticeable drop in the number of movies and TV shows added in 2021.

Month


Content Added month-wise
type
🟩 TV Show
🟪 Movie
count
0
100
200
300
400
500
1
2
3
4
5
6
7
8
9
10
11
12
month_added

Day

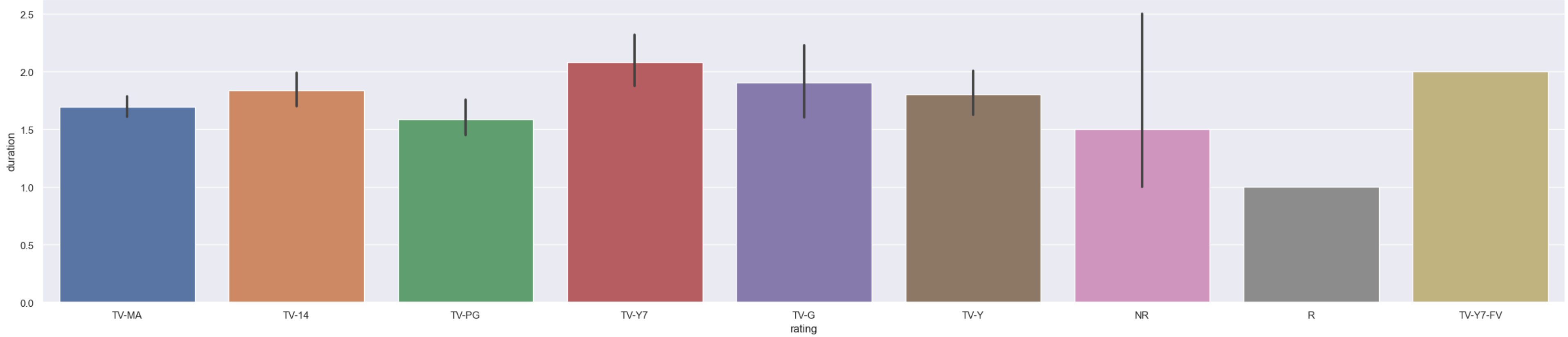
EDA
(Exploratory Data Analysis)

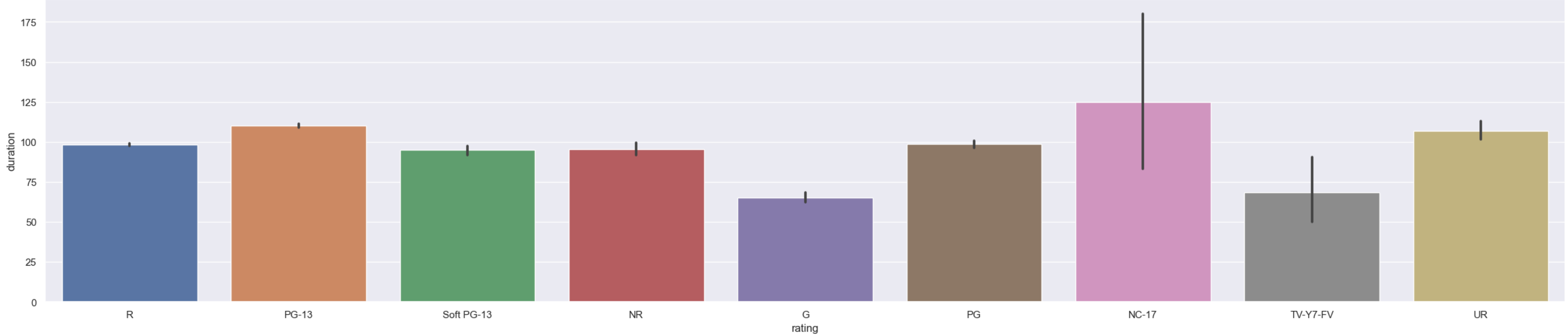


Most of the content gets uploaded in the beginning and the middle of the month

Rating and duration

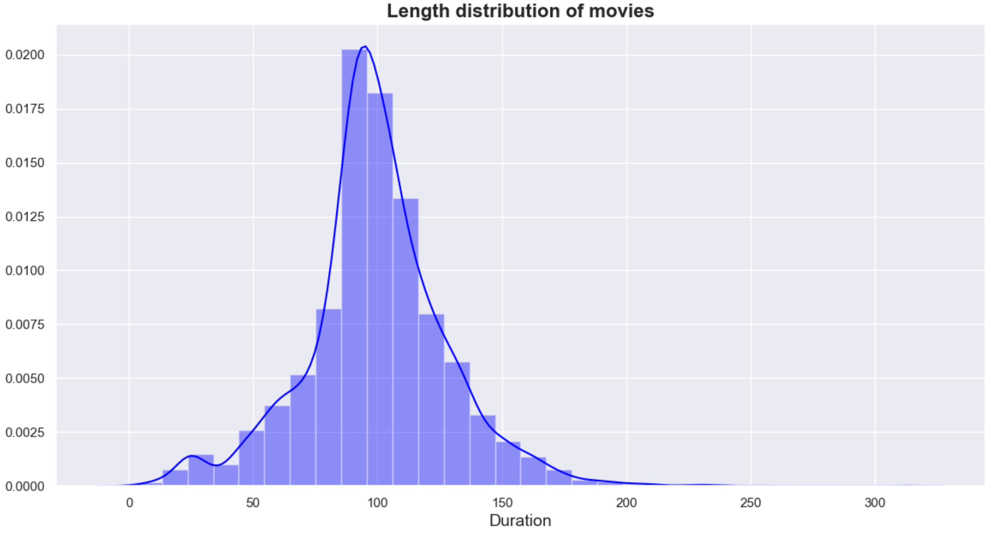
EDA
(Exploratory Data Analysis)

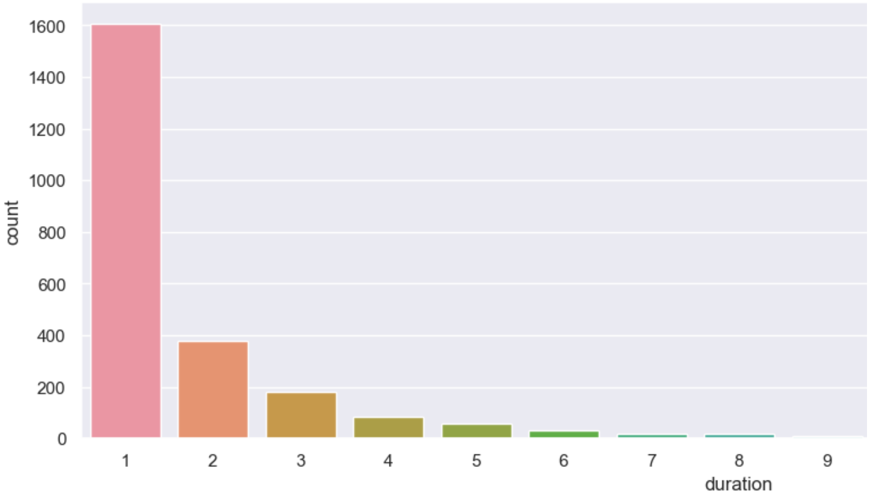




Length of movies and tv shows

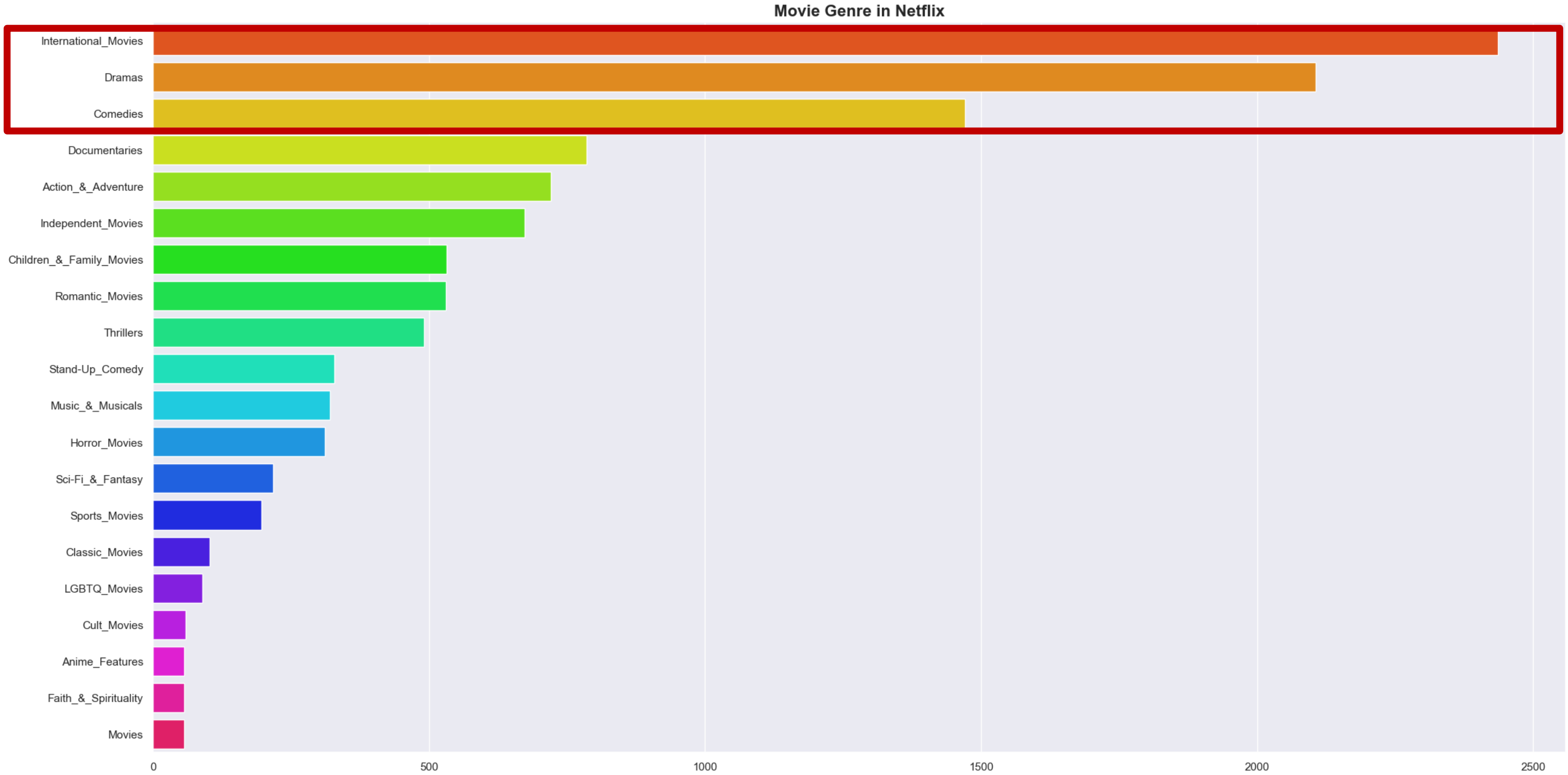
EDA
(Exploratory Data Analysis)





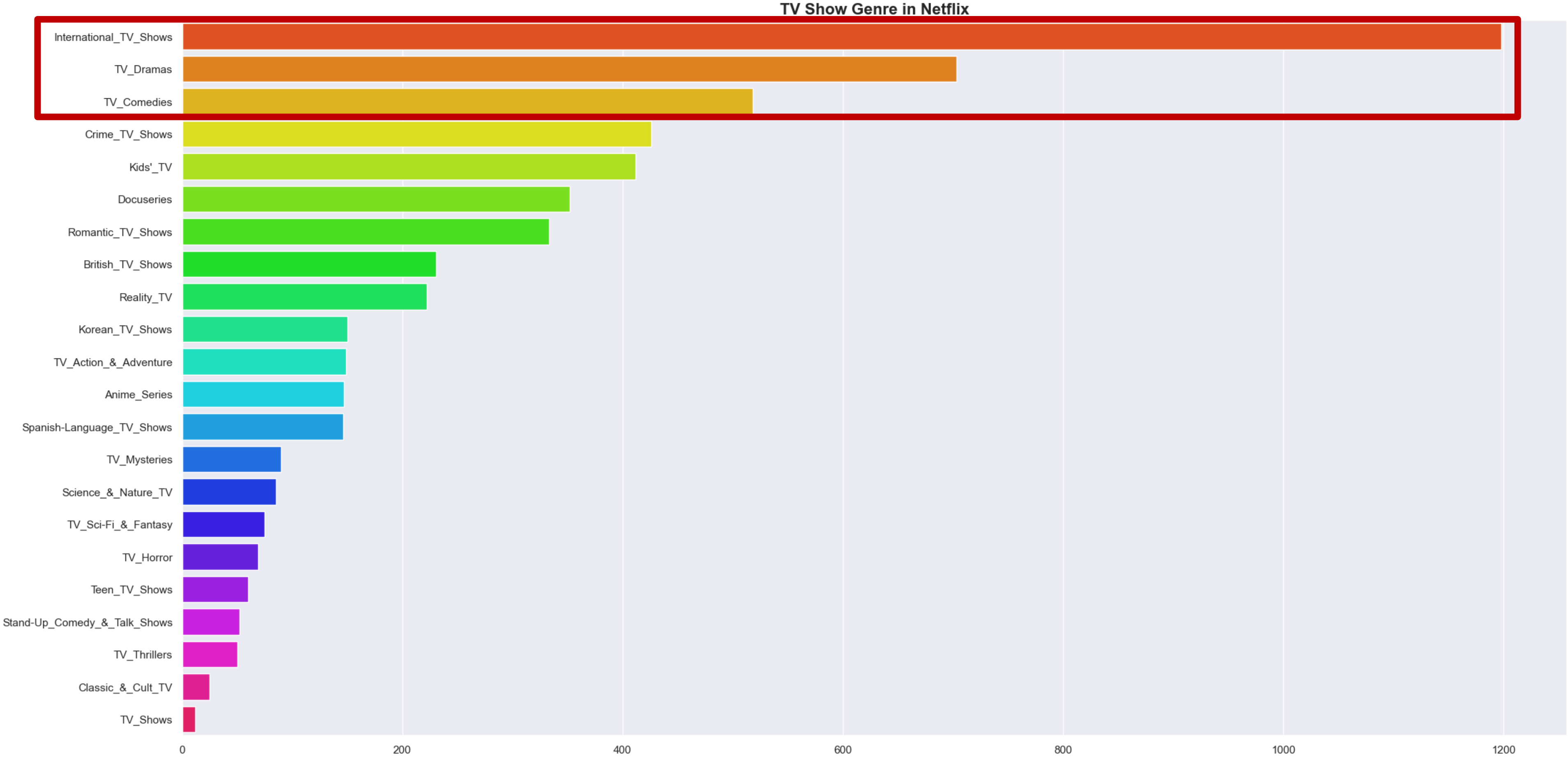
Genre (Movies)

EDA
(Exploratory Data Analysis)



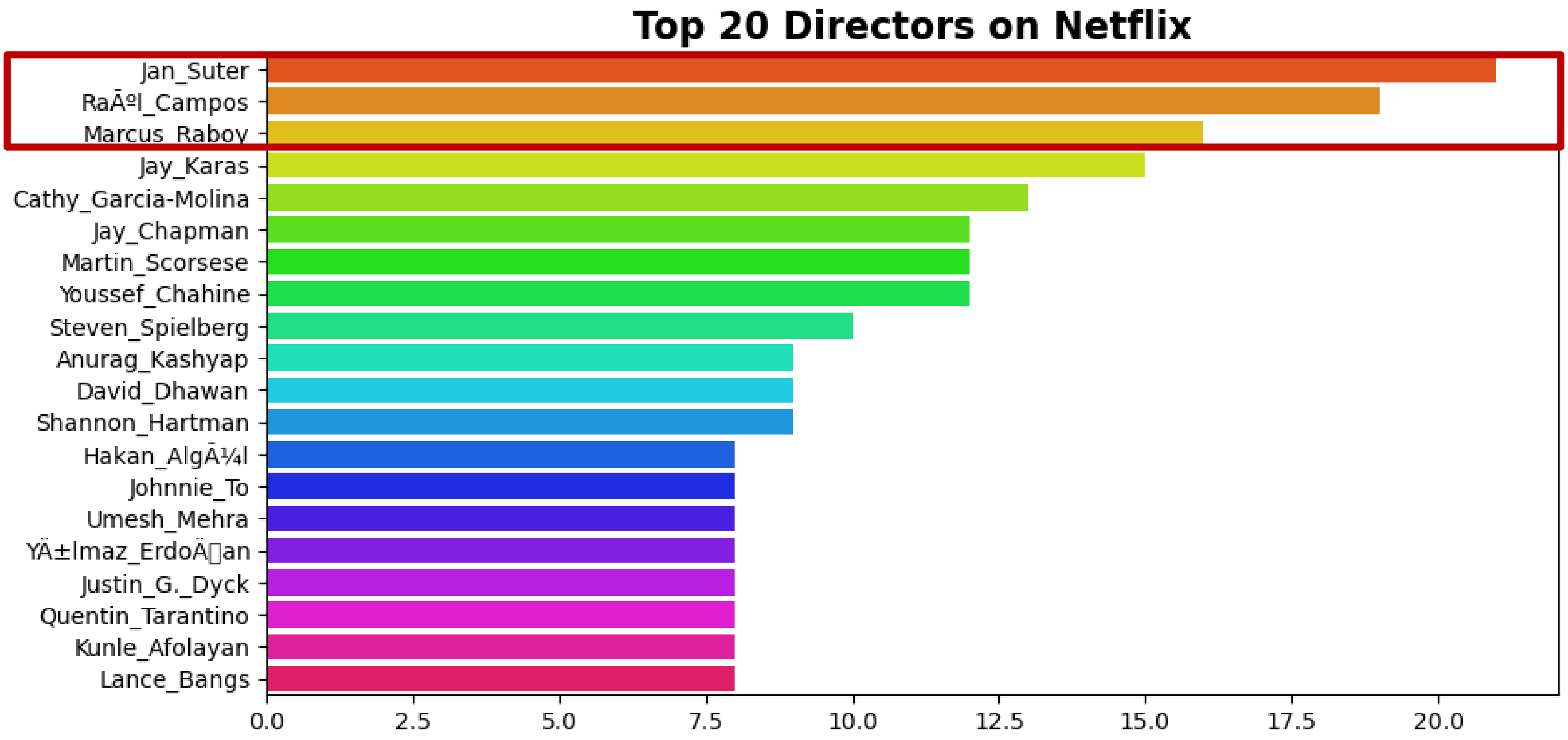
Genre (tv shows)

EDA
(Exploratory Data Analysis)



Directors

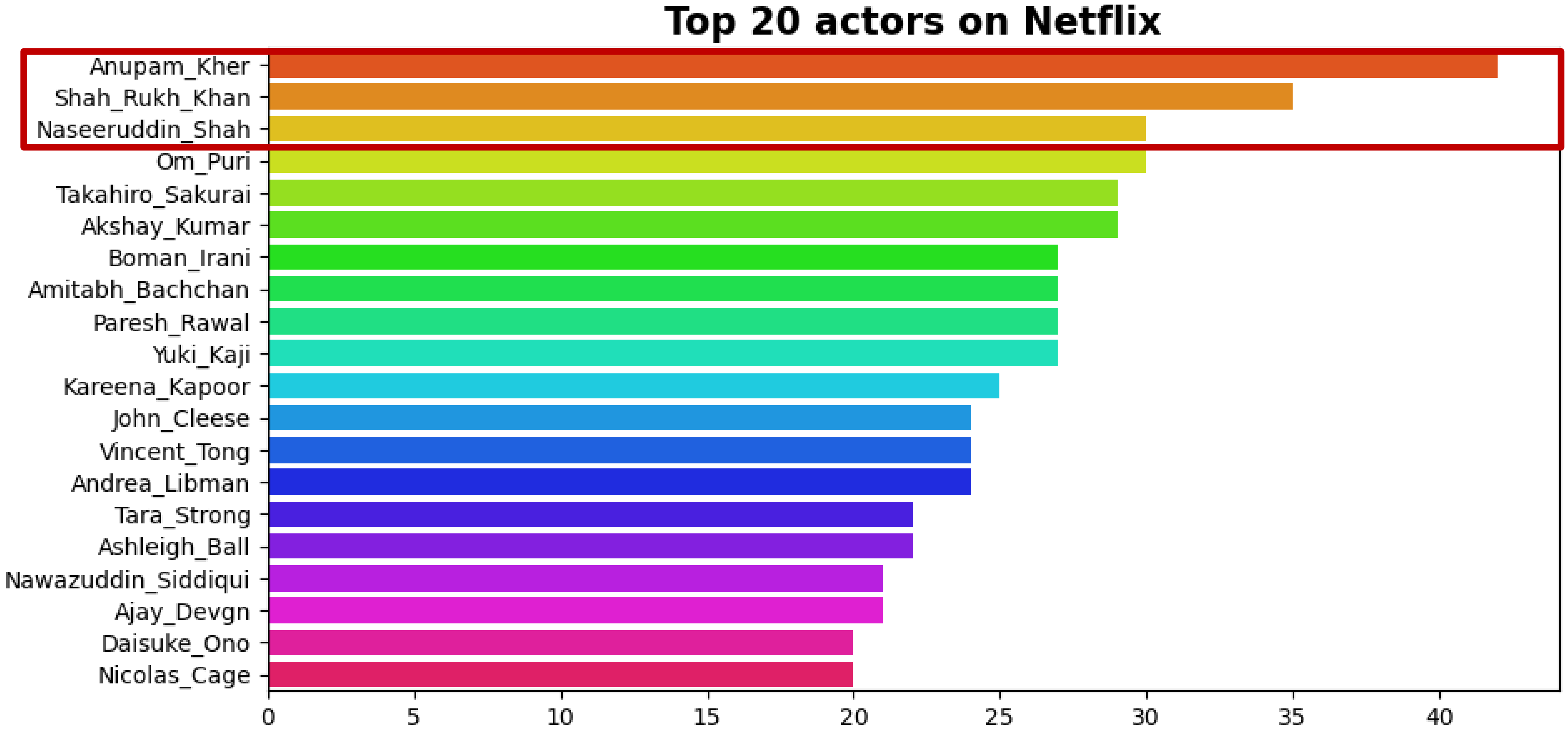
EDA
(Exploratory Data Analysis)



Actors

EDA
(Exploratory Data Analysis)

AI



Clustering

Clustering

Group 1

Group 2

Group 3

Group 4

Clustering

Group 1

1. 'director’
2. 'cast’
3. 'country’
4. 'listed_in '

Group 2

1. 'type’
2. 'release_year’
3. 'rating’
4. 'country_count’
5. 'number_of_directors’
6. 'number_of_casts’
7. 'number_of_countries’
8. 'number_of_genres'

Group 3

1. 'description'

Group 1

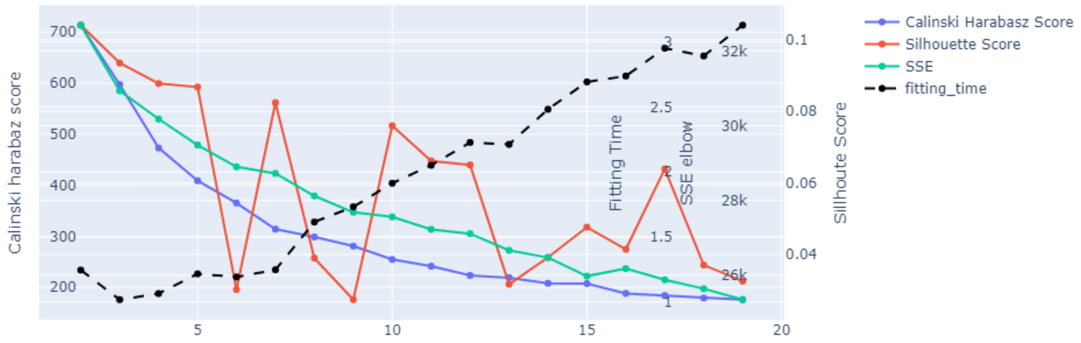
Clustering

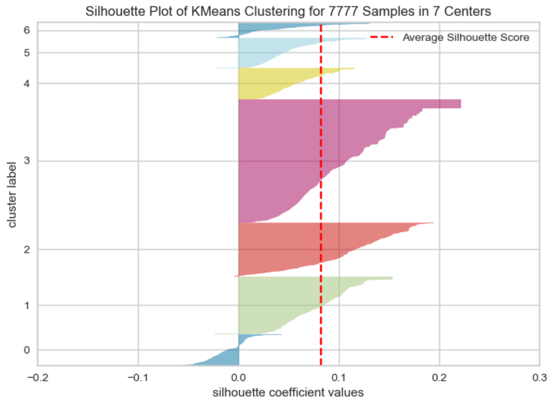
1. 'director’
2. 'cast’
3. 'country’
4. 'listed_in'

By applying k-means and agglomerative clustering on these columns after OHE and merging all the dataframes, we are trying to find similar shows based on their crew and category information. This approach can be useful to discover patterns in the production crew and show categories that can be used for content recommendations or to identify production trends in the entertainment industry.

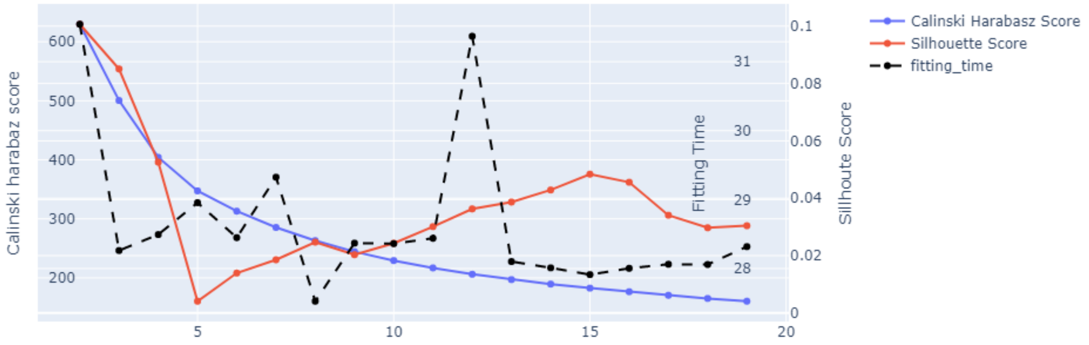
Group 1 | Clustering

KMeans





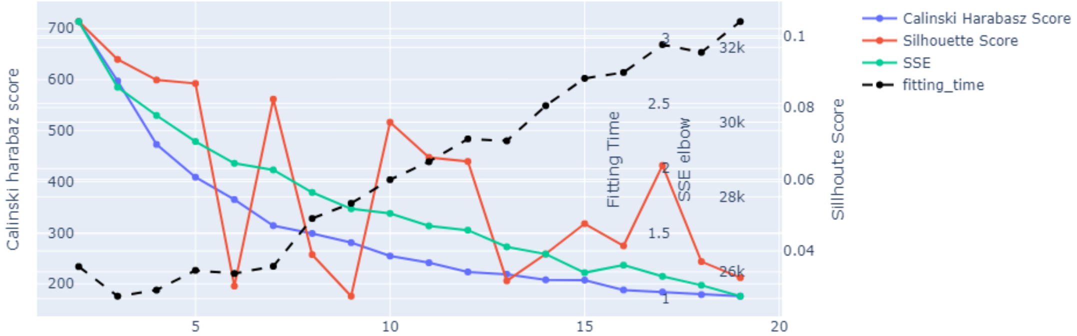
Agglomerative

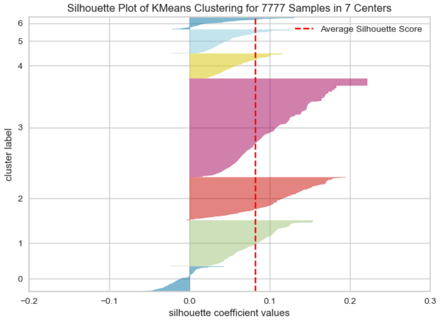


Here KMeans is performing better with vest k = 7

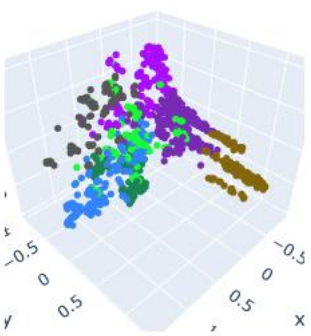
Group 1 | Clustering

KMeans

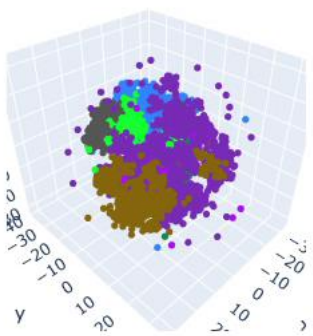




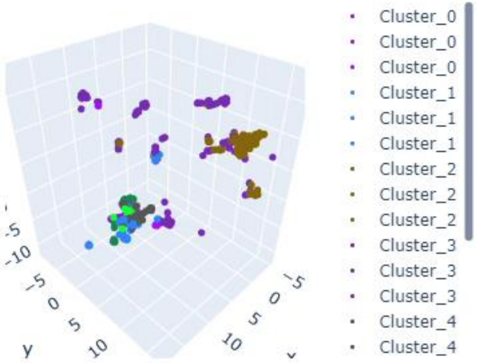
PCA



t-SNE



UMAP



Clustering

Group 1

1. 'director’
2. 'cast’
3. 'country’
4. 'listed_in '

Group 2

1. 'type’
2. 'release_year’
3. 'rating’
4. 'country_count’
5. 'number_of_directors’
6. 'number_of_casts’
7. 'number_of_countries’
8. 'number_of_genres'

Group 3

1. 'description'

Group 2

Clustering

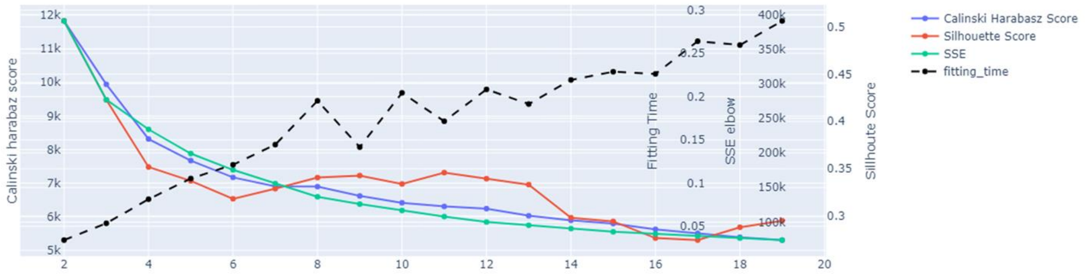
1. 'type’
2. 'release_year’
3. 'rating’
4. 'country_count’
5. 'number_of_directors’
6. 'number_of_casts’
7. 'number_of_countries’
8. 'number_of_genres'

By applying Clustering on these columns we are trying to find similar shows based on more quantitative data such as show type, date added, ratings, etc.

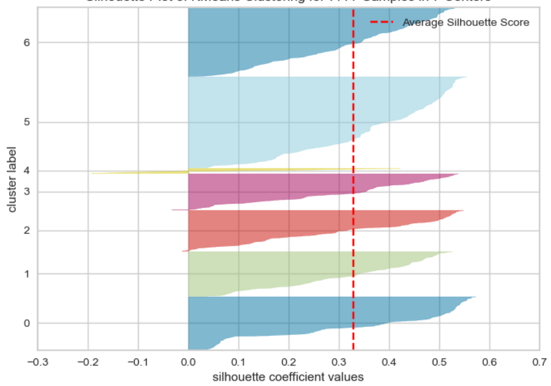
Group 2 | Clustering

KMeans

KMeans cluster metrics

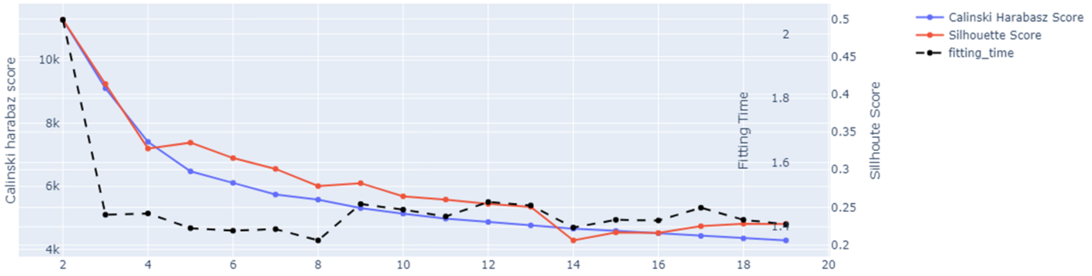


Silhouette Plot of KMeans Clustering for 7777 Samples in 7 Centers



Agglomerative

Agglomerative cluster metrics

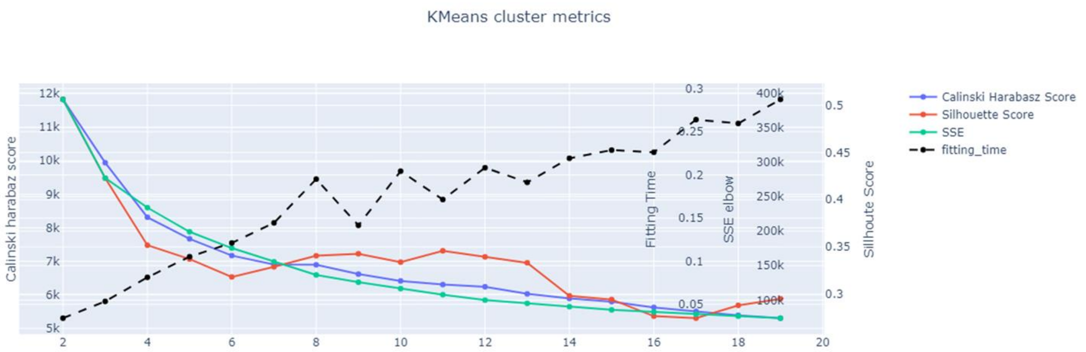


Here KMeans is performing better with vest k = 7

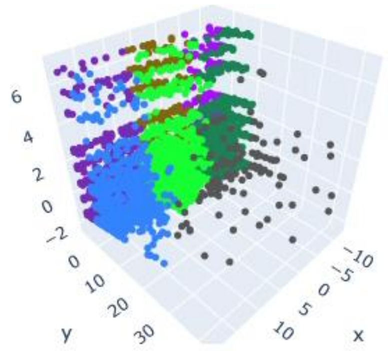
Group 2

Clustering

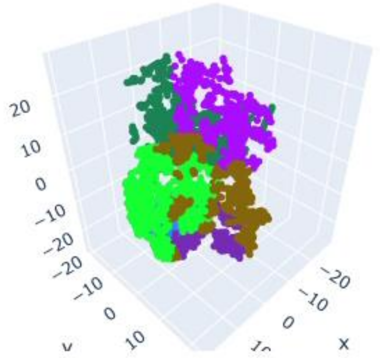
KMeans



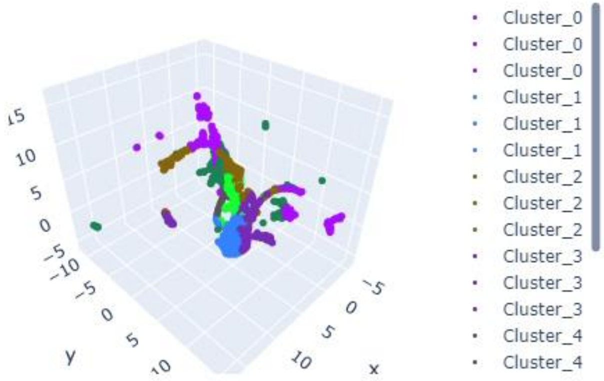
PCA



t-SNE



UMAP



Clustering

Group 1

1. 'director’
2. 'cast’
3. 'country’
4. 'listed_in'

Group 2

1. 'type’
2. 'release_year’
3. 'rating’
4. 'country_count’
5. 'number_of_directors’
6. 'number_of_casts’
7. 'number_of_countries’
8. 'number_of_genres'

Group 3

1. 'description'

Preprocessing For'description’

- lower casing
- Remove punctuation
- Tokenize the text
- Removing Stop words
- Stemming using porter stemmer
- lemmatization using word net lemitization
- And join all of them in a string

Clustering

1. 'description'

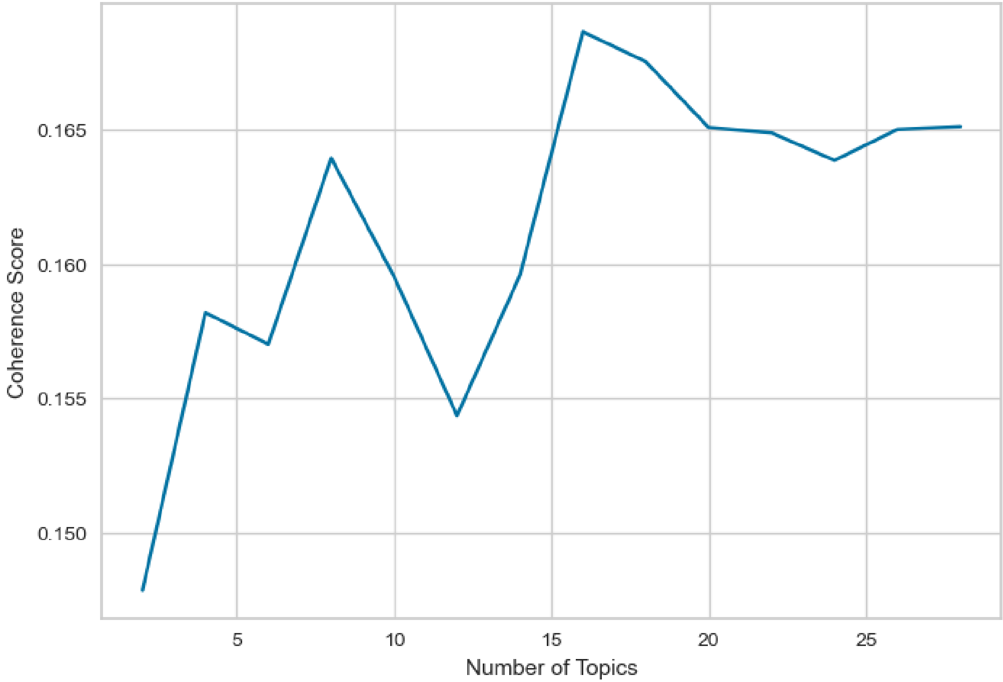
    1. By applying LDA (Latent Dirichlet Allocation) on the 'description' column, we are trying to extract topics or themes present in the show descriptions

    2. and then using k-means clustering to group similar shows based on these themes. This approach can be useful to understand the content and genre of shows, and to discover patterns in the storytelling, themes and genre of shows.

Group 3

Clustering

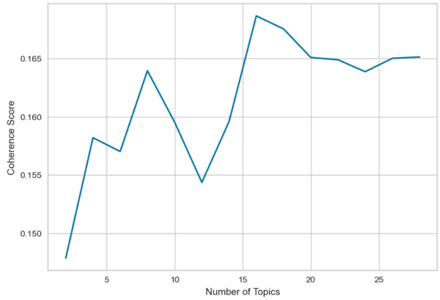
Coherence Score



By calculating coherence score we found best no. of topics are 16.

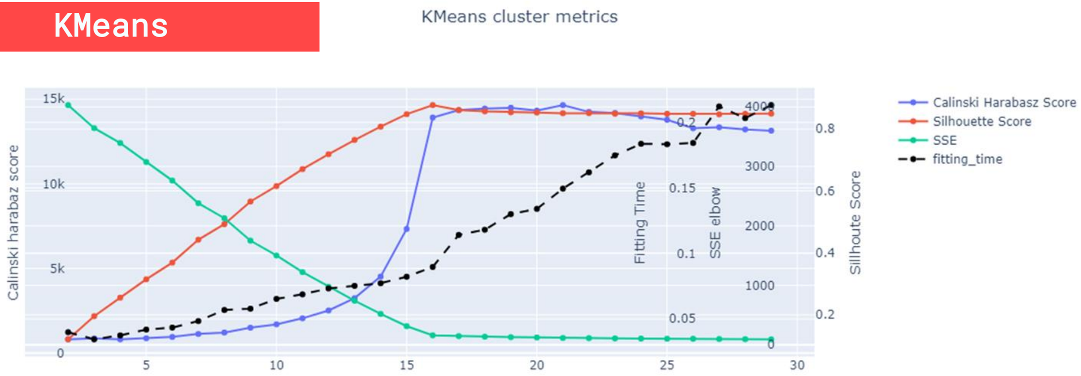
Group 3 | Clustering

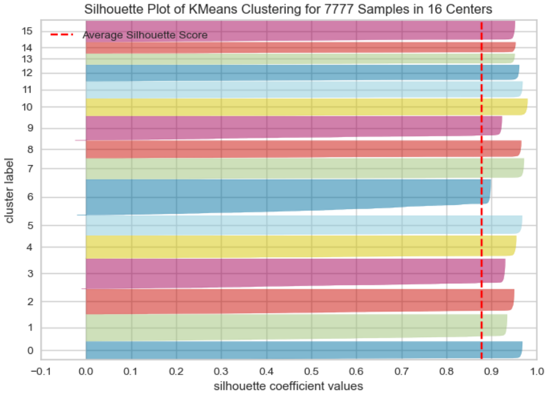
Coherence Score



By calculating coherence score we found best no. of topics are 16.

KMeans

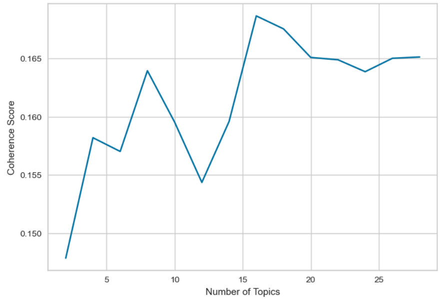




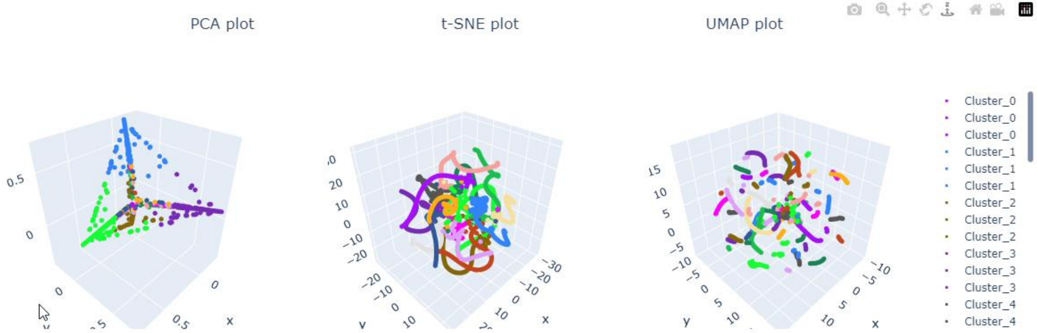
Group 3

Clustering

Coherence Score

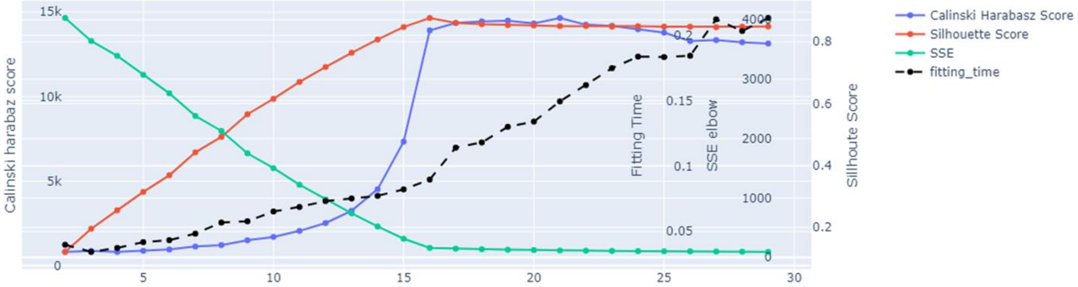


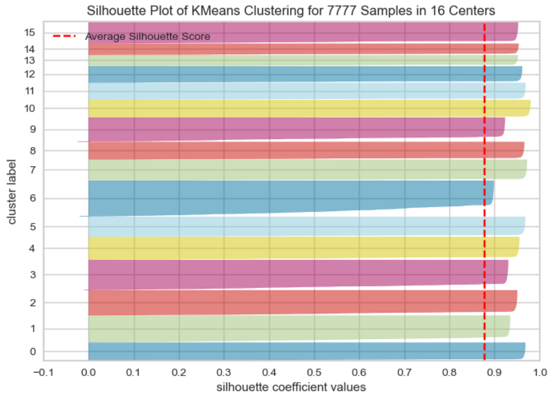
By calculating coherence score we found best no. of topics are 16.



KMeans

KMeans cluster metrics





Clustering

Group 1

Group 2

Group 3

Group 4

Group 4

Clustering

In this Group we merged all the data frames used in all the previous groups and then applied clustering on them

Group 1

1. 'director’
2. 'cast’
3. 'country’
4. 'listed_in
'

Group 2

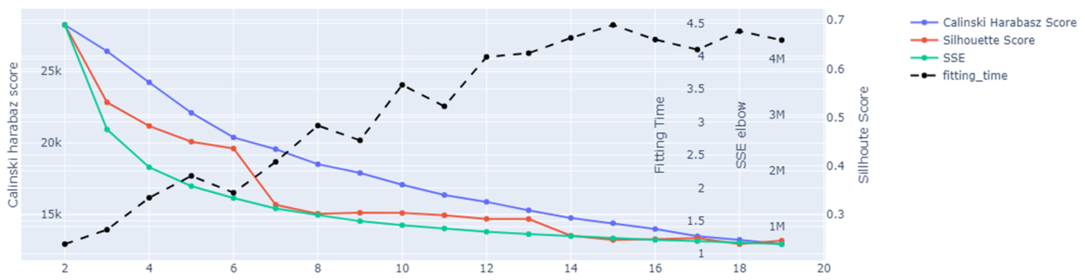
1. 'type’
2. 'release_year’
3. 'rating’
4. 'country_count’
5. 'number_of_directors’
6. 'number_of_casts’
7. 'number_of_countries’
8. 'number_of_genres'

Group 3

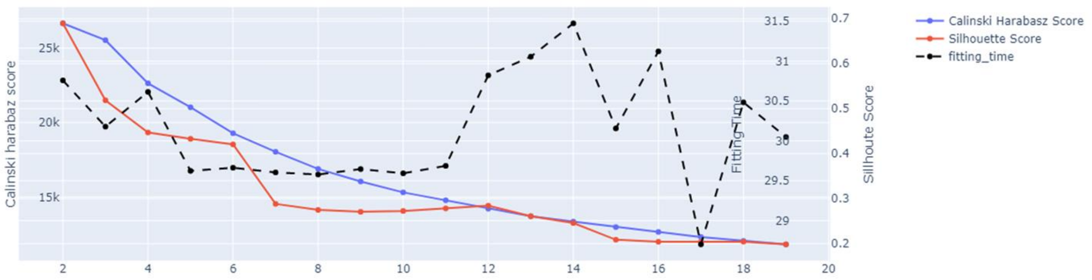
1. 'description'

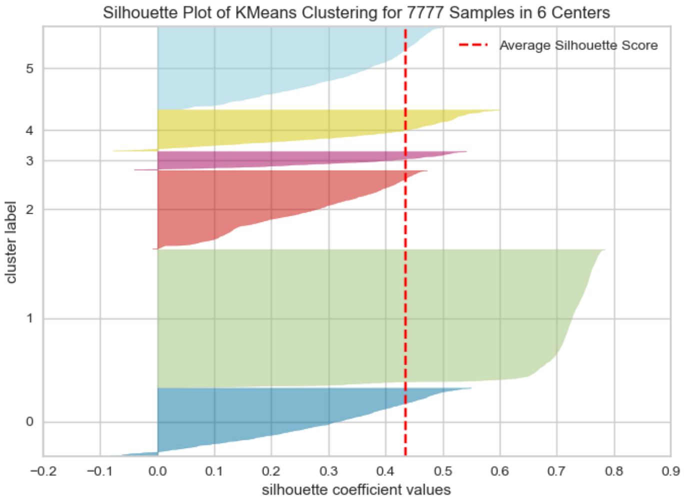
Group 4 | Clustering

KMeans



Agglomerative



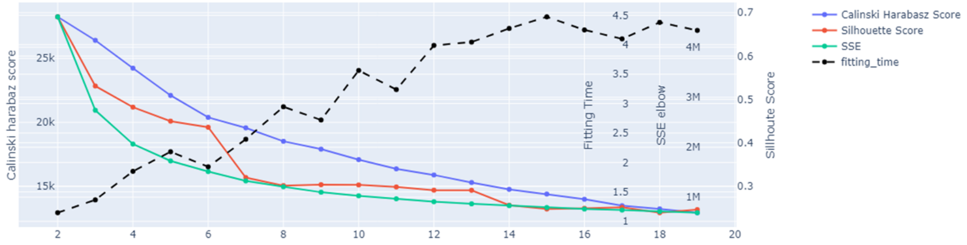


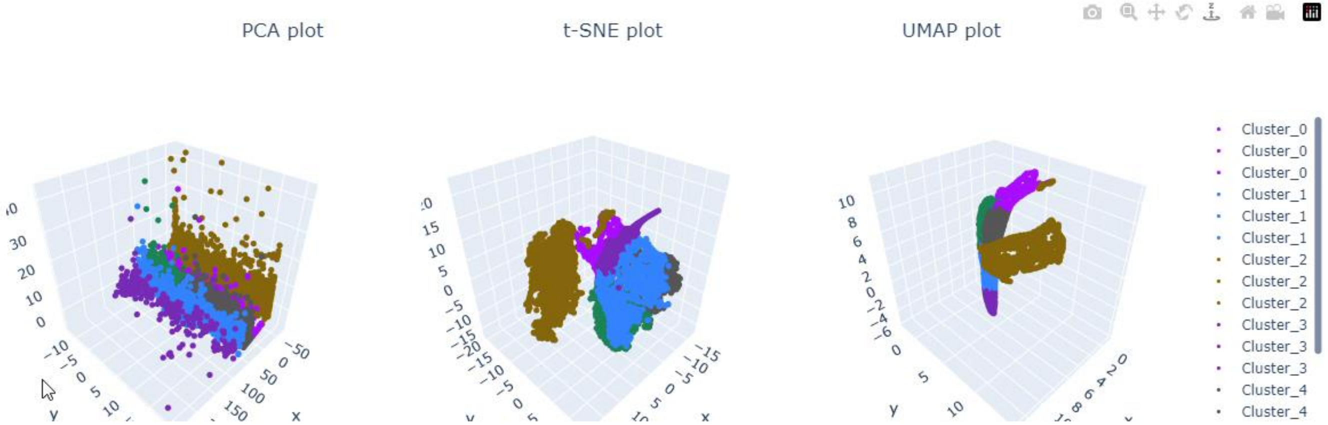
Here KMeans is performing better with vest k = 6

Group 4

Clustering

KMeans





Movie recommendation

Movie recommendation

Movie recommendation Using Cosine Similarity and defined cluster for filtering

This function takes in the following arguments:

1. netflix_df : a pandas dataframe containing information about the Netflix movies or shows.
2. Model : a trained Doc2Vec model from the gensim library
3. Title : a string representing the title of a movie or show in the netflix_df dataframe
4. cluster_column : a string representing the name of the column in netflix_df that contains the cluster labels for each movie or show
5. top_num : an integer representing the number of most similar movies or shows to return
6. filter_by_cluster : a boolean flag indicating whether to only search for similar texts within the same cluster label as the input text

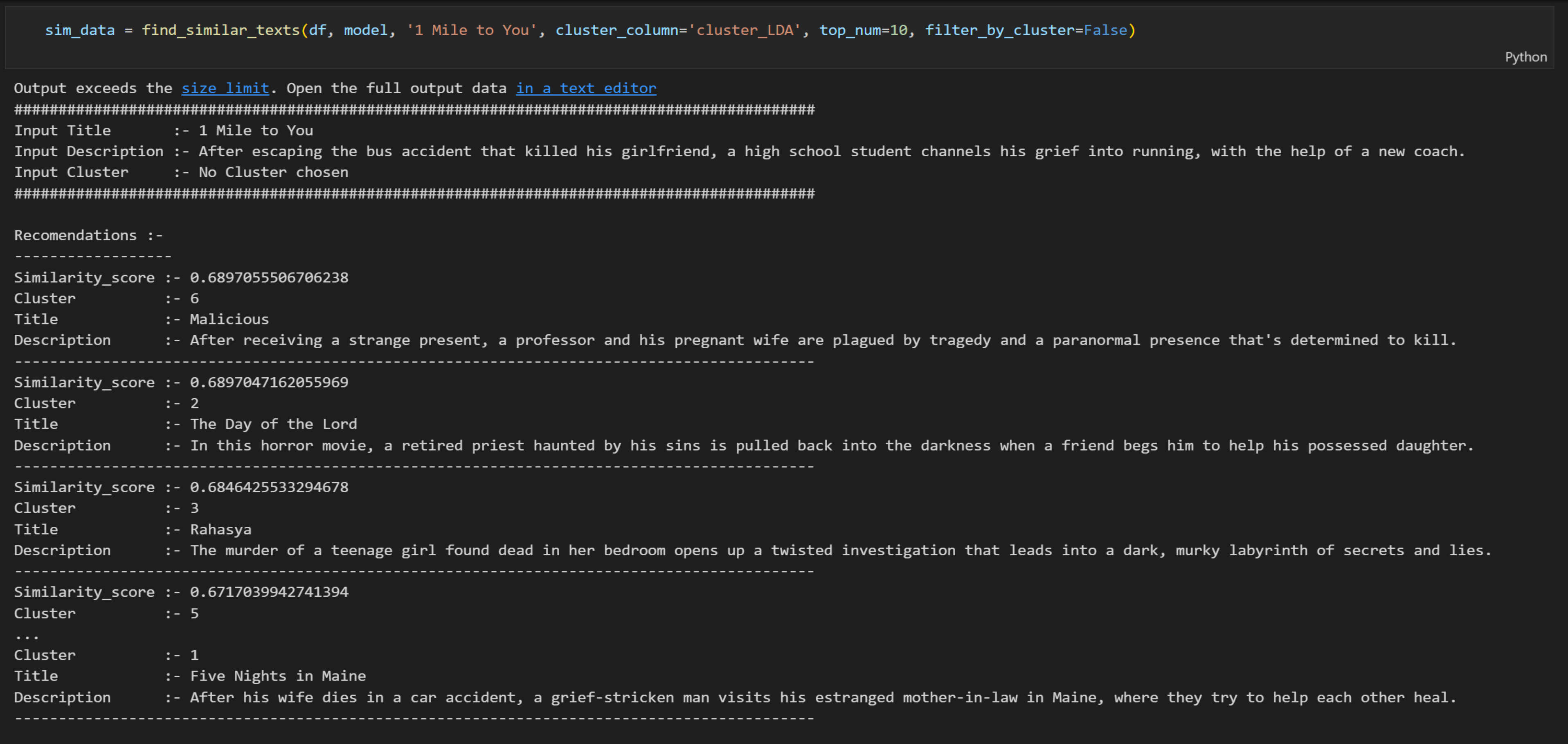
Movie recommendation

Movie recommendation Using Cosine Similarity and defined cluster for filtering

The function works as follows:

1. Retrieve the description of the input text based on the provided title.
2. Use the trained Doc2Vec model to infer the vector representation of the input text.
3. If filter_by_cluster is set to True, the function will find the cluster label of the input text and filter the netflix_df dataframe to only include texts with the same cluster label.
4. The function then uses the most_similar method from the model's Docvecs object to find similar texts. If filter_by_cluster is set to True, the search for similar texts will be limited to the same cluster as the input text, otherwise the search will be across all texts in the netflix_df dataframe.
5. The function returns the top_num most similar texts along with the cluster label and their descriptions, and prints a summary of the input text, its cluster label, and the most similar texts with their descriptions, similarity scores, and cluster labels.

Movie recommendation



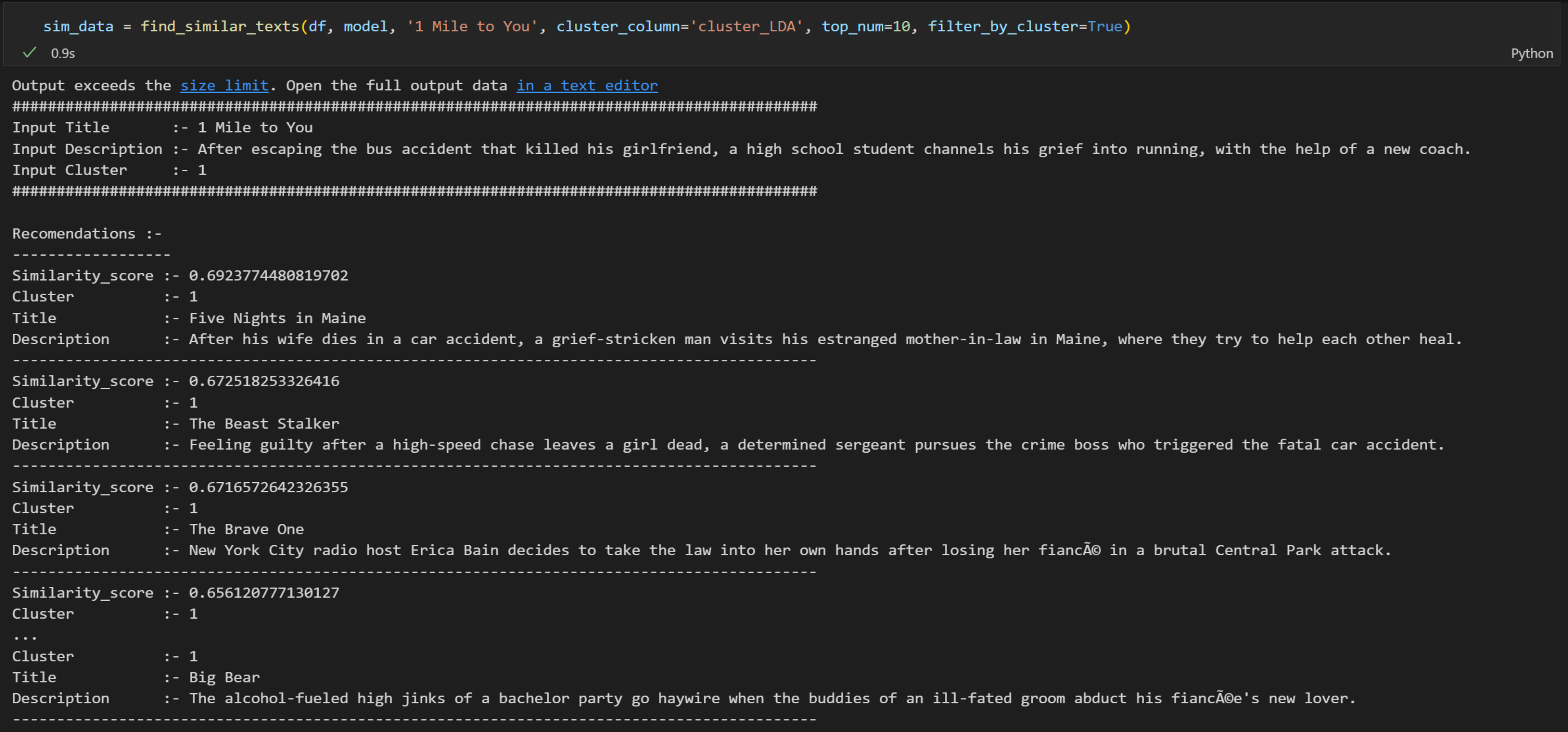
```python
sim_data = find_similar_texts(df, model, '1 Mile to You', cluster_column='cluster_LDA', top_num=10, filter_by_cluster=False)
```

Output exceeds the size limit. Open the full output data in a text editor

```
###########################################################################################
Input Title       :- 1 Mile to You
Input Description :- After escaping the bus accident that killed his girlfriend, a high school student channels his grief into running, with the help of a new coach.
Input Cluster     :- No Cluster chosen
###########################################################################################

Recomendations :-
------------------
Similarity_score :- 0.6897055506706238
Cluster          :- 6
Title            :- Malicious
Description      :- After receiving a strange present, a professor and his pregnant wife are plagued by tragedy and a paranormal presence that's determined to kill.
-------------------------------------------------------------------------------------------
Similarity_score :- 0.6897047162055969
Cluster          :- 2
Title            :- The Day of the Lord
Description      :- In this horror movie, a retired priest haunted by his sins is pulled back into the darkness when a friend begs him to help his possessed daughter.
-------------------------------------------------------------------------------------------
Similarity_score :- 0.6846425533294678
Cluster          :- 3
Title            :- Rahasya
Description      :- The murder of a teenage girl found dead in her bedroom opens up a twisted investigation that leads into a dark, murky labyrinth of secrets and lies.
-------------------------------------------------------------------------------------------
Similarity_score :- 0.6717039942741394
Cluster          :- 5
...
Cluster          :- 1
Title            :- Five Nights in Maine
Description      :- After his wife dies in a car accident, a grief-stricken man visits his estranged mother-in-law in Maine, where they try to help each other heal.
-------------------------------------------------------------------------------------------
```

Movie recommendation



```
sim_data = find_similar_texts(df, model, '1 Mile to You', cluster_column='cluster_LDA', top_num=10, filter_by_cluster=True)
```

✓ 0.9s

Python

```
Output exceeds the size limit. Open the full output data in a text editor
###########################################################################################
Input Title       :- 1 Mile to You
Input Description :- After escaping the bus accident that killed his girlfriend, a high school student channels his grief into running, with the help of a new coach.
Input Cluster     :- 1
###########################################################################################

Recomendations :-
------------------
Similarity_score :- 0.6923774480819702
Cluster          :- 1
Title            :- Five Nights in Maine
Description      :- After his wife dies in a car accident, a grief-stricken man visits his estranged mother-in-law in Maine, where they try to help each other heal.
-------------------------------------------------------------------------------------------
Similarity_score :- 0.672518253326416
Cluster          :- 1
Title            :- The Beast Stalker
Description      :- Feeling guilty after a high-speed chase leaves a girl dead, a determined sergeant pursues the crime boss who triggered the fatal car accident.
-------------------------------------------------------------------------------------------
Similarity_score :- 0.6716572642326355
Cluster          :- 1
Title            :- The Brave One
Description      :- New York City radio host Erica Bain decides to take the law into her own hands after losing her fiancÃ© in a brutal Central Park attack.
-------------------------------------------------------------------------------------------
Similarity_score :- 0.656120777130127
Cluster          :- 1
...
Cluster          :- 1
Title            :- Big Bear
Description      :- The alcohol-fueled high jinks of a bachelor party go haywire when the buddies of an ill-fated groom abduct his fiancÃ©e's new lover.
-------------------------------------------------------------------------------------------
```

Conclusion

Conclusion

1. In conclusion, the Netflix dataset is a rich resource for various types of analyses, such as exploratory data analysis and clustering.

2. Through our exploratory data analysis, we gained insights into the distribution of the dataset across various features such as type, directors, cast, country, date added, release year, rating, duration, genre, title, and description.

3. We also carried out clustering on the dataset and discovered that there are several ways to group the shows based on their features. This can be useful in creating a recommendation system for Netflix viewers. The analysis revealed interesting patterns and trends in the Netflix dataset, which can be used to enhance the user experience on the platform.

Q & A