Machine Learning
Capstone Clustering Project

Netflix Movies and

TV Shows

Fully EXplained

By
Diksha Shejao

CONTENT:

Problem statement:

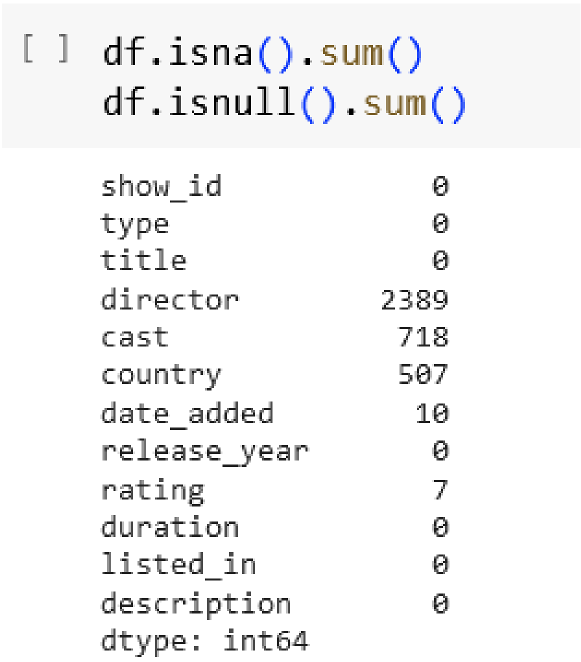
This dataset contains TV shows and movies available on Netflix as of 2019. The dataset is collected from Flexible which is a third-party Netflix search engine. In 2018 , they released an interesting report which shows that the number of TV shows on Netflix has nearly tripled since 2010. The streaming service's number of movies has decreased by more than 2,000 titles since 2010, while its number of TV shows has nearly tripled. IT will be interesting to explore what all other insights can be obtained from the same dataset.

Integrating this dataset with other external datasets such as IMDB ratings, rotten tomatoes can also provide many interesting findings.

Data Description:

1.show_id :   Unique ID for every Movie / Tv Show
2.type : Identifier - A Movie or TV Show
3.title : Title of the Movie / Tv Show
4.director : Director of the Movie
5.cast : Actors involved in the movie / show
6.country : Country where the movie / show was produced
7.date_added : Date it was added on Netflix
8.release_year : Actual Release Year of the movie / show
9.rating : TV Rating of the movie / show
10.duration : Total Duration - in minutes or number of seasons
11.listed_in: Genre
12.description: The Summary description

Null Value:

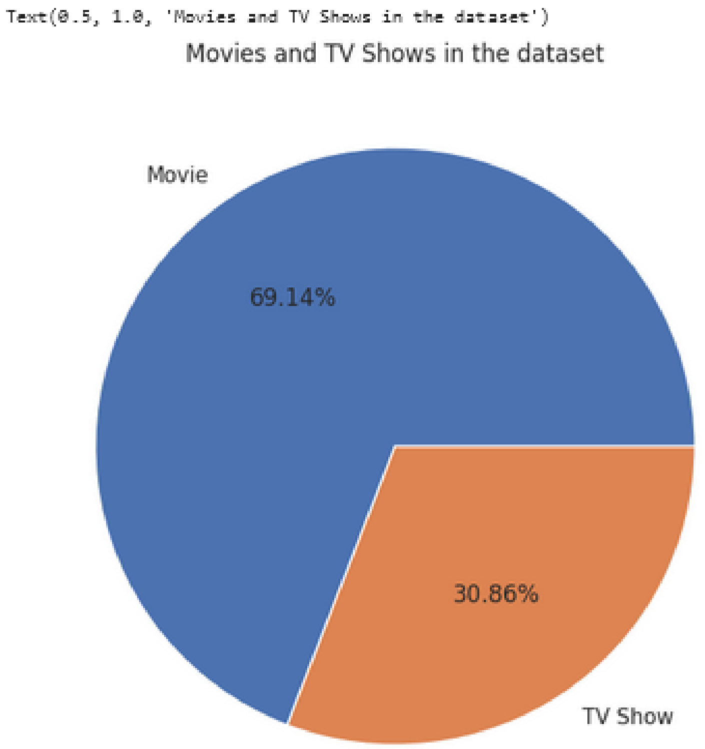


```
[ ]  df.isna().sum()
     df.isnull().sum()
```

```
show_id            0
type               0
title              0
director        2389
cast             718
country          507
date_added        10
release_year       0
rating             7
duration           0
listed_in          0
description        0
dtype: int64
```

- This dataset contains 7787 rows and 12 columns.
- Speaking about missing values , here around 5 columns contain null values.
- One column director has huge missing values.
- There is no null value in this dataset
- There are many missing values in director, cast, country, date_added, and rating columns.

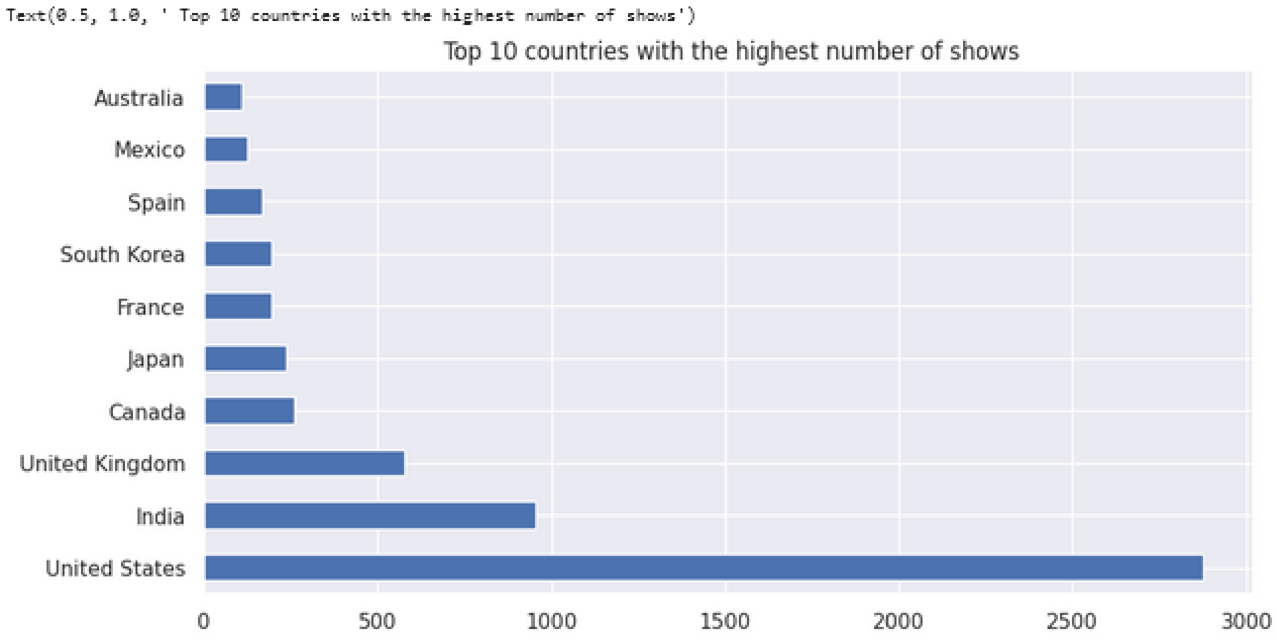
EDA:



```
Text(0.5, 1.0, 'Movies and TV Shows in the dataset')
```

Type of content available on Netflix

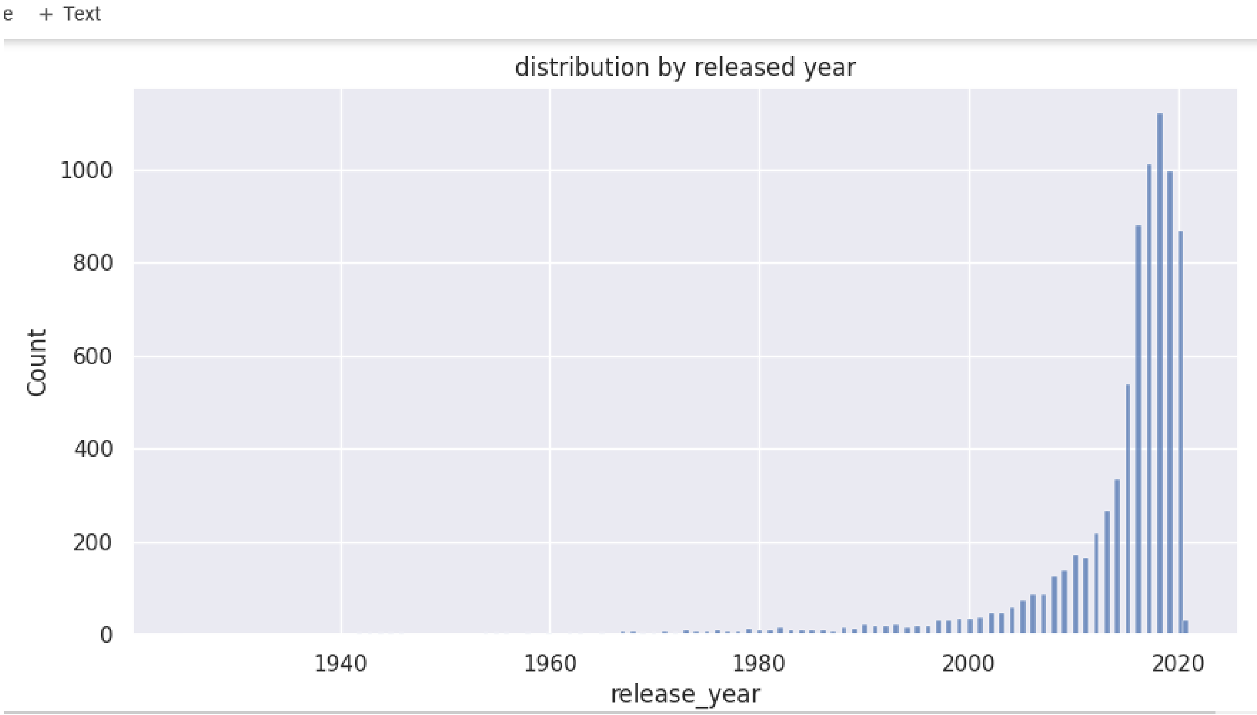
- It is evident that there are more movies on Netflix than TV shows
- There are more movies (69.14%) than TV shows (30.86%) in the dataset.

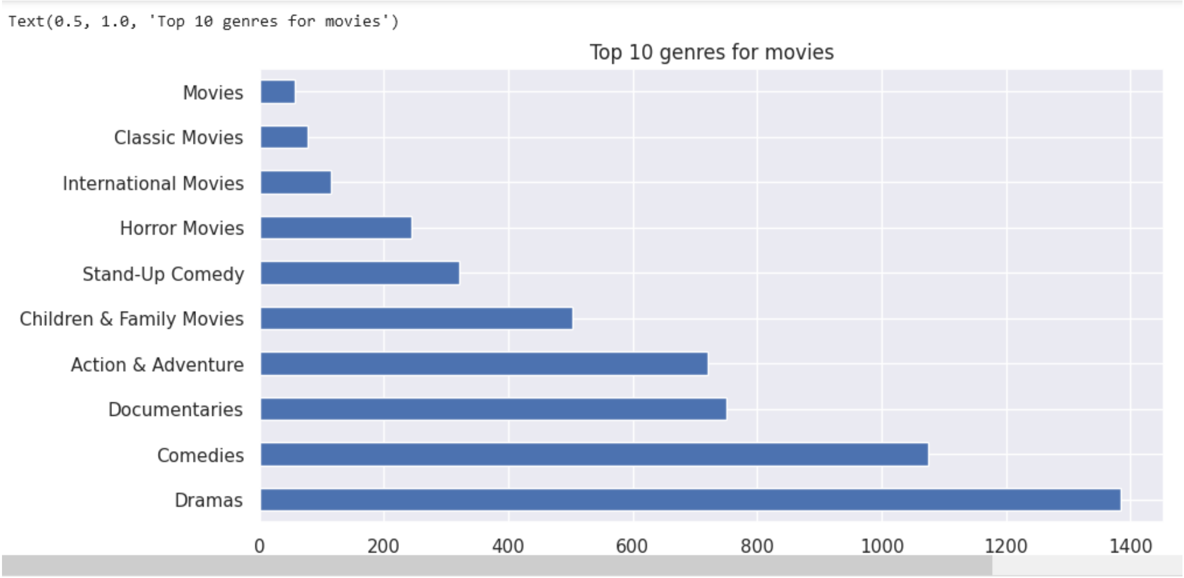


```
Text(0.5, 1.0, ' Top 10 countries with the highest number of shows')
```

- United States has the most number of content on netflix
- India has second highest content on Netflix
- Australia and Mexico has least number of content on netflix.

Distributed by released year





```
Text(0.5, 1.0, 'Top 10 genres for movies')
```

Dramas, comedies, and documentaries are the most popular genres for the movies on Netflix.

Data preprocessing:

Modelling Approach:

Select the attributes based on which you want to cluster the shows

Text preprocessing: Remove all non-ascii characters, stopwords and punctuation marks, convert all textual data to lowercase.

Lemmatization to generate a meaningful word out of corpus of words

We will cluster the shows on Netflix based on the following attributes:

Director

Cast

Country

Listed in (genres)

Description

Removing non_ASCII character

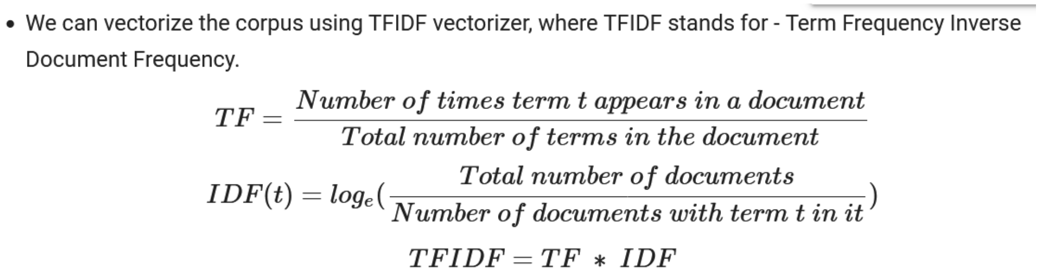
Removing stop words and lower case

Removing punctuations

Lemmatization

vectorization

* We can vectorize the corpus using TFIDF vectorizer, where TFIDF stands for - Term Frequency Inverse Document Frequency.



- We can vectorize the corpus using TFIDF vectorizer, where TFIDF stands for - Term Frequency Inverse Document Frequency.

$$TF = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

$$IDF(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it})$$

$$TFIDF = TF\ *\ IDF$$

Dimensionality reduction using pca

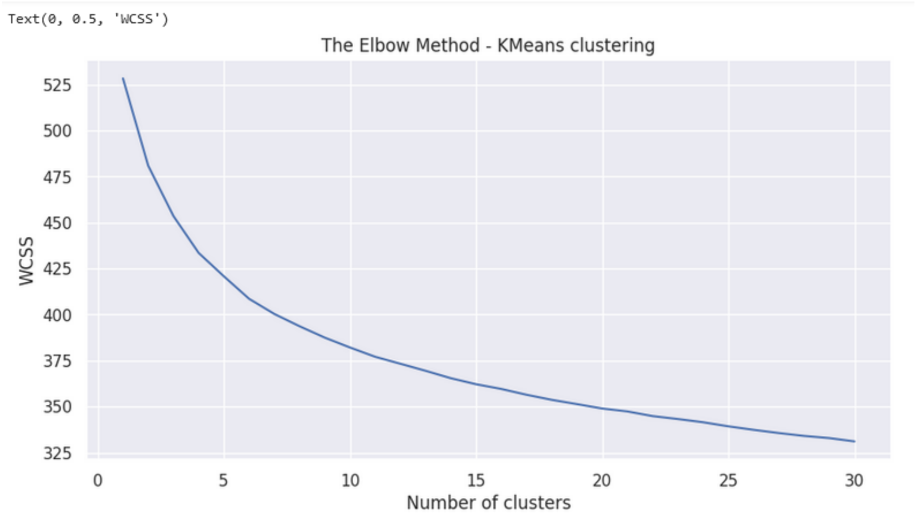
We can use PCA (Principal component Analysis) to reduce the dimensionality of data.

Model implementation:

Clusters implementation

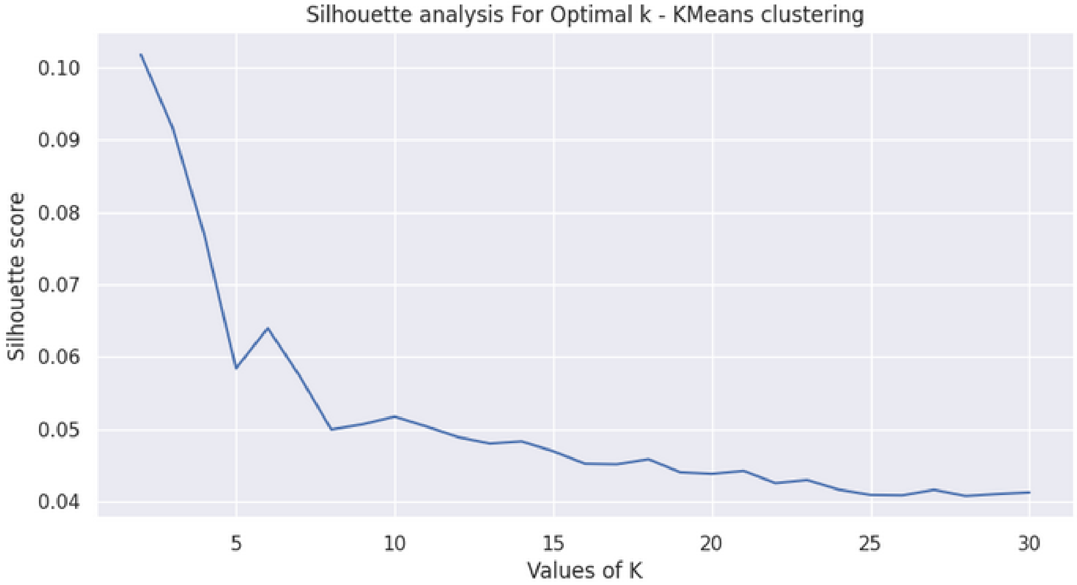
K-means clustering :

Building clusters using the K-means clustering algorithm Visualizing the elbow curve and Silhouette score to decide on the optimal number of clusters for K-means clustering algorithm.



```
Text(0, 0.5, 'WCSS')
```

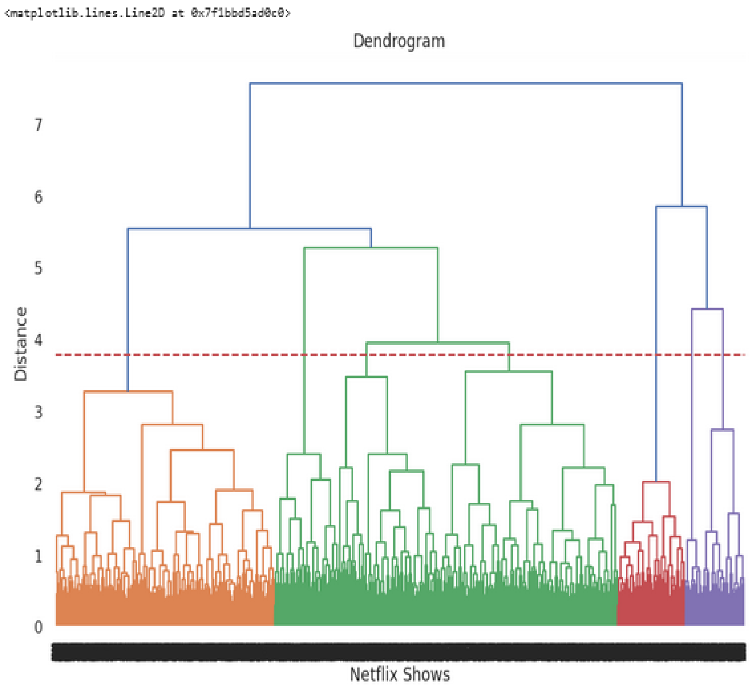
The sum of squared distance between each point and the centroid in a cluster (WCSS) decreases with the increase in the number of clusters.



The highest Silhouette score is obtained for 6 clusters.

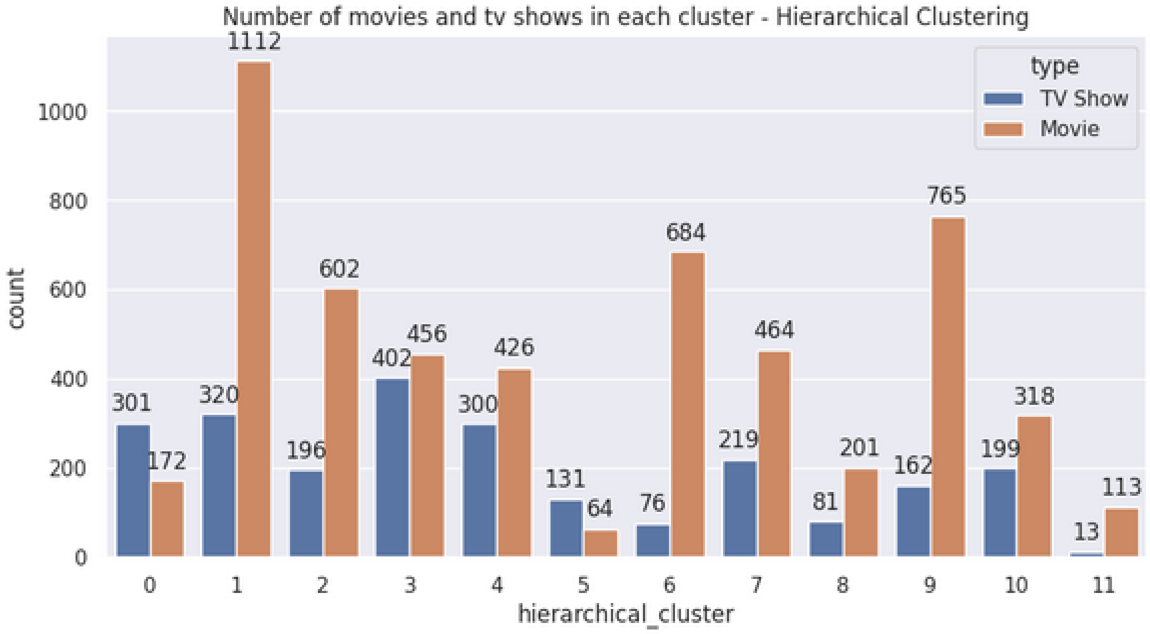
Hierarchical clustering:

Building clusters using the agglomerative (hierarchical) clustering algorithm.Visualizing the dendrogram to decide on the optimal number of clusters for the agglomerative (hierarchical) clustering algorithm:



```
<matplotlib.lines.Line2D at 0x7f1bbd5ad0c0>
```

At a distance of 3.8 units, 12 clusters can be built using the agglomerative clustering algorithm.Building 12 clusters using the Agglomerative clustering algorithm:



Successfully built 12 clusters using the Agglomerative (hierarchical) clustering algorithm.

Base Recommended system:

We can build a simple content based recommender system based on the similarity of the shows.

If a person has watched a show on Netflix, the recommender system must be able to recommend a list of similar shows that s/he likes.

To get the similarity score of the shows, we can use cosine similarity

The similarity between two vectors (A and B) is calculated by taking the dot product of the two vectors and dividing it by the magnitude value as shown in the equation below. We can simply say that the CS score of two vectors increases as the angle between them decreases.

$\mathrm{Cos}(\theta)=A . B / |A| . |B|$

Conclusion:

- In this project, we worked on a text clustering problem where in we had to classify/group the Netflix shows into certain clusters such that the shows within a cluster are similar to each other and the shows in different clusters are dissimilar to each other.
- The dataset contained about 7787 records, and 11 attributes.
- We began by dealing with the dataset's missing values and doing exploratory data analysis (EDA).
- It was found that Netflix hosts more movies than TV shows on its platform, and the total number of shows added on Netflix is growing exponentially. Also, the majority of the shows were produced in the United States, and the majority of the shows on Netflix were created for adults and young adults age group .
- It was decided to cluster the data based on the attributes: director, cast, country, genre, and description. The values in these attributes were tokenized, preprocessed, and then vectorized using TFIDF vectorizer.
- Through TFIDF Vectorization, we created a total of 20000 attributes.
- We used Principal Component Analysis (PCA) to handle the curse of dimensionality. 4000 components were able to capture more than 80% of variance, and hence, the number of components were restricted to 4000.
- We first built clusters using the k-means clustering algorithm, and the optimal number of clusters came out to be 6. This was obtained through the elbow method and Silhouette score analysis.
- Then clusters were built using the Agglomerative clustering algorithm, and the optimal number of clusters came out to be 12. This was obtained after visualising the dendrogram.
- A content based recommender system was built using the similarity matrix obtained after using cosine similarity. This recommender system will make 10 recommendations to the user based on the type of show they watched.