Netflix Movie and TV Show Clustering

Project Type - Unsupervised (Clustering, Content Based Recommendation System)

Hark Pun,
Data Science Trainee,
AlmaBetter, Bangalore.

Abstract:

With the advent of streaming platforms, there's no doubt that Netflix has become one of the important platforms for streaming. The dataset that we have used for EDA and clustering has been collected by Flixable, a third-party Netflix search engine. There are 12 features and around 7700 observations in the dataset and are mostly textual features. Through univariate and multivariate analysis, we found trends that will help in understanding what content is being consumed country-wise, depending on some categorical features like rating, type, genres, cast, directors, etc. Clustering was performed along with NLP on textual columns and then a mini-recommendation system was built out of it.

Keywords - Explanatory Data Analysis, NLP, Clustering, Recommendation System.

Introduction:

Unsupervised Learning is a machine learning technique in which the model is not supervised by the training set instead we find hidden patterns and insight from the given data. It is a machine learning technique in which model are trained on the unlabeled data set without any supervision.
A cluster is a collection of elements that are similar to each other but dissimilar to the elements belonging to other clusters. Clustering can be doneusing various kinds of distancessuch as Euclidean distance, Manhattan distance, gomer distance, etc. We can do different kinds of clustering based on the data pattern in space such asspherical clustering, K-means clustering, etc.

1. Problem Statement:

The dataset is collected from Flixable which is a third-party Netflix search engine. Netflix is the world's largest online streaming service provider, with over 220 million subscribers as of 2022-Q2. It is crucial that they effectively cluster the shows that are hosted on their platform in order to enhance the user experience, thereby preventing subscriber churn.

We will be able to understand the shows that are similar to and different from one another by creating clusters, which may be leveraged to offer the consumers personalized show suggestions depending on their preferences.

The goal of this project is 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.

Data Description:

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

2. EDA:

EDA or Exploratory Data Analysis is the critical process of performing the initial investigation on the data to find the anomalies in our data and shape it such that it is useful for taking some insights to solve our purpose.

Data pre-processing began with the visualization of raw data using descriptive statistics tables, skewness, and other descriptions such as min, max, percentile values, and mean. It also includes the identification and removal of missing values, as well as the textual data preprocessing for clustering purpose.

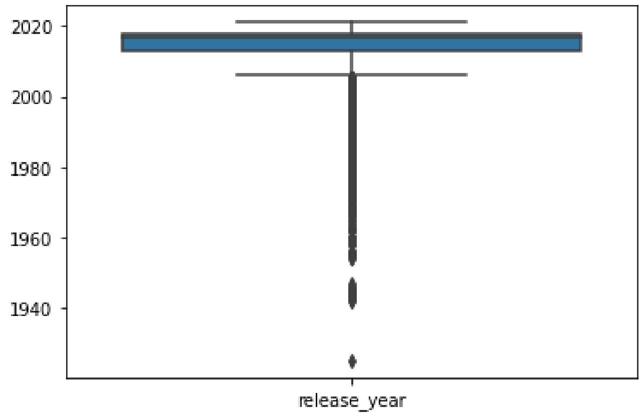
.

Let's have a look at the missing values present in our data set.

The cleaning of the data starts with the replacing of missing values using various techniques. The missing values in the director, cast, and country attributes can be replaced with 'empty string. Small amount of null value percentage present in rating and date_added column, if we drop these nan values it will not affect that much while building the model. So, we simply drop the nan value present in rating and date_added columns.

Replace missing values with empty string. Whichever feature having less than 5% missing values we have decided to drop them directly.

Further, outliers' removal technique to remove outlier from data the data using the **Capping method**.
Where Q1, Q3 represent the first quantile, the third quantile of each attribute.



Since the almost all of the data present in textual format Except release year and also the data that we need to create cluster are present in textual format. So, there is no need to perform handling outlier.

3. Modelling Approach:

1. Select the attributes based on which you want to cluster the shows
2. Text preprocessing: Remove all stop words and punctuation marks, convert all textual data to lowercase.
3. Stemming to generate a meaningful word out of corpus of words.
4. Tokenization of corpus and Word vectorization
5. Dimensionality reduction
6. Use different algorithms to cluster the movies, obtain the optimal number of clusters using different techniques.
7. Build optimal number of clusters and visualize the contents of each cluster using word clouds

Creating Cluster:
We create one cluster column based on the following features:

- Director
- Cast
- Country
- Rating
- Listed in (genres)
- Description

Before clusters implementation we need to pre-process the data. So that we filtered data with following steps:

4. Textual Data Pre-processing:

1. **Removing Stop words**
    - Stop words are common words like "the", "and" and "but" do not carry much meaning on their own and are often seen as noise in the data.

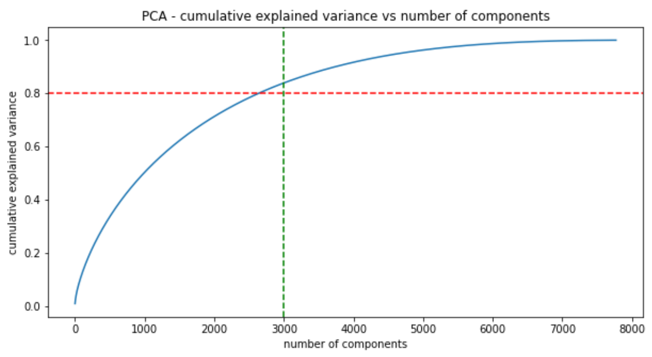
2. **Lowercasing words**
    - Lowercasing the words can also reduce the size of the vocabulary, which can make it easier to work with larger texts or texts in languages with a high number of inflected forms.

3. **Removing Punctuation**
    - Punctuation marks like periods, commas, and exclamation points can add noise to the data and can sometimes be treated as separate tokens, which can affect the performance of NLP models.

4. **Stemming**
    - used Snowball Stemmer to generate a meaningful word out of corpus of words.
    - For example, the words "run," "runs," "ran," and "running" are all different inflected forms of the same word "run," and a stemmer can reduce them all to the base form "run."

5. **Tokenization of corpus and Word vectorization** – TFIDF
    - This is important in NLP tasks because most machine learning models expect numerical input and cannot work with raw text data directly. Word vectorization allows you to input the words into a machine learning model in a way that preserves the meaning and context of the words.

6. **Dimensionality reduction** – PCA
    - Dimensionality reduction is the process of reducing the number of features or dimensions in a dataset while preserving as much information as possible. As high-dimensional datasets can be difficult to work with and can sometimes suffer from the curse of dimensionality.

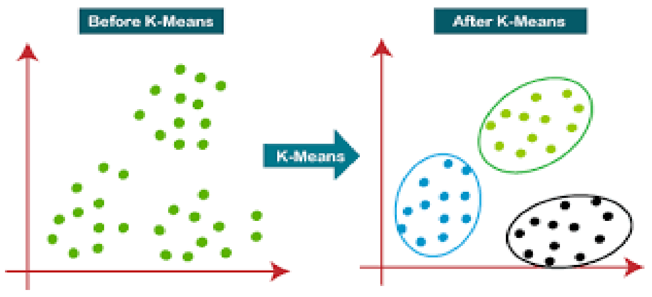


- As you can see that 100% of the variance is explained by about ~7500 components.
- Also, more than 80% of the variance is explained just by 3000 components.
- Hence to simplify the model, and reduce dimensionality, we take top 3000 components, which will still be able to capture more than 80% of variance.

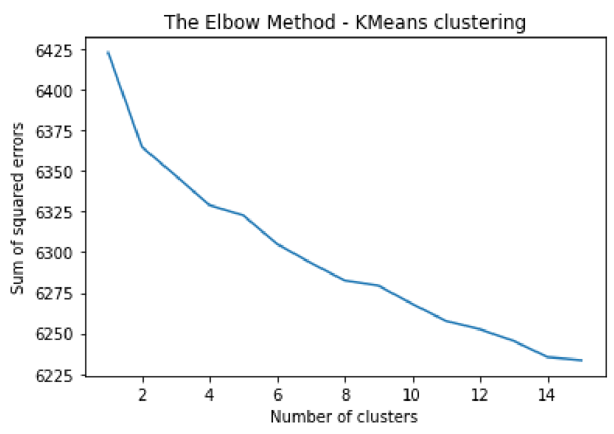
5. Clustering Algorithms:

K-Means Clustering:

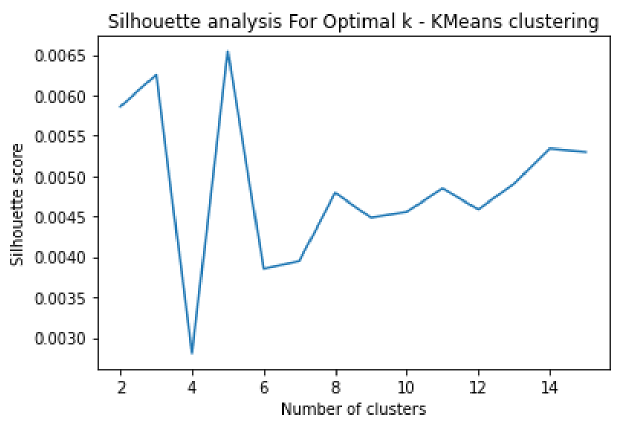
k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.



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



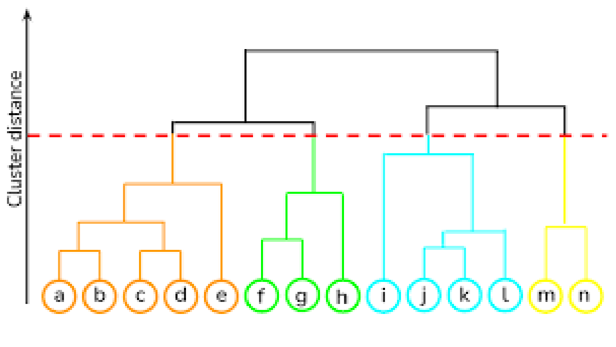
- The sum of squared errors between each point and the centroid in a cluster decreases with the increase in the number of clusters



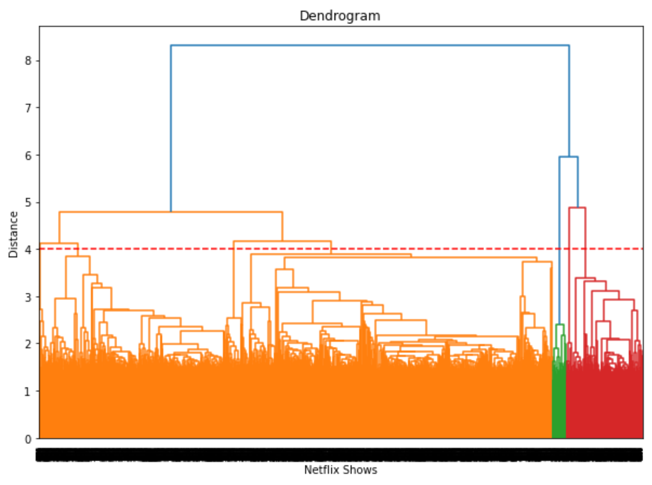
- The highest Silhouette score is obtained for 5 clusters.
- Building 5 clusters using the k-means clustering algorithm

Hierarchical clustering:

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other

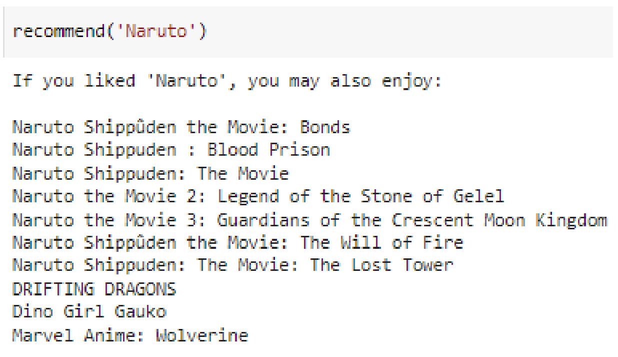


- Visualizing the dendrogram to decide on the optimal number of clusters for the agglomerative (hierarchical) clustering algorithm.
- At a distance of 4 units, 7 clusters can be built using the agglomerative clustering algorithm.



6. Recommendation System:

- We can build a simple content-based recommender system based on the similarity of the movie/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. We can simply say that the CS score of two vectors increases as the angle between them decreases.



```
recommend('Naruto')
```

```
If you liked 'Naruto', you may also enjoy:

Naruto Shippûden the Movie: Bonds
Naruto Shippuden : Blood Prison
Naruto Shippuden: The Movie
Naruto the Movie 2: Legend of the Stone of Gelel
Naruto the Movie 3: Guardians of the Crescent Moon Kingdom
Naruto Shippûden the Movie: The Will of Fire
Naruto Shippuden: The Movie: The Lost Tower
DRIFTING DRAGONS
Dino Girl Gauko
Marvel Anime: Wolverine
```

7 Conclusion:

In this project, we worked on a text clustering problem wherein 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 decided to cluster the data based on the attributes: director, cast, country, genre, rating and description. The values in these attributes were tokenized, preprocessed, and then vectorized using TFIDF vectorizer.
- Through TFIDF Vectorization, we created a total of 10000 attributes.
- We used Principal Component Analysis (PCA) to handle the curse of dimensionality. 3000 components were able to capture more than 80% of variance, and hence, the number of components were restricted to 3000.
- We first built clusters using the K-Means Clustering algorithm, and the optimal number of clusters came out to be 5. 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 7. This was obtained after visualizing 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.

8. References:

- https://towardsdatascience.com/
- https://www.analyticsvidhya.com/blog/
- https://github.com/rushter/data-science-blogs
- https://www.kaggle.com/
- https://stackoverflow.com/