Technical Documentation
CAPSTONE PROJECT
ON
Netflix Movie and TV Show Clustering

Abstract:

Without a question, Netflix has emerged as one of the key streaming platforms since the emergence of these services. A third-party Netflix search engine called Flexible gathered the dataset that we used for EDA and clustering. The dataset contains approximately 7787 observations and 12 features, the majority of which are textual features. Textual columns underwent clustering and NLP, and a mini-recommendation system was developed from it.

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

Introduction:

Unsupervised learning is a machine learning technique in which we extract insights and hidden patterns from the provided data rather than having the model be supervised by the training set. It is a method of machine learning where models are trained on sets of unlabeled data without any guidance. A cluster is a collection of items that are distinct from the elements found in other clusters but related to one another. Different types of distances, including the Euclidean distance, Manhattan distance, Gomer distance, etc., can be used to cluster data. Based on the spatial distribution of the data, we may do several types of grouping, including spherical clustering, K-means clustering, etc.

1. Problem Statement:

The dataset is collected from Flexible 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 personalised 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:

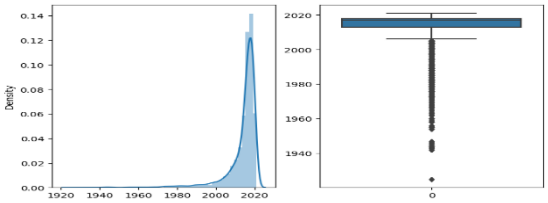
Exploratory Data Analysis, often known as EDA, is a crucial step in doing the initial research on the data in order to identify abnormalities and transform it so that it may be used to gain some insights into how to accomplish our goal.

Beginning with the visualisation of the raw data using descriptive statistics tables, skewness, and other descriptions like min, max, percentile values, and mean, the pre-processing of the data began. Additionally, it includes textual data preprocessing for clustering and the identification and removal of missing values.

Let's examine the missing values that are part of 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 unknown, no caste string and mode values in place of any missing values. Any feature with fewer than 5% missing values has been immediately dropped.

Additionally, the data using the Capping method to remove outliers from the data.



- The figures (release_year less than 2009) are being displayed as outliers
- Replacing outliers with mean value
- we don't have have any release_year which is greater than 2018

3. Modelling Approach:

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

Creating Cluster:

We build a single cluster column using the following criteria:
• Director
• Cast
• Country

- Rating
- Listed in (genres)
- Description

We must preprocess the data before implementing clusters. In order to do this, we took the 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

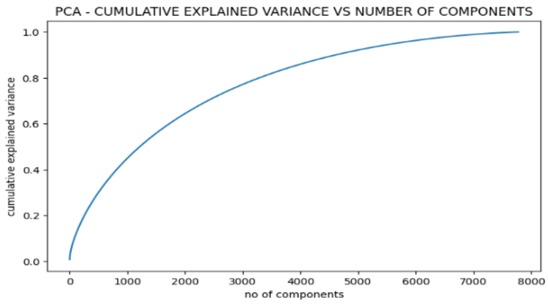
- used Snowball Stemmer to generate a meaningful word out of a corpus of words.
- For example, the words "run," "runs," "ran," and "running" are all different inflected forms of the same word "run," and a stemmer can reduce them all to the base from "run."

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.



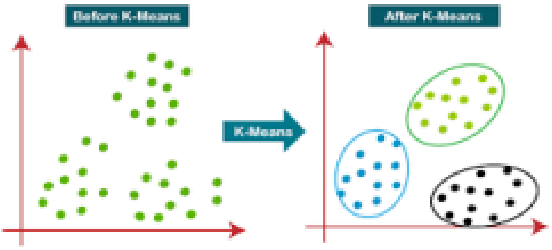
- We find that around 7500 components account for 100% of the variance
- Also, just 4000 components comprise more than 80% of the variation
- As a result, we can pull the top 4000 components out of the model to make it simpler and less dimensional while still

being able to account for more than 80% of variance.

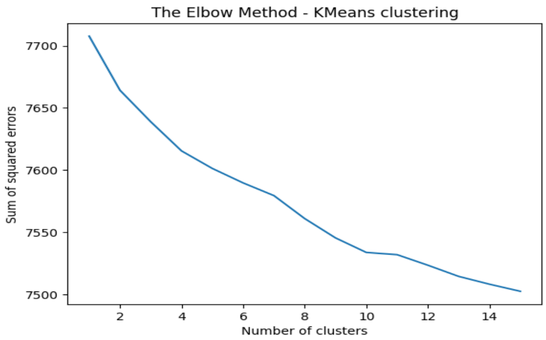
5. Clustering Algorithms:

K-Means Clustering:

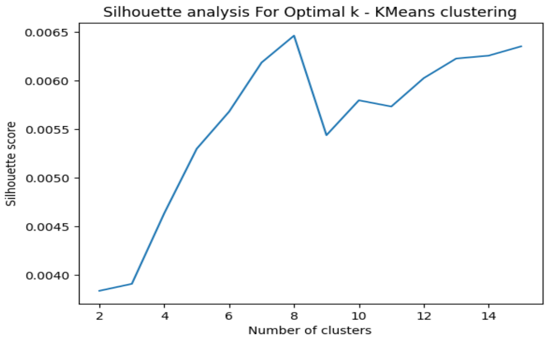
The goal of the vector quantization technique known as "k-means clustering," which has its roots in signal processing, is to divide a set of n observations into k clusters, each of which has as its prototype the observation with the closest mean**.**



Using visualisation, the K-means clustering algorithm determines the ideal number of clusters by examining the elbow curve and Silhouette score.



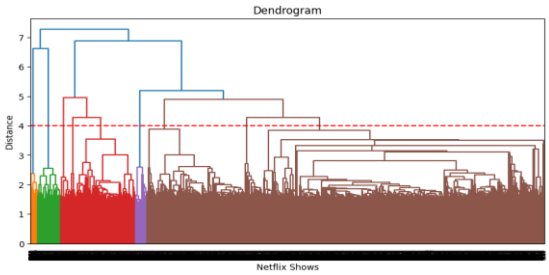
- As the number of clusters increases, the sum of squared errors between each point and the centroid in the cluster decreases.



- Using the k-means clustering technique, six clusters were built.

Hierarchical clustering:

- An algorithm called hierarchical clustering, commonly referred to as hierarchical cluster analysis, divides objects into clusters based on how similar they are. The endpoint is a collection of clusters, each of which differs from the others but having objects that are largely similar to one another.
- Using the dendrogram to determine the agglomerative (hierarchical) clustering algorithm's ideal number of clusters.



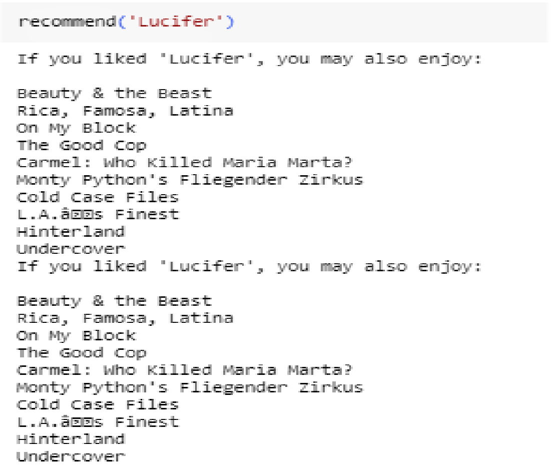
6. Recommendation System:

A simple content-based recommender system based on the similarities of the film or series can be created.

- If someone has seen a Netflix programme, the recommender system should be able to provide a list of similarly themed programmes.

- 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('Lucifer')
```

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

Beauty & the Beast
Rica, Famosa, Latina
On My Block
The Good Cop
Carmel: Who Killed Maria Marta?
Monty Python's Fliegender Zirkus
Cold Case Files
L.A.â⊠⊠s Finest
Hinterland
Undercover
If you liked 'Lucifer', you may also enjoy:

Beauty & the Beast
Rica, Famosa, Latina
On My Block
The Good Cop
Carmel: Who Killed Maria Marta?
Monty Python's Fliegender Zirkus
Cold Case Files
L.A.â⊠⊠s Finest
Hinterland
Undercover
```

<u>7 Conclusion:</u>

The Netflix shows were to be categorised or grouped into specific clusters in this project's text clustering task so that the shows within a cluster are similar to one another and the shows in various clusters are dissimilar from one another.

- The dataset contained about 7787 records, and 12 attributes. We began by dealing with the dataset's missing values and doing exploratory data analysis (EDA).
- The selection was made to group the data according to the director, cast, country, genre, rating, and description. These attributes' values underwent tokenization, preprocessing, and vectorization using TFIDF vectorizer.
- For the purpose of overcoming the dimensionality curse, we applied Principal Component Analysis (PCA). The number of components was limited to 4000 because they may account for more than 80% of the variance.
- The ideal number of clusters was found to be six when we first constructed clusters using the K-Means Clustering technique. The elbow technique and a score analysis of Silhouette data were used to accomplish this.
- The similarity matrix created after employing cosine similarity was used to construct a content-based recommender system. Based on the kind of show the user viewed, this recommender system will give them ten recommendations.

<u>Project by</u>

Sandip Dey