**Netflix Movies and TV Shows Clustering**

**Project Type - Unsupervised Clustering and Recommendation System**

**Contribution - Individual**

**Navneet Keshri**

**Data Science Trainee**

**AlmaBetter, Bangalore**

**Abstract:**

Netflix is without a doubt one of the most important streaming platforms now that streaming platforms have emerged. Flixable, a third-party Netflix search engine, collected the dataset that we used for EDA and clustering. The majority of the 12 features in the dataset are textual, with approximately 7700 observations. We discovered trends through univariate and multivariate analysis that will assist in comprehending the content consumed country-by-country based on categorical features like rating, type, genres, cast, and directors, among others. Clustering and natural language processing were used on textual columns to create a mini-recommendation system.

Keywords - Explanatory Data Analysis, Natural Language Processing, Clustering, and a Recommendation System.

**Introduction:**

Unsupervised learning is a method of machine learning in which we discover hidden patterns and insights from the given data rather than having the training set supervise the model. Models are trained without supervision on the unlabelled data set using this machine-learning technique.

A cluster is a collection of elements that share some similarities but are not the same as those in other clusters. Various distances, such as the Euclidean distance, the Manhattan distance, the Gomer distance, and so on, can be utilized in clustering. Based on the data pattern in space, we can perform various types of clustering, such as K-means clustering and spherical clustering.

**Problem Statement:**

The dataset is gathered from Flixable which is an outsider Netflix web crawler. Netflix is the world's biggest web-based real-time feature supplier, with more than 220 million endorsers starting around 2022-Q2.

It is vital that they successfully group the shows that are facilitated on their foundation to upgrade the client experience, in this way forestalling supporter beat.

Netflix is a streaming service that offers a wide variety of television shows and movies for viewers to watch at their convenience. With a monthly subscription, users have access to a vast library of content, including original series and films produced by Netflix. The platform also allows users to create multiple profiles, making it easy for family members or roommates to have their own personalized viewing experience. Additionally, Netflix allows users to download content to watch offline, making it a great option for those who travel frequently or have limited internet access. Overall, Netflix is a convenient and cost-effective way to access a wide variety of entertainment.

By creating clusters, we will be able to comprehend the shows that are alike and different from one another. These clusters can be used to provide customers with individualized show recommendations based on their preferences.

This project aims to classify and group Netflix shows into specific clusters in such a way that shows in the same cluster are similar to one another and shows in different clusters are different.

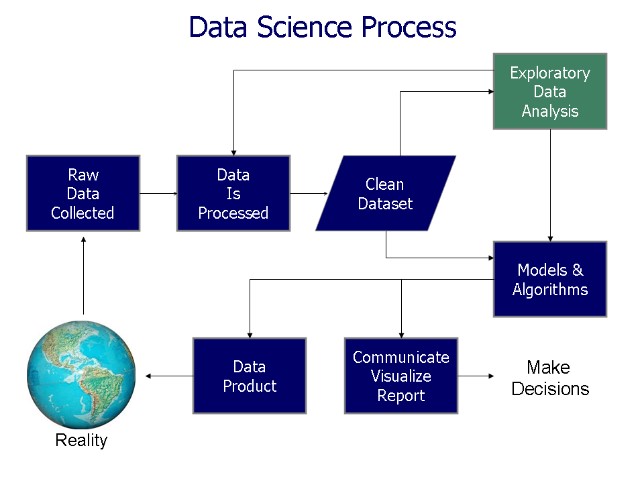
The objective of this undertaking is to order/bunch the Netflix shows into specific bunches with the end goal that the shows inside a bunch are like one another and the

**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: The 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

## **Exploratory Data Analysis:**

## Exploratory data analysis (EDA) is a method in statistics that uses statistical graphics and other data visualization techniques to summarize the main characteristics of data sets. A measurable model can be utilized or not, yet basically, EDA is for seeing everything that the information can say to us past the proper demonstration and in this manner contrasts conventional speculation testing. We were able to determine various aspects and relationships between the target and the independent variables with the assistance of EDA.



The first step in data pre-processing was to use descriptive statistics tables, skewness, and other descriptions like min, max, percentile values, and mean to show the raw data. It also includes textual data preprocessing for clustering and the detection and elimination of missing values.

Let's take a look at the values that are missing from our data set.

The first step in data cleaning is to use a variety of methods to replace missing values. "empty string" can be used to replace the values that are missing from the director, cast, and country attributes.

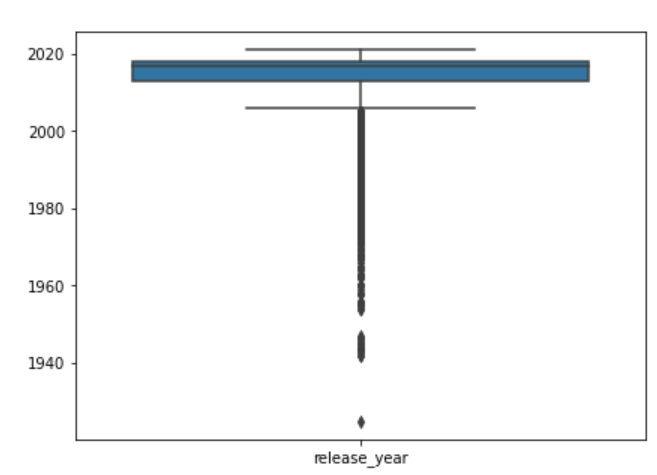
There is a small percentage of null values in the rating and date\_added columns; removing these nan values will have little effect on the model's construction. As a result, the nan value in the rating and date\_added columns is simply removed.

An empty string should be used to replace any missing values.

We have decided to eliminate any feature with fewer than 5% missing values immediately.

In addition, the caping method is used to remove outliers from data using the outlier removal technique.

Where Q1, Q3, and each attribute's third quantile are represented.



Since almost all of the data, except the release year, are in textual format, as are the data needed to create a cluster. Therefore, there is no need to handle outliers.

**Modeling Approach:**

1. Choose the attributes that you want to cluster the shows around
2. Preprocessing of text: Change all textual data to lowercase and eliminate all punctuation marks and stop words.
3. Attempting to construct a meaningful word from a word corpus.
4. Corpus tokenization and Word vectorization.
5. Dimensionality reduction.
6. Utilize a variety of methods to determine the ideal number of clusters and apply various algorithms to cluster the movies.
7. Choose the best number of clusters and use word clouds to show what's in each one.

* **Creating Cluster:**

We use the following characteristics to create a single cluster column:

• Director

• Cast

• Country

• Rating

• Listed in (genres)

• Description

We must preprocess the data before implementing clusters. To filter the data, we followed these steps.

**Textual Data Pre-processing:**

1. **Removing Stop words:** Words such as "a," "an," "the," and "is," are words that are commonly used in a language but do not convey much meaning. These words can add noise to the data and can sometimes affect the performance of NLP models, so they are often removed as a pre-processing step.
2. **Lowercasing words:** It is the process of converting all the words in a text to lowercase. This can be useful in tasks such as information retrieval or text classification where case differences are not important and also can reduce the size of the vocabulary making it easier to work with larger texts or texts in languages with a high number of inflected forms.
3. **Removing Punctuation:** Removing punctuation is the process of removing any punctuation marks (e.g., periods, commas, exclamation points, etc.) from text data. This is a common pre-processing step in natural language processing (NLP) tasks and text analysis, as punctuation marks often do not carry much meaning and can add noise to the data. Removing punctuation can also make it easier to tokenize text into words or sentences, as punctuation marks often act as delimiters between words or sentences. Additionally, removing punctuation can also help in reducing 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. It can be done using python libraries such as string, re, and nltk.
4. **Stemming:** Stemming is the process of reducing a word to its base or root form. This is a common preprocessing step in natural language processing (NLP) tasks and text analysis. The goal of stemming is to reduce words to their base form so that words with the same stem are treated as the same word, even if they are written in different forms. For example, stemming would reduce "running," "runner," and "ran" to the base form "run." This can be useful in tasks such as information retrieval or text classification where the specific form of a word is not important, and it can also help in reducing the size of the vocabulary. There are several stemmers available in python such as Porter stemmer, Snowball stemmer, and Lancaster stemmer.

We have utilized **SnowballStemmer**to construct a meaningful word from a word corpus.

1. **Tokenization of corpus and Word vectorization:**

Text vectorization is the process of converting text data into numerical vectors or feature representations that can be used for machine learning or data analysis tasks. In simple terms, it transforms the text data into numerical data which can be easily processed by machine learning algorithms. There are several text vectorization techniques available such as a bag of words, TF-IDF, Word2vec,  GloVe, etc.

We have used the **TF-IDF vectorizer**, which stands for Term Frequency Inverse Document Frequency

* TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document). The more often a word appears in a document, the higher its TF score.
* IDF(t) = IDF measures how rare a word is across all the documents in the corpus. The rarer a word, the higher its IDF score.
* The product of TF and IDF is used to calculate the overall weight of a word in a document, which is known as the TF-IDF score.
* Words with high TFIDF scores are considered to be more important and relevant to the document than words with low TF-IDF scores.

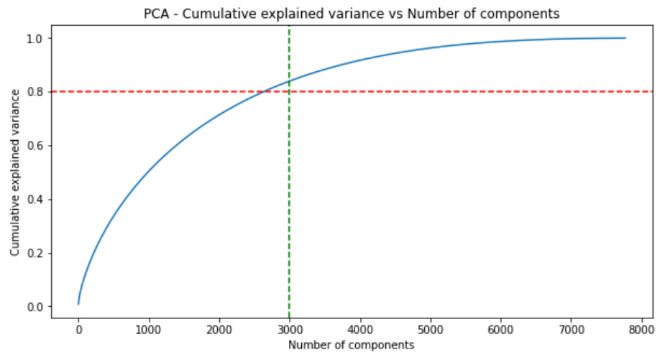
1. **Dimensionality reduction:**

Dimensionality reduction is the process of reducing the number of features or dimensions in a dataset while retaining as much information as possible. The main goal of dimensionality reduction is to simplify the data while minimizing the loss of information. It is a crucial step in machine learning and data analysis as it can help to improve the performance of models, reduce overfitting, and make it easier to visualize and interpret the data.

There are several techniques used for dimensionality reduction, such as:

Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), t-Distributed Stochastic Neighbor Embedding (t-SNE), Autoencoder, and Random Projection, etc.

We have used **Principal Component Analysis (PCA)** to reduce the dimensionality of data.



* We discovered that approximately 7500 components account for 100 percent of the variance.
* 3000 components alone account for more than 80% of the variance.
* Therefore, we can take the top 3000 components to reduce dimensionality and simplify the model while still being able to capture more than 80% of the variance.

**Clustering Algorithms:**

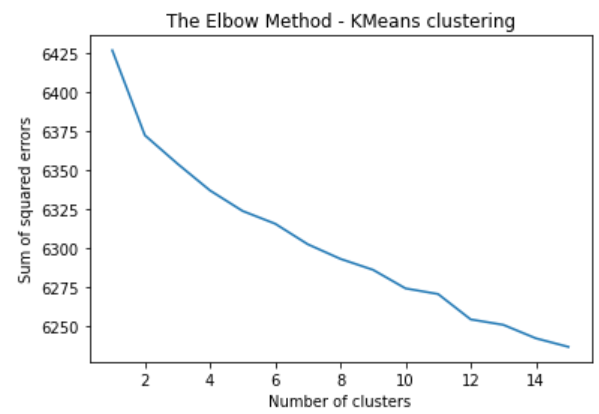
**K-Means Clustering:**

Unsupervised machine learning technique known as K-means clustering is used to group similar data points. The partitioning of a dataset into k clusters, each of which is represented by its centroid and contains comparable data points, is the objective of K-means clustering.

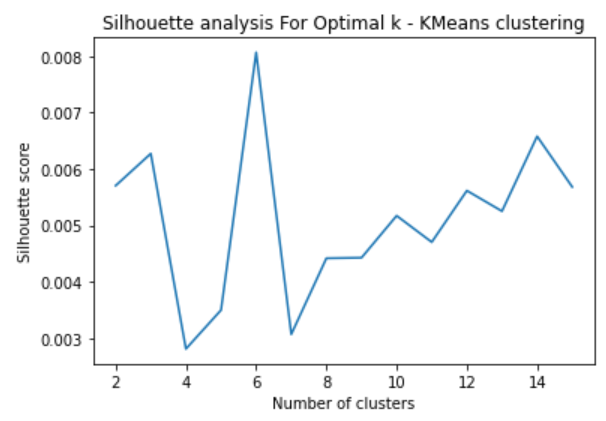
First, the k-means algorithm selects k centroids at random, one for each cluster. After that, it gives each data point to the cluster with the closest centroid. This procedure is repeated until either the number of iterations or the number of data points assigned to clusters stops changing.



By visualizing the elbow curve and silhouette score, we were able to determine the optimal number of clusters for the K-means clustering algorithm.



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



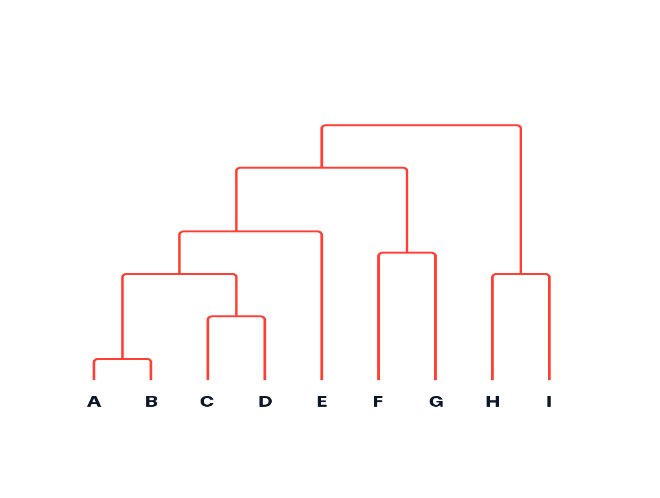
The highest Silhouette score is obtained for 5 clusters. Building 5 clusters using the k-means clustering algorithm.

**Hierarchical clustering:**

The process of grouping data points into a tree-like structure is known as hierarchical clustering. Similar data points can be grouped hierarchically with this alternative to k-means clustering.

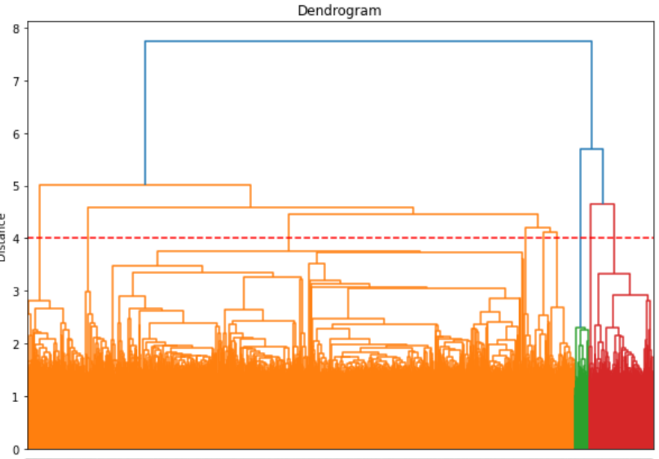
Hierarchical clustering can be broken down into two main categories: Divisive and agglomerative. The algorithm iteratively merges the closest clusters using the agglomerative method, which is a bottom-up strategy in which each data point is regarded as a distinct cluster. Divisive, on the other hand, is a top-down method in which the algorithm iteratively divides the clusters of all the data points into one.

A dendrogram can be used to represent the hierarchical clustering algorithm, making it simple to see how the clusters are organized.



Deciding on the best number of clusters for the agglomerative (hierarchical) clustering algorithm by visualizing the dendrogram.

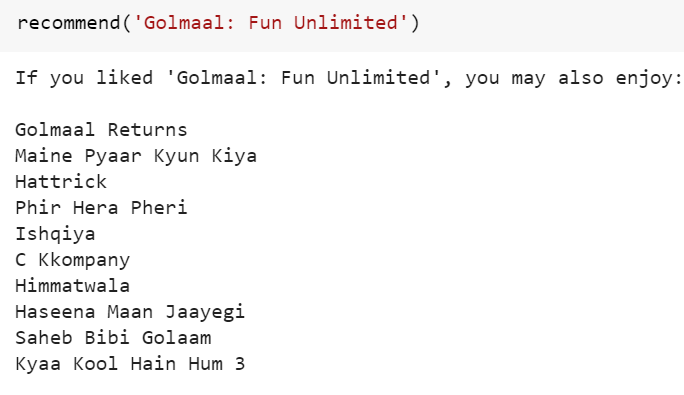
The agglomerative clustering algorithm allows for the creation of seven clusters at a distance of four units.



**Recommendation System:**

A type of recommendation system called a content-based recommendation system suggests products to users based on how similar they are to other products the user is interested in. It compares the items based on their characteristics or characteristics.

* We can make a straightforward content-based recommender system by looking at how much in common the movies and shows are.
* A person who has watched a show on Netflix likes must be able to get a list of similar shows from the recommender system.
* The similarity scores of the shows can be determined using cosine similarity.
* The similarity between A and B can be determined by dividing the magnitude values of the two vectors by their dot products. Simply put, the cosine similarity score increases when the angle between two vectors decreases.



**Conclusion:**

We tackled a text clustering problem in this project, where we had to group Netflix shows into specific clusters so that shows in the same cluster are similar to one another but not to others.

* The dataset contained approximately 7787 records and 11 attributes.
* To get started, we did exploratory data analysis (EDA) and worked on the dataset's missing values.
* Netflix's platform now hosts more movies than shows, and the total number of shows added to Netflix is growing at an exponential rate. Furthermore, the majority of the shows were produced in the United States.
* The following characteristics served as the foundation for the **clustering of the data: cast, country, genre, director, rating, and description** The values in these attributes were tokenized, preprocessed, and vectorized with the help of the TFIDF vectorizer.
* **TFIDF vectorization** resulted in the creation of a total of **'10000 attributes'.**
* **Principal Component Analysis (PCA)** was utilized to address the dimensionality issue. The total number of components was limited to 3000 because **3000 components were able to account for more than 80% of the variance**.
* We first created clusters by employing the **K-Means Clustering** algorithm, and the **optimal number of clusters was found to be 5**. This was obtained by employing the "**elbow method" and the "Silhouette score analysis."**
* After that, the **Agglomerative clustering algorithm** was used to form clusters, and the **optimal number of clusters was found to be seven.** After imagining the "**dendrogram,"** this resulted.
* **A content-based recommender system** was built using the similarity matrix created by using **"cosine similarity."** This recommender system will provide the user with ten recommendations based on the type of show they watched.

**References:**

* <https://www.kaggle.com/>
* <https://stackoverflow.com/>
* <https://towardsdatascience.com/>
* <https://www.analyticsvidhya.com/blog/>
* <https://github.com/>
* <https://www.geeksforgeeks.org/>