**ALMABETTER**

**Capstone Project-Unsupervised Machine Learning**

**Netflix Movies & TV Shows Clustering**

**Ajith R**

**Introduction:**

Netflix is a streaming service that offers a wide variety of award-winning TV shows, movies, anime, documentaries and more – on thousands of internet-connected devices. Netflix has an extensive library of feature films, documentaries, TV Shows, anime, award-winning Netflix originals, and more.

The Netflix Kids experience gives parents the control while kids enjoy family-friendly TV shows and films in their own space. Kids profiles come with PIN-protected parental controls that led you restrict the maturity rating of content kids can watch and block specific titles you don’t want kids to see.

Watch anywhere and anytime. All you need to do is to sign in with your Netflix account to watch instantly on the web at netflix.com from your personal computer or on any internet connected device that offers the Netflix app, including smart TVs, smartphones, tablets, streaming media players and game consoles.

**Abstract:**

The objective was to anticipate bunches of comparable substance by matching text-based elements.

Exploratory Data Analysis is done on the dataset to get the insights from the information however the principal invalid qualities are taken care of. Likewise, some hypothesis testing was additionally performed from the experiences from EDA. After that description segment is our objective variable must be highlighted where NLP activities are performed on it and after that vectorized by utilizing TFIDF.

From that point forward, all that was left was to track down the clusters and fit our models by knowing various clusters, and further, the model is assessed utilizing the metrics.

**Problem Statement:**

This dataset consists of tv shows and movies available on Netflix as of 2019. The dataset is collected from Flixable 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.

The dataset contains following columns:

* 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 content
* 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

**Steps involved:**

The following steps are involved in the project.

1. **Integrating IMDB Dataset**:

The IMDB dataset has the following parts:

**title.akas.tsv.gz** - Contains the following information for titles:

* titleId (string) - a tconst, an alphanumeric unique identifier of the title
* ordering (integer) – a number to uniquely identify rows for a given titleId
* title (string) – the localized title
* region (string) - the region for this version of the title
* language (string) - the language of the title
* types (array) - Enumerated set of attributes for this alternative title. One or more of the following: "alternative", "dvd", "festival", "tv", "video", "working", "original", "imdbDisplay". New values may be added in the future without warning
* attributes (array) - Additional terms to describe this alternative title, not enumerated
* isOriginalTitle (boolean) – 0: not original title; 1: original title

**title.basics.tsv.gz** - Contains the following information for titles:

* tconst (string) - alphanumeric unique identifier of the title
* titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
* primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
* originalTitle (string) - original title, in the original language
* isAdult (boolean) - 0: non-adult title; 1: adult title
* startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
* endYear (YYYY) – TV Series end year. ‘\N’ for all other title types
* runtimeMinutes – primary runtime of the title, in minutes
* genres (string array) – includes up to three genres associated with the title

**title.crew.tsv.gz** – Contains the director and writer information for all the titles in IMDb. Fields include:

* tconst (string) - alphanumeric unique identifier of the title
* directors (array of nconsts) - director(s) of the given title
* writers (array of nconsts) – writer(s) of the given title

**title.episode.tsv.gz** – Contains the tv episode information.

* tconst (string) - alphanumeric identifier of episode
* parentTconst (string) - alphanumeric identifier of the parent TV Series
* seasonNumber (integer) – season number the episode belongs to
* episodeNumber (integer) – episode number of the tconst in the TV series

**title.principals.tsv.gz** – Contains the principal cast/crew for titles.

* tconst (string) - alphanumeric unique identifier of the title
* ordering (integer) – a number to uniquely identify rows for a given titleId
* nconst (string) - alphanumeric unique identifier of the name/person
* category (string) - the category of job that person was in
* job (string) - the specific job title if applicable, else '\N'
* characters (string) - the name of the character played if applicable, else '\N'

**title.ratings.tsv.gz** – Contains the IMDb rating and votes information for titles.

* tconst (string) - alphanumeric unique identifier of the title
* averageRating – weighted average of all the individual user ratings
* numVotes - number of votes the title has received

**name.basics.tsv.gz** – Contains the following information for names.

* nconst (string) - alphanumeric unique identifier of the name/person
* primaryName (string)– name by which the person is most often credited
* birthYear – in YYYY format
* deathYear – in YYYY format if applicable, else '\N'
* primaryProfession (array of strings)– the top-3 professions of the person
* knownForTitles (array of tconsts) – titles the person is known for

Integrating all of these dataset leads to memory issue. Therefore only title.ratings.tsv and title.basics.tsv datasets were used to integrate with the actual dataset given to us. We got more insights in terms of number of votes, user ratings etc.

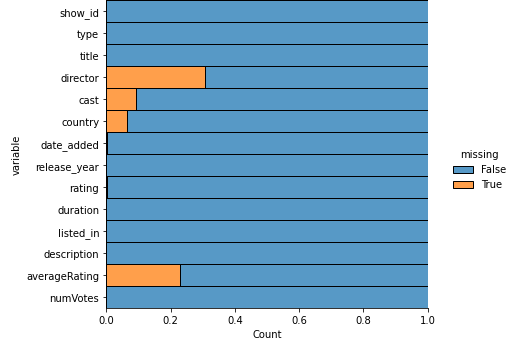
1. **Exploratory Data Analysis:**

After mounting our drive and fetching and reading the dataset given, we performed the Exploratory Data Analysis for it.

To get the understanding of the data and how the content is distributed in the dataset, its type and details such as which countries are watching more and which type of content is in demand etc has been analyzed in this step.

1. **Missing or Null value treatment:**

In this dataset, it is found that we have null values and it is as represented below.



In the above figure that part with yellow color is true for null values. We also tried treating null values with IMDBPY, which is an IMDB library that allows us to get the required null values. The problem with this library is that it requires internet connection and even when connected, the connection is disconnected after 100 requests, leading to waste of time. There are still other ways to treat these null values like scraping the IMDB website or through API services offered by TMDB which requires payment and web programming skills. In this dataset since we using only description column, the null values are not dropped nor treated.

1. **Hypothesis Testing**

Hypothesis testing is a form of statistical inference that uses data from a sample to draw conclusions about a population parameter or a population probability distribution. First, a tentative assumption is made about the parameter or distribution. This assumption is called the null hypothesis and is denoted by H0.

1. **Feature Engineering:**

The description column was added to the list and performed NLP operations on it like converting the entire text into lower case and removing stopwords, punctuations, URLs etc and then tokenized the text for further process.

1. **TF-IDF**

TF-IDF stands for Term Frequency Inverse Document Frequency. It is a measure, used in the fields of information retrieval (IR) and machine learning, that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc) in a document amongst a collection of documents.

PCA was also utilized to improve the performance at a very low cost of model accuracy. PCA’s benefits include reduction of noise in the data, feature selection and produce independent and uncorrelated features of the data.

So it's essential to transform our text into TF-IDF vectorizer, then convert it into an array so that we can fit into our model.

1. **Finding number of clusters :**

The goal is to separate groups with similar characteristics and assign them to clusters. We used the Elbow method and the Silhouette score to do so, and we have determined that 28 clusters should be an optimal number of clusters.

1. **Fitting into model**

In this task, we have implemented a K means clustering algorithm. K-means is a technique for data clustering that may be used for unsupervised machine learning. It is capable of classifying unlabelled data into a predetermined number of clusters based on similarities (k).

1. **Model Evaluation**

Evaluation of the model is performed using:

* Silhouette’s coefficient
* Calinski-Harabasz Index
* Davies-Bouldin Index

**Algorithms:**

**K Means Clustering:**

K-means algorithm searches for a pre-determined number of clusters within an unlabelled multidimensional dataset. It accomplishes this using a simple conception of what the optimal clustering looks like:

* The "cluster center" is the arithmetic mean of all the points belonging to the cluster.
* Each point is closer to its own cluster center than to other cluster centers.

## **k-Means Algorithm: Expectation–Maximization:**

### Expectation–maximization (E–M) is a powerful algorithm that comes up in a variety of contexts within data science. k-means is a particularly simple and easy-to-understand application of the algorithm.

### In short, the expectation–maximization approach here consists of the following procedure:

### Guess some cluster centers

### Repeat until converged

### E-Step: assign points to the nearest cluster center

### M-Step: set the cluster centers to the mean

### Here the "E-step" or "Expectation step" is so-named because it involves updating our expectation of which cluster each point belongs to.

### The "M-step" or "Maximization step" is so-named because it involves maximizing some fitness function that defines the location of the cluster centers — in this case, that maximization is accomplished by taking a simple mean of the data in each cluster.

### The literature about this algorithm is vast, but can be summarized as follows: under typical circumstances, each repetition of the E-step and M-step will always result in a better estimate of the cluster characteristics.

### 

**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.

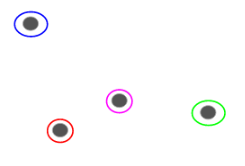
### There are mainly two types of hierarchical clustering:

### Agglomerative hierarchical clustering

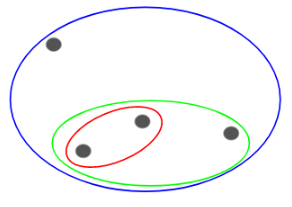
### Divisive Hierarchical clustering

## **Agglomerative Hierarchical Clustering:**

### Bottom Up Approach - We assign each point to an individual cluster in this technique. Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning.



### Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left:



Since we are merging the clusters at each step, this type of clustering is known as additive hierarchical clustering.

## **Divisive Hierarchical Clustering:**

### Top Down Approach or also known as DIANA (DIvisive Analysis) is the inverse of agglomerative clustering

### Divisive hierarchical clustering works in the opposite way. Instead of starting with n clusters (in case of n observations), we start with a single cluster and assign all the points to that cluster. So, it doesn’t matter if we have 10 or 1000 data points. All these points will belong to the same cluster at the beginning.

### 

### Now, at each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single point.

### 

### We are splitting (or dividing) the clusters at each step, hence the name divisive hierarchical clustering.

### DBScan:

### The **DBSCAN algorithm** is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points.

DBSCAN algorithm can be abstracted in the following steps:

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.
4. A point*a* and *b* are said to be density connected if there exist a point *c* which has a sufficient number of points in its neighbors and both the points*a* and *b* are within the epsdistance. This is a chaining process. So, if b is neighbor of c, c is neighbor ofd, d is neighbor of *e*, which in turn is neighbor of *a* implies that b is neighbor ofa.
5. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

# **Principal Component Analysis:**

## Principal component analysis is a fast and flexible unsupervised method for dimensionality reduction in data. Its behaviour is easiest to visualize by looking at a two-dimensional dataset.

## Step 1 : Take the whole dataset with dimension **d** (which corresponds to the **d** features)

## Step 2 : Compute the mean of every dimension of the whole dataset.

## Step 3 : Compute the covariance matrix of the whole dataset

### Cov(X,Y)=1n−1∑i=1n(Xi−X¯)(Yi−Y¯)

## Step 4 : Compute Eigenvectors and corresponding Eigenvalues for the covariance matrix A

### Let A be a square matrix, ν a vector and λ a scalar that satisfies Aν=λν, then λ is called eigenvalue associated with eigenvector ν of A.

## Step 5 : Sort the eigenvectors by decreasing eigenvalues and choose **k** eigenvectors with the largest eigenvalues to form a **d×k** dimensional matrix **W**.

## Step 6 : Transform the samples onto the new subspace

### In the last step, we use the d×k dimensional matrix W that we just computed to transform our samples onto the new subspace via the equation y=W'×x where W' is the transpose of the matrix W.

### Because of the versatility and interpretability of PCA, it has been shown to be effective in a wide variety of contexts and disciplines. Given any high-dimensional dataset, we can start with PCA in order to visualize the relationship between points, to understand the main variance in the data, and to understand the intrinsic dimensionality.

### Certainly PCA is not useful for every high-dimensional dataset, but it offers a straightforward and efficient path to gaining insight into high-dimensional data.

### PCA's main weakness is that it tends to be highly affected by outliers in the data. For this reason, many robust variants of PCA have been developed, many of which act to iteratively discard data points that are poorly described by the initial components.

# **UMAP(Uniform Manifold Approximation and Projection for Dimension Reduction):**

Uniform Manifold Approximation and Projection (UMAP) is a dimension reduction technique that can be used for visualisation similarly to t-SNE, but also for general non-linear dimension reduction. The algorithm is founded on three assumptions about the data.

1. The data is uniformly distributed on Riemannian manifold;
2. The Riemannian metric is locally constant (or can be approximated as such);
3. The manifold is locally connected.

UMAP is a fairly flexible non-linear dimension reduction algorithm. It seeks to learn the manifold structure of your data and find a low dimensional embedding that preserves the essential topological structure of that manifold. In this notebook we will generate some visualisable 4-dimensional data, demonstrate how to use UMAP to provide a 2-dimensional representation of it, and then look at how various UMAP parameters can impact the resulting embedding.

**Model performance:**

Model can be evaluated by various metrics such as:

**Silhouette’s Coefficient**

If the ground truth labels are not known, the evaluation must be performed utilizing the model itself. The Silhouette Coefficient is an example of such an evaluation, where a more increased Silhouette Coefficient score correlates to a model with better-defined clusters. The Silhouette Coefficient is determined for each sample and comprised of two scores:

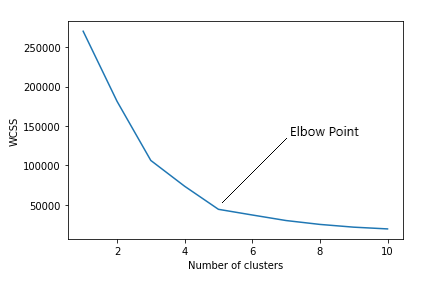
* The mean distance between a sample and all other points in the same class.
* The mean distance between a sample and all other points in the next nearest cluster.

The Silhouette Coefficient *s* for a single sample is then given as:

**Elbow Method:**

To find the optimal number of clusters i.e ‘k’, elbow method is used.

In the Elbow method, we are actually varying the number of clusters ( K ) from 1 – 10. For each value of K, we are calculating WCSS ( Within-Cluster Sum of Square ). WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1. When we analyze the graph we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to this point is the optimal K value or an optimal number of clusters.



**Conclusion:**

* If the null values were treated then we would have arrived at better EDA.
* In this dataset, Netflix is increasingly focusing on movies rather than TV shows.
* We found that different types of content are available in different countries, but TV-MA is the content that is available in the majority of countries. This could be because it shows that it is just for adult audiences, and the Netflix audience enjoys content like this.
* We've also explained different clusters based on their content; we've defined 28 clusters and enforced the K-means clustering algorithm. And then we determined that cluster number nine has the most clusters; we've also plotted a scatter plot in which we may interact with similar content about that cluster.

**REFERENCES:**

**Almabetter**

**Towards Data Science**

**Analytics Vidhya**

**sklearn**