**NETFLIX MOVIES AND TV SHOWS CLUSTERING**

Nitesh Bhowmick,

Ayush Goyal, M Sameer Ahmed,

Data science trainees

Alma Better, Bangalore

**Abstract:**

The study conducted was on the proposed “NETFLIX MOVIES AND TV SHOWS CLUSTERING”. The main purpose of this study was to build a DEC [(Deep Embedded Cluster)](https://arxiv.org/abs/1511.06335) with the given movie/tv show descriptions to find similar movies/tv shows as a recommender system. The model will consist of an autoencoder/decoder part and a clustering layer part. The researchers used the descriptive type of survey methods where they distributed questionnaires to the respondent of the study as a research instrument for data gathering.

Our experiment can help understand what could be the reason for the Clustering of such labels by feature selection, data analysis and prediction with Python Programming & ML algorithm taking into account previous trends to determine the correct Clustering.

**1. Problem Statement**

This dataset consists of 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, and rotten tomatoes can also provide many interesting findings.

**2. Introduction**

N[etflix, Inc](https://en.wikipedia.org/wiki/Netflix). is an American [technology and media services provider](https://en.wikipedia.org/wiki/Over-the-top_media_service) and [production company](https://en.wikipedia.org/wiki/Production_company) headquartered in [Los Gatos, California](https://en.wikipedia.org/wiki/Los_Gatos,_California). Netflix was founded in 1997 by [Reed Hastings](https://en.wikipedia.org/wiki/Reed_Hastings) and [Marc Randolph](https://en.wikipedia.org/wiki/Marc_Randolph) in [Scotts Valley, California](https://en.wikipedia.org/wiki/Scotts_Valley,_California). The company’s primary business is its subscription-based [streaming service](https://en.wikipedia.org/wiki/Streaming_service_provider), which offers online streaming of a library of films and television series, including those produced in-house.

Netflix is a popular entertainment service used by people around the world. This EDA will explore the Netflix dataset through visualizations and graphs using python libraries, matplotlib, and seaborn.

### 

### Netflix is known for its strong recommendation engines. They use a mix of content-based and collaborative filtering models to recommend tv shows and movies. In this task, one can create a recommendation engine based on text/description similarity techniques.

### 

I will proceed with reading the data, and then perform data analysis. The practice of examining data using analytical or statistical methods in order to identify meaningful information is known as data analysis. After data analysis, we will find out the data distribution and data types. We will train 3 Clustering ML algorithms to predict the output. We will also compare the outputs.

Let us get started with the project implementation.

**3. Key features of Retail Sales Prediction:-**

### Most of the fields are self-explanatory. The following are descriptions for those that aren't.

* **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 Releaseyear of the movie/show
* **Rating:** TV Rating of the movie/show
* **Duration:** Total Duration - in minutes or number of seasons
* **Listed\_in:** Genere
* **Description:** The Summary description

## **4. Methodology**

This dataset is a live dataset of Netflix. In analysing this problem we observe that the NETFLIX MOVIES AND TV SHOWS problem is a Clustering problem and The main purpose of this study was to build a DEC [(Deep Embedded Cluster)](https://arxiv.org/abs/1511.06335) with the given movie/tv show descriptions to find similar movies/tv shows as a recommender system. The model will consist of an autoencoder/decoder part and a clustering layer part.

Analysing the Dataset by using Exploratory Data Analysis. Using Exponential Moving Averages analyse Trends and Seasonality in the Netflix dataset. Analyse Clustering analysis using the following prediction analysis, **A.** K Means Clustering **B.** Dendrogram. **C.** Agglomerative Clustering.

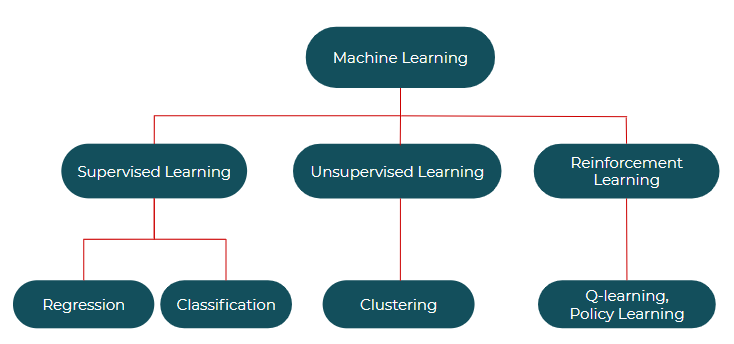
### **5. Machine Learning:**

**Machine Learning:**

Learning is any process by which a system improves performance from experience.

Machine Learning is concerned with computer programs that automatically improve their performance through experience.

**Task in Machine Learning:-**



**Supervised learning: -** Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Basically supervised learning is when we teach or train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labelled data.

**Types:-**

* Linear Regression
* Logistic Regression
* Classification
* Naive Bayes Classifiers
* K-NN (k nearest neighbours)
* Decision Trees
* Support Vector Machine

**Advantages:-**

* Supervised learning allows collecting data and produces data output from previous experiences.
* Helps to optimize performance criteria with the help of experience.
* Supervised machine learning helps to solve various types of real-world computation problems.

**Disadvantages:-**

* Classifying big data can be challenging.
* Training for supervised learning needs a lot of computation time. So, it requires a lot of time.

#### **Unsupervised learning:-**

Unsupervised learning is the training of a machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.

Unlike supervised learning, no teacher is provided which means no training will be given to the machine. Therefore the machine is restricted to finding the hidden structure in unlabeled data by itself.

**Types of Unsupervised Learning:-**

**Clustering:-**

1. Exclusive (partitioning)
2. Agglomerative
3. Overlapping
4. Probabilistic

**Clustering Types:-**

1. Hierarchical clustering
2. K-means clustering
3. Principal Component Analysis
4. Singular Value Decomposition
5. Independent Component Analysis

**6. Steps involved:**

* **Exploratory Data Analysis**

 In statistics, exploratory data analysis (*EDA*) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily *EDA* is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task.

* **Null values Treatment**

Our dataset contains nan values. There were a lot of nan values in some columns that had to be dealt with. So I wrote a code to specifically deal with the nan values of each column by replacing them with some data. Removing some rows of two columns which contain NaN values. This dataset does not contain any duplicate values.

* **Fitting different models**

For Analysis we tried various types of ML models:

1. **K Means Clustering.**
2. **Dendrogram.**
3. **Agglomerative Clustering.**

**7. Algorithms:**

1. **K Means Clustering:**

K-Means Clustering is an unsupervised learning algorithm that is used to solve clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

**K-Means Algorithm:-**

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

“It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.”

It allows us to cluster the data into different groups and is a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

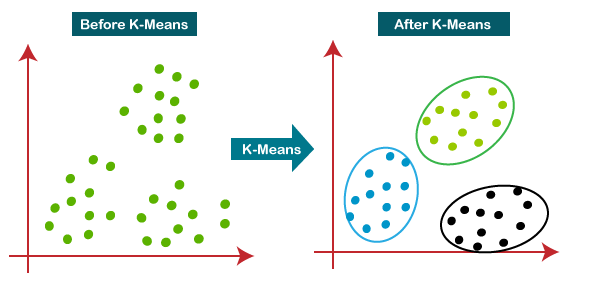
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

* Determines the best value for K centre points or centroids by an iterative process.
* Assigns each data point to its closest k-centre. Those data points which are near the particular k-centre, create a cluster.

Hence each cluster has data points with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



## **How does the K-Means Algorithm Work:-**

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other than the input dataset).

**Step-3**: Assign each data point to its closest centroid, which will form the predefined K clusters.

**Step-4:** Calculate the variance and place a new centroid of each cluster.

**Step-5:** Repeat the third steps, which means reassigning each datapoint to the new closest centroid of each cluster.

**Step-6:** If any reassignment occurs, then go to **step-4** else go to FINISH.

**Step-7:** The model is ready.

## **Advantages of k-means:-**

* Relatively simple to implement.
* Scales to large data sets.
* Guarantees convergence.
* Can warm-start the positions of centroids.
* Easily adapts to new examples.
* Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

## **Disadvantages of k-means:-**

* Choosing k manually.
* Being dependent on initial values.
* Clustering data of varying sizes and densities.
* Clustering outliers.
* Scaling with a number of dimensions.

**Import K Means Clustering:-**

from sklearn.cluster import KMeans

1. **Dendrogram:-**

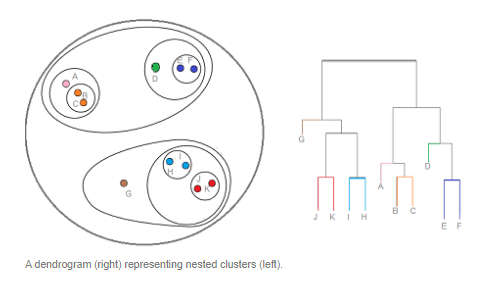
A dendrogram is a branching diagram that represents the relationships of similarity among a group of entities.

# **Hierarchical Clustering:**

Hierarchical clustering is where you build a cluster tree (a dendrogram) to represent data, where each group (or “node”) links to two or more successor groups. The groups are [nested](https://www.statisticshowto.com/nested-model-anova-factors/#model) and organized as a tree, which ideally ends up as a meaningful classification scheme.

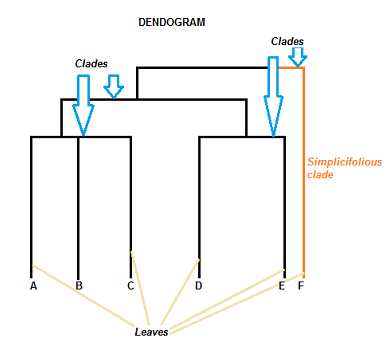
Each node in the cluster tree contains a group of similar data; Nodes group on the graph next to other, similar nodes. Clusters at one level join with clusters at the next level up, using a degree of similarity; The process carries on until all nodes are in the tree, which gives a visual snapshot of the data contained in the whole set. The total number of clusters is *not* predetermined before you start the tree creation.

## **Dendrogram:**



A dendrogram is a type of [tree diagram](https://www.statisticshowto.com/how-to-use-a-probability-tree-for-probability-questions/) showing hierarchical clustering — relationships between similar sets of data. They are frequently used in biology to show clustering between genes or samples, but they can represent any type of grouped data.

## **Parts of a Dendrogram:**



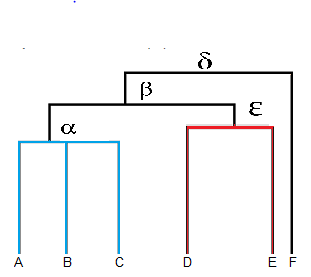
A dendrogram can be a column graph (as in the image below) or a row graph. Some dendrograms are circular or have a fluid shape, but the software will usually produce a row or column graph. No matter what the shape, the basic graph comprises the same parts:

* The *clade* is the branch. Usually labelled with Greek letters from left to right (e.g. α β, δ…).
* Each clade has one or more *leaves*. The leaves in the above image are
  + Single (simplicifolius): F
  + Double (bifolius): D E
  + Triple (trifolious): A B C

A clade can theoretically have an infinite amount of leaves. However, the more leaves you have, the harder the graph will be to read with the naked eye.

## **How to Read a Dendrogram:**

The clades are arranged according to how similar (or dissimilar) they are. Clades that are close to the same height are similar to each other; clades with different heights are dissimilar — the greater the difference in height, the more dissimilarity (you can measure similarity in many different ways; One of the most popular measures is [Pearson’s Correlation Coefficient](https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/)).



* Leaves A, B, and C are more similar to each other than they are to leaves D, E, or F.
* Leaves D and E are more similar to each other than they are to leaves A, B, C, or F.
* Leaf F is substantially different from all of the other leaves.

Note that on the above graph, the same clave, β joins leaves A, B, C, D, and E. That means that the two groups (A, B, C & D, E) are more similar to each other than they are to F.

**Advantages of Hierarchical Clustering:**

- We can obtain the optimal number of clusters from the model itself, human intervention not required.

- Dendrograms help us with clear visualization, which is practical and easy to understand.

**Disadvantages of Hierarchical Clustering:**

- Not suitable for large datasets due to high time and space complexity.

- There is no mathematical objective for Hierarchical clustering.

- All the approaches to calculating the similarity between clusters have their own disadvantages.

**Import Dendrogram:-**

from scipy.cluster.hierarchy import [dendrogram](https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram)

1. **Agglomerative Clustering:-**

Agglomerative Clustering is a type of hierarchical clustering algorithm. It is an unsupervised machine learning technique that divides the population into several clusters such that data points in the same cluster are more similar and data points in different clusters are dissimilar.

**Basically, there are two types of hierarchical cluster analysis strategies –**

**1. Agglomerative Clustering:-**

Also known as the bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all data.

**Algorithm:**

given a dataset (d1, d2, d3, ....dN) of size N

# compute the distance matrix

for i=1 to N:

# as the distance matrix is symmetric about

# the primary diagonal so we compute only the lower

# part of the primary diagonal

for j=1 to i:

dis\_mat[i][j] = distance[di, dj]

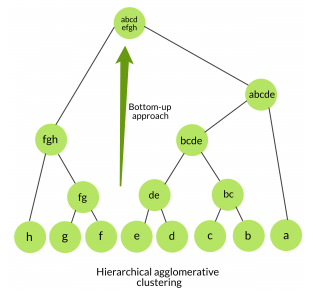
each data point is a singleton cluster

repeat

merge the two cluster having minimum distance

update the distance matrix

until only a single cluster remains.



**2. Divisive clustering:**

Also known as a top-down approach. This algorithm also does not require prespecify the number of clusters. Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.

**Algorithm:**

given a dataset (d1, d2, d3, ....dN) of size N

at the top we have all data in one cluster

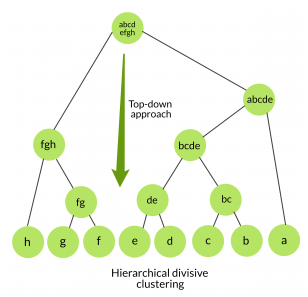
the cluster is split using a flat clustering method eg. K-Means etc

repeat

choosing the best cluster among all the clusters to split

that cluster by the flat clustering algorithm

until each data is in its own singleton cluster.

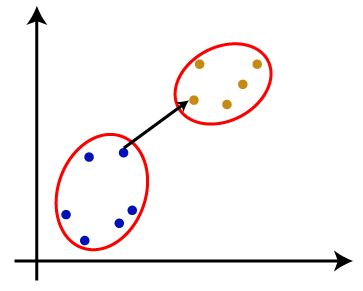


**Hierarchical Agglomerative *vs* Divisive clustering –**

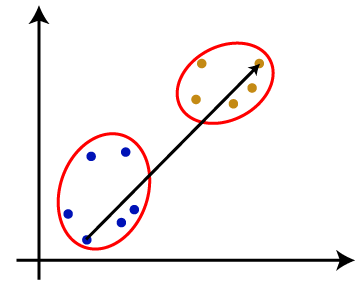
* Divisive clustering is more *complex* as compared to agglomerative clustering, as in the case of divisive clustering we need a flat clustering method as “subroutine” to split each cluster until we have each data having its own singleton cluster.
* Divisive clustering is more *efficient* if we do not generate a complete hierarchy all the way down to individual data leaves. The time complexity of a naive agglomerative clustering is O(n3) because we exhaustively scan the N x N matrix dist\_mat for the lowest distance in each of the N-1 iterations. Using priority queue data structure we can reduce this complexity to O(n2logn). By using some more optimizations it can be brought down to O(n2). Whereas for divisive clustering given a fixed number of top levels, using an efficient flat algorithm like K-Means, divisive algorithms are linear in the number of patterns and clusters.
* A divisive algorithm is also more *accurate*. Agglomerative clustering makes decisions by considering the local patterns or neighbour points without initially taking into account the global distribution of data. These early decisions cannot be undone. whereas divisive clustering takes into consideration the global distribution of data when making top-level partitioning decisions.

### **Measure for the distance between two clusters:**

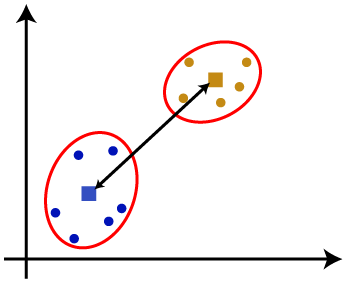
As we have seen, the closest distance between the two clusters is crucial for hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called Linkage methods. Some of the popular linkage methods are given below:

**1. Single Linkage:** It is the Shortest Distance between the closest points of the clusters. Consider the below image:

**2 . Complete Linkage:** It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.

**3. Average Linkage:-** It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

**4. Centroid Linkage:-** It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:



From the above-given approaches, we can apply any of them according to the type of problem or business requirement.

**Advantages of Agglomerative Hierarchical Clustering Algorithm:-**

* The agglomerative technique is easy to implement.
* It can produce an ordering of objects, which may be informative for the display.
* In agglomerative Clustering, there is no need to pre-specify the number of clusters.

**Disadvantages of Agglomerative Hierarchical Clustering Algorithm:-**

**Time and space complexity:** The time and space complexity of agglomerative clustering are more than K-means clustering, and in some cases, it is prohibitive.

**Final merging decisions:** The merging decisions, once given by the algorithm, cannot be undone at a later point in time. Due to this, local optimisation criteria cannot become global criteria. Note that there are some advanced approaches available to overcome this problem.

**Import Agglomerative Clustering:**

from sklearn.cluster import AgglomerativeClustering

**8.**  **CONCLUSION**

* From elbow and silhouette score, optimal of 26 clusters formed, K Means is best for identification than Hierarchical as the evaluation metrics also indicate the same. In kmean cluster 0 has the highest number of data points and is evenly distributed for other clusters
* Netflix has 5372 movies and 2398 TV shows, and there are more number movies on Netflix than on TV shows.
* TV-MA has the highest number of ratings for tv shows i.e adult ratings
* Highest number of movies released in 2017 and 2018

highest number of movies released in 2020 The number of movies on Netflix is growing significantly faster than the number of TV shows. We saw a huge increase in the number of movies and television episodes after 2015. there is a significant drop in the number of movies and television episodes produced after 2020. It appears that Netflix has focused more attention on increasing Movie content than TV Shows. Movies have increased much more dramatically than TV shows

* The most content is added to Netflix from October to January
* Documentaries are the top most genre on Netflix which is followed by standup comedy and Drams and international movies
* kids tv is the top most TV show genre in Netflix
* most of the movies have a duration of between 50 to 150
* highest number of tv\_shows consisting of single season
* Those movies that have a rating of NC-17 have the longest average duration.

When it comes to movies having a TV-Y rating, they have the shortest runtime on average

* united states has the highest number of content on Netflix, followed by India
* India has the highest number of movies on Netflix
* 30% of movies are released on Netflix.
* 70% of movies added on Netflix were released earlier by different modes.

**References-**

1. Geeks for Geeks
2. Analytics Vidhya
3. Nvidia
4. scikit-learn