**TECHNICAL DOCUMENTATION**

**Capstone Project: 4**

**Unsupervised Learning: Clustering**

**NETFLIX MOVIES AND TV SHOWS CLUSTERING**





**About the Dataset:**

This dataset consists of Tv-shows and movies available on Netflix as of 2019. The dataset is collected from Fixable 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.

**Problem Statement:**

* Need to perform Exploratory Data Analysis to fetch meaningful insight from the data to help the stakeholders.
* Understanding what type of content is available in different countries
* Netflix has increasingly focused on TV rather than movies in recent years.
* Clustering of similar content by matching text-based features.

## **Methodology**

**EDA:**

 Exploratory Data Analysis is a data analytics process to understand the data in depth and learn the different data characteristics, often with visual means. This allows you to get a better feel for your data and find useful patterns in it.

All of this can be done with Exploratory Data Analysis. It helps you gather insights and make better sense of the data, and removes irregularities and unnecessary values from data.

* Helps you prepare your dataset for analysis.
* Allows a [machine learning model](https://www.simplilearn.com/machine-learning-models-article) to predict our dataset better.
* Gives you more accurate results.
* It also helps us to choose a better machine-learning model.

**Using Python on Google Colaboratory we Performed EDA on the Global Terrorism Dataset to fetch meaningful insights.**

**Libraries used:**



**PANDAS** – Pandas is a library used for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

**NUMPY** - NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

**MATPLOTLIB** - Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

**PLOTLY:** Plotly makes it easy to share and edit your work privately, with your co-workers, or with the world. There is so much to learn from open government data and we are happy to feature tools like Plotly and CartoDB that make it easier for everyone to become a data scientist.

## **SEABORN** - Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top of the matplotlib library and is also closely integrated with the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

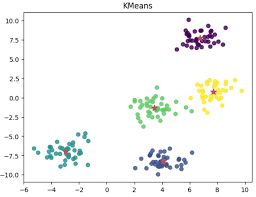
## **Sklearn -** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, Clustering, and dimensionality reduction via a consistency interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy, and Matplotlib.

**SCIPY**- SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, and other tasks common in science and engineering.

**NLTK -** [Natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) (NLP) is a field that focuses on making natural human language usable by computer programs. NLTK, or [Natural Language Toolkit](https://www.nltk.org/), is a Python package that you can use for NLP. A lot of the data that you could be analyzing is [unstructured data](https://en.wikipedia.org/wiki/Unstructured_data) and contains human-readable text. Before you can analyze that data programmatically, you first need to pre-process it. In this tutorial, you’ll take your first look at the kinds of text pre-processing tasks you can do with NLTK so that you’ll be ready to apply them in future projects. You’ll also see how to do some basic text analysis and create visualizations.

## **Model Used:**

**KMeans 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.



**Hierarchical - Clustering:** Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and is also known as hierarchical cluster analysis or HCA. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.

* **Agglomerative**: Agglomerative is a bottom-up approach, in which the algorithm starts by taking all data points as single clusters and merging them until one cluster is left.

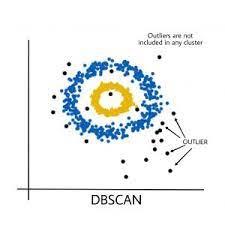
**DBSCAN:** DBSCAN stands for density-based spatial clustering of applications with noise. It is able to find arbitrarily shaped clusters and clusters with noise (i.e. outliers).

The main idea behind DBSCAN is that a point belongs to a cluster if it is close to many points from that cluster.

There are two key parameters of DBSCAN:

•Eps: The distance that specifies the neighborhoods. Two points are considered to be neighbors if the distance between them is less than or equal to eps.

•minPts: Minimum number of data points to define a cluster.



**## Classification model (SVM)**

*Applied SVM to Predict the cluster number of the data*

Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both- classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

**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.
* Missing values and duplicate values were handled adequately (features like Director is having around 30% cast: 9.22, country: 6.51% of missing values.)
* Outlier detection and removal techniques were performed to, on date added and released date features which are a kind of correlated features ( 61 outliers removed from the date\_added feature)
* We began by dealing with the dataset's missing values and doing exploratory

##### ----------------------- (EDA) and found various insights including---------------------------------

* In our dataset we have around 69% content as movies, Remaining 31% as TV shows, which signifies people generally prefer movies over TV-Shows.
* Most content on Netflix is rated for Mature Audiences and is over 14 years old.
* Documentaries  is the topmost genre available on Netflix, followed by Stand-up and Drama
* From the year 2015, there was a significant rise in the rate of making TV -shows but in 2021 this trend was discontinued due to the pandemic situation by covid-19, and further, the data is not available.
* The most vital & Popular words recapitulated in TV shows and movies are ‘Love', 'Christmas', 'World', 'Story', 'Man', 'Live', 'Girl', and 'Life’ these means mostly movies and TV-Shows contains content related to these keywords.
* The top countries with the most numbers of TV-Shows are
  1. USA – Not the highest in population but the highest content Creator
  2. India
  3. UK
  4. Japan
  5. South Korea

##### ---------------------------Conclusion on Modelling-----------------------------

* In this project, we worked on a text clustering problem wherein we had to classify/group the Netflix shows and movies into certain clusters such that the shows and Clusters within a cluster are similar to each other and the shows in different clusters are dissimilar to each other.
* It was found that Netflix hosts more movies than TV shows on its platform, and the total number of shows added on Netflix is growing exponentially. Also, the majority of the shows were produced in the United States, and the majority of the shows on Netflix were created for adults and young adults age group.
* It was decided to cluster the data based on the attributes: director, cast, country, genre, and description. The values in these attributes were tokenized, pre-processed, and then vectorized using the TFIDF vectorizer.
* Through TF-IDF Vectorization, we created a total of 20000 attributes.
* We used Principal Component Analysis (PCA) to handle the curse of dimensionality. 3000 components were able to capture more than around 90% of the variance, and hence, the number of components was restricted to 3000.
* We first built clusters using the k-means clustering algorithm, and the optimal number of clusters came out to be 10 with a good WCSS Value and a high Silhouette score of 0.0483. 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 20 at a distance of 20. This was obtained after visualizing the Dendogram.
* DBSCAN cluster didn't give satisfying results. it was a kind of biased model that clustered most of the data into a single cluster.

Estimated number of clusters: 19  
Estimated number of noise points: 7134

* Build an SVM Classification model with hyper-parameter tuning at the end to predict a cluster number for a data record.
* Classifier predicts good results with above 99% accuracy