Topic : NETFLIX MOVIES AND TV SHOWS CLUSTERING

For capstone project using unsupervised machine learning

Mohd Sharik

Data Science Trainee

Alma Better

Abstract:

1. Our study encompasses the findings done Predicting whether Understanding what type content is available in different countries
2. Is Netflix has increasingly focusing on TV rather than movies in recent years.
3. Clustering similar content by matching text-based features

Netflix is a company that manages a large collection of TV shows and movies, streaming it anytime via online. This business is profitable because users make a monthly payment to access the platform. However, customers can cancel their subscriptions at any time. Therefore, the company must keep the users hooked on the platform and not lose their interest. This is where recommendation systems start to play an important role, providing valuable suggestions to users is essential.

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.

In this project, you are required to do

1. Exploratory Data Analysis
2. Understanding what type content is available in different countries
3. Is Netflix increasingly focused on TV rather than movies in recent years?
4. Clustering similar content by matching text-based features.

Data Description

Attribute Information

---

1. show_id : Unique ID for every Movie / Tv Show
2. type : Identifier - A Movie or TV Show
3. title : Title of the Movie / Tv Show
4. director : Director of the Movie
5. cast : Actors involved in the movie / show
6. country : Country where the movie / show was produced
7. date_added : Date it was added on Netflix
8. release_year : Actual Release year of the movie / show
9. rating : TV Rating of the movie / show
10. duration : Total Duration - in minutes or number of seasons
11. listed_in : Genere
12. description: The Summary description

Approach taken:

This project is divided into multiple section where each section will have it's own importance towards our problem statements. the approach that we will be following in this project is given as-

Section 1: In this section we will simply loading our dataset into google colab and will explore the basic information about data.

Section 2: In section 2 we will be dealing with missing values of this dataset and will impute the values into missing places.

Section 3: In this section We fill focus on Exploratory data analysis of the dataset using various methods and visualization plot and will be extracting the information from this dataset as much as we can.

Section 4: In this section we will be dealing with outliers in out dataset and will see how to define our outlier criteria and deal with outliers.

Section 5: In this Section we will be applying NLP in order to get most similar contents like most similar movies.

Section 6: In this section we will be perform various clustering methods to find out best no. of clusters and for validation we will be using Silhouette score and elbow curve where ever it is applicable.

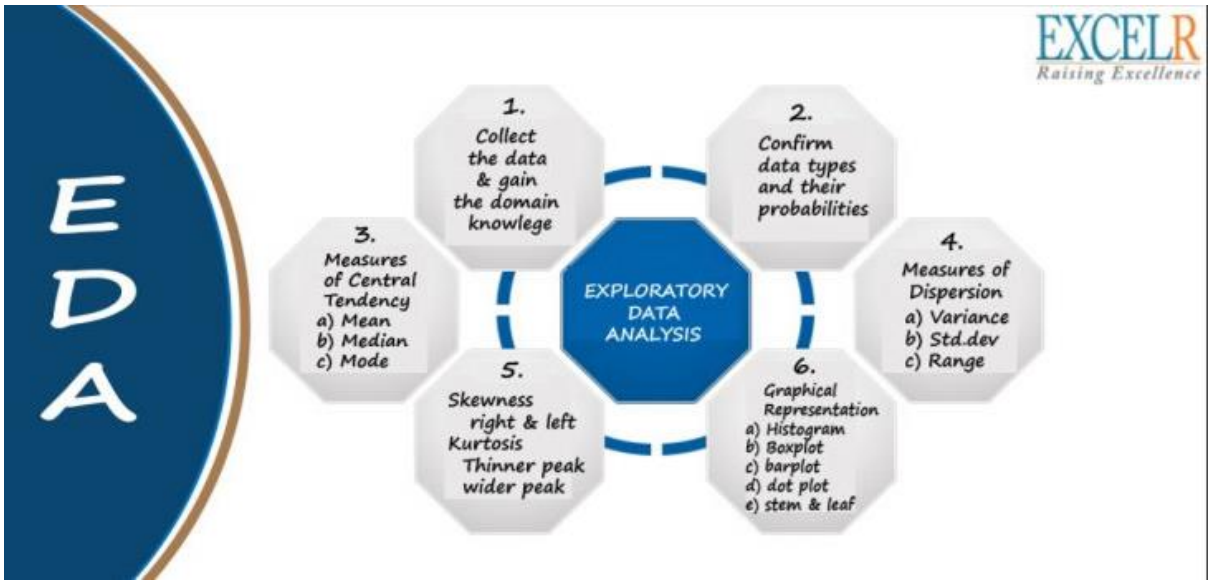
Section 7: In this section we will be giving a quick summary of entire notebook.

5. Steps involved:

Exploratory Data Analysis

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.



Feature Engineering:

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make machine learning work well on new tasks, it might be necessary to design and train better features. As you may know, a "feature" is any measurable input that can be used in a predictive model — it could be the color of an object or the sound of someone's voice. Feature engineering, **in simple terms, is the act of converting raw observations into desired features using statistical or machine learning approaches.**

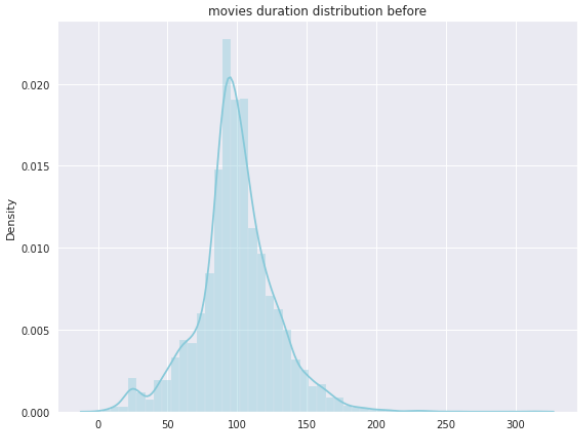
We used the following techniques;

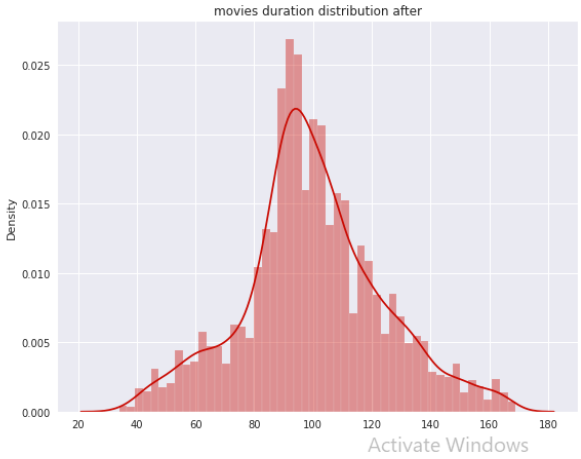
1. **Null values Treatment**

   We have seen that a lot information was missing in dataset hence we have filled these missing values possible string values as most of missing values were in categorical features which becomes difficult to fill with the help of some statistical methods like mean median etc.

2. Handling possible outliers:

This is a very task in any machine learning project as if not properly then a machine learning technique may lead to poor results. While performing EDA we have identified two feature where there are some extreme values and these values can lead to poor predictions. Although there are many ways to treat outliers or extreme values but simply have dropped we had very high no. observation and remove some observation will not affect our dataset randomness.

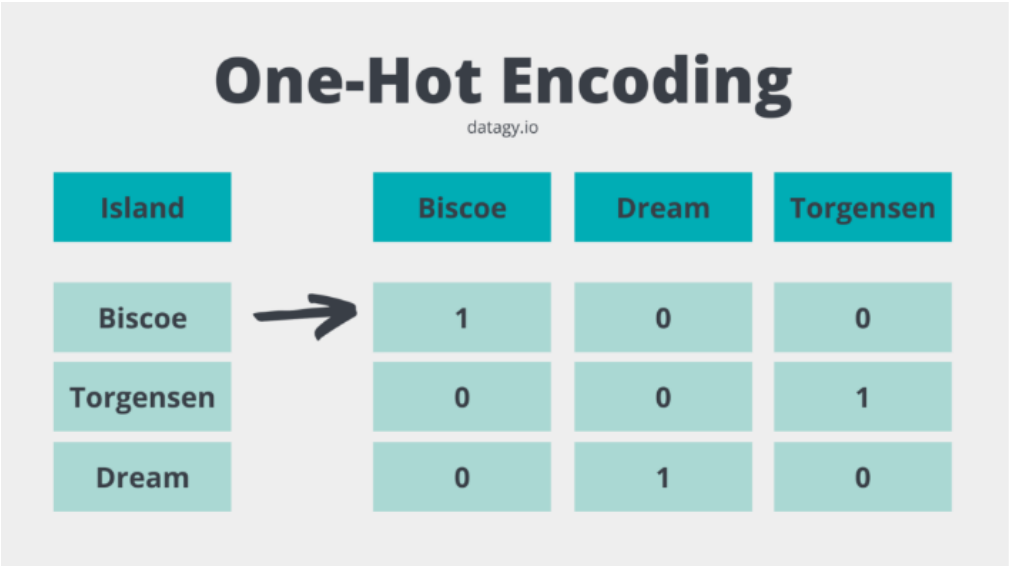




To remove the outliers from the movie dataset we have applied isolation forest technique.

3. One hot encoding:

One hot encoding can be defined as the essential process of converting the categorical data variables to be provided to machine and deep learning algorithms which in turn improve predictions as well as classification accuracy of a model. One Hot Encoding is a common way of pre-processing categorical features for machine learning models.



4. Standardization:

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Just to give you an example — if you have multiple independent variables like age, salary, and height; With their range as (18–100 Years), (25,000–75,000 Euros), and (1–2 Meters) respectively, feature scaling would help them all to be in the same range, for example- centered around 0 or in the range (0,1) depending on the scaling technique.

In order to perform feature scaling in our project we have used specific scaling technique name as Standard Scaler which transforms the data using below formula-

$$x' = \frac{x - \bar{x}}{\sigma}$$

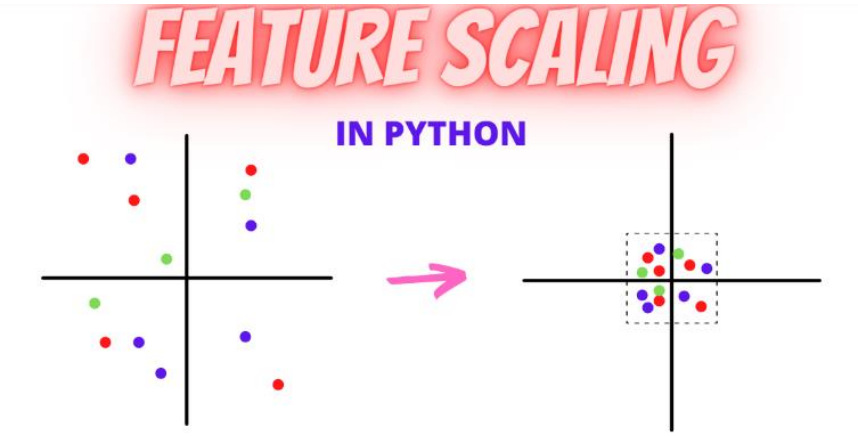
Where, x' = value after transformation

x = actual value of feature point

X_bar = mean of the all feature values

Sigma = Standard deviation of feature.

we can also see a figure here to understand how feature scaling works.



Natural Language Processing (NLP) Model:

For the NLP portion of this project, I will first convert all plot descriptions to word vectors so they can be processed by the NLP model. Then, the similarity between all word vectors will be calculated using cosine similarity (measures the angle between two vectors, resulting in a score

between -1 and 1, corresponding to complete opposites or perfectly similar vectors). Finally, I will extract the 5 movies or TV shows with the most similar plot description to a given movie or TV show.

Tf-idf vectorization:

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is a very common algorithm to transform text into a meaningful representation of numbers which is used to fit a machine learning algorithm for prediction.

We have also utilized the PCA because it can help us improve performance at a very low cost of model accuracy. Other benefits of PCA include reduction of noise in the data, feature selection (to a certain extent), and the ability to produce independent, 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.

Lemmatization

Stemming algorithm works by cutting the suffix from the word. In a broader sense cuts either the beginning or end of the word.

On the contrary, Lemmatization is a more powerful operation, and it takes into consideration morphological analysis of the words. It returns the lemma which is the base form of all its inflectional forms. In-depth linguistic knowledge is required to create dictionaries and look for the proper form of the word. Stemming is a general operation while lemmatization is an intelligent operation where the proper form will be looked in the dictionary. Hence, lemmatization helps in forming better machine learning features.

Code to distinguish between Lemmatization and

Stemming Code:

```
import nltk
from nltk.stem.porter import PorterStemmer
porter_stemmer  = PorterStemmer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
print("Stemming for {} is {}".format(w,porter_stemmer.stem(w)))
```

Output:

```
Stemming for studies is studi
Stemming for studying is studi
```

```
Stemming for cries is cri
Stemming for cry is cri
```

Lemmatization Code:

```
import nltk
        from nltk.stem import  WordNetLemmatizer
        wordnet_lemmatizer = WordNetLemmatizer()
        text = "studies studying cries cry"
        tokenization = nltk.word_tokenize(text)
         for w in tokenization:
               print("Lemma for {} is {}".format(w,
wordnet_lemmatizer.lemmatize(w)))
```

Output:

```
Lemma for studies is study
Lemma for studying is studying
Lemma for cries is cry
Lemma for cry is cry
```

Discussion of Output

If you look stemming for studies and studying, output is same (studi) but NLTK lemmatizer provides different lemma for both tokens study for studies and studying for studying. So when we need to make feature set to train machine, it would be great if lemmatization is preferred.

Use Case of Lemmatizer

Lemmatizer minimizes text ambiguity. Example words like bicycle or bicycles are converted to base word bicycle. Basically, it will convert all words having the same meaning but different representation to their base form. It reduces the word density in the given text and helps in preparing the accurate features for training machine. Cleaner the data, the more intelligent and accurate your machine learning model, will be. NLTK Lemmatizer will also saves memory as well as computational cost.

Real Time example showing use of Wordnet Lemmatization and POS Tagging in Python

```
from nltk.corpus import wordnet as wn
        from nltk.stem.wordnet import WordNetLemmatizer
        from nltk import word_tokenize, pos_tag
        from collections import defaultdict
        tag_map = defaultdict(lambda : wn.NOUN)
        tag_map['J'] = wn.ADJ
        tag_map['V'] = wn.VERB
        tag_map['R'] = wn.ADV

        text = "guru99 is a totally new kind of learning experience."
```

Code Explanation

- Firstly, the corpus reader wordnet is imported.
- WordNetLemmatizer is imported from wordnet.
- Word tokenize as well as parts of speech tag are imported from nltk.
- Default Dictionary is imported from collections.
- Dictionary is created where pos_tag (first letter) are the key values whose values are mapped with the value from wordnet dictionary. We have taken the only first letter as we will use it later in the loop.

- Text is written and is tokenized.
- Object lemma_function is created which will be used inside the loop.
- Loop is run and lemmatize will take two arguments one is token and other is a mapping of pos_tag with wordnet value.

Python Lemmatization has a close relation with wordnet dictionary, so it is essential to study this topic, so we keep this as the next topic.

Clustering:

- ## 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 8 clusters should be an optimal number of clusters.

Silhouette Score

Silhouette score is used to evaluate the quality of clusters created using clustering algorithms such as K-Means in terms of how well samples are clustered with other samples that are similar to each other. The Silhouette score is calculated for each sample of different clusters. To calculate the Silhouette score for each observation/data point, the following distances need to be found out for each observations belonging to all the clusters:

- Mean distance between the observation and all other data points in the same cluster. This distance can also be called a **mean intra-cluster distance.** The mean distance is denoted by **a**
- Mean distance between the observation and all other data points of the next nearest cluster. This distance can also be called a **mean nearest-cluster distance.** The mean distance is denoted by **b**

Silhouette score, **S,** for each sample is calculated using the following formula:

$$S = \frac{(b - a)}{max(a, b)}$$

The value of the Silhouette score varies from -1 to 1. If the score is 1, the cluster is dense and well-separated than other clusters. A value near 0 represents overlapping clusters with samples very close to the decision boundary of the neighboring clusters. A negative score [-1, 0] indicates that the samples might have got assigned to the wrong clusters.

K-Means clustering is an unsupervised learning algorithm. There is no labeled data for this clustering, unlike in supervised learning. K-Means performs the division of objects into clusters that share similarities and are dissimilar to the objects belonging to another cluster.

The term ‘K’ is a number. You need to tell the system how many clusters you need to create. For example, K = 2 refers to two clusters. There is a way of finding out what is the best or optimum value of K for a given data.

For a better understanding of k-means, let's take an example from cricket. Imagine you received data on a lot of cricket players from all over the world, which gives information on the runs scored by the player and the wickets taken by them in the last ten matches. Based on this information, we need to group the data into two clusters, namely batsman and bowlers.

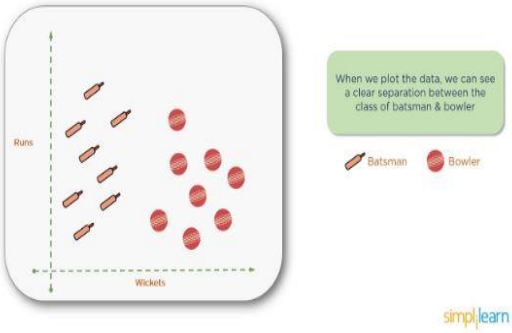
Let's take a look at the steps to create these clusters.

Solution:

Assign data points

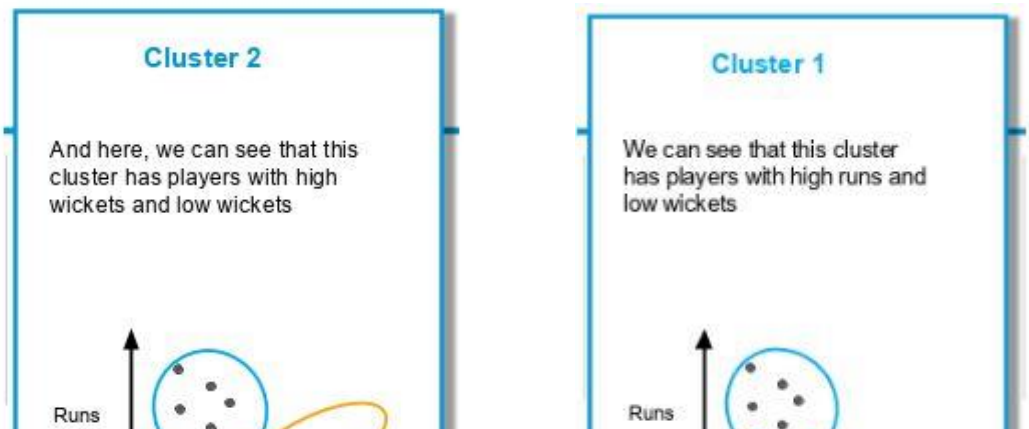
Here, we have our data set plotted on ‘x’ and ‘y’ coordinates. The information on the y-axis is about the runs scored, and on the x-axis about the wickets taken by the players.

If we plot the data, this is how it would look:



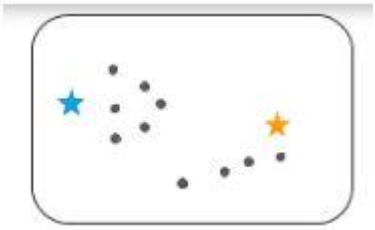
Perform Clustering

We need to create the clusters, as shown below:



Considering the same data set, let us solve the problem using K-Means clustering (taking K = 2).

The first step in k-means clustering is the allocation of two centroids randomly (as K=2). Two points are assigned as centroids. Note that the points can be anywhere, as they are random points. They are called centroids, but initially, they are not the central point of a given data set.



The next step is to determine the distance between each of the randomly assigned centroids' data points. For every point, the distance is measured from both the centroids, and whichever distance is less, that point is assigned to that centroid. You can see the data points attached to the centroids and represented here in blue and yellow.

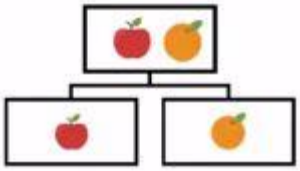
The next step is to determine the actual centroid for these two clusters. The original randomly allocated centroid is to be repositioned to the actual centroid of the clusters.

This process of calculating the distance and repositioning the centroid continues until we obtain our final cluster. Then the centroid repositioning stops.

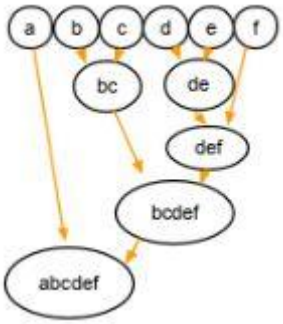
As seen above, the centroid doesn't need anymore repositioning, and it means the algorithm has converged, and we have the two clusters with a centroid.

Hierarchical Clustering :

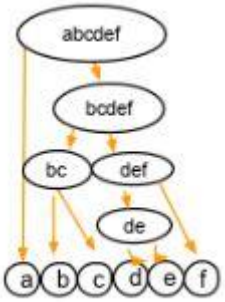
Hierarchical clustering uses a tree-like structure, like so:



In agglomerative clustering, there is a bottom-up approach. We begin with each element as a separate cluster and merge them into successively more massive clusters, as shown below:



Divisive clustering is a top-down approach. We begin with the whole set and proceed to divide it into successively smaller clusters, as you can see below:



DBSCAN:

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.
   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 *eps distance*. This is a chaining process. So, if *b* is neighbor of *c*, *c* is neighbor of *d*, *d* is neighbor of *e*, which in turn is neighbor of *a* implies that *b* is neighbor of *a*.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.