Clustering Analysis on

NETFLIX MOVIES AND TV SHOWS

Rohan s. Jagadale
Data science trainee at Almabetter

Abstract

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.

Introduction

Netflix's recommendation system helps them increase their popularity among service providers as they help increase the number of items sold, offer a diverse selection of items, increase user satisfaction, as well as user loyalty to the company, and they are very helpful in getting a better understanding of what the user wants. Then it's easier to get the user to make better decisions from a wide variety of movie products.

With over 139 million paid subscribers(total viewer pool -300 million) across 190 countries, 15,400 titles across its regional libraries and 112 Emmy Award Nominations in 2018 — Netflix is the world's leading Internet television network and the most-valued largest streaming service in the world. The amazing digital success story of Netflix is incomplete without the mention of its recommender systems that focus on personalization.

There are several methods to create a list of recommendations according to your preferences. You can use (Collaborative-filtering) and (Content-based Filtering) for recomendation.

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.

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

Objective

The project's main goal is to create a model that can perform Clustering on comparable material by matching text-based attributes.

Dataset Peeping

The dataset has 7787 rows and 12 attributes to work with.

1. We have NaN values in the dataset.
2. Changed the format of the Date.
3. Added some columns which are extracted from the Date column.

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 : Genre
12. description: The Summary description

Challenges Faced

The following are the challenges faced in the data analysis:

- ➢ Conversion of Datetime features, categorical features.
- ➢ Feature engineering
- ➢ Model Implementation

Approach

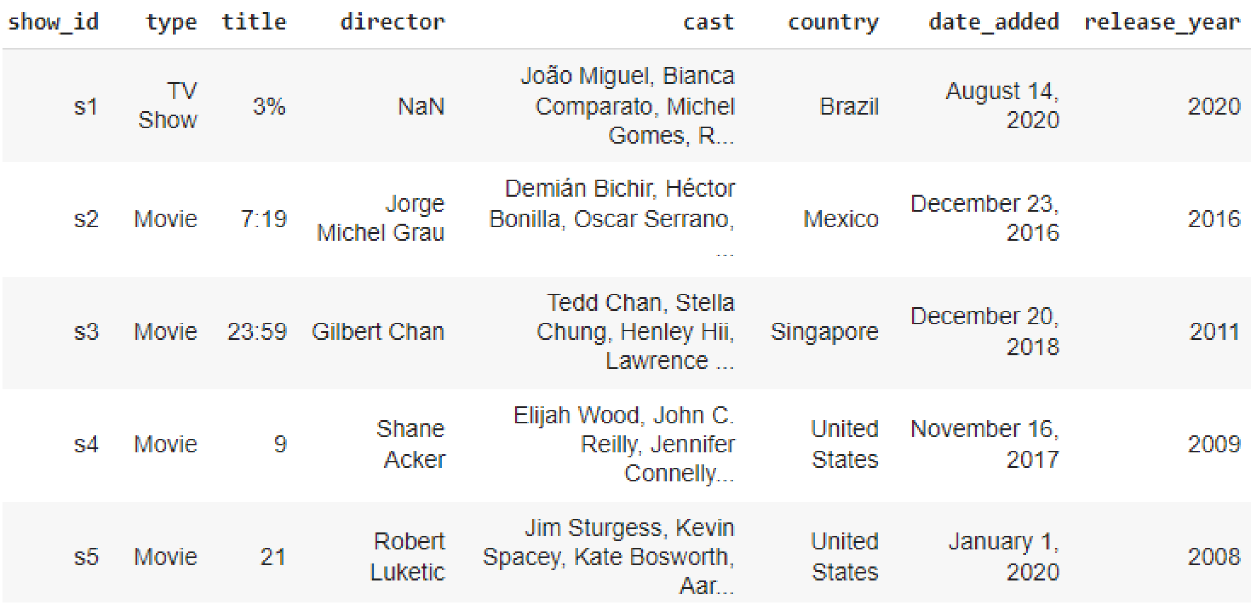
As the problem statement says, Understanding what type of content is available in different countries and Is Netflix increasingly focused on TV rather than movies in recent years we have to do clustering on similar content by matching text-based features. For that we used Affinity Propagation, Agglomerative Clustering, and K-means Clustering.

Tools Used

The whole project was done using python, in google colaboratory. Following libraries were used for analysing the data and visualizing it and to build the model to predict the bike count required at each hour for the stable supply of rental bikes.

- Pandas: Extensively used to load and wrangle with the dataset.
- Matplotlib: Used for visualization.
- Seaborn: Used for visualization.
- Datetime: Used for analysing the date variable.
- Warnings: For filtering and ignoring the warnings.
- Numpy: For some math operations in predictions.
- Sklearn: For the purpose of analysis and prediction.
- Datetime: For reading the date.

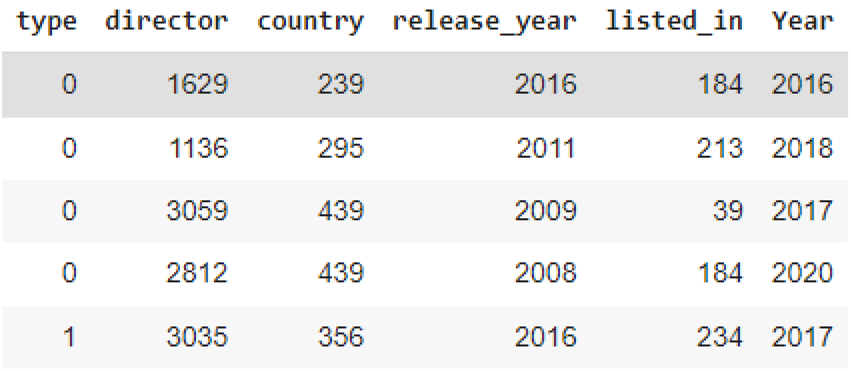
The below table shows the dataset in the form of Pandas DataFrame



show_id	type	title	director	cast	country	date_added	release_year
s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	2020
s2	Movie	7:19	Jorge Michel Grau	Demián Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	2016
s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hii, Lawrence ...	Singapore	December 20, 2018	2011
s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connelly...	United States	November 16, 2017	2009
s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	2008

Feature engineering

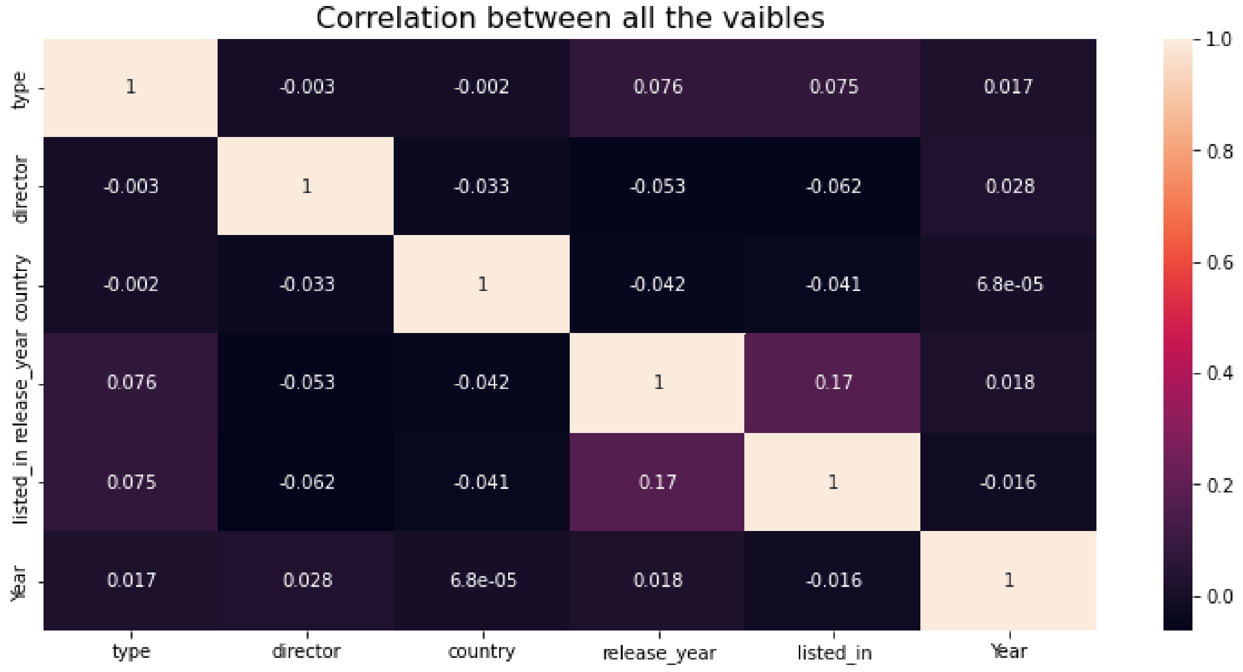
- ➔ There are Too much classes, so we just obtain the first 50 (the most common 50)
- ➔ Unify some of the similar types(genre)
- ➔ Make a dictionary with similar content by matching text-based features that we are going to use in clustering.



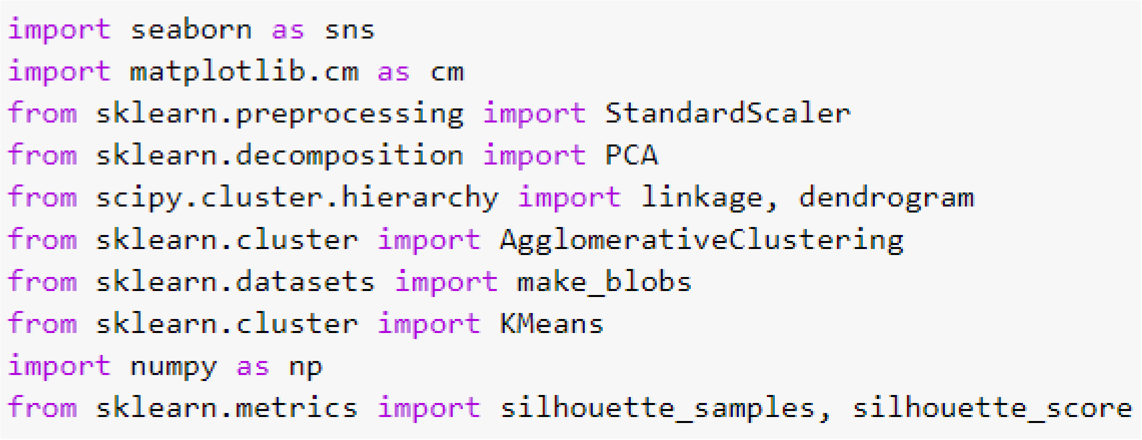
type	director	country	release_year	listed_in	Year
0	1629	239	2016	184	2016
0	1136	295	2011	213	2018
0	3059	439	2009	39	2017
0	2812	439	2008	184	2020
1	3035	356	2016	234	2017

These are the columns that we are going to use in Clustering.

Correlation Heatmap



Importing required libraries for clustering



```
import seaborn as sns
import matplotlib.cm as cm
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import linkage, dendrogram
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import numpy as np
from sklearn.metrics import silhouette_samples, silhouette_score
```

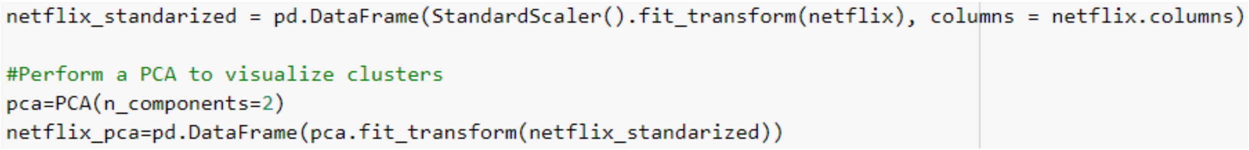
Building a clustering model

Clustering models allow you to categorize records into a certain number of clusters. This can help you identify natural groups in your data.

Clustering models focus on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics. In fact, you may not even know exactly how many groups to look for. This is what distinguishes clustering models from the other machine-learning techniques—there is no predefined output or target field for the model to predict. These models are often referred to as **unsupervised learning** models, since there is no external standard by which to judge the model's classification performance.

Scaling the data

We used standardScaler to transform the data.



```
netflix_standarized = pd.DataFrame(StandardScaler().fit_transform(netflix), columns = netflix.columns)

#Perform a PCA to visualize clusters
pca=PCA(n_components=2)
netflix_pca=pd.DataFrame(pca.fit_transform(netflix_standarized))
```

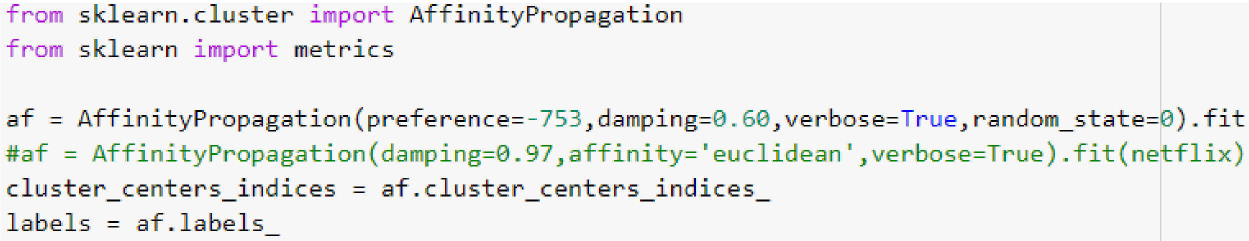
Silhouette Coefficient or silhouette score(meaning)

Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value **ranges from -1 to 1**. 1: Means clusters are well apart from each other and clearly distinguished. ... a= average intra-cluster distance i.e the average distance between each point within a cluster.

Model Implementation

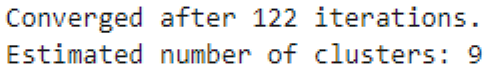
1. Affinity Propagation

Affinity propagation (AP) is **a graph based clustering algorithm** similar to k Means or K medoids, which does not require the estimation of the number of clusters before running the algorithm. Affinity propagation finds “exemplars” i.e. members of the input set that are representative of clusters.



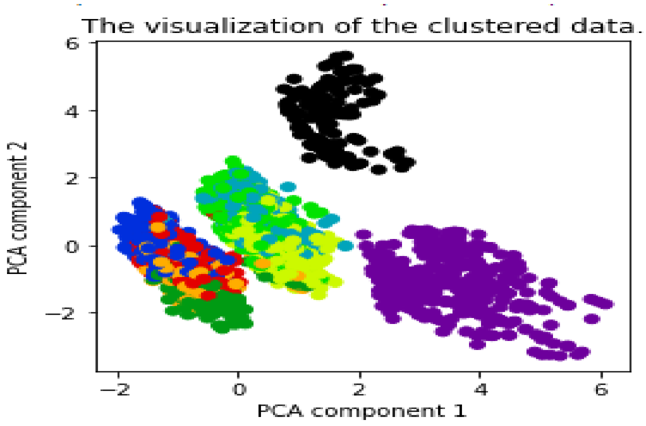
```
from sklearn.cluster import AffinityPropagation
from sklearn import metrics

af = AffinityPropagation(preference=-753,damping=0.60,verbose=True,random_state=0).fit
#af = AffinityPropagation(damping=0.97,affinity='euclidean',verbose=True).fit(netflix)
cluster_centers_indices = af.cluster_centers_indices_
labels = af.labels_
```

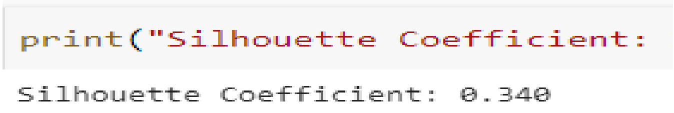
We used euclidean distance as an affinity estimator.
After that, number of clusters we got here:



```
Converged after 122 iterations.
Estimated number of clusters: 9
```

Visualization of custer and Silhouette Coefficient



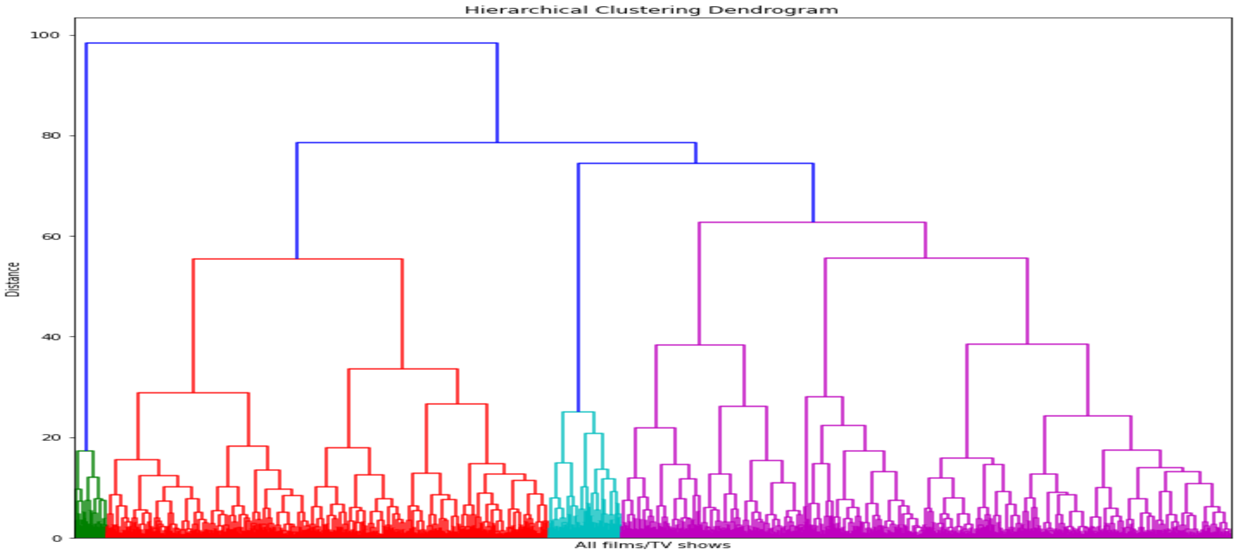


```
print("Silhouette Coefficient:
```

```
Silhouette Coefficient: 0.340
```

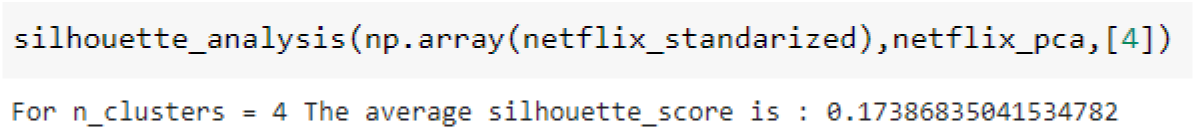
2.Agglomerative Clustering

The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. ... Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects.
We used a dendrogram to find the number of clusters.



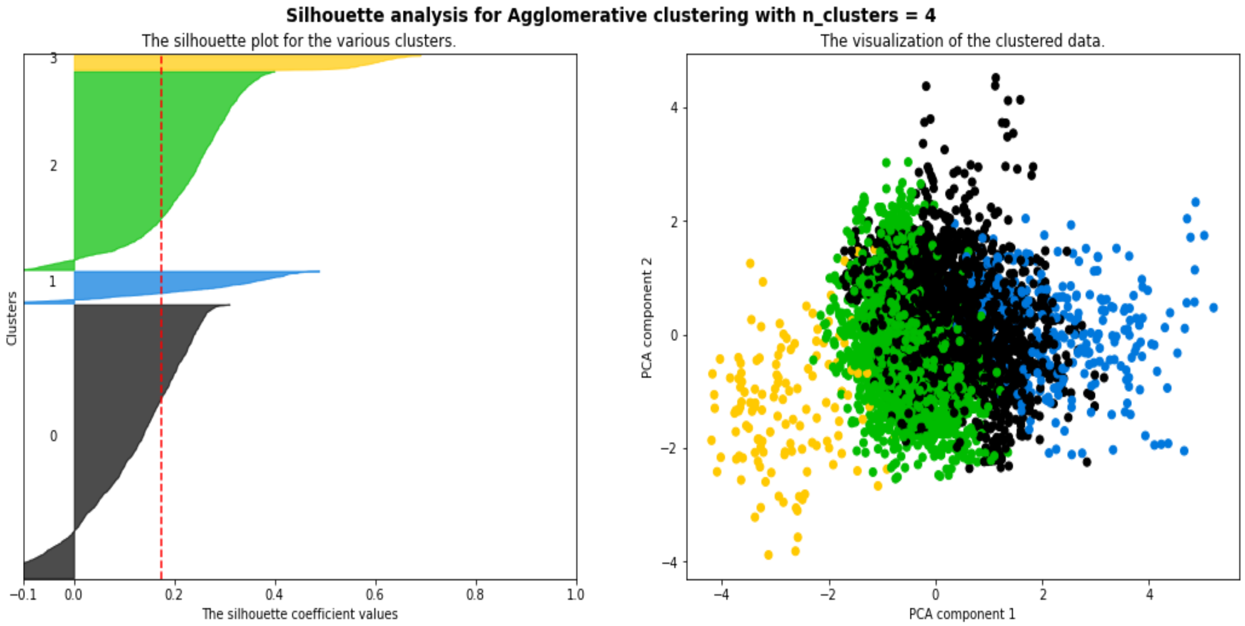
Assume we cut vertical lines with a horizontal line to obtain the number of clusters. **Number of clusters = 4**

Silhouette_score and visualization



```
silhouette_analysis(np.array(netflix_standarized),netflix_pca,[4])
```

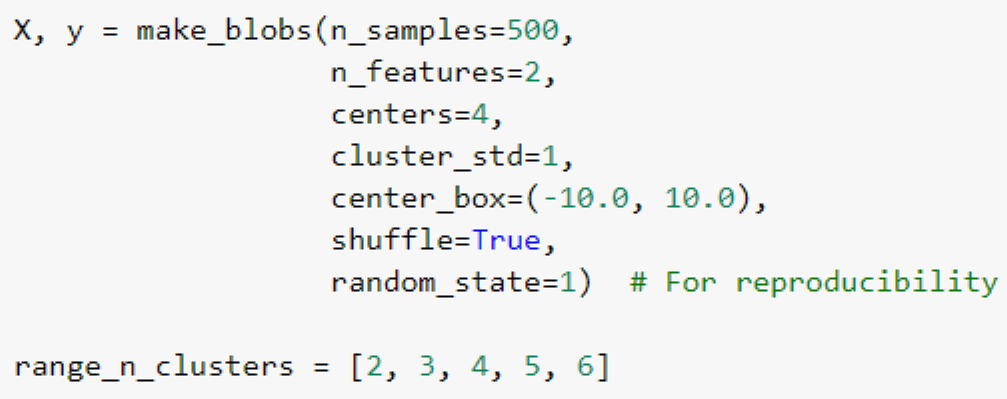
```
For n_clusters = 4 The average silhouette_score is : 0.17386835041534782
```



3.K-means Clustering

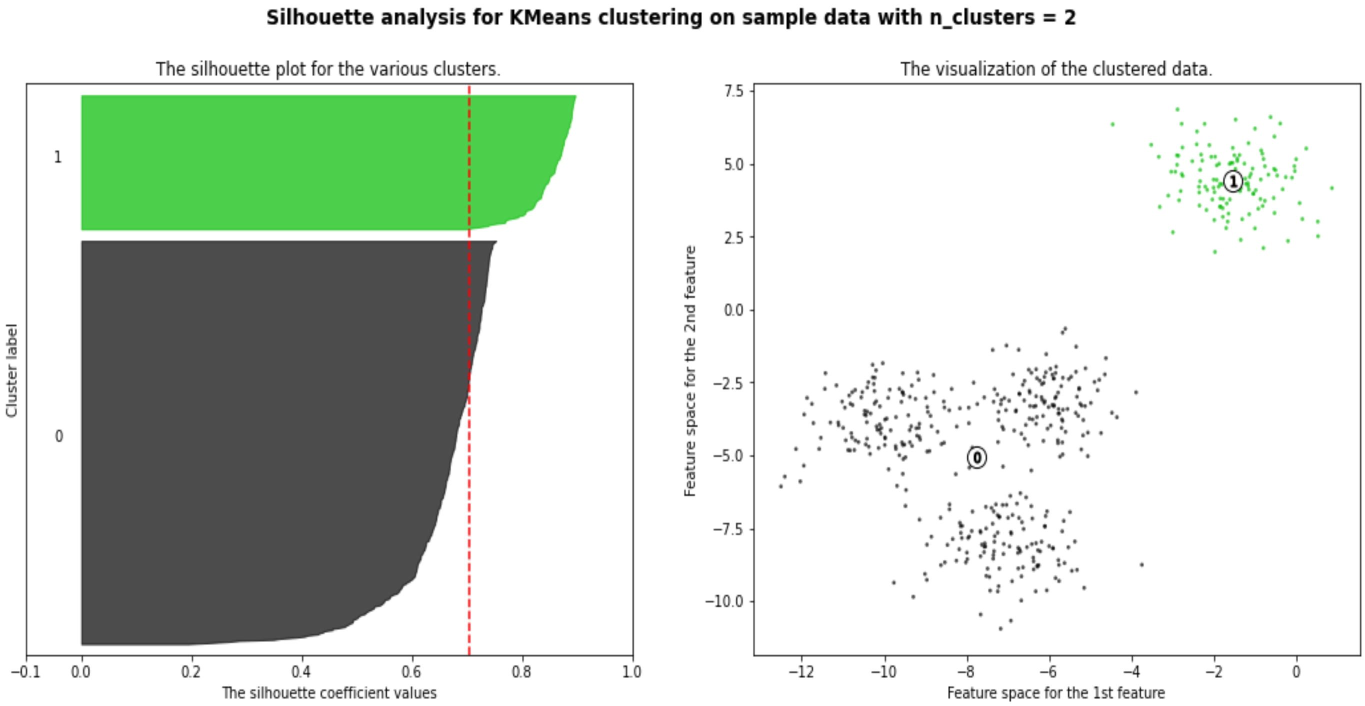
k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

We created the sample data using build blobs and used range_n_clusters to specify the number of clusters we wanted to utilize in k means.



```
X, y = make_blobs(n_samples=500,
                  n_features=2,
                  centers=4,
                  cluster_std=1,
                  center_box=(-10.0, 10.0),
                  shuffle=True,
                  random_state=1)  # For reproducibility

range_n_clusters = [2, 3, 4, 5, 6]
```

Silhouette_score and visualization

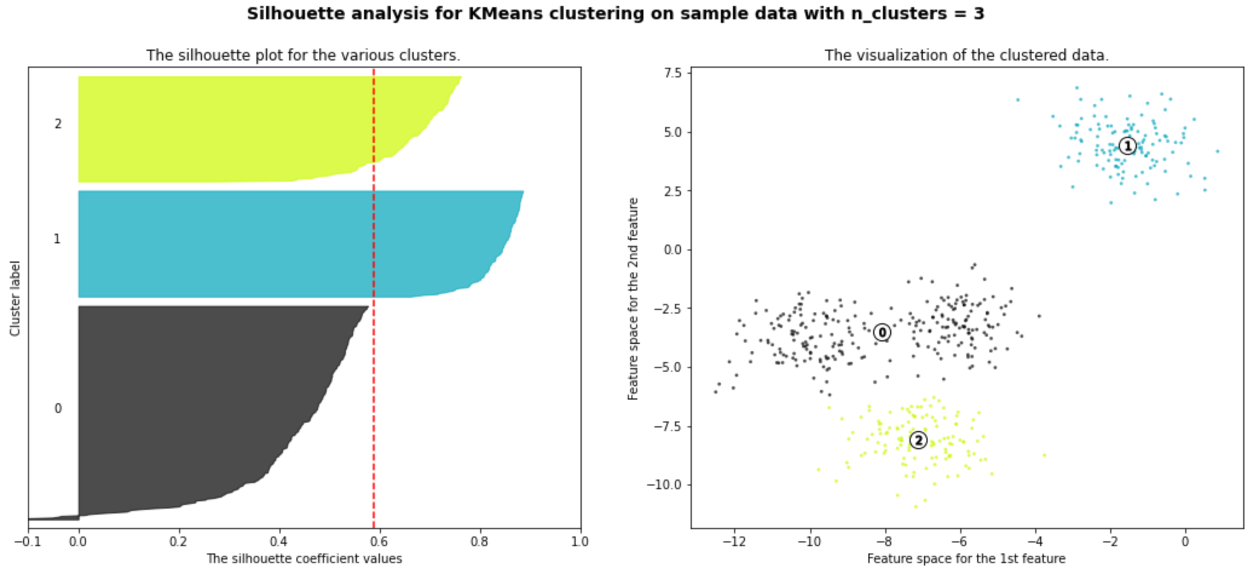


Silhouette analysis for KMeans clustering on sample data with n_clusters = 2

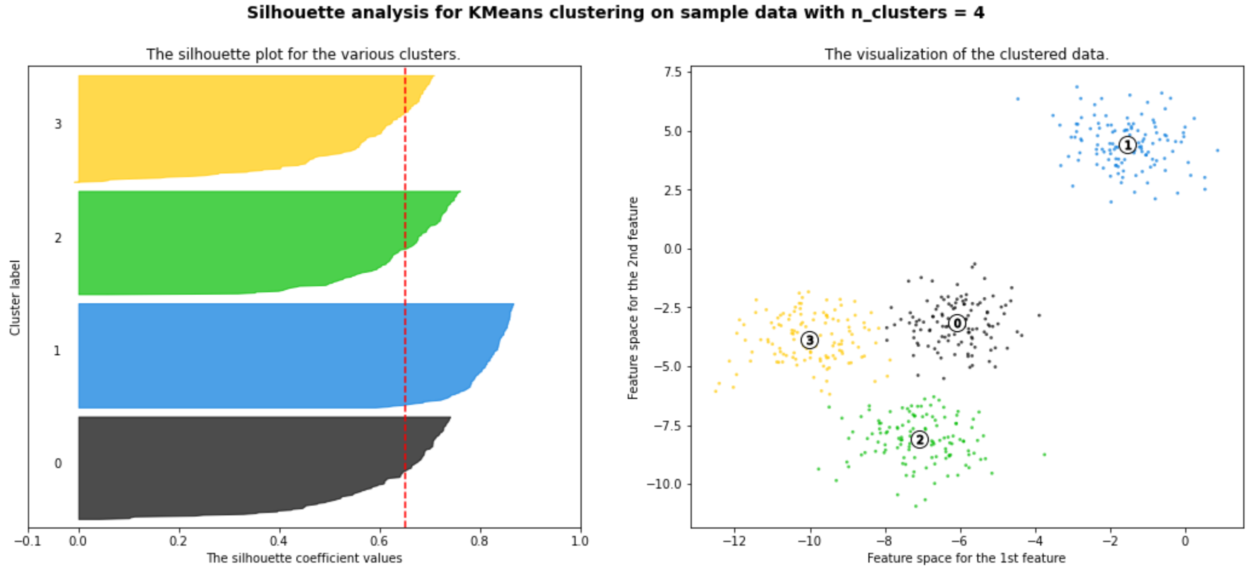
The silhouette plot for the various clusters.

The visualization of the clustered data.

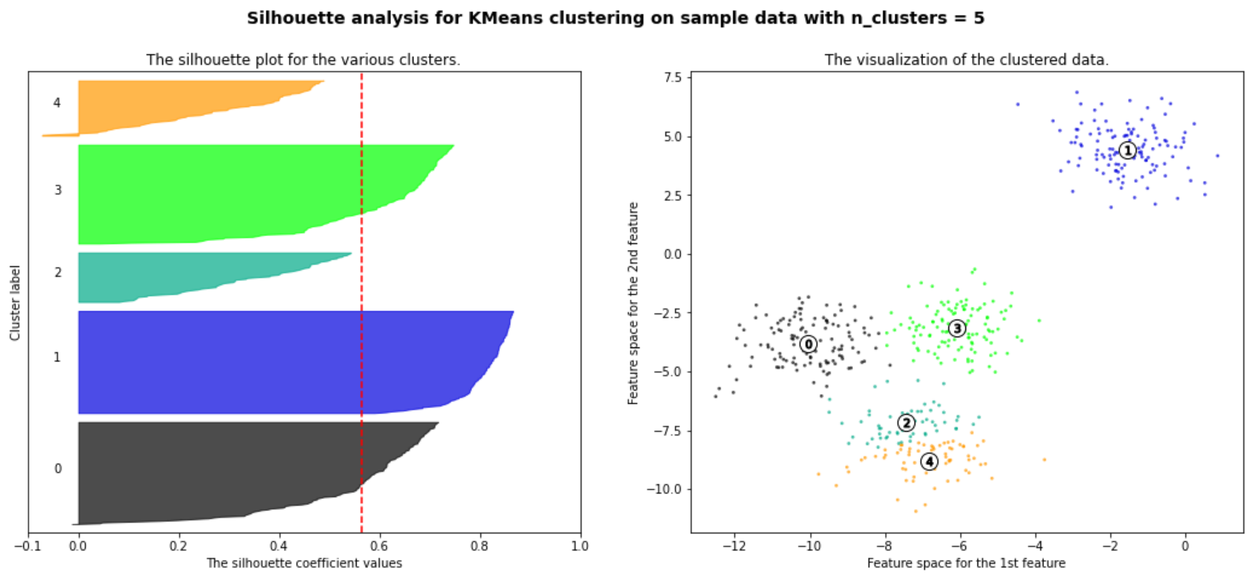
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262



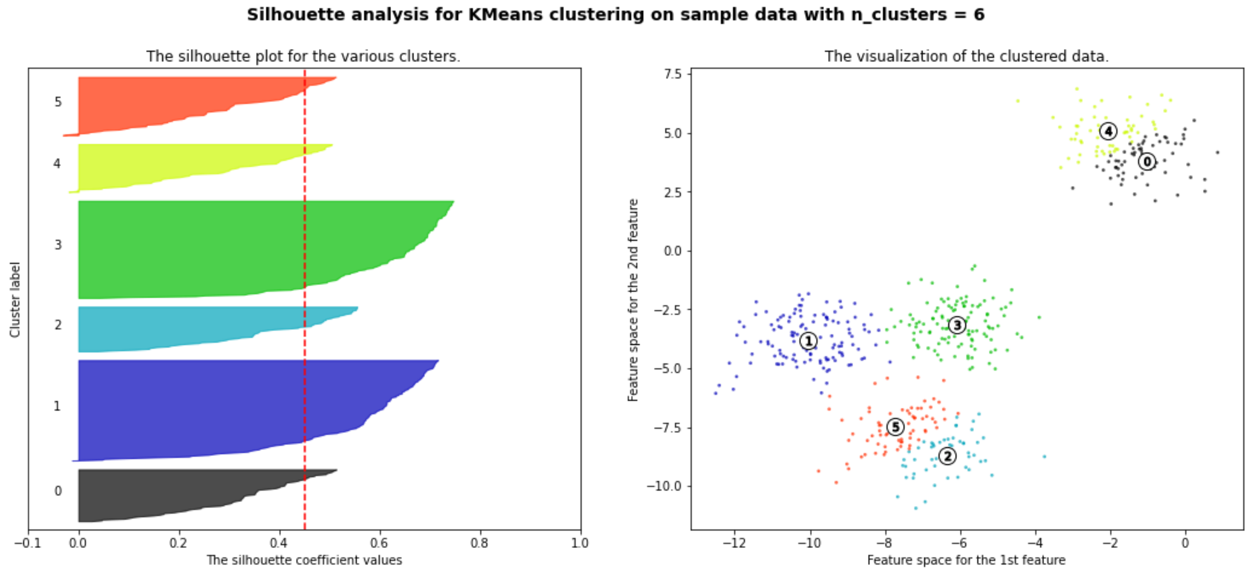
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721



For n_clusters = 4 The average silhouette_score is : 0.6505186632729437



```
For n_clusters = 5 The average silhouette_score is : 0.56376469026194
```

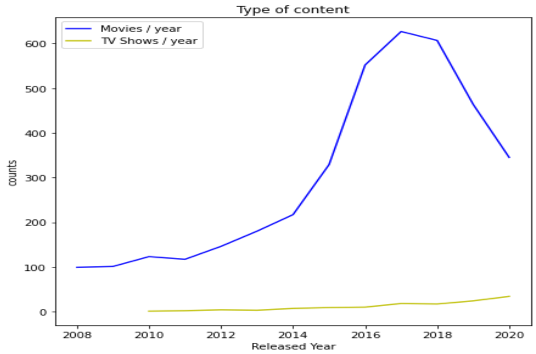


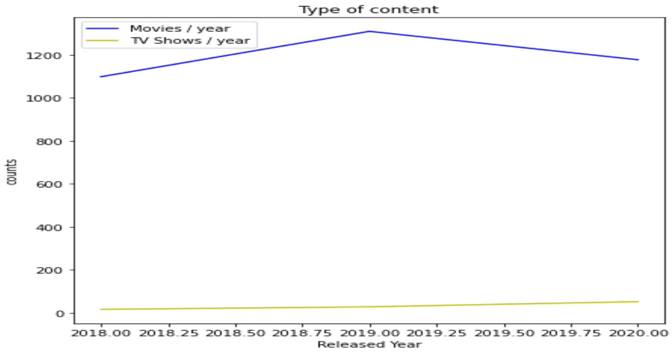
```
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

We plot line graph on

1.Actual Release year of the movie / show

2. Date it was added on Netflix



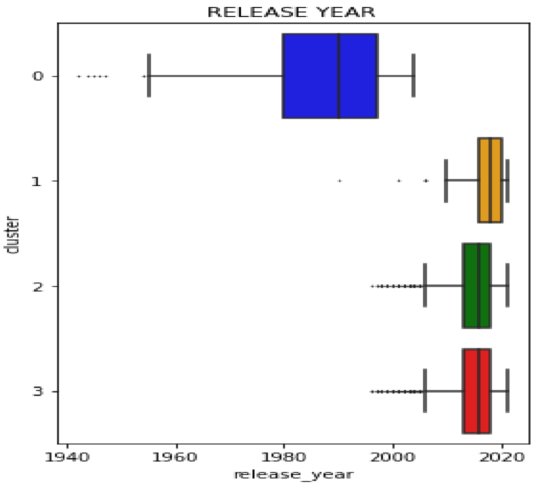


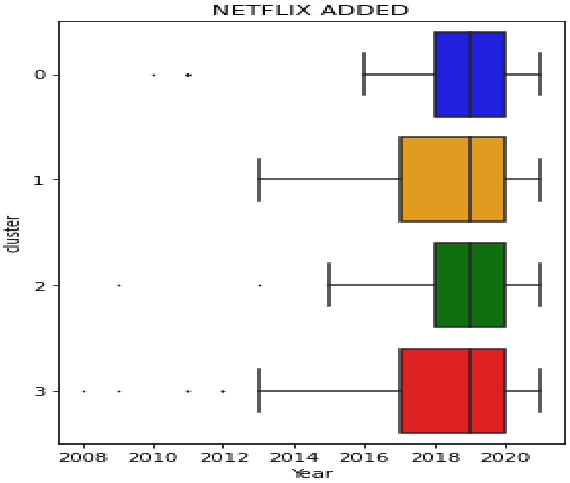
Hypothesis from the data visualized

1. According to the first graph, the number of TV shows launched in the previous few years is growing.

2. According to the second graph, the number of TV shows added to Netflix is stable.

We also plot some boxplots for our clusters





Hypothesis after clustering

1. After clustering, we can say that our alternative hypothesis is that the number of TV shows launched in the previous few years is NOT growing.

2. Our second alternative hypothesis is the number of TV shows added to Netflix is Higher.

Conclusion :

- **We started by removing nan values and converting the Netflix added date to year, month, and day using date time format.**
- **Most films were released in the years 2018, 2019, and 2020.**
- **The months of October, November, December and January had the largest number of films and television series released.**
- **TV shows account for 2.8 percent of the total, while movies account for 97.2 percent.**
- **The United States, India, the United Kingdom, Canada, and Egypt are the top five producer countries.**
- **Netflix has added a lot more movies and TV episodes in the previous years, but the numbers are still low when compared to movies released in the last ten years.**
- **We did feature engineering, which involved removing certain variables and preparing a dataframe to feed the clustering algorithms.**
- **For the clustering algorithm, we utilized type, director, nation, released year, genre, and year.**
- **Affinity Propagation, Agglomerative Clustering, and K-means Clustering were utilised to build the model.**
- **In Affinity Propagation, we had 9 clusters and a Silhouette Coefficient score of 0.340.**
- **A dendrogram was used to determine the number of clusters in Agglomerative Clustering. There were two clusters, with an average silhouette score of 0.56590662228136.**
- **The final model we used was k-means clustering, which consisted of 2,3,4,5,6 clusters. 4 numbers of clusters gives us good fitting.**
- **After clustering, we can say that the number of TV shows launched in the previous years is NOT growing.**
- **The number of TV shows added to Netflix is higher in the last three years.**