Capstone Project

Clustering Analysis on
NETFLIX MOVIES AND TV SHOWS

Rohan Jagadale
Data science trainee at
Almabetter

Flow of the Presentation

- [ ] Introduction
- [ ] Problem Statement
- [ ] Data Pipeline
- [ ] Exploratory Data Analysis
- [ ] Feature engineering
- [ ] Model Implementation
- [ ] conclusion

Introduction

Netflix:

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.

Methodology:

- Unsupervised Machine Learning (Clustering)

Database:

- Netflix Movies and TV Shows
- 7787 rows and 12 columns
- Data from last decade

Problem Statement

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 has increasingly focusing on TV rather than movies in recent years.
4. Clustering similar content by matching text-based features

Data Pipeline

- **Data processing:** At first phase checked for null Values and changed the datetime containing column in dataset.
- **EDA:** Exploratory analysis was done on the Features selected in the first phase.
- **Feature engineering:** Unify some of the similar Types (genre) and Make a dictionary with matching Text based features that we are going to use in clustering.
- **Prepare Dataframe:** Delete some features, we prepare df to feed the clustering algorithms.
- **Create a model:** Finally in this part created models on clustering.

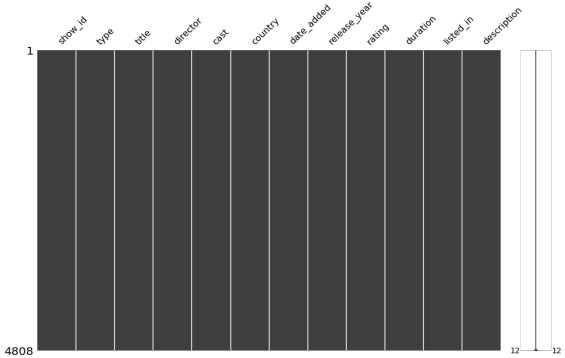
Data Description

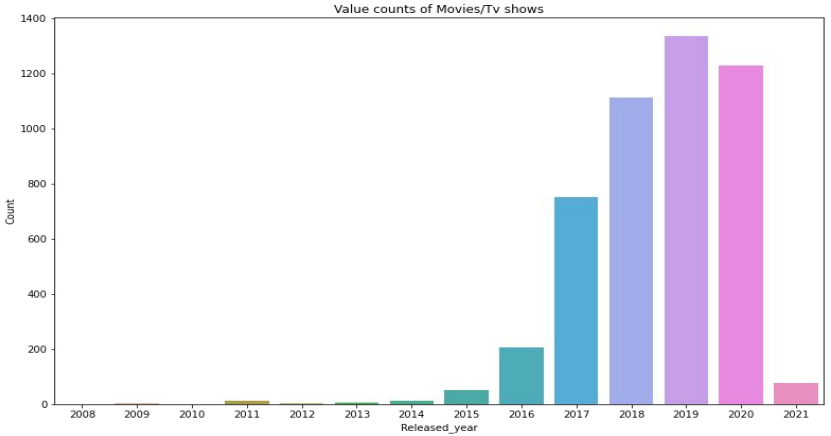
Attribute Information :

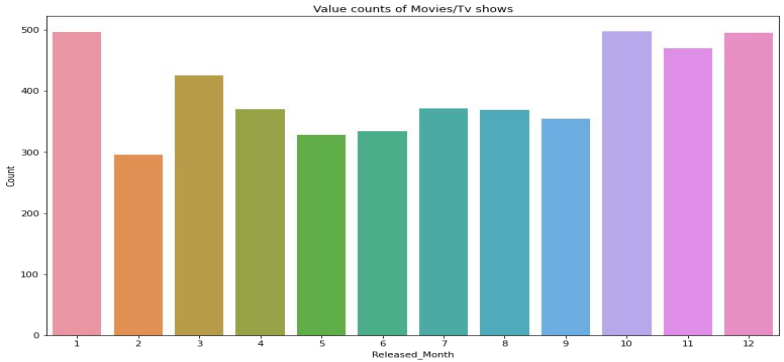
1. show_id : Unique ID for every Movie / Tv Show
2. type : Identifier - A Movie or TV Show
3. title : Title of the Movie / Tv Show
4. director : Director of the Movie
5. cast : Actors involved in the movie / show
6. country : Country where the movie / show was produced
7. date_added : Date it was added 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

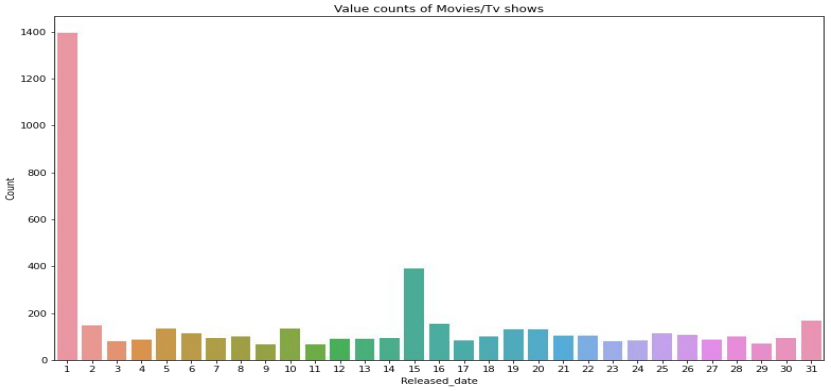
Exploratory Data Analysis

- Check Null Values
- Visualization on features

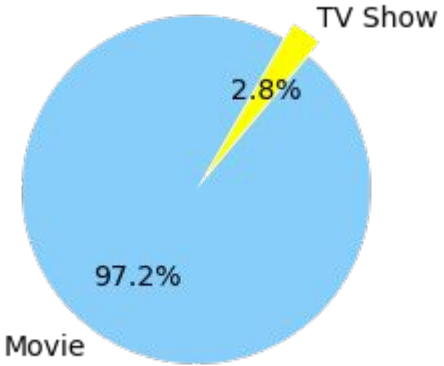


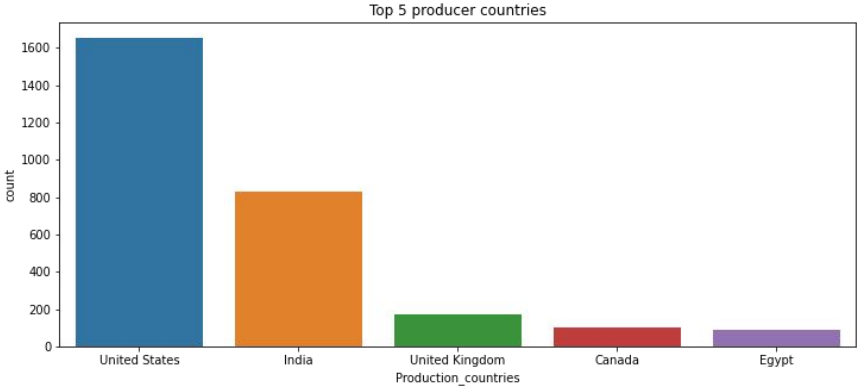


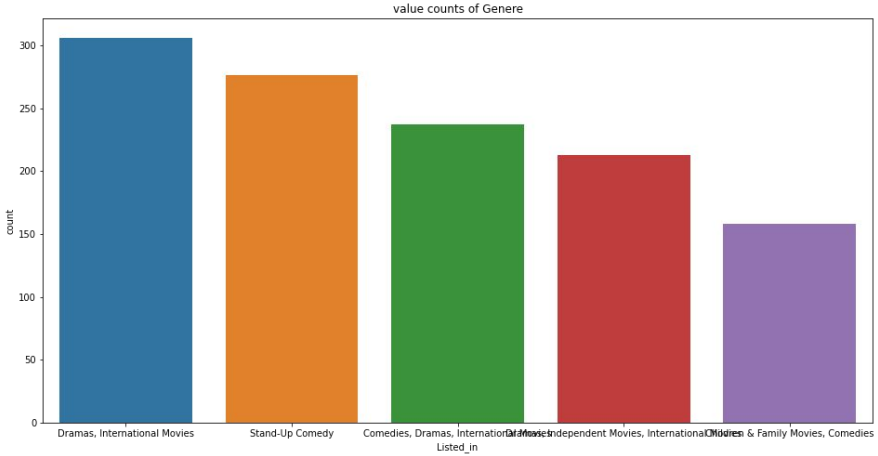


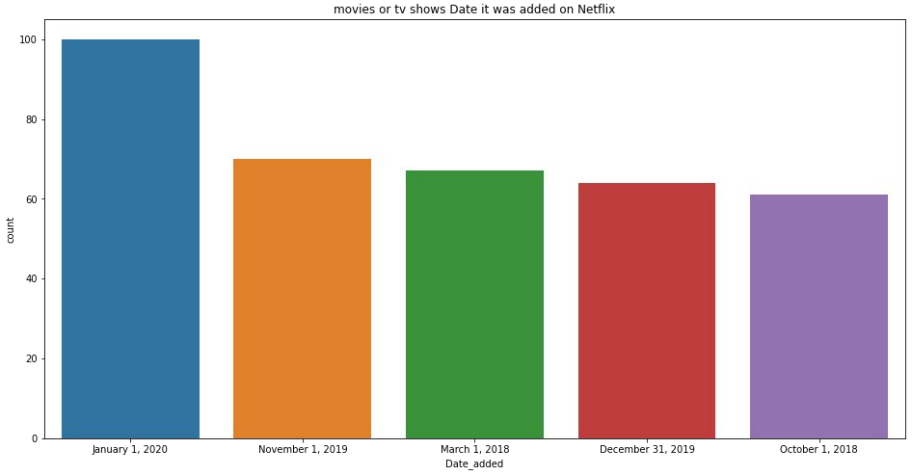


Exploratory Data Analysis



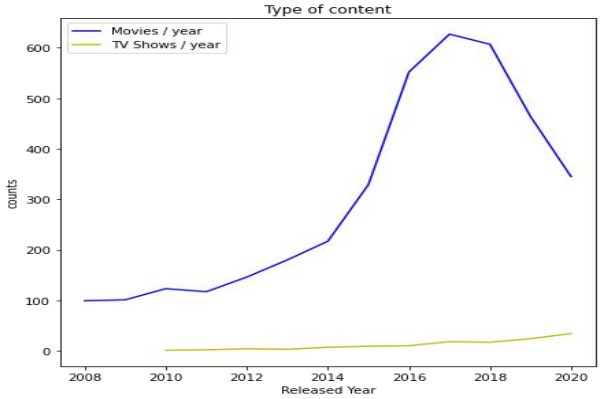


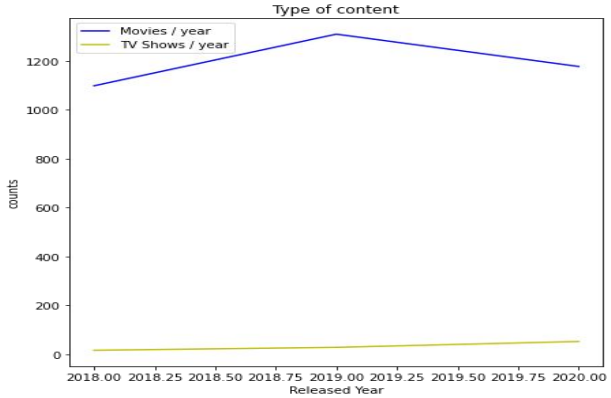




Exploratory Data Analysis

- Visualizing number of Movies and TV Shows
- Make DataFrame using certain columns for clustering
- Correlation of columns





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

Data Preparation

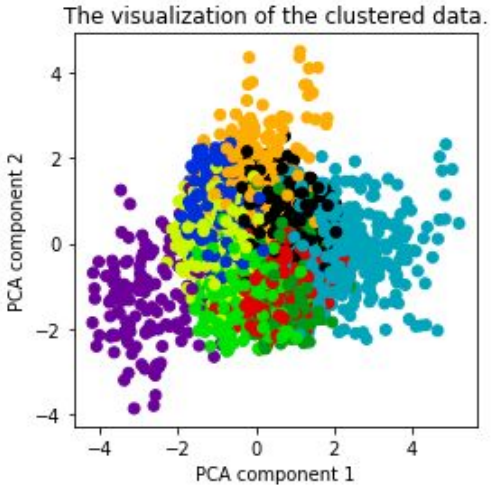
- Feature engineering
- Select required columns
- Import required libraries for clustering
- StandardScaler was used to scale the data.

```
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
```

```
# transform the data using StandardScaler
netflix_standarized = pd.DataFrame(StandardScaler().fit_transform(netflix)

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

Model Implementation

1. Affinity Propagation



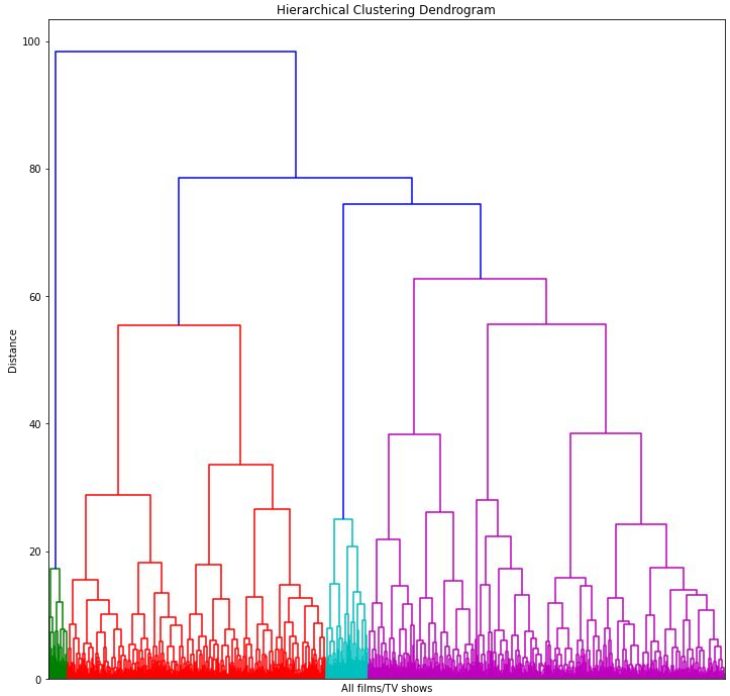

```
from sklearn.cluster import AffinityPropagation
from sklearn import metrics

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_

n_clusters_ = len(cluster_centers_indices)

print('Estimated number of clusters: %d' % n_clusters_)
```

```
Silhouette Coefficient: 0.308
```

```
Converged after 123 iterations.
Estimated number of clusters: 9
```

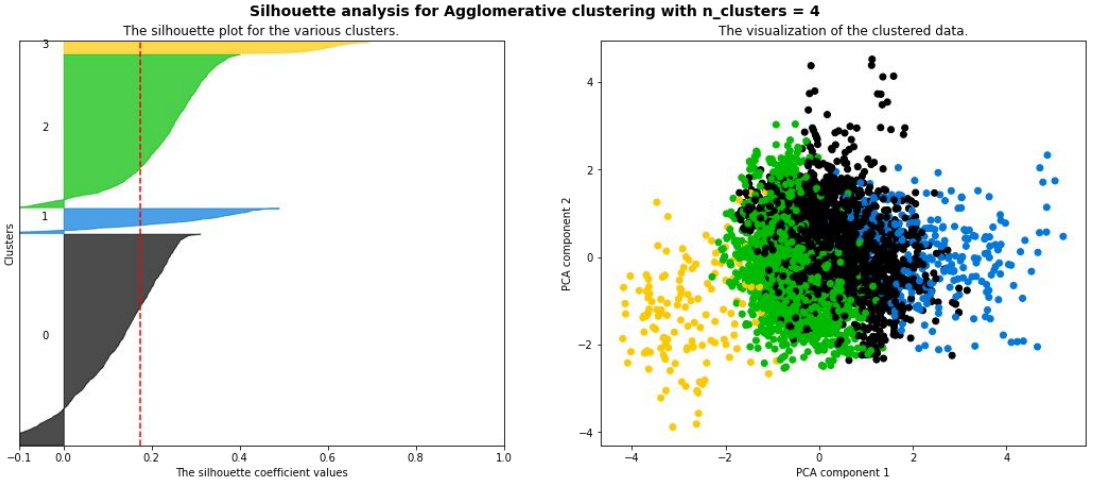
Model Implementation

2. Agglomerative Clustering



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

Number of clusters = 4

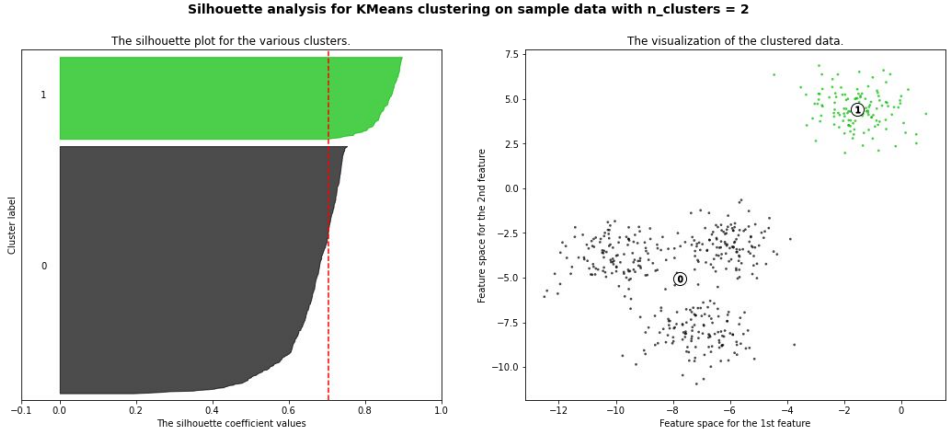


The average silhouette_score is : 0.17386835041534782 which is not good.

Model Implementation

3. k-means clustering

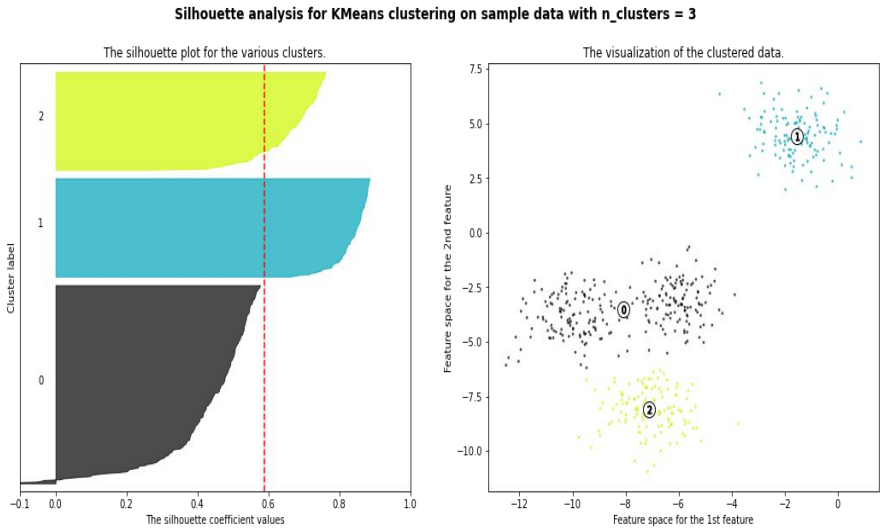
```
X, y = make_blobs(n_samples=500,
                  n_features=2,
                  centers=4,
                  cluster_std=1,
                  center_box=(-10.0, 10.0),
                  shuffle=True,
                  random_state=1)

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

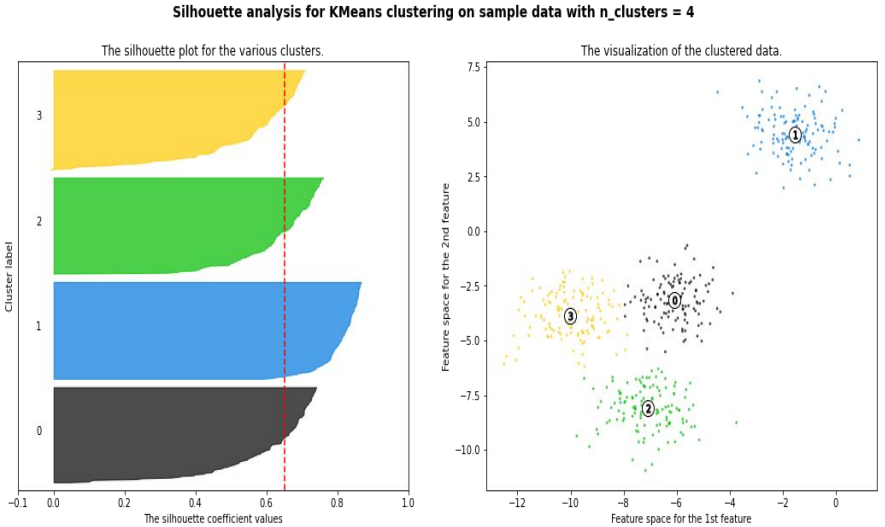



```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.56376469026194
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

Model Implementation

3. k-means clustering

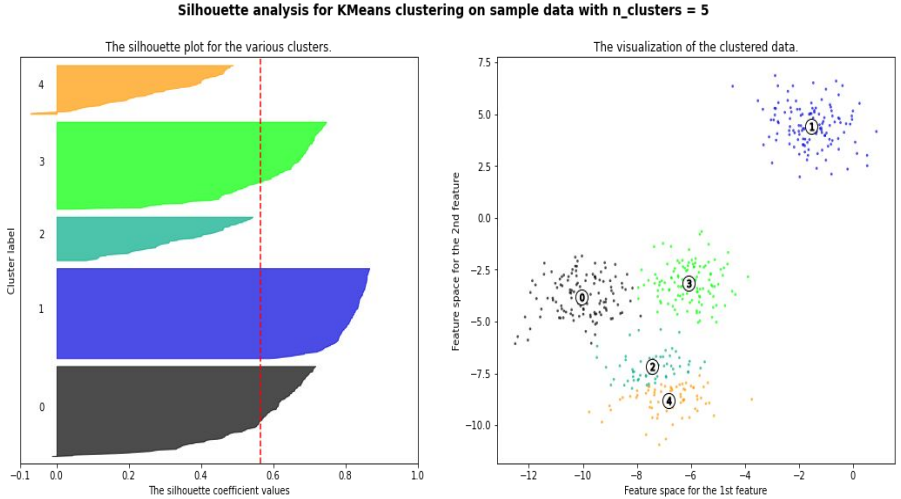


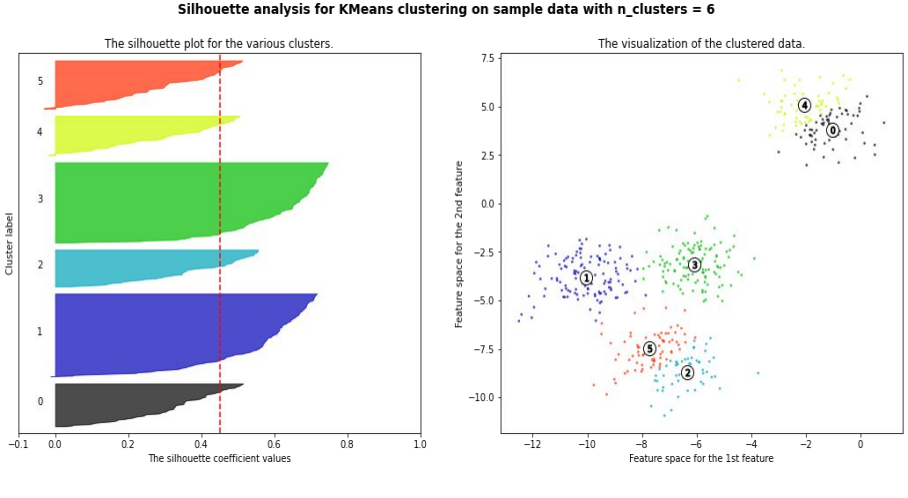



```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.56376469026194
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

Model Implementation

3. k-means clustering






```
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
```

Model Implementation

3. k-means clustering

Hypothesis from the data visualized

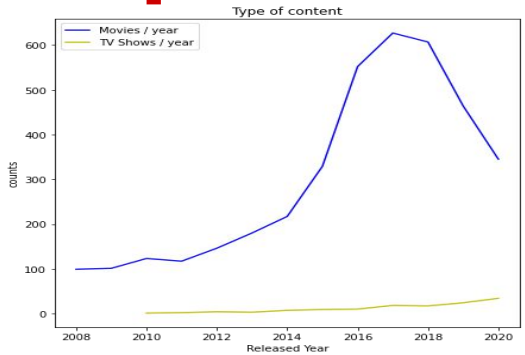
1. According to the first graph, the number of TV shows launched in the previous years is growing.

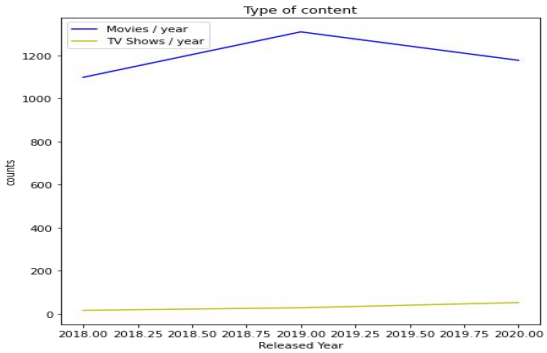
2. According to the second graph, the number of TV shows added to Netflix is stable.

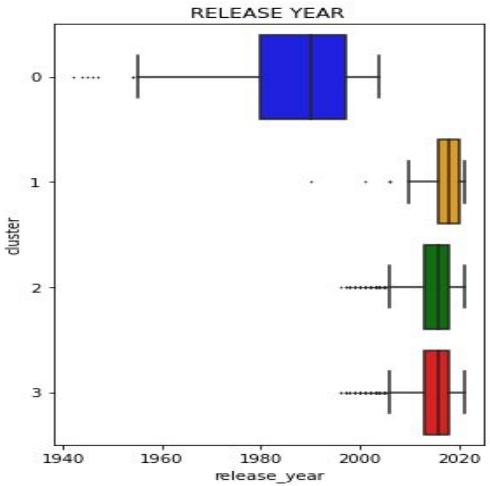
Hypothesis after clustering

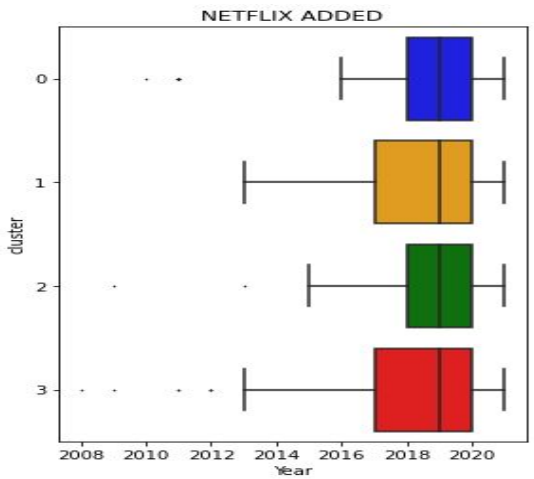
1. After clustering, we can say that our alternative hypothesis is that the number of TV shows launched in the previous years is NOT growing.

2. second alternative hypothesis is 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 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 tv shows released.
- TV shows account for 2.8 percent of the total, while movies account for 97.2 percent.
- The USA, India, the United Kingdom, Canada, and Egypt are the top five producer countries.
- 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 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 four clusters, with an average silhouette score of 0.17386835041534782.
- 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.
- We observed that number of TV shows launched in the previous years is NOT increasing.
- The number of TV shows added to Netflix is higher in the last three years.

Thank you