Netflix Movies and TV Shows Clustering

Premanand Gaikwad
Data science trainees
AlmaBetter, Bangalore

Abstract:

Netflix is an organization that manages a giant series of TV indicates and movies, streaming it each time by means of online. This commercial enterprise is worthwhile due to the fact customers make a month-to-month price to get entry to the platform. However, clients can cancel their subscriptions at any time. Therefore, the agency should hold the customers hooked on the platform and no longer lose their interest. This is the place advice structures begin to play an essential role, offering precious recommendations to customers is essential.

Keywords: Cluster, topic modelling, NLP, features, genres.

1. Problem Statement

Netflix content material varies through location and can alternate over the years. You can watch numerous award-winning Netflix originals, TV shows, movies, documentaries, and more. The more you watch, the better Netflix gets at recommending TV shows and movies I think you'll enjoy.

We know that, Users of such a platform have experienced it ourselves that endless scrolling spend more time deciding what to watch than looking at their movie.

This project's goal is to cluster the films or television indicates on Netflix and construct a movie recommendation device for users. To start with, I am going to research available records from Netflix to get insights to know data of the records with the aid of different feature wise

2. Introduction

Netflix, Inc. is an American subscription streaming service and production company based in Los Gatos, California. Founded on August 29, 1997, Netflix had 220.7 million subscribers worldwide, including 73.3 million in the United States and Canada, 73.0 million in Europe, the Middle East and Africa, 39.6 million in Latin America and 34.8 million in the Asia-Pacific region.

Netflix is a subscription-based streaming service that allows our members to watch TV shows and movies without commercials on an internet-connected device. You can also download TV shows and movies to your iOS, Android, or Windows 10 device and watch without an internet connection.

So I am going to cluster the motion pictures or TV suggests on Netflix and construct a film recommendation system for users. To begin with, I am going to lookup handy information from Netflix to get insights to know statistics of the data with the aid of different feature wise

3. Data Descriptions:

For this project 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. It will be interesting to explore what all other insights can be obtained from the same dataset. In the given datasets, 7787 records and 12 features are available.

Attribute Information:

- **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 on which film is added on Netflix.
- **Release year**: Actual Release Year of the movie / show.
- **Rating**: TV Rating of the movie / show.
- **Duration**: Total Duration - in minutes or number of seasons.
- **Listed_in:** Type of the movie or genres.
- **Description**: The Summary description.

4. Steps Involved

- **Problem Statement in ML Terms:**
  First I define the given problem statement in ML terms which tells us exactly what needs to be done with our dataset.

- **Data Overview:**
  After I have our data, simply take a look at it and perform sanity and null value checks. Also look for statistics of the overall data.

- **Exploratory Data Analysis:** In this I got different insights into the dataset and its correlation with the other features, also its distribution. From this analysis, important features for the clustering are decided.

- **Text Pre-processing:**
  A further step in this process involves text pre-processing it make the raw data ready for the machine learning model, basic objective is to clean the noise which is less relevant to the final outcome, it involve the following steps:
  - **Removing the stopwords**, short words (less than 3 letter): Stop words are those words in natural language that have very little meaning, such as "is", "an", "the", etc. To remove stop words from a sentence, divide your text into

words and then remove the word if it exists in the list of stop words provided by NLTK.

- **Lemmatization** is the process of converting a word to its base form. Lemmatization considers the context and converts the word to its meaningful base form. Lemmatization is that it is more accurate than stemming.
- **Label Encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated.
- **StandardScaler** removes the mean and scales each feature/variable to unit variance. Useful for the features that follow a Normal distribution.
- **Vectorization (TFIDF)** Vectorization is a step in feature extraction, the idea is to get some distinct features out of the text for the model to train on, by converting text to numerical vectors
- **PCA** utilized due to the fact it can assist us enhance overall performance at a very low price of mannequin accuracy. Other benefits of PCA consist of reduction of noise in the data, function choice (to a positive extent), and the potential to produce independent, uncorrelated facets of the data.

- **Topic Modelling/Clustering:** Clustering models allow you to categorize records into a certain number of clusters. This can help you identify natural groups in your data. I clustered the films data based on the description, genres of the film data and films other detail feature based (type, release year, director, cast, country, duration, and rating) because most of the audience choose the film based on the description and type (genres) of the movie or either cast of the film or availability of the time. So I have grouped them into two parts to cluster the data for a better recommendation.

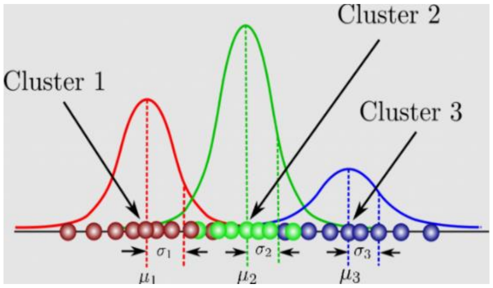
  Models used for clustering:
  - Gaussian Mixture Model (GMMs)
  - K-Means Clustering Model (KMC)
  - Latent Dirichlet Allocation Model (LDA)
  - Affinity Propagation Model (APM)
  - Agglomerative Clustering

- **Comparing Model Performance Metrics and Finding Number of Clusters:** After fitting all the models, I compared their metrics with each other to figure out the best number of cluster. In clustering used silhouettes score, coherence, AIC, Elbow method to obtain the best number of cluster for given data set.

5. Algorithms

1. Gaussian Mixture Model:

Gaussian mixture models (GMMs) is used to cluster the data into different categories based on the probability distribution. Gaussian distributions are assumed for each group and they have means and covariance's which define their parameters. GMM consists of two parts – mean vectors (μ) & covariance matrices (Σ). A Gaussian distribution is defined as a continuous probability distribution that takes on a bell-shaped curve.



Suppose there are K clusters (For the sake of simplicity here it is assumed that the number of clusters is known and it is K). So mu and Sigma is also estimated for each k. Had it been only one distribution, they would have been estimated by the maximum-likelihood method. But since there are K such clusters and the probability density is defined as a linear function of densities of all these K distributions, i.e.

$$p(X) = \sum_{k=1}^{K} \pi_k G(X|\mu_k, \Sigma_k)$$

2. K-Means Clustering Model:

The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, I will use the euclidean distance as measurement.

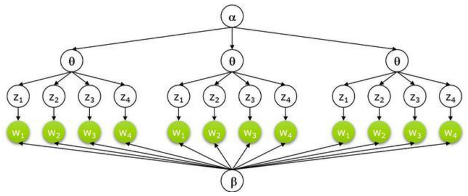
The algorithm works as follows:

1. First, initialize k points, called means or cluster centroids, randomly.
2. Categorize each item to its closest mean and update the mean's coordinates, which are the averages of the items categorized in that cluster so far.
3. Repeat the process for a given number of iterations and at the end, I have our clusters.

The "points" mentioned above are called means because they are the mean values of the items categorized in them. To initialize these means, I have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set.

3. Latent Dirichlet Allocation Model:

The aim behind the LDA to find topics that the document belongs to, on the basis of words contains in it. It assumes that documents with similar topics will use a similar group of words. This enables the documents to map the probability distribution over latent topics and topics are probability distribution.



Let's suppose we have D documents using the vocabulary of V-word types. Each document consists of an N-words token (can be removed or padded). Now, I assume K topics, this required a K-dimensional vector that represents the topic distribution for the document. Each topic has a V-dimensional multinomial beta k over words with a common symmetric prior. For each topic 1…k:

Draw a multinomial over words $\varphi \sim Dir(\beta)$.

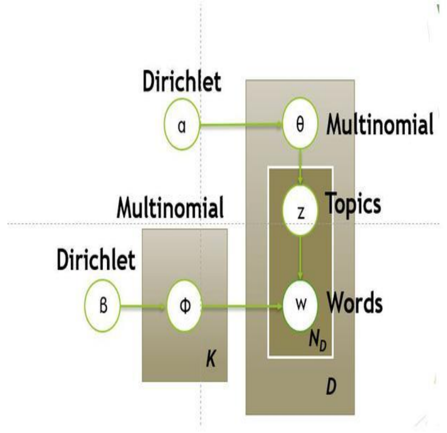
For each document 1…d:

Draw a multinomial over topics $\theta \sim Dir(\alpha)$

For each word w {Nd} :

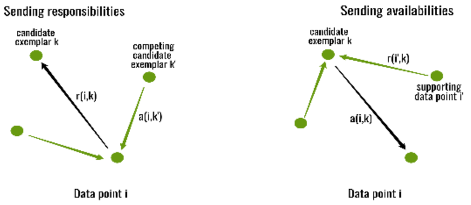
Draw a topic $Z_{N_d} \sim Mult(\theta_D)$ with $Z_{N_d} \epsilon [1..K]$

Draw a word w {Nd}\sim Mult (\varphi).



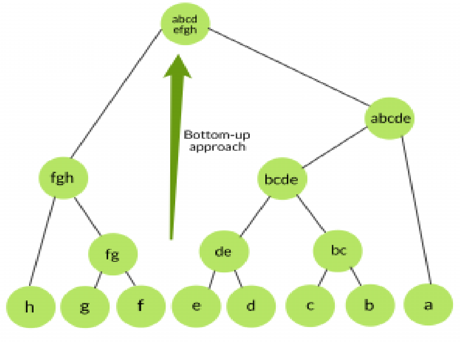
4. Affinity Propagation Model:

Affinity Propagation creates clusters by sending messages between data points until convergence. Unlike clustering algorithms such as k-means or k-mean, affinity propagation does not require the number of clusters to be determined or estimated before running the algorithm, for this purpose the two important parameters are the preference, which controls how many exemplars (or prototypes) are used, and the damping factor which damps the responsibility and availability of messages to avoid numerical oscillations when updating these messages. A dataset is described using a small number of exemplars, 'exemplars' are members of the input set that are representative of clusters. The messages sent between pairs represent the suitability for one sample to be the exemplar of the other, which is updated in response to the values from other pairs. This updating happens iteratively until convergence, at that point the final exemplars are chosen, and hence I obtain the final clustering.



5. Agglomerative Clustering:

This model is also known as 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 agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

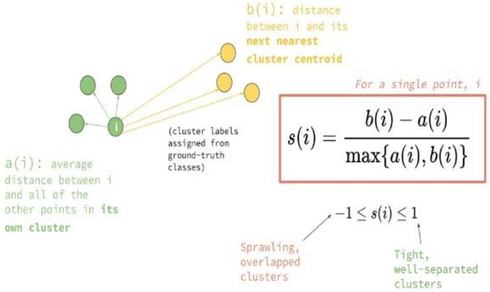


Hierarchical agglomerative clustering

6. Model performance:

Model can be evaluated by various metrics such as:

1. **Silhouettes Score:**
Silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.


b(i): distance between i and its next nearest cluster centroid
i
For a single point, i
$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$
(cluster labels assigned from ground-truth classes)
a(i): average distance between i and all of the other points in its own cluster
$-1 \leq s(i) \leq 1$
Sprawling, overlapped clusters
Tight, well-separated clusters

1: Means clusters are well apart from each other and clearly distinguished.

0: Means clusters are indifferent, or I can say that the distance between clusters is not significant.

-1: Means clusters are assigned in the wrong way.

2. **Coherence:**
Coherence score in topic modelling to measure how interpretable the topics are to humans. In this case, topics are represented as the top N words with the highest probability of belonging to that particular topic. Briefly, the coherence score measures how similar these words are to each other.

3. **Akaike Information Criterion(AIC):**
Akaike information criterion (AIC) is an estimator of prediction error and thereby relative quality of statistical models for a given set of data.

AIC = -2/N * LL + 2 * k/N

Where,
N is the number of examples in the training dataset
LL is the log-likelihood of the model on the training dataset
k is the number of parameters in the model.

The score, as defined above, is minimized, e.g. the model with the lowest AIC is selected.

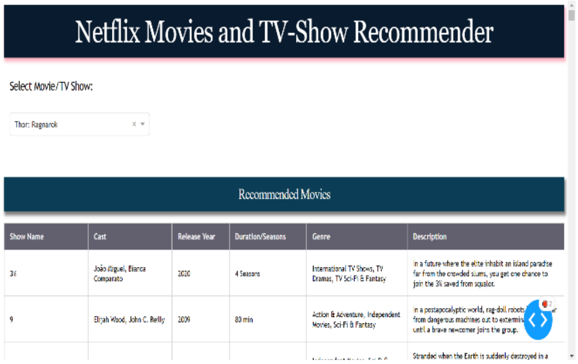
7. Tools/Libraries Used:

The total challenge was once executed the usage of python, in Google Collaboratory. Following libraries had been used for inspecting the facts and visualizing it and to construct the mannequin to Netflix films clustering.

- Pandas: it allows you to perform processing, wrangling and munging of data.
- NumPy: NumPy is a Python library used for working with arrays and mathematical functions.
- Matplotlib and Seaborn: this libraries are used for data visualization.
- Warnings: used to filtering and ignoring the warnings.
- Sklearn: sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction.
- NLTK: The Natural Language Toolkit (NLTK) is a platform used for building python programs that work with human language data for natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.
- Yellowbrick: used to visualize the frequency of the word in dataset also used to visualize the silhouette score and elbow method.
- Gensim: is used to process the text basically we used to import LDA model and coherence scorer.
- PyLDAvis: used to visualize the LDA models cluster.
- Dash: is used to create a recommendation app for films.

8. Dash App:

In this recommendation system I have used a Natural Language Processing (NLP) model and a K-Means Clustering model to make these recommendations. These models grouped the given data on the text based i.e. description and genres. I have created a Dash web app that utilizes used model in colab to recommendations the films based on a user's preferences.



Netflix Movies and TV-Show Recommender

Select Movie/TV Show:

Thor: Ragnarok

Recommended Movies

Show Name	Cast	Release Year	Duration/Seasons	Genre	Description
3%	João Miguel, Bianca Comparato	2020	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi & Fantasy	In a future where the elite inhabit an island paradise far from the crowded slums, you get one chance to join the 3% saved from squalor.
9	Elijah Wood, John C. Reilly	2009	80 min	Action & Adventure, Independent Movies, Sci-Fi & Fantasy	In a postapocalyptic world, rag-doll robots [illegible] from dangerous machines out to extermin[illegible] until a brave newcomer joins the group.
				Independent Movies, Sci-Fi &	Stranded when the Earth is suddenly destroyed in a

9. Conclusion

In exploratory data analysis, I found that:

- There are almost 70% of movies and 30% of TV shows listed on Netflix.
- For the mature audience (MA), there is much more movie content than TV shows. For the younger audience (under the age of 17), it is similar, though there are more movies than TV shows.
- Overall, there is much more content that comes from the United States (52 %) and India (18 %).

- The majority of films are released during the holiday season, i.e., January, October, November, and December, and there has been consistent growth since 2014.
- In the case of genres, international movies take a peak and are followed by dramas and comedies.
- Most of the movies listed on Netflix are about 90 minutes long, which seems to make sense, whereas most TV shows have one season.
- "Jan Suter" directed most of the movies. As stated previously regarding the top genres, it's no surprise that the most popular directors on Netflix with the most titles are mainly international as well. Anupam Kher is cast in most of the films.

In clustering, I clustered the data on the basis of the two types because most of the audience chose the film based on the description and type (genres) of the movie or either the cast of the film or the availability of the time. So I have grouped the features into two parts to cluster the data for a better recommendation.

- It seems that the K-mean cluster's silhouette score gives the highest score For 25 topics, the cluster and elbow methods yield a maximum of 15 topics.
- Gaussian mixture models-AIC score seems to hint at the 25 topics.
- If we rely on the LDA-coherence score, k = 12 is the highest.
- As I clustered the categorical data using the Affinity Propagation Model and Agglomerative Model, they didn't perform well and gave the worst clustering result.

As a result, we will simulate data from 25 latent/hidden topics while accounting for all scores.

References-

1. https://www.geeksforgeeks.org/
2. https://machinelearningmastery.com
3. https://www.analyticsvidhya.com/
4. https://www.freecodecamp.org/
5. https://www.analyticsvidhya.com/
6. https://www.w3schools.com/
7. https://neptune.ai/