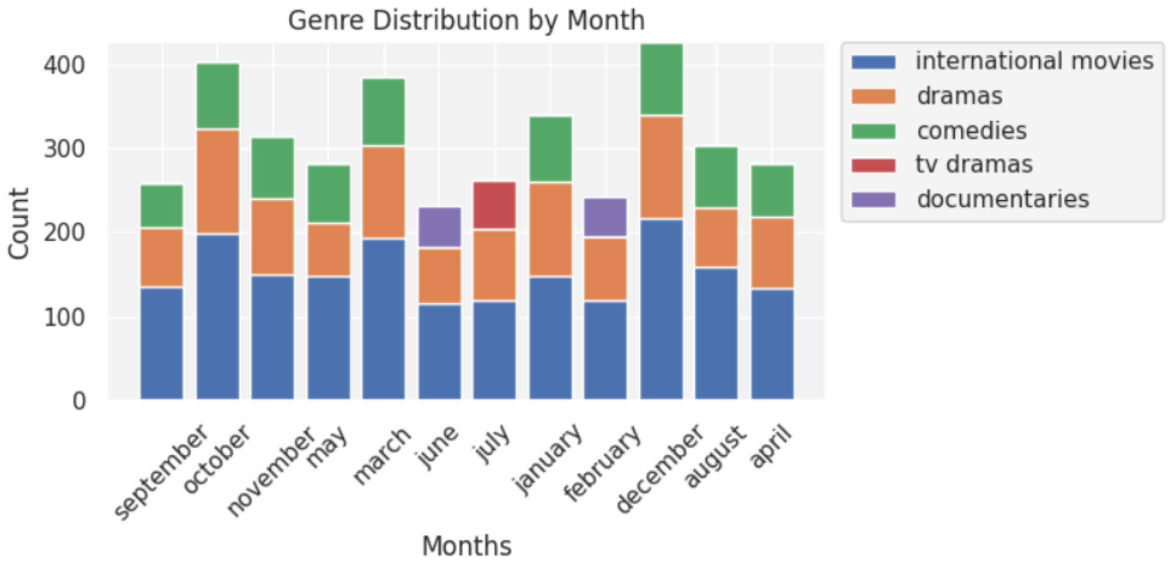
CSE587: Data Intensive Computing
Project - Phase 2 - Report

Name	UB-ID	UB Person No.	UB Email
Rachna Bhatia	rachnatr	50486853	rachnatr@buffalo.edu
Rounak Biswas	rounakbi	50450467	rounakbi@buffalo.edu
Shruti Gupta	sgupta55	50485160	sgupta55@buffalo.edu
Sujan Reddy	sujanred	50471029	sujanred@buffalo.edu

Exploratory Data Analysis (EDA)

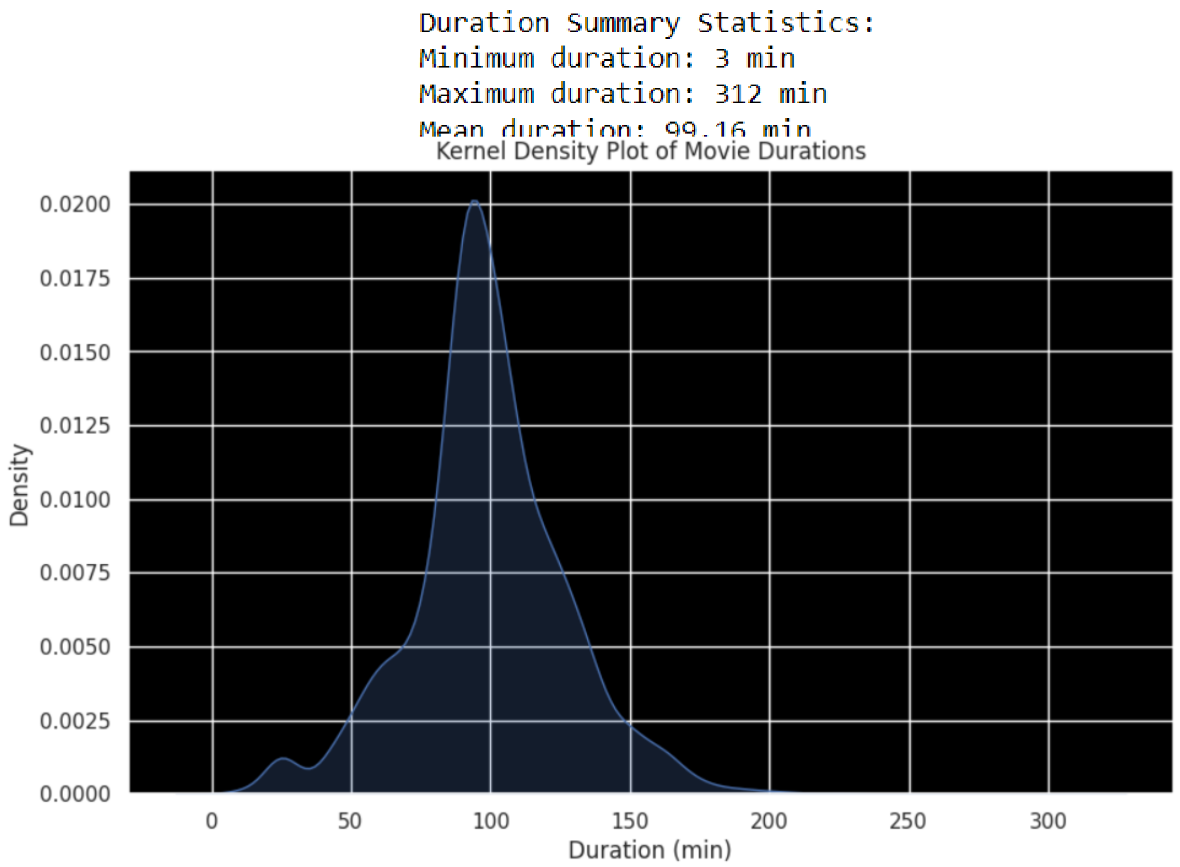
In EDA, We evaluated and summarised data sets to acquire insights, identify patterns, detect anomalies, and formulate hypotheses. It helped us in understanding the structure and characteristics of the data before applying more complex statistical or machine learning algorithms.

Genre Distribution By Month - Finding out the relation among months and the type or genre of content most released during that month.
For this section of analysis, we looked into two columns of the dataset, Date_added and Listed_in and found out the three most common types of content released.



- International movies, comedies, and dramas are popular worldwide due to their appealing narratives and relatable characters.
- Distributors and streaming services actively seek out and distribute foreign films to meet the high demand.
- International film festivals serve as platforms to showcase and promote foreign films, generating buzz and attracting audiences.
- Recognition at prestigious award shows like the Oscars contributes to the growing popularity of foreign films.
- Streaming platforms provide easy access to foreign content, allowing viewers to explore different cultures and narratives.
- The availability of on-demand content has increased the demand for frequent releases throughout the year.

Movie Duration Analysis - For this section of the analysis, we wanted to deduce the 'duration' of the movies that are or were released, recorded in our dataset. For the plot we have density as the number of movies plotted against duration(In mins) of the movies.

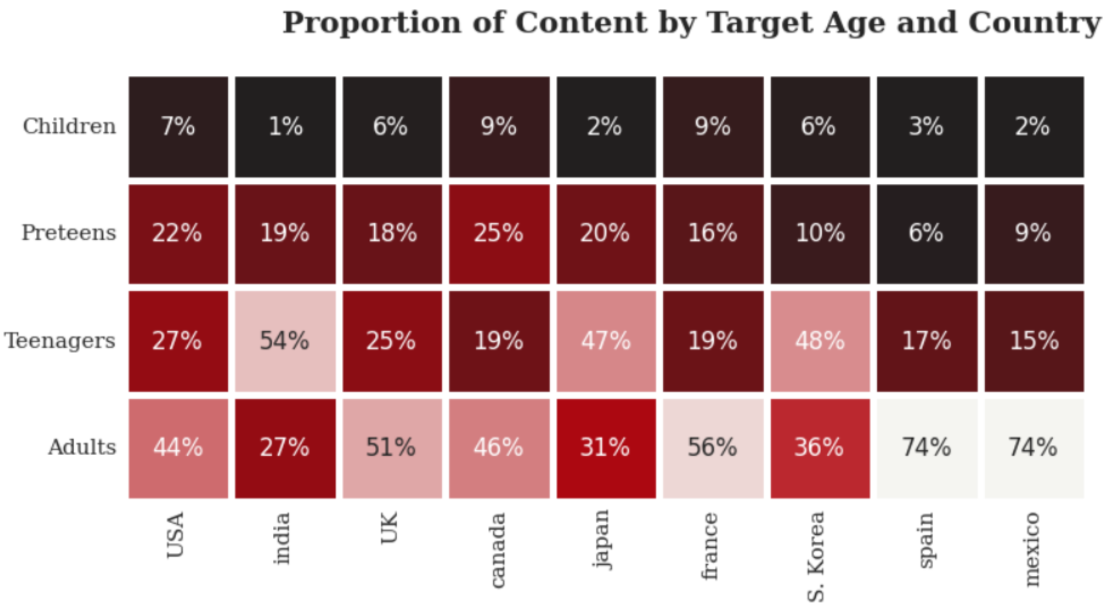


```
Duration Summary Statistics:
Minimum duration: 3 min
Maximum duration: 312 min
Mean duration: 99.16 min
```

We can deduce that 90-minute movie durations are released more than others, Our take on that is :

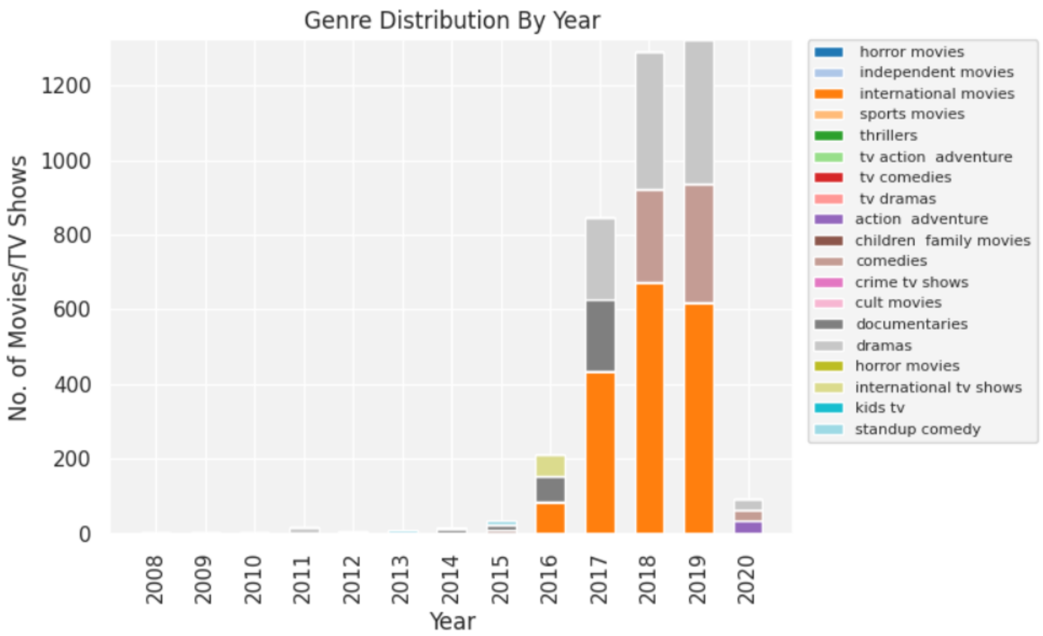
- It aligns with the scheduling needs of movie theatres, allowing for multiple screenings and maximising ticket sales.
- It caters to the average attention span of viewers, ensuring they remain engaged throughout the film.
- A shorter duration encourages focused storytelling, emphasising important plot points and character development while reducing unnecessary content.

Target Ages - For this section, we strive to understand the local audience's preferences and create content accordingly, taking into account the specific target age groups within each country.



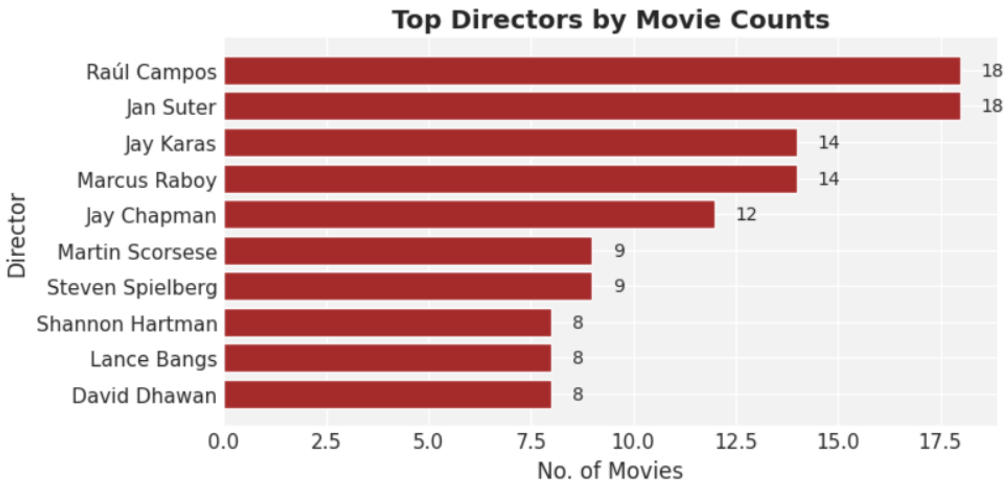
- Based on target age proportions, the graph demonstrates the variability of content choices between cultures.
- The US and Canada show similar target age distributions, indicating shared content preferences.
- India stands out with a majority of its content targeting teens at 54%. This suggests a strong emphasis on content catering to the younger demographic in the country.
- Spain and Mexico stand out with a larger proportion of content targeted towards adults, possibly indicating a preference for mature themes and storytelling in these countries.
- Understanding these differences can help content creators and streaming platforms in curating and promoting content that appeals to certain target groups in each country.

Type of content released each year - Here, we find out the underlying trends of the type of content that is released each year and try and understand what genre is released the most and why that is.



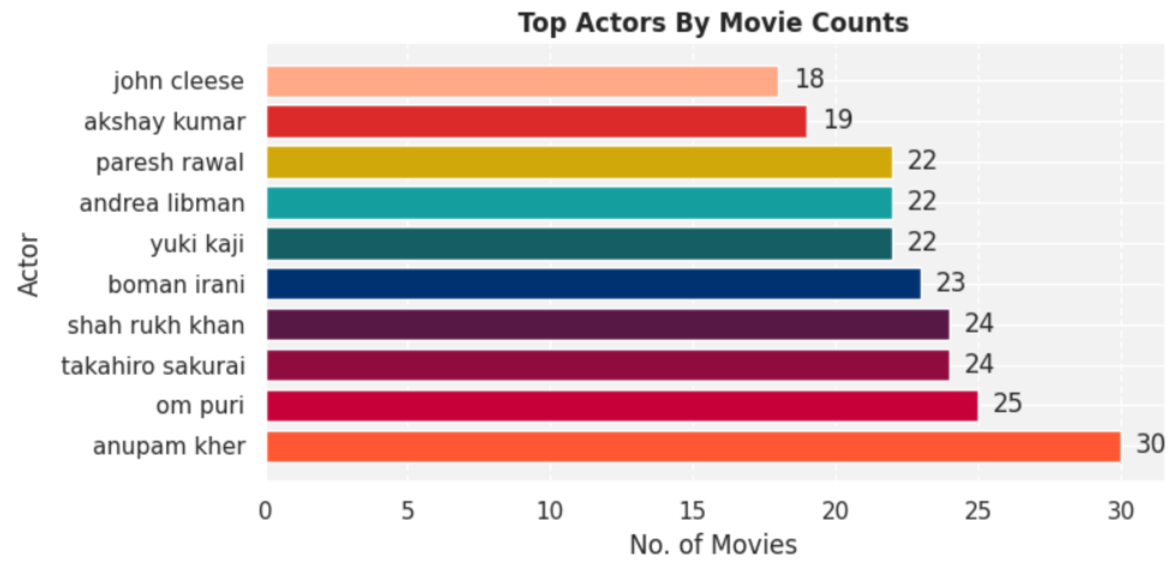
- In various years, International movies have continuously ranked among the top three most popular categories, suggesting continuing popularity.
- Dramas have maintained a steady place in the top three categories, suggesting long-term interest and viewership.
- Comedy is consistently ranked among the top genres, demonstrating its enduring popularity and need for lighthearted entertainment.
- Documentaries, family-friendly films, stand-up comedy, TV shows, and action-adventure films are among the genres represented in the data, catering to a wide spectrum of audience interests.
- Specific genres' popularity may fluctuate from year to year, reflecting changes in audience tastes and industry trends.

Top Directors By Movie Counts - Here, the plot shows the total counts of released contents directed by the top 10 directors.



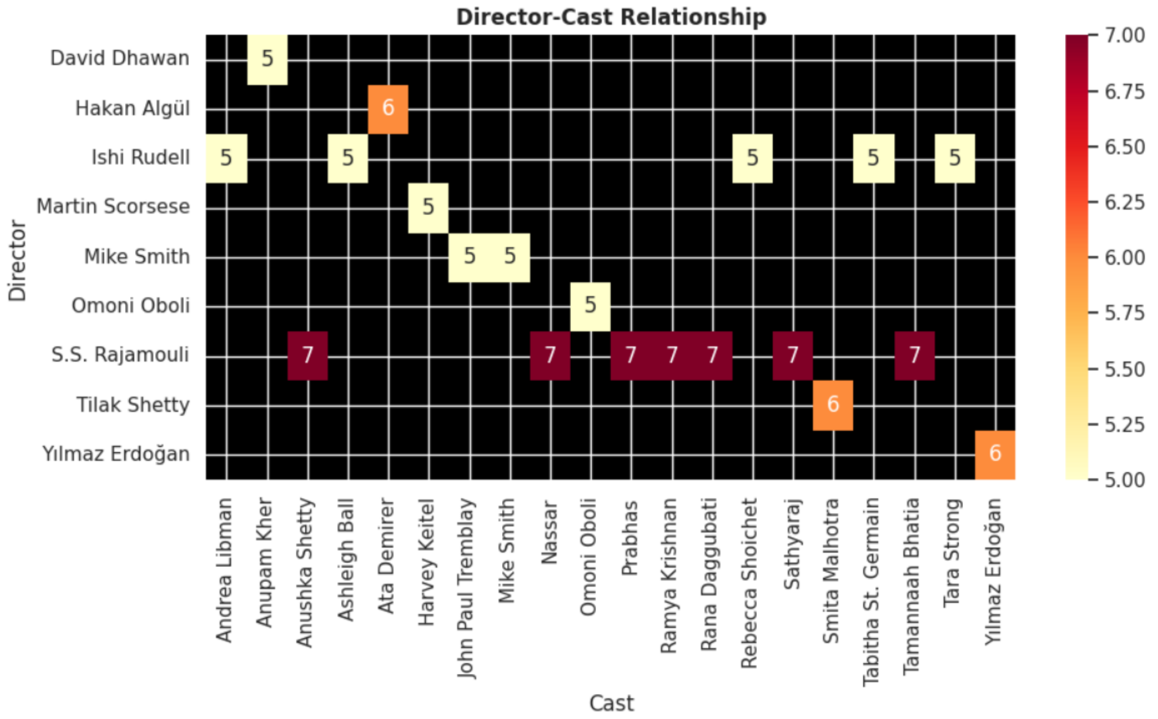
- Raul Campos and Jan Suter have the most films to their credit, both with 18 films.
- Jay Karas and Marcus Raboy have each directed 14 films since them.
- Jay Chapman comes in second with 12 films, while renowned directors Martin Scorsese and Steven Spielberg each have 9 films.
- Shannon Hartman, Lance Bangs, and David Dhawan round out the top ten filmmakers with 8 films each.
- This information will allow us to analyse the director column and potentially explore relationships between the directors and other attributes, such as the cast, in our EDA.

Frequently appearing Actors/Actresses - The plot shows the total number of releases performed by the top ten actors/actresses.



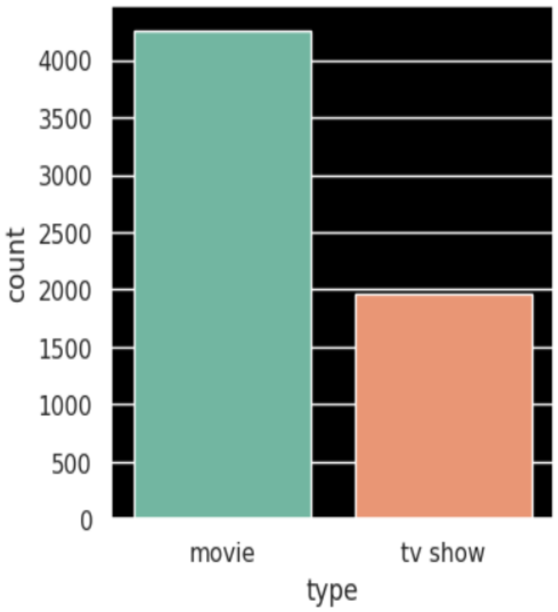
- Anupam Kher has the most films to his credit, with 30.
- Om Puri comes in second with 25 films, while Takahiro Sakurai and Shah Rukh Khan both have 24.
- Boman Irani, Yuki Kaji, and Andrea Libman have each appeared in 23 and 22 films.
- Paresh Rawal and Akshay Kumar each have 22 films, while John Cleese rounds out the top ten with 18 films.
- This data will allow us to analyse the cast column and maybe investigate links between the actors/actresses and other variables in EDA.

Director and Cast Analysis - Here we have deduced a relation between directors and actors and it reveals instances of strong collaborations between directors and actors, with some actors constantly appearing in the content of specific directors.



- S.S. Rajamouli has continuously collaborated with a cast that includes Prabhas, Rana Daggubati, Anushka Shetty, Tamannaah Bhatia, Sathyaraj, Nassar, and Ramya Krishnan, with each actor starring in seven of Rajamouli's films.
- In five productions, Ishi Rudell has worked with performers Ashleigh Ball, Andrea Libman, Rebecca Shoichet, Tabitha St. Germain, and Tara Strong, demonstrating a recurring casting choice.
- Some directors, such as Ylmaz Erdoan and Omoni Oboli, have also performed as the main character in their films, implying personal engagement and creative control.
- While certain director-actor pairings are common, others, such as Martin Scorsese and Harvey Keitel or Tilak Shetty and Smita Malhotra, have only collaborated on a few films, showing both ongoing connections and unpredictable collaborations.
- These findings highlight the dynamics and preferences within the film industry regarding casting choices and collaborative relationships between directors and actors.

Analysis of Movies vs TV Shows - The plot below contains the count of movies and TV shows.

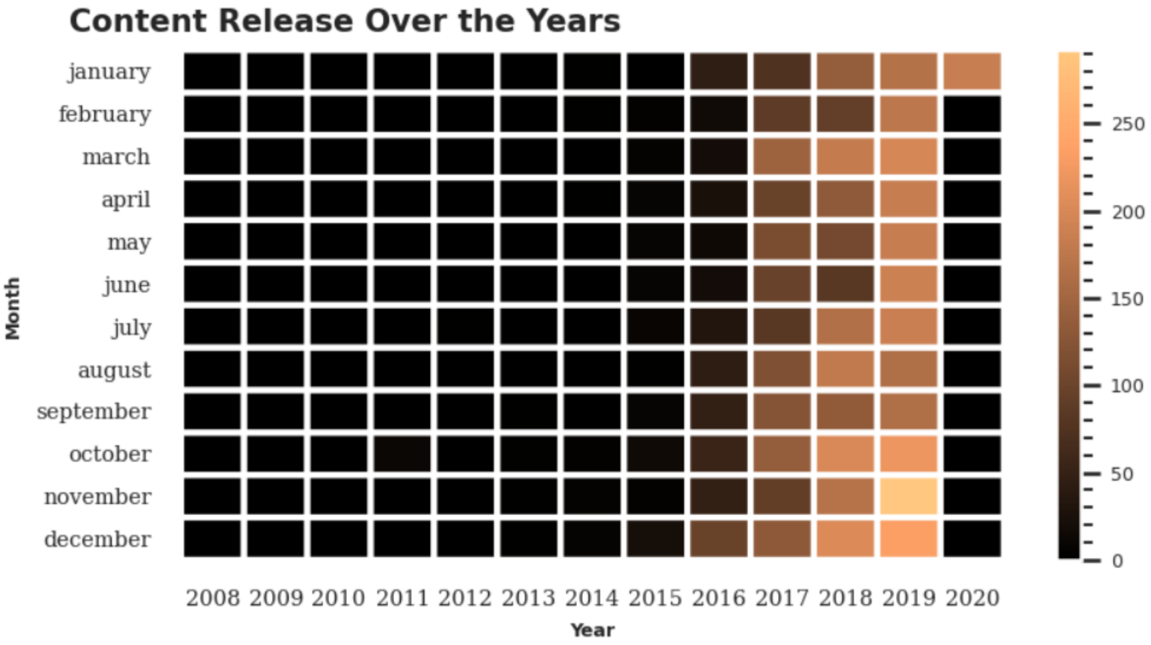


From the plot we can deduce that the number of movies released is more than double the number of TV shows. The reason for that can be :

- TV shows can run for several seasons whereas movies are a short form of entertainment, being released a lot more in the period of a running TV show which can go on for years.
- Hence, the number of new movies released exceeds the number of TV-shows released by a huge margin.

New Content Addition - For director to see which month will give him the success for a new release

F



From the above plot,

- The x-axis represents the years, showing the timeline from the earliest available year to the latest.
  While,
  The y-axis represents the months, displaying the twelve months of the year in reverse order.
- Each of the cells in the heatmap represents the count of content releases in a specific month and year.
- The heatmap provides an overview of how the content releases are distributed across different months and years.
- When we analyse the graph we can deduce the release pattern of content on Netflix.

Collectively this helps us understand the streaming platform's content strategy and forecasting annual audience trends.

Use case -

- Directors can use this statistic and release their contents accordingly, taking an example of the year 2018. If the Director wishes to produce their content, be it TV show or drama, during the month of June or February it may end up doing well unlike in the months of October or December, whereas the opposite might also be true, where when fewer contents are released, audience and no option but to view that content and that in turn may boost their show ratings.

ML Algorithms And Their Analysis

1. K Nearest Neighbours

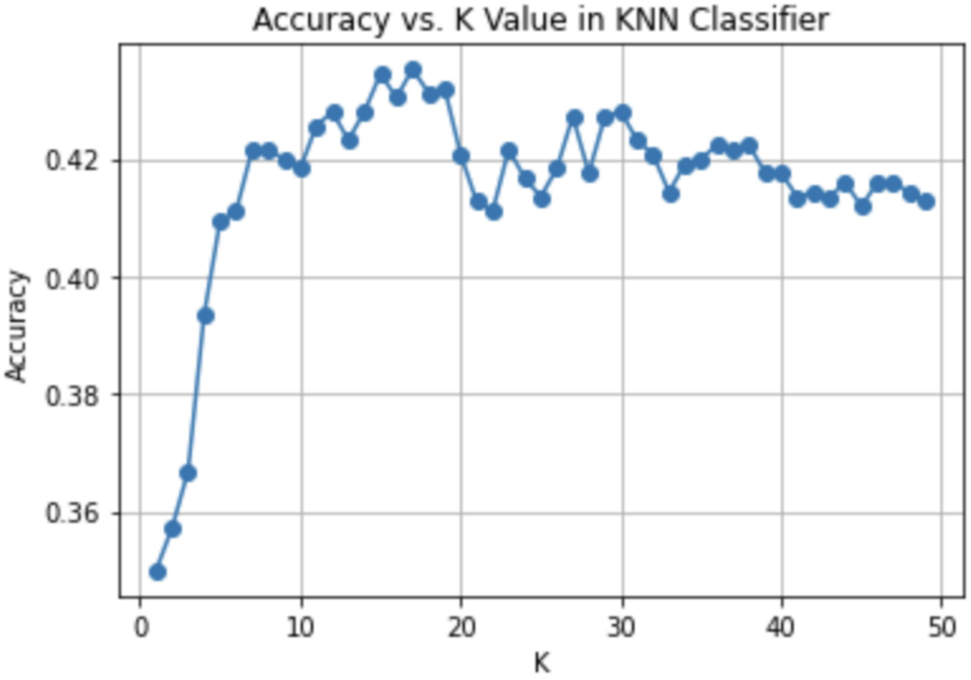
KNN algorithm is a very popular algorithm and our problem statement involves analysing a comprehensive dataset of Netflix movies and TV shows to develop a Content-Based Recommender System. It is popular in terms of prediction and we require the K-Nearest neighbours for the specific input features and this was the reason we have chosen this algorithm

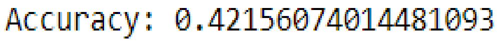
Algorithm -

The machine learning technique K-nearest neighbours (KNN) is used for classification and regression problems. It locates the k data points that are the closest to a new point, and then it chooses a label or value by majority vote or average.

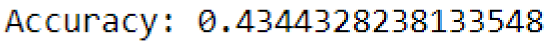
In order to create predictions, the algorithm calculates distances between points and saves training data. Selecting the right k is essential.
For big datasets, KNN may be computationally costly and requires feature scaling. It works well for both significant feature correlations and local patterns.

Explanation -



For k = 7 

```
Accuracy: 0.42156074014481093
```

For k = 15 

```
Accuracy: 0.4344328238133548
```

In this section, Netflix content ratings are predicted using the k-nearest neighbours (KNN) algorithm based on the director and cast information.
As the selected features - cast and director, and the target feature - rating is textual data, we can not directly apply KNN model to it.

Hence, we utilised TFIDF to transform the textual data (information about the director and cast) into a numerical representation that can be applied to machine learning methods like KNN.

We also plotted the Accuracy vs K value graph to find optimal value of k and to visualise how accuracy is changing on the basis of the number of neighbours considered. We also found the accuracy for k = 7 and k = 15 as 0.42 and 0.43 respectively. The reason for low accuracy could be the skewed distribution of ratings for some the ratings, the noise of the data could also be one of the contributing factors.

We can conclude that from the above accuracy, this is not the final output we are expecting and we need to come up with the better Machine learning algorithms,and we can deduce that the KNN assumes that all characteristics contribute equally to the prediction, sensitivity to the choice of k, and poorer performance with bigger datasets,and hence this may not be the optimal algorithm and we need to look up with better machine learning algorithms

2. Random Forest Algorithm

As we have seen from the above algorithm, it is not yielding the required accuracy and hence we have chosen to see the results of the Random forest algorithm. The genre (listed_in), actors, directors, release year, and user ratings are just a few of the characteristics of Netflix material that were analysed using the Random Forest algorithm. When creating or acquiring new content, Netflix may use this information to spot patterns, assess viewer preferences, and make data-driven choices.

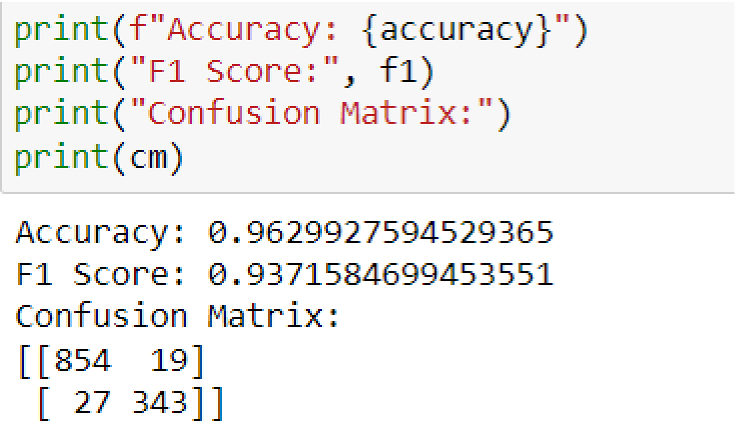
Algorithm -

An ensemble learning system called Random Forest combines many decision trees to provide predictions. By training each tree on various subsets of the data and choosing features at random for splitting, it builds a forest of trees.

When making predictions, it either uses majority voting (for classification) or averaging (for regression) to combine the predictions of all trees.

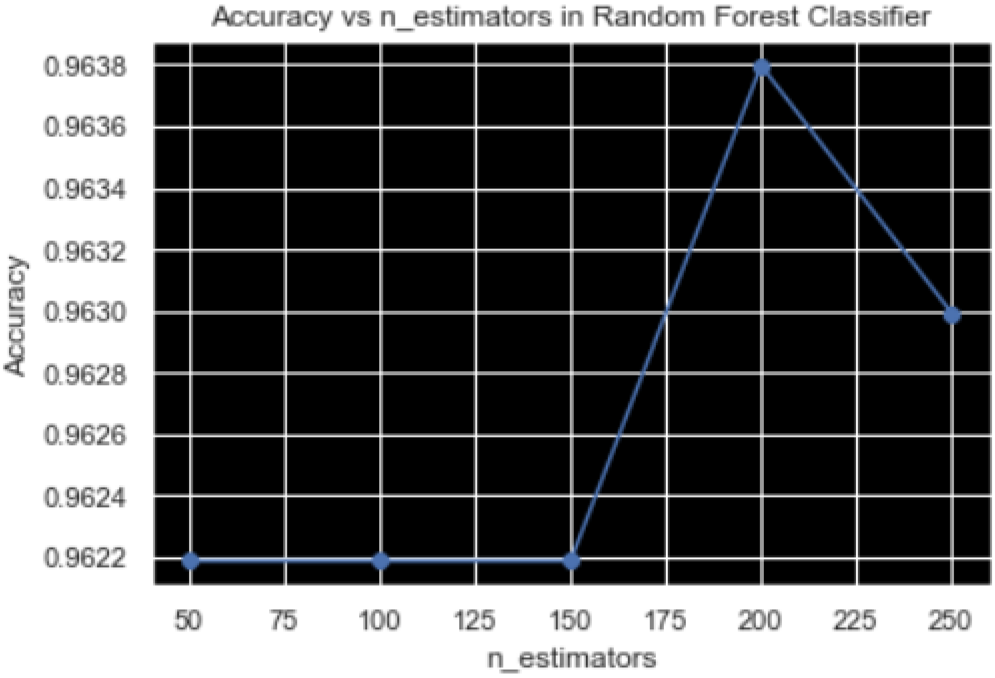
Overfitting is lessened by Random Forest, which also handles big feature sets and offers estimates of feature relevance. It is a flexible and strong algorithm that is renowned for its precise forecasts and insightful data.

Explanation -



```
print(f"Accuracy: {accuracy}")
print("F1 Score:", f1)
print("Confusion Matrix:")
print(cm)
```

```
Accuracy: 0.9629927594529365
F1 Score: 0.9371584699453551
Confusion Matrix:
[[854  19]
 [ 27 343]]
```



For applying Random Forest to our dataset, we have taken the director, cast, and release year as the input features and have used the Label Encoder to encode categorical labels into numerical values. From the above usage of the Ml algorithm, we can see that when we consider the change in the n_estimator values, there is a change in the Accuracy.

We can see that the highest accuracy can be seen when we have the n_estimator value at 200 which yields an accuracy of 0.9638 and it gradually drops when there is an increase in the n_estimator, but whereas when we have changed the input labels to the Director and the cast, the accuracy was highest when the n_estimators are 100 and the lowest when the n_estimators are at 150, so change in the different labels and no. of n_estimators yields different accuracy.

Here, the goal was to predict the type (movie or TV show) of content based on specific selected features. These features include the title, director, cast, country, date added and rating of each content. One-hot encoding is applied to the features to convert categorical variables into a numerical representation.
This was done to ensure that the features are in a suitable format for classification. And similarly encode the target variable, type, Label Encoder is utilised. It converts the categorical labels of movies or TV shows into numerical values. We then used the Random Forest classifier for the required classification. An accuracy of 0.96 was observed. In the confusion matrix, F1 score came out to be 0.93.

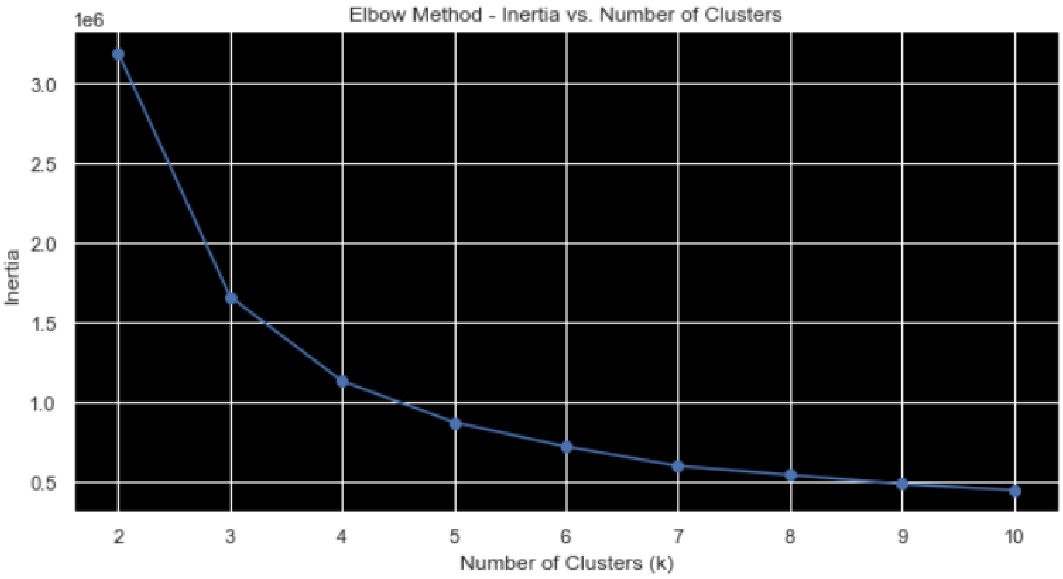
3. K-means clustering

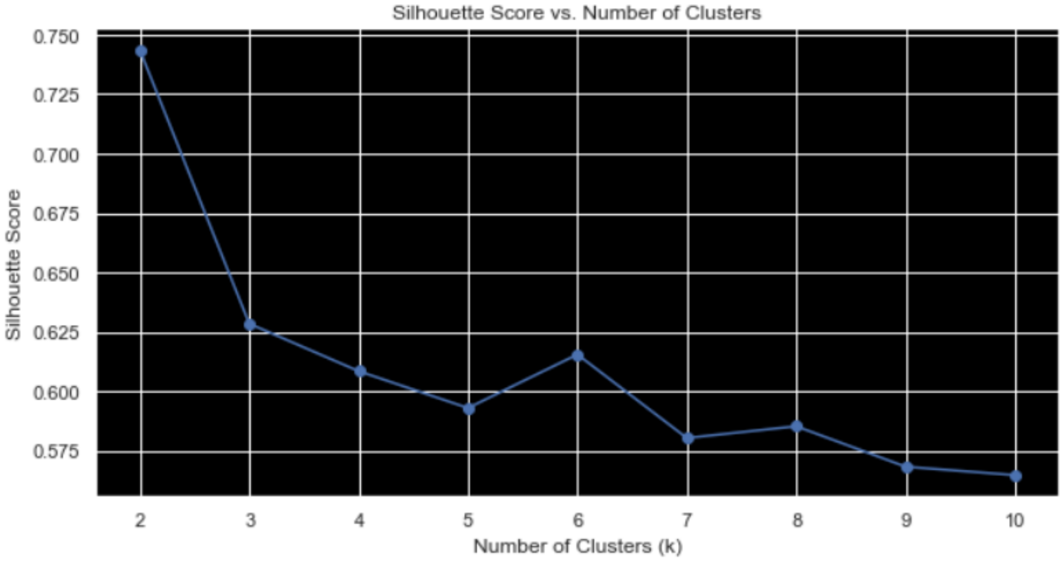
The reason for choosing the K-means clustering is - We wanted to know how the Unsupervised Algorithms may perform on our dataset and how the quality of the data, the feature selection, and the interpretability of the generated clusters affect the results of K-means clustering. To rate the calibre of the clustering outcomes, evaluation measures like silhouette score will be utilised. Additionally, the clusters may be validated and valuable insights can be gained from the Netflix data with the use of visualisations and subject knowledge.

Algorithm -

An unsupervised machine learning approach called K-means clustering is used to cluster comparable data points into groups. It seeks to divide the data into k clusters, where k is a user-selected, pre-defined integer. Iteratively assigning data points to the closest centroid, the method first initialises k cluster centroids at random before recalculating the centroids based on the mean position of the allocated points.

The centroids keep moving until convergence, at which point they stop moving considerably. The goal of K-means clustering is to produce compact and well-separated clusters by reducing the within-cluster sum of squared distances.
It offers insights into the underlying structure of the data and is frequently used for tasks like data exploration, pattern detection, and segmentation.





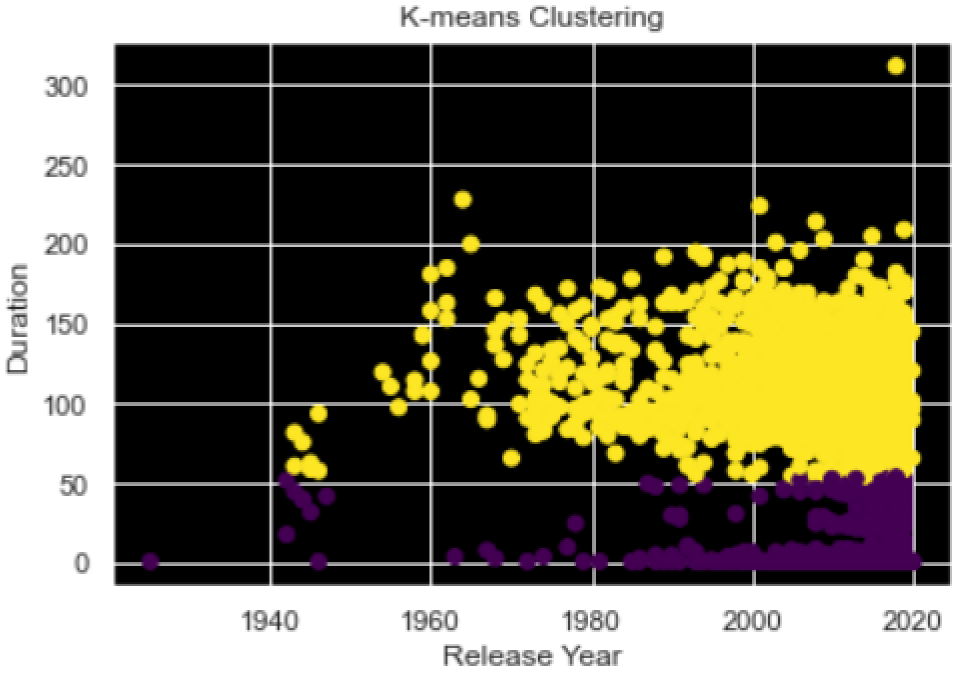
Explanation -

Here, we plotted the elbow curve, to find the optimal number of clusters, but it can be observed that this method of finding the value of k is not appropriate.

We do not get a clear elbow shape, this could be due to many reasons, one of them being the lack of proper distinct clusters or the presence of outliers.

We then plotted silhouette scores for 2 to 10 clusters. The highest score was at 2 clusters - 0.74 and the minimum was at 10 - 0.57, we can assume with an increasing number of clusters, the silhouette score would decrease.

For k = 2, we have well-defined clusters, but as we increase the value of k, the clusters become less and less distinguishable, i.e. it has weak separation.



Explanation -

Now, Considering the silhouette score, for plotting we have considered k = 2. We have plotted a scatter plot to visualise the clusters generated by the K-means clustering algorithm using the features 'release_year' and 'duration' from the Netflix dataset.

The x-axis represents the 'release_year', the y-axis represents the 'duration', and the points are coloured based on the assigned cluster labels using the 'cluster' column.

By visualising the scatter plot, we observe how the data points are grouped into different clusters based on their similarity in terms of release year and duration. Each cluster is represented by a different colour in the plot.

4. Principal Component Analysis (PCA)

PCA may be used to investigate and examine the connections between the numerical features 'release_year' and 'duration' in the context of our dataset. We can more effectively view and comprehend the data by using PCA to minimise the dimensionality of the data while maintaining the most important information. We wanted to know the valuable insights on our data after applying the dimensionality reduction and finding the correlation analysis through PCA.

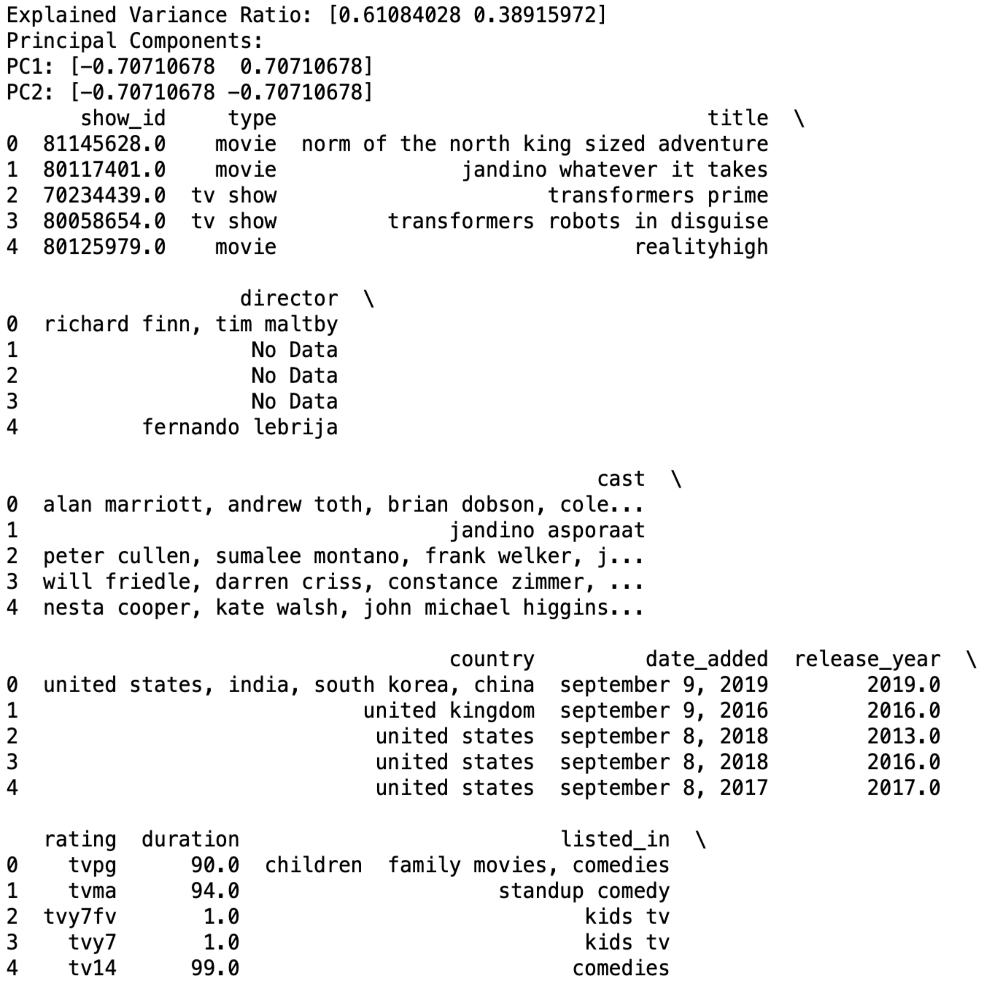
Algorithm -

Principal component analysis, or PCA, is a dimensionality reduction approach that is used to convert highly dimensional data into a lower-dimensional representation while maintaining the data's fundamental properties.

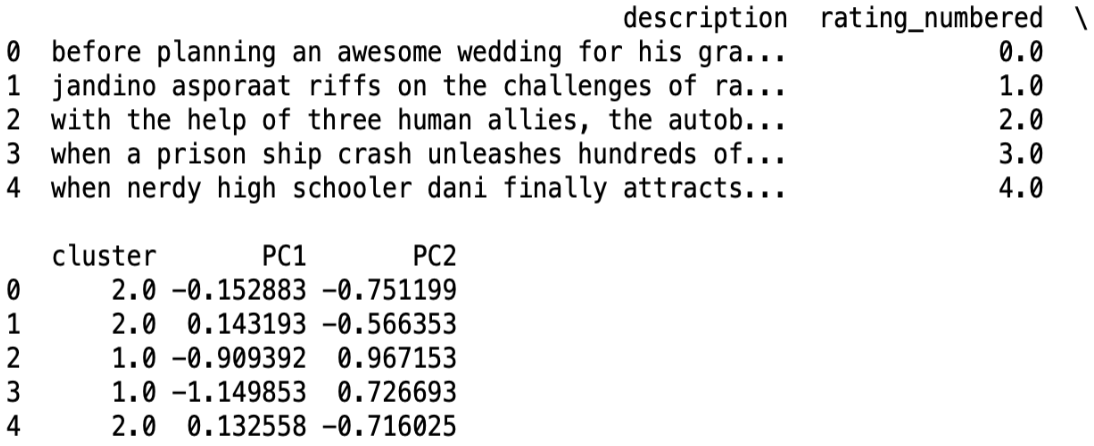
This is accomplished by locating the main components—linear combinations of the initial attributes that encapsulate the data's greatest volatility. The direction with the greatest variation is represented by the first principal component, which is followed by subsequent components in decreasing order.

One may successfully decrease the dimensionality of the data while keeping the majority of the information by choosing a subset of these components. PCA is frequently used for feature extraction, noise reduction, and data visualisation, making it easier to comprehend complicated datasets and to do further analysis and modelling.

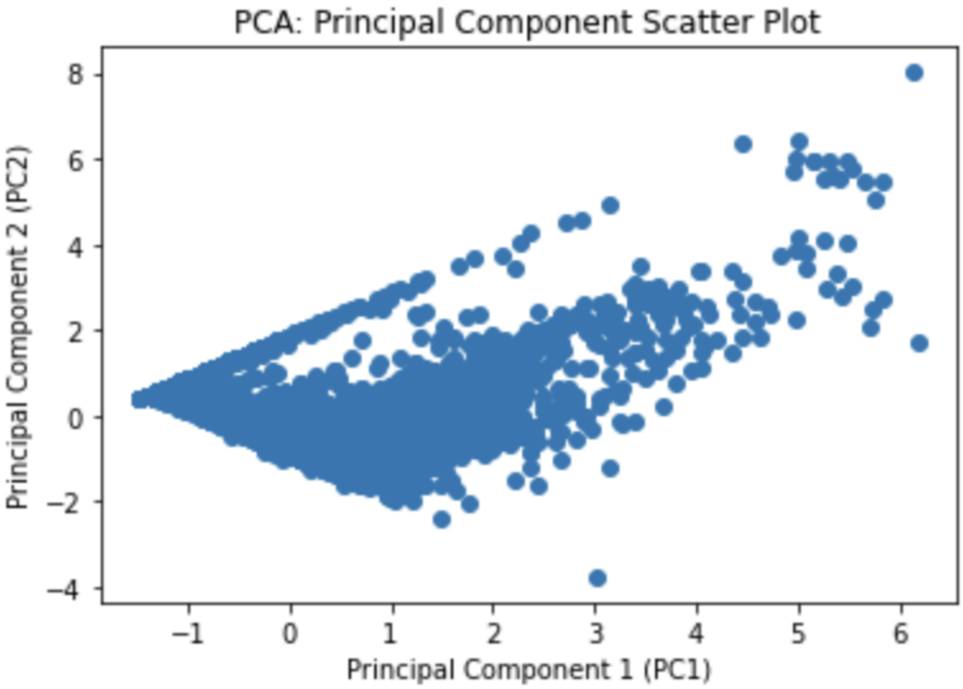
Explanation -



```
Explained Variance Ratio: [0.61084028 0.38915972]
Principal Components:
PC1: [-0.70710678  0.70710678]
PC2: [-0.70710678 -0.70710678]
      show_id     type                                   title  \
0  81145628.0    movie  norm of the north king sized adventure   
1  80117401.0    movie               jandino whatever it takes   
2  70234439.0  tv show                      transformers prime   
3  80058654.0  tv show         transformers robots in disguise   
4  80125979.0    movie                             realityhigh   

                   director  \
0  richard finn, tim maltby   
1                   No Data   
2                   No Data   
3                   No Data   
4          fernando lebrija   

                                               cast  \
0  alan marriott, andrew toth, brian dobson, cole...   
1                                  jandino asporaat   
2  peter cullen, sumalee montano, frank welker, j...   
3  will friedle, darren criss, constance zimmer, ...   
4  nesta cooper, kate walsh, john michael higgins...   

                                    country         date_added  release_year  \
0  united states, india, south korea, china  september 9, 2019        2019.0   
1                            united kingdom  september 9, 2016        2016.0   
2                             united states  september 8, 2018        2013.0   
3                             united states  september 8, 2018        2016.0   
4                             united states  september 8, 2017        2017.0   

   rating  duration                          listed_in  \
0    tvpg      90.0  children  family movies, comedies   
1    tvma      94.0                     standup comedy   
2  tvy7fv       1.0                            kids tv   
3    tvy7       1.0                            kids tv   
4    tv14      99.0                           comedies   
```



```
                                         description  rating_numbered  \
0  before planning an awesome wedding for his gra...              0.0   
1  jandino asporaat riffs on the challenges of ra...              1.0   
2  with the help of three human allies, the autob...              2.0   
3  when a prison ship crash unleashes hundreds of...              3.0   
4  when nerdy high schooler dani finally attracts...              4.0   

   cluster       PC1       PC2  
0      2.0 -0.152883 -0.751199  
1      2.0  0.143193 -0.566353  
2      1.0 -0.909392  0.967153  
3      1.0 -1.149853  0.726693  
4      2.0  0.132558 -0.716025  
```



For applying PCA to our dataset, we selected columns 'release_year' and 'duration' as they have numerical characteristics. Since it is crucial to normalise the features and scale them similarly, StandardScaler is used to normalise the feature data, resulting in each feature having a mean of 0 and a standard deviation of 1.

PCA is instantiated with 'n_components=2' to indicate that we want to extract two main components from the data. The standard feature data ('X_scaled') is then fitted to PCA, to extract the primary components, the PCA's 'components_' property is used. The contribution of each feature to each primary component is shown on the printed output.

The explained variance ratio, which shows the percentage of variation in the data that is explained by each principal component, is one of the code's outputs. Each of the principle components, which are linear combinations of the original characteristics, represent a separate pattern or piece of knowledge from the data using the reduced-dimensional representation offered by PCA.

5. Isolation Forest

The Isolation Forest algorithm is useful for detecting anomalies or outliers in the dataset. The reason for choosing this algorithm for our dataset of Netflix, is that it can help identify unusual or abnormal patterns in the dataset.

For instance, it can detect movies or TV shows with unusual ratings, extreme durations, or content that deviates significantly from the average. These anomalies can indicate potential errors in data or highlight unique and distinct content that may require some addressing.

Algorithm -

It is a method for detecting anomalies that accurately locates abnormalities or outliers in a dataset. It achieves this by creating an isolation forest, a collection of arbitrary decision trees. During construction, the algorithm chooses at random a feature and a split value to recursively partition the data into a tree structure.

In comparison to normal data points, anomalies are anticipated to have shorter average route lengths inside the trees. The method can successfully isolate anomalies and identify them as instances that need fewer divisions to be separated by using the idea of isolation.

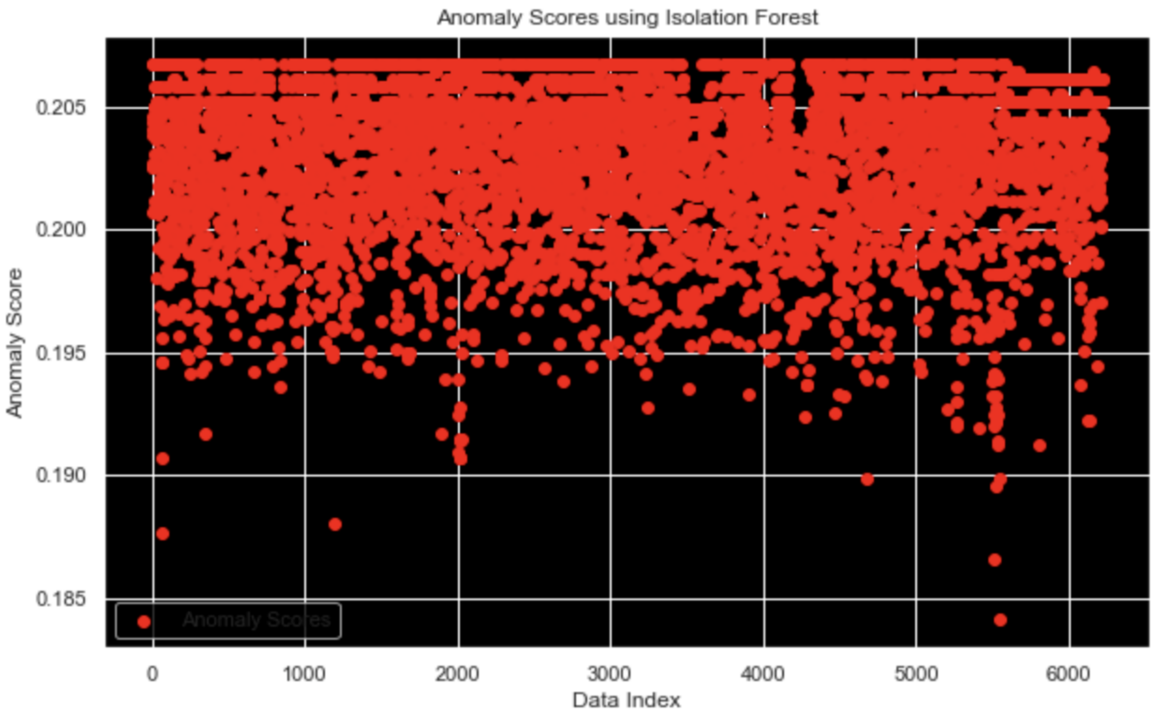
As a scalable, effective technique that can handle high-dimensional data, isolation forest is well suited for finding anomalies in a variety of fields, including fraud detection, network intrusion detection, and outlier analysis.

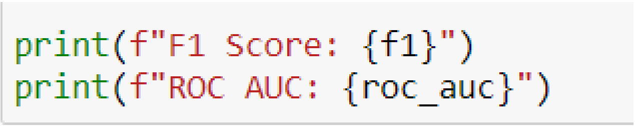
Explanation -

Here, the selected features are columns - 'duration', 'release_year', 'director', 'cast', 'country', 'listed_in', and 'description'. The features (X) and the target variable (y) are extracted from the dataframe. X contains the selected features, and y contains the target variable that is in the 'type' column.

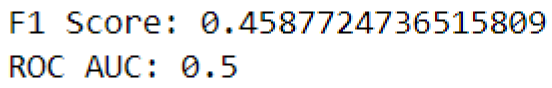
With the train_test_split function, we are specifying a test size of 0.2, 20% of the data will be used for testing, and the remaining 80% will be used for training. Text columns ('director', 'cast', 'country', 'listed_in', 'description') are then processed using TF-IDF vectorization. Numeric columns ('duration' and 'release_year') are extracted from X_train and X_test and converted to a numeric data type using astype(float).

The encoded text features and the numeric features are combined which in turn helped us to combine the feature matrix for both the training and testing sets. Anomalies are predicted on the testing data using the predict() method of the IsolationForest model. The predictions are stored in y_pred_test.





```
print(f"F1 Score: {f1}")
print(f"ROC AUC: {roc_auc}")
```



```
F1 Score: 0.4587724736515809
ROC AUC: 0.5
```

In the plot above -

The x-axis denotes the order or position of the data points in the dataset. Each data point is assigned a unique index starting from 0. The y-axis denotes the degree of abnormality or outlier-ness of a data point. Higher scores suggest a higher likelihood of being an anomaly.

A red dot is drawn on the graph to represent each data point's anomaly score from the dataset. The data index is represented by the horizontal location on the x-axis and the magnitude of the anomaly score by the vertical position of the dot on the y-axis.

By analysing the graph we can deduce the distribution of the anomaly scores across the dataset. Higher anomaly scores, represented by dots that are farther from the x-axis, indicate a higher probability of being an outlier or anomaly.

Thus with help of the plot and the algorithm we can analyse the distribution and identify potential outliers or anomalous data points.

6. One-Class SVM

SVMs are useful for managing high-dimensional data and may be used to classify text, which is relevant to the issue of predicting genres from textual data.we wanted to know the performance of the data and the anomaly detection in our data and hence we have used this algorithm.

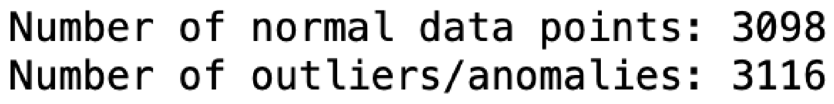
Algorithm -

An anomaly detection machine learning approach is called One-Class SVM (Support Vector Machine). It is intended to learn a decision boundary that includes most of the training data and classifies it as the typical class, while recognizing outliers or anomalies as data points beyond this boundary.

One-Class SVM works in a single-class environment where only normal data is accessible during training, in contrast to classic SVM, which functions in a binary classification scenario. In order to successfully generalise to new data, it builds a hyperplane that optimises the margin around the typical data points.

When anomalies are uncommon or poorly represented in the training data, one-class SVM is useful. It may be used in a variety of areas for fraud detection, network intrusion detection, and outlier identification.

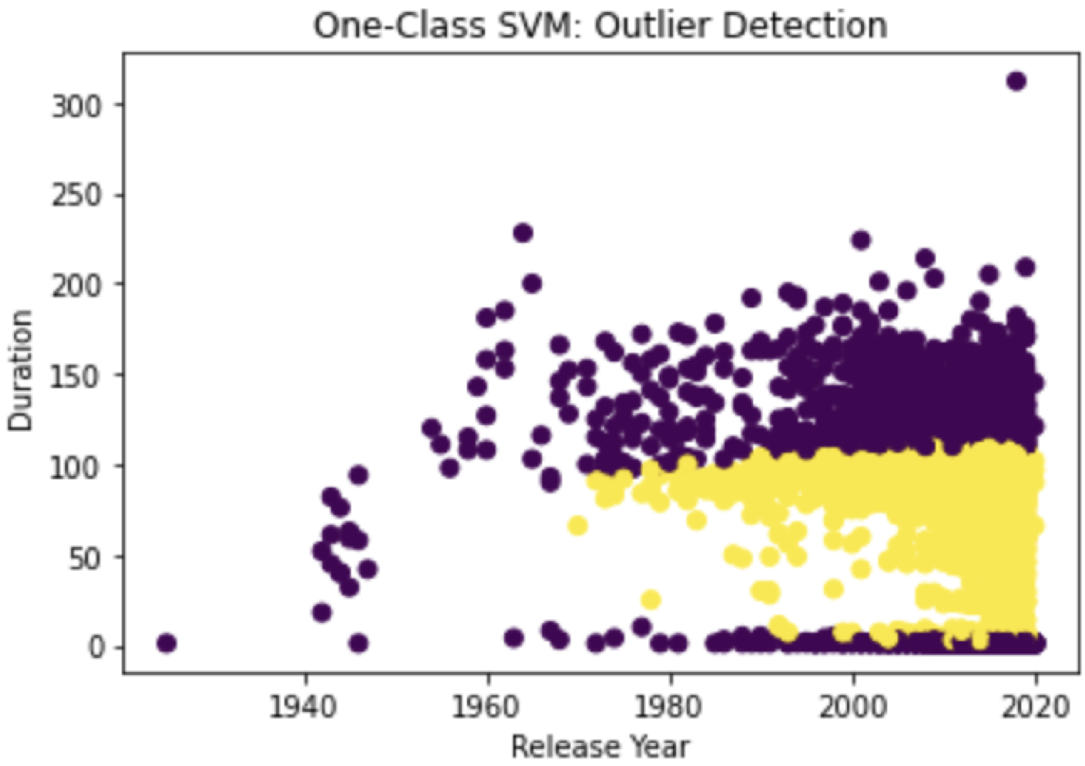
Explanation -



```
Number of normal data points: 3098
Number of outliers/anomalies: 3116
```

We have chosen this ML model to find outliers in the given input column. We used numerical features 'release_year' and 'duration' to conduct outlier identification using the One-Class SVM method.

The OneClassSVM class is instantiated without any hyperparameters being given. The model is then adjusted to the numerical feature data ('X') that was retrieved.



We can see the number of outliers from the output of scatter plot, Here we can see two classifications in the release year, the yellow data points correspond to the data points that are predicted as normal points by the One-Class SVM model. These are the data points that are considered to be within the learned pattern of the majority of the data.

On the other hand, the purple data points correspond to the data points that are predicted as outliers/anomalies by the One-Class SVM model. These data points are considered to deviate from the learned pattern of the majority of the data and are identified as unusual or anomalous instances.

7. Cosine similarity using TF-IDF

The similarity between two vectors in a high-dimensional space may be assessed using the cosine similarity metric. Cosine similarity may be used to determine how similar the TF-IDF (Term Frequency-Inverse Document Frequency) vectors of various things, such as movies or TV series, in the context of content-based recommendation systems. Here, we are evaluating our data with respect to the various pairs of columns such as Listed_in and Title and as well as Description and Title.

Algorithm -

To compare things based on their textual content, recommendation systems often utilise cosine similarity utilising the TF-IDF (Term Frequency-Inverse Document Frequency) method. This is how it goes:

Calculating the TF-IDF:

- Calculates the frequency of each word (term) in a text using the TF (Term Frequency) algorithm. Terms that occur more often in the manuscript are given greater weight.
- Measures a term's prominence over the whole document collection using IDF (Inverse Document Frequency). Less frequent words across papers are given more weight.
- In order to get a weighted score for each phrase in a document, TF-IDF integrates both TF and IDF.

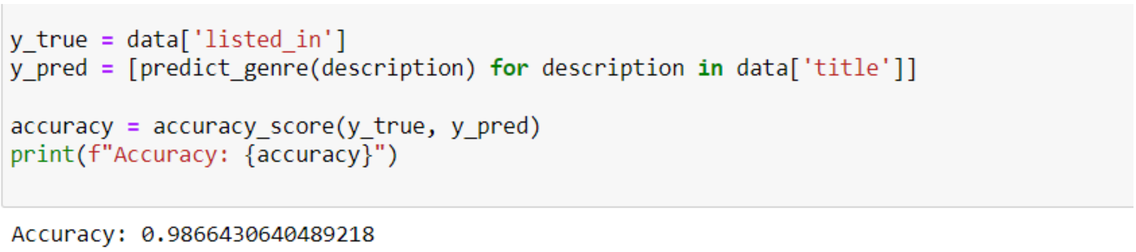
Representation of the Document:
Each dimension's value in the vector corresponds to a term's TF-IDF score in the document.

How to calculate cosine similarity:

- By computing the cosine of the angle between two vectors, cosine similarity gauges how similar they are.
- The vectors used in recommendation systems reflect the textual content of two different things (such as books, articles, or goods).
- A value of 1 indicates that the vectors are identical (high similarity), a value of 0 shows no similarity, and a value of -1 indicates full dissimilarity. Cosine similarity has a range of -1 to 1.
- Making recommendations

The system determines the cosine similarity between the user's profile (expressed as a vector) and the vectors of accessible objects in order to suggest things to a user.
Higher cosine similarity ratings indicate that the item is more likely to match the user's tastes, and so it is recommended.

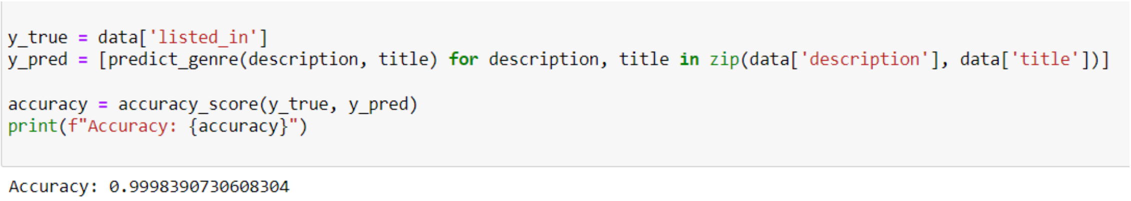
Checking accuracy with column ‘Listed_in’ scaled against column, ‘Title’.



```
y_true = data['listed_in']
y_pred = [predict_genre(description) for description in data['title']]

accuracy = accuracy_score(y_true, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9866430640489218
```

Checking accuracy with column ‘Listed_in’, scaled against two columns, ‘Description’ and ‘Title’.



```
y_true = data['listed_in']
y_pred = [predict_genre(description, title) for description, title in zip(data['description'], data['title'])]

accuracy = accuracy_score(y_true, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9998390730608304
```

Explanation -

The way how the cosine similarity works is by calculating the dot product and the magnitude. This is the widely used formula in the recommendation system and since our dataset and project is based on the recommendation of the content, this algorithm has yielded the best result of accuracy when compared to the other ML algorithms which we have used.

We aimed to predict the genre of any content based on their description and title. As both description and title columns are textual, we used TF-IDF to vectorize them. And then calculated the cosine values. So, we calculated cosine similarity between the TF-IDF vectors of different descriptions and titles.

We determined how similar the descriptions and titles are to each other based on their textual content. By comparing the cosine similarity values, we identified the most similar description and title to a given input and predicted its associated genre. For only title, the accuracy was 0.98 and for only description and title it was 0.99.

Using this, we were able to predict the genre of content based on the title and description. This is by far the best algorithm we have come up with, which gives a good accuracy.

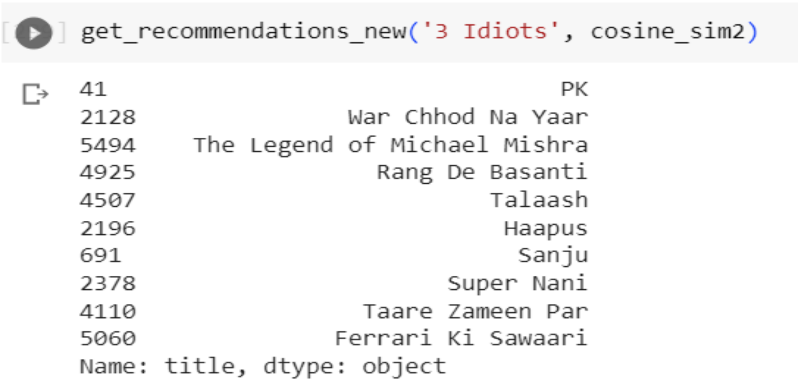
With the accuracy wavering around .98 and .99 we deduced that there might be a case of overfitting when we use this algorithm for our dataset. And if we add some regularisation techniques like L1 or L2, we may be able to reduce the overfitting and get a better accuracy which in turn would help us deduce what the best algorithm is, for our dataset.

Recommendation system

Using Cosine similarity, the user will input the name of a movie/TV show, and on the basis of the columns - title, director, description, cast and the genre, the system will recommend 10 similar movies/tv shows with highest cosine similarity index,

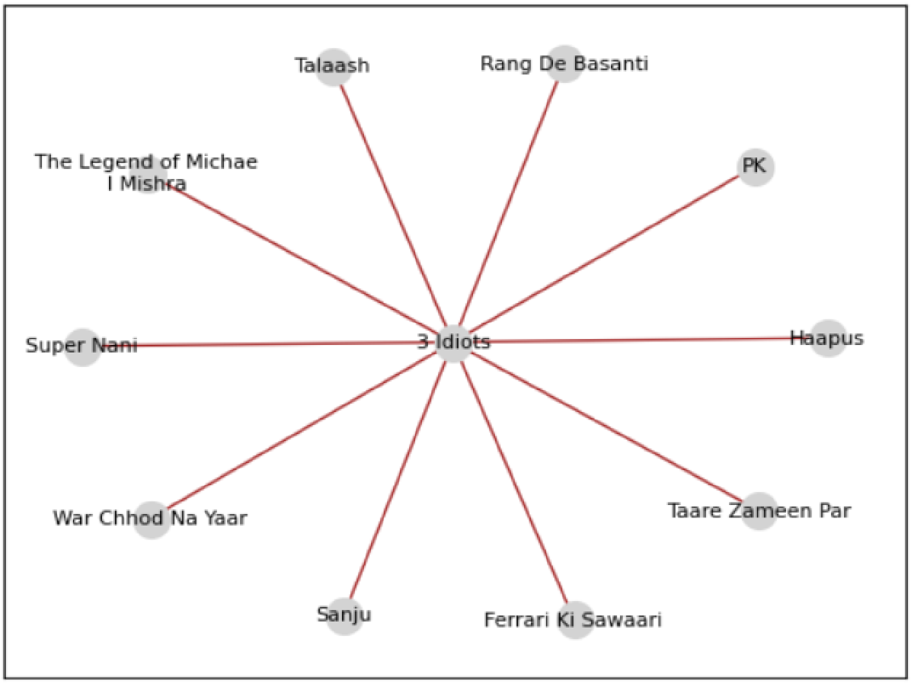
Algorithm -

Countvectorizer with cosine similarity (based on multiple metrics -title, cast, director, description, listed in)-Text documents are transformed into a matrix representation of token counts using the text preprocessing method CountVectorizer.



```
get_recommendations_new('3 Idiots', cosine_sim2)
```

```
41                                     PK
2128                     War Chhod Na Yaar
5494       The Legend of Michael Mishra
4925                       Rang De Basanti
4507                               Talaash
2196                                Haapus
691                                  Sanju
2378                            Super Nani
4110                      Taare Zameen Par
5060                    Ferrari Ki Sawaari
Name: title, dtype: object
```



Movie Recommendations Network for "3 Idiots"

It is expected that the cosine similarity between movies are represented as a matrix or array in the variable cosine_sim. This line creates a collection of tuples from the cosine similarity scores that correspond to the input title.

The index of the movie and its cosine similarity score to the input movie are both included in each tuple. In this example we have taken '3 Idiots' as the title for the cosine_sim of the recommendation system. Then, we calculated the similarity scores to select the top 10 movies that have the highest score.

Finally, We have used the network graph for the visualisation of the results, where we used nodes and edges to represent recommended contents for the given content based on multiple features.

References

- https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article
- https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=Summary-,The%20k%2Dnearest%20neighbors%20(KNN)%20algorithm%20is%20a%20simple,that%20data%20in%20use%20grows.
- https://builtin.com/data-science/step-step-explanation-principal-component-analysis

- https://medium.com/geekculture/understanding-tf-idf-and-cosine-similarity-for-recommendation-engine-64d8b51aa9f9
- https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html
- https://www.analyticsvidhya.com/blog/2022/06/one-class-classification-using-support-vector-machines/
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
- https://www.ibm.com/topics/random-forest
- https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm