Priyasri Sankaran
Instructor: Dr. Rishabh Iyer
Course: CS 6375
UTD -Fall 2024
Title: Netflix Movies and TV shows Genre Prediction

Project Outline: Genre Prediction- Netflix Movies and TV show Content

Motivation for the Project:

Time is precious. After a day of hard work, people sit down to relax. Due to the modern era, there are abandon ways to relax. One of the easy and comfortable way of entertainment is watching TV. With overwhelming options and variety of choices are available on our figure tip. Often it takes a good amount of time to find what we wish to watch. Scrolling through various shows or movie descriptions takes our precious time. Genre prediction helps in recommendation system to provide the views to choose something that they would enjoy.

Genre prediction project addresses the challenges the streaming services like Netflix. It helps them to enhance the user-friendly options, recommendation system, marketing their products and so on. This project gives me real life application using machine learning techniques. The idea of the project is endless. The application is potential for many real-life projects.

Problem Statement:

Given metadata about movies and TV shows, the goal is to predict the genre using various variables.

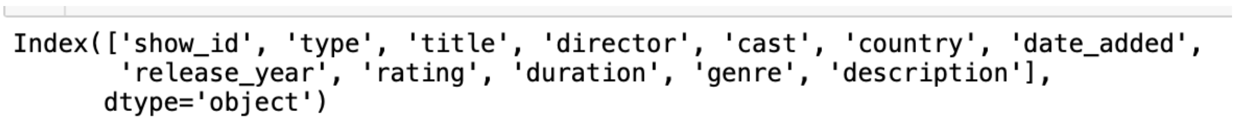
For the genre prediction-it involves **multi-label classification** task (since there are more than one genre).

Objective:

- Predicting genres based various features like type, cast, rating, director, description
- Exploratory Data Analysis
- Making predictions using Machine learning techniques.
- Evaluating the performances

Dataset

- **Source**: Kaggle.com
- Dataset size-8807 entries
- **Features**: The variables in the dataset are show_id, type, title, director, description, cast, country, data_added, release year, rating, genre and duration.



```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'genre', 'description'],
      dtype='object')
```

Features:

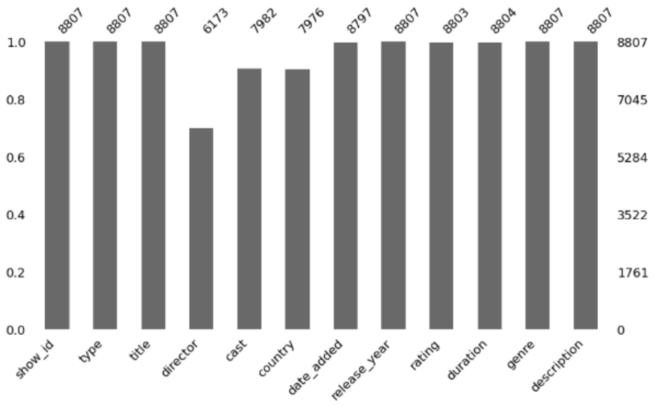
- Show_id : Unique ID for each Movie or TV show
- Type : Type of entertainment- Movie or TV Show
- Title: Name of the Movie or TV Show
- Director: Director of the Movie or TV Show
- Description: A brief summary of the Movie or TV Show
- Cast: Actors in the Movie or TV Show
- Country: Country where the Movie or TV Show produced
- data_added: Data content added to Netflix
- Release year: Original year the Movie/TV Show was released
- Rating: Rating of the Movie or TV Show
- Genre:Genre of the Movie or TV Show
- Duration: The time duration of the Movie or TV Show

Exploratory Data Analysis (EDA)

Data analysis is an important part of any machine learning model. Performing a deep analysis of data is a key factor. This process is essential for the consequence data processing steps.

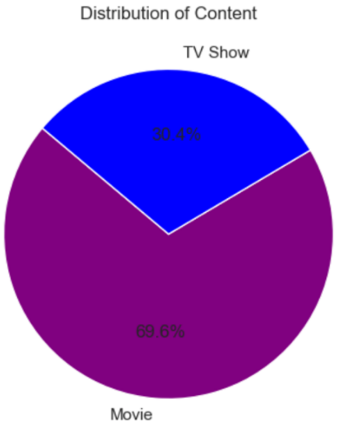
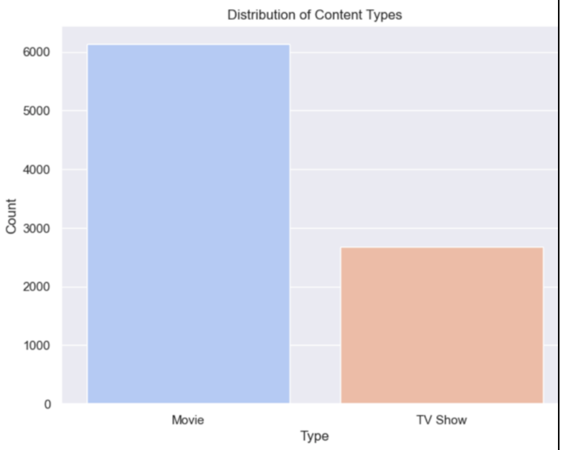
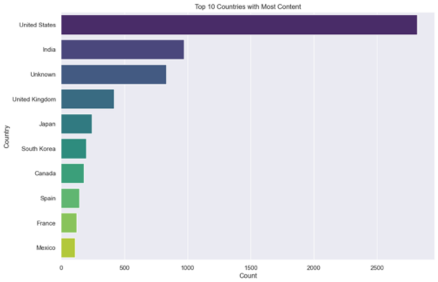
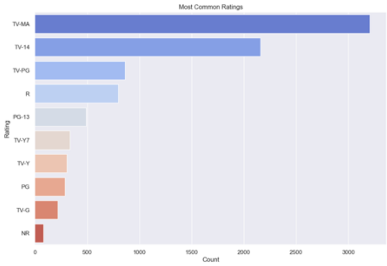
While preprocessing the data and performing the EDA, I came across several empty cells, such as director names are missing, actors names are missing in the cast column, country information is missing and the date_added column was missing values.

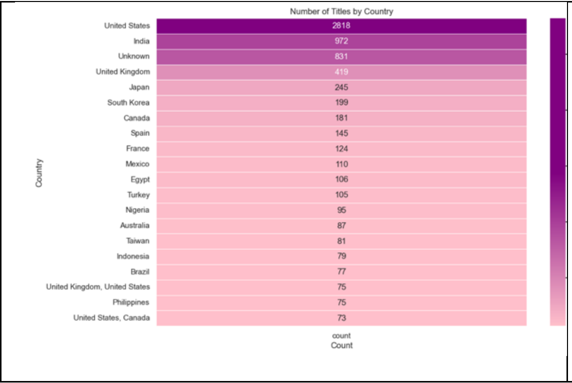
```
<Axes: >
```

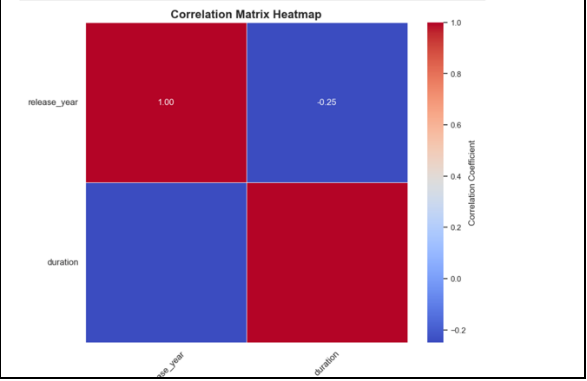


→The plot to the left gives a visual model of the missing values on the features of the dataset. We observe that 'director' column has more missing values. Cast and country are also has notable missing values.

- **Data Distribution**: Performed various combinations of variable distributions and the correlation between various features. An in-depth data analysis is performed, and the outcomes are analyzed as below.

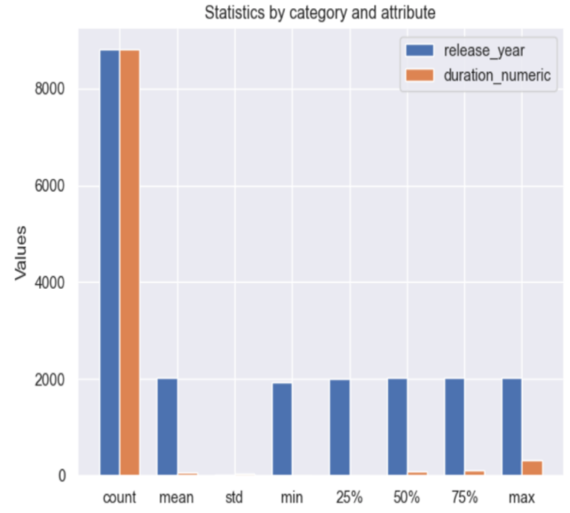
Figure 1: Show the distribution of the TV show and Movie, we clearly see that the distribution is not normal. 69.6% of movie content and 30.4 % of TX shows in the dataset.	Figure 2: Is a distribution of the type(movies and TV shows). Observe that it’s not normal distribution between movies and the TV shows
Figure 1  	Figure 2 
Figure 3: Shows the distribution of the countries by count. Definitely not notmal distribution. Observe that United States movie count is has most. And Mexico has the least count.	Figure 4: Shows the distribution of the rating by count. Looks like the TV-MA(mature Audience only) has a higher count of content in the dataset.
Figure 3  	Figure 4 
Figure 5: Shows the count by the country. Same us figure 2, but with exact numbers.	Figure 6: The below correlation matrix shows Heatmap shows the release_year and duration variable.
Figure 5	Figure 6

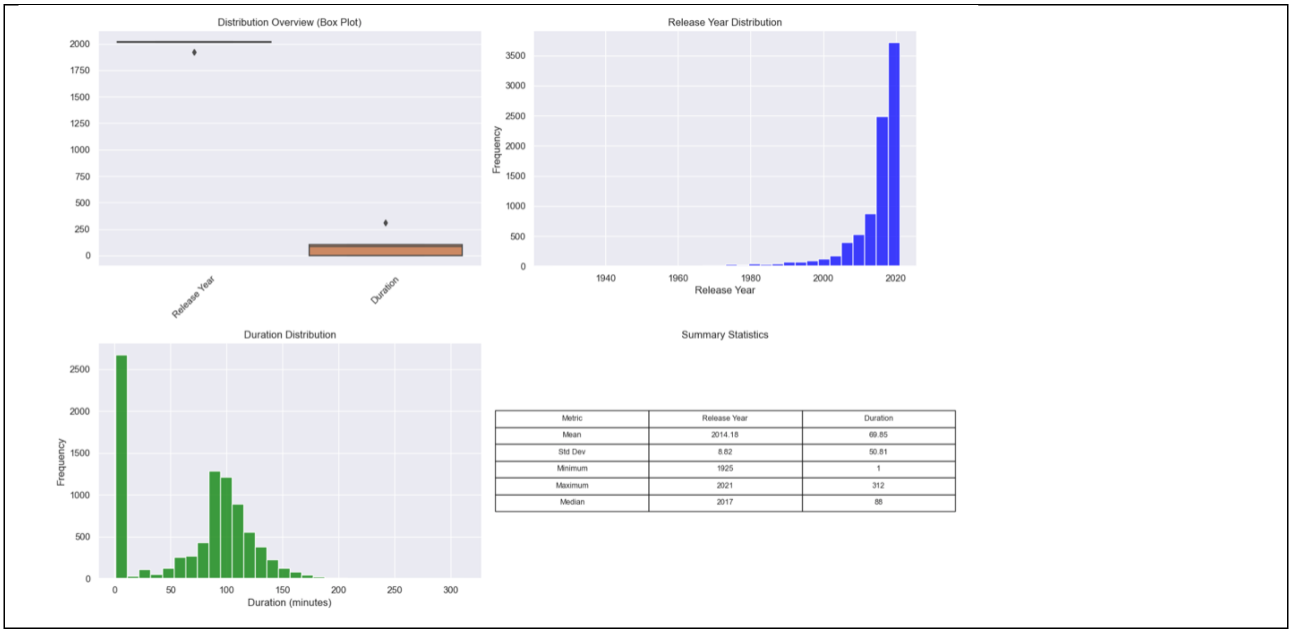




Summary statistics: Below the are the summary statistics for the data. Noted that the mean= 2014 and the standard deviation=8.819.

```
       release_year  duration_numeric
count   8807.000000       8804.000000
mean    2014.180198         69.846888
std        8.819312         50.814828
min     1925.000000          1.000000
25%     2013.000000          2.000000
50%     2017.000000         88.000000
75%     2019.000000        106.000000
max     2021.000000        312.000000
```



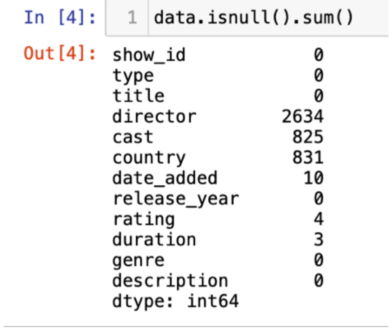


Summary Statistics

Metric	Release Year	Duration
Mean	2014.18	69.85
Std Dev	8.82	50.81
Minimum	1925	1
Maximum	2021	312
Median	2017	88

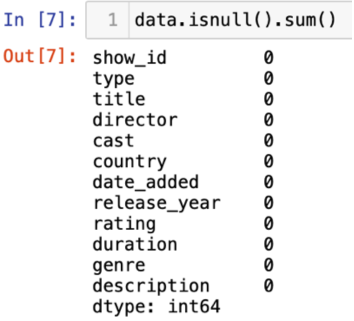
Feature Extraction: Have done the below steps, notice that the data is cleaned up and applied various necessary steps.

Missing Values: Missing values are taken care by cleaning up the data. Below are the results of the checking for null content before and after taking care of the missed values.



```
In [4]:  data.isnull().sum()

Out[4]:  show_id            0
         type               0
         title              0
         director        2634
         cast             825
         country          831
         date_added        10
         release_year       0
         rating             4
         duration           3
         genre              0
         description        0
         dtype: int64
```

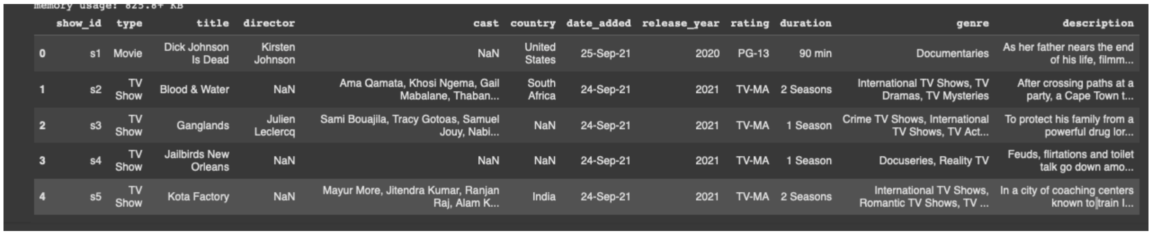


```
In [7]:  data.isnull().sum()

Out[7]:  show_id         0
         type            0
         title           0
         director        0
         cast            0
         country         0
         date_added      0
         release_year    0
         rating          0
         duration        0
         genre           0
         description     0
         dtype: int64
```

This step helped me identify the missing values and clean up.

Some of the columns like title, director, genre, description and other columns are in text format. Categorical data. Please see the below image.



memory usage: 825.8+ KB

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	genre	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	25-Sep-21	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	24-Sep-21	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	24-Sep-21	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	24-Sep-21	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	24-Sep-21	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

There were 12 columns are available on the dataset. Not all of them are of our interest. So we selected only what we are going to use. After the columns are selected, we performed some cleaning. We removed the punctuations, Stopwords that appears the most are removes. Cast column has the actors' name, removed the space, to avoid any confusion of picking up the wrong word.

Used the NLTK library is very popular Python library, that can help with text cleanup. It give us with easy interfaces to use. Use for tokenization, converting to lower case, punctuation, stemming and Stopwords.

TFID Vectorization:

There are various techniques to convert text to word embeddings. Calculates originality of a word by comparing the number of times a word appears in a document with the frequency of the word. It takes the product of two terms.

Model Selection and Training

Tried various models to find the best performing ones based on the accuracy.

I used the following classifier machine learning models to perform classification.

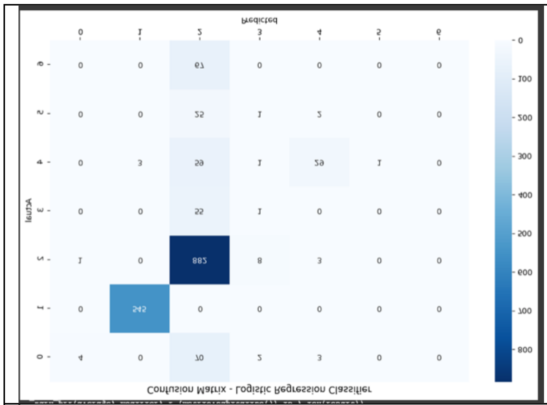
- Logistic Regression
- Naïve Bayes
- Support Vector Machine
- Decision Tree
- Random Forest
- Gradient Boosting

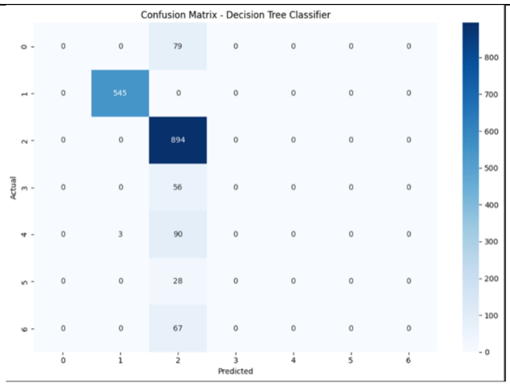
SKlearn has inbuild models for both supervised and unsupervised data. We used to the library for the above-mentioned classifiers. Standardized the dataset.

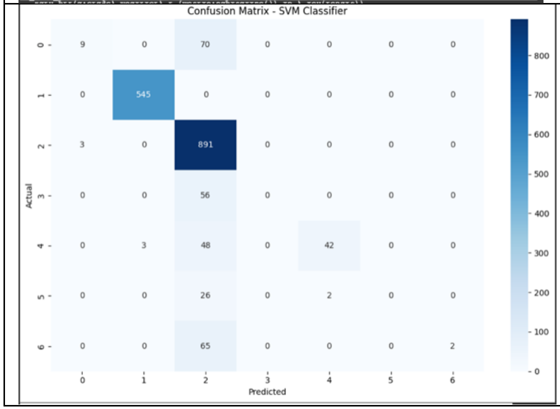
<table>
<tr><td colspan="2">Logistic Regression: Popular classifier, it calculates the probability, bounded between 0 and 1. Performed the logistic regression without hyperparameter tuning and with tuning, below are the accuracy obtained.</td></tr>
<tr><td>Before Hyperparameter tuning
Logistic Regression Accuracy:0.90</td><td>After Hyperparameter tuning
Logistic Regression Accuracy: 0.83</td></tr>
<tr><td colspan="2">Naïve Bayes: Uses the Bayes rule to predictions for the probabilities for the datapoints. It assumes that the distribution is normal and features are conditionally independent. It's used for text classification. Below are the results obtained before and after hyperparameter tuning.</td></tr>
<tr><td>Before Hyperparameter tuning
Naïve Bayes Accuracy: 0.49</td><td>After Hyperparameter tuning
Naïve Bayes Accuracy: 0.53</td></tr>
<tr><td colspan="2">SVM-Is a great technique to divide the data points into classes by creating hyperplane. The aim is to maximize the margin. Below are the values of accuracy obtained.</td></tr>
<tr><td>After Hyperparameter tuning</td><td>After Hyperparameter tuning</td></tr>
</table>

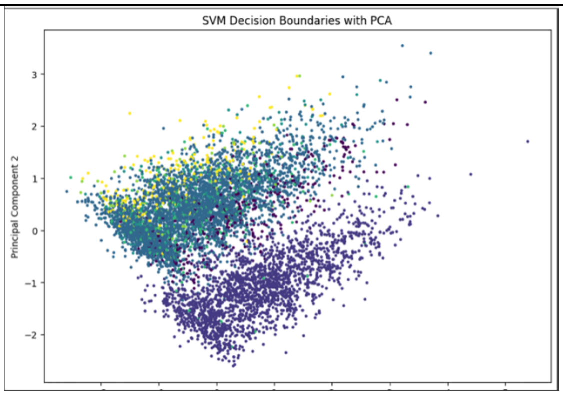
SVM Accuracy: 0.16	SVM Accuracy 0.85
Decision Tree: is a ono-parametric supervised learning algorithm, It a structed way of making the decision. Performed with and without hyperparameter tuning. Below are the results obtained.	
Before Hyperparameter tuning Decision Tree Accuracy: 0.82	After Hyperparameter tuning Decision Tree Accuracy: 0.82
Random Forest:-Random forest consist of multiple decision trees. Takes vote to make the final decision. Performed with and without hyperparameter tuning. Below are the results obtained.	
Before Hyperparameter turning Random Forest Accuracy: 0.82	After Hyperparameter tuning: Random Forest Accuracy: 0.82
Gradient Boosting: Is a ensemble machine learning technique that combines weak models into single, efficient model. Below are the results with and without hyperparameter tuning.	
Before Hyperparameter tuning Gradient Boosting Accuracy: 0.51	After Hyperparameter tuning Gradient Boosting Accuracy: 0.51

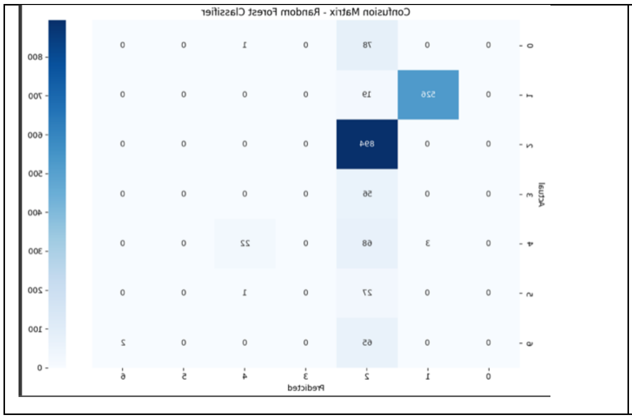
Confusion Matrix: Confusion matrix provides us a with the breakdown of the true positive, true negative, false positive and false negative values. With this we can calculate the accuracy and misclassification to evaluate the performance of the model. Using this we have achieved the accuracy of various model's performance and jotted on the above table.

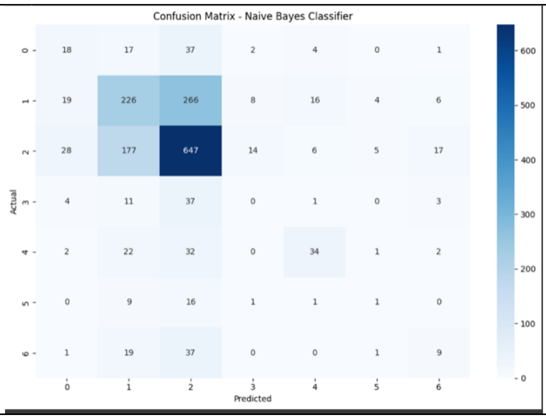








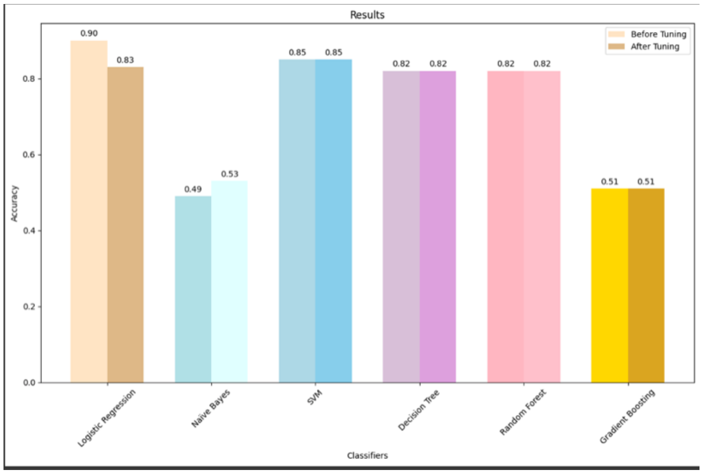




Confusion Matrix - Naive Bayes Classifier

Actual \ Predicted	0	1	2	3	4	5	6
0	18	17	37	2	4	0	1
1	19	226	266	8	16	4	6
2	28	177	647	14	6	5	17
3	4	11	37	0	1	0	3
4	2	22	32	0	34	1	2
5	0	9	16	1	1	1	0
6	1	19	37	0	0	1	9

Analysis: The accuracy of the SVM is 0.85. We notice the Logistic regression accuracy before and after the hyperparameter tuning, we notice that the accuracy is reduced to 0.83. It might be because it's struggling to capture the data structure. Then decision tree and random forest with both with accuracy value of 0.82 . Naïve Bayes performace is moderate with accuracy of 0.53. Gradient boosting gives the accuracy of 0.51. Overall performance of SVM and Logistic Regression is good ~ 85% accuracy. Below is the visual of the combined results for comparison. Also performed precision, recall, F1 score and crossvalidation.



Lesson Learnt:

Genre Prediction project gave a great opportunity to handle categorical data. The key factor is the preprocessing and model development stage. The libraries are wonderful resources, this project helped me to explore. This project helped for real world projects. Starting from the introductory data analysis to model analysis, encountered challenges and learnt a lot. The introductory data analysis helped me understand the why it's so important to perform these steps. In order to clean the data or use the correct variable, we need to understand the know the data. I never worked with a raw data before. Helped me understand the workflow of building a model. Error handling and optimization took me lot of time. Standardizing the data and hyperparameter tuning definitely improved the performance. Overall, it’s a great learning process. Looking forward to applying the knowledge gained through this in projects.

References

Kaggle.com
PCA-https://online.stat.psu.edu
https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.StandardScaler.html
https://www.geeksforgeeks.org/what-is-standardscaler/
http://betterstck.com/
http://stackoverflow.com/
https://docs.python.org/
https://www.analyticsvidhya.com/blog/
https://domino.ai/data-science-dictionary/sklearn
https://www.ibm.com/topics/logistic-regression
https://www.nltk.org
https://www.ibm.com
https://www.snowflake.com/guides/gradient-boosting/