Capstone Project Submission

**Instructions:**

1. Please fill in all the required information.
2. Avoid grammatical errors.

|  |
| --- |
| **Member’s Name, Email and Contribution: (Individual project)** |
| **Member : Harshad Savle**  **Email:** [harshad.savle@gmail.com](mailto:harshad.savle@gmail.com)  **Tasks :**   1. Clean and prepare the data for analysis. 2. Done Initial analysis and visualization. 3. Prepared Project Summary 4. Prepared Key Notes and conclusion 5. Done the visualization for analysis. 6. Added Useful Codes to simplify the analysis. 7. Prepared conclusions and PPT 8. Prepared introduction and key finding 9. Prepared Technical Documentation 10. Prepared Project Presentation |
| **Please paste the GitHub Repo link.** |
| **GitHub Link:** <https://github.com/harshadsavle/Netflix-Movies-And-TV-Shows>  **Google Drive link:** <https://drive.google.com/drive/folders/1SQMyucsmxYtVWmPnBaSiuN0sNmEAmhZh?usp=sharing> |

|  |
| --- |
| **Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions. (200-400 words)**  **Problem Statement:**  The dataset is collected from Flixable which is a third-party Netflix search engine. Netflix is the world's largest online streaming service provider, with over 220 million subscribers as of 2022-Q2. It is crucial that they effectively cluster the shows that are hosted on their platform in order to enhance the user experience, thereby preventing subscriber churn.  We will be able to understand the shows that are similar to and different from one another by creating clusters, which may be leveraged to offer the consumers personalized show suggestions depending on their preferences.  The goal of this project is to classify/group the Netflix shows into certain clusters such that the shows within a cluster are similar to each other and the shows in different clusters are dissimilar to each other.  **Our Approach:**  The approach followed here is to first check the sanctity of the data and then understand the features involved. The events followed were in our approach:   * + Select the attributes based on which you want to cluster the shows   + Text preprocessing: Remove all stop words and punctuation marks, convert all textual data to lowercase.   + Stemming to generate a meaningful word out of corpus of words.   + Tokenization of corpus and Word vectorization   + Dimensionality reduction   + Use different algorithms to cluster the movies, obtain the optimal number of clusters using different techniques.   + Build optimal number of clusters and visualize the contents of each cluster using word clouds |

# Conclusion from EDA:

* Greater number of count belong from movie class than Tv show class.
* 69% of data belong from movie class and 31% of data belong from Tv show.
* Alstair Fothergill directed highest shows in datalist which is 3.
* Raul campos and jan suter together have directed 18 higher than anyone in the dataset.
* Anupam kher, Shahrukh khan, Om Puri play highest number of role in the movies.
* Takahiro sakurai, Yuki Kaji, Daisuke Ono play highest role in the Tv shows.
* Most of the movie and tv shows have rating of TV-MA (Mature Audience) then followed by tv-14 (younger audience).
* Highest number of rating given for the movies as compared to TV shows it is pretty obevious because gighet number of category belong from movie class as we can seen earlier in type column
* Highest number of genre belong from classic and cult tv, tv thriller, stand-up comedy and talk show.

# Conclusion from Machine Learning:

* The dataset contained about 7787 records, and 11 attributes. We began by dealing with the dataset's missing values and doing exploratory data analysis (EDA).
* It was decided to cluster the data based on the attributes: director, cast, country, genre, rating and description. The values in these attributes were tokenized, preprocessed, and then vectorized using TFIDF vectorizer.
* Through TFIDF Vectorization, we created a total of 10000 attributes.
* We used Principal Component Analysis (PCA) to handle the curse of dimensionality. 3000 components were able to capture more than 80% of variance, and hence, the number of components were restricted to 3000.
* We first built clusters using the K-Means Clustering algorithm, and the optimal number of clusters came out to be 5. This was obtained through the elbow method and Silhouette score analysis.
* Then clusters were built using the Agglomerative clustering algorithm, and the optimal number of clusters came out to be 7. This was obtained after visualizing the dendrogram.
* A content-based recommender system was built using the similarity matrix obtained after using cosine similarity. This recommender system will make 10 recommendations to the user based on the type of show they watched.