**MOVIE RECOMMENDATION SYSTEM**

**PG Level Advanced Programme in Applied Data Science and Machine Learning**

ADSML Talent Sprint Cohort 4

**Group 2**

|  |  |
| --- | --- |
| **S:NO** | **NAMES** |
| **1** | **Deepika V** |
| **2** | **Harmit Singh** |
| **3** | **Priti Manickavachakam** |
| **4** | **Mrityunjay Balkrishnan** |
| **5** | **Sourav Sahu** |
| **6** | **Mohd Arif Ansari** |
| **7** | **Ipsita Roy Chowdhury** |
| **8** | **Kritika Thadani** |
| **9** | **Varun Sharma** |
| **10** | **Bharath Vasudevan** |

THE PROBLEM STATEMENT:

To build a movie recommendation system for the provided movie data that recommends the movie as per user’s preferences and find new, desirable content for them automatically based on the pattern between their likes and rating of different items. This system aims to enhance user experience by suggesting movies that users are likely to enjoy, thus increasing user engagement and satisfaction. The key goals of a Movie Recommender System include:

1. Personalization: Tailoring recommendations to individual users' tastes, preferences, and viewing history.
2. Accuracy: Providing accurate and relevant recommendations that align with users' interests.
3. User Engagement: Reducing user’s browsing time and encouraging users to explore new movies and genres, thereby increasing user engagement and user retention.

Dataset Description :

In order to achieve this objective, we decided to build a movie recommendation system, that rank order the movie for User as per ratings of movies unseen by users. Ratings are predicted according to the genres and tags associated with the movies they've liked and disliked.

The dataset was obtained from group lens (University of Minnesota, <https://grouplens.org/datasets/movielens/>), specifically the "MovieLens 25M

**Dataset** : <http://files.grouplens.org/datasets/movielens/ml-25m.zip> (250 MB)

The resulting Data Frame produced by this system will display the top ten movies with the highest predicted ratings

**Dataset contains 6 files listed below**:

|  |  |
| --- | --- |
| rating.csv | All Ratings for each Movie id by user contains user Id, movie Id, rating, timestamp. Ratings ranges between 0.5 and 5 |
| movie.csv | Contains movie related information movie id, title, genres. All genres that movie belongs to are same row separated by "|". Title also has year in which movie was released. |
| genome\_scores.csv | Contains tag relevance scores for movies. |
| genome\_tags.csv | Contains tagId and tags |
| *tag.csv* | Contains tags data. Tags given to one movie by one user. |
| link.csv | Identifiers that can be used to link to other sources of movie data are contained in the file links.csv. We decided not to use this file for movie recommendation. |

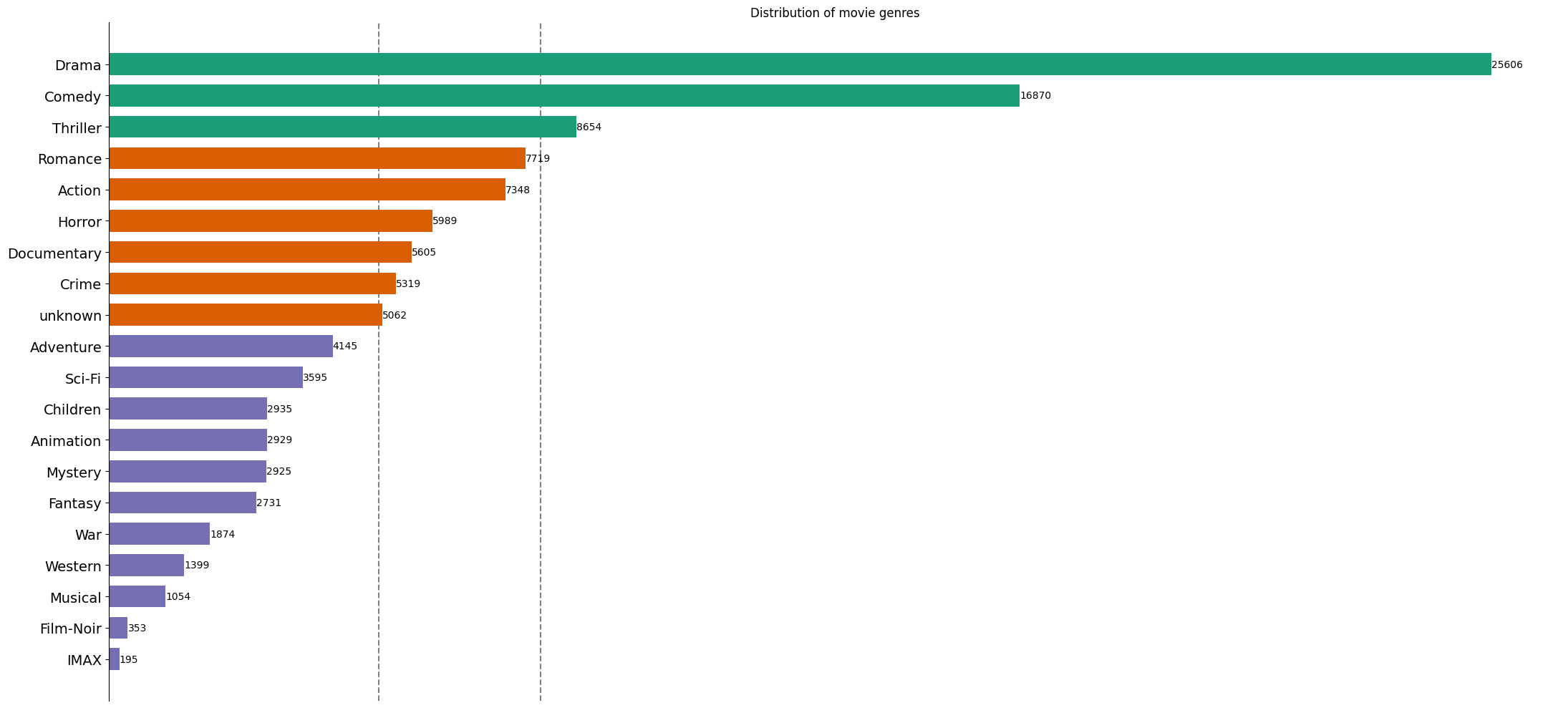
METHODOLOGY :

To build a movie recommendation system, you can follow a methodology that combines collaborative filtering (such as Singular Value Decomposition - SVD) for user-item interactions and content-based filtering (using TF-IDF and cosine similarity) for item-item relationships. Here's a step-by-step methodology you could use:

1. **Data Analysis and Pre-processing**
   * Download the data on user ratings and movie.
   * Pre-process the data, handle missing values, and ensure data consistency.
2. **Exploratory Data Analysis (EDA):**
   * Explore the dataset to understand the distribution of ratings, genres, user preferences to identify the relevant features to build the model.
   * Identify trends and patterns in user behaviour, such as popular genres or highly rated movies.
3. **Collaborative Filtering (CF) with SVD:**
   * Split the data into training and testing sets.
   * Apply SVD algorithm using the surprise package to the training data to decompose the user-item matrix and extract latent factors representing user preferences and item features.
   * Use the SVD model to predict ratings for unseen user-item pairs in the test set.
   * Evaluate the model's performance using metrics like RMSE (Root Mean Squared Error) or MAE (Mean Absolute Error).
4. **Content-Based Filtering (CBF) with TF-IDF and Cosine Similarity:**
   * Pre-process movie metadata, such as titles, genres, and tags
   * Apply TF-IDF (Term Frequency-Inverse Document Frequency) to convert textual data into numerical vectors, capturing the importance of words in each movie's description.
   * Compute pairwise cosine similarity between movie vectors to measure their similarity based on content.
   * Use the cosine similarity matrix to recommend movies similar to a given movie.
5. **Random Forest**
   * Combine collaborative filtering and content-based filtering to identify the top 10 movies from both approaches based on user’s last watched movies
   * Predict the ratings for these top 20 movies from both the approach and top 10 movies from user’s top 3 preferred genres released in Mar 2009 or Apr 2009.
6. **Model Evaluation:**
   * Evaluate the system's overall performance using metrics like R2 Score and RMSE for top-10 recommendations for Apr 2009
7. **Next Steps:**
   * Explore the Deep Learning Models
   * Explore the ALS for Movie recommendation System
   * Explore User Segmentation approach to improve the model training and scoring time.

**MAJOR FINDINGS FROM EXPLORATORY DATA ANALYSIS**

**THE PLOT OF THE NUMBER OF MOVIES IN EACH GENRE:**



**THE PLOT OF TOP 10 MOVIES WATCHED FREQUENTLY**

**A green and yellow bar graph

Description automatically generated**

**DISTRIBUTION OF MOVIE RATINGS**

**A blue bar graph with white text

Description automatically generatedA blue and black diagram

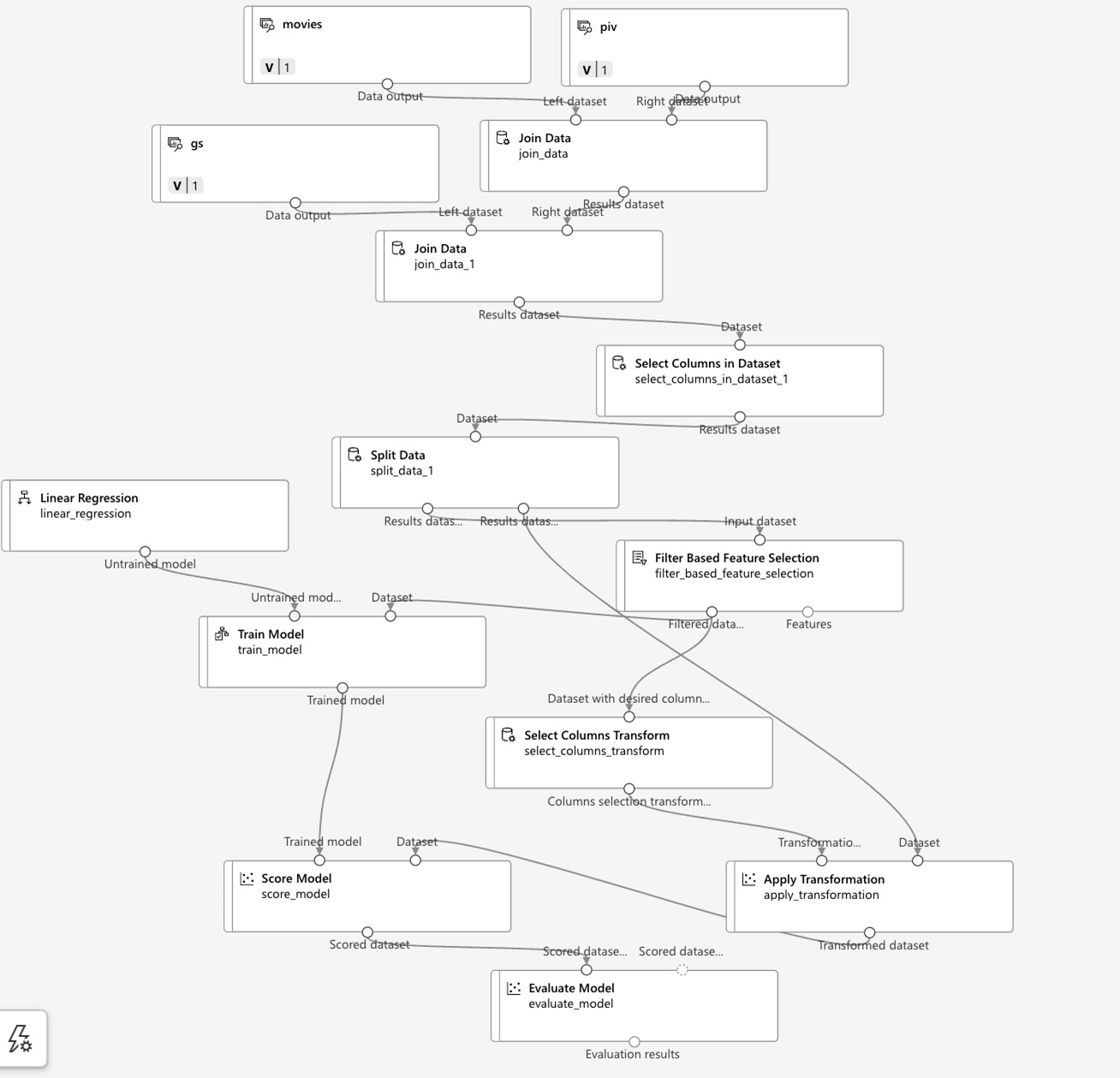
Description automatically generated**

**COLLABORATIVE FILTERING WITH SVD:**

To build a SVD Algorithm, we used only the ratings file and calculated 8 Latent Features which gave us the below Model Performance Metrics

****

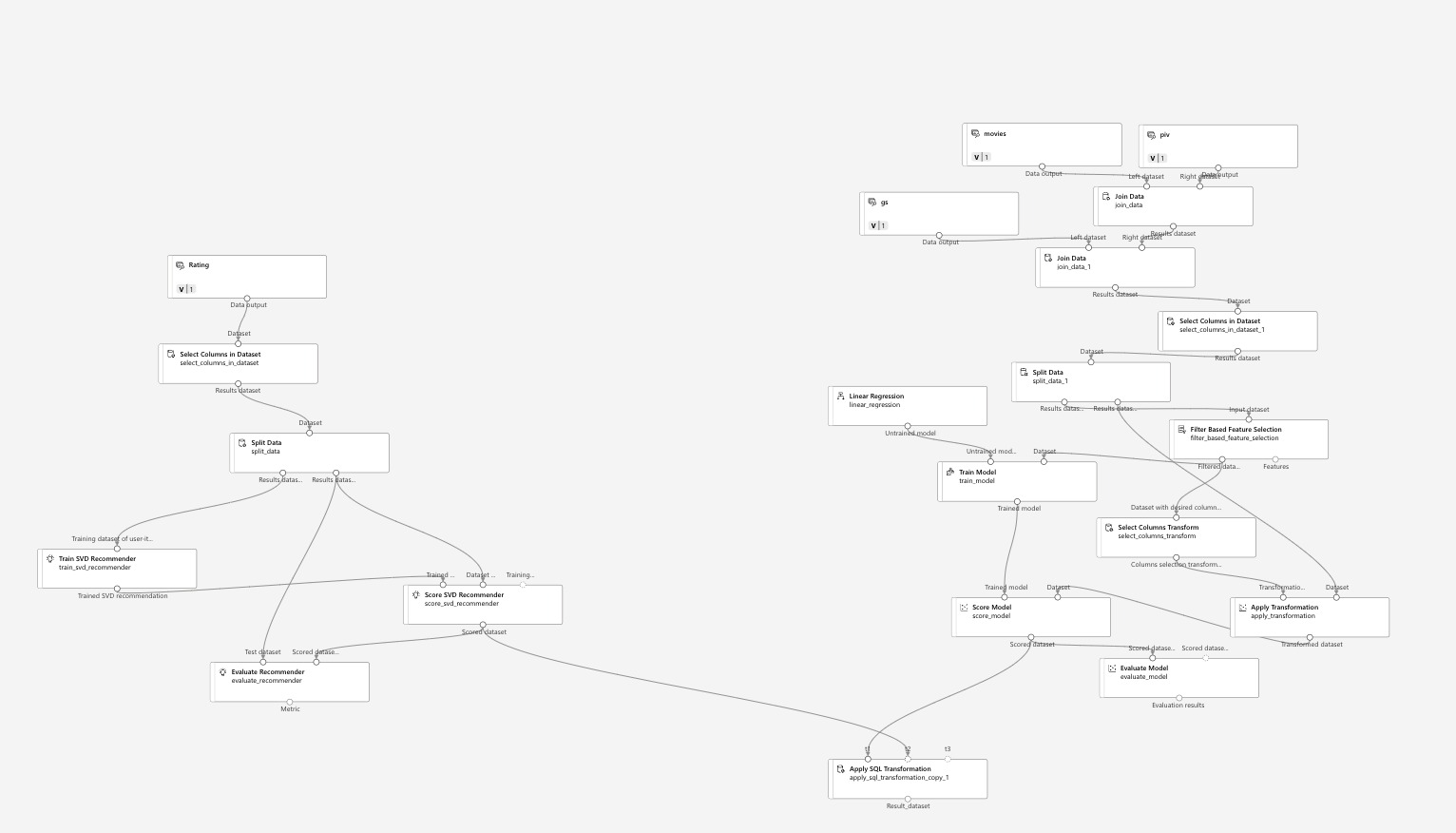
**CONTENT BASED FILTERING - TF-IDF**

****

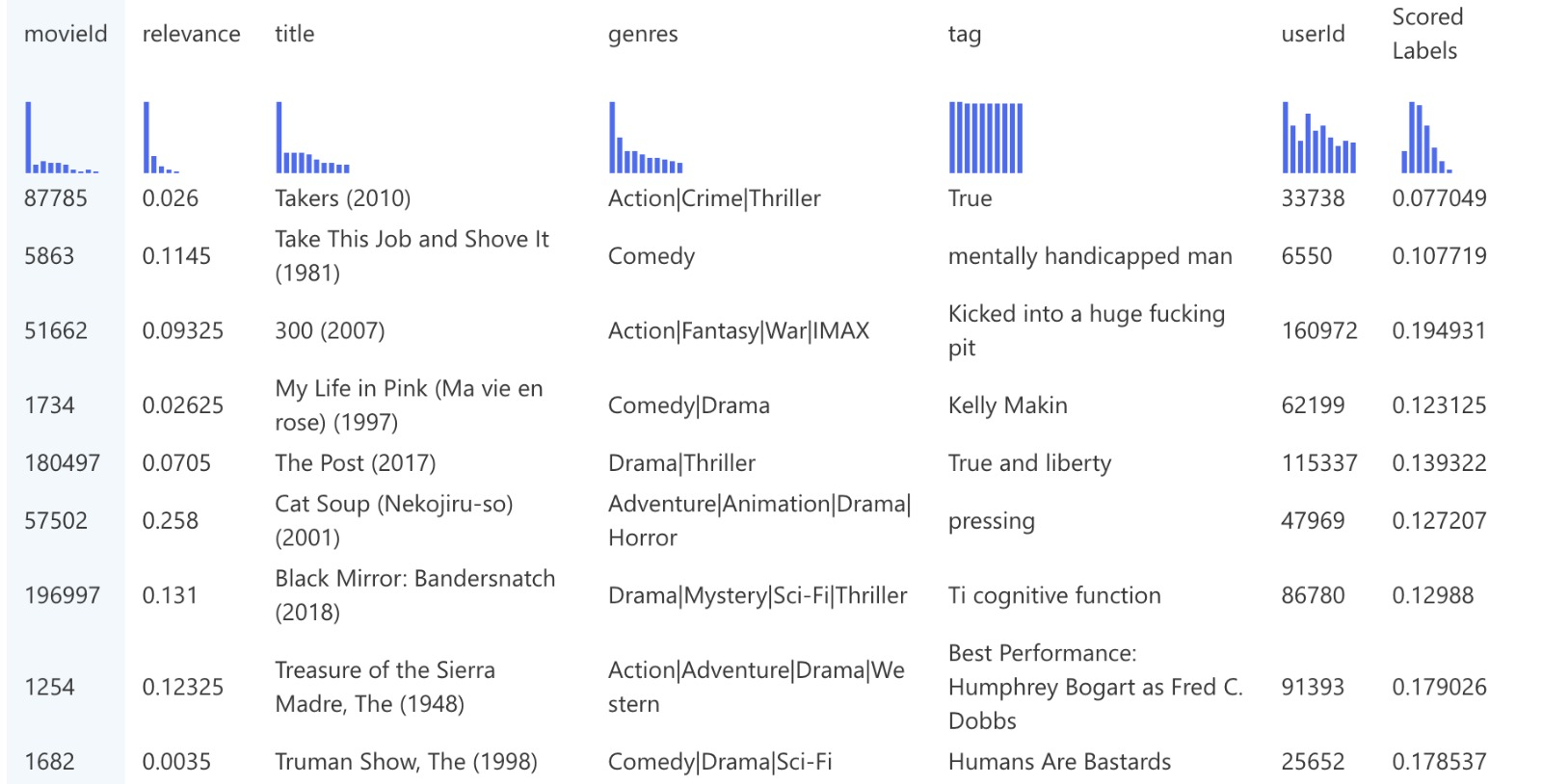
**CONFUSION MATRIX FOR TF-IDF:**

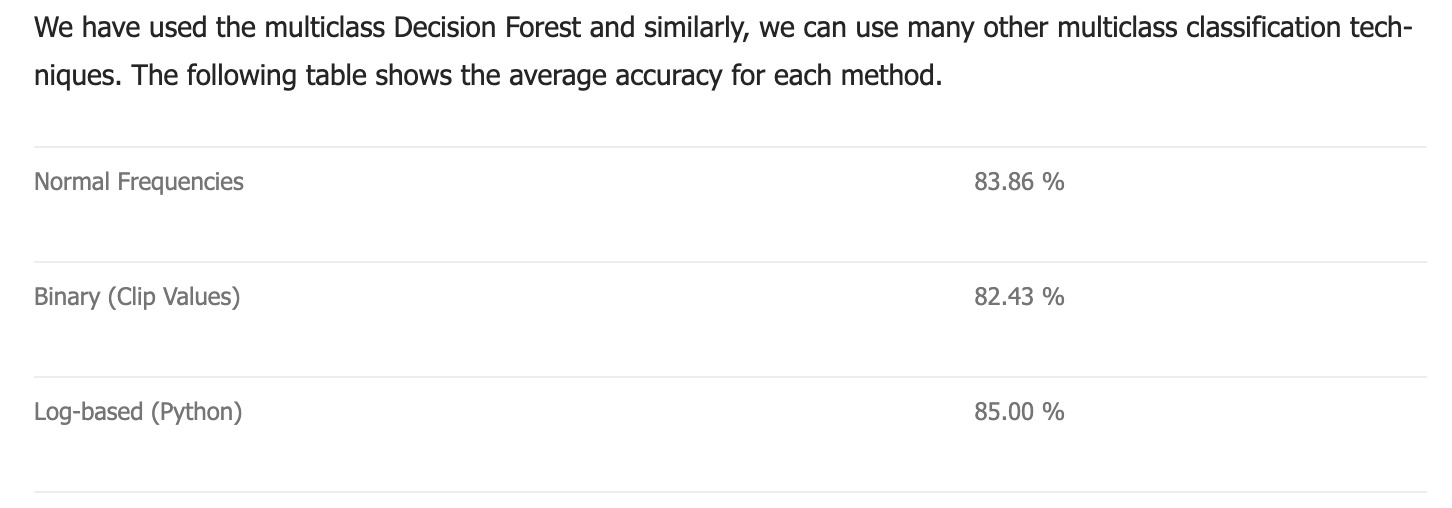


**OVERALL FLOW- SVD + TF-IDF**

****

**OUTPUT – SVD+TF-IDF:**

****

****

**RANDOM FOREST :**

**Features:**

1. Number of times the movie was watched
2. Average Rating
3. Number of times the movie was watched in last month
4. One hot encoding for genome scores and values were updated by relevance score when available else fill 0.
5. Ratings by User
6. Random forest model was evaluated using

'R² Score': 0.08158589350458367, 'RMSE': 0.793708011836573}

**A diagram of a company

Description automatically generated with medium confidence**

**INTEGRATED SOLUTION :**

1. User’s Last watched Movie was identified.
2. Using Collaborative filtering, top 10 movies were identified for each user
3. Using Content-based filtering, top 10 movies were identified for each MovieId that was last watched by movie.
4. User’s preferred Genre was identified based on relevance score for genre and user’s rating was predicted for top 10 movies in that genre.
5. User’s predicted rating was rank ordered to display the top 10 movies for each User.

**CHALLENGES:**

1. Movie Lens 25M dataset is very huge dataset. It has insufficient data in Genome Scores and Tags file due to which accuracy drops drastically.
2. To overcome these issues, we created a catalog of movies which had relevance score in genome scores while applying the random forest.
3. User’s last watched movie was further filtered to have movies that had relevance score for its genre.
4. It was very challenging to train the XgboostClassfier model or Deep Learning Models to improve the performance of the recommendation system.

**ALTERNATE APPLICATION OF RECOMMENDED SYSTEMS :**

Recommender systems find applications across various domains but it differs in each domain with unique characteristics and challenges. It is essential to understand the domain-specific features required to train the recommendation mode. Here are some common applications of recommender systems along with how they differ from movie recommendation systems:

1. E-commerce:
   * Recommending products to users based on their browsing history, purchase behaviour, and preferences.
   * In addition to considering factors like product attributes, user demographics, seasonality, and pricing dynamics are to be included for model training.
2. Travel and Hospitality:
   * Recommending travel destinations, accommodations, flights, activities, and travel packages to users based on their preferences, budget, travel history, and reviews.
   * Travel recommendation systems incorporate location-based services, user reviews, ratings, booking patterns, loyalty programs, seasonal trends, and travel constraints (e.g., visa requirements, weather conditions). They may also integrate external APIs for real-time pricing and availability.
3. Content (News/Article/Video) Recommendation:
   * Personalizing news articles, blogs, videos, and other content for users based on their interests, reading habits, engagement levels, and topical relevance.
   * Additionally, Content recommendation systems needs to analyse text, metadata, user feedback, click-through rates, and topic modelling techniques to suggest relevant content. They also consider freshness, diversity, and editorial guidelines while balancing personalized recommendations
4. Social Media:
   * Recommending posts, articles, videos, or connections to users on social platforms based on their interests, social network structure, and engagement patterns.
   * Social media recommendation systems leverage user-generated content, social graphs, user interactions (likes, shares, comments), and real-time updates. They often incorporate contextual information such as trending topics, events, and personalized notifications.
5. Job Portals:
   * Matching job seekers with relevant job listings based on their skills, experience, location, industry preferences, and job search history.
   * Job recommendation systems focus on matching candidates to job requirements, considering factors like skills gaps, career progression, salary expectations, company culture, and geographic preferences.
6. Banking and financial services:
   * Recommend financial products such as credit cards, loans, investment options, insurance plans, and savings accounts to customers based on their financial profile, needs, and preferences.
   * Tailoring recommendations based on individual financial goals, risk tolerance, life stages (e.g., students, professionals, retirees), and specific needs (e.g., travel, education, retirement planning). Assessing credit risk, and Mitigating biases in recommendation algorithms related to age, gender, ethnicity are critical success factors.

**ACKNOWLEDGEMENT :**

We, the team members of Group 2/ Cohort 4 in PG Level Advanced Program in Applied Data Science and Machine Learning are very grateful to the mentor support throughout the course

Also, we would like to thank Mohit, our Mentor for guiding us in each step to complete the project and make it a success