Name: - Hitarth Padaliya

AI/ML Project

Movie – Recommender

Problem Statement: -

This project focuses on developing an AI-powered recommendation system that delivers personalized content suggestions for users. By analyzing factors such as user profiles, browsing and search history, demographic characteristics, and preferences of similar users, the system aims to generate tailored recommendations. The objective is to enhance user experience, satisfaction, and engagement through intelligent content curation.

Keywords: -

Collaborative Filtering, KNN, PCA, SVD, Hybrid Model

1 Introduction

1.1 Background

The project revolves around exploring traditional machine learning (ML) models commonly used in movie recommendation systems. These models exploit collaborative filtering techniques such as user-based and item-based filtering, which uses similarities between users or items to generate recommendations. User-based filtering identifies users with similar preferences and recommends movies based on their collective behavior, while, item-based filtering suggests similar movies given one movie as an input.

The objective is to develop an efficient and reliable movie recommendation system that provides personalized suggestions based on user behavior and preferences.

1.2 Overview

Many different techniques to be used in movie recommendation system wee explored. Mainly, two types of methods are used, user-based collaborative filtering and item based collaborative filtering. User based collaborative filtering is implemented using 3 different models which are KNN, PCA and SVD. Item based collaborative filtering were also implemented using same models. After analyzing for efficiency and reliability of results, one can conclude that PCA and SVD are similar techniques, thereby giving similar results.

Apart from this, a hybrid model is created that uses K-means clustering and Random forest to generate recommendations based on user's activity as well as analyzing various features of movie. By creating this model, I have been able to link different ML techniques in one model.

2 Dataset Used

The dataset utilized in this project originates from **MovieLens**, a movie recommendation platform, capturing 5-star rating data and free-text tagging activity. It comprises **100,836 ratings** and **3,683 tag applications** across **9,742 movies**, with all included users having rated at least **20 movies**. The dataset consists of four primary files: **links.csv, movies.csv, ratings.csv, and tags.csv**.

Additionally, two supplementary dataset files, **imdb.csv** and **tmdb.csv**, were created by scraping data from IMDb and TMDb. These datasets include key features such as **ratings, user reviews, and critic evaluations**, enhancing the recommendation system's effectiveness.

1. **'Links.csv':** It contains 3 columns, movieId (which is consistent throughout all the files), imdbId and tmdbId through which we can access the movie lens, imdb and tmdb webpage of the particular movie using id as suffix in the url.

2. **'Movies.csv':** Movie information is contained in this file and each row represents one movie. It contains 3 columns, movieId, title and genres. Title of the movie also contains year in which it was released. Genres are divided into 18 different categories and one movie can have multiple.

3. **'Ratings.csv':** Each row of this file represents one rating of one movie by one user. It has 4 columns, userId, movieId, rating and timestamp. Ratings are made on 5-star scale with half-star increment. Timestamp represents seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

4. **'Tags.csv':** Each row represents one tag applied to one movie by one user. It has 4 columns, userId, movieId, tag and timestamp. Tags are user generated metadata about movies. Each tag is typically a short phrase or word. The meaning, value and purpose of a particular tag is determined by each user.

3 Approaches Tried

Following approaches are used:

1. Collaborative filtering using KNN
2. Collaborative filtering using PCA
3. Collaborative filtering using SVD
4. Hybrid Model

3.1 Collaborative filtering using KNN

Using **K-Nearest Neighbors (KNN)**, I have implemented both **user-based and item-based collaborative filtering**. First, a **movie-user matrix** is constructed to represent the ratings each user has given to different movies. By identifying the **k nearest neighbors** to a user's rating vector, personalized movie recommendations are generated based on the preferences of similar users. This approach is particularly beneficial for streaming platforms, where customized content suggestions appear on a user's homepage, providing recommendations without genre bias.

To extend this further for **genre-specific recommendations**, I filtered the movie-user matrix to include only movies from a particular genre and applied KNN in the same manner.

Additionally, the model supports **item-based recommendations**, where instead of focusing solely on user preferences, it identifies **similar movies** using KNN. This feature is especially useful when users search for a specific movie or finish watching one, as it helps suggest related content, enhancing the overall viewing experience.

3.2 Collaborative filtering using PCA

Similarly, using this model, I have implemented three types of filtering: **general and genre-specific user-based filtering** and **item-based filtering**. For **item-based filtering**, a similar **movie-user matrix** is created. The **covariance matrix** is then computed from the **normalized movie matrix** to determine **eigenvalues and eigenvectors**, which play a crucial role in calculating **cosine similarity scores**. By leveraging cosine similarity, the model identifies the **top "n" movies** that are most similar to a given movie, enabling tailored recommendations based on movie similarities.

For **user-based filtering**, item-based filtering is applied to all the movies rated by a user. From the resulting recommendations, **two movies per rated movie** are selected for the final recommendation list. To ensure high-quality suggestions, movies rated by the user are sorted in **descending order of rating**, ensuring that the recommendations are aligned with the user's top-rated content. For **genre-specific recommendations**, only movies belonging to the **same genre** as the user's preferences are selected from the recommendations, providing a more targeted and relevant viewing experience.

3.3 Collaborative filtering using SVD

SVD splits the **normalized movie-user matrix** into three simpler matrices: one capturing the relationship between **users and latent factors**, another representing the **importance of each latent factor**, and the third depicting the relationship between **items and latent factors**. Using **cosine similarity**, recommendations were generated for **item-based filtering**.

For **user-based filtering**, the approach was similar to **PCA**, where **item-based filtering** was applied to movies rated by the user, and recommendations were derived from its output. As expected from theory, it was also observed in practice that **PCA and SVD** are similar techniques, yielding comparable results.

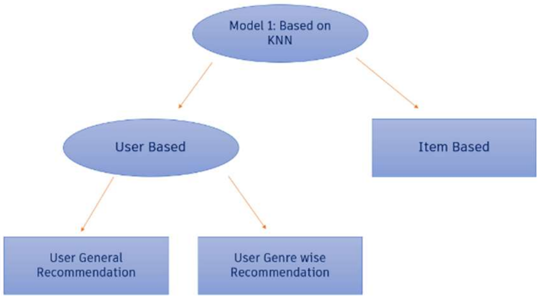
3.4 Hybrid Model

In this model, the **K-means clustering algorithm** was initially used to group users based on their ratings, ensuring that users with similar preferences were clustered together. A **training dataset** was then created using the movies rated by users within each cluster. To build the recommendation model, a **Random Forest Regressor** was trained on this dataset, as it was well-suited for handling the numerous features available. Along with genres, additional features extracted from various sources were also incorporated.

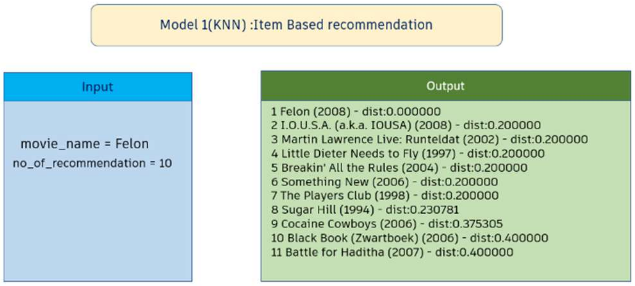
Once trained, the model predicted ratings for all movies, and recommendations were generated accordingly. This **two-step approach** ensures that recommendations are **personalized within user clusters** while also leveraging the **predictive power of Random Forest** to improve recommendation accuracy by considering multiple influential features.

4 Results

1. Collaborative filtering using KNN



(a) Model


Model 1(KNN) :Item Based recommendation

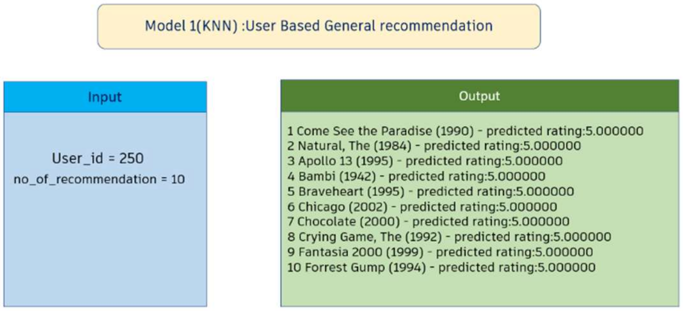
Input

movie_name = Felon
no_of_recommendation = 10

Output

```
1 Felon (2008) - dist:0.000000
2 I.O.U.S.A. (a.k.a. IOUSA) (2008) - dist:0.200000
3 Martin Lawrence Live: Runteldat (2002) - dist:0.200000
4 Little Dieter Needs to Fly (1997) - dist:0.200000
5 Breakin' All the Rules (2004) - dist:0.200000
6 Something New (2006) - dist:0.200000
7 The Players Club (1998) - dist:0.200000
8 Sugar Hill (1994) - dist:0.230781
9 Cocaine Cowboys (2006) - dist:0.375305
10 Black Book (Zwartboek) (2006) - dist:0.400000
11 Battle for Haditha (2007) - dist:0.400000
```

(b) Item based using KNN


Model 1(KNN) :User Based General recommendation

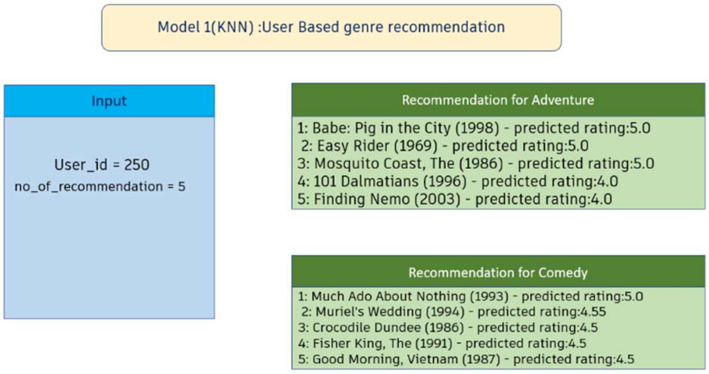
Input

User_id = 250
no_of_recommendation = 10

Output

```
1 Come See the Paradise (1990) - predicted rating:5.000000
2 Natural, The (1984) - predicted rating:5.000000
3 Apollo 13 (1995) - predicted rating:5.000000
4 Bambi (1942) - predicted rating:5.000000
5 Braveheart (1995) - predicted rating:5.000000
6 Chicago (2002) - predicted rating:5.000000
7 Chocolate (2000) - predicted rating:5.000000
8 Crying Game, The (1992) - predicted rating:5.000000
9 Fantasia 2000 (1999) - predicted rating:5.000000
10 Forrest Gump (1994) - predicted rating:5.000000
```

(c) User based general


Model 1(KNN) :User Based genre recommendation

Input

User_id = 250
no_of_recommendation = 5

Recommendation for Adventure

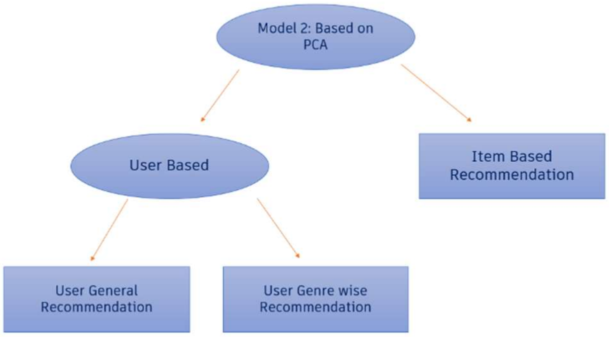
```
1: Babe: Pig in the City (1998) - predicted rating:5.0
 2: Easy Rider (1969) - predicted rating:5.0
3: Mosquito Coast, The (1986) - predicted rating:5.0
4: 101 Dalmatians (1996) - predicted rating:4.0
5: Finding Nemo (2003) - predicted rating:4.0
```

Recommendation for Comedy

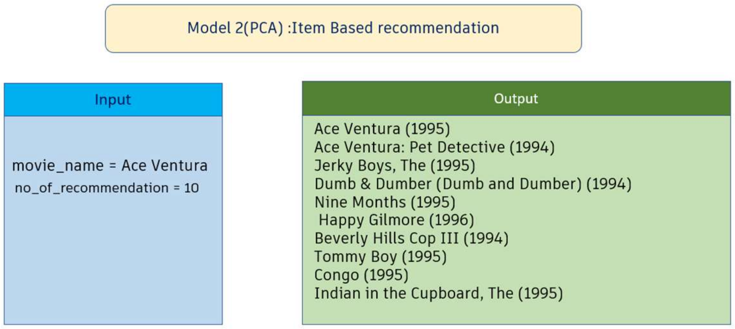
```
1: Much Ado About Nothing (1993) - predicted rating:5.0
 2: Muriel's Wedding (1994) - predicted rating:4.55
3: Crocodile Dundee (1986) - predicted rating:4.5
4: Fisher King, The (1991) - predicted rating:4.5
5: Good Morning, Vietnam (1987) - predicted rating:4.5
```

(d) User based genre wise

2. Collaborative filtering using PCA


Model 2: Based on PCA
User Based
Item Based Recommendation
User General Recommendation
User Genre wise Recommendation

(a) Model


Model 2(PCA) :Item Based recommendation

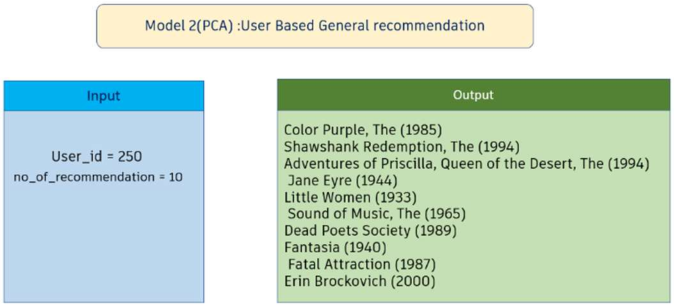
Input

movie_name = Ace Ventura
no_of_recommendation = 10

Output

```
Ace Ventura (1995)
Ace Ventura: Pet Detective (1994)
Jerky Boys, The (1995)
Dumb & Dumber (Dumb and Dumber) (1994)
Nine Months (1995)
 Happy Gilmore (1996)
Beverly Hills Cop III (1994)
Tommy Boy (1995)
Congo (1995)
Indian in the Cupboard, The (1995)
```

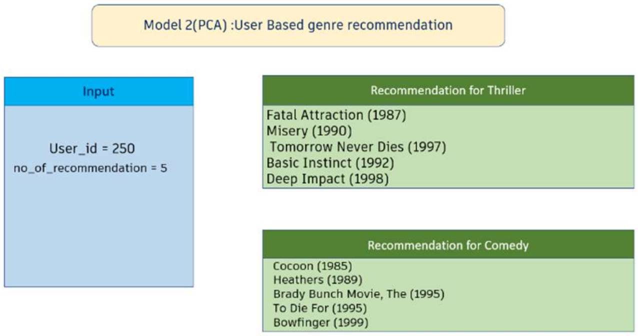
(b) Item based using PCA



User_id = 250
no_of_recommendation = 10

Color Purple, The (1985)
Shawshank Redemption, The (1994)
Adventures of Priscilla, Queen of the Desert, The (1994)
Jane Eyre (1944)
Little Women (1933)
Sound of Music, The (1965)
Dead Poets Society (1989)
Fantasia (1940)
Fatal Attraction (1987)
Erin Brockovich (2000)

(c)User based general


Model 2(PCA) :User Based genre recommendation
Input
Recommendation for Thriller
Recommendation for Comedy

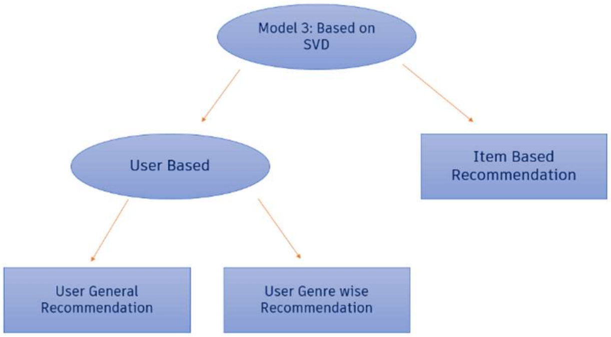
User_id = 250
no_of_recommendation = 5

Recommendation for Thriller:
Fatal Attraction (1987)
Misery (1990)
Tomorrow Never Dies (1997)
Basic Instinct (1992)
Deep Impact (1998)

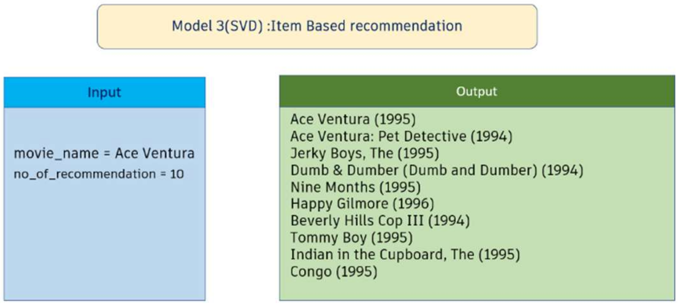
Recommendation for Comedy:
Cocoon (1985)
Heathers (1989)
Brady Bunch Movie, The (1995)
To Die For (1995)
Bowfinger (1999)

(d) User based genre wise

3. Collaborative filtering using SVD


Model 3: Based on SVD
User Based
Item Based Recommendation
User General Recommendation
User Genre wise Recommendation

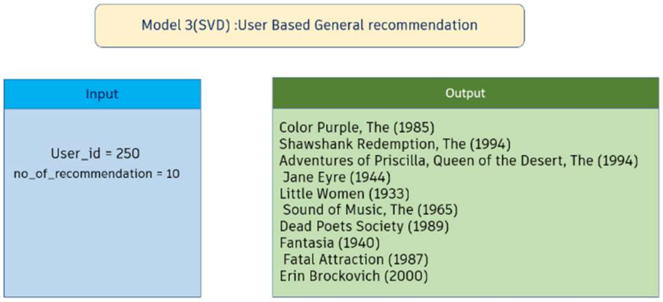
(a) Model



movie_name = Ace Ventura
no_of_recommendation = 10

Ace Ventura (1995)
Ace Ventura: Pet Detective (1994)
Jerky Boys, The (1995)
Dumb & Dumber (Dumb and Dumber) (1994)
Nine Months (1995)
Happy Gilmore (1996)
Beverly Hills Cop III (1994)
Tommy Boy (1995)
Indian in the Cupboard, The (1995)
Congo (1995)

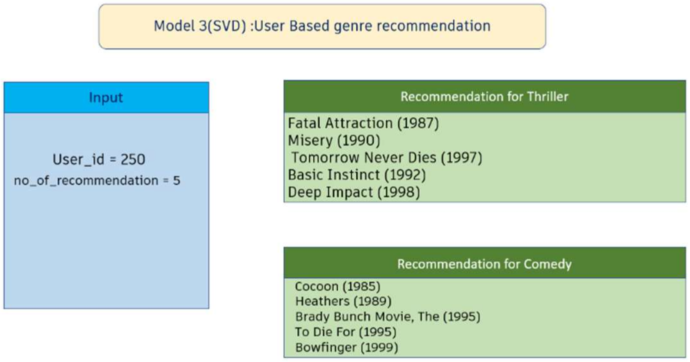
(b) Item Based using SVD



User_id = 250
no_of_recommendation = 10

Color Purple, The (1985)
Shawshank Redemption, The (1994)
Adventures of Priscilla, Queen of the Desert, The (1994)
Jane Eyre (1944)
Little Women (1933)
Sound of Music, The (1965)
Dead Poets Society (1989)
Fantasia (1940)
Fatal Attraction (1987)
Erin Brockovich (2000)

(c)User based general


Model 3(SVD) :User Based genre recommendation
Input
Recommendation for Thriller
Recommendation for Comedy

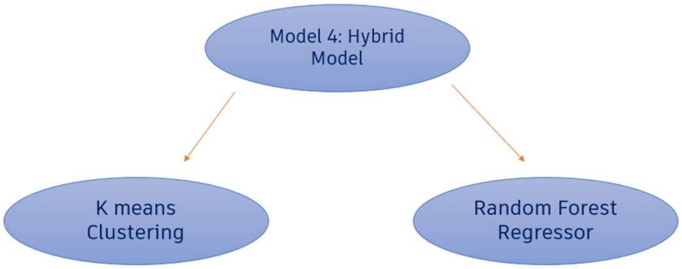
User_id = 250
no_of_recommendation = 5

Recommendation for Thriller:
Fatal Attraction (1987)
Misery (1990)
Tomorrow Never Dies (1997)
Basic Instinct (1992)
Deep Impact (1998)

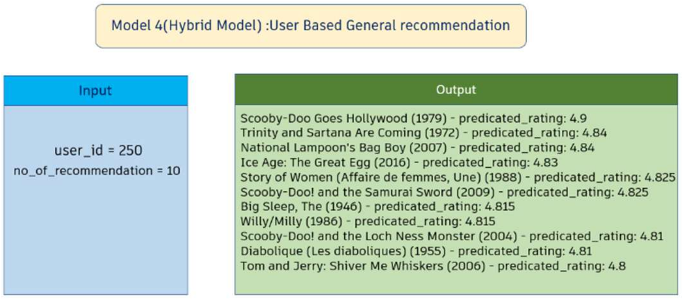
Recommendation for Comedy:
Cocoon (1985)
Heathers (1989)
Brady Bunch Movie, The (1995)
To Die For (1995)
Bowfinger (1999)

(d) User based genre wise

4. Collaborative filtering using Hybrid Model



(a) Model



user_id = 250
no_of_recommendation = 10

Scooby-Doo Goes Hollywood (1979) - predicated_rating: 4.9
Trinity and Sartana Are Coming (1972) - predicated_rating: 4.84
National Lampoon's Bag Boy (2007) - predicated_rating: 4.84
Ice Age: The Great Egg (2016) - predicated_rating: 4.83
Story of Women (Affaire de femmes, Une) (1988) - predicated_rating: 4.825
Scooby-Doo! and the Samurai Sword (2009) - predicated_rating: 4.825
Big Sleep, The (1946) - predicated_rating: 4.815
Willy/Milly (1986) - predicated_rating: 4.815
Scooby-Doo! and the Loch Ness Monster (2004) - predicated_rating: 4.81
Diabolique (Les diaboliques) (1955) - predicated_rating: 4.81
Tom and Jerry: Shiver Me Whiskers (2006) - predicated_rating: 4.8

(b) User based using Hybrid Model

5 Summary

The project presents a **movie recommendation system** leveraging traditional **machine learning models**, with a primary focus on **collaborative filtering techniques** such as **KNN, PCA, and SVD**. Additionally, a **hybrid approach** combining **K-means clustering** and **Random Forest** enhances recommendation accuracy. The system is designed to deliver **personalized movie suggestions** by analyzing user behavior and preferences. Future improvements can incorporate **advanced ML techniques**, such as **neural networks**, to further expand the model's scope and accuracy.

References

[1] Yuliia Kniazieva. Introduction to the movie recommendation system architecture. URL https://labelyourdata.com/articles/movie-recommendation-with-machine-learning.