**Movie Recommendation System**

This repository contains the code for a movie recommendation system using various collaborative filtering and matrix factorization techniques. It includes extensive exploratory data analysis (EDA) and visualizations to better understand the dataset and the recommendation results.

**Table of Contents**

* [Dataset](#dataset)
* [Requirements](#requirements)
* [Exploratory Data Analysis (EDA)](#exploratory-data-analysis-eda)
* [Recommendation Models](#recommendation-models)
  + [User-Based Collaborative Filtering](#user-based-collaborative-filtering)
  + [Item-Based Collaborative Filtering](#item-based-collaborative-filtering)
  + [Matrix Factorization (SVD)](#matrix-factorization-svd)
  + [Content-Based Filtering](#content-based-filtering)
  + [Hybrid Model](#hybrid-model)
* [Evaluation](#evaluation)
* [Visualizations](#visualizations)
* [Usage](#usage)

**Dataset**

The dataset used in this project is the MovieLens 100k dataset. It consists of 100,000 ratings from 943 users on 1682 movies.

**Requirements**

The following Python libraries are required to run the code:

* pandas
* matplotlib
* seaborn
* surprise
* sklearn
* implicit

You can install these libraries using pip:

bash

Copy code

pip install pandas matplotlib seaborn scikit-surprise scikit-learn implicit

**Exploratory Data Analysis (EDA)**

The EDA script includes the following steps:

1. Loading and understanding the dataset
2. Descriptive statistics of the ratings
3. Visualizing rating distributions
4. Visualizing popular movies
5. Visualizing user behavior
6. Correlation analysis

**Recommendation Models**

The following recommendation models are implemented in this project:

**User-Based Collaborative Filtering**

Uses k-nearest neighbors (KNN) to find similar users and recommend movies based on their ratings.

**Item-Based Collaborative Filtering**

Uses k-nearest neighbors (KNN) to find similar items and recommend movies based on item similarity.

**Matrix Factorization (SVD)**

Uses Singular Value Decomposition (SVD) to predict ratings and recommend movies.

**Content-Based Filtering**

Uses TF-IDF vectorization to recommend movies based on their titles.

**Hybrid Model**

Combines collaborative filtering and content-based filtering to provide recommendations.

**Evaluation**

The models are evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics.

**Visualizations**

The script includes visualizations for:

1. Rating distribution
2. Top 10 most rated movies
3. Distribution of the number of ratings per user
4. Distribution of average ratings given by users
5. Top 10 recommendations for each model

**Usage**

To run the code, follow these steps:

1. Clone the repository:

bash

Copy code

git clone https://github.com/yourusername/movierecommendation.git

cd movierecommendation

1. Install the required libraries:

bash

Copy code

pip install -r requirements.txt

1. Run the Jupyter Notebook or Python script to perform EDA and generate recommendations.

**Example Code**

**EDA and Visualizations**

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load datasets

u\_data = pd.read\_csv('path/to/u.data', sep='\t', names=['user\_id', 'item\_id', 'rating', 'timestamp'])

u\_item = pd.read\_csv('path/to/u.item', sep='|', names=['item\_id', 'title'], usecols=[0, 1], encoding='latin-1')

# Merge datasets to get movie titles

data = pd.merge(u\_data, u\_item, on='item\_id').drop('timestamp', axis=1)

# Rating distribution

plt.figure(figsize=(10, 6))

sns.countplot(data['rating'])

plt.title('Rating Distribution')

plt.xlabel('Rating')

plt.ylabel('Count')

plt.show()

# Top 10 most rated movies

top\_movies = data['title'].value\_counts().head(10)

plt.figure(figsize=(12, 6))

sns.barplot(x=top\_movies.values, y=top\_movies.index, palette='viridis')

plt.title('Top 10 Most Rated Movies')

plt.xlabel('Number of Ratings')

plt.ylabel('Movie Title')

plt.show()

**Recommendation Models**

Refer to the detailed implementation in the script for each recommendation model.

**Evaluation and Visualizations**

Refer to the detailed implementation in the script for evaluation and visualizations.