Cheptoi-Millicent /
DS-PHASE4-PROJECT

Code Issues Pull requests Actions Projects Wiki Security Insights Setting

0 stars 0 forks 1 watching Branches Activity
Tags

Public repository

1 Branch 0 Tags Go to file t Go to file Add file Code

Cheptoi-Millicent Presentation-pdf 826ca71 · 6 minutes ago

.gitignore	Initial commit	2 days ago
Notebook.ipynb	ipynb file	8 minutes ago
README.md	README.md	1 hour ago
movies.csv	Dataset	2 days ago
notebook.pdf	Notebook-pdf	7 minutes ago
presentation.pdf	Presentation-pdf	6 minutes ago
ratings.csv	Dataset	2 days ago

README

DS-PHASE4-PROJECT

Overview

This project aims to develop a Collaborative Filtering-based Movie Recommendation System that helps users discover movies tailored to their preferences. The system achieves this by analyzing user ratings and recommending the top 5 movies that align with their viewing history and preferences. Additionally, the system predicts the rating a user would likely give to a movie they haven't rated yet, ensuring more accurate recommendations.

To measure the effectiveness of the system, key evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are used.

Problem Statement

The objective of this project is to build a movie recommendation system that help the user to find the right item by minimizing the options since all entertainment websites or online stores have alot of items. It becomes challenging for the customer to select the right one.

Objectives

The objectives:

- To create a Collaborative Filtering based Movie Recommendation System.It provides top 5 recommendations to a user, based on their ratings of other movie.
- Predict the rating that a user would give to a movie that he has not yet rated.
- Minimize the difference between the predicted and actual rating (RMSE and MAE).

Data

The data i worked with was accessed throught the link:(https://grouplens.org/datasets/movielens/latest/), the focus for this project being the movies and the ratings csv's

The source of the data

- MovieLens Dataset to an external site (Grouplens)

Description of data

- Movies data - the datatypes include: int64(1), object(2)
- Ratings data - the datatypes include: int64(2), float64(1)
- Data Manipulation, the module used include: pandas and numpy
- Data Visuaalization, the module used include: seaborn, matplotlib
- Modelling, the modules were accessed from the surprise they include: Reader,Dataset,accuracy,train_test_split,cross_validate,GridSearchCV,KNNBasic, KNNBaseline, SVD, mean_squared_error.
- Algorithms used: The modules were accessed from surprise they include KNNBasic,KNNBaseline and the Matrix Factorization-based algorithm: SVD

Loading of the Data

- First, i had to connect to my google drive, the load the data using the file path, which refers to the google drive as well as the folder where the data is located.

file_path = "/content/drive/MyDrive/Data" movies = pd.read_csv(file_path + "/movies.csv") ratings = pd.read_csv(file_path + "/ratings.csv")

- Feature Engineering, by drooping the timestamp column
- Merging the two datasets onto one usibg the common column the "movieid": data = pd.merge(ratings, movies, on="movieId")

Data Preparation

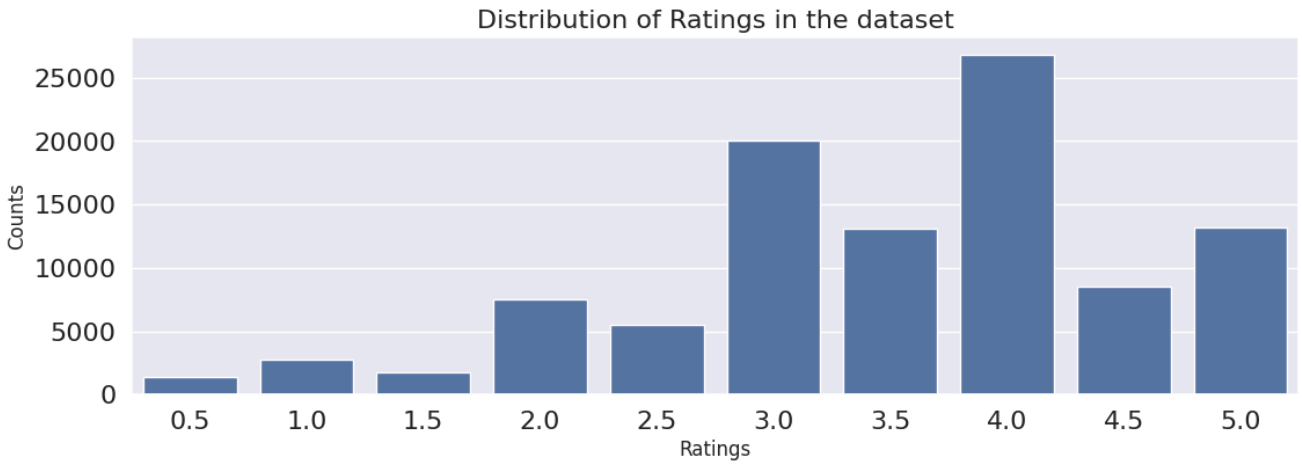
Cleaning of Data

The data did not have any missing values or duplicates, so no cleaning was carried out for this specific data.

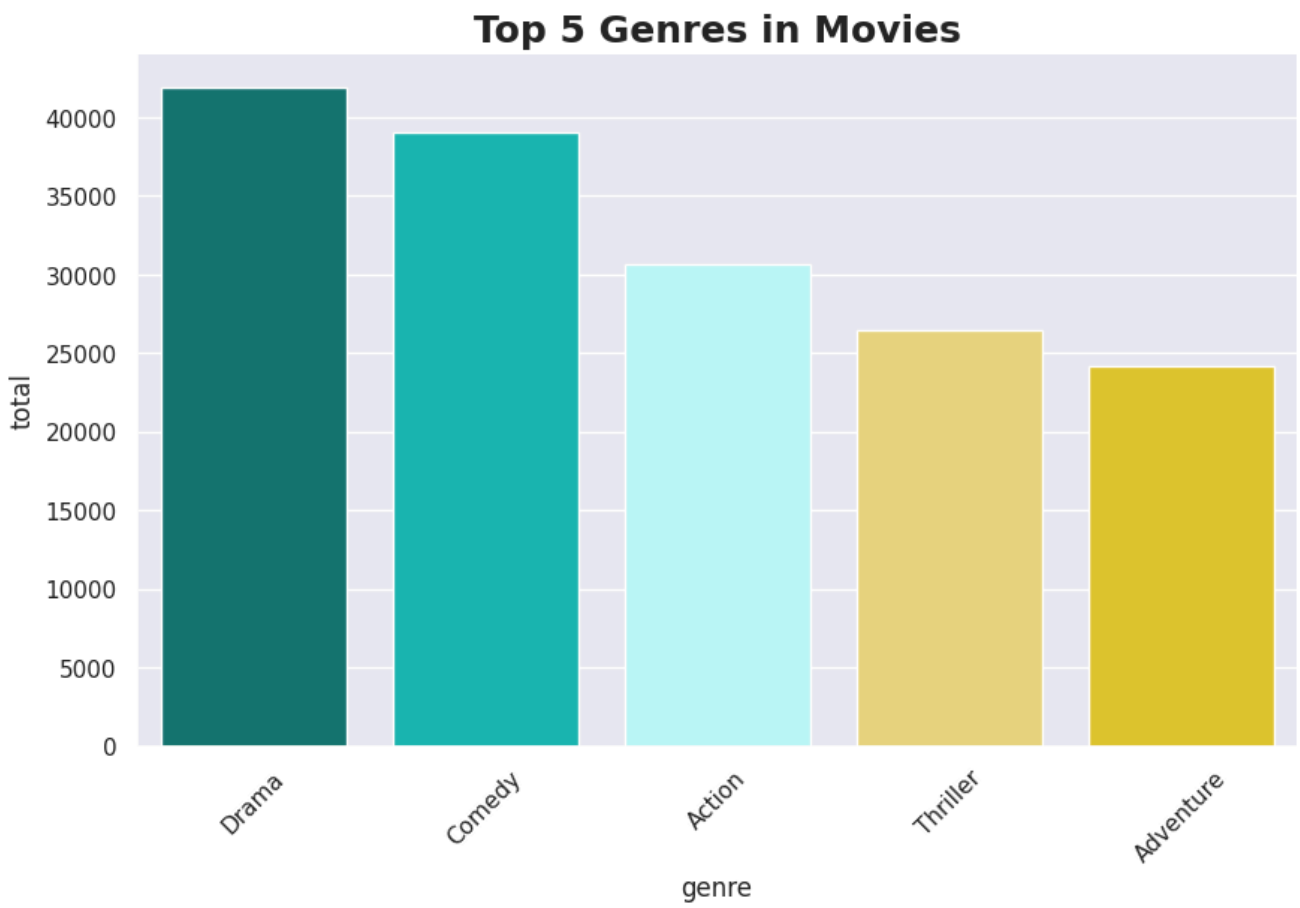
Exploratory Data Analysis(EDA)

Univariate Analysis

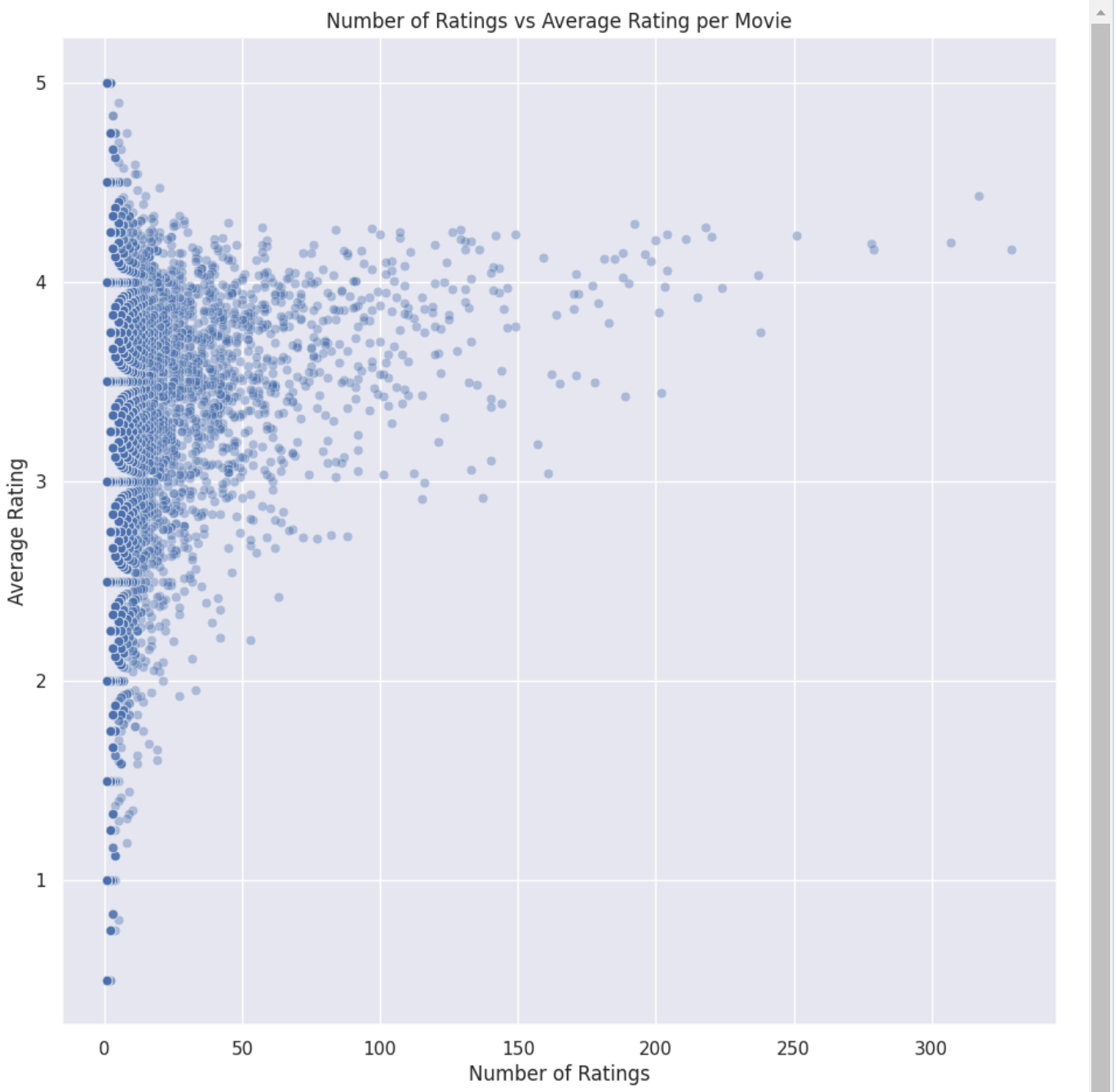
- Checking the "movieID Feature
- Checking the "userID" Feature
- Checking the "title" Feature
- Visualization of the "Rating" feature



- Visualization of the "Genre" Feature

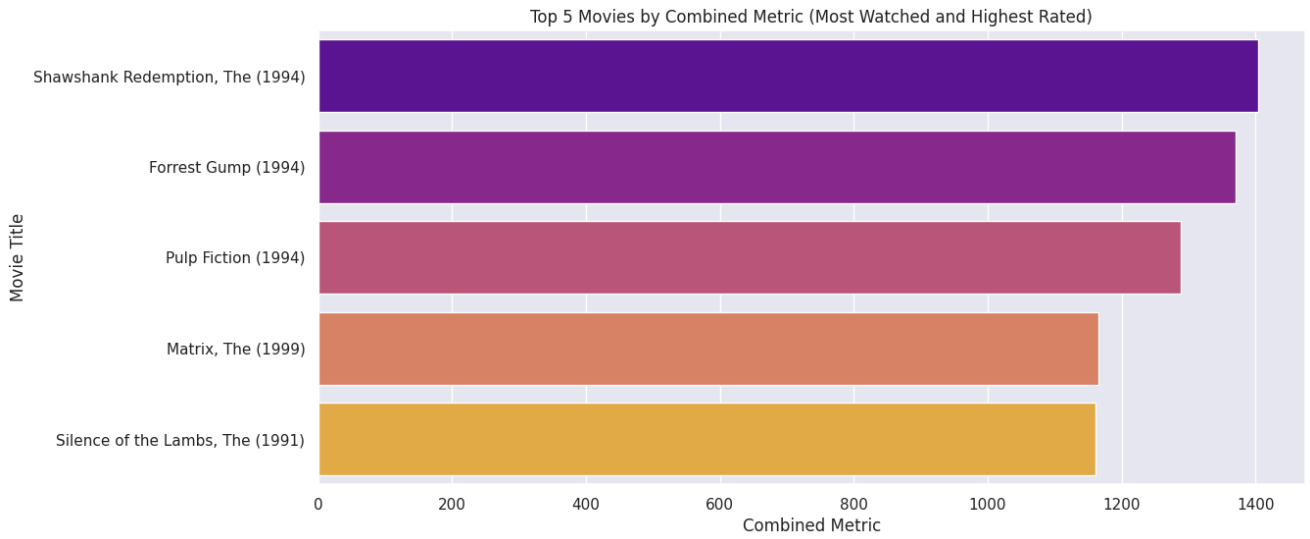


- Visualization of the relationship between the number of ratings a movie got and the average rating.



Bivariate Analysis

- Visualization of the top 5 movies with the best combined score



Modelling

- The surprise library to be used in the modelling process.
- The algorithms used include: (KNN basic algorithm), (KNNBaseline algorithm), the Matrix Factorization_based algorithm(SVD).
- During implementation the SVD model to be used in the recommendation system.
- Convertion of the dataframe to a surprise dataset.
- Splitting the dataset: train_set(size 80) and test_set(size=20)

A baseline memory-based model (KNN basic)

- User-Based (user_based=True): Finds users with similar rating patterns and Predicts a movie's rating for a user by averaging ratings from similar users.
- Similarity metrics used, cosine similarity to measure the similarity between users

KNNBaseline

- It improves upon KNNBasic by incorporating baseline estimates to address biases in user and item ratings.
- Item-Based (user_based=False): Finds similar movies based on their bias-adjusted rating patterns and Predicts a rating based on how a user rated similar movies.
- When working with moderate-sized datasets and when interpretability is important

SVD

- It is one of the most popular and effective algorithms for handling large and sparse recommendation datasets.
- Used when better accuracy is required and when capturing hidden patterns in user behavior

Model Evaluation

- The key focus being the accuracy which is the RMSE and MAE. They both vary for the three models, KNNBasic(RMSE - 1.0100 and MAE - 0.7773), KNNBaseline (RMSE - 0.9758 and MAE - 0.7397) , SVD (RMSE - 0.6448 and MAE - 0.4996)
- KNNBasic only finds similar users/items but doesn't adjust for rating biases.
- KNNBaseline improves accuracy by accounting for user/item biases.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%