mkiganane20 / **Group-5_Phase_Four_Project**

`<> Code`   Issues 1   Pull requests   Actions   Projects   Wiki   Security   Insights   Sett

☆ **0** stars   **0** forks   ⊙ **0** watching   Branches   Activity
Tags

🌐 Public repository

1 Branch   **0 Tags**   🔍 Go to file t   Go to file   +   Add file ▾   Code   ⋯

🔶 **mkiganane20** Group_5_Phase_4_Project_Notebook.ipynb ⋯		40a6c5f · now 🕐
📄 Group_5_Phase_4_Project_No...	Group_5_Phase_4_Project_Notebook...	46 minutes ago
📄 Group_5_Phase_4_Project_No...	Group_5_Phase_4_Project_Notebook...	41 minutes ago
📄 Group_5_Phase_4_Project_Re...	Group_5_Phase_4_Project_Repositor...	37 minutes ago
📄 Presentation.pdf	Added the Presentation and Update...	yesterday
📄 Presentation.pptx	Added the Presentation and Update...	yesterday
📄 README.md	README.md	4 minutes ago

📖 README   ✏️ ☰

MovieLens Analysis and Prediction

Overview

This project explores the development of a personalized movie recommendation system using the MovieLens dataset, leveraging advanced supervised learning techniques to solve a regression problem — predicting user ratings for movies. The core business problem is to improve content discovery and user engagement on a movie streaming platform by offering accurate, data-driven movie recommendations.

Key Objectives

- Which model (SVD or KNNBasic) gives the most accurate movie recommendations?
- What is the optimal number of latent factors (n_factors) to achieve the lowest error in SVD?
- How do different KNN settings (k-values and similarity metrics) impact prediction accuracy?

Business Problem

Can we leverage user ratings and tagging behavior to improve personalized movie recommendations, thereby increasing user engagement and satisfaction for a movie recommendation platform?

Data

The dataset from [MovieLens](#), a movie recommendation platform, captures user activity in the form of 5-star ratings and free-text tags. It includes 100,836 ratings and 3,683 tag entries for a total of 9,742 movies. These interactions were recorded by 610 randomly selected users between March 29, 1996, and September 24, 2018. The dataset itself was compiled on September 26, 2018. Only users who had rated at least 20 movies were included. The dataset is organized into four CSV files: ratings.csv, movies.csv

Process and Workflow

The project uses Python and tools such as pandas,numpy, scikit-learn, Surprise, matplotlib, and seaborn for data handling, modeling, and visualization.

Data Preparation: Merging multiple CSVs into a relational structure, cleaning and encoding categorical features, and creating user-item interaction matrices.

Exploratory Data Analysis : Looking further into the data including distribution, average ratings, and Top ten Movie ratings. With the Mean global rating being 3.5, the Average number of ratings per user is 165.3, and `Shawshank Redemption` , `The Godfather` , and `The Usual Suspects` are the most highly rated movies.
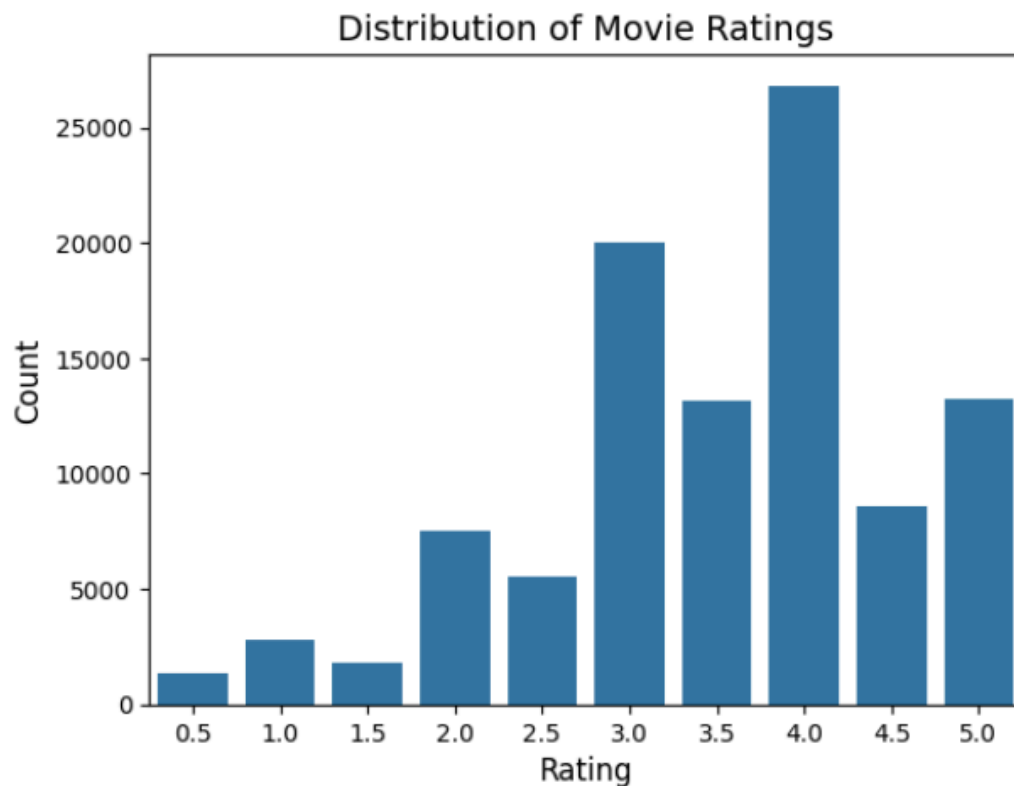

The distribution shows that ratings tend to be positive with most lying from 3.0 to 4.0

Modeling Approach: For collaborative filtering, the Surprise library's KNNBasic algorithm was used, while cosine similarity was employed for content-based filtering on genres and tag relevance.

Model Evaluation: Model performance was validated using train-test split and cross-validation, with evaluation metrics including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Precision@k. After tuning, the SVD model achieved the lowest RMSE and MAE—both on the test set and in cross-validation. While KNNBasic and KNNBaseline delivered similar results, neither surpassed SVD. Because lower error scores indicate more accurate predictions, we selected SVD as our final model.

Recommendation Model Analysis
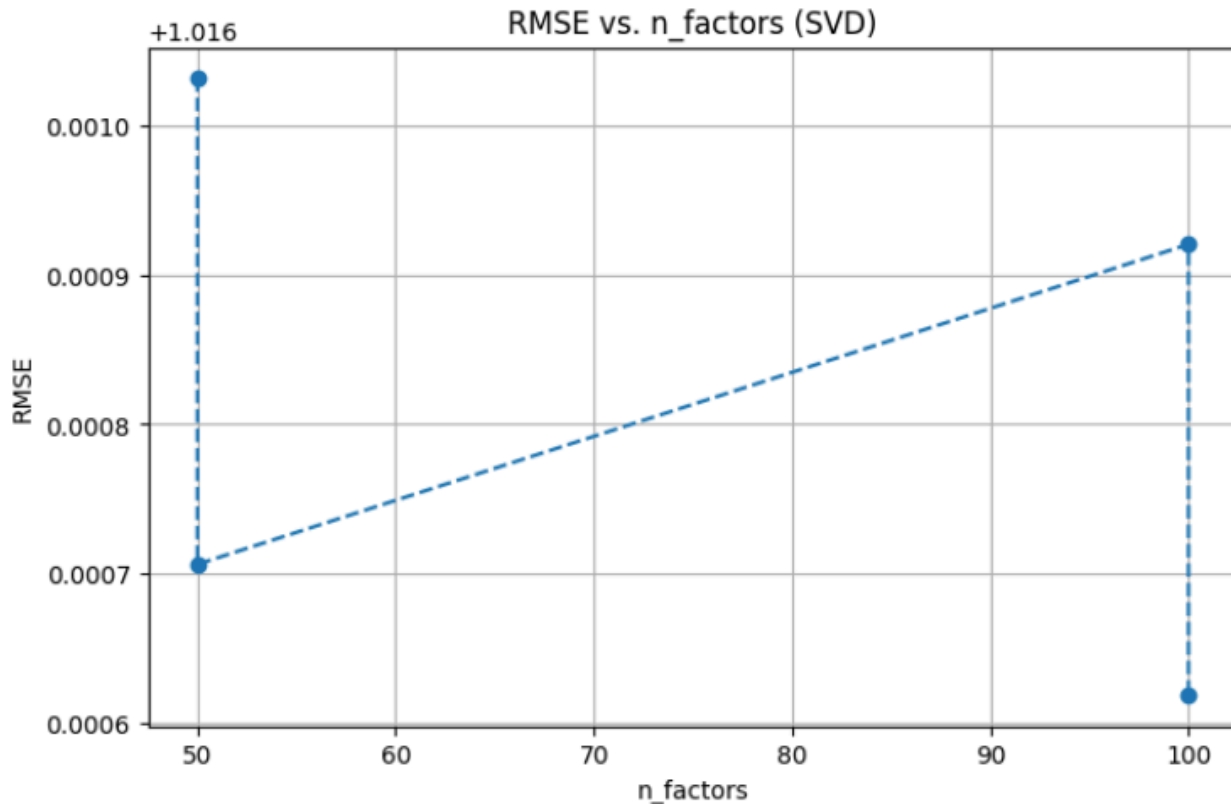
This graph shows that SVD performs slightly better, with lower error scores. Lower scores are better, therefore SVD is the preferred model for predicting movie ratings accurately.

Comparison of SVD and KNNBasic (Lower is Better)
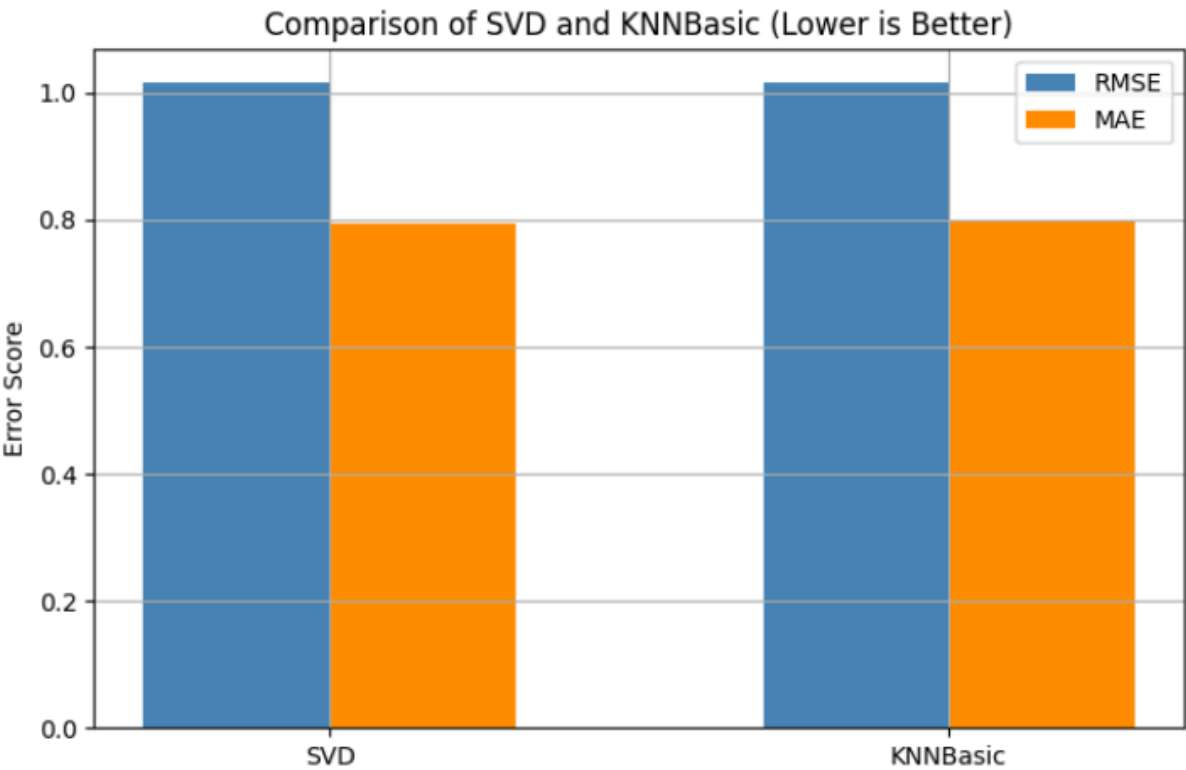
The graph shows how RMSE changes with different n_factors in the SVD model.Lowest RMSE occurs at n_factors = 50, meaning best performance.

Findings and Recommendations

This project showcases a scalable and practical recommendation system tailored for movie platforms. First. The

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

mkiganane20 Myrajoy Kiganane

Tintin-dataObsessed Michelle Chekwoti

Joshport1

Languages

- ● **Jupyter Notebook** 100.0%