1. Introduction

The e-commerce landscape is highly competitive, and providing personalized experiences has become vital to retaining customers and boosting sales. Recommendation systems are at the core of this personalization, as they provide users with product suggestions that align with their preferences and behavior. This project aims to build a personalized recommendation system for the Flipkart GRiD 5.0 coding challenge.

Problem Statement: The aim is to enhance user experience by implementing a personalized product ranking system. Your task is to develop an algorithm or model that can generate accurate and relevant product rankings for individual users. The ranking system should consider factors such as user preferences, past interactions, product popularity, and user similarity. It should be able to predict the most suitable products for a user based on their unique characteristics and preferences.

2. Data Loading and Preprocessing

2.1 Data Loading

- **Dataset Description:** The dataset contains Amazon Clothes, Shoes, and Jewelry review data, including user IDs, product IDs, and ratings.
- **Loading Process:**
  - The data is loaded using the Pandas library.
  - It includes two main parts: review data and product metadata.
  - The review data contains user IDs, product IDs, and ratings, while the metadata provides additional information about the products.

2.2 Data Cleaning

- **Handling Missing Values:** Missing values in the review data are handled by dropping the corresponding rows.
- **Outliers:** Ratings outside the range of 1 to 5 are capped to fit within this range.
- **Conversion to Surprise Format:** The data is converted into a format suitable for the Surprise library, which is used for building recommendation models.

3. Model Building and Training

3.1 Algorithms Used

- **SVD (Singular Value Decomposition):** A matrix factorization technique used to decompose the user-item interaction matrix into multiple matrices representing latent factors.
- **KNN-Based Algorithms:** Various K-Nearest Neighbors (KNN) algorithms are used, including:
  - **KNNBasic:** A basic collaborative filtering algorithm.
  - **KNNWithMeans:** Similar to KNNBasic but takes the mean ratings of each user into account.
  - **KNNWithZScore:** Takes the z-score normalization of each user into account.

3.2 Hyperparameter Tuning

- **Grid Search:** A systematic search through a specified parameter grid is performed to find the optimal hyperparameters for the SVD algorithm.

- **Parameters Tuned:**
    - Learning rate
    - Regularization terms
    - Number of factors

3.3 Training Process

- The best algorithm (SVD with optimal hyperparameters) is trained on the data.
- The trained model is saved as a pickle file for future use, allowing for quick loading and prediction.

4. Evaluation and Recommendations

4.1 Evaluation Metrics

- **RMSE (Root Mean Square Error):** Measures the differences between the predicted and actual ratings. It provides an overall accuracy metric for the predictions.
- **Precision and Recall:** Used to evaluate the relevance and completeness of the recommendations.

4.2 Recommendation Retrieval

- The trained model is used to predict ratings for unseen user-item pairs.
- The top N recommendations for each user are retrieved based on these predictions.

5. Results and Evaluation

- **RMSE:** The final trained model achieved an RMSE, indicating a good fit between the predicted and actual ratings.
- **Precision and Recall:** The model achieved precision and recall values demonstrating the relevance and completeness of the recommendations.

6. Implementation Details

- **main.py:** The main execution script where data loading, model training, evaluation, and recommendation retrieval are orchestrated.
- **app.py:** Contains the application for serving the recommendations through a web interface. It includes endpoints for fetching recommendations and interacting with the system.

7. Debugging and Improvements (Conversation Files)

- The conversation files detail the iterative process of building, debugging, and improving the recommendation system. Key areas of focus include:
    - Enhancing the recommendation algorithm
    - Debugging issues related to data loading and preprocessing
    - Implementing additional features and improvements

8. Conclusion

The project successfully implemented a personalized recommendation system that met the requirements of Flipkart GRiD 5.0 coding challenge. By leveraging various algorithms, careful preprocessing, and tuning, the system is capable of providing relevant and accurate product recommendations.

9. Future Work

Potential areas for future enhancement include:

- Incorporating additional user and item metadata to create hybrid recommendations.
- Exploring deep learning approaches for more complex modeling.

10. References

- Surprise library documentation: Link
- Dataset: Amazon Review Dataset for Clothing, Shoes, and Jewelry