Recommendation System Project

Institute: Jio Institute
Submission Date: 6 January 2025
Professor: Ashish Tendulkar
Teaching Assistant: Nikita Divay

Group Name: The 5 Astronauts

Members:

- Abhishek Sahoo
- Ayush Chakraborty
- Nishant Pandey
- Vikram Pathak
- Yash Joglekar

---

1. Introduction

Problem Definition

The vast number of products available on e-commerce platforms like Amazon often leads to information overload for users. Recommender systems aim to alleviate this problem by providing personalized suggestions based on users' preferences and behavior. This project develops a recommendation system that suggests Amazon Beauty products to users based on historical reviews. The system leverages collaborative filtering algorithms to predict ratings and improve product discovery.

Project Objective

The goal of this project is to develop a recommendation system that can accurately suggest products to users based on their preferences, which are inferred from past reviews and ratings. The system's performance is evaluated based on RMSE (Root Mean Squared Error), which measures the prediction accuracy compared to actual user ratings.

Importance of Recommendation Systems in E-commerce

In modern e-commerce, recommendation systems have become an integral part of enhancing user experience. By suggesting products that a user might be interested in, these systems increase user engagement, conversion rates, and sales. This project focuses on the beauty product category, providing users with personalized product recommendations, improving product discoverability and satisfaction.

---

2. Dataset Description

Overview of Dataset (Amazon Beauty Product Reviews)

The dataset used in this project is a subset of Amazon's product review data, specifically the "5-core" dataset, which contains reviews from users who have made at least five entries. This dataset includes reviews and associated metadata for beauty products on Amazon. The data was sourced from the following link: Amazon Review Data.

Key Dataset Attributes

- **reviewerID**: A unique identifier for each user who left a review.
- **asin**: A unique identifier for each product.
- **overall**: The rating given by the reviewer (ranging from 1 to 5).
- **reviewText**: The content of the review, which was not used in this particular project.

The dataset comprises:

- **198,502 reviews** from **22,363 unique users**.
- **12,101 unique products**.

Data Cleaning and Preprocessing

The dataset was preprocessed to remove duplicates and missing values. The key columns used for building the recommendation system were:

- **reviewerID**
- **asin**
- **overall**

These columns were formatted and loaded into the Surprise library, which is used for building recommendation models. After filtering for users with at least five reviews, the dataset was split into training and test sets.

---

3. Methodology

Collaborative Filtering

Collaborative filtering is the core methodology used in this project. It assumes that users who agreed in the past will agree in the future, making it ideal for recommending products based on similar users' preferences. We used two types of collaborative filtering models:

1. **User-based collaborative filtering**: Recommends products that similar users have liked.
2. **Item-based collaborative filtering**: Recommends products that are similar to those a user has liked before.

Matrix Factorization (SVD, SVD++)

Matrix factorization methods, such as Singular Value Decomposition (SVD), are used to break down the user-item interaction matrix into smaller, dense matrices that can predict ratings for unseen products. These methods capture hidden factors that explain patterns in user behavior.

- **SVD**: A basic matrix factorization technique that factors the user-item matrix into latent factors.
- **SVD++**: An extension of SVD that incorporates implicit feedback (like clicks and views) along with explicit ratings.

Model Evaluation Metrics

- **RMSE (Root Mean Squared Error)**: The primary evaluation metric used to assess the prediction accuracy of the model. It measures the difference between predicted and actual ratings. A lower RMSE indicates better predictive accuracy.
- **Cross-Validation**: We performed 5-fold cross-validation to ensure that our models generalize well and avoid overfitting.

---

4. Implementation

Data Preprocessing

The dataset was first loaded into the Surprise library, which provides tools for building collaborative filtering models. Data preprocessing steps included:

- Removing null values and duplicates.
- Splitting the data into training and test sets.
- Converting data into the format required by the Surprise library.

Collaborative Filtering Implementation

We implemented three models:

1. **NormalPredictor**: A baseline model that randomly predicts ratings based on the average rating.
2. **BaselineOnly**: A model that uses the baseline estimate (user and item biases) to predict ratings.
3. **SVD and SVD++**: Matrix factorization models to capture latent factors for user-item interactions.

Model Training

Each model was trained on the training set using the Surprise library. Hyperparameters such as the number of factors, epochs, and learning rate were optimized using grid search to find the best-performing configuration.

Hyperparameter Optimization

We performed grid search to find the optimal hyperparameters for the models, particularly focusing on the number of latent factors and the number of epochs.

---

5. Evaluation

Model Evaluation Using RMSE

The performance of the models was evaluated using RMSE, which measures how closely the predicted ratings match the actual ratings from users. We calculated the RMSE for each model using the test set.

Cross-Validation Process

We used 5-fold cross-validation to evaluate the models' performance on multiple splits of the dataset, ensuring the results were robust and not overfitted to a single partition of the data.

Benchmark Comparison

We compared the performance of several models, including SVD, SVD++, BaselineOnly, and NormalPredictor, based on RMSE and fit time. The results helped identify which model performed best in terms of both accuracy and efficiency.

---

6. Results

Model Performance (SVD, SVD++, BaselineOnly, NormalPredictor)

The following models were evaluated:

- **SVD**: RMSE = 1.0820
- **SVD++**: RMSE = 1.0880
- **BaselineOnly**: RMSE = 1.0890
- **NormalPredictor**: RMSE = 1.4988

RMSE Comparison Across Models

- **SVD** performed the best with the lowest RMSE of 1.0820, indicating that it provided the most accurate recommendations.
- **SVD++** followed closely behind, but it was slightly less accurate.
- **BaselineOnly** and **NormalPredictor** models performed poorly compared to matrix factorization-based models.

Interpretation of Results

The results showed that SVD was the most effective model for this dataset. The RMSE values for SVD and SVD++ were significantly lower than the baseline models, indicating that matrix factorization techniques captured the latent factors influencing user preferences better than simpler models.

---

7. Conclusion

Summary of Findings

The project successfully developed a recommendation system for Amazon Beauty products using collaborative filtering algorithms. The SVD model achieved the best performance with an RMSE of 1.0820, outperforming both the BaselineOnly and NormalPredictor models.

Model Effectiveness and Improvements

The SVD model provided highly accurate recommendations, suggesting that matrix factorization is a powerful technique for collaborative filtering tasks. Future improvements could involve incorporating additional features like product descriptions and images to further enhance recommendation quality.

Future Work

Future work could focus on integrating content-based filtering, which would allow the system to recommend products based on item metadata, such as product category and description. Additionally, exploring deep learning models, such as neural collaborative filtering, could further improve the accuracy of the recommendations.

---

8. References

1. **Surprise Library Documentation**: https://surprise.readthedocs.io/en/stable/
2. **Amazon Review Dataset**: https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews
3. **Collaborative Filtering: A Review and New Directions** by Yehuda Koren, Robert Bell, Chris Volinsky, IEEE Computer Society.
4. **Matrix Factorization Techniques for Recommender Systems** by Yehuda Koren, Robert Bell, Chris Volinsky, Computer Science Department, Stanford University.