Amazon Recommendation Systems

MIT-DSML-JUNE-2024-C

Date: 10/1/2024

Contents / Agenda

- Business Problem and Data Overview
- Exploratory Data Analysis
- Rank Based Model
- User-User Similarity-based Model
- Item-Item Similarity-based Model
- Matrix Factorization based Model
- Conclusion and Recommendations

Business Problem

In today's fast-paced digital world, information overload has become a significant challenge for consumers, especially on large e-commerce platforms like Amazon. With millions of products available, consumers face difficulty choosing the right product. This is where we see the need for a well-designed recommendation system that helps alleviate this problem by providing personalized suggestions that keep users engaged and make shopping more convenient. Our task is to build such a recommendation system for Amazon, using customer ratings to predict future preferences.

Data Overview

- In order to develop this recommendation system, our approach is to review our dataset of previous ratings given by our customers for a variety of products.
- The dataset contains customer reviews of electronic products, including the following attributes:
    - **userId:** A unique identifier for each user.
    - **productId:** A unique identifier for each product.
    - **Rating:** The rating given by the user to a product.
    - **timestamp:** The time the rating was given (not used in this analysis).

- Initially, the dataset had over 7.8 million entries. To make it computationally feasible and meaningful, we reduced the dataset by selecting users who had reviewed at least 50 products and products that had received at least 5 ratings. After this filtering process, the dataset was reduced to 65,290 entries.

Exploratory Data Analysis

- After filtering, the data contains 65,290 rows and 3 columns (**userId, productId, Rating**)

Data Types:

- userId and productId are categorical variables.
- Rating is a numerical variable (float).

- After filtration, there were no missing values as we made sure to filter our insignificant data points.

Summary Statistics for Rating:

- Mean Rating: 4.29
- Standard Deviation: 0.99
- Min Rating: 1
- Max Rating**:** 5

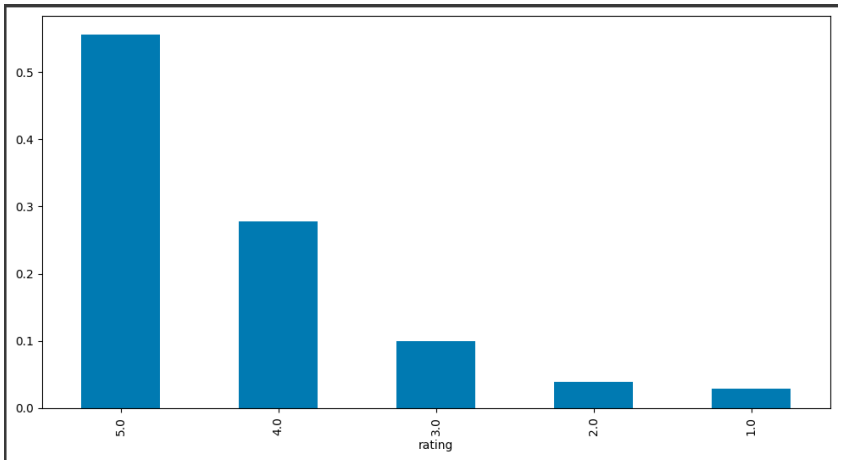
The data shows that most products have relatively high ratings, with a significant concentration around 4 and 5. This suggests that users generally provide favorable feedback on products they purchase.

Exploratory Data Analysis

- The bar plot shows that the majority of ratings are between 4 and 5. This reflects a positive skew in the dataset, indicating overall customer satisfaction with the products they rate.

After the filtration process, we were able to deduct:

- 65290 total observations
- 1540 unique users
- 5689 unique items



Rank Based Model

- **Approach:**
  The Rank-Based Model is a simple baseline model that ranks products based on their average ratings across all users. It recommends the top-rated products to everyone, regardless of individual user preferences.

- **Observations:**
  - The model effectively identifies and promotes products that have received consistently high ratings.
  - However, it lacks personalization and may recommend popular products that a user has already purchased or is not interested in.
  - This model can be useful as a quick recommendation for new users who haven't provided enough data for personalized suggestions.

Rank Based Model

Example:

- In the dataset, a product like *Product A* (productId 1400501466) has an average rating of 4.7 from 500 users. This product will be recommended to all users based on its high average rating, even if some users may not be interested in this specific product type. This approach works well for popular products, but may miss the mark for users with unique tastes.

- **Challenges:**

  - Popularity Bias: The model tends to recommend popular products, which may overshadow niche products that could be more relevant to certain users.

  - Limited Personalization: This approach does not account for the unique preferences of individual users, limiting its effectiveness for personalized recommendations.

User-User Similarity-based Model

Approach:

- The User-User Similarity Model uses collaborative filtering to find users with similar preferences based on their past ratings. The model recommends products to a user that similar users have rated highly.

Default Parameters:

- With default parameters, the model made some good recommendations but often leaned toward more popular items.

Tuned Parameters:

- After tuning the parameters *User X* (userId A1A5KUIIIHFF4U) has rated several electronic products highly, such as *Product B* (rating 5) and *Product C* (rating 4). The model identifies that *User Y* (userId A3LDPF5FMB782Z) has similar ratings for other products and recommends *Product B* and *Product C* to *User Y*.

User-User Similarity-based Model

Observation:

- After tuning the parameters (k = 20, min_k = 6), the model became more precise in finding users with closely matched preferences. For instance, it started recommending niche products like *Product D* (productId B005EOWBHC), which has a lower rating count but fits *User Y*'s interests.

Takeaway:

- The model excels at finding relevant items for users who have rated similar items, but may struggle with users who have rated few products.
- This is important to consider because many users will not have rated a vast variety of products if we are to always depend on their ratings

Item-Item Similarity-based Model

Approach:

- In contrast to the User-User Model, the Item-Item Similarity Model recommends products based on the similarity of products a user has already rated highly. If a user likes a particular product, this model suggests similar products.

Observations:

- **Default Parameters:** Similar to the User-User Model, the default parameters produced relevant recommendations but were more focused on popular products.
- **Tuned Model:** After tuning (k = 20, min_k = 6, and using 'msd' similarity), the model provided more accurate suggestions by finding products that closely matched the user's previous preferences

Item-Item Similarity-based Model

In the model we see that User Z (userId A1E3OB6QMBKRYZ) rated *Product E* (productId B00BB72WX4) highly. The model identifies that *Product F* (productId B00B9AB26G), which shares similar characteristics (same brand, similar features), is likely to be of interest to *User Z*. It thus recommends *Product F* based on item similarity.

Comparison

- The Item-Item Similarity Model generally produced better results than the User-User Model for users who had specific tastes in certain product categories. It was able to recommend items similar to what the user had already shown interest in.

- This shows a capability for more personalized recommendation and therefore would be a better model to use than the User-User Model

Matrix Factorization based Model

Approach:

- Matrix Factorization identifies latent factors that explain the relationship between users and products. This model is particularly effective for sparse datasets where users have rated only a small number of products.

For example, User A (userId A2XIOXRRYX0KZY) has rated only a few products, like *Product G* (rating 5) and *Product H* (rating 3). The matrix factorization model identifies hidden factors based on *User A*'s ratings and predicts that they might also like *Product I* (productId B00B9AB26G), even though *User A* hasn't rated it yet.

This prediction is based on similar latent factors between the rated and recommended products. Examples of these latent factors can be price sensitivity, brand loyalty, interest in tech, etc.

Matrix Factorization based Model

Analysis:

- Because this model takes advantage of latent factors we can gather extremely useful information. Using the previous example we looked at and the latent factors that we discussed we can gather :
    - That *Product I*—a budget-friendly, mid-range electronics product—would align well with *User A*'s preferences, even though they haven't rated it yet by considering such latent factors.

Conclusion

- While matrix factorization models is a powerful model, it also requires a sufficient amount of significant data to accurately infer these "latent factors". Users that do not provide many ratings can pose a challenge because the model has less data to work with.
- However, combining matrix factorization with other techniques, like user-user or item-item similarity models, can help mitigate this issue.

Conclusion

1. **Rank-Based Model:**
   - **Key Finding:** This model ranks products purely by average ratings, which means it recommends the most popular products to everyone.
   - **Finding:** This model doesn't account for individual preferences and may repeatedly recommend products a user has already purchased or isn't interested in.

2. **User-User Similarity-Based Model:**
   - **Key Finding:** This model identifies users with similar preferences and recommends products that those users rated highly.
   - **Finding:** While it provides good personalization, it struggles when users have few ratings or when preferences diverge significantly.

3. **Item-Item Similarity-Based Model:**
   - **Key Finding:** This model finds similar products based on what a user has already rated highly.
   - **Finding:** This model excels when users have clear product preferences, such as favoring specific brands or product categories. It's especially effective for users who shop within a narrow range of products.

4. **Matrix Factorization-Based Model:**
   - **Key Finding:** The most advanced model, matrix factorization, breaks down user-product interactions into latent factors, allowing it to predict ratings even for products a user hasn't rated yet.
   - **Finding**: Handles Sparse data well, provides good quality personalized recommondations, but will struggle if majority of data is of users with few and divergent ratings.

Recommendations

- The **Matrix Factorization Model** should be the model of choice for Amazon's recommendation system due to its scalability, precision, and ability to uncover hidden user preferences through latent factors. This model is especially fit for Amazon's needs because we have the capability to provide an endless amount of meaningful ratings and reviews on a vast array of customers.
- *As additional backup support models in order to provide more meaningful data for the Matrix Factorization model, we can u*se the **Item-Item Similarity Model** for users with strong product loyalty and **User-User Similarity Model** for those with sufficient rating history. The **Rank-Based Model** can serve as a fallback for new users.
- As time passes and we say a wider/different group of customers, regular tuning of **model parameters** (like k and min_k) and **hyperparameters** (such as similarity measures) will play a huge role in maintaining and improving the Matrix Factorization Model for the recommendation system.