Developing Collaborative Filtering Recommendation Systems for Amazon Movies and TV

Problem Identification Overview

Amazon has a vast quantity of items available in the marketplace. Customers can only be aware of and only be exposed to a finite number of items. The principal problem is for Amazon to find a way to introduce new products that those customers may be interested in. This project aims to create a recommendation model that recommends relevant Movies and TV shows individualized for each customer.

Business Impact[1]

Implementing an effective recommendation system for Amazon's products can positively affect the business by providing personalized recommendations. The recommendations will allow customers to more easily discover new content that matches their preferences, leading to increased satisfaction and loyalty. Relevant recommendations can boost sales by promoting products users are more likely to purchase, thereby increasing the average order value and overall revenue. Highlighting a broader range of products can optimize inventory turnover and reduce unsold stock.

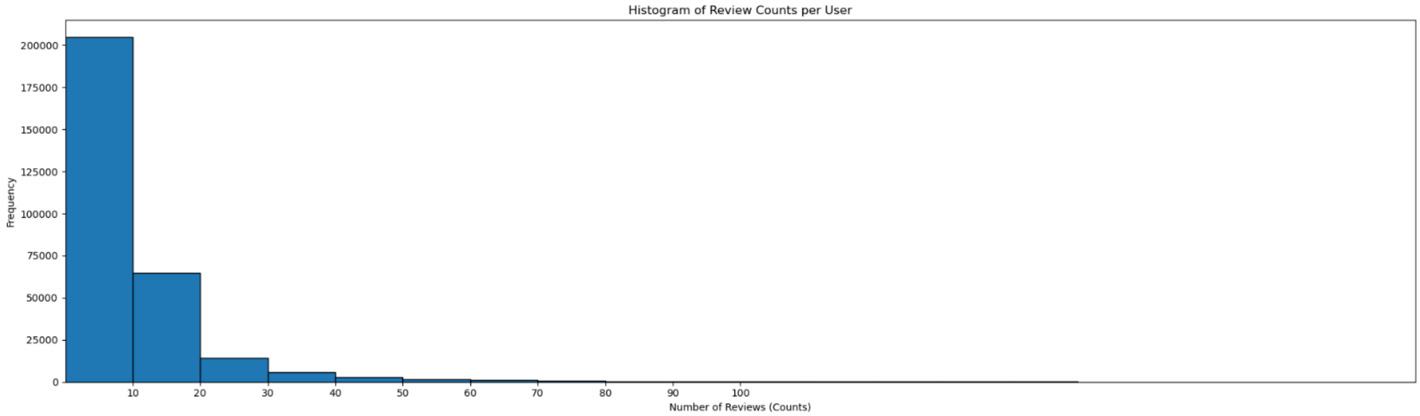
A superior recommendation system will also differentiate Amazon from their competitors, potentially increasing market share by enhancing the customer experience. Understanding customer preferences through recommendation interactions can inform targeted marketing strategies, making them more effective and data driven. Overall, the recommendation system contributes to customer retention, revenue growth, and a stronger competitive position in the market.

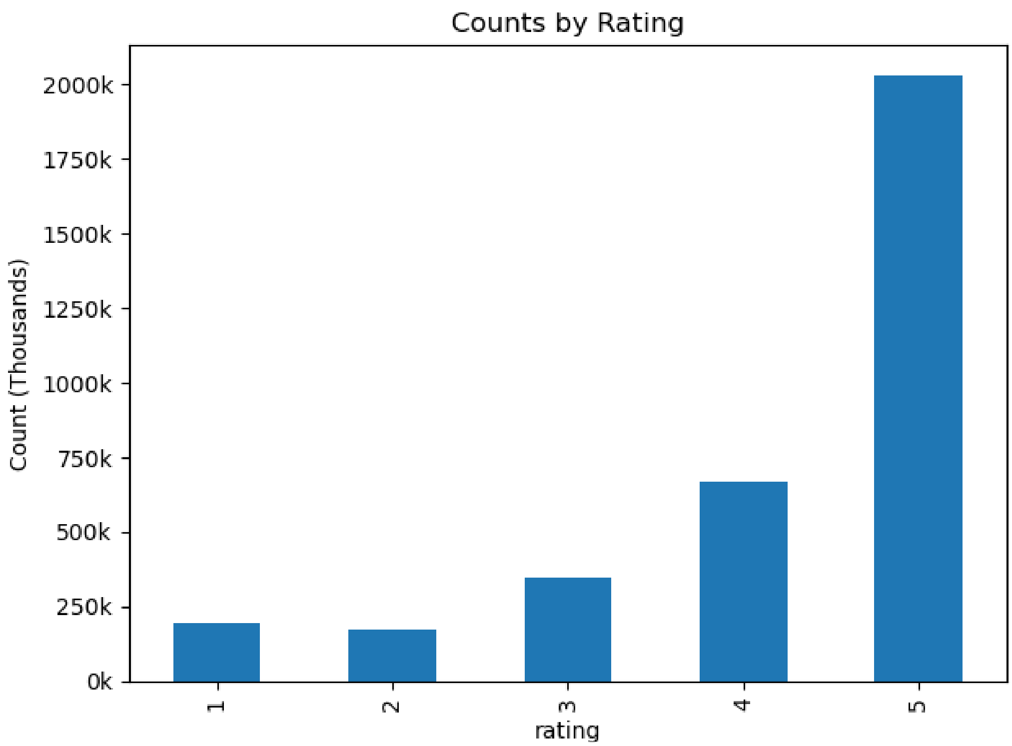
The Data

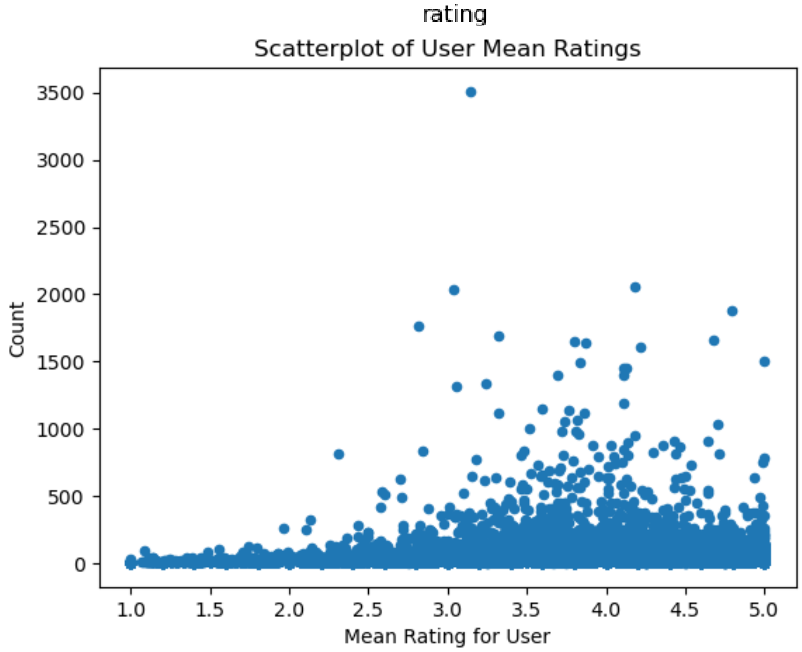
The data used for this project comes from public Amazon review datasets collected by researchers at the University of California, San Diego[2]. The dataset used is the 5-core Movies_and_TV category from the 2018 collection. This dataset contains 3,410,019 reviews from 297,529 users, covering 60,175 unique Movies and TV shows (identified by ASINs). The dataset ensures that each user and each item have at least five reviews, providing sufficient data for collaborative filtering models.

The reviews span from May 1996 through October 2018. The dataset includes several features, but for this project, only three key attributes were used:

- User_ID: A unique identifier for each user.
- ASIN: The Amazon Standard Identification Number for each product.
- Rating: The user's rating of the product, ranging from 1 to 5 stars.


Histogram of Review Counts per User
Frequency
Number of Reviews (Counts)
0
25000
50000
75000
100000
125000
150000
175000
200000
10
20
30
40
50
60
70
80
90
100


Counts by Rating
Count (Thousands)
rating
0k
250k
500k
750k
1000k
1250k
1500k
1750k
2000k
1
2
3
4
5


Scatterplot of User Mean Ratings
Count
Mean Rating for User
0
500
1000
1500
2000
2500
3000
3500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

EDA

Most reviews are 5-star ratings, followed by 5 and then 3-star ratings. One-star reviews are more frequent than two-star reviews, but both are considerably fewer in quantity compared to higher ratings. Analysis indicates that users with more authored reviews tend to have higher average ratings, suggesting a positive correlation between review quantity and more favorable ratings. Most items and users each have less than 10 reviews, but there are a few outliers with thousands of reviews. The most reviews by a single customer is 3509 and their average item rating is 3.14.

Data Preprocessing steps of Note

This dataset had some strong advantages in its favor. Missing values in the key feature were already handled. Normalization for this dataset was also optional and ultimately not performed for this project.

- Data Type Conversions:
  - The user_id and asin features needed to be converted to strings which is required for the Surprise library.
  - The rating feature needed to be converted to floating-point numbers for compatibility with the Alternating Least Squares (ALS) model which requires numerical inputs.
- ID Mapping:
  - Created mappings for user_id and asin to integer indices for compatibility with ALS which requires numerical indexing.
- Train-Test Split
  - Dataset was split into training (80%) and testing (20%) sets.
  - Guaranteed that all user_id and asin were represented in the training and test sets. This is necessary for collaborative filtering models because they cannot make predictions if the user_id or asin are not represented in the test set.

Models

A total of four collaborative filtering models were developed:

- KNN: Chosen because it is a simple but effective algorithm for capturing user or item similarities based on patterns of interactions.
- SVD: Widely used in recommendation systems because of its ability to identify latent factors or patterns.
- ALS: A good model for developing recommendation systems that also handles large datasets efficiently.
- Ensemble:
  - These models can perform well leveraging the strengths of multiple models to improve recommendation accuracy.
  - This ensemble model is an inverse weighted model using the RMSE of the constituent models. This was a simple model to build that gave more weight to the better performing model.

Libraries

1. Surprise: from its own website it describes itself as "a Python scikit for building and analyzing recommender systems that deal with explicit rating data."[3] Surprise handles datasets well and has many ready-to-use prediction algorithms.
2. PySpark: this is a great library to use with ALS and it handes large datasets efficiently.

Model Performance

The models were evaluated using several performance metrics:

- Root Mean Squared Error (RMSE): Measures the average magnitude of prediction errors.
- Precision@10: Proportion of top 10 recommended items that are relevant.
- Recall@10: Proportion of relevant items that are recommended in the top 10.
- F1 Score@10: Harmonic mean of Precision@10 and Recall@10.

Model	RMSE	Precision@10	Recall@10	F1Score@10
KNN	0.922	0.2037	0.9094	0.3329
SVD	0.623	0.2054	0.9107	0.3352
ALS	0.997	0.0013	0.0073	0.0023
Ensemble	0.715	0.2054	0.9107	0.3352

The SVD model performed the best across all metrics, with the ensemble model closely approaching its performance. The SVD model achieved the lowest RMSE, indicating higher prediction accuracy. Both the SVD and ensemble models showed similar Precision@10 and Recall@10 scores, suggesting they are equally effective in recommending relevant items. The KNN model performed reasonably well but was outperformed by the SVD and ensemble models. The ALS model exhibited poor performance, with significantly higher RMSE and substantially lower precision and recall, leading to its exclusion from the final ensemble.

Next Steps

The models have demonstrated their effectiveness, but more should be done before they can be considered for use in a live production environment:

- Hyperparameter Tuning: The models have not undergone extensive hyperparameter tuning due to computational limitations. Exploring different hyperparameter configurations could lead to significant gains in performance. Utilizing more efficient optimization techniques or allocating additional computational resources are recommended.
- Include More Features: Incorporating additional features could improve the performance of the models. Both users and items have extensive metadata that can be leveraged for this. For users,

demographic information, browsing history, and purchase patterns could be valuable. For items, attributes like genre, release date, and price might enhance recommendation accuracy. This would also aid in addressing the Cold Start Problem and allow for content-based filtering models to be developed.

- Expand to All Product Categories: The current models are trained on only the Movies and TV category. Eventually this will need to be expand to include all product categories on Amazon. During this expansion it can be determined whether a single model can handle the diverse categories or if specialized models could be developed. If those specialized models offer performance gains, the business must consider whether maintaining multiple models is justified by the benefits.

Conclusion

The goal of this project, to develop an effective collaborative filtering model, was successful demonstrating the potential to deliver personalized and relevant content to Amazon's customers. The dataset was effective and is well maintained. The Singular Value Decomposition (SVD) model delivered the best performance across key metrics, achieving the lowest Root Mean Squared Error (RMSE) and high Precision and Recall. While the other models were less effective, hyperparameter tuning could increase the effectiveness of all models further.

While these models show promise, the next steps outlined are crucial to the continued development of the model. More features and including content-based filtering models can further enhance the accuracy of and handle the Cold Start Problem for new users and items. Developing and continuously improving and training recommendation systems is a key resource for Amazon to benefit for delivering personalized content, increasing customer satisfaction, and driving revenue growth.

Citations

1. Maruti Techlabs. (2022, January 5). How can product recommendation system benefit your business? *Medium*. https://medium.com/geekculture/how-can-product-recommendation-system-benefit-your-business-65afd9cabfd8
2. **Bridging Language and Items for Retrieval and Recommendation**
   Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, Julian McAuley
   arXiv
3. Hug, N. (n.d.). *Home*. Surprise. https://surpriselib.com/