Collaborative Filtering based Groceries Recommendation System

Hasnain Gandhi
Rutgers University
hg410@scarletmail.rutgers.edu

Mansi Borole
Rutgers University
mb2208@scarletmail.rutgers.edu

Tarini Divgi
Rutgers University
td508@scarletmail.rutgers.edu

ABSTRACT

This paper investigates the application of Collaborative Filtering techniques such as Matrix Factorization, flavors of KNN, Slope One, and Singular Value Decomposition (SVD) in the Gourmet and Grocery food section of the Amazon dataset. As recommendation systems play a crucial role in E-commerce businesses, it is imperative to identify the best recommendations for users based on their preferences. This research aims to provide insights on finding the most appropriate recommendations for users using various techniques in the domain of recommendation systems. Results and findings of the study can be beneficial for E-commerce businesses that utilize Collaborative Filtering to improve customer experience and sales.

KEYWORDS

Recommendation Systems; Collaborative Filtering; Neighborhood-Based Methods; Matrix Factorization; KNN; Slope One; SVD;

1 INTRODUCTION

In today's digital age, the vast amount of data available can be overwhelming for users trying to find the products that best fit their needs. This is where recommendation systems come in, offering a solution by providing personalized recommendations based on user behavior and preferences, while also benefiting the e-commerce businesses by increasing sales. As such, the need for effective recommendation systems has become increasingly important.

To design an effective recommendation system, it is crucial to understand user behavior and preferences. This can be achieved through analyzing user interactions with the system, such as purchase history or product ratings. The challenge, however, lies in generating accurate and personalized recommendations based on this data.

We use various Collaborative Filtering to address this challenge, by analyzing user interactions with the system to identify patterns and recommend items that are similar to what the user has interacted with in the past. Our project focuses on comparing different methods used to generate product recommendations, specifically Collaborative Filtering, Singular Value Decomposition (SVD), K-Nearest Neighbors (KNN), Matrix Factorization, and Slope One. By analyzing the strengths and weaknesses of each method, we hope to gain insights into which method(s) work best for different types of e-commerce businesses and customer preferences.

The paper is divided into 3 parts. We first start by briefly describing the dataset including the feature set and the methods of data exploration and preprocessing. We then explain the different Collaborative filtering methods addressed above. Lastly we discuss the different properties of Recommendation engines learned through the project and future work possible in this domain.

2 DATA PRE-PROCESSING

We based our recommendation on the Grocery and Gourmet foods data. [source] The dataset comprises reviews of products and includes features such as product ID (ASIN), reviewer ID, review text, a summary of the product, review timestamp, and more. Additionally, we are provided with the metadata associated with each product in the Groceries and Gourmet category on Amazon. The metadata file also includes information on other frequently purchased items that are related to the product under consideration.

3 EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is an essential preliminary step in data analysis that facilitates the identification and characterization of relevant features and patterns in our given dataset. Through the use of statistical and visualization techniques, EDA provides a comprehensive understanding of the underlying data distribution, enabling us to detect outliers, data quality issues, and other anomalous behavior. It is also critical in the identification of relationships and dependencies between different variables, which can inform subsequent modeling and analytical decisions. We ask the dataset questions that will help us to understand it more.

(1) Figure 1: The barchart shows that Beverages, Cooking and Baking supplies and Snack foods is the most reviewed category. We can draw conclusions about the items' popularity using this.
(2) Figure 2: We see that there are numerous outliers, some reviewrs making more that 500 reviews. But the vast majority of seem to making 9 reviews each.
(3) Figure 3: Typically an item has about 27 ratings, but this metric tapers down quickly. Some outliers having more than 6000 reviews.
(4) Figure 4: Top 9 items containing fancy teas, coffees, organic candies and chocolate delicacies.

3.1 Database Statistics

Our initial analysis of the reviews dataset lead us to decide the different approaches to data cleaning and selecting the best features as well as reviews. Below we show the initial findings.

- Number of Reviews: 1143860.
- Number of unique reviewers: 127496

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

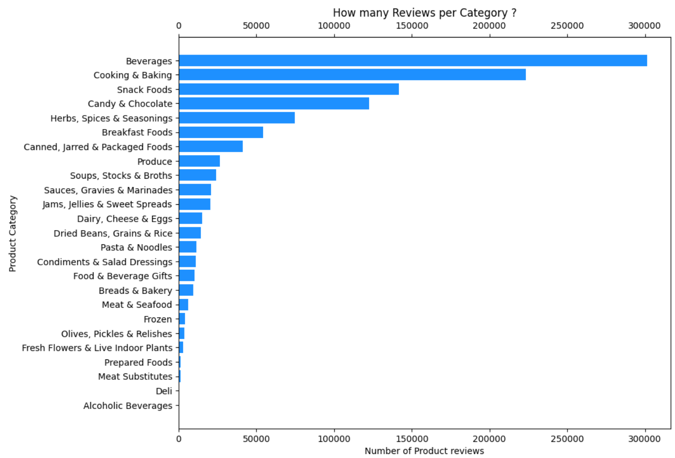

Figure 1: Popularity of various categories

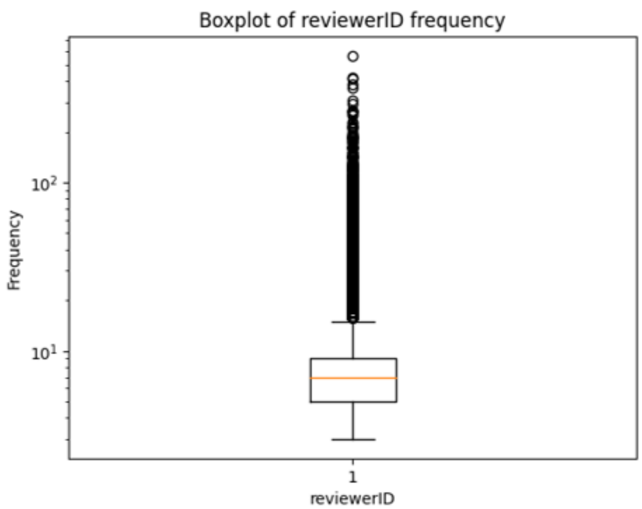

Figure 2: Typically a reviewer makes 9 reviews

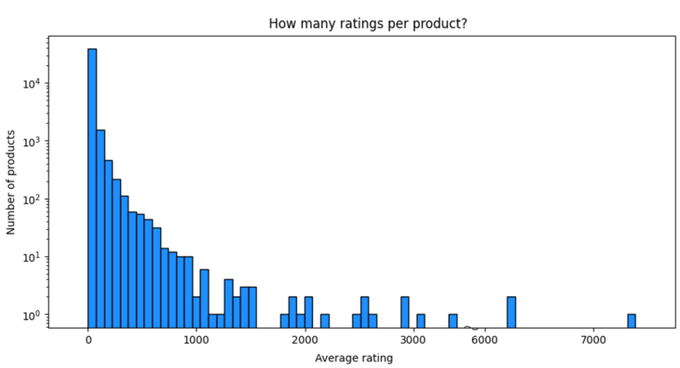

Figure 3: Typically an item has 27 reviews

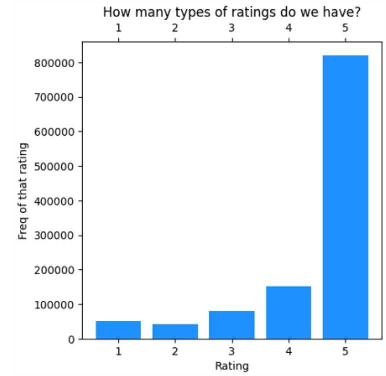

Figure 4: Most common rating is 5*

Figure 5: Top N products being sold

- Number of unique products: 41320
- Rating values: [1,2,3,4,5]

The initial observation of the dataset reveals that the ratings are integers and the User-Item matrix is sparse. This conclusion is drawn based on the considerable difference between the number of reviews and the product of unique reviewers and unique products. To confirm this suspicion, we analyzed the sparsity of the dataframe using the sparse.density function. Our findings indicate a sparsity of approximately 0.8, which indicates a substantial number of NaN values in the matrix. These NaN values correspond to cases where the reviewer did not rate the product. As a result, we were motivated to undertake the following pre-processing steps.

3.2 Data Cleaning and Preparation

The following steps were taken

- Drop columns which have no impact to the rating prediction. These include columns like image of the product, summary of product, Reviewername. Below image shows a few items.
- A lot of reviewers are not verified meaning there could be a bias in those ratings and that could influence the overall predictions. So we remove those reviewers who are not verified by Amazon.

[6]:

	overall	verified	reviewTime	reviewerID	asin	reviewerName	reviewText	summary	unixReviewTime	vote	style
0	5	True	11 19, 2014	A1QVBUH9E1V6I8	4639725183	Jamshed Mathur	No adverse comment.	Five Stars	1416355200	NaN	NaN
1	5	True	10 13, 2016	A3GEOILWLK86XM	4639725183	itsjustme	Gift for college student.	Great product.	1476316800	NaN	NaN
2	5	True	11 21, 2015	A32RD6L701BIGP	4639725183	Krystal Clifton	If you like strong tea, this is for you. It mi...	Strong	1448064000	NaN	NaN
3	5	True	08 12, 2015	A2UY1O1FBGKIE6	4639725183	U. Kane	Love the tea. The flavor is way better than th...	Great tea	1439337600	NaN	NaN

Figure 6: Data Description with columns

- A common observation is that, product reviews tend to be biased towards positive reviews, and that these reviews are authored by a small number of infrequent reviewers. These are not regular reviewers, and their ratings would be either very critical or highly generous. So we now remove reviewers who have reviewed less than 20 products. This step was undertaken to ensure that our analysis is based on a more representative subset of reviewers who have reviewed a substantial number of products.
- Now for the User-based filtering, we group by the rows based on unique reviewers and make the dataframe such that, each reviewer is mapped to all the items they have reviewed and rating in different columns.
- Lastly based in this dataframe, we split the data such that for each reviewer we take 80 percent of the entries in the training set and the remaining in the testing set. The image below shows the structure of database.

[270]:

reviewerID	asin	overall
A10030KC6GYK89	[B0009VFDEI, B0009VFDEI, B00117YT4Y, B00117YT4...	[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 5]
A100RH4M1W1DF0	[B001NJJOCW, B003ZRXRIC, B003ZXCA2U, B003ZXCAA...	[5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 4, 5, 5]
A100UD67AHFODS	[B001E5E3JY, B001XSMANI, B00474ASJI, B004OLKF6...	[5, 5, 4, 5, 5, 1, 5, 5, 4, 5, 4, 5, 5, 5, 5, 5]
A100WO06OQR8BQ	[B00099XOQO, B000E1HUVC, B000H7ELTW, B000H7ELT...	[5, 5, 5, 5, 5, 5, 5, 5, 5, 2, 5, 5, 5, 1, 5, ...
A102JNFLL0KW7I	[B000E1FZHS, B000FFIL92, B000FOIYS6, B000U0OUP...	[5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5]
...	...	...
AZUUU81LB0NYV	[B0001LO3FG, B000F4DKAI, B000GG0BQ6, B000GG0BN...	[3, 3, 5, 5, 4, 4, 4, 4, 5, 4, 3, 4]
AZV26LP92E6WU	[B0009AFWZE, B000E1FZHS, B000E1FZHS, B000E1HUV...	[5, 5, 5, 3, 5, 5, 5, 5, 5, 5, 5, 3, 5, 5, 5, ...
AZWCQL30IJZR8	[B000Z978SS, B000Z978SS, B001K2KR46, B002863BI...	[5, 5, 5, 5, 5, 3, 5, 5, 5, 5, 5]

Figure 7: Dataframe use for train-test split

4 MODELS FOR RATING PREDICTION

In this section, we describe all the models used for comparative analysis of popular CF-based methods.

4.1 Baseline Predictors

Collaborative Filtering methods attempt to calculate a predicted rating for an item and a predictee user by modelling the user-item interactions [1]. However, many observed ratings could be the result of factors external to user-item interactions – they could be the result of inherent biases within the item and user, which are independent of user-item interaction. These biases are taken into account by using baseline predictors. For a given user u, item i and the corresponding rating r_{ui}, the baseline prediction b_{ui} is given by –

$$b_{ui} = \mu + b_i + b_u \tag{1}$$

Where, μ is the overall average rating. b_u and b_i are parameters which represent the observed deviations of user u and item i from the average. These parameters can be estimated using the Least Squares method with Regularization –

$$\min_{b*} \sum_{(u,i)\epsilon\chi} (r_{ui} - \mu - b_i - b_u)^2 + \lambda(b_i^2 + b_u^2) \tag{2}$$

where χ is the set of all user-item pairs (u, i) with known ratings in the training set.

4.2 K Nearest Neighbour

Collaborative methods are usually based on past interactions recorded between users and items in order to produce new recommendations. This memory based approach is essentially based on nearest neighbours search. The gist of the approach is that for a target user, we create clusters based on other users or items to suggest new items. The most common are the User-User and Item-Item systems.

4.2.1 User based CF. This method Tries to identify users with the most similar 'Interaction Profile', meaning the items purchased by user cluster are similar and our user could be interested in those products. From figure 9 we see that users are similar profiles based on their purchase of chocolates and ice-creams and we now suggest John donuts and ice cream.

4.2.2 Item based CF. The idea of item-item method is to find items similar to the ones the user already "positively" interacted with. Two items are considered to be similar if most of the users that have interacted with both of them did it in a similar way. From the figure 9 we see that as the sweets are part of same cluster and hence we can see John is recommended ice-cream scoops.

4.2.3 Implementation details. We propose the use of different KNN based algorithms like KNN Basic, KNN with Means and KNN with Baseline for the rating prediction. There is some difference between each of them.

(1) KNN basic: Basic approach which calculates the cosine/pearson similarity between different users/items. The predicted rating for an item is the weighted average of the ratings given by the k most similar users or items.
(2) KNN with Means: Takes into consideration mean ratings of users. We normalize the ratings by subtracting the mean rating of the user or item from the weighted average of the ratings given by the k most similar users or items.
(3) KNN with Baseline predictors: Consists of baseline parameter for each user. The predicted rating for an item is the sum of the baseline estimate, the weighted average of the ratings given by the k most similar users or items, and a regularization term.

4.3 Slope One

Slope One is an item—based collaborative filtering method based on the concept of the popularity differential – the rating for an item for a user is predicted using the average pairwise deviation of ratings

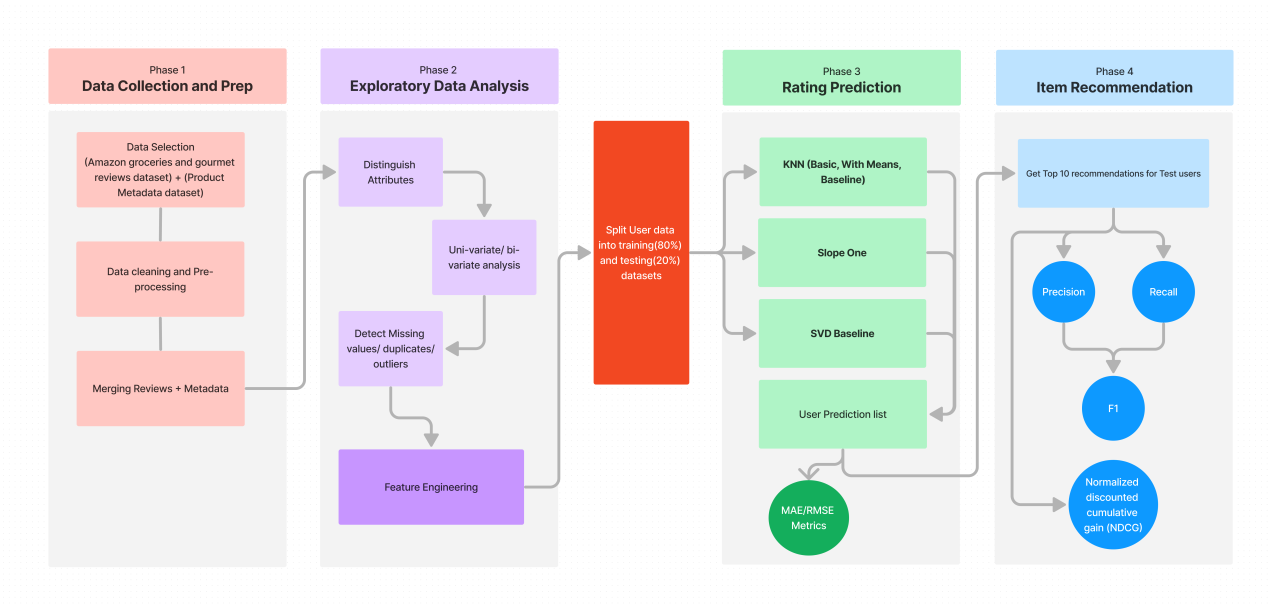

Figure 8: Recommendation System flow diagram

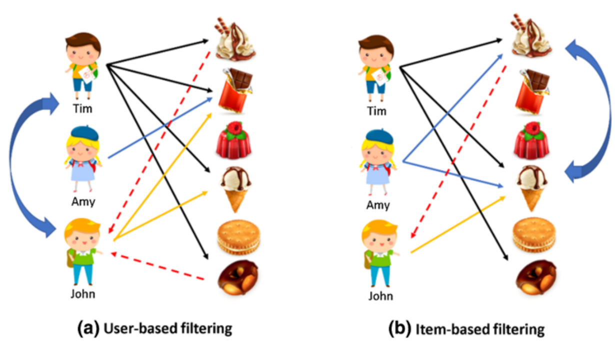

Figure 9: Approach of KNN based recommendation

of common items rated by other users as well as the predictee user [2].

Most recommender algorithms consider a subset of data to produce a predicted rating. In the case of KNN methods, this subset consists of information from other users who rated the same item and from other items rated by the same user. However, Slope One considers additional information, such as the ratings of items that the predictee also rated, and ratings by users who have rated some common item with the predictee [2]. The predicted ratings by the Slope One scheme can be further refined using a weighted average. Hence, in this project, we use the **weighted Slope One** scheme as detailed in [2].

We refer to the set of ratings by a single user as an evaluation which is an incomplete array $x(u)$, where $x(u)_i$ is the rating that the user has given to item i. The set of all the items rated in evaluation $x(u)$ is given by $S_{x(u)}$ .Let χ be all the evaluations in the training set. The set $S_i(\chi)$ is the set of all evaluations $x(u)\epsilon X$ such that they contain item i $(i\epsilon S_{x(u)})$. Taking two items i and j, $S_{i,j}(\chi)$ represents the set of all evaluations $x(u)\epsilon\chi$ such that they contain item i and j. Given a set S, the number of elements in S is given by $|S|$. Therefore, the predicted rating of an item i by a user with evaluation $x(u)$ can be represented as follows (using the weighted slope one scheme) -

$$\hat{r}_{ui} = \frac{\sum_{j\epsilon S_{x(u)}-\{i\}}(dev_{i,j} + x(u)_j)|S_{i,j}(X)|}{\sum_{j\epsilon S_{x(u)}-\{i\}}|S_{i,j}(X)|} \tag{3}$$

where $dev_{i,j}$ is the average pairwise deviation of item j with respect to item i. The deviation is defined as follows –

$$dev_{i,j} = \sum_{x(u)\epsilon S_{i,j}(X)} \frac{x(u)_i - x(u)_j}{|S_{i,j}(X)|} \tag{4}$$

4.4 SVD

SVD (Singular Value Decomposition) is a matrix factorization technique. In the context of recommender systems, it is a model-based technique which uses learned parameters to predict ratings. This method maps users and items to a new joint latent factor or ”concept” such that the user-item interactions are modelled as inner products in the new space [1]. If the latent factor space has a dimensions d, in the latent factor space each item i is represented by a vector $q_i\ \epsilon\mathbb{R}^d$, and each user u is represented by vector $p_u\ \epsilon\ \mathbb{R}^d$.

Each element in q_i gives the extent to which item i contains the corresponding latent factor. Each element in p_u gives the extent to which the user u prefers items that have high measures for the corresponding latent factor. These values can range from negative to positive. Therefore, the overall interest of the user u in an item i, and by extension, the predicted rating for item i by user u is given by $p_u q_i^T$.

We add baseline predictors to refine the predicted rating ($\hat{r}_{ui}$) -

$$\hat{r}_{ui} = \mu + b_i + b_u + p_u q_i^T \tag{5}$$

In order to learn the model parameters (b_u; b_i; p_u and q_i) we minimize the regularized squared error -

$$\min_{b*,q*,p*} \sum_{(u,i)\epsilon\chi} (r_{ui} - \mu - b_i - b_u - p_u q_i^T)^2 + \lambda_4(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2) \tag{6}$$

We use the Funk approach to gradient descent [3], where for a particular training case, the parameters are updated as follows -

$$b_u = b_u + \gamma(e_{ui} - \lambda \cdot b_u)$$
$$b_i = b_i + \gamma(e_{ui} - \lambda \cdot b_i)$$
$$q_i = q_i + \gamma(e_{ui} \cdot p_u - \lambda \cdot q_i)$$
$$p_u = q_u + \gamma(e_{ui} \cdot q_i - \lambda \cdot p_u)$$

Here, the learning rate γ, regularization parameter λ, and the number of latent factors d are hyperparameters which are tuned using cross-validation. For our dataset, we have found the following hyperparameter values to be the optimal values -

$$\gamma = 0.009;\ \lambda = 0.05;\ d = 100$$

5 ITEM RECOMMENDATION

Now we need to use the predicted rating, and leverage our knowledge of dataset to create a recommendation list of new products which the user has not rated based on the similarity from other users. We use the following approach.

(1) Create a prediction matrix such that we have data in the form of reviewerID, productID and estimated rating.
(2) We then create a dataframe such that we have all the test users and their product rating in the test data. We sort that data based on the best rating so that we can use this to create our top K recommended list.
(3) We perform the same transformation with the test data as well so that we have the labels against whom we have to test our recommendations.
(4) Next step is to write functions for precision, recall and then F1 score. Precision is calculated as the percentage of true positive over the total number of recommendations.
(5) Recall is basically ratio of true positive over all good entries. Good entries are basically top 10 best test value for each reviewer.
(6) Now we use the harmonic mean of Precision and recall to get the F1 score.

6 RESULTS AND ANALYSIS WORK

In order to demonstrate the functionality of recommender systems, we examine the top 10 recommendations generated by two different models - Slope One, which exhibits the poorest performance with respect to RMSE, and SVD Baseline, which exhibits the strongest performance with respect to RMSE. This comparison allows us to gain insight into the relative strengths and weaknesses of these models and highlights the impact of mode l selection on the accuracy of the recommendations. We sample a single user with reviewerId = $'A100WO06OQR8BQ'$.

As can be observed in Figure 10, we compare Slope One and SVD baseline predictions with the top 10 recommendations according to the Test Data and Train Data.

The items marked in brown are all related to coffee and Keurig items. We can see that these products have a high frequency among the top items in the test data (50%) and train data (50%). Consequently, they are also among the top items in the Slope One (70%) and SVD baseline (70%) predictions.

SVD baseline was also able to recommend certain items like the 'Bai Coconut Flavored Water', 'V8 Orginal Low Sodium 100% Vegetable Juice', and 'Benchmark Bouquets' with reasonable accuracy.

Slope One also performed reasonably well, showing trends that can be observed in both the Test and Train data by recommending coffee-based and peanut butter-based items.

However, one of the drawbacks of Slope One is that it is very sensitive to "popular opinion" i.e. an item that is highly rated by the vast majority of users is more likely to be recommended by Slope One irrespective of the predictee user's own rating behaviour [2]. This is exemplified by the recommendation of 'Bai Flavored Water, Sumatra Dragonfruit' which is the second recommended item by Slope One, but it actually ranks at the bottom of the Test data with an observed rating of 1 (not shown in figure).

The below tables shows the RMSE and MAE error along with the F1 scores of the prediction. Note that F1 score is less, this is due to the fact the data is sparse and not all products are rated by reviewers so, the estimation is certainly biased.

Metrics	RMSE	MAE
KNN Basic	0.9765	0.6351
KNN with Means	0.9745	0.6325
KNN Baseline	0.9808	0.646
Slope One	0.9929	0.6404
SVD Baseline	0.8558	0.6123

Metrics	Precision	Recall	F1 Score	NDCG
KNN Basic	0.6443	0.0987	0.1711	0.3203
KNN with Means	0.7076	0.1060	0.1844	0.3318
KNN Baseline	0.7589	0.1196	0.2066	0.3256
Slope One	0.7784	0.1029	0.1819	0.3403
SVD Baseline	0.7723	0.1016	0.1796	0.3140

7 CONCLUSIONS AND FUTURE WORK

The project gave us opportunity to work on real Amazon data for Gourmet foods and we were able to get reasonable results based on the data processing, models used. We also want to discuss a few points about food based recsys, to analyze the strengths, weaknesses, threats and opportunity for them to improve.

- **Transparency and Explain-ability:** The system should be able clearly present how it reaches the conclusion and with the dataset we are having we understand the process the algorithm uses like User based similarity so the factors are transparent. Explain-ability on other hand could be a challenge as we dont have a dense matrix to compare each user with its preferred items.

Index	Slope One	Slope One Rating	SVD baseline	SVD Rating	Test Data	Test Ratings	Train Data	Train Ratings
1	Bell Plantation PB2 Powdered Peanut Butter, 1 lb Jar (2-pack)	4.591134037	Bai Coconut Flavored Water, Molokai Coconut, Antioxidant Infused Drinks, 18 Fluid Ounce Bottles, 12 count	4.932835821	Knorr Pasta Sides, Beef 4.3 oz	5	Keurig, Coffee People, Jet Fuel, K-Cup Counts,Dark Roast Coffee 50 Count	5
2	Bai Flavored Water, Sumatra Dragonfruit, Antioxidant Infused Drinks, 18 Fluid Ounce Bottles, 12 count	4.54800356	Brooklyn Beans Oh Fudge Coffee Pods for Keurig K Cups Coffee Maker, 40 Count	4.614654003	V8 Original Low Sodium 100% Vegetable Juice, 5.5 oz. Can (8 packs of 6, Total of 48)	5	Napalm Coffee, EXTRA DARK ROAST, 100% Arabica, Single Serve Cups for Keurig K-Cup Brewers, 12 Count	5
3	Community Coffee Café Special Medium Dark Roast Single Serve, 36 Ct Box, Compatible with Keurig 2.0 K Cup Brewers, Full Body Smooth Full Flavor, 100% Arabica Coffee Beans	4.537324286	V8 Original Low Sodium 100% Vegetable Juice, 5.5 oz. Can (8 packs of 6, Total of 48)	4.434175681	Coffee People DARK Roast Variety Sampler * JET FUEL & BLACK TIGER * Extra Bold 48 K-Cups for Keurig Brewers	5	ALMOSTcoffee Coffee Substitute, 16 Oz. Brews Like Real Coffee, Great Tasting, Healthy, Naturally Caffeine-Free and Acid-Free, Non GMO.	5
4	Brooklyn Beans Oh Fudge Coffee Pods for Keurig K Cups Coffee Maker, 40 Count	4.438403759	Coffee People DARK Roast Variety Sampler * JET FUEL & BLACK TIGER * Extra Bold 48 K-Cups for Keurig Brewers	4.434175681	Tassimo GEVALIA Dark Italian Roast Coffee, 12 Count T-Discs, (Pack of 3)	5	College Care Healthy Premium Care Package and Military Variety Bundle (30 Count)	5
5	V8 Original Low Sodium 100% Vegetable Juice, 5.5 oz. Can (8 packs of 6, Total of 48)	4.383668126	Tassimo GEVALIA Dark Italian Roast Coffee, 12 Count T-Discs, (Pack of 3)	4.434175681	Cafe Escapes Milk Chocolate Hot Cocoa Keurig Vue Portion Pack, 32 Count	5	Green Mountain Coffee Nantucket Blend,Medium Roast K-Cup Portion Pack for Keurig Brewers 72-Count	5
6	Coffee People DARK Roast Variety Sampler * JET FUEL & BLACK TIGER * Extra Bold 48 K-Cups for Keurig Brewers	4.383668126	Cafe Escapes Milk Chocolate Hot Cocoa Keurig Vue Portion Pack, 32 Count	4.434175681	32 Count - Green Mountain Dark Magic Vue Cup Coffee For Keurig Vue Brewers	5	Reeses Puffs, Peanut Butter, 22.9 Ounce (Pack of 3)	5
7	Tassimo GEVALIA Dark Italian Roast Coffee, 12 Count T-Discs, (Pack of 3)	4.383668126	32 Count - Green Mountain Dark Magic Vue Cup Coffee For Keurig Vue Brewers	4.434175681	Bell Plantation PB2 Powdered Peanut Butter, 1 lb Jar (2-pack)	5	Senseo Extra Strong Coffee Pods 48-count Pads	5
8	Cafe Escapes Milk Chocolate Hot Cocoa Keurig Vue Portion Pack, 32 Count	4.383668126	32 Count - Starbucks House Blend & French Roast Coffee Vue Cup For Keurig Vue Brewers	4.434175681	Community Coffee Café Special Medium Dark Roast Single Serve, 36 Ct Box, Compatible with Keurig 2.0 K Cup Brewers, Full Body Smooth Full Flavor, 100% Arabica Coffee Beans	5	Bell Plantation PB2 Powdered Peanut Butter and PB2 with Premium Chocolate, 6.5 Ounce (Pack of 2)	5
9	32 Count - Green Mountain Dark Magic Vue Cup Coffee For Keurig Vue Brewers	4.383668126	The Original Donut Shop Coffee Keurig Vue Pack, 32 Count	4.434175681	Bai Coconut Flavored Water, Molokai Coconut, Antioxidant Infused Drinks, 18 Fluid Ounce Bottles, 12 count	5	Walden Farms Fruit Spread, Raspberry, 12 Ounce	5
10	32 Count - Starbucks House Blend & French Roast Coffee Vue Cup For Keurig Vue Brewers	4.383668126	Benchmark Bouquets Red Roses and White Oriental Lilies, No Vase (Fresh Cut Flowers)	4.434175681	Benchmark Bouquets Red Roses and White Oriental Lilies, No Vase (Fresh Cut Flowers)	5	Kevala Organic Sesami Tahini, 16 Ounce	5

Figure 10: Analysis of top 10 Items for a single reviewer

- **Fairness and Unbiases:** The system can be ask user feedback to understand the quality of recommendations. This way the users can recommendations that they do not think are relevant.
- **Privacy Protection:** An important feature in e-commerce business as the system should only take in information required to predict, it should not ask irrelevant questions to user. Also they should be able to control their data.

There are still certain domains which could be explored with the project. We list down some thoughts on how to better the recommendations with the constraints we have on size and sparsity.

- We can incorporate more features in the prediction like price of products and how it influences the purchase, seasonality of products. The thought process here is that maybe statistical models like ARIMA/SARIMA could be used to predict the rating as there could be some trend with items around the year.
- We can think of using content based Recommendation by performing vectorization using techniques like Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF). Then based on similar products to the user's history, we can recommend new products to people.
- NLP can be combined with other recommendation techniques such as matrix factorization to create hybrid models that capture implicit feedback of users and the semantic similarity of items.

ACKNOWLEDGEMENT

The authors would like to express sincere appreciation to Prof. Yongfeng Zhang, who provided us with an opportunity to work on this project also guided us throughout the project and helped us to learn a lot of new things. We also express our gratitude to the resources provided by Rutgers University which was used to run the simulations. Lastly, we would like to thank the TAs who were there to solve our problems throughout the project.

REFERENCES

[1] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*. Springer, 2022.
[2] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. 2008.
[3] Yuefeng Zhang. An introduction to matrix factorization and factorization machines in recommendation system, and beyond, 2022.
[4] Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. 2009.
[5] Diego Fernández Fidel Cacheda, Víctor Carneir. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. 2011.