Product Recommender System

Akshay Pandey
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, USA
pandey5@buffalo.edu

Manish Sharma Addanki
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, USA
maddanki@buffalo.edu

Divya Pandey
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, USA
dpandey2@buffalo.edu

Praveer Kothari
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, USA
praveerk@buffalo.edu

Yeshwanth Pabbathi
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, USA
ypabbath@buffalo.edu

***Abstract*— Recommender System is one of the most effective**

applications of Machine Learning which is being widely used in various industries to provide personalized recommendation. This technology has a wider scope in customer-centric environments. The intent of this project is to implement a Machine Learning algorithm to build a product recommender system while comparing multiple aspects of this approach in the real world.
Our model on is based on collaborative filtering which is one the most popular type of recommendation technique. We have proposed a UBCF approach which is rating based analysis. Our approach combines calculating similarities like Cosine, Pearson, Jaccard & Manhattan to understand the similarities among the users & then applying UBCF model to make the predictions. The proposed system recommends music based on the ratings given by user for different music products. The various graphs have been plotted to understand the data like ratings per product count, rating per user count etc. from the dataset obtained from Amazon site. Furthermore, the accuracy has been calculated based on RMSE, MSE & MAE score, and the results are described in terms of improved accuracy to achieve significantly better results than the traditional approaches.

***Keywords—Collaborative Filtering, UBCF, Cosine, Jaccard,* Manhattan.**

I. INTRODUCTION

Recommender System is a machine learning technique providing suggestions for items to be of use to a user. It can be built using both supervised and unsupervised approaches. Our approach is rather simple and is appropriate for small systems. Then Users are enticed with appropriate recommend (or) suggestions depending on their decisions by the systems. And this Recommender System is one of the most effective applications of Machine Learning which is being widely used in various industries to provide personalized recommendation.

A. Importance with Stats

The quality of a RS can be evaluated using different types of measurement which can be accuracy, precision, RMSE etc. *Statistical accuracy metrics* evaluate accuracy of a filtering technique by comparing the predicted ratings directly with the actual user rating. Mean Absolute Error. (MAE), Root Mean Square Error (RMSE) and Correlation are usually used as statistical accuracy metrics. MAE is the most popular and commonly used; it is a measure of deviation of recommendation from user's specific value.

B. Brief Related Work

The previous work related to the recommender system is discussed in this section. Prior research describes the related concepts of recommenders such as information filtering and recommendation algorithms previously used to develop recommender systems [20] which help to understand and realize the need of recommenders in the modern era of web technologies.

Jannach et al. worked on the RS which utilized the regression-based methods and item-based models for accurate recommendations. Ibrahim et al. presented a personalized intelligent information model to examine hotel services. Bouras and Tsogkas have used the user clustering Word Net-enabled k-means algorithm to recommend improved news articles. Chen and Chuang [18] optimized the performance of a ubiquitous hotel recommender system by using a nonlinear and fuzzy programming approach over the hotels dataset.

C. Bridging the Research Gap

One of the prevalent research challenges in the field of recommender system is to do better user profiling. There are some advanced user profiling techniques found in the literature to achieve the same. User profiling aims to understand the user well and as a result recommending the most relevant items to the user, where relevant means items returned as a result of intelligent techniques from various fields, mainly from data mining.

D. Summary of Contribution

This technology has a wider scope in customer-centric environments. Recommendation systems have been designed using various approaches, namely content-based, collaborative and hybrid approaches. Collective filtering algorithms are divided into two subcategories: memory-based and model-based approaches. In memory-based approaches, values of recorded interactions are directly used, assuming that no model exists, and they are essentially based on nearest neighbors. The goal of model-based approaches is to find an underlying generative model that explains user-item interactions and to make predictions with it. In all, it actually rotates around the use case, whether it is a personalized or non-personalized and the dataset available for extracting the relevant insights [1].

II. MOTIVATION

A. Scope

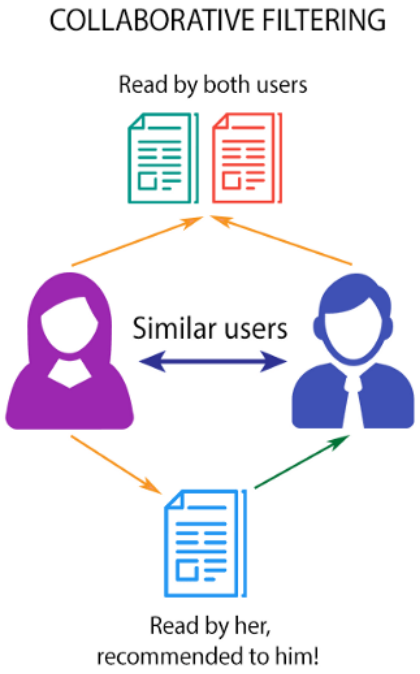
One of the major motivations for this topic is the scope of Recommendation systems which have the power to impact businesses on various scales. Since the inception of modern recommendation systems in 1992, recommendation systems have made a significant progress [2]. From an organizational standpoint, Recommendation Systems play a huge role in consumer satisfaction and retention. Amazon has over 90 million prime users which makes it an ideal use case for recommendation systems.

B. Real-World Applications

Recommendation systems have impacted various industries including Ecommerce, News and Advertising. Additionally, it has huge applications in social media like Facebook, Instagram, and Twitter. Another important area of impact has been the entertainment industry where popular platforms like Amazon Prime Video and Netflix has been focusing on recommending the most relevant media content with the primary motive of increasing user engagement on their respective platforms. So basically, recommendation systems have an application in every

III. APPROACH

We are using Amazon reviews datasets for our project. The datasets consist of several users and their rating for a variety of electronic equipment. The goal of the project is to use this dataset to create a recommender system that uses the information available to recommend similar products to similar users. Two users are said to similar if they have purchased common products & share common reviews about the products. Thus if two users are similar it is possible to recommend one user the products that are not common among them.



A. Data Exploration

We will use Digital Music dataset from Amazon. It contains 836,006 ratings on Digital Music in the "Data Group 2.csv" file. We have also tried this approach on other datasets provided by Amazon. Once such dataset is "Magazine Subscriptions" dataset which has 89,689 ratings. It includes columns of Product ID, User ID, Rating and Timestamp. The dataset is about 4 MB. We will create a user-products ratings matrix with rows as User ID and columns as Product ID.

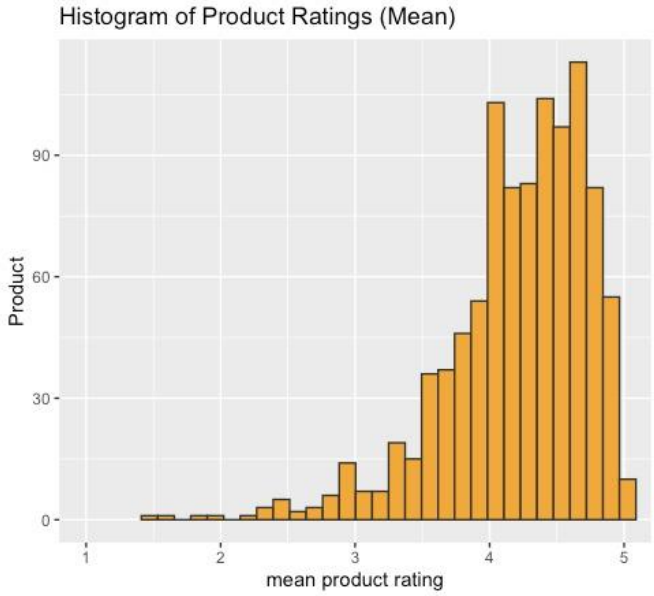
```
> head(prod_ratings,10)
           userId  productId rating  timestamp
 1: A2EFCYXHNK06IS 5555991584      5  978480000
 2: A1WR23ER5HMAA9 5555991584      5  953424000
 3: A2IR4Q0GPAFJKW 5555991584      4 1393545600
 4: A2V0KUVAB9HSYO 5555991584      4  966124800
 5: A1J0GL9HCA7ELW 5555991584      5 1007683200
 6: A3EBHHCZO6V2A4 5555991584      5 1158019200
 7: A340XJYJDFSMUG 5555991584      3 1190419200
 8: A3Q1J7VFGG80EK 5555991584      5  975628800
 9: A1REP2FMPOXV4A 5555991584      5  993427200
10: A3QEKUPBPQ7A2S 5555991584      5 1055635200
```

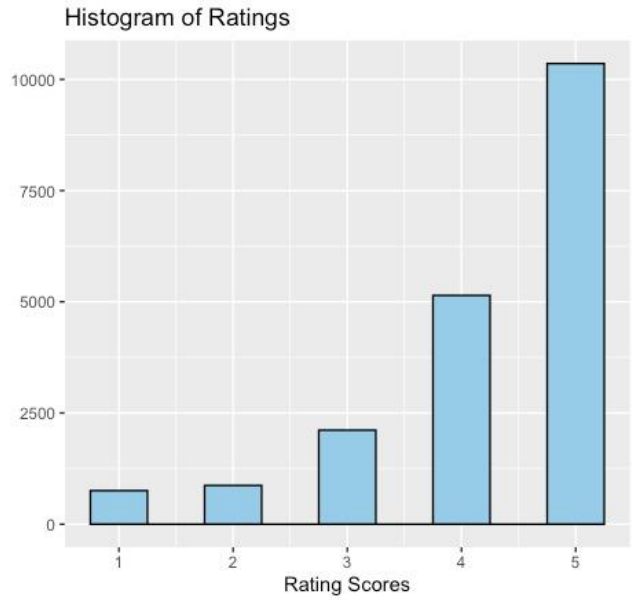
Figure 1 Data Sample

We started by exploring some relevant terms for our understanding, such as:

- *Average Rating Scores Per Product*
- *Number of Ratings Per User*

- *Most number of Ratings*
- *Count of Products Rated*



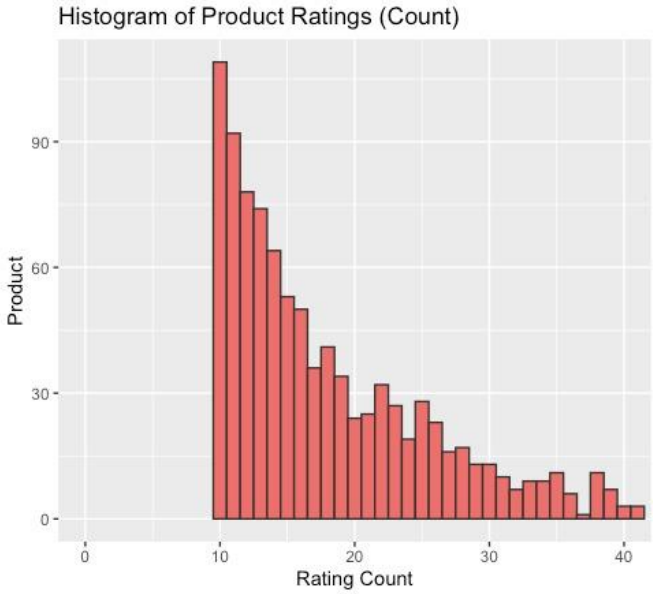


We analyzed the rating count of products and visualized it in the form of a Histogram. We can infer from the histogram that the minimum number of ratings received by any product is 10.

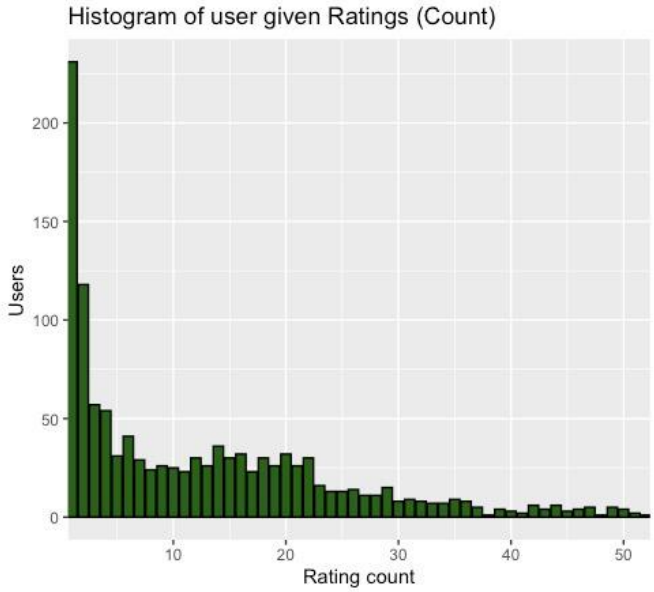
B. Data Preprocessing and Visualization

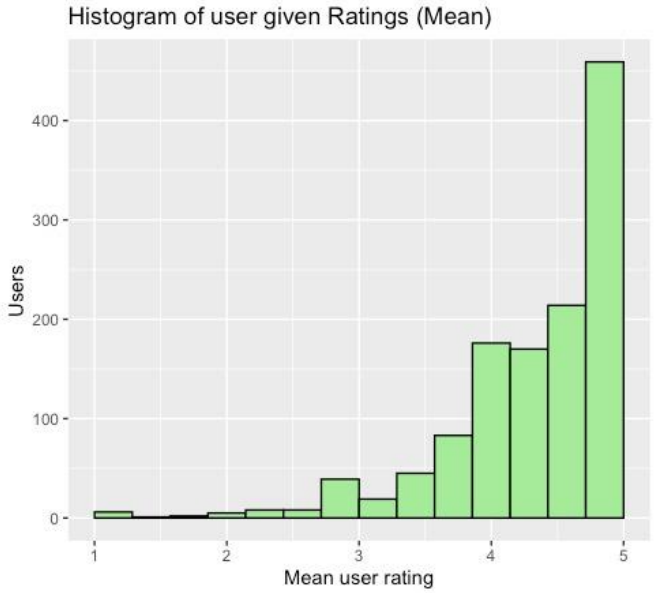
The dataset had 836,006 user rating records, ideally one would assume that each user must have rated a product once, it's only logical. But after exploration we found out that there were some duplicate entries in the data when filtered with both userID and productID. Now, it is important to remove duplicates because we need to create a user-item-rating matrix and for that to be generated, there has to be a unique set of user-item pair in the ratings. Otherwise, the code will fail mentioning that it received two arguments instead of one in the required field.

In order to make the best fit for user-item-rating matrix, we took a subset of 19236 records keeping in mind that the resource required for the calculation of similarity matrix would be reduced.



We also analyzed the number of ratings provided by the users which shapes their vector that is used to compare the similarities between the users. To our dismay, we found out that maximum users had given out ratings for only one or two products.





C. Distance Calculation

We used the following methods to calculate the distances between the user vectors to find their similarity values.

- Cosine Similarity: It is a metric that mainly measures how similar two or more vectors are. The vectors are usually non-zero and belong to an inner product space. The cosine of the angle between the vectors is cosine similarity. The divide between the dot product of vectors and the product of the Euclidean norms or magnitude of each vector describes the cosine similarity mathematically.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

- Jaccard Distance: The Jaccard index which is also known as Intersection over Union is a statistic for calculating set of samples similarity and diversity. The size of intersection is divided by size of sample sets' union. It's the overall number of related entities between sets divided by the total number of entities in practice.

$$D(x, y) = 1 - \frac{|x \cap y|}{|y \cup x|}$$

- Manhattan Distance: The Manhattan distance, also known as the Taxicab distance is a formula for computing the distance between the two real-valued vectors. Consider vectors that define items on a chessboard-like uniform grid pattern. The Manhattan represents the distance connecting two vectors assuming they can only move in the same direction. The distance is computed without any diagonal movement.

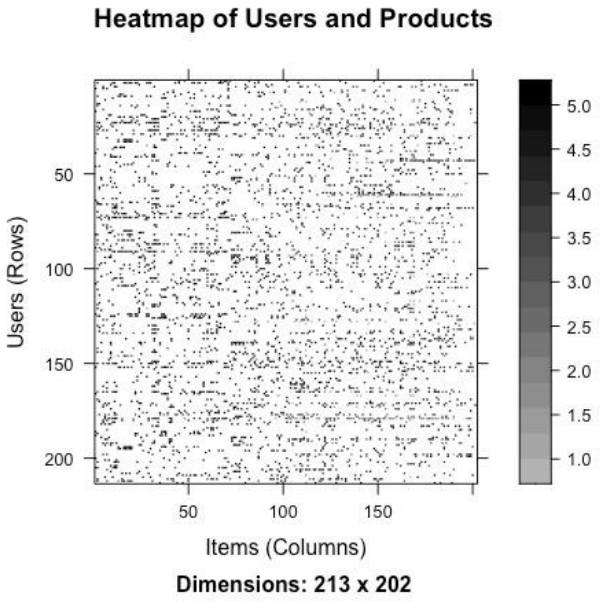
$$D(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$

- Pearson's Correlation: The Coefficient of correlation is an indicator of the magnitude of a connection between two sets of data. The Pearson's correlation test is derived by dividing the overlap of two factors by the product of the each data sample's mean difference.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

In our machine, the rating matrix takes around as much as 1.4 GB size. This size is due to the large number of zero's in the matrix. It can be called a "sparse matrix". Let's convert this into a dense matrix by removing the zeros. The package we will use for recommendations is Recommender lab.

Recommender lab provides various algorithms for testing and developing recommender algorithms. Some of them are User-based collaborative filtering (UBCF), Item-based collaborative filtering (IBCF), Alternating Least Squares (ALS), Randomly chosen items for comparison (RANDOM).



We will use real Rating Matrix from recommender lab to remove the zeros and reduce the size of matrix. This reduces the size of matrix to just 1.4 MB, which is much, much smaller. To get unbiased recommendations, we need to normalize the matrix. When calculating similarity between user ratings, we have the option of using the following methods: Jaccard Similarity, Cosine Similarity, Pearson's Correlation and Manhattan Similarity.

We used User-based collaborative filtering (UBCF) and analysed it with multiple distance calculation formulas, namely "Cosine", "Jaccard", "Pearson" and "Manhattan". We will use the Recommender() function in the recommender lab package for our recommendation model. After building our model, we can check recommendations for multiple users with different similarities.

IV. RESULTS

A. Model Results

The model recommended a list of product IDs for a specific user based on the user's past reviews and similarity with other users. We evaluated our UBCF models with different distance calculation formulas and predicted top 5 products for a specific user. According to our dataset, the predictions for a specific user were identical.

```
> Top_5_List_cos = as(Top_5_pred, "list")
> Top_5_List_cos
[[1]]
[1] "B00005N7RA" "B000IJ7RQ8" "B0037STB02" "B00WKNNN5Y" "B00007AWXX"
```

In most of the cases, they came out to be somewhat similar.

	Algorithm	Top5_Results_for_UserID_AOSFI0JEYU4XM
1	UBCF_Cosine	c("B00005N7Q2", "B00005NIND", "B00005QJE5", "B000069YW9", "B00006LB29")
2	UBCF_Jaccard	c("B00005N7Q2", "B00005NIND", "B00005QJE5", "B000069YW9", "B00006LB29")
3	UBCF_Pearson	c("B00005N7Q2", "B00005NIND", "B00005QJE5", "B000069YW9", "B00006LB29")
4	UBCF_Manhattan	c("B00005N7Q2", "B00005NIND", "B00005QJE5", "B000069YW9", "B00006LB29")

B. Model Evaluation

We used evaluation metrics for checking the effectiveness of our designed model, primarily we used RMSE to check the results for minimum RMSE value. As Root Mean squared error captures the square root of the squared error between the predicted value and the truth value, the scale remains same.

	RMSE	MSE	MAE
UBCF_Manhattan	1.076	1.157	0.851
UBCF_Cosine	1.095	1.200	0.800
UBCF_Jaccard	1.095	1.200	0.800

V. BROADER IMPACT OF RESEARCH

As a result of our study of recommender systems and it's applications for industrial use cases, we have evaluated some practical theories that must be considered while designing the near-perfect recommender system.

A. Limitations of Content Based Filtering Technique

This technique is good for the initial phases of any project where the system can do well without the requirement of a user specific or personalized recommendations. So, if we want to check the most popular product / article / blog, the recommender system is rather simple.
When we dive deeper into this requirement, we get to see the branching of user preferences. It is very common to have unique preferences, which sounds like a paradox. When considering differences between users and their preferences, that where the content based filtering technique does not help. We can still make do with this approach on a smaller scale by classifying users / products into separate segments based on the dataset. But, this can only be applied on a much smaller scale, because since this is solely based on the types of users, it can fail to provide good results once the number of classes or segments increase.
Another limitation of Content Based Filtering Technique is that we cannot expand the area of interest for any user. Since the user will be shown items similar to what they have viewed, the system would not be able to recommend new products that are not similar to the user's mentioned preferences that the user might like. This limits the thinking and expansion of the recommendations.

B. Collaborative Filtering: An Effective Approach

To explain UBCF as an effective approach, we must see how it overcomes the drawbacks that are encountered in Content Based Filtering Technique.

This technique is user to make recommendations based on user-user or item-item similarity, without worrying about the increasing number of features. So, all we need to have is the ratings provided by the users and the details of the products. The task then becomes easier to handle in two segments. The first segment evaluates the user's ratings and creates a user-item-rating matrix. This is the most important part of the recommender system because it deals with the calculation of similarity. All the users are given specific scores with respect to each other in the form of a similarity matrix.

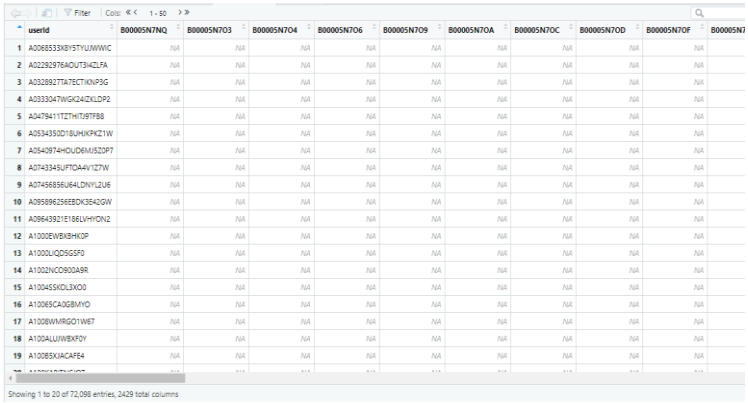
If two users have similar preferences, there is a high correlation which means their distance score is low whereas if two users are not that similar, their correlation is low. So, if two users are similar, the products positively reviewed by them that are not common between the two users are recommended to each other due to their high similarity. This technique overcomes the drawback of content based filtering approach that if a user has not reviewed product of a certain different category, they are not recommended that product.

C. Challenges to UBCF Recommender System

While working on this project, we faced a major challenge with the input data. When creating a user-item-rating matrix, we get the actual view of the table that shows how many user-item pairs have no value. It means that the particular item has not been used / rated by the user. If there are a lot of such blank cells, then it is a big challenge for the recommender model. This happens when there are lots of users that have rated only a few products from a list of thousand products.

We have filtered the input data such that we consider building recommendation system for users who have more than just a few ratings so that their similarities can be calculated accurately. What happens is that the system runs multiple rounds trying to find similar users and identify a score in order to fill this matrix. If we begin with very limited data then the recommendation might not be accurate.

So, we need to make sure we check our dataset for such conditions based on filters to avoid sparse data condition. We have shown below the output of our initial user-item-rating matrix which shows most of the values as "NA" due to missing user-item pairs in the dataset.



	userId	B00005N7NQ	B00005N7O3	B00005N7O4	B00005N7O6	B00005N7O9	B00005N7OA	B00005N7OC	B00005N7OD	B00005N7OF	B00005N7
1	A0068533XBYSTYUMWIC	NA	NA	NA	NA	NA	NA	NA	NA	NA	
2	A0229297EAOUT3I4ZLFA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
3	A0328927ITA7ECTIKNP3G	NA	NA	NA	NA	NA	NA	NA	NA	NA	
4	A0333047WGK24IZKLDP2	NA	NA	NA	NA	NA	NA	NA	NA	NA	
5	A0479411TZTHITJ9TFB8	NA	NA	NA	NA	NA	NA	NA	NA	NA	
6	A0534350D1BUHJKPKZ1W	NA	NA	NA	NA	NA	NA	NA	NA	NA	
7	A0540974HOUD6MJSZ0P7	NA	NA	NA	NA	NA	NA	NA	NA	NA	
8	A0743345UFTOA4V1Z7W	NA	NA	NA	NA	NA	NA	NA	NA	NA	
9	A0745685EU64LDNYL2U6	NA	NA	NA	NA	NA	NA	NA	NA	NA	
10	A0958962S6EBDK3E4ZGW	NA	NA	NA	NA	NA	NA	NA	NA	NA	
11	A0964392YE186LVHYON2	NA	NA	NA	NA	NA	NA	NA	NA	NA	
12	A1000EWBXBHK0P	NA	NA	NA	NA	NA	NA	NA	NA	NA	
13	A1000LIQDSGSF0	NA	NA	NA	NA	NA	NA	NA	NA	NA	
14	A1002NCO900A9R	NA	NA	NA	NA	NA	NA	NA	NA	NA	
15	A1004SSKDL3XO0	NA	NA	NA	NA	NA	NA	NA	NA	NA	
16	A1006SCA0GBMYO	NA	NA	NA	NA	NA	NA	NA	NA	NA	
17	A100BWMRGO1W67	NA	NA	NA	NA	NA	NA	NA	NA	NA	
18	A100ALUWBXF0Y	NA	NA	NA	NA	NA	NA	NA	NA	NA	
19	A100BSXJACAFE4	NA	NA	NA	NA	NA	NA	NA	NA	NA	

Showing 1 to 20 of 72,098 entries, 2429 total columns

VI. SUMMARY

We have studied the different approaches to recommender systems based on the user requirements and studied various challenges with different types of recommender systems. To summarize the study, we would like to highlight the following points:

- Recommender system is a modern approach towards an industrial requirement where companies try to make their product more customer-centric and focus more on designing products for specific user requirements. For example: Amazon is one of the largest network of buyers and sellers. Amazon has optimized its operations and used recommender systems to give personalized ads to target users. In doing so, it has increased its revenue by a greater margin.

- After explaining the impact of recommender systems on businesses, we went on to find how many categories of approaches are there for designing a recommender system. We found out that the model or approach should be selected on the basis of our requirements and the availability of relevant data.

- We studied the limitations of Content Based Filtering which made us consider User Based Collaborative Filtering for our use case and dataset.

- We highlighted the issues we faced with our dataset. We encountered sparse data also estimated the impact of it in our similarity matrix calculation which is one of the most important aspect of a recommender system.

- We used various formulas for distance calculation between the users, namely, Cosine, Euclidean and Jaccard. We evaluated the results based on these models and noted the similarities.

VII. FUTURE WORK

We have noted several points of discussion for future work related to this study. Following points cover the areas for future work:

- Incorporating More Features: We began our study with few features, we selected the dataset provided by Amazon. For future work, we can start by selecting proper dataset that has a lot of features.
- Generating Rich Data: We have encountered sparse data issue in our project, mostly so because of the cases where users did not actively reviewed products. While that cannot always be a roadblock, we can come up with other methods of dealing with sparse data that could take the users information into consideration.
- Designing Scalable Models: Another drawback of a lot of models is the issue where they cannot incorporate increasing number of users after a point, so instead of increasing resources for higher computation, we could study to find out how to optimize the distance calculation and reduce processing time even when we scale up.
- Impact of Hybrid Approaches: We have studied the concept and application of both content based filtering and collaborative filtering approaches, we can come up with alternative approaches where we use a combination of modules from both techniques.

VIII. ACKNOWLEDGEMENT

We would take this opportunity to thank our professor Dr. Nazmus Sakib in the department of Computer Science and Engineering at University at Buffalo for accepting out proposal and guiding us throughout the project implementation.

IX. REFERENCES

[1] Kunal Shah, Akshaykumar Salunkhe, Saurabh Dongare and Kisandas Andala "Recommender systems: An overview of different approaches to recommendations," in 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), in press

[2] Jan Boehmer, Yumi Jung, and Rick Wash, Recommender Systems, pp.1– 4 at Michigan State University.

[3] "10 Charts That Will Change Your Perspective of Amazon Prime's Growth" by Forbes, Mar 4, 2018

[4] "The distance function effect on k-nearest neighbor classification for medical datasets" by Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke and Chih-Fong Tsai US National Library of Medicine National Institutes of Health, in press