Customer-Based Predictive Analytics to Find The Next Best Offer

by

Amany AbdElrahman

Esraa Mohamed

Youmna Alsayed

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

September 2025

Abstract

The Next Best Offer (NBO) is the term used to describe the "correct" offer made to a consumer at "the right" time in an online setting while maximising company goals. By using information filtering technology, the appropriate goods and services for each consumer are suggested based on their explicit and implicit preferences. There are various kinds of recommendation services: a) Collaborative recommendations: Recommendations are made on the basis of similar cases of other users. (b) Content-based recommendations: Recommendations are made on the basis of the user's past cases. (c) Hybrid approaches: Recommendations are made using a combination of content-based and collaborative methods.

Acknowledgments

First and foremost we are extremely grateful to our supervisors Dr. Anwar Hossain, and Dr. Mona Farouk for their help, and supervision on this project.
We would like to thank Eng. Ashraqat Mustafa, and Eng. Mohamed Ibrahim for their help during the project.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Customers may have an unfavorable opinion of your business if you send them many or unrelated products offers. Additionally, research demonstrates that giving customers additional options might decrease their satisfaction and decrease their likelihood of making a purchase. To prevent spam and enhance product offerings, marketers need a mechanism to decide when and how to engage customers. Winning and keeping consumers is now crucial to the success of organizations. As a result, any large customer-focused organizations face a strong demand for systematic and computer-based customer relationship management (CRM). Predictive models on consumer behavior are intended to enable effective direct marketing campaigns and high-quality cross-selling services.

The Next Best Offer (NBO) is an expansion of the recommendation system that keeps corporate goals in mind. NBO is the term used to describe the "correct" offer made to a consumer at "the right" time in an online setting while maximising company goals. Next best offer asks, "Which is the offer that is going to help and appeal to the customer the most right now?"[2]

1.2 Problem

In today's business, a lot of companies are falling in sales because there are a lot of customers who have become not interested or motivated to buy products from their websites, there are old marketing techniques and bad locations. Nowadays companies try to focus on marketing their products to attract a lot of customers in different ways like suggesting related products to the customer's interests, and to the customer's previous purchase.

1.3 Objective

The recommendation system helps keep business objectives in focus. Offering customers of online retailers suggestions about what they might like to buy, based on their past history of purchases and/or product searches. Beside that, this will increase the company's profits and sales, so will reflect on the company improving its vision and plans on how to satisfy the customer in the best way.

1.4 Organization of Thesis

We proceed by introducing conformance checking. Discussing the important concepts, and taking a brief background on the application, and the domain we will work at in Background in the next chapter 2. In the same chapter 2, we discussed the related works to our application in the Related Work. In chapter 3, we introduce the dataset we will use to implement our work in section Data Collection, and the tools we will use in section Tools and Libraries. Besides, discussing the preprocessing steps that we will go through in section Data Preprocessing, and algorithms that we will use in our implementation in section Algorithm. In the same chapter 3, we started with discussing how we will evaluate our work in the Evaluation section. Besides, make some data exploration in section Data Visualization. Then, by starting in chapter 4, we discuss the results of all work that we applied to the dataset we have. Furthermore, we discussed the challenges, and the limitation we faced in Challenges and Limitation section, we summarized and concluded all the work, we made in section Conclusion, and talked about the future work that we plan to do in chapter 5.

Chapter 2

Background and Related work

2.1 Background

A recommendation system is an AI algorithm, usually associated with machine learning, that uses Big Data to suggest or recommend additional products to consumers. These can be based on various criteria, including past purchases, search history, demographic information, and other factors. Recommender systems are highly useful as they help users discover products and services they might otherwise have not found on their own.

Recommender systems are trained to understand the preferences, previous decisions, and characteristics of people and products using data gathered about their interactions. These include impressions, clicks, likes, and purchases. Because of their capability to predict consumer interests and desires on a highly personalized level, recommender systems are a favorite with content and product providers. They can drive consumers to just about any product or service that interests them, from books to videos to health classes to clothing. There are several types of RSS, which base their recommendations on different types of information. RSs are classified as follows:

Collaborative recommendations: Recommendations are made on the basis of similar cases of other users. Collaborative filtering filters information by using the interactions and data collected by the system from other users. It's based on the idea that people who agreed in their evaluation of certain items are likely to agree again in the future.There are two types of collaborative filtering:

User-based collaborative filtering (UBCF): It is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user.

Item-based collaborative filtering (IBCF): It is based on the similarity between items calculated using the rating users have given to items. IBCF applying dimension reduction uses an optimized data by reducing dimention of items. R-IBCF provides better quality of predictions than IBCF and UBCF.[5]

Content-based recommendations: Produce suggestions based with respect to the clients inclinations and profile. They attempt to match clients to things which they've enjoyed beforehand. The degree of comparability between things is by and large settled in light of properties of things loved by the client. Dissimilar to most Collaborative separating models which influence appraisals between target client and different clients. Content put together models center with respect to the evaluations given by the objective client themselves. Fundamentally, the content based approach uses various wellsprings of information to produce suggestions.

Hybrid approaches: Recommendations are made using a combination of content-based and collaborative methods the model used or the algorithm used. Hybrid recommendation system whose architecture is based on agent technology it employs a multi-agent system (MAS) with a recommendation system that is capable of learning through a case-based reasoning (CBR) model. MASs are composed of a series of agents that collaborate with each other to fulfil a common goal that a single agent would not be able to achieve.[6]

2.2 Related Work

The goal of recommendation systems is to foresee the preferences a consumer would have for a given product. The two recommendation frameworks were examined:

Adaptive Cognizance (AC): By taking the arithmetic average of all three: cosine similarity, Euclidean similarity, Manhattan similarity, and importance index. The recommendations are improved compared to the approach using individual similarity measures. AC involves a process of recommending recharge packages in the form of rankings based on similarity calculation modules of usage and product profiles.

Bayesian Network (BN): based recommendation approach was also employed. Through the fusion of structural equation modelling and probabilistic graphic modelling, structural causal modelling suggested a causal language. The causal networks record the joint probability distribution of each user's product consumption and purchase history.

The best recommender model was BN with feature engineering developed. The daily aggregated usage data is better than the monthly aggregated usage data 95.38% conversion and 36.26% revenue.[2]

Collaborative filtering works by predicting which products a user will find interesting based on their preferences. It is a technology to recommend items based on similarity. There are two types of collaborative filtering:

The user-based collaborative filtering algorithm (UBCF): uses the hunch that users will probably favour the things that other users like to offer helpful content to users. In order to combine the neighbour user's rating score, the algorithm first attempts to identify the user's neighbours based on user similarities. This is done by using either supervised learning techniques like the k-nearest neighbour's algorithm and Bayesian network or unsupervised learning techniques like the k-means algorithm.

Item-based collaborative filtering(IBCF): is much the same as that for user-based collaborative filtering. It examines a group of items rather than just the closest neighbours; the target user has already rated items.

The goal of the proposed R-IBCF is to provide a better quality of prediction in terms of the MAE measure and to make a faster execution time. To evaluate the accuracy of a recommendation system, they used statistical accuracy metrics, Mean Absolute Error **(MAE)**.[5]

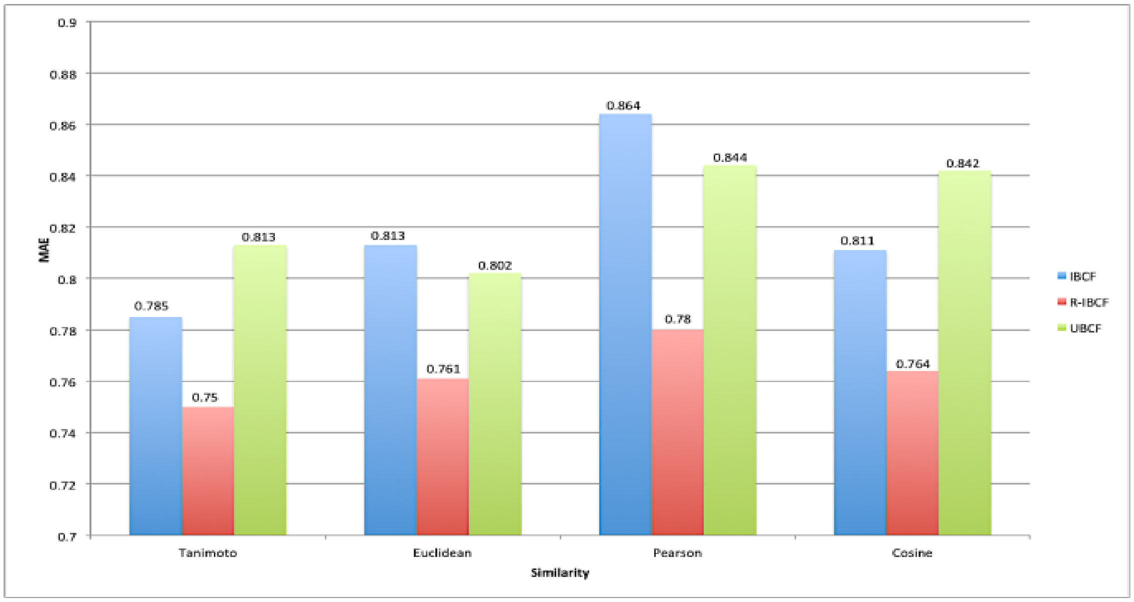

Figure 2.1: Prediction quality of IBCF, R-IBCF, and UBCF

Another algorithm used for recommending 'next-best offers' is iPrescribe. iPrescribe a scalable low-latency architecture which uses an ensemble of deep learning and machine learning algorithms for prediction. For machine learning (gradient boosting, XGBoost), and deep learning (Long Short Term Memory (LSTM)) with optimisation to avoid having to store and process user-histories for every prediction.

The Observation from this paper is that Combining multiple models' predictions yields better accuracy than traditional collaborative filtering and Deep learning models give higher accuracy, however, model inference using RNNs (such as LSTMs) requires each user's history for each prediction, leading to higher model inference time.The model is configurable and can be built for any business domain since iPrescribe uses meta-models, which define business transaction data sets and functions for creating business-specific features [8]

Automated recommender systems create individualized predictions about product liking by shifting consumer preference statements and historical behavior to lessen the complexity of the decision-making process.We define group value as the unweighted mean of the value that the group members gain from consuming a product. The suggestions that arise from this calculation are intended to maximize group value and are generated through collaborative filtering or similar methods.

The model runs two kinds of analyses: In the first regression, we compared the group and single recommender conditions by regressing group value.The second regression, comparing the group and no recommender conditions. The regression results shed new light on H2. The proposed positive effect of group recommenders on group value, though not relevant in all situations, emerged in situations in which agents were characterized by their positive attitude toward recommenders. In other words, our rejection of a general effect of group recommenders can be attributed primarily to agents who do not think highly of automated recommenders and in practice would hardly use them.[4]

Algorithm	Type	Mean Absolute Error	Compared with Study Algorithm	RMSE	Compared with Study Algorithm
Algorithm used (user-to-user, Euclidian distance)	Collaborative filtering	17.385		22.866	
User-to-user, Pearson	Collaborative filtering	16.921	–2.6%	22.158	–3.0%
User-to-user, cosine	Collaborative filtering	17.373	–.0%	22.551	–1.3%
Item-to-item, Pearson	Collaborative filtering	16.807	–3.3%	22.174	–3.0%
Item-to-item, cosine	Collaborative filtering	17.214	–.9%	22.521	–1.5%
Funk (2006)	Matrix factorization	16.951	–2.4%	22.139	–3.1%
Koren, Bell, and Volinsky (2009)	Matrix factorization	16.769	–3.6%	21.111	–7.6%
Average deviation			–2.1%		–3.2%

Figure 2.2: Algorithm Performance

Chapter 3

Methodology

3.1 Data Collection

The three datasets being used for the analysis are a list of books(Bx-books) - including ISBN, author, title, and year published; a list of books that have been rated(Bx-book-Rating) - including ISBN, book rating, and user id who rated the book; and finally, the list of users(Bx-Users) who are rating the books using the application - including their location, age, and user id. The files are csv type.

We merged the dataset of the two datasets[Bx-book-Rating, Bx-Users] by the column [User-ID] then we merged the results of this merge with the last dataset [Bx-books] by the column [ISBN]. The observation of merged dataset is that: it's shape is (1031175, 12). there are 1 missing value in Book-Author column, 2 missing value in Publisher column and 277845 missing value in Age. After that we performed preprocessing on the merged dataset. Table 3.1 shows the description of the fields of the dataset.

Field	**Description**
ISBN	ISBN of book
Book-Title	Title of book
Book Author	Book author
Year-Of-Publication	Year book was published
Publisher	Book publisher
Image-URL-S	Image of book with small size
Image-URL-M	Image of book with medium size
Image-URL-L	Image of book with large size
User-ID	Book reader's unique user ID
Book-Rating	Book rating by individual user
Location	Location where user is from
Age	Age of user

Table 3.1: Description of fields.

3.2 Tools and Libraries

We used the Kaggle code platform, and Jupyter notebook to run our files.

We used multiple libraries to implement our code like the following:

- Pandas to read the dataset CSV files
- Matplotlib, PIL, and WordCloud to make the data exploration vistualization
- Numpy, and math to use some mathematical operations

- sklearn.metrics.pairwise to make the pairwise calculations
- sklearn.metrics, and surprise to evaluate the models used

3.3 Data Preprocessing

Real-world data is in most cases incomplete, noisy, and inconsistent. Data preprocessing resolves such issues and makes datasets more complete and efficient to perform data analysis.

Handle the missing values in the dataset, the missing data we presume will not drastically affect the end results. The dataset has a 277845 missing values of Age column, 2 missing values of publisher column and one missing value of author, so we handle the missing value in 2 ways the first way replaced the missing values in publisher and author columns with constant values. the secound way by dropping the column that has a lot of missing values.

Handling the duplicated value While it might be possible that each user could review multiple books, there are no duplicated rows in all the dataframe.

Features cleaning by dropping some of features unnecessary such as: [Location, Year-Of-Publication, Image-URL-S, Image-URL-M, Image-URL-L].

Feature engineering by make new feature from the existing features such as: from the location column extract the city, country and status columns. from the rating column create new feature is the number of user that rating the book. We realized that we have duplicated row between [user-id] column and]book-title] column however this column should be unique row.so we dropped the duplicated values.

The initial reviews of the data shows some missing pieces. Therefore, I made some assumptions about the data: we removed any Book Ratings = 0. As a user of the Good Reads app, I know that that you cannot give a 0 rating. Therefore, we are assuming they are unrated entries, then removed any books that had a Publishing Year of 0. I made the assumption that this is an improperly created entry and there could be other data problems with the entry.

3.4 Algorithm

3.4.1 Item-based Collaborative Filtering

The Item-based collaborative filtering approach compares users and their ratings to a certain set of items. The algorithm calculates the nearest neighbors to a given user. Every neighbor's previously rated items are compared to the item in focus with an item similarity computation. When the most similar items have been determined, the prediction is computed by taking the weighted average of every neighbor's ratings of those items. The calculation of item similarities can be done with several methods including cosine-based similarity, correlation-based similarity, and adjusted cosine similarity.

Correlation-based Similarity

The correlation-based similarity between two items m and n is calculated by computing the Pearson correlation [20]. The Pearson correlation for the items m and n is rated by a set of users U is computed as follows

3.5 Evaluation

Recommender algorithm performance can be measured by testing against different datasets. Cross-validation is used in order to test algorithm prediction performance. Algorithm prediction performance is measured with different metrics. Which metric to use depends on the purpose of the algorithm and the goal of the measurement. The metrics relevant to this paper will be outlined below.

RMSD, MAE, and k-fold cross-validation are examples of common statistical accuracy measures that are used to assess a recommender's accuracy.

RMSD (Root Mean Square Deviation):

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \hat{x}_i\right)^2}{N}}$$

Figure 3.1: RMSD

MAE (Mean Absolute Error):

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} \left|y_i - x_i\right|}{n}$$

Figure 3.2: MAE

K Fold Cross Validation: Through accuracy measures, K fold cross validation can be utilised to deduce the model's outcomes. The recommendation system is trained independently on each individual training set or fold before the effectiveness of the resulting systems is evaluated against the test set. [1]

3.6 Data Visualization

Figure 3.3 shows the distribution of the ratings (0 to 10) in the dataset, the dataset contains both explicit ratings, on a 1–10 scale, and implicit actions of unspecified nature. 0 values indicate all interactions without rating values. Analyzing the rating distribution and the value counts of the ratings column in the dataset, we notice that the number of implicit ratings is significantly high.

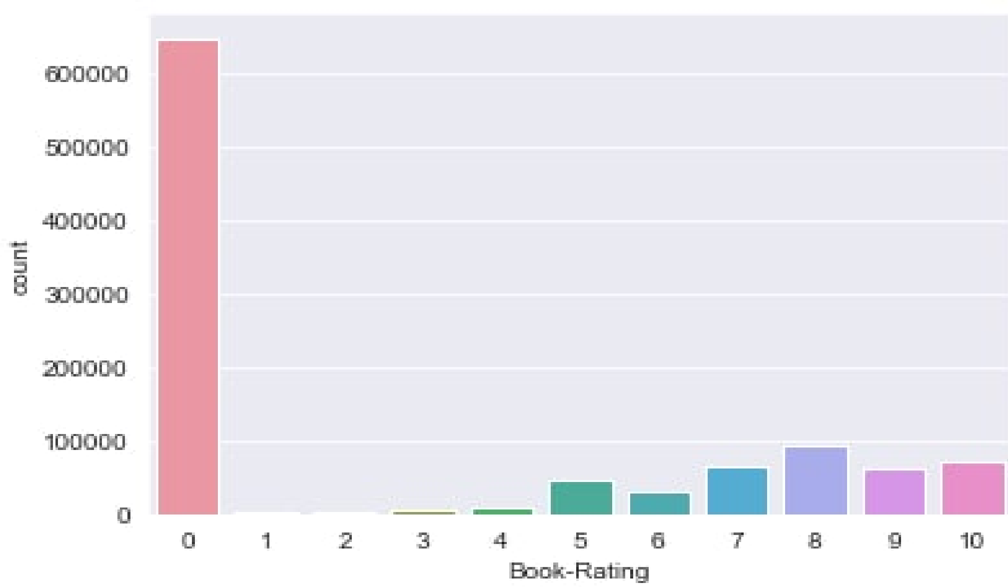

Figure 3.3: Rating Distribution

Figure 3.4 shows the distribution of the explicit ratings in the dataset, users generally give higher ratings to books as per the above distribution.

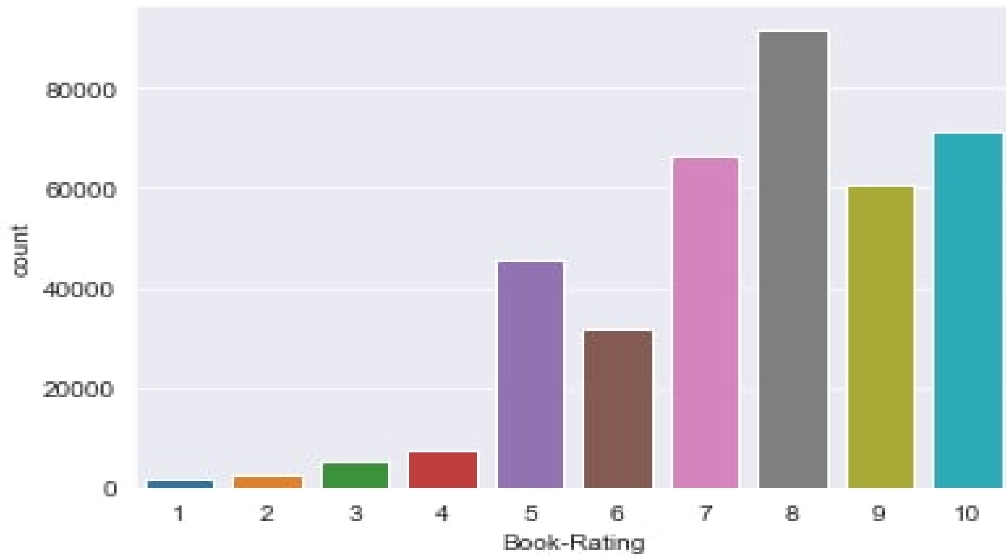

Figure 3.4: Explicit Rating Distribution

Figure 3.5 shows the distribution of the top 10 books title in the dataset. From the figure, we found that "Wild Animus" book has the highest rating distribution. We also observed that from the third book to the last book of the top 10 have a close rating distribution to each other.

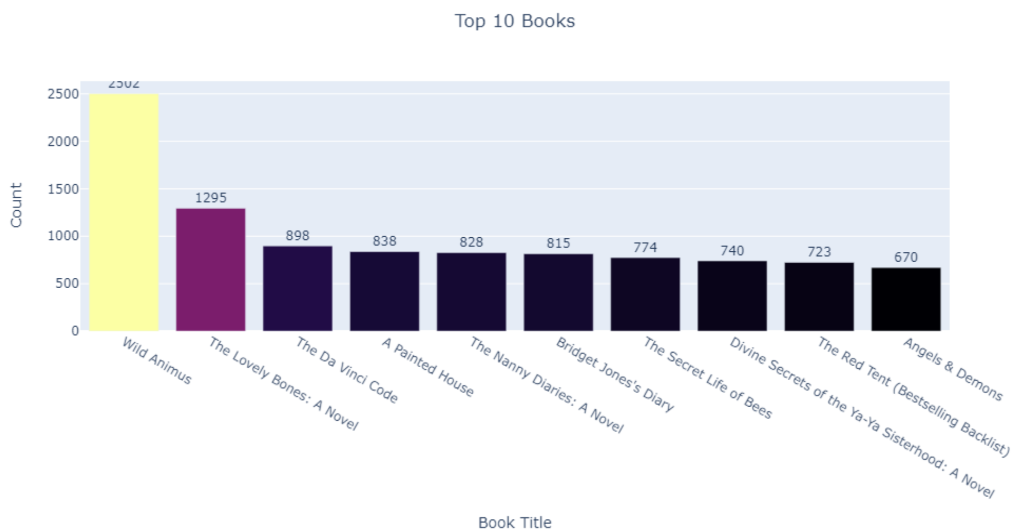

Figure 3.5: Top 10 Books

Figure 3.6 shows the distribution of the top 20 books publishers in the dataset. From the figure, we observed that "Stephen King" is the most popular publisher in our dataset. We also observed that the third and the fourth publishers have a close rating. Besides, for the rest publishers, the difference between their ratings is not high.

Figure 3.6: Top 20 Publishers

Figure 3.7 shows the percentage of the most reviewed books. We observed that the most popular book has the highest percentage is "Wild Animus". Besides, the lowest-reviewed book is "The Secret Life of Bees".

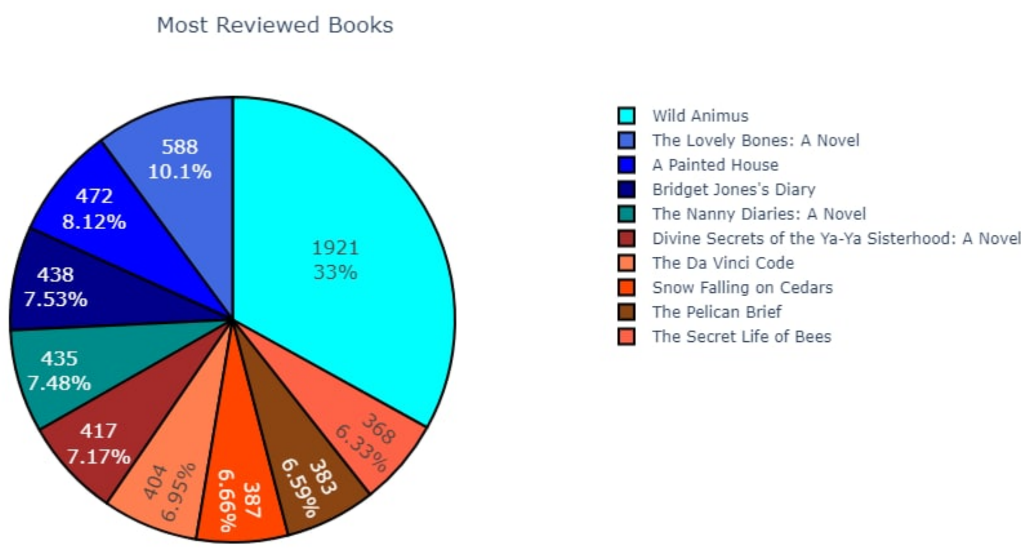

Figure 3.7: Most Reviewed Books

Chapter 4

Results

In this section we present our results of applying Memory based collaborative filtering and Model Based collaborative filtering techniques for generating predictions. In assessing the quality of recommendations. Recommender systems has used several types of measures for evaluating the quality of a recommender system. and we used Root mean square error to evaluate our model and the results we got from the two models:

- RMSE Model based : 1.6352
- RMSE for memory based : 7.9499 for both item and user based

As we know the lower the mean is the mean is the better so Model based perform very well then the Memory based.

We made a comparison between different papers use different a proches in recommendation system and the following table 4.1 will show the results.

Author	**Dataset**	**Approach**	**Evaluation**
Rekha Singhal [1]	PAKDD	iPrescribe with XGBoost and LSTM	AUC = 0.67 F1 =0.3843
Animesh Deval [2]	not mentioned	Bayesian Network (BN)	performed very well
Paolo Falcarin[3]	MySQL database system	GenericItemBased	performed very well
Greg Linden [4]	Amazon database	Collaborative Filtering	performed very well
Kiran Gajanan Javkar[7]	Book dataset	Best offer recommendation server	performed very well
Alberto Rivas [6]	not mentioned	Hybired recommender system	performed very well

Table 4.1: Comparison.

Chapter 5

Discussion

5.1 Challenges and Limitation

At the beginning we had some issues in the working time because we could not find suitable dataset to work on, and define the domain that we will work in well. As most of the datasets that we found were huge, so we would need fast resources like GPU to finish our preprocessing, and running the models we need. We also wanted to use other methods like content-based, and Hybrid approach, but we could not find enough time to work on them. So, we will cover them in the future work.

5.2 Conclusion

Recommender systems are a strong new technology for extracting additional value for a business. These systems help users and items they want to buy from a business. Recommender systems benefit users by enabling them to and items they like. They help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Offering customers of online retailers suggestions about what they might like to buy, based on their past history of purchases and/or product searches, and that's will increase the company's profits and sales. New technologies are needed that can dramatically improve the scalability of recommender systems. In this paper we presented and experimentally evaluated a new algorithm for CF-based recommender systems. Our results show that item-based techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations.

5.3 Future Work

Our vision for the project first, to use more algorithms that are possible to give more accurate predictions like a Content-based filter and Hybrid filter.
Second, to find new ways to evaluate the models, and methods we use.
Besides, to create a website that can handle the idea of our project easily.

Bibliography

[1] Recommendation systems explained. URL `https://towardsdatascience.com/recommendation-systems-explained-a42fc60591ed`.

[2] A. Deval, R. Kumar, and R. Vyas. A recommendation of right product to right customer, next best offer: A comparative study. In *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–5. IEEE, 2019.

[3] P. Falcarin, A. Vetrò, and J. Yu. A recommender system for telecom users. 2020.

[4] T. Hennig-Thurau, A. Marchand, and P. Marx. Can automated group recommender systems help consumers make better choices? *Journal of Marketing*, 76 (5):89–109, 2012.

[5] Y. Lee. Recommendation system using collaborative filtering. 2015.

[6] A. Rivas, P. Chamoso, A. González-Briones, R. Casado-Vara, and J. M. Corchado. Hybrid job offer recommender system in a social network. *Expert Systems*, 36(4): e12416, 2019.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering

recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[8] R. Singhal, G. Shroff, M. Kumar, S. R. Choudhury, S. Kadarkar, R. Virk, S. Verma, and V. Tewari. Fast online'next best offers' using deep learning. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 217–223, 2019.