Building a Personalized Recommendation System for Yelp Restaurants

Biswas Nandamuri

The University of Oklahoma

Norman, OK, USA

Abstract

Recommendation systems are a powerful tool that can be used to improve the user experience by providing personalized recommendations. In this research project, I compare the performance of different recommendation systems for restaurants located in Pennsylvania, Florida, and Los Angeles, USA. I build recommendation systems using random, popularity, and collaborative filtering techniques. I evaluate the performance of the recommendation systems using metrics such as root mean square error, mean absolute error, mean absolute recall at k and catalog coverage. I find that the collaborative filtering models provide more relevant results to the users and have the most catalog coverage. I also find that SGD is the best collaborative filtering model for Yelp Dataset.

I believe that my findings will be of interest to researchers and practitioners who are interested in building recommendation systems for restaurant catalog businesses.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Recommendation systems have revolutionized the way businesses interact with their customers. With the rise of e-commerce platforms, music streaming services, and social media, personalized recommendations have become more critical than ever. By providing users with personalized recommendations, businesses can enhance the user experience and increase engagement, loyalty, and revenue.

The history of recommendation systems dates back to the 1990s when content-based filtering was the dominant approach. Content-based filtering recommends items similar to those a user has interacted with or liked. For instance, if a user frequently searches for vegan recipes, a content-based filtering system will recommend other vegan recipes. However, this approach has limitations, as it relies on explicit user feedback, which can be incomplete or inaccurate. And always recommending vegan recipes that lack the variety and exploration aspect needed to keep customers re-visiting the website.

Collaborative filtering emerged as a popular technique for building recommendation systems to address this limitation of content-based filtering. Collaborative filtering works by leveraging users' collective intelligence to identify items relevant to a user's preferences. For example, if a group of users frequently buys the same type of book, the collaborative filtering system will recommend that book to other users with similar purchasing habits. Collaborative filtering has become a popular technique because it makes recommendations without relying on explicit user feedback. For example, Amazon's recommendation system uses collaborative filtering[2] to recommend products to its users.

In this project, I focus on building recommendation systems for restaurants located in Pennsylvania, Florida, and Los Angeles, USA[1], using collaborative filtering techniques. I utilize the Yelp dataset, which contains information about businesses, reviews, and users, to

[1] Explanation about why just these three cities are stated in Data chapter

build these recommendation systems. The primary objective of this project is to compare the performance of different collaborative filtering techniques, including memory-based and model-based methods.

With the rapid growth of online businesses, recommendation systems have become a vital component of many digital platforms. These systems have proven to be a critical success factor for companies like Amazon, Netflix, and Spotify. By analyzing past user behavior and identifying patterns, recommendation systems can predict which items a user is likely to prefer, increasing the likelihood of conversion and customer retention.

In conclusion, this project aims to contribute to the growing research on recommendation systems, specifically in the restaurant catalog industry, similar to Yelp. By leveraging collaborative filtering techniques and a dataset of hypothetical restaurant listings, I hope to provide valuable insights into which filtering techniques are most effective in recommending restaurants to customers. The goal is to help a hypothetical restaurant catalog business enhance its customer experience and online presence and increase revenue. As a data science and analytics practitioner, my expertise and knowledge in the field will be utilized to develop recommendation systems that can help the business achieve its objectives.

Additionally, the insights gained from this project can be applied to other businesses similar to the restaurant catalog industry, helping them to improve their recommendation systems and ultimately enhance their customer experience. By conducting this research, I hope to contribute to advancing recommendation systems in the restaurant catalog industry, helping businesses better serve their customers and stand out in a highly competitive market.

Chapter 2

Objectives

This project addresses the need for personalized recommendations for customers searching for restaurants. Although numerous restaurant review websites and apps are available, the vast number of options can be overwhelming for users. Often, customers end up choosing restaurants based on their limited knowledge and preferences rather than tailored recommendations that could better meet their needs.

The specific nature of the problem I am solving is building recommendation systems for restaurants in Pennsylvania, Florida, and Los Angeles, USA. These systems will use collaborative filtering techniques to generate personalized customer recommendations based on their past restaurant interactions, as well as the collaborative interactions of similar users. The goal is to provide customers with tailored restaurant recommendations that meet their preferences, leading to enhanced customer experience, online presence, and increased revenue for hypothetical restaurant catalog businesses.

The specific objectives of this project are as follows:

- To develop and compare the performance of different collaborative filtering techniques, including memory-based and model-based methods, to identify the most effective technique for building recommendation systems for the Yelp dataset.
- Build recommendation systems to generate personalized customer recommendations based on their past interactions with restaurants, including ratings, reviews, and check-ins.
- To evaluate the performance of the recommendation systems using metrics such as Root Mean Square Error, Mean Absolute Error, Mean Average Recall at K, and Catalog Coverage, to ensure that they effectively provide personalized recommendations to customers.

To know whether the objectives are reached, I will evaluate the performance of the recommendation systems using various metrics, such as the following:

- Root Mean Square Error (RMSE): RMSE is a commonly used metric for evaluating the accuracy of recommendation systems. It measures the root mean squared error between predicted and actual rating users give. A lower RMSE value indicates better performance.
- Mean Absolute Error (MAE): MAE is another widely used metric for evaluating recommendation systems. It measures the average absolute difference between the predicted ratings and the actual ratings given by users. Like RMSE, a lower MAE value indicates better performance.
- Mean Average Recall at K (MAR@K): This metric evaluates the relevance of the recommendations by measuring the fraction of relevant items in the top-K recommendations. It considers the predicted and actual ratings of items and measures how well the system predicts what the user likes. A higher MAR@K value indicates better performance.
- Catalog Coverage: This metric measures the percentage of the total catalog items recommended to at least one user. A higher catalog coverage indicates that the system can recommend more items to users, ensuring variety.

By comparing the performance of different collaborative filtering techniques, I will determine the most effective technique for building recommendation systems for the Yelp Dataset.

In conclusion, the specific problem this project aims to address is the need for personalized recommendations for customers searching for restaurants. The objectives of the project are to develop and compare the performance of different collaborative filtering techniques, build recommendation systems that can generate personalized recommendations for customers, evaluate the performance of these systems, provide insights into the factors that influence their effectiveness, and explore the potential impact of these systems on hypothetical restaurant catalog businesses. The project's success will be evaluated based on the performance of the recommendation systems and the potential impact on restaurant catalog businesses.

Chapter 3

Data

The success of any data-driven project depends on the quality and suitability of the data being used. This chapter outlines the data environment for this project, including the steps taken to ingest, prepare, and explore the data. The dataset used in this project is from Yelp, a popular online review platform. To make the data more amenable to analysis, a number of techniques were applied, including data cleaning, filtering, and reshaping. These steps are described in detail in the following sections. Finally, the chapter explores the dataset, providing insights into the data and informing the subsequent analysis.

3.1 Ingestion and Preparation

The first step in this project was to obtain the data from a reliable source. For this purpose, Yelp's public dataset is available for download on their website. The dataset includes information on businesses, users, and reviews, including tips and check-ins. The data is provided as multi-level JSON files, which can be challenging to work with directly. The multi-level JSON objects in the files is shown in Figure 3.1.

To prepare the data for analysis, PySpark was used to convert the multi-level JSON objects to single-level CSV files. This reshaping involved creating dummy variables (with values of 0 and 1) where needed, such as business attributes and business categories. Once the single-level information was extracted from the JSON objects, the resulting CSV files were pushed to a PostgreSQL database. Data in the PostgreSQL database helped reduce the system load and made it easier to manage and work with the data using SQL.

During the initial exploration of the data, I discovered that there was a fair amount of sparsity in the user-business rating matrix. To reduce the sparsity, some filters were applied to the dataset. First, the ratings were limited to only those given by users with at least ten

```
{
    "business_id": "lk9IwjZXqUMqqOhM774DtQ",
    "name": "Caviar & Bananas",
    "address": "2031 Broadway",
    "city": "Nashville",
    "state": "TN",
    "postal_code": "37203",
    "stars": 3.5,
    "review_count": 159,
    "attributes": {
        "RestaurantsTakeOut": "True",
        "RestaurantsReservations": "False",
        "RestaurantsAttire": "'casual'",
        "OutdoorSeating": "True",
        "BestNights": "{'monday': False, 'tuesday': False, 'friday': False, 'wednesday': False, 'thursday': False, 'sunday': False, 'saturday': False}",
        "RestaurantsTableService": "False",
        "Ambience": "{'touristy': False, 'hipster': False, 'romantic': False, 'divey': False, 'intimate': False, 'trendy': True, 'upscale': False, 'classy': True, 'casual': True}"
        // shortened for brevity
    },
    "categories": "Coffee & Tea, Restaurants, Wine Bars, Bars, Nightlife, American (Traditional), Event Planning & Services, Food, Caterers, Breakfast & Brunch, Cafes, Diners",
    "hours": {
        "Monday": "7:0-17:0",
        "Tuesday": "7:0-17:0",
        // shortened for brevity
    }
}
```


Figure 3.1: Complex multi-level JSON objects

ratings in the database. This filtering step helped ensure that the data contained more active users, which would indicate their preferences. Similarly, the ratings were limited to businesses with at least ten ratings in the database. This step helped to ensure that more popular businesses would be more representative of the Yelp ecosystem.

Additionally, the dataset was limited to ratings related to businesses that were categorized as restaurants. This step helped to ensure that the data is from a more focused domain, which would be more indicative of restaurant preferences. Finally, the dataset was limited to businesses in the top three cities, determined based on the number of businesses in each city. This filtering helped to ensure that the dataset used is from a representative sample of Yelp's business ecosystem. The SQL query used to construct the final merged dataset is as below:

```
SELECT
    r.user_id, r.business_id, r.stars,
    bb.name, bb.state, r.text
FROM review r
JOIN (
    SELECT user_id FROM review
    GROUP BY user_id HAVING COUNT(*) >= 10
) u ON r.user_id = u.user_id
```

```
09 |  JOIN (
10 |      SELECT business_id FROM review
11 |      GROUP BY business_id HAVING COUNT(*) >= 10
12 |  ) b ON r.business_id = b.business_id
13 |  JOIN business bb ON bb.business_id = r.business_id
14 |  JOIN business_cat2 bc ON bc.business_id = r.business_id
15 |  WHERE bc.cat_restaurants = 1;
```

Overall, the ingestion and preparation phase of the project was critical for ensuring that the models are trained on clean and relevant data that is representative of Yelp's business ecosystem. Using PySpark and PostgreSQL helped make the data more manageable and reduced the load on the system. Applying various filters to the dataset helped reduce sparsity and focus our analysis on relevant data.

3.2 Exploration

The Yelp dataset is a large and comprehensive dataset of reviews, businesses, and users. It contains over 6.99 million reviews, 150,350 businesses, 52,270 restaurants, and 1.99 million users. The dataset is a valuable resource for researchers and businesses alike. It can be used to understand consumer behavior, identify trends, and make informed decisions, and in the current project context to build recommendation systems.

The wrangled dataset is organized into several tables. The businesses table contains information about each business, such as its name, address, phone number, and category. The reviews table contains information about each review, such as the user who wrote it, the business it is about, and the rating. The users table contains information about each user, such as their name, email address, and location.

The histogram in Figure 3.2 shows that the average ratings received by businesses are slightly skewed to the right. This means that there are more businesses with ratings of 4 stars and 5 stars than there are businesses with ratings of 3 stars and below. There are a few possible explanations for this:

- People are more likely to write positive reviews than negative reviews. This is known as the positivity bias.
- Businesses with higher ratings tend to receive more reviews. This is known as the Matthew effect.
- Some businesses are simply better than others. This could be due to the quality of their products or services, the customer service they provide, or their location.

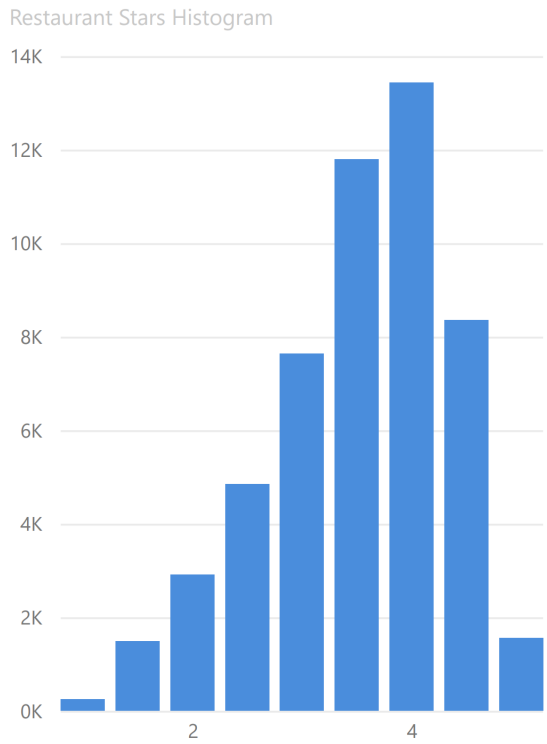

Figure 3.2: Average stars distribution received by all businesses

It is important to note that the histogram only shows the average rating for each business. It does not take into account the number of reviews each business has received. A business with a high average rating may have only received a few reviews, while a business with a low average rating may have received many reviews. It is therefore important to consider the number of reviews when evaluating a business's rating.

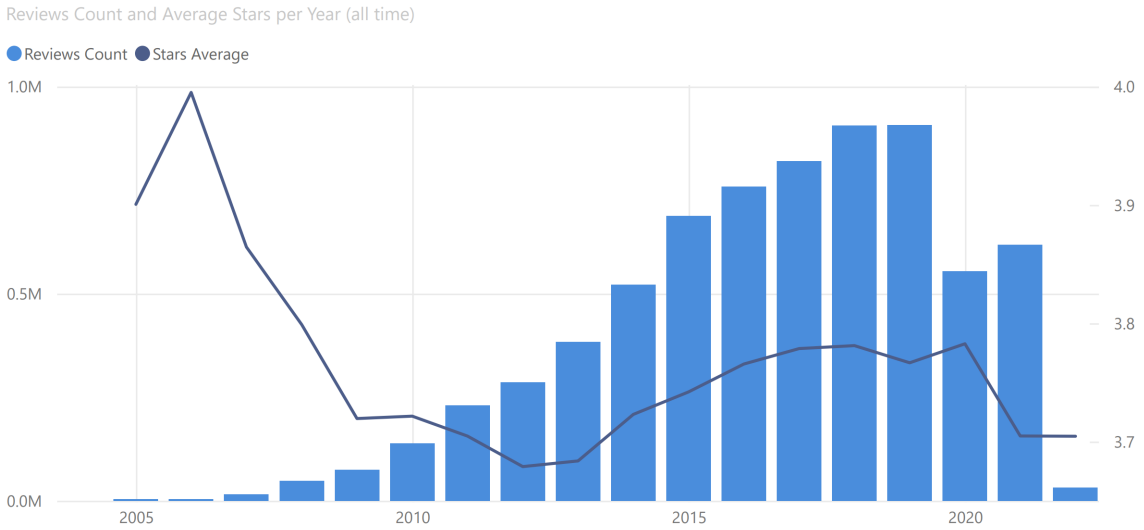

Figure 3.3: Number of Reviews and Average Rating Received by Businesses Each Year

This Figure 3.3 shows the number of reviews and average rating received by businesses

each year from 2005 to 2022. The bar chart shows the number of reviews, and the line chart shows the average rating.

As you can see, the number of reviews received by businesses has increased steadily over the years. The highest number of reviews was received in 2018 and 2019, with a big drop in 2020. This is likely due to the COVID-19 pandemic, which caused many businesses to close or operate at reduced capacity. The average rating received by businesses has also increased over the years. The highest average rating was received in 2018 and 2019, with a slight drop in 2020. This suggests that businesses have been able to maintain or improve their quality of service despite the challenges posed by the COVID-19 pandemic.

It is also worth noting that the initial average reviews in 2006 seems to be 4 out of 5 stars despite having the least number of reviews in that year. This could be due to a number of factors, such as the fact that businesses were newer and had not yet had a chance to accumulate a lot of reviews, or that customers were more likely to leave positive reviews in the early days of Yelp.

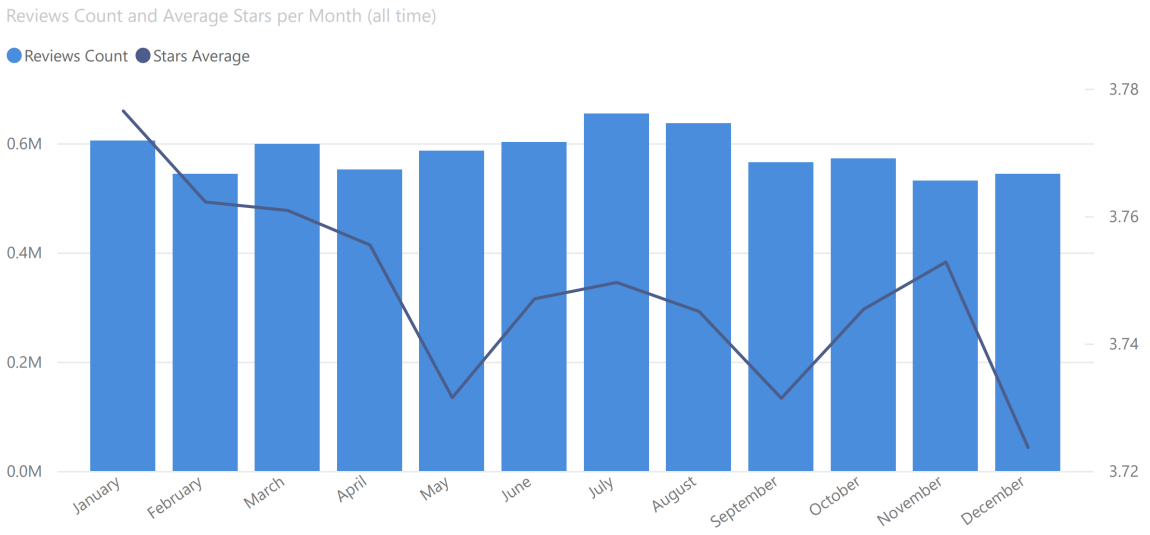

Figure 3.4: Number of Reviews and Average Rating Received by Businesses Each Month

This Figure 3.4 shows the number of reviews and average rating received by businesses each month. The bar chart shows the number of reviews, and the line chart shows the average rating.

As you can see, the number of reviews received by businesses is relatively uniform throughout the year. However, there are some slight fluctuations. For example, there are slightly more reviews in January and July, and slightly fewer reviews in February and November. The average rating received by businesses also varies slightly throughout the year. The highest average rating is received in January, while the lowest average rating is received in

December. There is also a slight spike in the average rating in November, which is likely due to the holiday season.

It is also worth noting that the average rating in January is higher than the average rating in December. This could be due to the fact that people are more likely to leave positive reviews when they are feeling good, such as after the holidays.

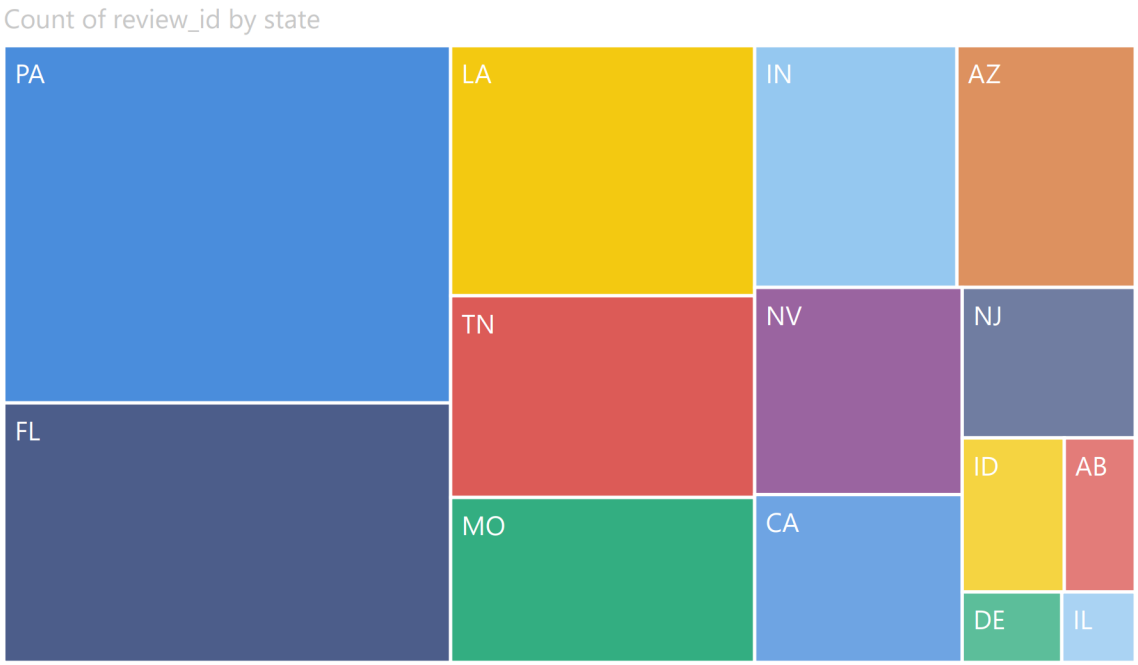

Figure 3.5: State-wise Distribution of Reviews

This final Figure 3.5 shows the state-wise distribution of reviews on Yelp. The treemap shows the number of reviews received by each state.

As you can see, the state with the most number of reviews is Pennsylvania, followed by Florida and Los Angeles. The state with the least number of reviews is Illinois.

There are a number of factors that could contribute to this distribution. For example, Pennsylvania has a large population, which means that there are more potential reviewers. Florida and Los Angeles are also popular tourist destinations, which means that there are more people who are likely to leave reviews of businesses they visit. Illinois, on the other hand, is a smaller state with a lower population, which means that there are fewer potential reviewers.

Overall, this figure shows that the distribution of reviews on Yelp is not uniform. There are some states that receive more reviews than others, which could be due to a number of factors, such as population size, tourism, and economic activity.

Chapter 4

Methodology

4.1 Techniques

The following techniques were used in this study to address the research problem of developing a recommendation system for Yelp dataset:

1. Random selection
2. Popularity based filtering
3. Collaborative filtering
    (a) User-based Collaborative Filtering
    (b) Item-based Collaborative Filtering
    (c) Matrix Factorization-based Collaborative Filtering

4.1.1 Random selection

Random recommendation is a naive approach where the system recommends items randomly to the user. It is the simplest recommendation technique that requires no knowledge of the user or the items. The idea behind this approach is to introduce some randomness in the recommendations to prevent the user from getting bored with the same items being recommended repeatedly. This technique is also useful for new users who have not provided enough information to generate personalized recommendations. Mathematically, the recommendation of items using random approach can be represented as:

Let U be the set of all users and I be the set of all items. The recommendation function $R_{random} : U \rightarrow I$ selects an item i uniformly at random from the set of all items I. Hence, for a user u, $R_{random}(u) = i$ where $i \in I$ is a random item.

The major advantage of this technique is its simplicity and easy implementation. However, the major drawback is that it does not take into account the user's preferences, which makes it less effective in generating personalized recommendations.

4.1.2 Popularity based filtering

Popularity-based recommendation is a non-personalized recommendation method that recommends the most popular items to all users. In this approach, items that are frequently rated or reviewed by users are recommended to new or existing users. This technique is suitable for new users who have no or very little data available for personalization. The most popular items are determined by counting the number of times an item has been rated or reviewed.

The popularity-based recommendation can be mathematically represented as follows: Let i be an item, u be a user, and $R(u,i)$ be the rating given by user u to item i. The popularity score $P(i)$ of item i can be calculated as the sum of all ratings given to item i by all users:

$$P(i) = \sum_{u} R(u,i) \tag{4.1}$$

Then, the items are ranked in descending order of popularity score $P(i)$. The top k items are recommended to new or existing users.

This approach is simple and easy to implement but has some limitations. For example, it does not consider individual users' preferences or interests. It also assumes that all users have the same interests and preferences, which may not be accurate in reality. Additionally, it can result in a lack of diversity in the recommended items, as popular items tend to be recommended repeatedly.

4.1.3 Collaborative filtering

Collaborative filtering[3] is a widely used technique for recommendation systems that relies on analyzing the interactions between users and items (restaurants in the current project) to predict user preferences for items they have not yet rated or purchased.

User-based Collaborative Filtering

User-based collaborative filtering (UBCF) works on the principle that users who have rated or liked similar items in the past are likely to have similar preferences in the future. UBCF predicts the rating for a given item for a user by finding other users who have rated or reviewed the item and computing the average rating or a weighted average of the ratings of the item by those users. The formula for UBCF can be expressed as follows:

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_i(u)} w_{u,v} r_{v,i}}{\sum_{v \in N_i(u)} w_{u,v}} \tag{4.2}$$

where $\hat{r}_{u,i}$ is the predicted rating for user u on item i, $N_i(u)$ is the set of users who have rated item i, $w_{u,v}$ is the similarity between users u and v, and $r_{v,i}$ is the rating of user v on item i.

Item-based Collaborative Filtering

Item-based collaborative filtering (IBCF) is similar to UBCF, but instead of finding similar users, it finds similar items. IBCF predicts the rating for a given item for a user by finding other items that the user has rated or reviewed and computing the average rating or a weighted average of the ratings of those items. The formula for IBCF can be expressed as follows:

$$\hat{r}_{u,i} = \frac{\sum_{j \in N_u(i)} s_{i,j} r_{u,j}}{\sum_{j \in N_u(i)} s_{i,j}} \tag{4.3}$$

where $\hat{r}_{u,i}$ is the predicted rating for user u on item i, $N_u(i)$ is the set of items that user u has rated, $s_{i,j}$ is the similarity between items i and j, and $r_{u,j}$ is the rating of user u on item j.

Matrix Factorization-based Collaborative Filtering

Matrix factorization-based collaborative filtering (MF) is a model-based approach that learns latent factors[1] for users and items and predicts the rating for a given user-item pair as the dot product of the user and item latent factors. MF decomposes the user-item rating matrix into two matrices, a user-latent factor matrix and an item-latent factor matrix, using techniques like Singular Value Decomposition (SVD) or Stochastic Gradient Descent (SGD). The formula for MF can be expressed as follows:

$$\hat{r}_{u,i} = q_i^T p_u \tag{4.4}$$

where $\hat{r}_{u,i}$ is the predicted rating for user u on item i, q_i is the item latent factor vector for item i, and p_u is the user latent factor vector for user u.

[1] Latent factors in matrix factorization refer to the unobserved factors that are used to estimate user-item ratings. These factors are derived from the matrix decomposition and represent underlying characteristics or features of users and items that are not explicitly stated in the data.

In this project, memory-based collaborative filtering is performed using the k-Nearest Neighbors (KNN) algorithm and model-based collaborative filtering using the SGD[1] algorithm. These techniques are chosen because they are well-established and have shown promising results in previous studies on recommendation systems.

4.2 Procedure

In order to address the business problem and achieve the objectives of the project, a structured approach was employed using the Cross-Industry Standard Process for Data Mining (CRISP-DM) process as shown in the Figure 4.1. The following steps were taken:

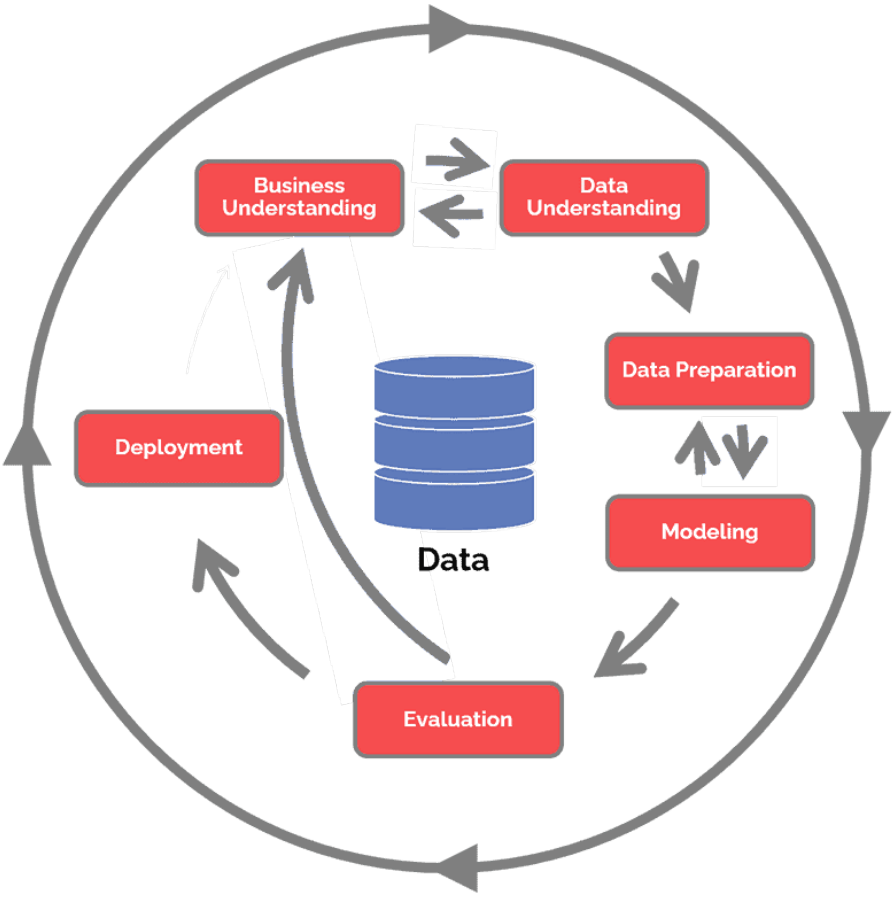

Figure 4.1: Cross-Industry Standard Process for Data Mining

- Business Understanding: The first step involved understanding the business problem and objectives. The aim of the project was to build recommendation systems for restaurants in Pennsylvania, Florida, and Los Angeles, USA, with the goal of providing personalized recommendations to customers searching for restaurants. The objectives were to develop and compare the performance of different collaborative

filtering techniques, build recommendation systems based on past customer interactions, and evaluate the performance of these systems using various metrics.

- Data Understanding and Data Preparation: The second and third steps involved gathering and understanding the data. In the data understanding phase, the Yelp public dataset was obtained, which included information on businesses, users, and reviews, as well as tips and check-ins. The dataset was provided in the form of multi-level JSON objects, which were challenging to work with directly. To address this, PySpark was used to reshape the data into single-level CSV files and pushed it into a PostgreSQL database. During the exploration of the data, sparsity was seen in the user-business rating matrix, so applied filters to the dataset to reduce it. These filters included limiting the ratings to users and businesses with at least ten ratings in the database and businesses that were categorized as restaurants.
- Modeling: The fourth step involved building models based on the prepared data. Collaborative filtering techniques, including memory-based and model-based methods, were employed to generate personalized recommendations for customers based on their past interactions with restaurants. The performance of the models was evaluated using various metrics such as RMSE, MAE, MAR@K, and Catalog Coverage.
- Evaluation: The fifth step involved evaluating the performance of the models. The models were assessed based on their ability to provide personalized recommendations to customers and their potential impact on restaurant catalog businesses. The performance of the models was compared using the metrics mentioned above, and the most effective technique was identified.
- Deployment: The final step involved deploying the models in a production environment. In this project, the deployment entailed integrating the models into hypothetical restaurant catalog businesses and providing personalized recommendations to customers searching for restaurants.

By following this structured approach, I was able to achieve its objectives in a rigorous and systematic manner.

Chapter 5

Results and Analysis

The collaborative filtering models that were evaluated include both memory-based modles such as K-Nearest Neighbor using User-User similarity and Item-Item similarity, and model-based models such as SVD, NMA, ALS and SGD. The best performing model was found to be the SGD model, which achieved the lowest RMSE and MAE scores of 1.109 and 0.868 respectively. The RMSE and MAE values for all the models are as follows:

Table 5.1: MAE and RMSE scores for different models

Model	MAE	RMSE
KNN User-User	0.89	1.175
KNN Item-Item	0.875	1.162
ALS	0.886	1.115
SGD	0.868	1.109
SVD	0.876	1.115
NMF	1.093	1.571

As you can see in Table 5.1, the Stochastic Gradient Descent (SGD) performed best in the initial scored. The SGD is a popular optimization technique used in machine learning for training various models, including collaborative filtering models. It works by iteratively updating the model parameters using a gradient of the loss function with respect to the parameters. Unlike batch gradient descent, which updates the parameters using all the training data at once, SGD updates the parameters using a randomly selected subset of the training data (a mini-batch) at each iteration. This makes it computationally efficient and able to handle large datasets such as Yelp Dataset.

In the context of collaborative filtering, SGD is used to learn the latent factors (features) that represent the user-item interactions. These latent factors are learned by minimizing

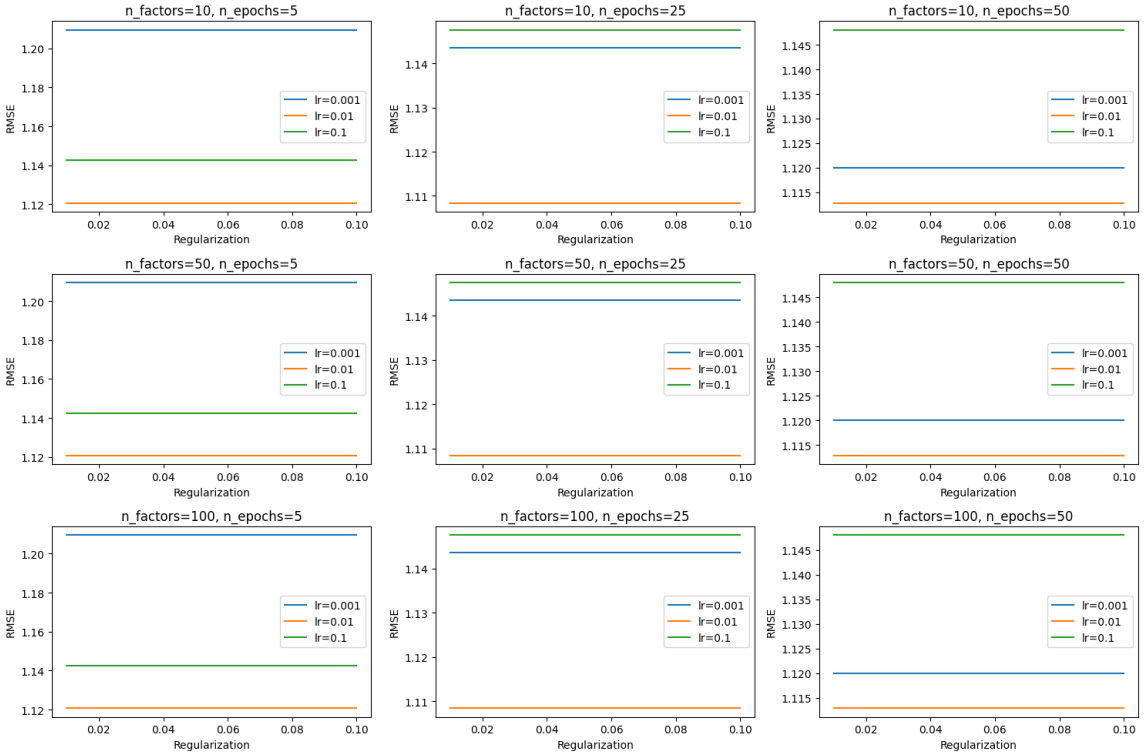

Figure 5.1: SGD tuned for best RMSE value

the difference between the predicted ratings and the actual ratings in the training data, RMSE in the current case. The model hyper-parameters, such as the number of epochs, user regularization strength, and item regularization, were tuned using cross-validation to find the values that give the best performance. The best RMSE value of 1.108 for SGD was achieved at the following hyper-parameters – $learning\_rate : 0.01, regularization : 0.01, n\_epochs : 25, n\_factors : 10$. The tuning chart can see seen in Figure 5.1. Thus, the latent factors found by SGD are 10, this would make storing the User-Restaurant ratings in memory less expensive.

Using the tuned SGD model as final Collaborative Filtering model, the following Figures 5.2 and 5.3 shows the performances of the Random, Popularity and Collaborative filtering based recommendation systems.

The Figure 5.2 shows the mean average recall @ K vs K for three recommendation systems: Random, Popularity, and Collaborative Filtering. Recall is a measure of how many of the items that a user actually likes are recommended to them. K is the number of items recommended to each user. As you can see, the Random recommendation system has the lowest recall for all values of K. This is because it simply recommends items at random, and there is no guarantee that any of the items will be of interest to the user. The Popularity recommendation system has a slightly higher recall than the Random system, but it is still

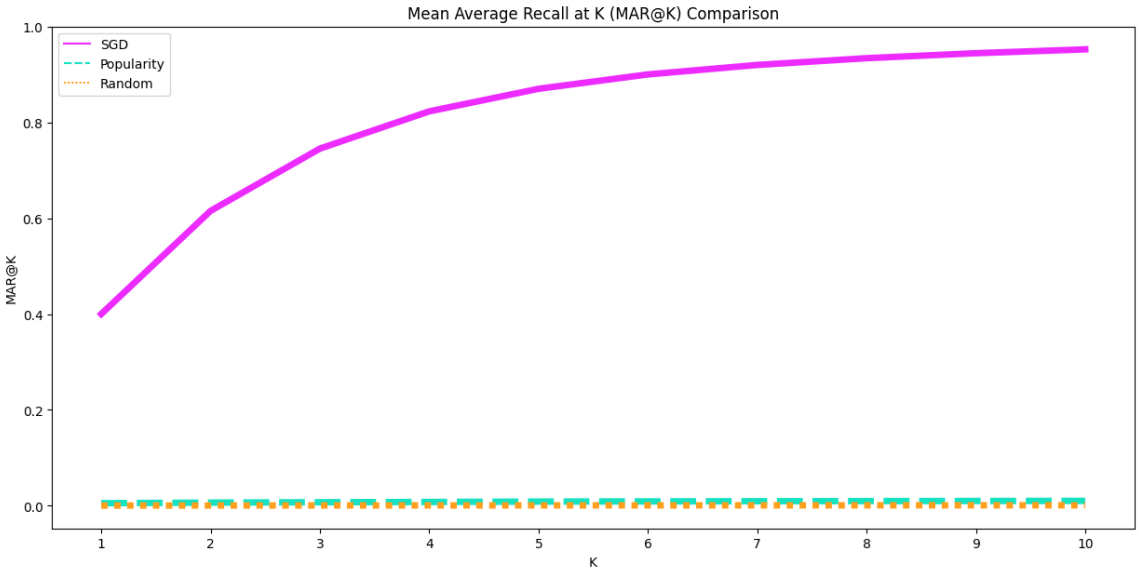

Figure 5.2: Mean Average Recall at k for the 3 recommendation systems

not good. This is because it simply recommends the most popular items, which may not be of interest to all users. The Collaborative Filtering recommendation system has the highest recall of all three systems. This is because it takes into account the user's past behavior to recommend items that they are likely to like. Overall, the Collaborative Filtering recommendation system is the best choice for most users. It is more likely to recommend items that the user will actually like, and it is more personalized than the other two systems.

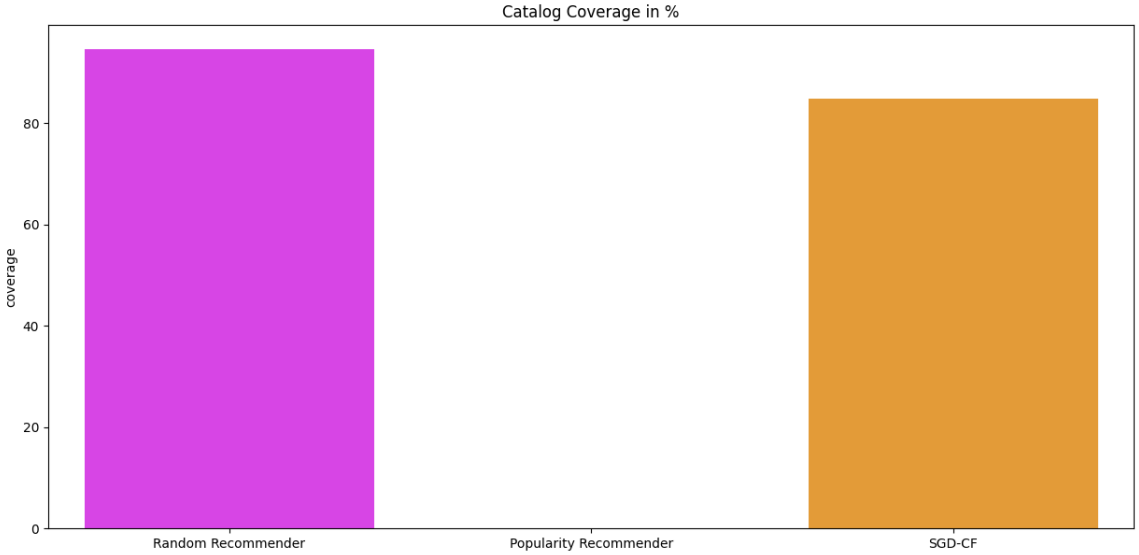

Figure 5.3: Catalog Coverage for the 3 recommendation systems

The bar chart in Figure 5.3 compares the coverages for all three recommendation systems: Random, Popularity, and Collaborative Filtering. Coverage is a measure of how many of the items in the dataset are recommended by each system. As you can see, the Random

recommendation system has the highest coverage. This is because it simply recommends items at random but there is no guarantee that any of the items will be relevant to the users. The Popularity recommendation system has a lowest coverage, because it simply recommends the most popular items. The Collaborative Filtering recommendation system has the second highest coverage of all three systems. This is because it takes into account the user's past behavior to recommend items while also showing new and relevent items to the users.

Overall, the Collaborative Filtering recommendation system is the best choice for most users. It is more likely to recommend items that the user will actually like, and it has a higher coverage resulting in diverse set of recommendations.

Chapter 6

Deliverables

The deliverables of the project encompass a set of technical outcomes that have been achieved through the development and implementation of various recommendation models. These models have been evaluated based on performance metrics, including RMSE and MAE scores and MAP@k and coverage.

Through this project, the primary problem of providing accurate and personalized recommendations to users has been addressed. The technical deliverables provide insights into the performance of various models and offer recommendations for selecting the best model for a given dataset.

The impact of the project outcomes on research objectives has been significant. The project has contributed to recommender systems by developing effective recommendation models and provided practical solutions for personalized recommendation problems. Furthermore, the project has also provided insights into the factors that impact the performance of recommendation models, which could inform future research in the field.

From a business perspective, the project has significant implications for companies that rely on recommendation systems for their revenue streams. Businesses can improve customer satisfaction and increase sales by providing accurate and personalized recommendations to users. Therefore, the project outcomes can improve the profitability and sustainability of businesses relying on recommendation systems.

Appendix A

Self Assessment

My individual learning objectives for this project were to:

- Learn how to build recommendation systems using different techniques.
- Evaluate the performance of different recommendation systems.
- Apply data science skills to a real-world problem.

I believe that I accomplished all of my learning objectives. I was able to build recommendation systems using random, popularity, and collaborative filtering techniques. I was also able to evaluate the performance of these systems using metrics such as root mean square error (RMSE), mean absolute error (MAE), mean absolute recall at k (MAR@k), and catalog coverage. Finally, I was able to apply data science skills to a real-world problem by building a recommendation system for restaurants in Pennsylvania, Florida, and Los Angeles, USA.

The most useful DSA skills in this project were:

- Data wrangling
- Data cleaning
- Exploratory data analysis

I had to learn independently the following skills to complete this project:

- How to build data dashboards on Power BI
- How to use PySpark along with Google Colab for Big Data handling
- How to build recommendation systems using Python
- How to evaluate the performance of recommendation systems

This practicum was for 4 credit hours. It was an unpaid internship. My supervisor was Mr. Triet Tran Minh, a PhD Candidate in Data Science and Analytics at the University of Oklahoma. His contact information is triet.m.tran-1@ou.edu.

Bibliography

[1] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–9, 2011.

[2] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[3] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. *The adaptive web: methods and strategies of web personalization*, pages 291–324, 2007.