Erin Ballar

Restaurant Recommendation System

Introduction

As a direct result of the information age, review platforms featuring user-generated feedback have become an essential part of the customer decision-making process. With so much information available within a platform, it often becomes difficult to find a business that fulfills one's needs and individual preferences. This could be detrimental for a review platform, as users' inability to find any valuable businesses could result in a decrease in user retention and online foot traffic. Therefore, it is crucial for a platform to innovate and take advantage of the readily available data the platform generates. By leveraging the wealth of data available that is contributed by users, we propose the implementation of a recommendation system. A recommendation system will help users discover businesses that are specifically catered to their tastes. When users are able to successfully use the platform and find what they need, it leads to an increase in user satisfaction that builds the trust needed for user retention. Additionally, it encourages users to contribute reviews to the platforms themselves, fostering greater user involvement. This proposal is not without evidence. Amazon's recommendation system generates 35% of the e-commerce giant's total revenue (MacKenzie et al., 2013). Similarly, Netflix has also found success, with 80% of the hours streamed on their platform being derived from their recommendation system (Singh, 2020). In this paper, we will explore popular machine learning methods for creating recommendation systems, and develop a restaurant recommendation system.

Literature Review

Personalized Book Recommendation System using Machine Learning Algorithm

The authors Sarmal, Mittra, and Hossain (2021) highlight common methods to create recommendation systems including collaborative, content-based, hybrid, and cross domain filtering algorithms. To create a book recommendation system, they took a hybrid approach by using k-means clustering with cosine distance and cosine similarity to produce e-book recommendations for users.

E-commerce Platform Based on Machine Learning Recommendation System

This paper discusses the importance of having a recommendation system in the e-commerce industry and proposes a system that mimics a personal stylist using content-based filtering techniques (Tahir et al. , 2021). Content-based filtering is a method that analyzes a user's profile and online activity to generate personalized recommendations. Through their clothing recommendation system, they discovered that color similarities in clothes more accurately predicted user's preferences than user's purchase history.

Recipe Recommendation System using Machine Learning Models

With data scraped from Yummly.com, the authors Maheshwari and Chourey (2019) created a recipe recommendation system that was able to find ingredient pairings and substitutions of different cuisines. They achieved this with TF-IDF (Term Frequency-Inverse Document), which is a mathematical tool that calculates the importance of a word in a document. Once calculated, the TF-IDF scores were used to help calculate the cosine similarity of two ingredients. To find ingredient alternatives, the authors used Word2vec, which converts words into vectors that are able to capture the relationships between words.

Personalized Travel Recommendation System: A Study of Machine Learning Approaches in Tourism

This study investigates the potential of machine learning applications in the travel industry, with a focus on enhancing travel planning. The authors, Badouch and Boutaounte (2023) outline the various stages of planning a vacation and identify the most suitable models. For example, they recommend using

Markov models to predict sequence of destinations, which essentially creates an itinerary, or neural networks to provide recommendations based on past travel patterns.

Restaurant Recommender System Using User-Based Collaborative Filtering Approach

Using collaborative filtering methods the authors, Alif Azhar Fakhri, Z K A Baizal, and Erwin Budi Setiawan (2019) create a restaurant recommendation system. Collaborative filtering is a method that generates recommendations for users based on the preferences and behaviors of similar users. To identify similar users, the authors used the Pearson correlation formula to calculate the similarities between users, and then applied k nearest-neighbors to find the top N similar users.

Method

We will develop and evaluate our restaurant recommendation system using two different collaborative filtering approaches: K nearest-neighbors and co-clustering. K nearest-neighbors for collaborative filtering is a technique that finds relationships between users by measuring distance or similarity between vectors within a space. While traditional k nearest-neighbors models use distance measures to find the nearest neighbors, for this project we will be using cosine similarity. Cosine similarity is measured by finding the cosine angle between two vectors within a space. It can be calculated with the following formula:

$$similarity(A, B) = cos(\theta) = \frac{A \cdot B}{||A||\,||B||}$$

Where θ is the angle between vectors, $A \cdot B$ is the dot product of A and B, and $||A||$ is the L2 norm of A (Karabiber, n.d.).

Co-clustering for collaborative filtering is an unsupervised learning technique that simultaneously finds clusters between users and between items. The model computes centroids by determining the average within each cluster and subsequently evaluates the error. If the error is too large, the model will continue to repeat the process, refining the clusters until convergence or achieving a low error. By identifying these clusters, it reveals relationships between user clusters and item clusters. These relationships are then used by the model to generate its recommendations.

Implementation & Experiment Set Up

The dataset used for our recommendation system is the Yelp dataset, available free online for academic purposes. This data includes information on businesses, reviews, users, check-ins, and photos. Due to the dataset being extremely large (over 6 million rows), it was subsetted to only include restaurants in California that are in business.

To familiarize ourselves with the data we were working with, we performed exploratory data analysis. First, we examined the distribution of businesses' averaged ratings. As shown in Figure 1, we can see that most businesses have an average rating between 3.8 and 4.8 stars (out of 5). This was much higher than expected, as we predicted it to have a more normal shaped distribution centered around 3.0 stars.

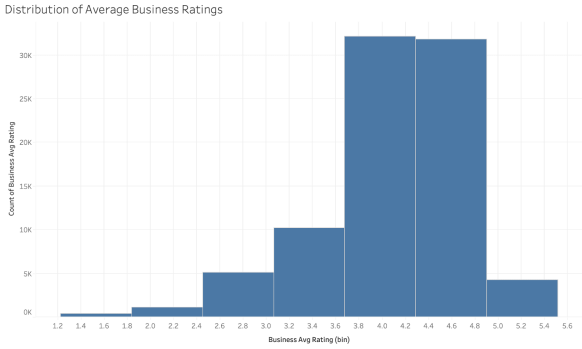

Figure 1. Distribution of Average Business Ratings

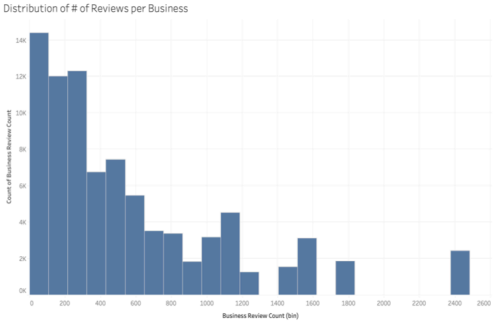

Figure 2. Distribution of Number of Reviews per Business

In the following histogram (Figure 2), we discover that most businesses have zero reviews. For our restaurant recommendation system, we decided to not include any of these unreviewed businesses as it could potentially impact the model. It was also decided to remove any businesses rated less than four stars, because we want our system to only recommend above average restaurants. The average number of reviews made by users (Figure 3) presents a similar distribution with many users having not made any reviews. These users were filtered out, as a collaborative filtering model cannot find recommendations for a user without existing reviews. The last visual (Figure 4) we examine is on Yelp users' average ratings. From this visual we discover that most users rate businesses a 4.0, which we found surprising. We had assumed Yelpers would be much more critical in their ratings. This discovery could potentially explain our first graph looking at business' average ratings.

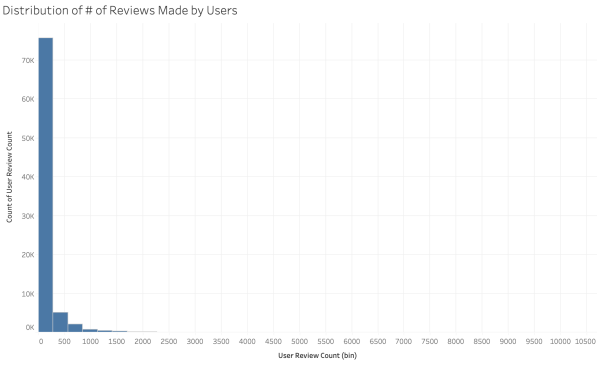

Figure 3. Distribution of Number of Reviews Made By Users

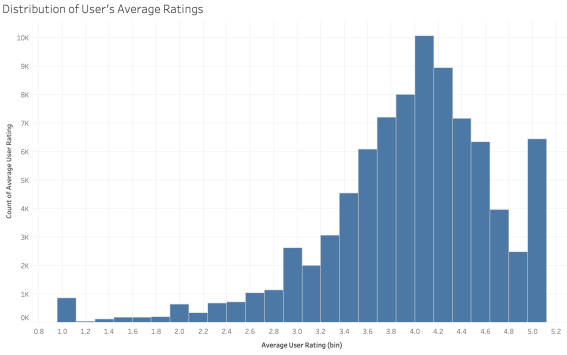

Figure 4. Distribution of User's Average Ratings

For machine learning implementation we used the Python Scikit Surprise library, a versatile library that enables users to create their own recommendation system using collaborative filtering. In the Surprise library, there are four different k-nearest neighbors models we will be testing with: `KNNBasic()`, a standard k nearest-neighbors approach; `KNNWithMeans()`, which standardizes ratings by taking the average rating of each user; `KNNWithZScore()`, which uses z-score normalization; `KNNBaseline()`, which incorporates baseline estimates for ratings. To incorporate cosine similarity, we will adjust the model parameter, `sim_options`, to `'cosine'`. For co-clustering, there is only one basic model in the library, `CoClustering()`. We used the default parameters for this model since it is computationally expensive.

To find the best model, we performed k-fold cross validation with all models to check the root mean square errors and mean absolute errors. In cross-validation, we split the data into k folds. For each iteration, we use one of the folds as the test set and the remaining $k - 1$ folds as the training set. After completing the cross validation tests, we picked the model with the lowest average scores as our final model.

Results

The five-fold cross-validation results for all models yielded very close average scores. Among the five models tested, the k-nearest neighbors with baseline model exhibited the lowest scores, with an average root mean squared error of 1.1233 and average mean absolute error of 0.834. On the other hand, the co-clustering model performed the worst in terms of accuracy. Yet, despite its inferior performance, the co-clustering model demonstrated the quickest fitting and testing times compared to all the other models. While we will proceed using the k nearest-neighbors with baseline model due to its superior accuracy, it's worth noting that with additional optimization, co-clustering has the potential to out-perform all the other models.

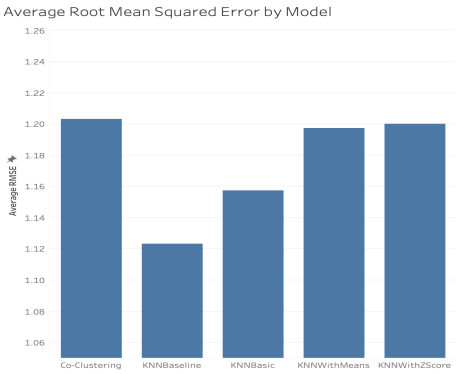

Figure 5. Average Root Mean Squared Error by Model

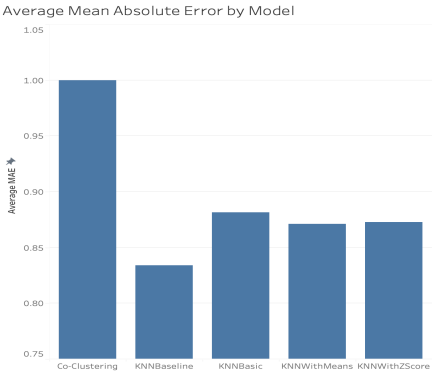

Figure 6. Average Mean Absolute Error by Model

Figure 7. Average Fit & Test Time by Model

Conclusion

Recommendation systems hold the power to improve the user experience on review platforms, which results in increased user retention and engagement. This power extends beyond the review platform industry, benefiting other industries such as the e-commerce and streaming industry. Through our restaurant recommendation system, we discovered that the k nearest-neighbors with baseline model excelled in accuracy, while the co-clustering model demonstrated the best fit and test times. With a machine with greater computing capacity, we recommend further testing with data that encompasses a broader region beyond California. Additionally, the recommendation system could be further improved by developing a method that allows the models to recommend new businesses with potential despite their biased averaged ratings due to limited reviews. Lastly, we highly recommend testing the co-clustering model with different parameters to see if it yields better error scores than with the default settings. With these additional enhancements, our recommendation system would become even more capable of providing high-quality recommendations across various domains.

Example Output by Restaurant Recommendation System:

```
Hi Erin! Here are some food spots we think you would like:
   1. The Black Sheep
   2. Empty Bowl Gourmet Noodle Bar
   3. Uncorked Wine Tasting and Kitchen
   4. Buena Onda at Mosaic Locale
   5. In-N-Out Burger
```

Works Cited

Badouch, Mohamed, and Mehdi Boutaounte. “Personalized Travel Recommendation Systems: A Study of Machine Learning Approaches in Tourism.” *April-May 2023*, vol. 3, no. 33, 26 Apr. 2023, pp. 35–45, https://doi.org/10.55529/jaimlnn.33.35.45.

Fakhri, Alif Azhar, et al. “Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region.” *Journal of Physics: Conference Series*, vol. 1192, Mar. 2019, p. 012023, https://doi.org/10.1088/1742-6596/1192/1/012023.

Karabiber, Fatih. “Cosine Similarity.” *Www.learndatasci.com*, www.learndatasci.com/glossary/cosine-similarity/. Accessed 3 June 2024.

MacKenzie, Ian, et al. “How Retailers Can Keep up with Consumers.” *McKinsey & Company*, 1 Oct. 2013, www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers. Accessed 3 June 2024.

Maheshwari, Suyash, and Manas Chourey. “Recipe Recommendation System Using Machine Learning Models.” *International Research Journal of Engineering and Technology*, vol. 6, no. 9, Sept. 2019.

Pandian, Shanthababu. “K-Fold Cross Validation Technique and Its Essentials.” *Analytics Vidhya*, 17 Feb. 2022, www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/.

Sarma, Dhiman, et al. “Personalized Book Recommendation System Using Machine Learning Algorithm.” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021, https://doi.org/10.14569/ijacsa.2021.0120126.

Spandana Singh. “Why Am I Seeing This?” *New America*, 25 Mar. 2020, www.newamerica.org/oti/reports/why-am-i-seeing-this/. Accessed 3 June 2024.

Tahir, Muhammad, et al. *E-Commerce Platform Based on Machine Learning Recommendation System*. Research Gate, Nov. 2021.