Team88: Personalized Restaurant Recommendations based on Yelp User Profiles

Jeffrey Rodriguez (jrodriguez396), Kristine K. Tran (ktran311), Prateek Shukla (pshukla64)

Joshua Cooper (jcooper301), Edmund Tan (etan47), Sahil Suri (ssuri8)

1 INTRODUCTION

The present work introduces a collaborative filtering-based web application that provides personalized restaurant recommendations to users, leveraging a publicly available Yelp dataset. Yelp, with its impressive reach of 73 million unique visitors on desktop and mobile and 6.3 million active claimed local business locations as of February 2023, represents an unparalleled source of information on consumer preferences in the restaurant industry. Our application aims to generate customer value by connecting users with businesses loved by similar people and helping businesses reach more customers who love their goods or services.

The application's front-end has been meticulously designed to offer users an intuitive and user-friendly interface, allowing them to select cuisine, type of restaurant, and location to generate recommendations based on the preferences of similar users. We utilize the maximum accuracy rating prediction algorithm within the open-source 'Surprise' library to generate predicted ratings for restaurants that the user has not rated, resulting in a list of top recommendations based on top ratings. We split the data into training and test datasets and evaluate the accuracy of our predictions by measuring the prediction error rates for the test data.

We evaluated the effectiveness of our collaborative filtering-based approach for restaurant recommendations through a proof-of-concept study and user survey. The study showed our algorithm delivers highly accurate and personalized recommendations with low prediction error rates, and the survey results were overwhelmingly positive. Our approach leverages publicly available data and contributes to collaborative filtering research. These findings have important implications for future recommendation algorithms in various domains.

2 LITERATURE SURVEY

In recent years, the research topic of recommending restaurants to users has garnered increasing attention, leading to the development of various recommendation methods. These methods aim to provide personalized recommendations based on user preferences to improve the recommendation process (Lops et al. 73). Content-based filtering is a well-known approach that recommends similar restaurants based on the user's previous restaurant ratings and restaurant features. However, this method is limited by its ability to only recommend restaurants within the interests already known about the user.

Collaborative filtering is another popular approach that uses similarities between users and items to provide recommendations simultaneously (Mohan et al. 8-es; Lee and Kim; Guo et al. 1; He et al. 173-182; Suganeshwari and Syed Ibrahim). This method recommends a restaurant to a user based on the ratings of similar users who have liked that restaurant. Several methods have been developed to improve collaborative filtering, including matrix factorization, neighborhood-based collaborative filtering, and hybrid collaborative filtering (Mohan et al. 8-es).

Clustering is a method that has been used to group users with similar restaurant ratings into clusters and recommend restaurants based on group ownership (Zhang et al. 2018). This method is similar to collaborative filtering but focuses on grouping users into clusters instead of looking for similarities between users.

Yelp, a well-known platform for restaurant recommendations, currently does not recommend businesses based on the similarity of other users. Yelp's recommendation system uses the user's search terms, distance, ratings, user engagement data, and reviews of relevant businesses to provide recommendations (Hu and Liu 1; Asani et al. 100114).

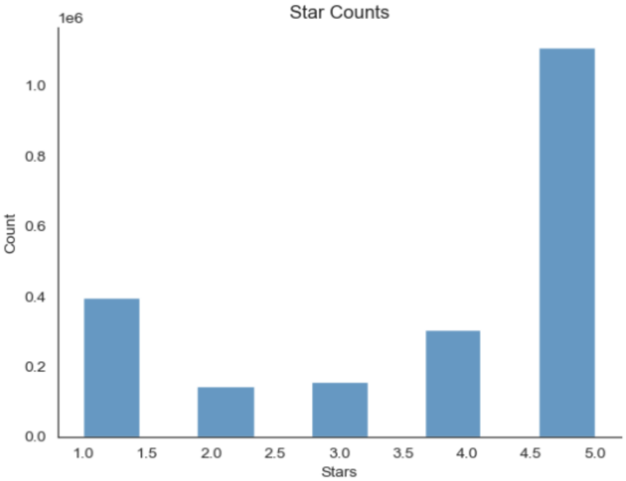
Our proposed approach to recommending restaurants on Yelp is innovative and personalized, utilizing user ratings to build profiles and group users into similar profile groups. In contrast to content-based and collaborative filtering methods, Lops et al's approach uses machine learning algorithms and a large dataset to provide accurate and robust recommendations (Lops et al. 73; Mohan et al. 8-es; Lee and Kim; Guo et al. 1; Rendle et al. 239). By using this approach, we can provide more personalized recommendations to Yelp users, making it easier for them to find restaurants that match their preferences.

We will be using sentiment analysis in this research to analyze the tone and emotion of user reviews and provide recommendations based on the user's sentiment towards a restaurant (Hu and Liu; Sun 2020). Additionally, natural language processing techniques will be utilized to extract relevant information from user reviews, providing valuable insights into user preferences and sentiment towards a restaurant (Sun et al. 2017).

Our web application with visualized data at its center offers a user-friendly interface for exploring recommendations, setting it apart from existing systems that output a list of recommendations. This feature makes our application a valuable tool for Yelp users looking for personalized and accurate restaurant recommendations. We are confident that our proposed approach, which incorporates sentiment analysis-based recommendations, will provide significant benefits to Yelp users and enhance the overall user experience of the platform.

3 PROPOSED METHOD

This section outlines our approach to addressing the limitations of the Yelp Dataset + platform that emerged during our exploratory analysis. The most notable limitation is the skewed and unreliable nature of the star rating system, which is the primary metric that users consider when evaluating Yelp's recommended restaurants. Specifically, we observed that over 50% of ratings are five stars, and middle ratings (2, 3, and 4 stars) are used infrequently. Additionally, most users tend to rate restaurants either highly positively or negatively, further exacerbating the skewed distribution. Another limitation we identified is the inadequacy of Yelp's review labeling system, which categorizes reviews as "useful," "funny," or "cool." However, these categories provide little guidance to users when making decisions, and we propose alternative tools that can more effectively assist users in understanding user reviews.



Our research aims to develop a novel approach to restaurant recommendations that overcomes the limitations of the Yelp Dataset and platform. We propose a two-fold solution to improve the current state-of-the-art approaches: utilizing collaborative filtering at scale and presenting sentiment scores of reviews to users. Our approach will employ sentiment analysis to generate a sentiment score for each restaurant, providing a more nuanced grading system. Collaborative filtering algorithms will be used to generate personalized recommendations for users by identifying similar users and recommending highly rated restaurants. We will innovate in two areas by scaling these algorithms to a large dataset and combining collaborative filtering and sentiment analysis.

Our dataset is significant, with almost 2 million users and 150,000 businesses, and our approach will employ well-known algorithms for collaborative filtering and sentiment analysis. We will benchmark at least five collaborative filtering recommendation algorithms, using KNN collaborative filtering, the Slope One algorithm, and a Baseline algorithm. Our choice of algorithm will balance scalability and model performance. The experiments section will provide a detailed description of our testing methodology, including how we validate and select the most appropriate algorithm. Our approach will contribute to the wider research on collaborative filtering techniques and their applications in the domain of restaurant recommendations.

We have selected the KNN collaborative filtering algorithms for their simplicity and intuitive nature, as they create a neighborhood of similar users to generate recommendations. In our testing, we will evaluate four KNN algorithms: KNNBasic, KNNWithMean, KNNWithZScore, and KNNBaseline.

- KNNBasic - which creates a prediction by aggregating the predictions of all similar users and dividing by the number of users.
- KNNWithMean - adds onto the method of KNNBasic by taking into account the mean ratings of each user.
- KNNWithZScore - modified KNNBasic by subtracting the mean and dividing by the variance for each user before aggregating user scores.
- KNNBaseline - adds onto the method of KNNBasic by adding a baseline rating for each user. A baseline rating is calculated for each user in the neighborhood and the user requesting the predicted rating. A Baseline rating attempts to capture the effect that some users may systematically rate all items higher or lower than others.

We will compare these four KNN algorithms with another Collaborative Filtering algorithm called Slope One, which is calculated using the formula:

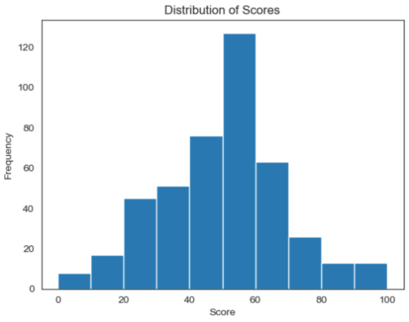
$$\hat{r}_{ui} = \mu_u + \frac{1}{|R_i(u)|} \sum_{j \in R_i(u)} \text{dev}(i, j), \tag{1}$$

where $\text{dev}(i, j) = \frac{1}{|U_{ij}|} \sum\limits_{u \in U_{ij}} r_{ui} - r_{uj}$.

In this study, we define $R_i(u)$ as the set of items that have been rated by the user requesting recommendation, which shares at least one common user with the set of items to be rated. Additionally, we introduce dev(i, j) as the average difference between the ratings of items to be rated and those already rated by the user.

To compare the effectiveness of different recommendation algorithms, we will employ a training dataset to train each algorithm and subsequently deploy them on a test dataset. The test dataset contains actual ratings of the items from the user, which enables us to evaluate the model's performance by calculating Root Mean Square Error (RMSE).

Our aim is to identify the best-performing model for our data, and we will accomplish this by selecting the algorithm with the lowest RMSE. Once the most effective model is chosen, we will further optimize its performance through a tuning process. The outcome of this research will provide insights into the most suitable recommendation algorithm for the given dataset.



```
Mean: 49.980420127793536
Standard Deviation: 18.519997844248753
```

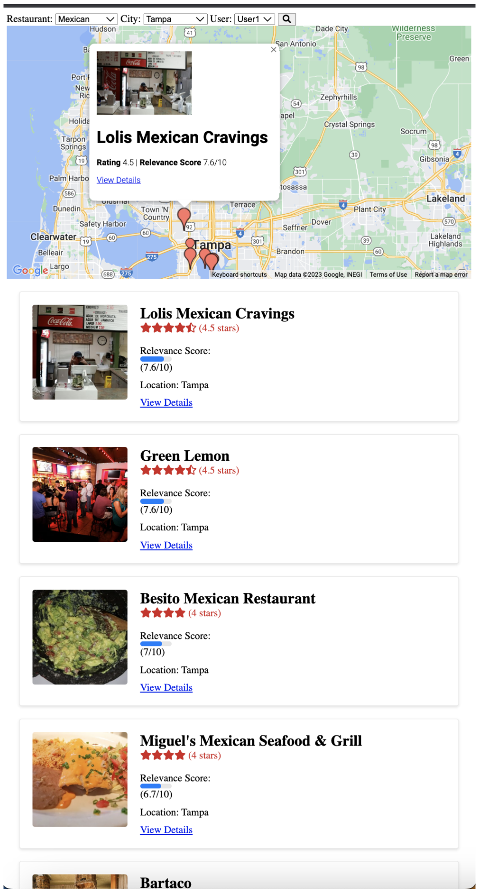
For semantic analysis, we aimed to provide a more granular metric than stars and summarize the sentiment of all the reviews with a single score per restaurant, ranging from 0 to 100, with 50 being the average. To achieve this, we utilized the nltk library and SentiWordNet for natural language processing and sentiment analysis, specifically to obtain a score for how positively people perceived a particular restaurant. Additionally, we used Textblob to determine polarity and subjectivity for each store.

Our first step was to tokenize the words from the review text with nltk and extract their tags to understand their parts of speech and grammar. We then converted the nltk tags to match the ones in the WordNet database and calculated the sentiment scores for words using SentiWordNet. Combining these scores gave us a numeric value for each review.

The second step involved grouping the review data by restaurant and aggregating the scores into a single rating for each restaurant. While looking at the distribution, we found a small subset of reviews that were 4.3 standard deviations outside the mean. A few restaurants also required more reviews to perform our sentiment analysis. Before standardizing the ratings to give us our score out of 100, we removed these outliers from the calculation and then gave them our average score, which would increase or decrease as they received genuine reviews.

We then used TextBlob with the same tokenized data to receive Polarity and Subjectivity Scores. This was important because we wanted to present more objective scores instead of using modified scoring in the ranking algorithm. The method of addressing outliers did affect the distribution of scores, creating a prominent peak at 50, but as can be seen, this is a much more realistic score than the one provided by the star system. After comparing the two algorithms, we used the TextBlob Polarity score in our recommendation algorithm and provided it to users in their recommendations of restaurants. Our tool provides a more meaningful and informative way to compare restaurants based on the text of their reviews.

For our backend, we will leverage Python and Flask to build an API that our frontend can call to obtain user recommendations and other information on a restaurant, such as reading reviews. We will use sqlite3 to provide a fast database system to store our user, restaurant, and review data. Recommendations will not be stored in the database but will be calculated on demand as users request them.



Restaurant: Mexican City: Tampa User: User1

Lolis Mexican Cravings

Rating 4.5 | Relevance Score 7.6/10

View Details

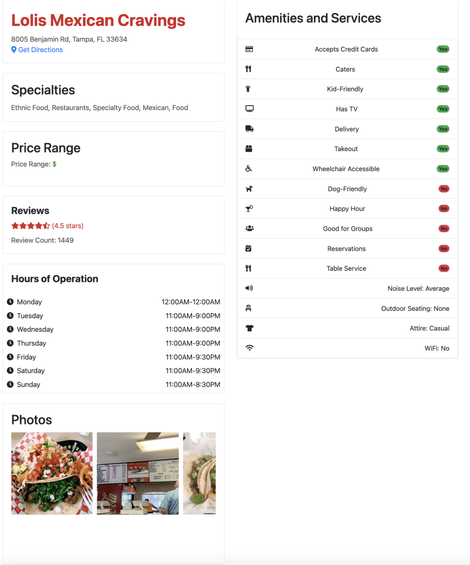
Lolis Mexican Cravings
★★★★☆ (4.5 stars)
Relevance Score:
(7.6/10)
Location: Tampa
View Details

Green Lemon
★★★★☆ (4.5 stars)
Relevance Score:
(7.6/10)
Location: Tampa
View Details

Besito Mexican Restaurant
★★★★ (4 stars)
Relevance Score:
(7/10)
Location: Tampa
View Details

Miguel's Mexican Seafood & Grill
★★★★ (4 stars)
Relevance Score:
(6.7/10)
Location: Tampa
View Details

Bartaco



Lolis Mexican Cravings

8005 Benjamin Rd, Tampa, FL 33634

Get Directions

Specialties

Ethnic Food, Restaurants, Specialty Food, Mexican, Food

Price Range

Price Range: $

Reviews

★★★★☆ (4.5 stars)

Review Count: 1449

Hours of Operation

Monday	12:00AM-12:00AM
Tuesday	11:00AM-9:00PM
Wednesday	11:00AM-9:00PM
Thursday	11:00AM-9:00PM
Friday	11:00AM-9:30PM
Saturday	11:00AM-9:30PM
Sunday	11:00AM-8:30PM

Photos

Amenities and Services

Accepts Credit Cards	Yes
Caters	Yes
Kid-Friendly	Yes
Has TV	Yes
Delivery	Yes
Takeout	Yes
Wheelchair Accessible	Yes
Dog-Friendly	No
Happy Hour	No
Good for Groups	No
Reservations	No
Table Service	No
	Noise Level: Average
	Outdoor Seating: None
	Attire: Casual
	WiFi: No

For our user interface, we have a frontend web page that features search functionality using multiple search options and a search button. The search options allow users to search for restaurant recommendations using various categories, such as a city or cuisine type. Once the user clicks on the search button, a large map loads, displaying the top recommended restaurants for the user based on their preferences. Clicking on a restaurant's icon reveals additional information about the restaurant, including sentiment analysis of its reviews. Clicking on the restaurant's entry in our list of recommendations provides a typical summary of the restaurant's details, as one would expect on a site like Yelp. The restaurant icons are sized proportionately to the user's predicted likelihood of enjoying the restaurant.

4 EXPERIMENTS & EVALUATION

Our objective is to build a restaurant recommendation system using Collaborative Filtering (CF) and Sentiment Analysis on the Yelp dataset. Our recommendations are powered by the Scikit-Surprise package. To determine the best Scikit-Surprise algorithm for our project, we performed experiments related to scalability and accuracy. We also added a Textblob sentiment score to the recommendations and manually inspected the outputs to ensure they were reasonable. Finally, we conducted experiments to understand how potential users perceive the usability of our system.

4.1 Scalability

The Yelp dataset contains 150,000 businesses, of which 1,200 are restaurants and bars with a minimum of 500 reviews. The dataset also contains 1.1 million user reviews for restaurants and bars. Our first experiment was to determine if there was an upper bound on the volume of data used to train our model. We selected a collaborative filtering model, KNNBaseline, to train using the dataset. Since our progress report, we have further refined our experiment by filtering the dataset to contain only reviews for restaurants and bars. Using an Apple Silicon M1 MacBook Pro with 16GB RAM, we performed cross-validation and obtained the following results:

Table 1: Training/Testing Times (business dataset fixed at 2800 businesses)

Size of review dataset	Training time (seconds)	Testing time (seconds)	Memory (RAM) errors
60,000	6.954	0.248	None
70,000	9.695	0.353	None
80,000	13.276	0.448	None
90,000	17.174	0.578	None
100,000	21,971	0.705	None
110,000	27.552	0.843	None
120,000	-	-	RAM limit exceeded - error

Based on observed training times, our recommendation model requires an additional 4 seconds of training time for every 10,000 reviews added to the training set. Assuming this trend continues, and given that model training happens only occasionally in an offline process, scaling for training times should not be a problem. Similarly, testing time increases by 0.1 seconds for every 10,000 reviews added, which is also not a significant issue. However, the challenge lies in the fact that our training algorithm is memory-bound, meaning larger training sets will require more RAM to run. To work around this memory consumption issue, we scaled our model training by splitting the dataset by cities, using our existing hardware. By doing so, we were able to train models for most cities without running into memory contention issues. This approach enables us to train on more of the dataset and likely achieve more accurate recommendations for each city, but the downside is that our system may do a less effective job of capturing user similarity across cities.

Further Research:

- As we trained our model using only a subset of the Yelp data, it is likely that our accuracy metrics are sub-optimal. Future research could focus on using highly scalable cloud services to optimize model training by utilizing the full dataset to improve recommendation accuracy.

- Another area of focus for future research could be finding ways to optimize the size of the training dataset. To reduce hardware memory requirements in our project, we pruned the training dataset by excluding businesses with less than 500 reviews. Future research could determine the threshold values (e.g., number of reviews per business) that would offer the best balance between dataset size and training results.

4.2 Accuracy

To provide accurate recommendations, it’s crucial to select the best collaborative filtering (CF) algorithm for the Yelp dataset. We experimented with different CF algorithms and performed 3-fold cross-validation using a dataset of approximately 1,200 businesses and 60,000 reviews to obtain accuracy metrics (Root Mean Squared Error). Our findings are as follows:

Table 2: Accuracy Metrics of different Collaborative Filtering Algorithms

Algorithm	RMSE	Train time (seconds)	Test time (seconds)
KNNBaseline	1.180	13.227	0.259
KNNBasic	1.228	12.933	0.251
KNNWithZScore	1.260	13.484	0.260
KNNWithMeans	1.261	13.158	0.252
SlopeOne	1.261	0.123	0.052
NormalPredictor	1.591	0.021	0.071

Based on the results, all KNN algorithms outperform our control NormalPredictor algorithm, with the KNNBaseline algorithm performing the best (lowest RMSE). We further optimized the KNNBaseline algorithm by using GridSearchCV to tune its hyperparameters. GridSearchCV returned the following optimal set of hyperparameters for KNNBaseline:

```
{
    'bsl_option': {
        'method': 'als'
    },
    'k': 3,
    'sim_options': {
        'name': 'pearson_baseline',
        'min_support': 5,
        'user_based': True
    }
}
```

We used the optimal set of hyperparameters to instantiate KNNBaseline and ran it through a 3-fold cross-validation. Additionally, we instantiated an unoptimized KNNBaseline and ran it through a 3-fold cross-validation. The results are as follows:

Table 3: Optimized KNNBaseline vs Un-optimized Using the Same Dataset

Algorithm	RMSE	Train time (seconds)	Test time (seconds)
Optimized KNNBaseline	1.158	19.086	0.824
Unoptimized KNNBaseline	1.173	14.215	0.930

The hyperparameter tuning has resulted in a slight improvement in the Root Mean Squared Error (RMSE) metric, thereby instilling more confidence in the results produced by our recommendation system. The increased training time, although a tradeoff, is justified given the lowered RMSE. In addition to the Scikit-Surprise predictions, our recommendation system incorporates sentiment analysis. We utilized the Textblob package to analyze the sentiment of the review text in our training dataset and assign a polarity score to each review. The polarity score ranges from -1 (highly negative) to 1 (highly positive) and enables us to capture the positive or negative experiences of customers with a business. Our system combines the Scikit-Surprise prediction and the polarity score by using a weighted system, as follows:

Recommendation score = (0.7 ×Scikit-Surprise prediction)+ (0.3 × polarity score)

To evaluate the effectiveness of our recommendation system, we performed a manual inspection of the recommendations generated for a user in our test dataset. Specifically, we selected the user ’ET8n-r7glWYqZhuR6GcdNw’ and examined the top-10 recommendations provided by our system. Of these recommendations, the user had already visited 8 restaurants and had not yet visited the remaining 2. Interestingly, the two unvisited restaurants were highly rated and served international cuisines - French and Italian. This is noteworthy as the user had rated other restaurants serving international cuisine such as Greek, Indian, and Middle Eastern very highly. Additionally, in the training dataset, the user had written a comment expressing their appreciation for a cafe with a French name. After inspecting the user’s comments, we found no evidence to suggest that French or Italian food would be inappropriate recommendations for the user. Although this is an anecdotal observation, it provides us with confidence that our system’s recommendations appear to be suitable for this particular user.

Future research:

- Optimize the weighting of the Scikit-Surprise prediction and Textblob polarity score to improve recommendation accuracy. A possible approach is to devise a backtesting methodology to evaluate predicted ratings using various weight combinations against actual Yelp data. By finding the optimal weights, our recommendation system can potentially provide more accurate and personalized recommendations to users.

4.3 Usability

To evaluate the user experience of our web application, we conducted a usability test with neutral participants who were not involved in the development process. The participants were asked a series of questions to assess their perception of the website. These questions included:

- Question 1: What do you think this website does?
- Question 2: How long did it take you to reach that conclusion?
- Question 3: On a scale of 1 to 5, with 1 being difficult and 5 being easy, what score would you give this app for “ease of use”?
- Question 4: On a scale of 1 to 5, with 1 being not at all and 5 being often, what score would you give this app for “how often would you use this website”?
- Question 5: What do you think is the best feature?
- Question 6: What do you think is the worst feature?

The usability of the web application was assessed through feedback from four users.

User 1 found the primary function to be providing restaurant recommendations, rated the app’s ease of use as average with a score of 3, and liked the informative view details page. However, the search bar was found to be clunky.

User 2 identified the website as a tool for finding restaurants, appreciated the detail in finding restaurants but rated the app’s ease of use as low with a score of 2.

User 3 took 2 minutes to conclude that the website was a restaurant finder, rated the app’s ease of use as moderate with a score of 2.5, and appreciated the feature of not having to search for reviews. However, they found the UI lacking in contextual information.

User 4 identified the website as a tool for recommending restaurants and appreciated the app’s ability to narrow down options. They rated the app’s ease of use as moderate with a score of 3 but could not use it due to the limited coverage of cities.

Overall, the usability test provided valuable insights on the user experience, with users appreciating the informative view details page and the ability to narrow down options. However, the search bar was found to be clunky, and some users noted a lack of contextual information. These insights can be used to improve the web application for future users.

5 PLAN OF ACTIVITIES

All team members have contributed a similar amount of effort.

6 CONCLUSIONS

In this study, we developed a restaurant recommendation website by leveraging a publicly available Yelp dataset. Our approach builds upon existing research by combining collaborative filtering and sentiment analysis techniques to generate novel recommendations. Our application demonstrates the feasibility of scalable recommendations given sufficient compute power, particularly system memory (RAM). Through exploratory analysis and evaluation using RMSE metrics, our novel recommendation engine was successful in generating recommendations that users are likely to enjoy.

The developed backend is built using Python-Flask-SQLite technology and allows text searches of the entire Yelp dataset, encompassing all reviews and businesses, along with their attributes. The frontend interface consists of dropdowns for selecting cuisine, city, and user, which then searches the data to generate restaurant recommendations. The recommendations are displayed on a map, and a summary section provides information about the restaurants, including the recommendation score.

The findings of our UI survey suggest that our web application offers significant value to users, as evidenced by its above-average value proposition scores. While the usability scores and feedback were average, we view this as an opportunity for further improvement in the user experience provided by the application. Therefore, we plan to conduct user clinics or focus groups to enhance the visuals and usability of the application. These efforts will build upon our current success and ensure that our concept continues to meet and exceed user expectations.

Future research topics of interest include exploring scaling collaborative filtering models to the full Yelp dataset in a memory-constrained training and production environment. Additionally, research focused on backtesting for better model tuning could lead to even better results with our collaborative filter plus sentiment analysis approach. Our study provides a significant contribution to the restaurant recommendation domain, highlighting the effectiveness of combining collaborative filtering and sentiment analysis in generating novel recommendations, as well as providing insight into the scalability of the approach.

7 REFERENCES

(1) Asani, Elham, Hamed Vahdat-Nejad, and Javad Sadri. "Restaurant recommender system based on sentiment analysis." Machine Learning with Applications 6 (2021): 100114.

(2) Guo, Guibing, Jie Zhang, and Neil Yorke-Smith. "A novel evidence-based Bayesian similarity measure for recommender systems." ACM Transactions on the Web (TWEB) 10.2 (2016): 1-30.

(3) He, Xiangnan, et al. "A Neural Collaborative Filtering Framework for Recommendations." Proceedings of the 26th International Conference on World Wide Web, ACM, 2017, pp. 173-182.

(4) Hu, Minqing, and Bing Liu. "Mining and summarizing customer reviews." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. 2004.

(5) Jindal, Nitin, and Bing Liu. "Opinion spam and analysis." Proceedings of the 2008 international conference on web search and data mining. 2008.

(6) Keller, Benjamin J., Bharath Kumar Mohan, and Naren Ramakrishnan. "Scouts, promoters, and connectors: The roles of ratings in nearest-neighbor collaborative filtering." ACM Transactions on the Web (TWEB) 1.2 (2007): 8-es.

(7) Kim, Kyoungok, and Cheong Rok Lee. "An improved similarity measure for collaborative filtering-based recommendation system." International Journal of Knowledge-based and Intelligent Engineering Systems 26. IOS Press. 2022

(8) Lee, Cheong Rok, and Kim, Kyoungok. "An improved similarity measure for collaborative filtering-based recommendation system." International Journal of Knowledge-based and Intelligent Engineering Systems 26. IOS Press. 2022

(9) Lops, Pasquale, Marco de Gemmis, and Giovanni Semeraro. "Chapter 3: Content-based Recommender Systems: State of the Art and Trends." The Adaptive Web, edited by Peter Brusilovsky et al., Springer, 2007, pp. 73-120.

(10) Ma, Yue, Guoqing Chen, and Qiang Wei. "Finding users preferences from large-scale online reviews for personalized recommendation." Electronic Commerce Research 17 (2017): 3-29.

(11) Mohan, Bharath Kumar, Benjamin J. Keller, and Naren Ramakrishnan. "Scouts, promoters, and connectors: The roles of ratings in nearest-neighbor collaborative filtering." ACM Transactions on the Web (TWEB) 1.2 (2007): 8-es.

(12) Rendle, Steffen, et al. "Neural Collaborative Filtering vs. Matrix Factorization Revisited." Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 239-247

(13) Sawant, Sumedh. "Collaborative filtering using weighted bipartite graph projection: a recommendation system for yelp." Proceedings of the CS224W: Social and information network analysis conference. Vol. 33. 2013.

(14) Seo, Young-Duk, Young-Gab Kim, Euijong Lee, and Doo-Kwon Baik. "Personalized recommender system based on friendship strength in social network services." Expert Systems with Applications 69. Elsevier. 2016

(15) Suganeshwari, G., and S. P. Syed Ibrahim. "A survey on collaborative filtering based recommendation system." Proceedings of the 3rd international symposium on big data and cloud computing challenges (ISBCC–16'). Springer International Publishing, 2016.

(16) Sun, Jiancong. NLP Analysis and Recommendation System for Yelp. University of California, Los Angeles, 2020.

(17) Sun, Shiliang, Chen Luo, and Junyu Chen. "A review of natural language processing techniques for opinion mining systems." Information fusion 36 (2017): 10-25.

(18) Yang, Wenqing, Yuan Yuan, and Nan Zhang. "Predicting Yelp ratings using user friendship network information." Available at SSRN (2015).