Yelp Who? A New Approach to Restaurant Recommendation

Introduction

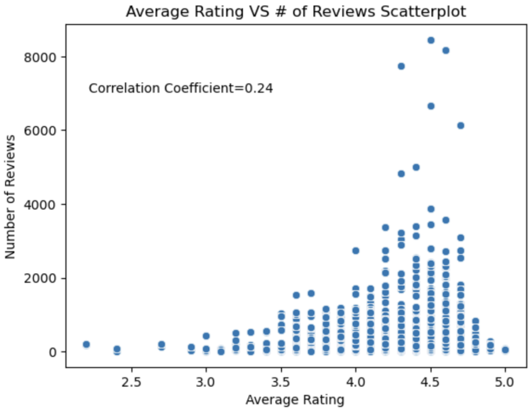
Restaurant review databases are an integral part of the food and beverage industry and determine the success of a given establishment compared to other eateries. This concept is so ingrained in our culture today, given the abundance of restaurants in a given locality, that entire businesses have been built with the purpose of getting the user to provide feedback on restaurants they have visited in the past in order to gain insight into user preference and provide more accurate recommendations for what restaurants they might like in the future.

With this in mind, our goal was to use a dataset of local Google reviews to create a predictive model that used similarity to suggest restaurants that might be appealing to a particular customer.

Part 1: Dataset/EDA

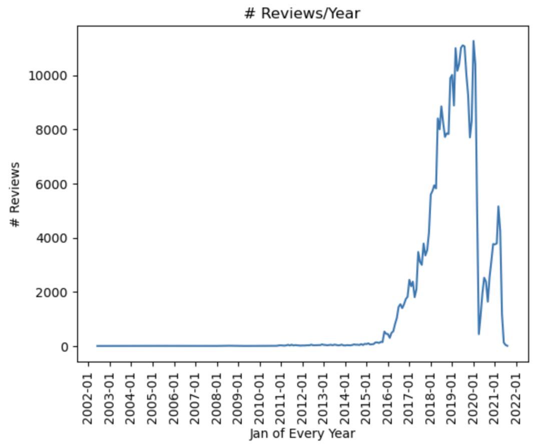
The dataset we chose for our model was a merged dataset containing local Google reviews from Google Maps as well as metadata containing the details of each establishment, such as name, description, hours of operation, and other unique features. While we did attempt to use the entirety of the review and metadata datasets, we found that the size of the file was too large to use in its entirety, and the wisest course of action for efficient analysis and modeling would be to use a subset of the data. Therefore, we used the full Hawaii metadata dataset, but only the 10-core rating dataset. In order to filter this dataset to serve our purpose, we used the “category” column and focused solely on the “Restaurant” shops. Next, we performed some basic data preprocessing, dropping columns that we deemed irrelevant to our analysis and duplicates, and transforming columns such as the “Time” column into its proper datetime format. Then, we went on to explore the individual components of the resulting data frame.

We found that the average rating and number of reviews for a restaurant were positively correlated, and as the average rating increases, the number of reviews also increases proportionally. This is logically intuitive since more people would be likely to visit a restaurant with higher ratings. However, we did notice that the scatterplot has a correlation coefficient of 0.24, indicating a relatively weak relationship between the two variables.

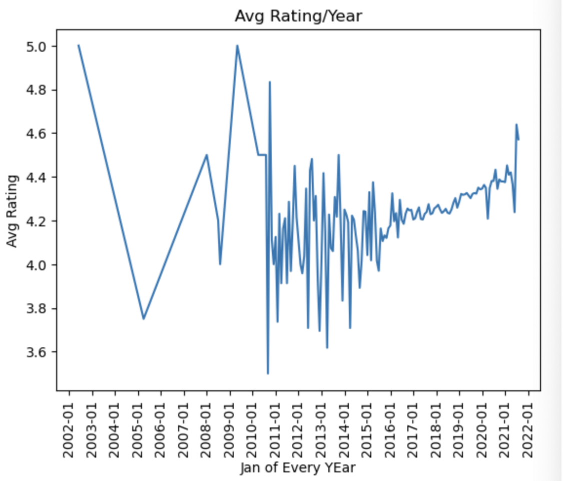


We then further explored how the number of reviews and average ratings fluctuate over time. In doing this, we were able to see a dip in the number of reviews due to COVID-19 in 2020, and were able to accurately visualize how the evolution of rating platforms enabled more users to rate restaurants as the years passed. We were also able to observe from the “Avg Rating/Year” graph that the evolution of these aforementioned platforms resulted in a more accurate portrayal of average rating as a metric for restaurant quality.

# Reviews / Year From 2002 to 2022



Average Rating / Year From 2002 to 2022



Next, we delved into the average number of reviews per restaurant. In particular, our aim in this step of the EDA was to explore the distribution of ratings and see whether it was skewed in some cases, resulting in potential bias during our model development. We found that while the average number of reviews per restaurant was around 38, the number of restaurants with the most reviews had 858 reviews. We, therefore, had to perform further investigation to determine whether this value was an outlier, or the dataset was large and therefore had a larger variance. Upon further scrutiny, we concluded that, due to the number of reviews for the top 20 restaurants, the dataset simply had a larger variance, and 858 was not an outlier.

After forming these initial conclusions about the nature of the data and performing extensive data cleaning, it was time to proceed to our predictive task.

Part 2: Predictive Task

After conducting the EDA on our model described above, we weighed the characteristics of the data with possible prediction tasks to decide what recommendation system the data is best suited for. The predictive task we selected was to predict whether a given user would visit a given restaurant. The model would make this prediction based on the restaurants the user has previously visited and based on the users that have previously visited the restaurant. In terms of preprocessing, the dataset initially included a variety of businesses such as stores, banks, and schools. We filtered the dataset to exclude non-restaurant businesses, we also dropped any columns that were redundant or included unnecessary information for the prediction task. We then created two data structures that were the backbone of our model: one mapping users to each restaurant they have visited (usersPerItem) and another mapping restaurants to each user (itemsPerUser). These were used in the biggest section of our model, the Jaccard similarity portion. The Jacquard similarity was based on the user history of each restaurant, meaning we needed to store the format in an easily accessible structure that is indexed by the restaurant. We also created a data structure to track restaurant popularity, which was measured through visitor counts.

To evaluate the model, we primarily used two metrics: accuracy and ROC AUC score. We chose to use two metrics because we believe they reveal different things about model performance, strengths, and weaknesses. Our evaluation was based on an 80-20 split of our dataset into training and testing data. This was done so that our model could be tested on unseen data and would not produce an inaccurately high evaluation metric because of overlapping training and testing data. Additionally, we compared our model to two other baseline models: one random prediction model and one cosine similarity model. We also used k-fold cross-validation to check the performance of the model against different train-test splits of data.

Part 3: Model

The final model we chose to use was a combination of multiple prediction techniques we discussed in class, specifically a popularity model and a similarity model. The bulk of our model's decision-making happens within the Jaccard similarity model. As a reminder, the objective of the model is to predict whether a given user visits a given restaurant. The first Jaccard similarity attempts to answer this question. `MaxSim` is a variable that represents the maximum similarity score for a user-restaurant set. This is found by comparing the given restaurant to every other restaurant in the user history and finding the similarity score of the most similar restaurant. If this similarity score is higher than a certain threshold, which we call the similarity threshold, we would say that the user is likely to visit that restaurant. The second part of our model is based on the restaurant's popularity. We set a popularity threshold based on the number of visitors in the overall dataset. If the given restaurant is more “popular”, and has a number of visitors over the popularity threshold, then we always predict that a user will visit. Including the popularity, threshold helped make the model more accurate and made sense in the real-life context, given that all kinds of users are likely to go to extremely popular restaurants regardless of their history.

We chose to use this combination of models because we thought that the behavior of similar users is the best indicator of a user's behavior. We assumed that users who frequent the same restaurants, have similar tastes, and have similar eating habits are likely to go to the same places. Jaccard similarity is a model that uses these user interactions as the main information for its prediction. We ran the model and found an accuracy of 61% on the testing data. While this is better than random prediction, we wanted to improve the model. We ran an accuracy-based grid search to find the best similarity and popularity thresholds. We also ran parameter tuning based on the ROC AUC score. Both parameter searches had similar results. Both outputted 0.1 as the best similarity threshold and outputted 400 and 300 as the best popularity thresholds. Finally, we decided to use 0.1 and 300 as the thresholds since they were the results of the ROC AUC taunting, and we thought that ROC AUC was a more robust evaluation metric.

Scalability was an issue for us because finding Jaccard Similarity in many pairs of restaurants is both time and memory-consuming, not to mention redundant. To save the cost of repeatedly calculating these scores, we created a data structure that found all possible pairs and their jacquard similarities once. While this is still time and memory-heavy, it only needs to run once instead of many times, and this structure can be referenced later.

We used two other models as baselines or comparisons, as mentioned above. We used a random model and a cosine model. The random model, arbitrarily decides whether someone is likely to visit a restaurant. This model is extremely simple and quick, but it makes no logical sense, and it doesn't take into account any information about the user or the restaurant. However, because it randomly selects a binary, it has roughly 50% accuracy. We also used a cosine similarity model because this takes into account the ratings of each review, which could also be a good predictor. Our first idea was to combine the Jacquard and the cosine similarity. However, despite parameter tuning, the best performance of cosine similarity was 60% so we thought it would drag down the performance of the rest of the model, which peaked at 79% accuracy.

A strength of the Jaccard similarity model is that it can personalize recommendations based on user history. However, since the whole prediction is based on previous data, this is a weakness for new users who haven't visited any/many places or for restaurants with no/few customers. Since there is no known information, the model doesn't have many data points to compare to and therefore can't predict well. To help deal with this weakness we introduced the popularity-based model. While it is less personalized, it is good in scenarios where user data is unavailable.

Part 4: Literature

The Google Local Reviews (2021) dataset is a dataset containing review information up to September 2021 from Google Maps business metadata, and was created in 2022 by Jiacheng Li, Jingbo Shang, Julian McAuley, An Yan, Zhankui He, and Tianyang Zhang. This dataset also has two variants, including a version with item images, and a version specific to user review information.

This dataset was webscraped originally for the composition of two research papers. The first, titled “Personalized Showcases: Generating Multi-Modal Explanations for Recommendations“ created a multi-modal framework to perform a “personalized showcase” task, which, given a recommended item/business, gave a mixture of textual and visual information to advance user experience. This new model, while unique, cannot be compared to our own model.

The second paper published based on this dataset was titled “UCTopic: Unsupervised Contrastive Learning for Phrase Representations and Topic Mining”, and aimed to improve the quality of phrase representation by finding topics and related terms in documents within the domain of topical phrase mining. Again, while this paper did highlight interesting conclusions formed by the researchers, the model is not comparable to our own, and therefore accuracy cannot be weighed against this paper.

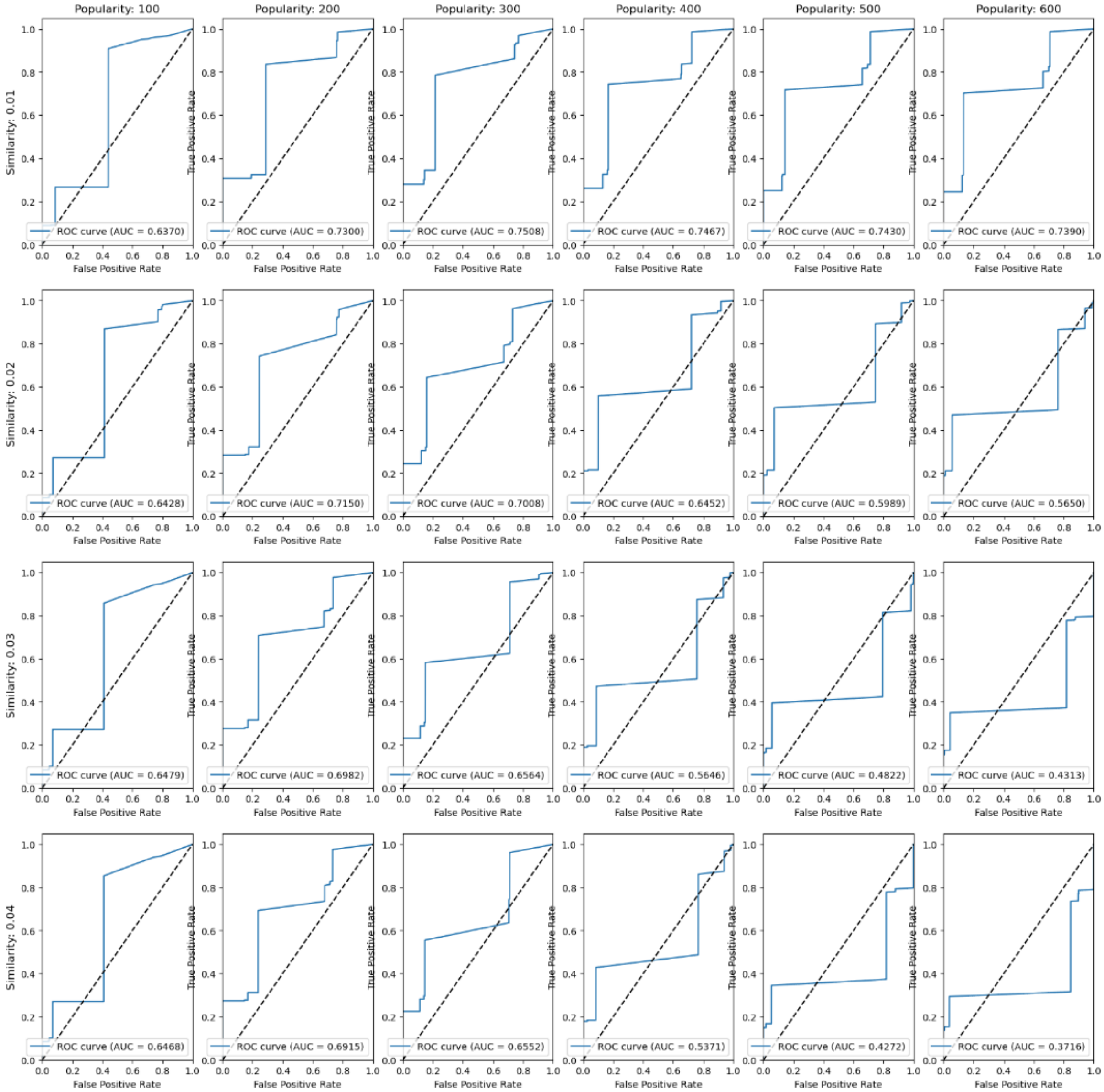
We also explored related works that focused on recommendation systems for restaurants, and how those model evaluations compared to our own. In order to accurately gauge the performance of our model compared to others, when the widely used performance metric for many of these models, we decided to additionally compute the F-1 score of our model, which was 0.78.

The first paper we compared our results to was titled “Extraction of Atypical Aspects from Customer Reviews: Datasets and Experiments with Language Models”, and personalized restaurant recommendations based on a “surprising aspect” of the restaurant that was not disclosed in advance. This model had drastically varying results for the fine-tuned model, which ranged from an F1 of 62.3% and 84.5%. This paper took advantage of LM models, specifically FLAN-T5 and ChatGPT, which was an extremely different approach from our own.

The second paper we compared our results to was titled “MenuAI: Restaurant Food Recommendation System via a Transformer-based Deep Learning Model”, and focused on creating a recommendation system that recommended food based on user food preferences. This model utilized MenuRank, a dataset constructed by the authors, and containing 5626 menus, as well as a tedious algorithm in which the menu is digitized, text extraction is performed, and the text is inputted into an LTR model. This model had an accuracy ranging from 65.2% and 96.1%.

The final research paper we referenced is titled “MealRec: A Meal Recommendation Dataset”, which introduces a dataset constructed by the authors containing over 1500 users and 3800 meals. The model proposed first uses OCR to perform text extraction and obtain dish names from menu images. Then, the vectors of words are inputted into a transformer model that ranks dishes based on user preference. This model had an HR@5 of 0.4838. Although this performance metric cannot be compared to our own because we did not have a ranked list of restaurants in our code, it was interesting to see how this paper evaluated their newly constructed dataset.

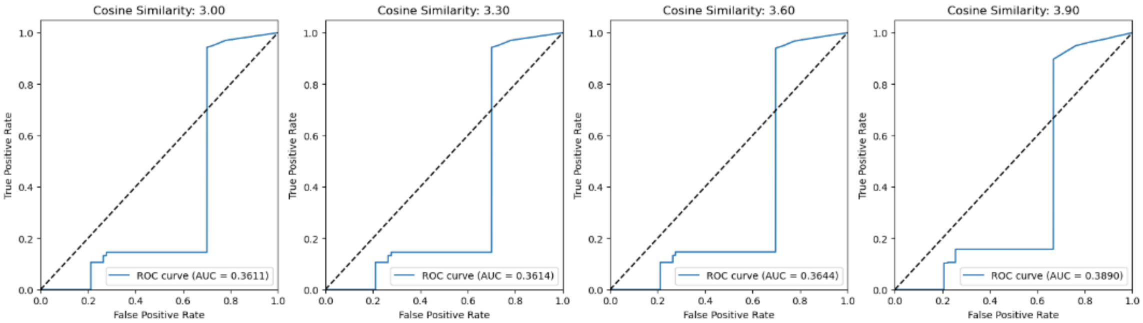
In general, based on the research papers we read for contextual purposes, our model’s performance metrics fell within the range of evaluation benchmark standards described within each respective paper.



Part 5: Results

After completing the training of our model, we evaluated it on the test set and observed our results. The figure above shows the results of a grid search in our attempt to find the best combination of hyperparameters. A Jaccard similarity threshold and a popularity threshold were the two parameters that were used in our model that would help us recommend a restaurant to a given user. Jaccard similarity attempts to measure the similarity between two sample sets. In the case of our model, we quantified the similarity of a given restaurant to all of the restaurants the user had reviewed. The Jaccard similarity threshold determined the value of Jaccard similarity at which we would recommend the user to a given restaurant. We also added a measure of popularity that could recommend a restaurant regardless of the user's history of reviews. After our findings of correlation between rating of reviews and number of ratings, we believed that if a restaurant had a certain number of reviews, we deem it to be "popular" and thus, should be recommended regardless of user preferences. The popularity threshold then essentially determines the number of reviews the restaurant should have to be recommended to the user. If either of these thresholds were exceeded, we predict the user to visit the restaurant.

Based on the grid search, we observed the best combination of hyperparameters to be a Jaccard similarity threshold of 0.01 and a popularity threshold of 300. This combination of parameters obtained an AUC of 0.7508 which is fairly good for a recommendation model. Looking at the figure above, the combinations of parameters with 0.01 as the similarity threshold were all fairly close in performance. When doing a grid search based on accuracy instead of AUC, the best combination of parameters was 0.01 and 400 for the similarity and popularity threshold respectively which obtained an accuracy of 0.79. This gives us stronger evidence that 0.01 is the best threshold for similarity while there may be less certainty for the popularity threshold as it can range between 300 and 400. As we believe that AUC is a better and more complex metric of evaluation, we used 300 as our popularity threshold in our final model.



Out of curiosity, we also trained a few models using a different similarity measure to see if we could obtain better results. Instead of Jaccard similarity, we used Cosine similarity and took out our popularity measure. We evaluated multiple models that used different Cosine similarity thresholds and the results can be seen in the figure above. As you can, they all performed really poorly with AUC's of around 0.36. We believe this significant range in performance to be a result of the nature of our dataset. Jaccard similarity outperforms Cosine similarity in cases where duplication does not matter and vice versa. During our stages of preprocessing, we removed duplicate reviews and our sample of negative reviews do not contain any duplicate user and restaurant pairs. The nature of our dataset suggests that duplication is non-existent and thus Jaccard similarity is better as a similarity metric which is why our final model performed considerably better.

In conclusion, we found that a model with a Jaccard similarity threshold of 0.01 and a popularity threshold of 300 was the best at recommending restaurants to users, obtaining an AUC of 0.75 and an accuracy of 0.7. With the personal taste of food being so subjective from person to person, there are many factors to consider in food recommendation such as what you had the day before, price, proximity. It is extremely difficult to take into consideration all these factors in a recommendation model but our model aimed to tackle this problem with an objective, user-history based approach and we believe that we obtained respectable results with our recommender system.

References

[1] Yan, An, et al. "Personalized Showcases: Generating Multi-Modal Explanations for Recommendations." *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 6 Apr. 2023, https://doi.org/10.1145/3539618.3592036.

[2] Li, Jiacheng, et al. "UCTopic: Unsupervised Contrastive Learning for Phrase Representations and Topic Mining." *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 27 Feb. 2022, https://doi.org/10.18653/v1/2022.acl-long.426.

[3] Nannaware, Smita, et al. "Extraction of Atypical Aspects from Customer Reviews: Datasets and Experiments with Language Models." 5 Nov. 2023, https://arxiv.org/pdf/2311.02702.pdf.

[4] Ju, Xinwei, er al. "MenuAI: Restaurant Food Recommendation System via a Transformer-based Deep Learning Model." 15 Oct. 2022, https://arxiv.org/abs/2210.08266.

[5] Li, Ming, et al. "MealRec: A Meal Recommendation Dataset." 24 May 2022, https://arxiv.org/abs/2205.121