GENIE: SIMPLIFYING THE RESTAURANT FINDING PROCESS

INTRODUCTION - MOTIVATION

With the vast amount of restaurant review data available, it is often tedious and time-consuming to find the perfect restaurant. We present Genie, an innovative restaurant recommendation system with visualization capabilities to simplify finding and comparing the best restaurants along a route.

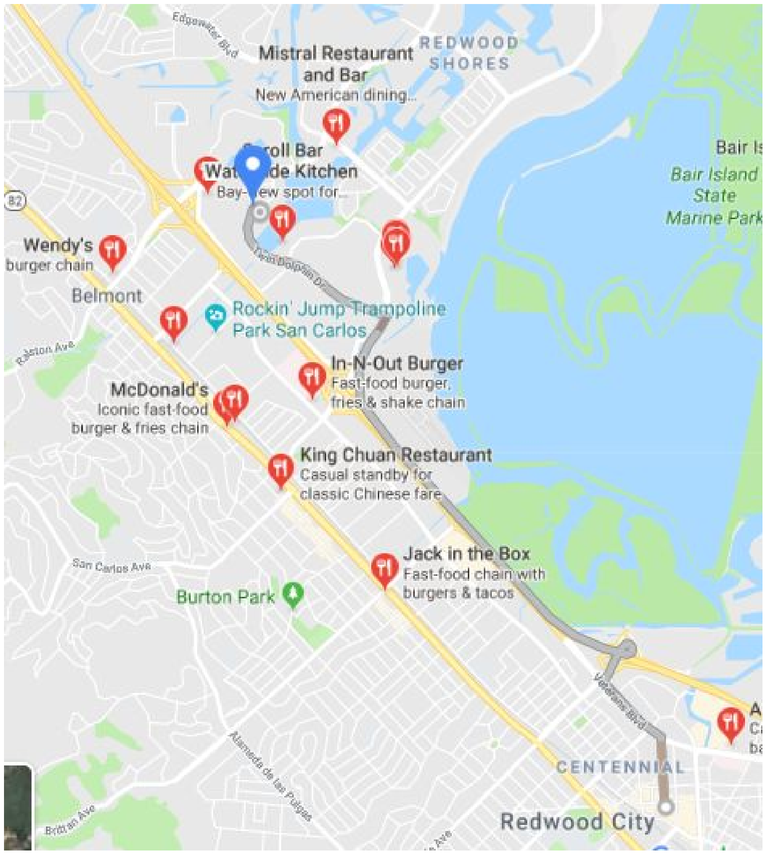

Mistral Restaurant and Bar
New American dining...
REDWOOD SHORES
Wendy's
burger chain
Belmont
Rockin' Jump Trampoline Park San Carlos
In-N-Out Burger
Fast-food burger, fries & shake chain
McDonald's
Iconic fast-food burger & fries chain
King Chuan Restaurant
Casual standby for classic Chinese fare
Jack in the Box
Fast-food chain with burgers & tacos
Burton Park
Bair Island State Marine Park
CENTENNIAL
Redwood City

Figure 1. Google Maps displaying restaurants around a route. Genie will summarize and compare the best restaurants with visualizations

PROBLEM DEFINITION

There are more than one million restaurants in the United States (https://upserve.com/restaurant-insider/industry-statistics/), and 90% of restaurant guests research a restaurant online before dining - more than any other business type. Websites and apps like Yelp, Google, and TripAdvisor, are commonly used to find and compare restaurants.

With the vast amount of restaurant review data available, it is often difficult and time-consuming to weed through all of the content in order to find the perfect restaurant. The restaurant location is also a critical factor, especially when having to meet up with friends, or when planning trips. Even after a few restaurant candidates are selected for comparison, they usually have to be manually inspected and compared. This project combines a restaurant recommendation system using ML techniques with summary comparison visualizations in order to allow the user to quickly select the best restaurant for their preferences.

SURVEY

- [1]: The paper focuses on the relevance component of the user review, to determine how certain user's reviews are more influential than others, based on the social attributes of the users. This is useful to gain insights into how we can improve the quality of the recommendation system, by filtering out more relevant users' reviews. This paper also gives insights into the Yelp dataset and dataset processing. Only numerical ratings analysis has been considered as the context for determining the user influence. It may be better if techniques like sentiment analysis and ML are applied.
- [2], [6]: These papers explore the challenges of visualizations, such as transparency, controllability, explorability, and context-awareness. Having awareness of strategies for useful visualizations is critical for our project. A taxonomy strategy is introduced which could help to improve user engagement and chances to accept the recommended items. The proposed taxonomy needs to be further developed to be applied to more solutions.
- [3], [15], [16], [17]: Instead of having to read through numerous product reviews, it may be more useful to analyze and present visualizations to better understand the differences between a set of products. These papers can be referenced as to what types of visualizations will work best for the visual comparison part of our project. Some papers also use text analysis techniques, which we had planned to use for our project. However, none of the papers' algorithms used sentiment analysis, and some of the visualizations used may not be the best choices.
- [4]: An incremental learning algorithm for sentiment analysis is to use tree-based classifiers to create trees, then average the predictions from all trees. Choosing an appropriate split in a tree is critical for achieving high accuracy and improving the model performance. This paper compares the accuracy of different approaches but doesn't compare the running time.
- [5], [14]: Route recommendation systems based on points of interest are the main ideas for these papers. They will be useful when considering optimizing routes based on chosen restaurants. These are good references for creating custom route for optimization. There are slight differences exist especially available resources to apply algorithm. For example, paper uses RFID/contextual information to generate desired results. We need to define our own parameters.

- [7], [12], [13]: These papers deal with algorithms for recommendation systems: collaborative filtering(CF) and content based filtering(CBF) hybrid system involving sentiment analysis with CF and CBF. They give us good insight into what ML techniques are being used in food recommendation systems, and which can be applied to the recommendation engine for our project. The challenge is to accurately understand the food domain and come up with the best model for optimal results.
- [8]: This paper focuses on developing Maps based mashups and challenges involved in integrating data from different services. It touches upon a wide variety of technologies and applications developed in this space. We have a feature to present the restaurants on a map along a route. This paper can be foundational in developing a Google Maps based visualization.
- [9]: This book provides several different technologies to establish content-based recommendation systems. One of the main topics of this chapter is creating decision tree to data partitioning. We can use this to pick best restaurants and/or categorize user's preference to match specific groups of restaurants.
- [10]: This paper lays out one way that ML algorithms have been applied towards the Yelp dataset which we can build on it by adding interactive visualizations. This paper only focuses on the analysis and does not have many visualizations.
- [11]: A survey of the six components of big data analytics: data generation, acquisition, storage, advanced analytics, visualization, and value-creation, which will be useful to apply to our project. Although this report has detail on all steps of a big data visualization project, it does not mention interactive visualizations.

PROPOSED METHOD

1. Intuition

Our project expands on current state of the art with the following innovations:

1. Recommends and compares restaurants along a route rather than around a single location.
2. Displays useful summary comparison visualizations so the user does not have to spend time reading through the reviews.
3. Implements a recommendation system with customized algorithms to provide the user with more information than just star ratings to allow the user better judgement.

2. Description of Approaches

A high-level flow chart is shown below in Figure 2.

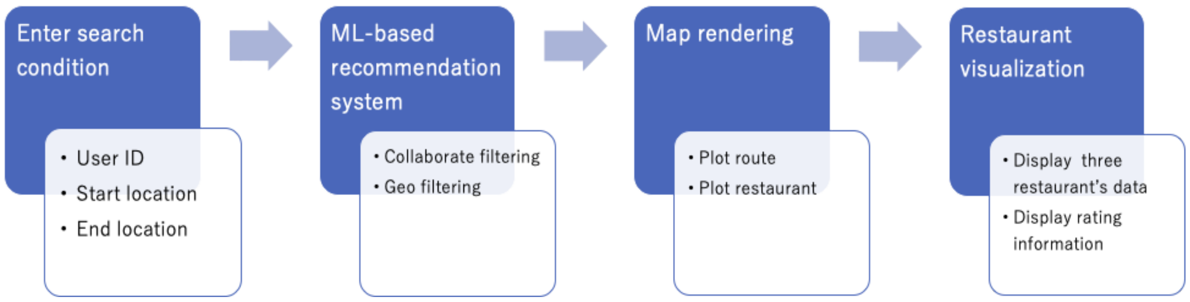
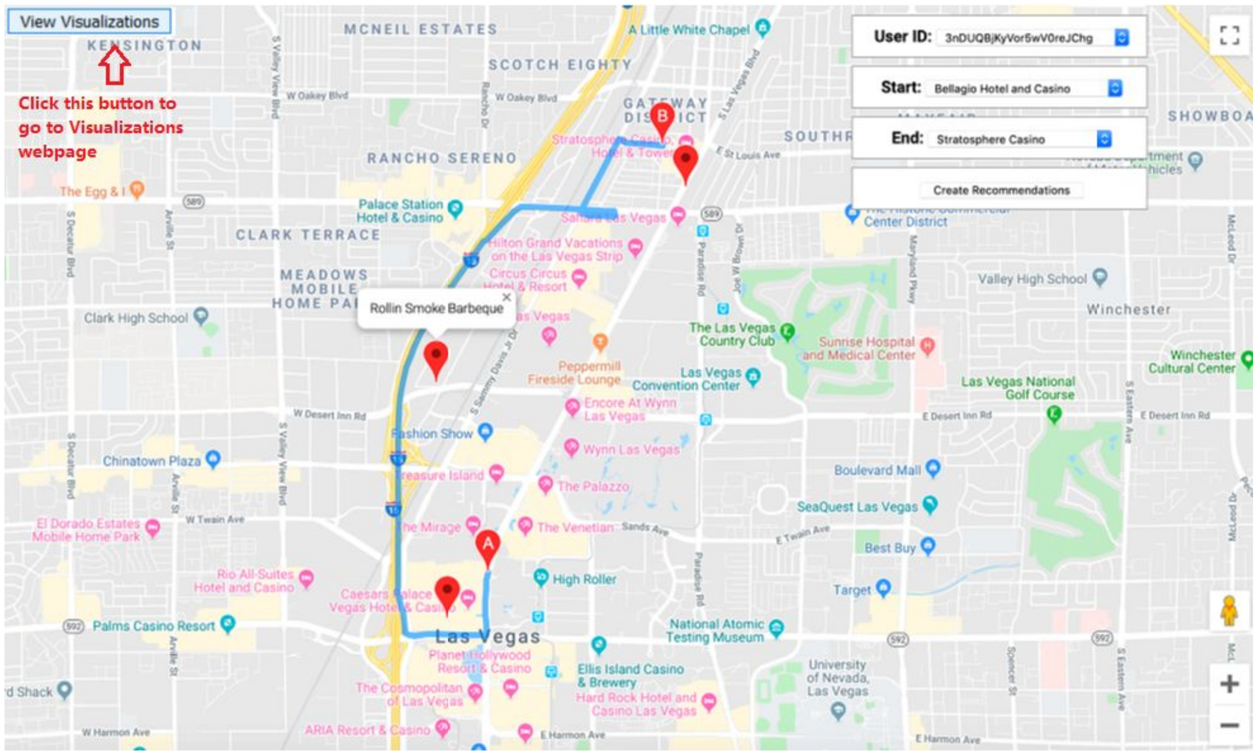


Figure 2. High-Level Flow Chart

Displaying Route with Map


User ID: 3nDUQBjKyVor5wV0reJChg

Start: Bellagio Hotel and Casino

End: Stratosphere Casino

Figure 3. Map displayed with three recommended restaurants along route

The map and directions used in this project were implemented using Google's Maps and Directions API. The user will input a start and end location, and the map will display a route. The map will also pin-point the restaurants selected from our recommendation system using the latitude and longitude coordinates provided from the Yelp Fusion API. Upon mouseover, the name of the restaurant will appear above the marker for each recommended restaurant. Due to the limitations of the Yelp dataset and for demo purposes, predetermined User ID and locations will be selected via a drop-down list.

Restaurant Recommendation System

The restaurant recommendation system uses a singular value decomposition (SVD) algorithm combined with the geofilter in order to output the top three recommended restaurants for a particular user along the user-determined route. The recommendation system is the driving technology of this project, and a lot of experimentation and evaluation was performed to develop the system. The details of the experiments can be found below in the Experiments/Evaluation section.

Interactive Visualizations

The bar chart compares the rating of the three restaurants. Upon mouseover a bar, it will show the corresponding rating and total number of reviews for that restaurant.

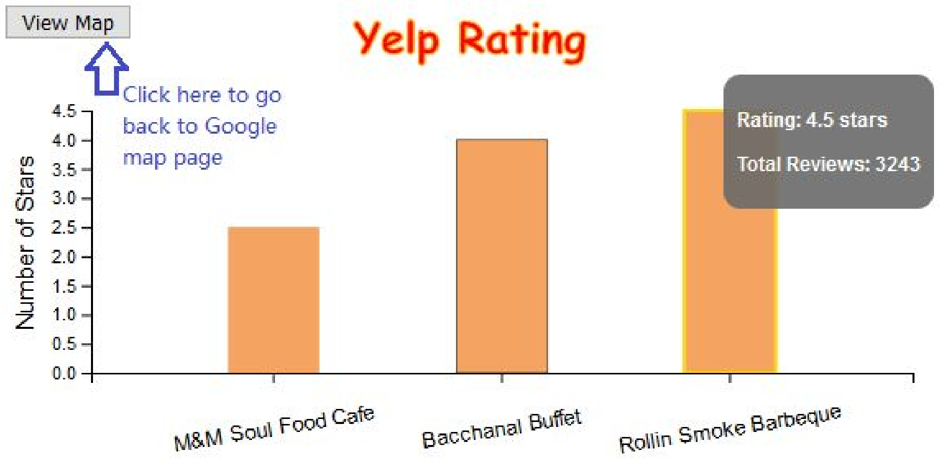


Figure 4. Bar chart comparison visualization with mouseover on Rollin Smoke Barbeque

For each restaurant, the restaurant name, food categories, price, and business hours are displayed. If the user wants more detail about a specific restaurant, clicking the restaurant name will open that restaurant's Yelp page in a new window.

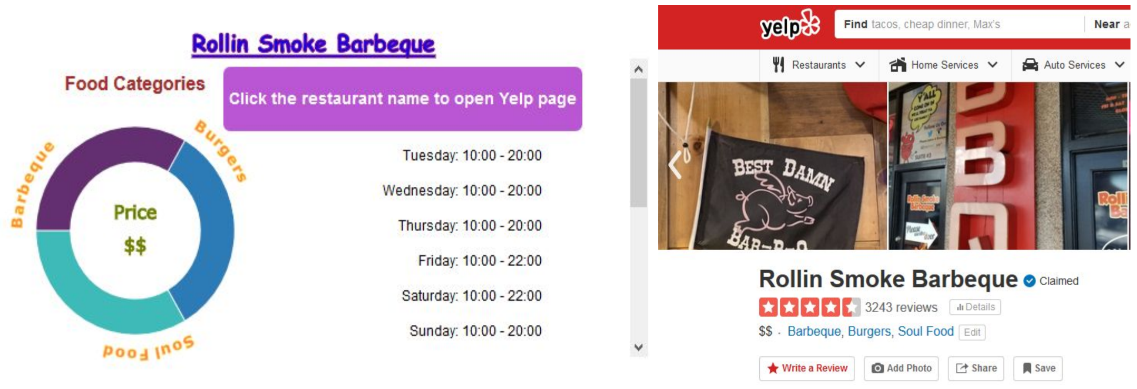


Figure 5. Summary and Yelp page for Rollin Smoke Barbeque

The full comparison visualization is shown in Figure 6 below:

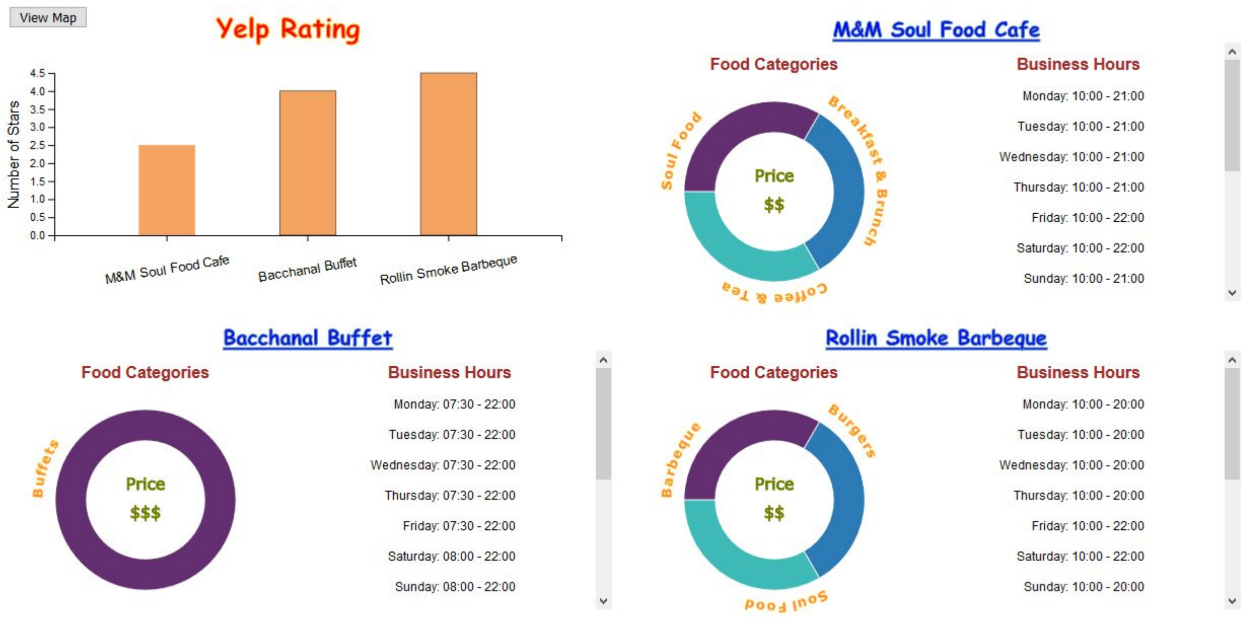


Figure 6. Example comparison visualization of three recommended restaurants

EXPERIMENTS / EVALUATION

1. Description of testbed

The recommendation system was evaluated by splitting the dataset into training and test sets, and RMSE was used to test the efficiency of each algorithm tested.

For evaluating the features of map rendering and visualizations, we relied on peer reviewing using the following questions:

- Is the product useful? (range of 5)
- Does the product meet the needs of the user? (range of 5)
- Is the product easy and intuitive to use? (range of 5)
- Does the user like the way the product looks and feels? (range of 5)
- Does the user prefer this product over similar products? (range of 5)
- What could be improved? (Comments)

For the visualization part of the project, a user preference test was executed with a user sample size of 10. For each user, a rating of ≥ 3 was obtained for each question. The most common feedback for the comments was related to the limitation of locations available. This was expected due to the inherent limitation of the Yelp dataset and because the User ID's for testing were taken from the test set, as opposed to a unique User ID with already personalized preferences.

2. Details of experiments

When we first began the project, we wanted to focus on recommendations for all restaurants in the U.S. Also, we wanted to make a general recommender system which could recommend a restaurant to any user. However, we realized that the publicly available Yelp dataset (8.7 GB) did not have all of the data for the entire U.S. compared to what can be found on Yelp's website. Based on this, we performed exploratory analysis on the Yelp dataset. Figure 7 shows a treemap of the count of businesses by state from the Yelp dataset by using Python's matplotlib and seaborn libraries. This shows states like CA and NY had very few businesses listed in the dataset, while places like AZ(56686) and NV(36312) had many businesses listed.

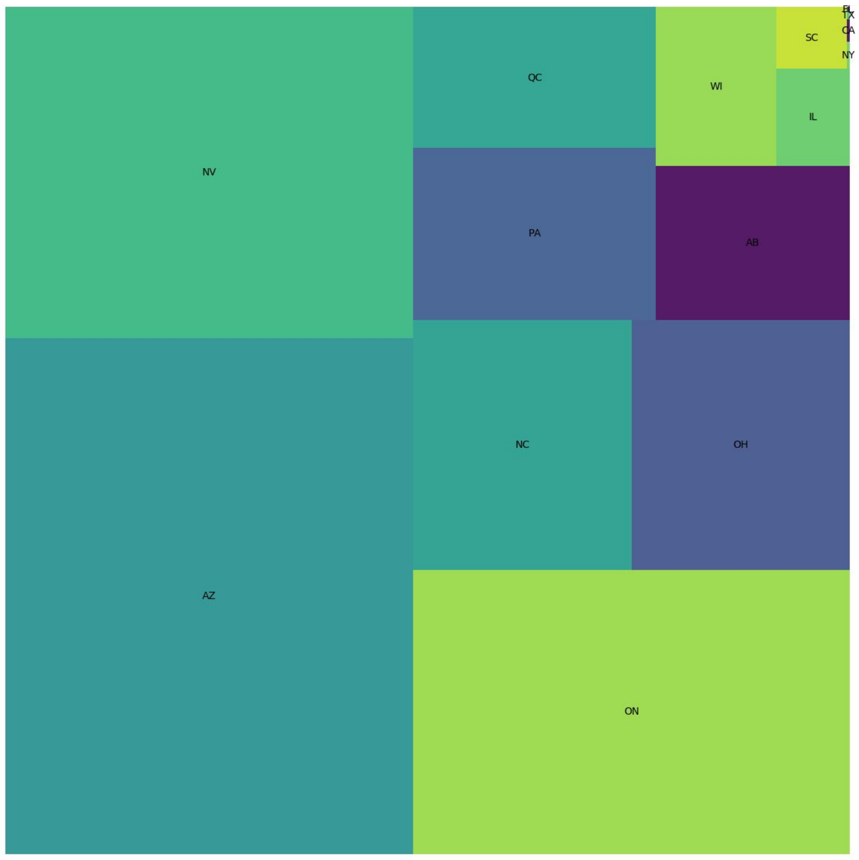
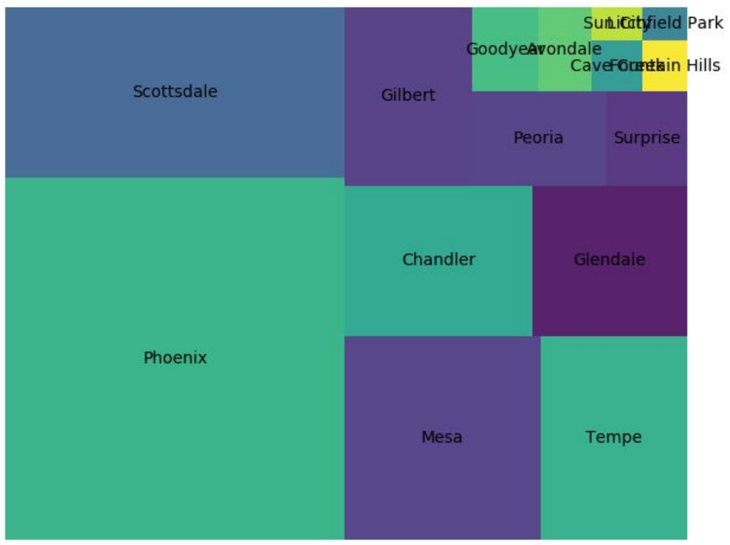


Figure 7. Treemap of number of businesses by state

From this result, we drilled down one more level to see the distribution by cities. We created treemaps for both AZ and NV to pick the city with the most businesses:



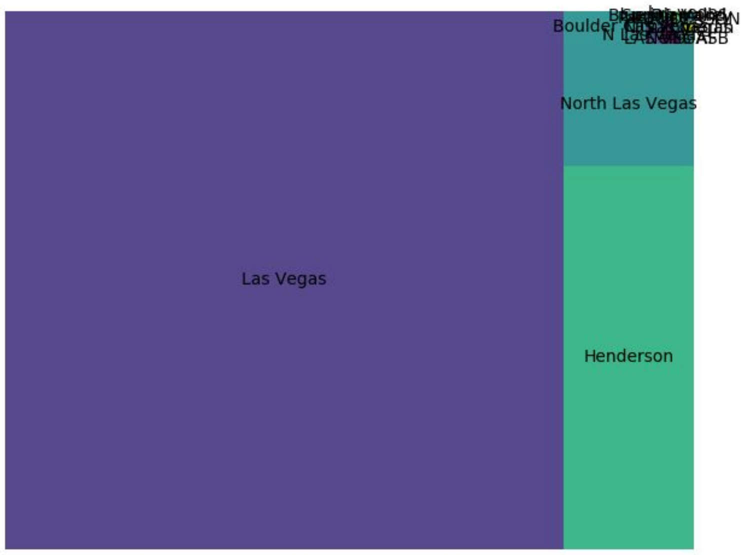


Figure 8. Number of businesses by city in Arizona

Figure 9. Number of businesses by city in Nevada

From the figures above, we see that Las Vegas is the city with the most businesses listed (29361). The next step is to confirm enough number of food businesses in Las Vegas. Figure 10 shows that restaurants top the number of businesses among all businesses in Las Vegas that are present in the Yelp dataset.

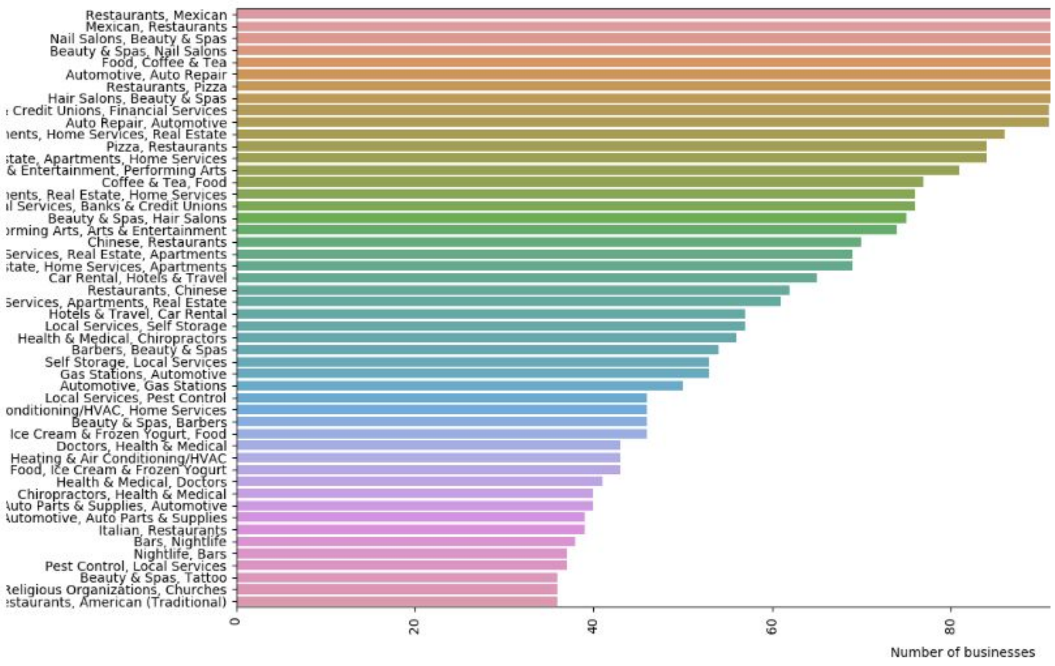


Figure 10. Sorted bar plot of business categories in Las Vegas

For exploratory purposes, we created a word cloud in Figure 11 with our filtered data. It shows a lot of Mexican restaurants in Las Vegas.

Mexican
Divino
Proteins
Taco
Rae
Stage
BBQ
Party
Restaurant
Bistro
Name
Jessie
Steak
Vince
object
Naco
Potato
Dairy
Eat
Bakery
dtype
Queen
Maria
Deli
Neil
Vegas
Drink
Tatuado
Length
Cinnaholic
Greens
LasGreat

Figure 11. Sorted bar plot of business categories in Las Vegas

Having performed the preliminary data analysis, we decided to focus our project on Las Vegas.

ML Pipeline

In this segment, we walk through the high-level steps performed end-to-end with respect to the recommender system and dwell into finer details in the later part.

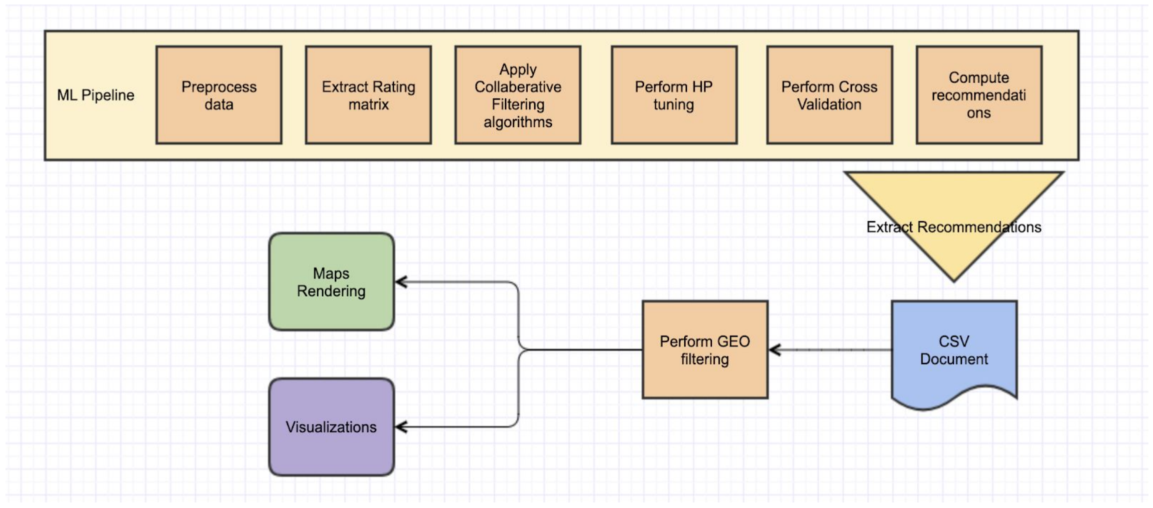


Figure 12. Flow chart of ML pipeline to map rendering and visualizations

Preprocessing Data:

First, we perform conversion and filtering of the data. We convert the business.json, user.json and review.json data from the Yelp dataset into csv for loading into the Pandas dataframe. These files are further filtered to obtain only food-related businesses in Las Vegas.

Extracting Rating Matrix:

Next, we merge pre-processed data and only extract the data required for collaborative filtering containing the [user_id, business_id, star_rating]. We also split the data into training and test sets. Training data is used to train the collaborative filtering algorithms, and test data to perform predictions for the recommended ratings.

Collaborative Filtering Algorithms:

We apply 13 different algorithms (combination of neighborhood based (KNN) and matrix factorization based) on the training data. Detailed algorithm information and how we pick the best algorithm are described in later sections.

Perform Hyperparameter tuning:

We perform hyperparameter tuning for all the algorithms using GridSearchCV. The optimal set of parameters are picked for each algorithm.

Perform Cross-Validation:

Once optimal sets of parameters are determined, Cross-Validation on the training set is performed for all the algorithms, and RMSE and MAE are collected. The algorithm with the least RMSE is picked as the best algorithm after this step. The reason behind choosing RMSE as a metric is that intuitively RMSE squares the errors before computing the mean, thus giving more weight to outliers or large errors. In this case, we desire to not have large errors and therefore use RMSE.

Compute Recommendations:

After the best algorithm is selected, we compute the recommendations for the test dataset using the chosen algorithm. The resultant predictions for the test users are stored in a csv file.

Geo Filtering:

In this step, based on the source and destination coordinates that are provided, we generate a bone-shaped contour, and filter out all the businesses that are not present in that bone-shaped contour. This is customizable by the diameter and width of the contour.

Algorithms and Implementation

Collaborative Filtering:

The basic principle governing collaborative filtering is that a user will be given the best recommendations from like-minded people with similar tastes. The path we chose to follow was to explore as many algorithms as possible and perform cross validation to pick the best algorithm based on our defined metrics. We used a Python scikit library called Surprise, which has builtin algorithms for developing recommender systems based on explicit ratings. We explore two main approaches - Neighborhood Based and Matrix Factorization Based methods.

Neighborhood Based:

These are based on nearest neighbor approach.
KNNBasic, KNNBaseLine, with similarity options of pearson, pearson_baseline and cosine, KNNWithMeans, and KNNZScore. In all of these KNN inspired algorithms, a similarity matrix is constructed based on user-user similarity or item-item similarity, using a slight variation of the formula below:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

or

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)}$$

Figure 13. Basic Formula for KNN-based algorithms

Since User X User similarity matrix is very large for the Yelp dataset, the memory footprint for computing similarity matrix based on user-user was too big. We could not run it under machines with 64GB RAM. Therefore, we had to use item-item similarity based KNN for all of the above algorithms. This required less memory and ran to completion in reasonable time.

Matrix Factorization Based Methods:
Singular Value Decomposition:
SVD is a technique in which we try to reduce the dimensionality space from N to K where K<N. In the case of collaborative filtering, we are interested in the matrix factorization part of SVD. The matrix in this case is formed by rows representing users and columns representing items (restaurants) and cells of the matrix representing the rating given by the user to the item.

$$\hat{r}_{xi} = q_i \cdot p_x^T$$

Where r_{xi} is the rating matrix, q_i is item matrix, and p^T_x is the user matrix. SVD gives the minimum reconstruction error, or sum of squared errors (SSE).

$$\underline{\min_{U,V,\Sigma}} \sum_{ij \in A} \underbrace{\left(A_{ij} - [U \Sigma V^T]_{ij}\right)}_{SVD}{}^2$$

SSE and RMSE are monotonically related, that is

$$RMSE = \frac{1}{c}\sqrt{SSE}$$

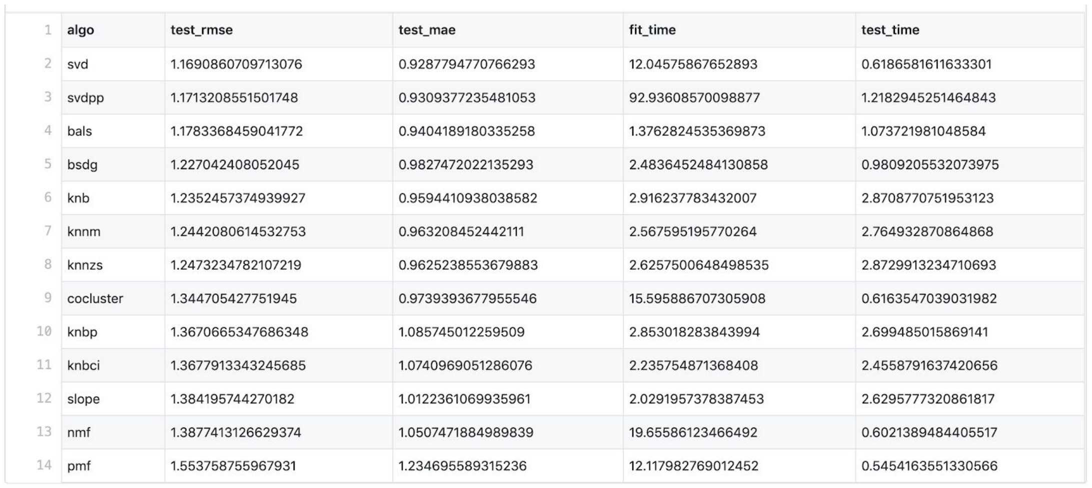
In other words, SVD is minimizing RMSE. SVD gives the best possible combination of user, item, rating matrix that minimizes RMSE. The equation of the objective function with regularization is

$$\min_{\underline{P,Q}} \underline{\sum_{training}(r_{xi} - q_i p_x^T)^2} + \boxed{\lambda}\left[\underline{\sum_x \|p_x\|^2 + \sum_i \|q_i\|^2}\right]$$

SVD uses gradient descent to minimize the objective function.
Refer to [20][21] for elaborate explanation of the above equations.

In addition to SVD, we also tested **SVD++, PMF, NMF, SlopeOne, CoClustering**, **BaseLine Alternating Least Squares** and **Baseline Stochastic gradient descent**.

Evaluation:
With these algorithms at our disposal, we had to research to pick which one would best meet our needs. We decided to use an experimental approach where we perform 5-fold cross-validation with all of these algorithms, and pick the algorithm with the lowest RMSE. The intuition for picking RMSE as a metric is described above in the section "Perform Cross-Validation"**.** Below are the results after hyperparameter tuning:



algo	test_rmse	test_mae	fit_time	test_time
svd	1.1690860709713076	0.9287794770766293	12.04575867652893	0.6186581611633301
svdpp	1.1713208551501748	0.9309377235481053	92.93608570098877	1.2182945251464843
bals	1.1783368459041772	0.9404189180335258	1.3762824535369873	1.073721981048584
bsdg	1.227042408052045	0.9827472022135293	2.4836452484130858	0.9809205532073975
knb	1.2352457374939927	0.9594410938038582	2.916237783432007	2.8708770751953123
knnm	1.2442080614532753	0.963208452442111	2.567595195770264	2.764932870864868
knnzs	1.2473234782107219	0.9625238553679883	2.6257500648498535	2.8729913234710693
cocluster	1.344705427751945	0.9739393677955546	15.595886707305908	0.6163547039031982
knbp	1.3670665347686348	1.085745012259509	2.853018283843994	2.699485015869141
knbci	1.3677913343245685	1.0740969051286076	2.235754871368408	2.4558791637420656
slope	1.384195744270182	1.0122361069935961	2.0291957378387453	2.6295777320861817
nmf	1.3877413126629374	1.0507471884989839	19.65586123466492	0.6021389484405517
pmf	1.553758755967931	1.234695589315236	12.117982769012452	0.5454163551330566

Figure 14. Cross-Validation RMSE and MAE results for 13 tested algorithms

For more detail on the hyperparameter tuning performed, refer to [22]. Figure 14 shows that SVD out-performed all of the tested algorithms and was picked as our algorithm for performing predictions.

Limitations:
Our recommendation system is based on collaborative filtering, which inherently suffers from the cold start problem, in that the system cannot draw any inferences about the users or items that have not interacted with the system. This being the fundamental limitation, this manifests into not showing up any recommendations for users along a particular route because there is lack of any data about restaurants or users along that route. Nevertheless routes along which data is available for restaurants and users, the system

was able to perform recommendations. There are ways in which this problem could be overcome by collecting metadata about new users while they register into the system and adding the feature of content based filtering and making this recommendation system an ensemble learner. Some of the approaches to overcome cold start problems have been discussed here.[23]

CONCLUSIONS AND DISCUSSION

This report documents the idea behind Genie, an innovative restaurant recommendation system that finds the best restaurants along a user-determined route, then displays visualizations to compare the recommended restaurants. Since the Yelp dataset was limited, Las Vegas was chosen as the city to demo the project, because it had the most number of restaurants over any other city in the U.S. The recommendation system described in "Machine Learning Pipeline" processes the data and returns the top 3 restaurants based on the user's taste and selected route.

We were able to successfully demonstrate an end-to-end proof of concept. For demo purposes, the available users were part of the existing filtered dataset. Ideally, the recommendations would be personalized for each logged in user, and the system would use the user's preferences to generate custom recommendations. For future work, the project could be expanded to cities outside of Las Vegas, especially if the Yelp dataset or other open source restaurant datasets were more up-to-date and included virtually all of the restaurants in the U.S. Also, the geofilter could be improved by using optimal routes between locations as a contour for filtering, as opposed to the bone-shaped adjustable contour that we used for our project. Lastly, although we felt the visualizations displayed important information, there could be more research and user testing performed to determine which visualizations would be the most useful for comparing restaurants.

DISTRIBUTION OF TEAM MEMBER EFFORT

All team members have contributed a similar amount of effort.

REFERENCES

1. Bejarano, Andres, et al. "Measuring User's Influence in the Yelp Recommender System." *PSU Research Review*, vol. 1, no. 2, 2017, pp. 91–104., doi:10.1108/prr-02-2017-0016.
2. Bresciani, Sabrina, and Martin J. Eppler. "The Pitfalls of Visual Representations: A Review and Classification of Common Errors Made While Designing and Interpreting Visualizations." *SAGE Open*, vol. 5, no. 4, 2015, doi:10.1177/2158244015611451.
3. Danone, Yaakov, et al. "Visualizing Reviews Summaries as a Tool for Restaurants Recommendation." *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval - IUI '18*, 2018, doi:10.1145/3172944.3172947.

4. Doan, Tri, and Jugal Kalita. “Sentiment Analysis of Restaurant Reviews on Yelp with Incremental Learning.” *University of Colorado, Colorado Springs*, 2016, cs.uccs.edu/~jkalita/papers/2016/DoanTriICMLA2016.pdf.
5. Hang, Lei, et al. “Design and Implementation of an Optimal Travel Route Recommender System on Big Data for Tourists in Jeju.” *Processes*, vol. 6, no. 8, 2018, p. 133., doi:https://www.researchgate.net/publication/327090246_Design_and_Implementation_of_an_Optimal_Travel_Route_Recommender_System_on_Big_Data_for_Tourists_in_Jeju
6. Keck, Mandy, and Dietrich Kammer. “Exploring Visualization Challenges for Interactive Recommender Systems.” *CEUR Workshop Proceedings*, May 2018, ceur-ws.org/Vol-2108/paper3.pdf.
7. Kumar, Sudhanshu, et al. “Movie Recommendation System Using Sentiment Analysis from Microblogging Data.” *ArXiv*, 27 Nov. 2018, arxiv.org/pdf/1811.10804.pdf.
8. Li, S., and J. Gong. “Mashup: A New Way of Providing Web Mapping/GIS Services.” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII, no. B4, 2008, pp. 639–648.
9. Pazzani, Michael J., and Daniel Billsus. “Content-Based Recommendation Systems.” *The Adaptive Web Lecture Notes in Computer Science*, pp. 325–341., http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.8327&rep=rep1&type=pdf
10. Perez, Luis. Predicting Yelp Star Reviews Based on Network Structure with Deep Learning. 2017. arxiv.org/pdf/1712.04350.pdf.
11. Saggi, Mandeep Kaur, and Sushma Jain. “A Survey towards an Integration of Big Data Analytics to Big Insights for Value-Creation.” *Information Processing & Management*, vol. 54, no. 5, 2018, pp. 758–790., doi:10.1016/j.ipm.2018.01.010.
12. Sawant, Sumedh, and Gina Pai. “Yelp Food Recommendation System.” *Stanford University*, 2013, cs229.stanford.edu/proj2013/SawantPai-YelpFoodRecommendationSystem.pdf.
13. Trattner, Christoph, and Elsweiler, David. “Food Recommender Systems: Important Contributions, Challenges and Future Research Directions.” *arXiv:1711.02760v2 [cs.IR]* 10 Nov 2017
14. Tsai, Chieh-Yuan, and Shang-Hsuan Chung. “A Personalized Route Recommendation Service for Theme Parks Using RFID Information and Tourist Behavior.” *Decision Support Systems*, vol. 52, no. 2, 2012, pp. 514–527., https://lms.ctl.cyut.edu.tw/sysdata/22/27122/doc/983f1e1f02a96e69/attach/1520796.pdf.
15. Vartak, Manasi, Huang, Silu, Siddiqui, Tarique, Madden, Samuel, and Parameswaran, Aditya. “Towards Visualization Recommendation Systems.” *SIGMOD Record,* Vol. 45, No. 4. December 2016
16. Wang, Ji, et al. “Clustered Layout Word Cloud for User Generated Review.” *Yelp*, 2013, www.yelp.com/html/pdf/YelpDatasetChallengeWinner_WordCloud.pdf.

17. Wang, Yong, et al. “Towards Easy Comparison of Local Businesses Using Online Reviews.” *Computer Graphics Forum*, vol. 37, no. 3, 2018, pp. 63–74., doi:10.1111/cgf.13401.
18. Liao, K. (2018, November 19). Prototyping a Recommender System Step by Step Part 1: KNN Item-Based Collaborative Filtering. Retrieved from https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea
19. Hug, N. (n.d.). k-NN inspired algorithms. Retrieved from https://surprise.readthedocs.io/en/stable/knn_inspired.html
20. Rajaraman, L. (2016, April 13). JP Search 9 Avatar image 0:01 / 14:16 Lecture 55 — Latent Factor Recommender System | Stanford University. Retrieved from https://www.youtube.com/watch?v=E8aMcwmqsTg&list=PLLssT5z_DsK9JDLcT8T62VtzwyW9LNepV&index=55
21. Rajaraman, L. (2016, April 13). Lecture 56 — Finding the Latent Factors | Stanford University. Retrieved from https://www.youtube.com/watch?v=GGWBMg0i9d4&list=PLLssT5z_DsK9JDLcT8T62VtzwyW9LNepV&index=56
22. LAT35N Team91 CSE6242 Group Project. (n.d.). Retrieved from https://github.gatech.edu/team91-cse6242-fa19/genie/blob/master/YR/hp_run.txt
23. Oshiba, K. (2018, June 5). Tackling the Cold Start Problem in Recommender Systems. Retrieved from https://kojinoshiba.com/recsys-cold-start/