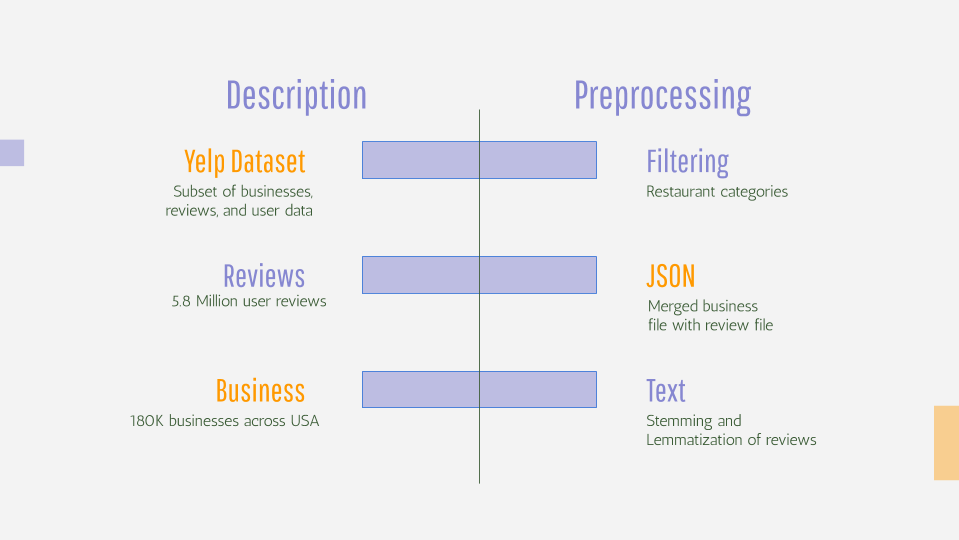


Business review websites such as Yelp, TripAdvisor allows users to rate businesses and provide reviews based on their experience and services provided by the businesses. These business reviews have a considerable influence on business growth as customers prefer businesses that have higher ratings and good reviews. Hence, we plan to build a system for both users and restaurants where restaurants can gain insights about their customers and services and customers can gain the benefit of recommended businesses using Unsupervised Learning techniques.

These reviews and ratings can also be used by the businesses in order to identify weak spots in their service, also to identify key issues provided by the users, and addressing such issues and weak spots can optimize their businesses and allow them to gain more revenue by improving their services. We used various Topic modeling techniques such as LDA, LSA, and Non-Negative Matrix Factorization to allow businesses to gain information about their weak spots in businesses such as High wait time, food quality, etc.

Not just a business side, but Yelp has a huge potential to gain more revenue by suggesting businesses to the users based on their previous likes and reviews as well as the requirements provided by the users. A single user cannot review all the businesses, and using the various reviews posted by other users, we can predict the user’s favorite businesses and recommend those businesses to gain more revenue. We used Collaborative and Text-based Recommender systems to help users identify more likable restaurant places based on similar user ratings and required attributes provided by the users.

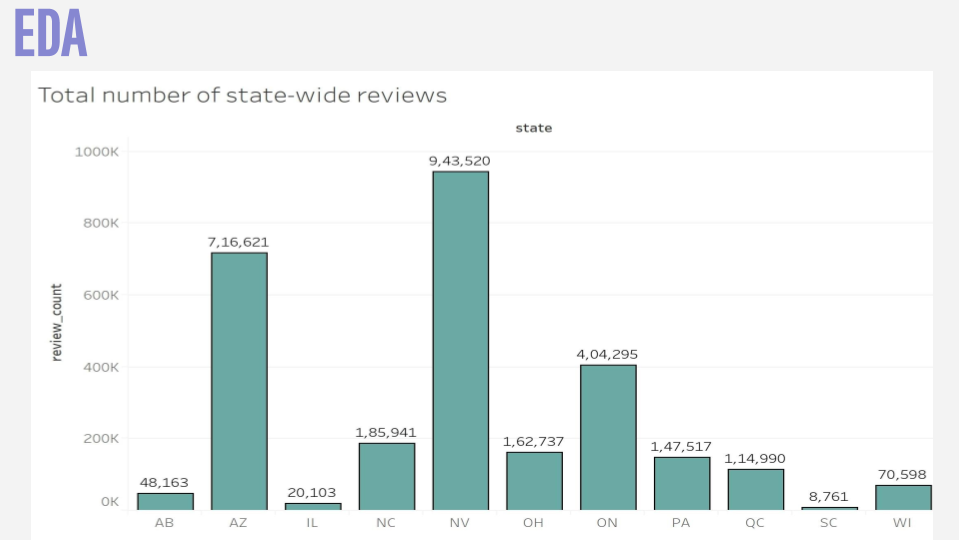
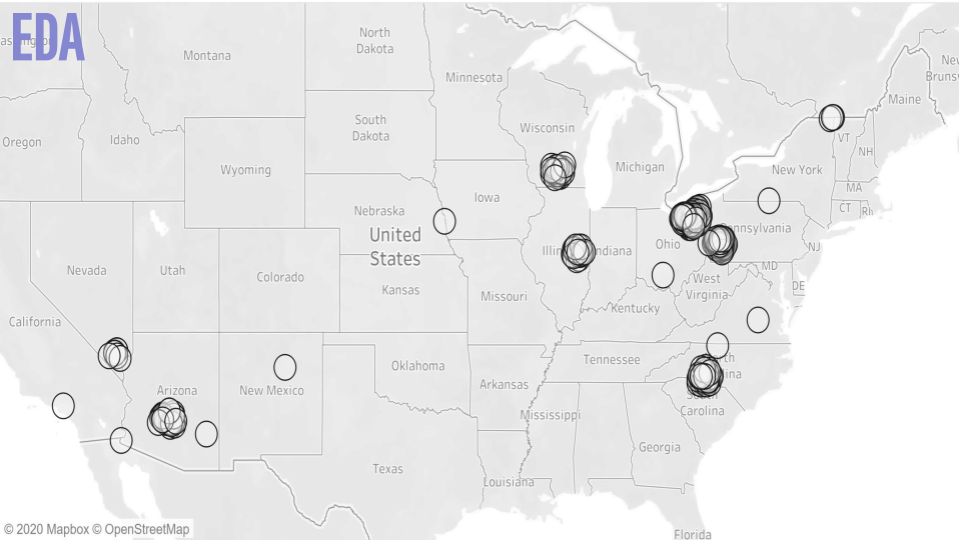


The dataset was provided by Yelp Inc. which contains 5 different files containing almost 5.8 million reviews given by almost 1.5 million users. There are a total of 180K businesses combined which are mainly located in Tier-1 cities of the USA and Canada. Along with the reviews, there are various business attributes provided by Yelp such as Pet Friendliness, Parking Availability, Noise Level, Price range, etc. Same as a business file, there is also a large dataset available for the user data such as user id, review count, etc.

For this project, we used two files reviews.json and business.json which contained information about reviews and businesses. There are various kinds of businesses such as Auto Care, Medical services. However, the scope of this project is only limited for the Restaurant businesses hence we only worked on restaurants that are open such as bars, cafes, restaurants, etc.

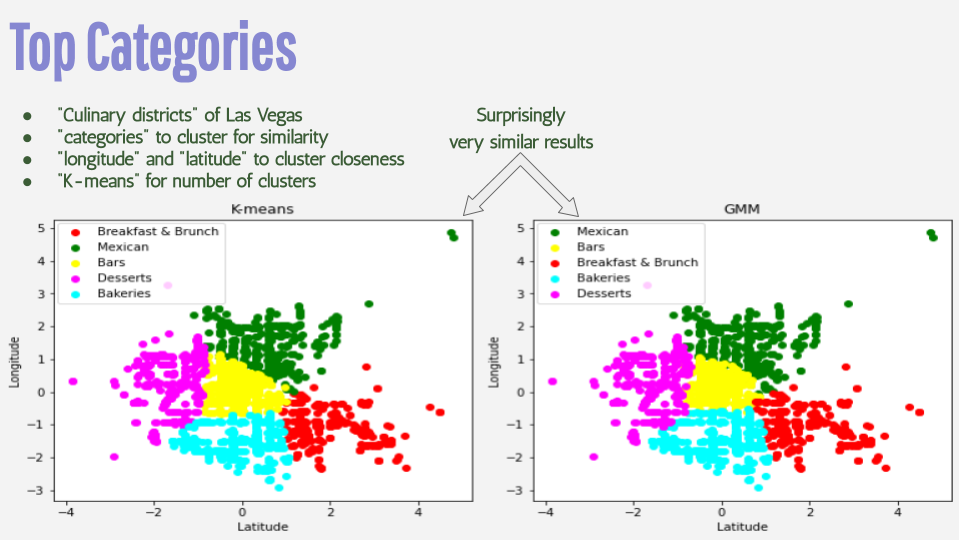
The business and reviews file was merged to filter out non-restaurant reviews.For the reviews, we performed various preprocessing tasks in order to gain accurate results. All the empty/null reviews were removed. Along with that, we also filtered out all the stopwords in the English language from the text. We also performed Stemming and Lemmatization of words in order to reduce the vocabulary size and to improve the accuracy of the results.

For the recommendation system, we chose to work on Las Vegas City restaurant businesses as that city had the maximum number of reviews. We avoided creating one global recommender system for all the restaurants, as it is very unlikely that the user will travel to different cities for the food.



It is important to know that our data is concentrated only on a few states in the USA. A map of the states having reviews indicates this. This makes sure that the user knows there is a bias involved in terms of preferring certain geographical locations over others

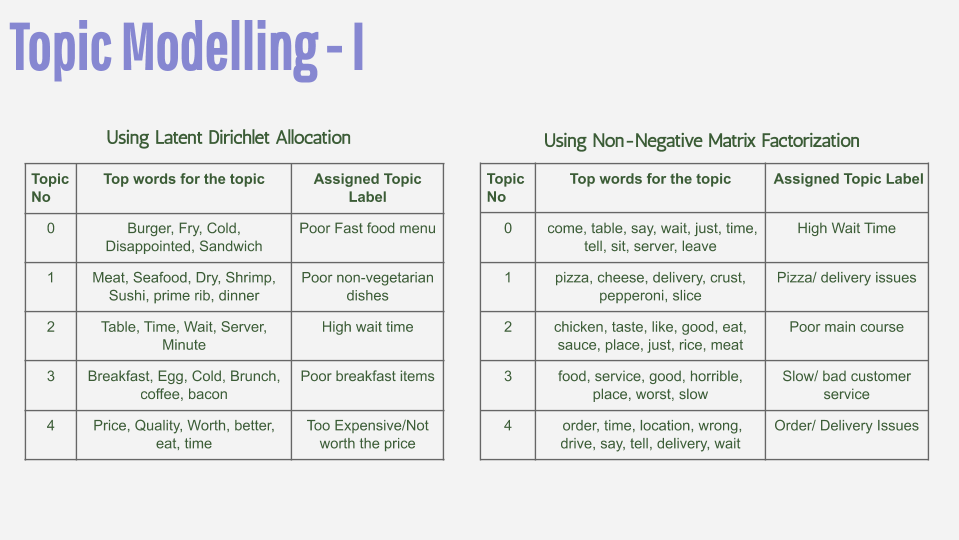
A bar plot of the total number of state-wide reviews shows us that Nevada and Arizona have the highest reviews. This insight is taken into consideration when deciding which cities/states are to be used for clustering and recommendation system



We parsed the data from the JSON file to a pandas data frame. The final data frame used for clustering has a total of 17 columns. Two of which are latitude and longitude and the remaining 15 are the columns of the top 15 categories of Restaurants in Las Vegas. If a category is present for a restaurant then its value for that column is 1 or else 0. Then performed feature scaling on latitude and longitude by subtracting the mean of the entire column from each value and then dividing by the standard deviation.

Also, we plotted error versus a number of clusters to select the appropriate number of clusters for kmeans++ method. As observed from Error vs Number of Cluster plot, there is a sharp decline till 5 and then the error decreases slowly. Hence, We selected 5 clusters for clustering the restaurants. We also performed Gaussian Mixture Modeling using co-variance as 'spherical' so that each component has its own single variance.

We also visualized the clusters by plotting the longitude/latitude of the restaurants in a scatter plot and labeled each cluster with a category. After calculating the ratio of each category present in the cluster with the total number of restaurants in Las Vegas of that category, we selected the category having a maximum ratio as the label for that particular cluster. Thus, avoiding dominance by a particular category having a large number of restaurants. K-means has more well-defined clusters because it does hard-clustering i.e. each point belongs to one cluster only. also, since the function used is kmeans++, the initial points are calculated based on a probability function. GMM uses soft-clustering and hence has overlapping clusters. It calculates the probability of a point belonging to each cluster and hence is more likely to give better clustering. It gives clustering similar to K-means but with some overlapping points. Eg: Centre clusters in both GMM and K Means are Bars.

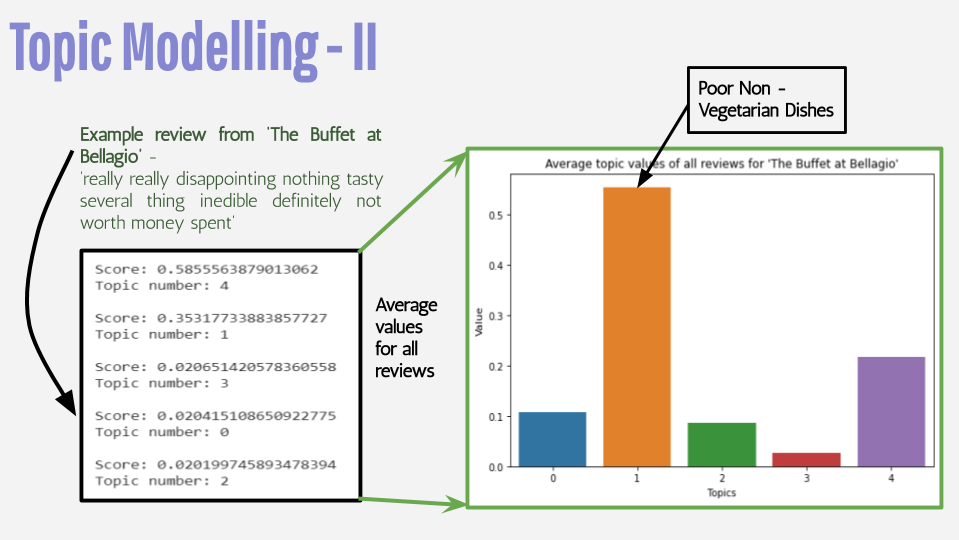


The idea behind this was to help restaurant owners understand the sentiment of customers towards their restaurants and identify potential areas of improvements using topic modeling (eg service, ambiance, parking, etc).

Topic Modelling is an unsupervised Machine Learning Technique that attempts to cluster a set of documents together by detecting word and phrase patterns within them. We are basically using this method to group different reviews together into a set defining the most occurring problems and get insights on what is going wrong or what can be further improved. LDA or Latent Dirichlet Allocation is one of the most popular topic modeling methods. The aim of LDA is to find topics a document belongs to, using 2 things- The words that belong to a document, that we already know and the words that belong to a topic or the probability of words belonging to a topic. The latter can be calculated by converting the text into a feature vector using CountVectorizer or using TfidfVectorizer.

Firstly we classified reviews as positive if the review ratings were 4 or 5 and classified them as negative if the review ratings were 1 or 2. Looking at the distribution of the reviews we realized that most of the reviews were positive. But in order to find out potential areas of improvement, we decided to focus our analysis on negative reviews only. To reduce the number of reviews we thought of selecting the restaurants with the highest number of reviews only. As a last step before extracting topics, we preprocessed the text reviews as explained before.

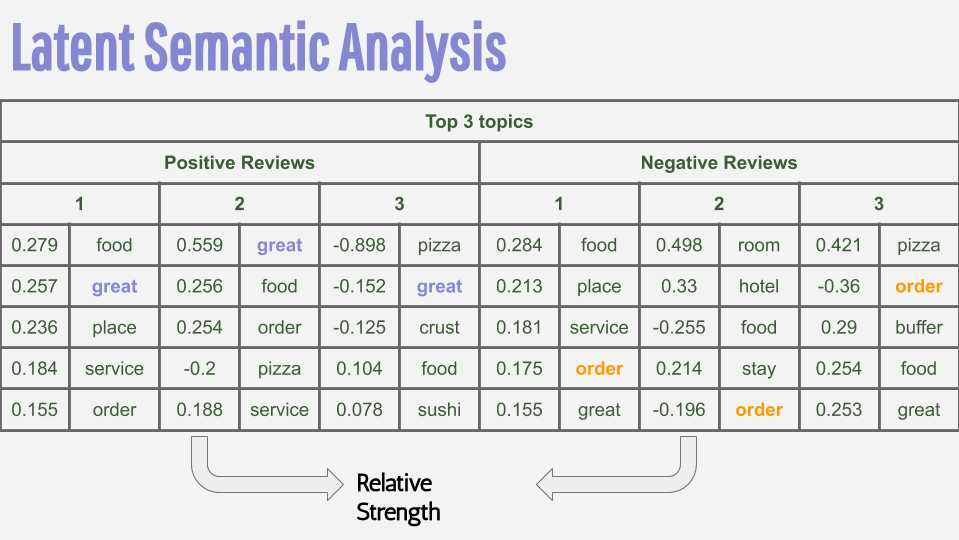
We experimented with a different number of topics and realized that larger values gave less distinct topics subjectively. Below is a table of topics when k=5, top occurring words for that topic and the assigned topic label. There was also some overlap of words in the topics. The overlap was more for NMF.



We tried generating topics using Non negative matrix factorization too which is another topic modelling technique that uses matrix factorization approach as opposed to the probabilistic approach used in LDA and figured that LDA does a slightly better job at generating well-defined topics and has less overlapping words. So we decided to use LDA for further analysis.

We tried to classify an unseen review from the restaurant ‘The Buffet at Bellagio’ and generated topic scores as can be seen in the slide. The model correctly classifies the review by generating the highest score for topic number 4 whose assigned label is ‘Too expensive/ not worth the price’. Now that we have a way to associate a topic to each negative review, we needed to figure out a way to put it all together so that restaurant owners can gain value out of it. For this, we generated topic scores from all the reviews for the above restaurant and calculated an average score for all the different topics. The above bar chart of the result gives the restaurant owner information on which areas are the customers complaining about and what the specific complaints are. It can be seen that topic 1 and 4 have the highest score indicating that the restaurant should improve the nonvegetarian dishes in their menu and maybe lower the prices since the customers find it too expensive/ not worth the price.

From a data mining standpoint, since we had a huge number of reviews for different restaurants across different cities we had to restrict ourselves to only the top restaurants with the most number of reviews. Also since topic modeling is an unsupervised technique, and since there is no objective way to determine the correct value of k - there was a lot of manual interpretation required. The topics we got were not individual distinct topics like we initially expected but a mixture of topics. In future, we could try tuning hyper-parameters of the algorithms(LDA and NMF) and see if we can get better topics.



LSA is an information retrieval technique that analyzes and identifies the pattern in an unstructured collection of text and the relationship between them. LSA itself is an unsupervised way of uncovering synonyms in a collection of documents. First, we check how the review lengths are distributed. We found that the distribution of the length of reviews is similar for each star rating. That's good - that means our tokenization should have about equal text data for both positive and negative reviews.

However, we had a lot more reviews that are positive than negative. We split 1 and 2-star reviews into negative reviews and 4 and 5-star reviews into positive reviews. Removed additional stopwords. Then we used TF-IDF. TF-IDF is an information retrieval technique that weighs a term’s frequency (TF) and its inverse document frequency (IDF). Each word has its respective TF and IDF score. The product of the TF and IDF scores of a word is called the TFIDF weight of that word. Put simply, the higher the TF-IDF score (weight), the rarer the word and vice versa. Store TFIDF vectors in a Pandas DataFrame to investigate further. Got mean for each column (word): highest means are the most important words.

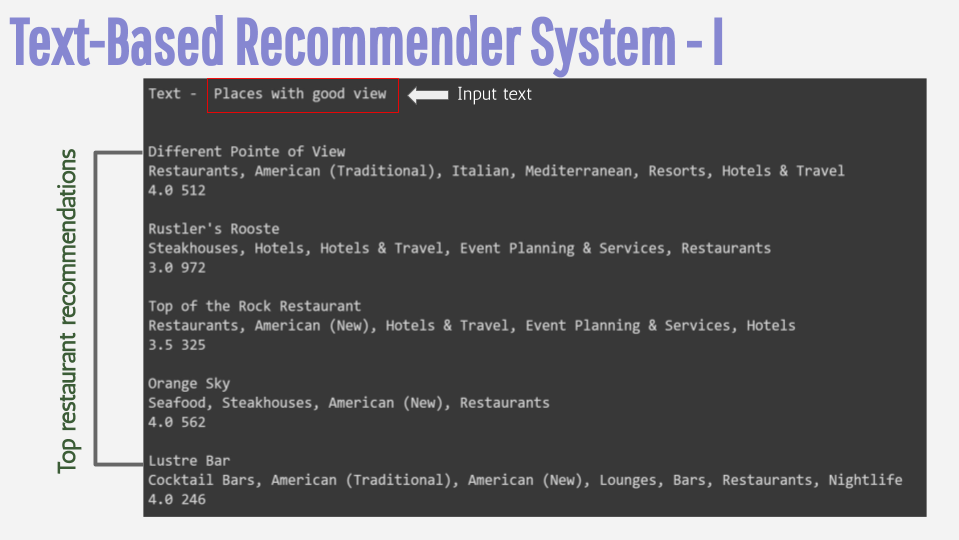
Then, the top 3 topics for both positive and negative reviews are found. A few takeaways: looks like the top words for both positive and negative Yelp Restaurant reviews mainly revolve around service order and food. There are subtle differences though. This shows that topic analysis is needed here to move past which words occur the most and analyze which words tend to group together in similar spaces. We find difficulty in interpretation with LSA as several of our topics have very strong negative weights for words.



The idea of collaborative filtering is finding users in a community that share appreciations. If two users have the same or almost the same rated items in common, then they have similar tastes. Such users build a group or a so-called neighborhood.

Collaborative filtering has basically two approaches: User-based approach: In this approach, items that are recommended to a user are based on an evaluation of items by users of the same neighborhood, with whom he/she shares common preferences. Item-based approach: Referring to the fact that the taste of users remains constant or changes very slightly, similar articles build neighborhoods based on appreciations of users. Afterward, the system generates recommendations with articles in the neighborhood that a user might prefer.

We got the unique users and restaurants from the dataset, and then split the dataset into train and test set (80:20) using train\_test\_split of sklearn. Here category, attributes, star, and ratings are used as features. We then created a user-item matrix (for train and test data), which consists of a rating of each user-item pair. Also, we created a user-user matrix(user\_similarity) which consists of a similarity score for each pair of users using pairwise\_distances of sklearn. After creating an item-item matrix(item\_similarity) which consists of a similarity score for each pair of restaurants using pairwise\_distances of sklearn, we calculated the predicted rating for each user-item pair for both the user-based and item-based approach. In the end, we evaluated the models using RMSE and MAE. Given user id, calculated the similarity between the other user’s and the input user’s profile using cosine similarity, we recommended the top 10 restaurants to the user.

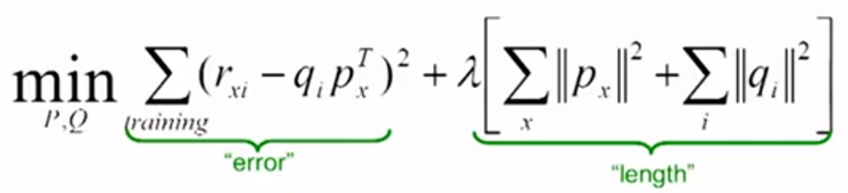


In order to build a recommender system using text reviews, we use the review dataset and the business dataset. The review dataset has a list of all users with individual reviews and the business dataset has the reviews attached to each restaurant. The first step is to read the data and clean it. We use the nltk library to remove stopwords and punctuations.

For each user, we combine all the reviews to form a single paragraph, after we combine it all then we apply the TFIDF Vectorizer to extract the features from the text. This is done after splitting the data-set into train and test set (85:15) using sklearn train\_test\_split. A similar approach is followed for each restaurant. Then, we create a matrix of users and businesses with the ratings.

We now have two matrices (user-features, business-features) that we can multiply to predict the ratings that a user gives to a restaurant. We now update the values in the features of our two matrices according to the Least Square Error (LSE). To prevent the model from overfitting, we also add regularization to our LSE formula. We then minimize the error using gradient descent.

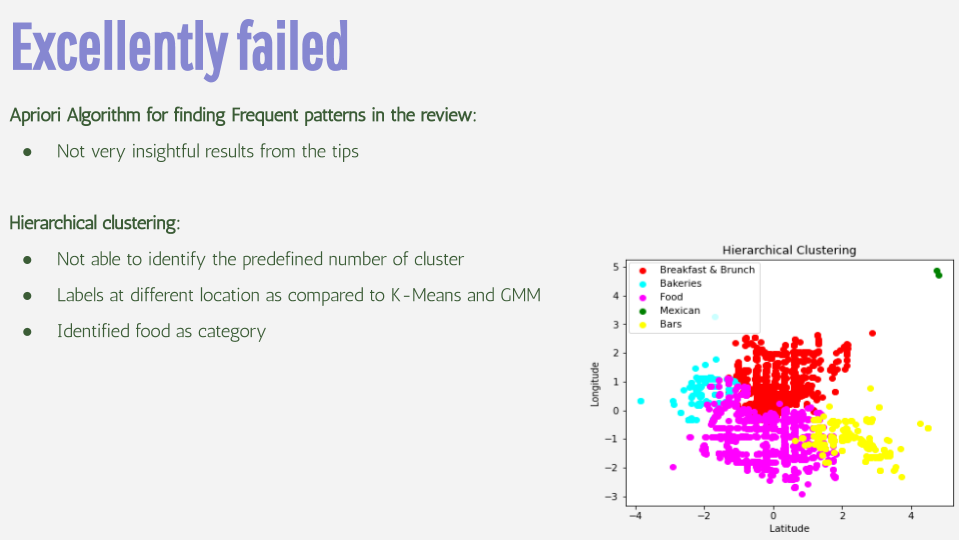
Here’s what the LSE looks like with regularization:





As seen in the two examples, this recommender system is extremely helpful not just to find suitable cuisines, but also for other aspects of dining people may find important like a place with a good view, a place with great ambience, a place with ample parking space or a restaurant that is child/pet-friendly.

Thus, given a text input, the model recommends restaurants with a similar match.



We applied Apriori Algorithm on the tips and review dataset in order to find frequent review patterns in the reviews. However, this method did not give us any interpretable results, as users used different words for explaining their reviews, which might have hindered the results of the Apriori algorithm. Also, the tips only contained few words and thus, lack of data also might have reduced explainability of the results. Thus, to get more category wise results like food quality, wait time etc., we found the LDA method more effective to perform topic modeling in a better and efficient way.

We used a complete link method for hierarchical clustering i.e finding the distance between the two points farthest from each other in order to combine clusters during the agglomerative clustering method. Since Hierarchical Clustering considers each point as a cluster on its own and then combines the points on the basis of distance between the points. As seen from the graph above, it gives a result which varies quite a lot from the K-means and GMM clustering methods since it does not use any random points. The points at the top right corner are considered as one cluster which should not happen. The clustering can have more error because agglomerative clustering is susceptible to chaining.