Machine Learning and Visualization with Yelp Open Dataset

Kunlun Liu: kunlun_liu@brown.edu
Tong Zhang: tong_zhang1@brown.edu
Zhiwei Zhang: zhiwei_zhang@brown.edu

Blog: Blog1-5 & Final Blog

https://medium.com/@zhiwei_zhang

Git repo:

https://github.com/zzhang83/Yelp_Sentiment_Analysis

Table of Contents

Introduction:

Yelp is currently the most widely used restaurant and merchant information software across United States. However, Yelp only provides us a holistic view about restaurant, such as giving overall review score or ratings and only a few reviews out of thousands of reviews. In order to improve Yelp users' experience, we dived deep into Yelp's open datasets as well as other data centers to retrieve useful information. Utilizing our complementary plug-in prototype, yelp users, whether they are business owners or consumers, are able to find information that better meet their preferences. Specifically, our group mainly focused on improving the customers' understanding of merchants, market knowledge for new business owners, and existing merchants' awareness about restaurants' features.

Vision and Goal:

In the first part of this work, we are examining the fake review detection data, which consists of 350,000 user reviews. The true and fake reviews in the data set help us train a model that predicts if a given review is fake or not.

In the second part of this work, we examined Yelp's merchant review data in the hope to retrieve useful information for customers to better understand a particular merchant and for merchants to improve their businesses. We obtained our data from Yelp's online data challenge, utilized machine learning techniques as well as natural language processing tools to retrieve insights from the data, and fit statistical models to the data so that we are able to access the most relevant keywords in the reviews that affect review scores.

In addition to the data processing pipeline we have developed, we also created a database to store the Yelp's review data. Our database are consists of multiple tables that describe different aspects of the data which allow us to do effective joining and querying. The database enables the flexibility of extracting, modifying, and storing the data as well as making our work easier to deploy as a web service. After constructing the database and the data processing pipeline, we packaged the program into a API which could handle HTTP requests and reply to users with analytical results.

In the last part of our work, we have incorporated the household income dataset which describes the income information in 6 states, namely Pennsylvania, Nevada, North Carolina, Illinois, Ohio, and Arizona. Combined with the merchant price range in our Yelp dataset, we mapped the average income and the average price range of restaurants in each county to help our new restaurants' owners to realize the relationship between the two, determine the potential size of the customer market, and ultimately to develop optimal pricing strategies.

Data Overview:

a. **Data:** Yelp Open Dataset
   Source: Yelp Dataset Challenge
   https://www.yelp.com/dataset/challenge
   This dataset includes information on reviews, users, businesses, checkin, photos and tips. The dataset is 5.79 gigabytes uncompressed in json format (6 json files, including business.json, check-in.json, photos.json, review.json, tip.json and user.json)

   Description: Our project mainly focused on reviews, users, and business datasets from Yelp open data source. For reviews, we kept the unique id of each review, user id for people wrote the review, business id for restaurants that the user wrote it for, the review content, the rating according to the review, and the date when the review was wrote. We also added in one geolocation character into the review data, which indicates the restaurant's location where the review implied. We kept user id, the number of reviews that an user has written, the time since an user joined Yelp, and the average ratings of reviews that an user has written from the user.json file. Last by not the least, we only included business id, name, business categories, geolocation information (city, state, postal codes, latitude, longitude), price range, ratings, number of reviews, and whether the restaurant is open or not from the business dataset.

   Data Cleaning: We only kept business information from 6 states in the United States. Since we are only interested in restaurants, we filtered out all other types of businesses in the business dataset. Moreover, we noticed that the closed restaurants did not contain much useful review information, so we removed restaurants that are not open by "is_open" column. With a cleaned business dataset, we matched business ids in both business and review datasets in order to remove all the reviews that are irrelevant to the restaurants we cleaned. As a result, we obtained cleaned versions of review and business datasets.

b. **Data:** Yelp Reviews with Deceptive or True reviews' labels

This dataset contains labeled customer reviews from Yelp (Deceptive reviews and True reviews), which is used for training our predictive models on Fake Review machine learning approach.

Source: It is provided by Professor Rayana, a professor from the department of Computer Science at Stony Brook University. Rayana's research group worked with this dataset to test their fake review detection model.

Data Cleaning: The methods for extracting columns and merging datasets are listed under fake review machine learning model section.

c. **Data:** Income Dataset and Zip code Dataset

Source: 1) American FactFinder under United States Census Bureau
2) Gaslamp Media

1) https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_16_1YR_S1902&prodType=table
2) https://www.gaslampmedia.com/download-zip-code-latitude-longitude-city-state-county-csv/

Description: The goal for bringing in both datasets is to compare the restaurants' price range to the average income level at their counties. With an indication of the relationship, existing or potential restaurants' owners are able to understand the macroeconomic trends of the market, which could help them to develop more profitable price strategies. The income dataset has information on household income based on family members, social status, and racial characteristics separated by county. Our project mainly focused on the average household income by county and state. The zip code dataset contains information of zip code in each city of each county in each state. We merged the income and zip code dataset together to add a column of zip code in the income dataset, which serves as the primary key of the dataset. In the succeeding analysis, we simultaneously mapped the business location data (longitude and latitude) and income zip code data onto an U.S. map.

Data Cleaning: Extracting relevant columns from raw data for both datasets, we noticed different columns have varies format that may cause problems in the merging step. So we changed state names into shorthand, lowered all cases, deleted all spaces, eliminated the suffix “county” in the county column, and created a combined column in both datasets. The combined column is unique, taking in a county as the first part and a state shorthand as the second part.

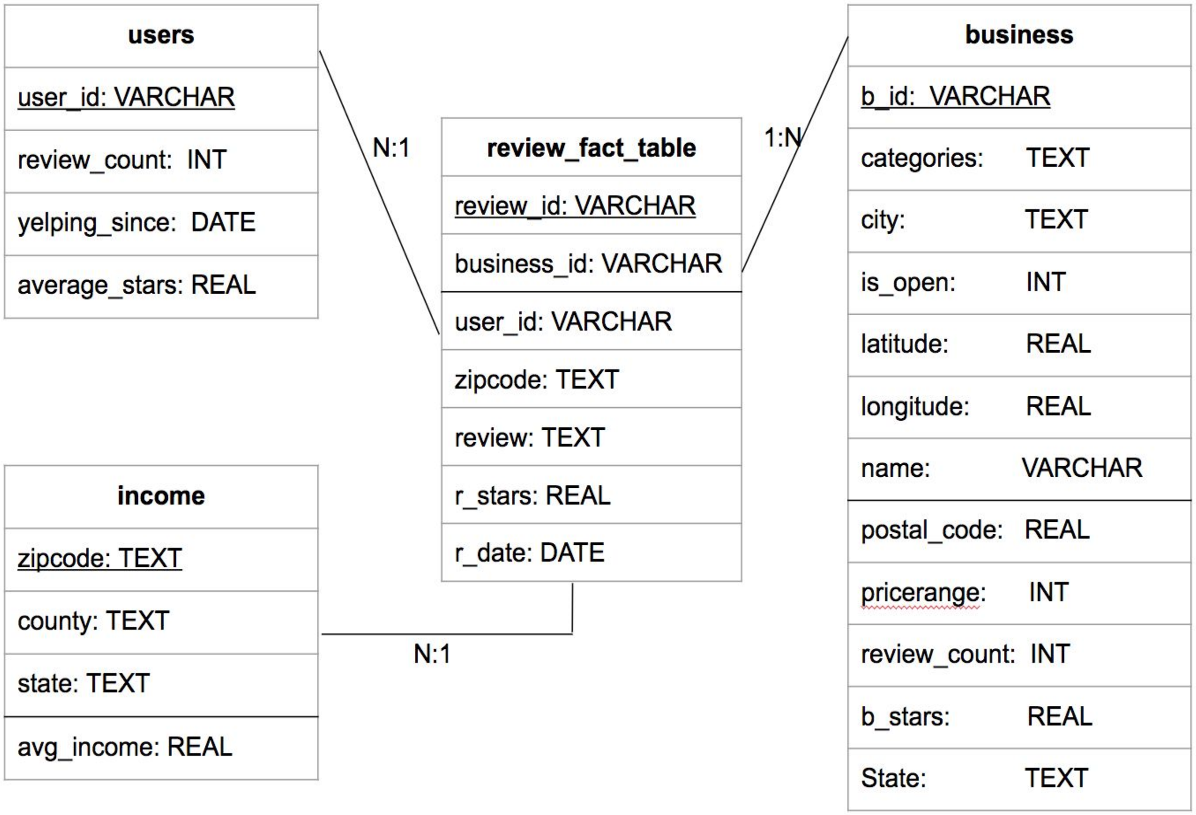
SQL Database:

With cleaned reviews, users, business, income & zip code, and fake review labeled data in five separated csv files, it is essential to connect them in order to reduce the time required for merging and extracting data for later models. It also enables scalability in the future if we have to merge or update more data into these datasets. With all the benefits, we built a SQL database with four out of five csv files (reviews, business, income & zip code, and users) utilized the following relationships:

- A business may contain multiple reviews.
- An user may write many reviews for different restaurants.
- Each review can only be written by one user about one restaurant. Review id is the primary key for the *review_fact_table*, which connects *users* table and *business* table.
- *Users* table and *review_fact_table* has N: 1 relationship, meaning 1 user id has N review ids, but 1 review id can only define 1 user id.
- *Business* table and *review_fact_table* has N:1 relationship, indicating 1 business id has N review ids, but 1 review id can only define 1 business id.
- *Income zip code* table and *review_fact_table* has a N:1 relationship, since a zip code can have N review ids, but 1 review id can only imply 1 zip code.
- Business_id, user_id, and zip code are all foreign keys in the *review_fact_table*.
- All user_id, business_id, and review_id are mixture of letters and numbers, so we used VARCHAR instead of integers. Ids are pre-populated by Yelp, so we do not need to create unique ids by ourselves.

Fake review label dataset was used for the fake review detection model only, so we did not include the dataset into our SQL database.

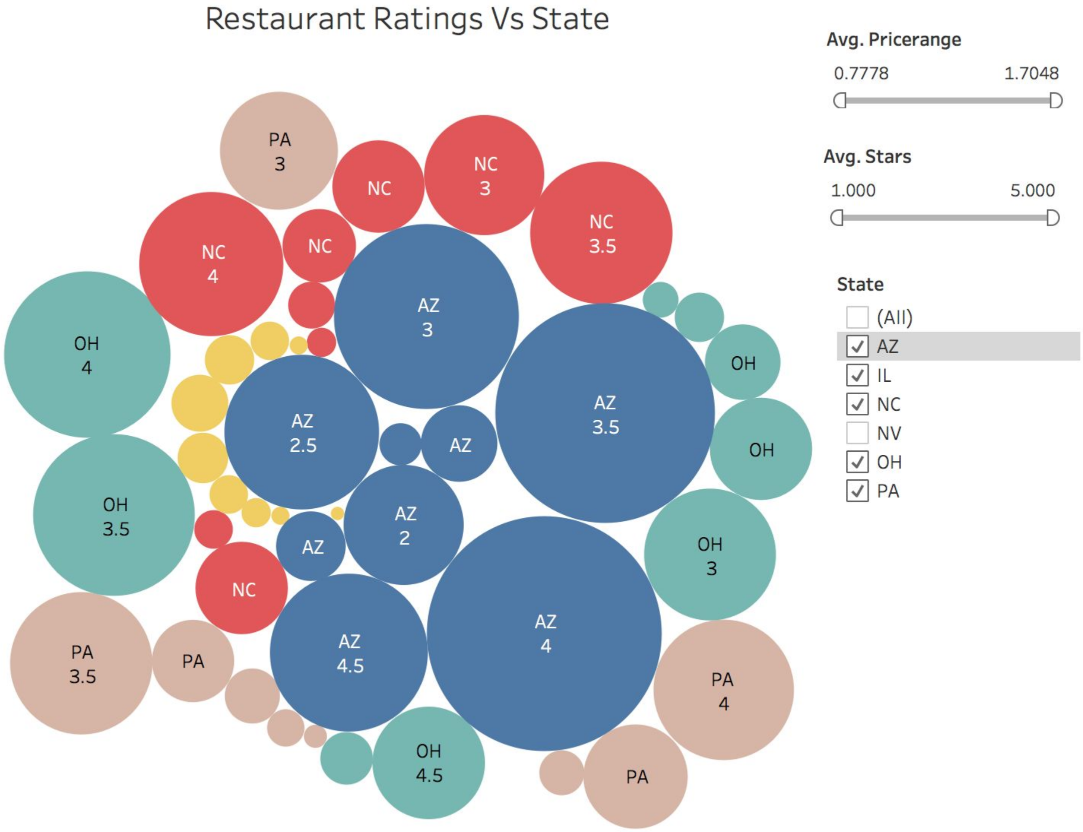
Yelp Database Star Schema



Exploratory Statistics:

Before jumping right into our machine learning models, we explored and familiarize ourselves with these datasets through graphs and some preliminary analysis. For each of the above dataset, we tried to find patterns and potential problems which could be useful information or traps in the later machine learning processes. Three main areas we explored on are star ratings' distribution, restaurants categories, and income versus restaurants' price range in each of the six states.

a. Star ratings' distribution



Restaurant Ratings Vs State

Avg. Pricerange

0.7778 1.7048

Avg. Stars

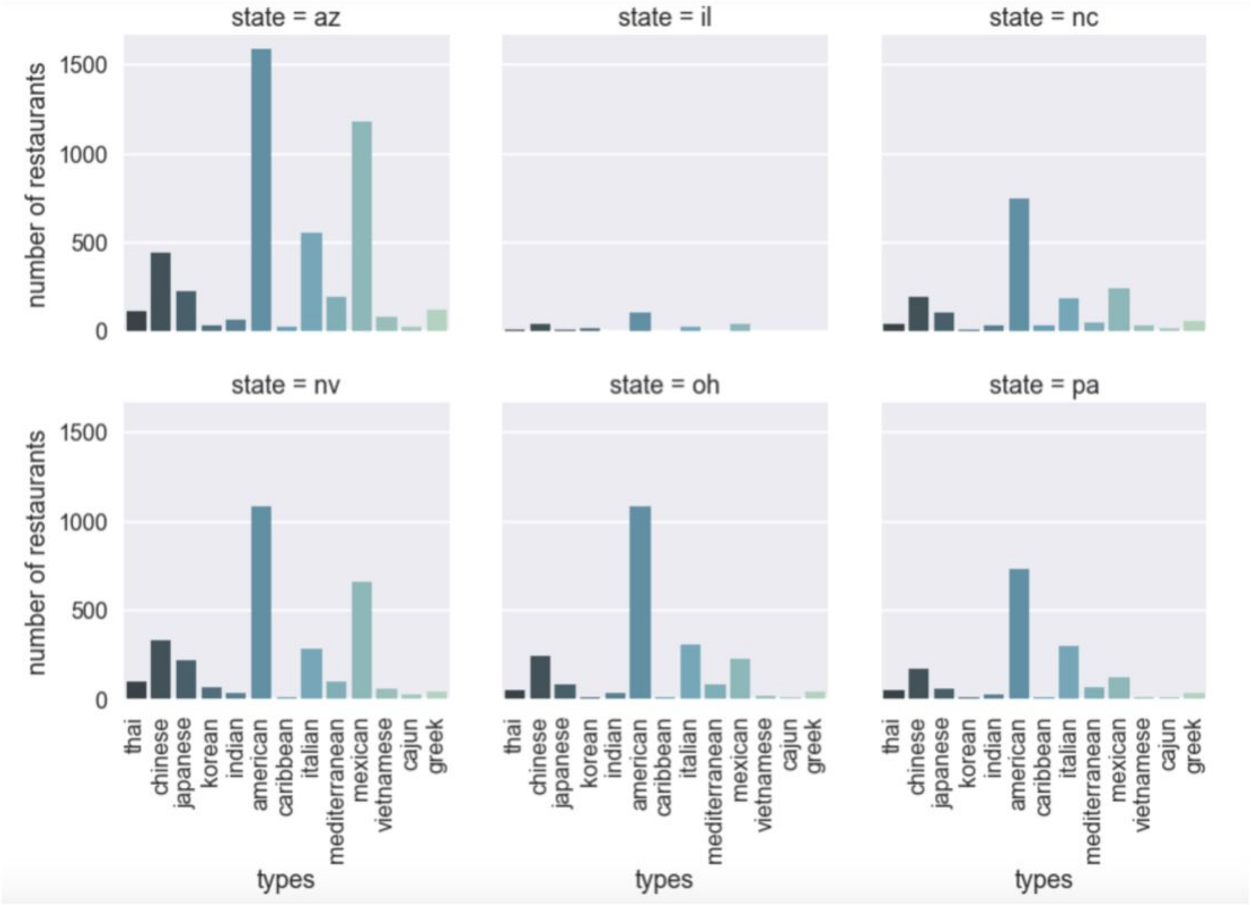
1.000 5.000

State

☐ (All)
☑ AZ
☑ IL
☑ NC
☐ NV
☑ OH
☑ PA

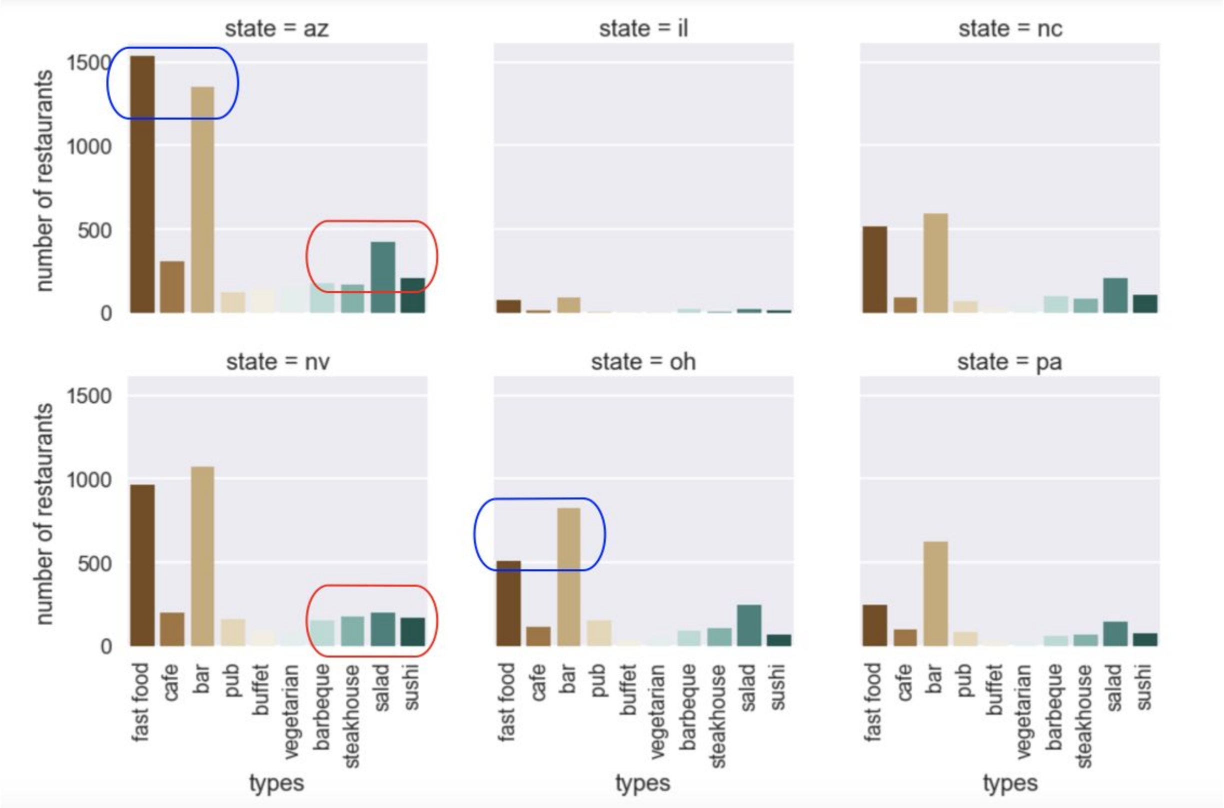
Each color indicates a state. Grouped by the ratings from 1 to 5, the number of restaurants in each stars is demonstrated by the size of the circle. For example, the majority of the restaurants are rated 3.5 to 4 in Arizona because the size of the circles is the largest at these two ratings. In fact, most of the restaurants are rated 3.5 to 4 stars in all six states, which alarmed us the existence of fake reviews. Restaurants are too heavily concentrated in the above-average-rating range to be realistic. In order to comprehend key insights of businesses, it is necessary to filter out the fake reviews from yelp restaurants first.

b. Restaurant categories I



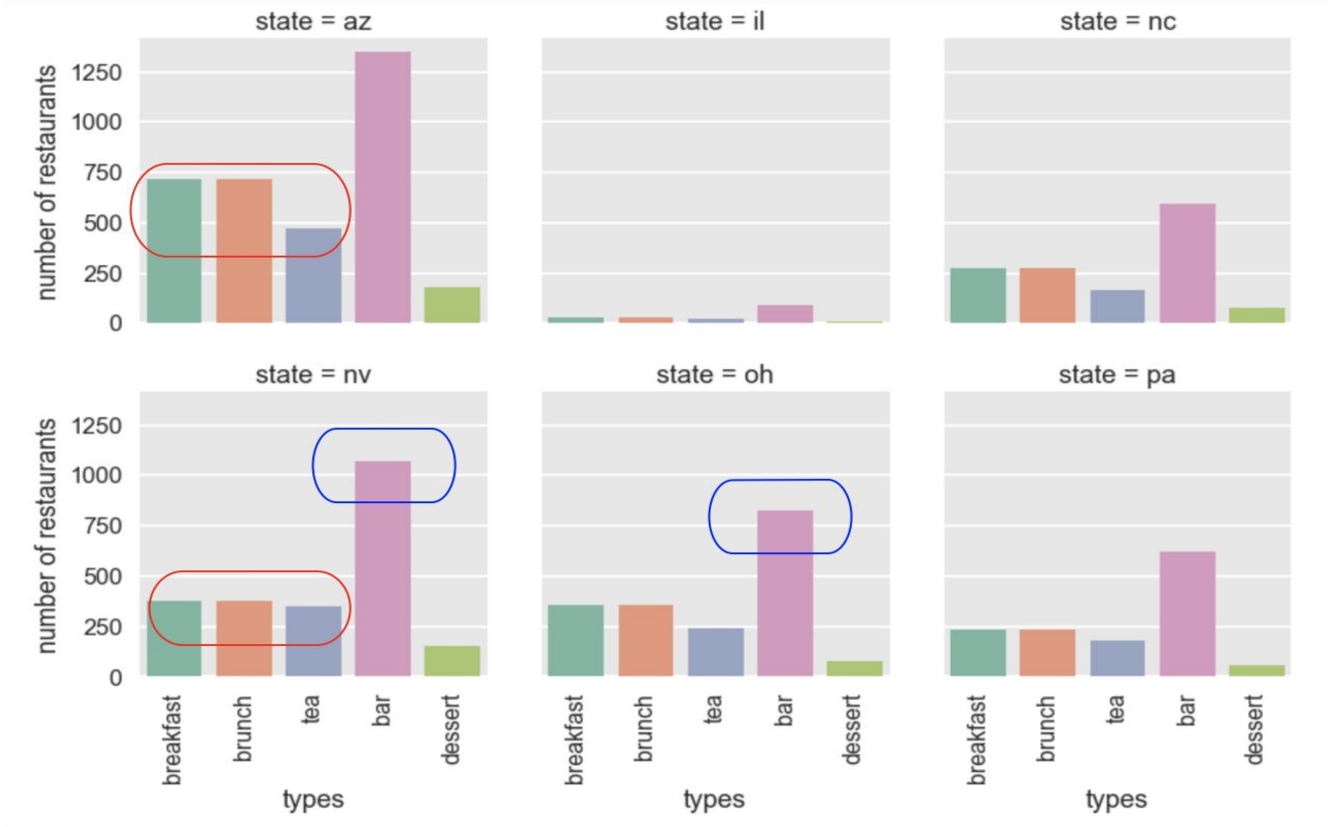
We picked 13 most common restaurant categories based on countries (Thai, Chinese, Japanese, Korean, Indian, American, Caribbean, Italian, Mediterranean, Mexican, Vietnamese, Cajun, Greek) and visualized the difference on the number of each type restaurants in six states. In the graphs above, Arizona state has the most number of restaurants, where Illinoise has the least. This may rise a problem for us if we would like to investigate Illinois restaurants in the later sections. In all states, both American and Mexican restaurants have the highest numbers, with Italian and Chinese restaurants in the third or fourth. The above graph may indicate the preference and the demand of consumers in the six states, though other factors and confounding variables may fill in additional knowledge about consumers' tastes.

c. Restaurant categories II



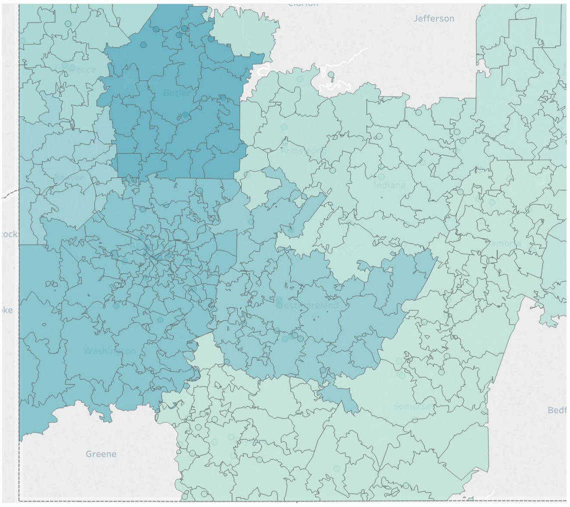
Instead of categorize restaurants based on countries, we would like to investigate the number of restaurants in various types. Extracting keywords from the "category" column of the business dataset, we separated them into fast food, salad, buffet, cafe, bar, pub, vegetarian, BBQ, steakhouse, and sushi. From the observations, we noticed that Arizona has the highest numbers of fast food restaurants and bars among all 6 states, with 200 more fast food restaurants than bars. Comparing to Ohio and Pennsylvania, we learned that fast food restaurants are much less in numbers than bars in these two states. Again, compared to Nevada, the number of BBQ, steakhouse and sushi are about the same as that in Arizona. However, Arizona has much more salad restaurants than Nevada's salad places. Combining both findings together, we came to the conclusion that Arizona consumers prefers more "quick" food (food needs short time to prepare or eat) than the consumers in the other five states. This could be an useful information for existing and new merchants who might think about developing new dishes or opening new restaurants, since time maybe the keyword for consumers' demand. Later in the natural language processing section, we will present a more precise model and analysis on key words about restaurants.

d. Restaurant categories III

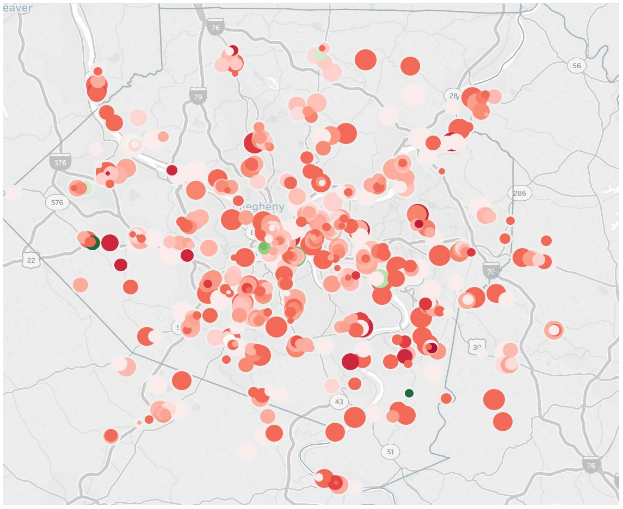


Furthermore, we also examined the restaurants based on the time of the day that the restaurants are defining themselves with. We broke down into five categories: breakfast, brunch, tea, bar, and dessert, where tea represents afternoon tea and milk tea places, and dessert is an indication for late night meals. Since lunch and dinner keywords are not present in the "category" column of the business dataset, we excluded them from this graph. With similar numbers of breakfast, brunch, tea, and dessert restaurants, Nevada has more bars than Ohio. Also, the demand for tea places is higher in Nevada than in Arizona because Nevada's number of tea places is only slightly lower than that of Arizona, when Nevada has much lower number of breakfast, brunch, dessert, and bars. Combining the two observations, we determined one of Nevada customers' preference is leisure, since both bars and tea stations are for people to relax, chat, and enjoy the time. Again, we would like to prove or use the patterns we found in these graphs to compare and apply to our machine learning models, so that we could summarize our results more precisely by excluding some of the unknown but confounding factors.

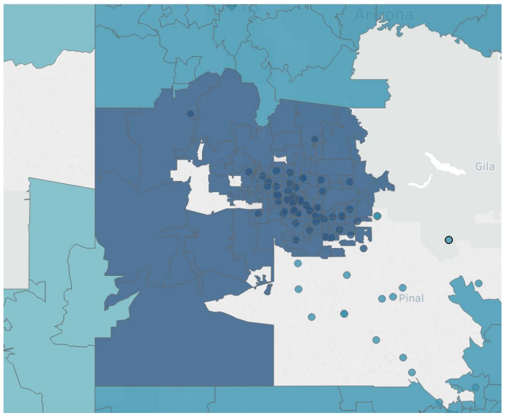
e. Price Range VS. Income Level



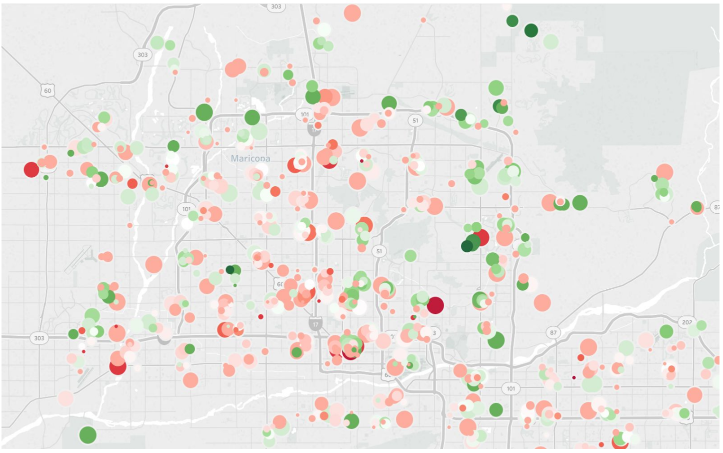
PA Income Level



PA-County Restaurant Pricerange



AZ Income Level



AZ-County Restaurant Pricerange

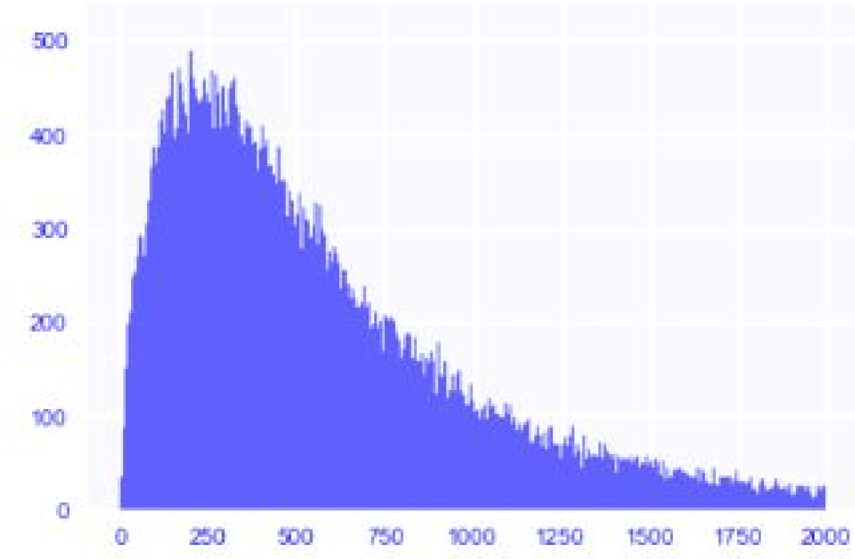
Utilized income zip code and Yelp business datasets by extracting zip code, average income, price range, and business geolocation information, we plotted the income level by county and mapped each restaurant's longitude and latitude of every state. Setting both maps side by side, we determined a direct relationship between the restaurants' price level and people's average income in these nearby restaurants. As income level increases, demonstrated by the gradually shading to dark blue/green color, the price range of restaurants rise, with changing from pink to green. This insight could be helpful for new or existing business owners: new merchants are able to set up food and service price (including tips) based on the most current income level of people in the nearby neighborhoods, and existing merchants can adjust item prices based on the current income level if the macro-economy has a recovery or recession.

Methodology:

A. **Fake Review Detection:**

Data Analysis:

1) By working with data analysis, we found that our data is very unbalanced. The amount of true reviews is about 10 times more than the number of fake reviews. To train a proper machine learning model, we solved this problem by replicating fake reviews 10 times to have the same amount of sample as the number of true reviews.



2) By graphing the distribution of the length of each review, we also noticed that the length of most reviews are within 500, which helped us to choose the best embedding parameters for our deep learning model.
3) There are also reviews wrote by other languages such as Chinese and French. Since these comments are rare, and we are only interested in English, we removed all foreign reviews.

Machine Learning Model

- **Model #1**: Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM)

Pre-process: Before the training process took place, we transformed our text data into numerical form. We utilized Kera's pre-processing module called tokenizer, which pre-processes text automatically by its insertion functions. The first step of tokenzier is to split the text by space, filter out all punctuations, and convert text to lowercase. Keras then transforms each word into numerical representations. In an given dataset, keras presents the most common word as 1, and the second most common word as 2, and so on. Since rare words cannot provide useful information for neural network models but only add noise, Keras is very useful in ignoring these rare words.

1) **Embedding layer:** The layer expands each token to a larger vector, allowing the network to represent words in a meaningful way. We passed 20,000 as the first argument, which takes in the most important 20,000 terms in the model. This parameter is flexible for tuning. We chose 20,000 because there is a good tradeoff between computational expenses and accuracy. To obtain a significant improvement, we would need a much longer training time.
2) **Convolutional layer:** The layer passes a filter over the data, and calculates a higher-level representation. The convolutional layer has been performed surprisingly well for text, too. It is also faster because the different filters can be calculated independently of each other. By adding convolutional layer before Long Short-Term Memory layer (LSTM), we allowed LSTM to learn sequences of chunks instead of sequences of words.
3) **Dropout layer:** a dropout layer is applied before convolutional layer because we would like to reduce a certain amount of data before feeding it into the neural networks to avoid overfitting problems.
4) **Max-pooling layer:** A max pooling layer followed by a convolutional neural networks is always efficient because it extracts higher-level representation of the dataset, which indicates more information will be expressed by max pooling.
5) **Long Short-Term Memory:** LSTM is a one layer recurrent neural network, which is known to perform very well on the text data. LSMT is designed to learn sequences of data, since it learns terms where each word is related to those before and after it in a sentence. Therefore, it works well for our goals. In addition, LSMT allows the neural network to pay more attention to certain parts of a sequence and largely ignore words which are not useful.

- **Model #2**: Support Vector Machine Classifier

Pre-process: We used feature extraction module called *TfidfVectorizer*, which is a scheme that transformed each review to a large sparse matrix with each cell represents a word and the frequency it appears in that review. Then TfidfVectorizer normalizes the counts by dividing the total number of times that the word appears in all reviews. Setting ngrams to 2, we considered all phases that contain two words. By assumption, a larger nram will have a better performance, but requires a larger dataset. With a try and error, we determined a good tradeoff by using bigrams. We also set min_df to 3, which removes all the words that appear less than 3 times in the document. The reason is that words with very rare appearance in the text are not useful for model improvement. Besiders, it only adds noise to our model.

Conclusion:

We trained two models: a deep learning model with convolution and long short-term memory layers using tokenized text data, the other is a linear support vector machine, utilizing 2-gram vectorized text data. The cross validation scores for deep learning and support vector machine models are 0.8976 and 0.8973. Though both methods have roughly the same performance, support vector machine is much faster than deep learning model. With a try and error, we decided to apply ensemble methods in order to combine the results of the two techniques. As a result, the performance is improved significantly.

B. Identifying Restaurant Features via Sentiment Analysis

The overall rating for each restaurant in Yelp is only useful to convey the general experience. There is not enough information for independently judging other aspects, such as the service, food quality and environment. If we only look at the ratings, it is difficult to understand why the restaurant is rated as 4 stars or 2 stars.

We are going to help both customers and merchants to understand the restaurant better by providing essential features behind all kinds of restaurants using Support Vector Machine model.

Text preprocessing:

1) In order to perform machine learning on the feature vectors, we required to process our text data to convert text into vector format by splitting a review into individual words and returning a word list.
2) We also removed punctuations and the stop words such as ‘the’, ‘a’, ‘this’, etc utilizing the NLTK library.

3) After converting each review into a word list, we applied *WordNetLemmatizer,* a process of grouping together different inflected forms of a word, to analyzed them as a single item. For example, the base form ‘go’ may appear as went, gone, going, and goes in reviews. If we utilized lemmatizer, we can analyzed all these variations as “go”.

Machine Learning Model:

1) Instead of employing the counts of each word in reviews with the bag of words in our machine learning model, we assigned *tf-dif* method to each term in our reviews. *Tf-idf* normalizes the count by dividing each word count by the number of reviews this word appears in.
2) After we convert our reviews as lists of tokens, we used scikit-learn *TfidfVectorizer* to combine our reviews’ vectors into m x n matrix containing our tf-idf scores, where each row of the matrix represents a single labeled review and each column represents a word term. The result is a 2-D matrix where each row is a review and each column is a unique word.
3) We applied support vector machines on the transformed data (tfidf matrix), utilizing review ratings as the model’s label. (If the stars ≥ 3, the review is positive, indicated by ‘1’, and if stars < 3, the review is negative, indicated by ‘0’)
4) We would like to learn the importance of words on creating positive or negative reviews in a restaurant. Considering SVM is an efficient model for text classification is because the linear SVM creates a hyperplane by using supported vectors to maximize the distance between the two classes. The weights obtained from svm.coef_ and the absolute size of these coefficients represent the importance of a feature for separating the two classes, which is a indicator of a feature influence level on positive and negative reviews.

Results & Visualization:

A. Fake Review Detection:

- Support Vector Machine:

We tried to train support vector machine on the unbalanced dataset. As demonstrated from the confusion matrix, the model has highly preference on true reviews.

	Predicted False	**Predicted True**
Actual False	177	14552
Actual True	550	128304

Table1: Confusion Matrix before balancing the training set

In order to train a proper machine learning model, we duplicated fake reviews ten times to have the same amount of sample as the number of true reviews. The result under the confusion matrix is listed below.

	Predicted False	**Predicted True**
Actual False	69642	3854
Actual True	10323	118780

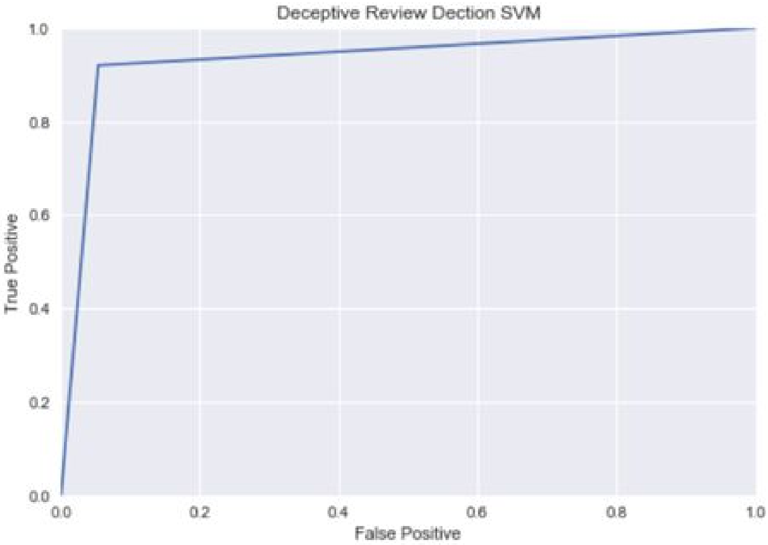
Table2： Confusion Matrix after balancing the training set

True positive decreased slightly, but true negative improved a lot. Though the overall accuracy is slightly lower, we chose to utilize balanced data to train our model because we would like to detect fake reviews as much as possible.

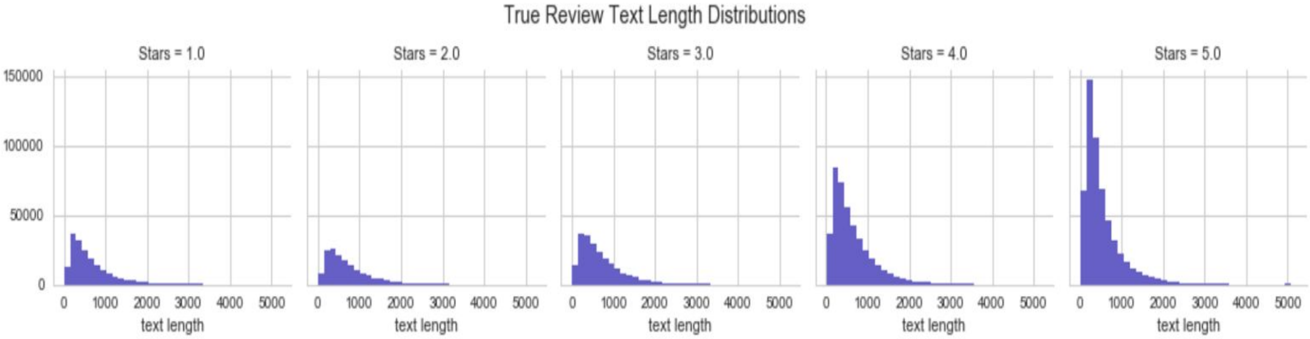
Statistics for Support Vector Machine:				
	Precision	**Recall**	**F1-score**	**Support**
	0.87	0.95	0.91	73499
	0.97	0.92	0.94	129100
avg/total	0.93	0.93	0.93	202599

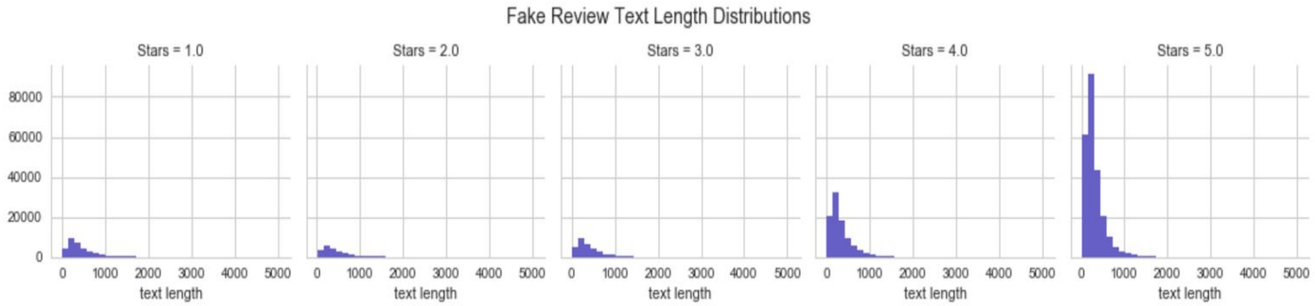
Table3: Statistics for Ensembled Model

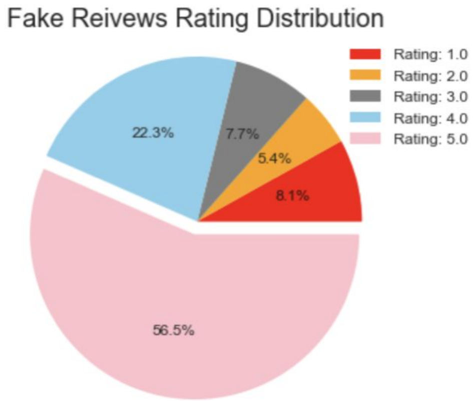
Here we have a ROC curve that demonstrates our results.

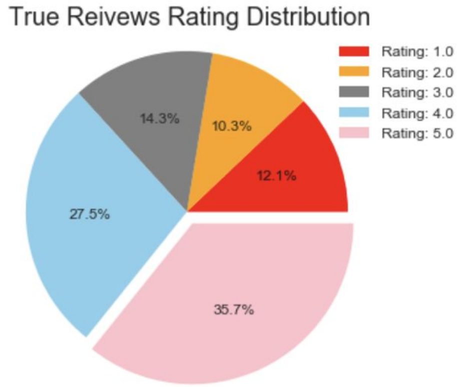


We also compared the text length distribution between true review and the fake review we detected.







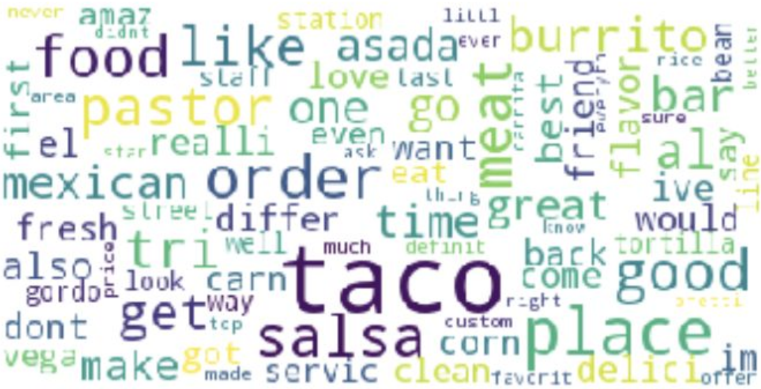


The pie chart demonstrates fake reviews tend to give more five star ratings than true reviews would give. As illustrated in the left pie chart, over **fifty percent** of the fake reviews are in the five stars' reviews.

B. Information retrieval based on true reviews:

Fake reviews created bias in users' judgements about the quality of a restaurant in Yelp. Thus, filtering out fake reviews is an important step for us to let customers really understand a restaurant and let Yelp better serves its original purpose.

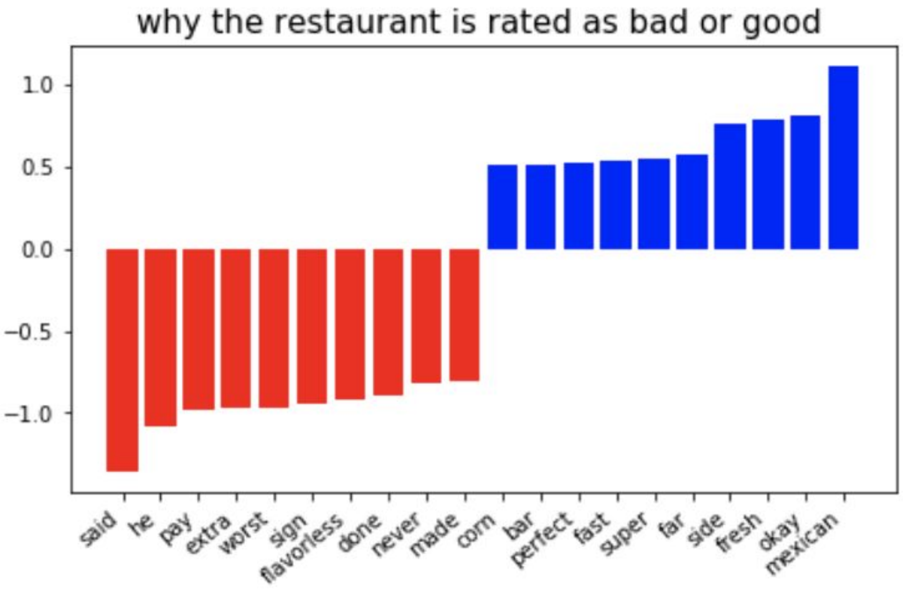
With the true reviews for a restaurant, we first created a **Word Cloud** to give customers an overview of a word appearance frequency in the reviews for a particular restaurant at one glance. As the size of the text grows larger, the word is mentioned more frequent. For instance, below is the word cloud for Chipotle, and words like *mexican, taco and burrito* recur quite often.



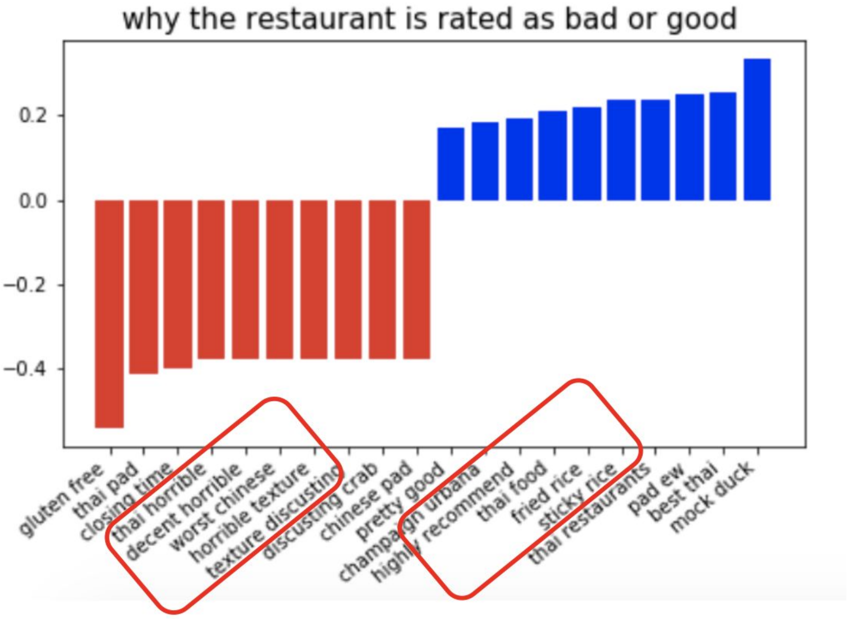
After we generated top keywords that affect review sentiment using support vector machine, we developed functions that could repeat the process for any given merchant and generated visualizations of the result.

Here we would like to illustrate words that are more likely to appear in good reviews and bad reviews, which provide strategic value to business owners that enables them to understand customers' demand and preference.

The bar chart is created using weights obtained from svm.coef_ and important features based on the absolute size of coefficients. Blue bars correspond to important words in good reviews, and red bars correspond to important words in bad reviews. For instance, using one gram, we can see words "fresh" and "fast" associated within good review.

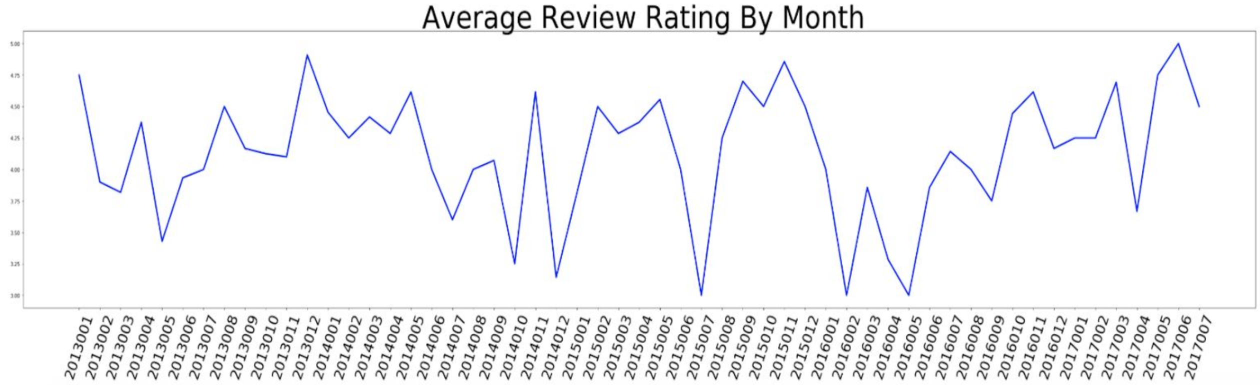


In order to present more informative results, we also performed restaurant important features extraction for bigram.



For this particular restaurant, we can visualize words like “thai horrible”, “horrible texture”, and “disgusting crab” associated within bad reviews. So we would suggest this restaurant to consider improving its food quality. Within good reviews, we can visualize phases such as “customers appreciate”, “like sticky salad”, “sticky rice”, and “duck” are items that people like most about this restaurant.

- In addition to the top keywords, we presented the average review score over time in a monthly scope for one particular restaurant.



The graph is able to present merchants a timeline of their business performance in any given month of the year. Combined with the top review keywords, merchants could gauge into the areas they are doing well and improve their weakness. This also enables merchants to notice shift in customers' attitudes in response to changes in business. Suppose they have a modification on the menu or service, they can monitor the changes in reviews to reflect its impact in business.

API & User Interface:

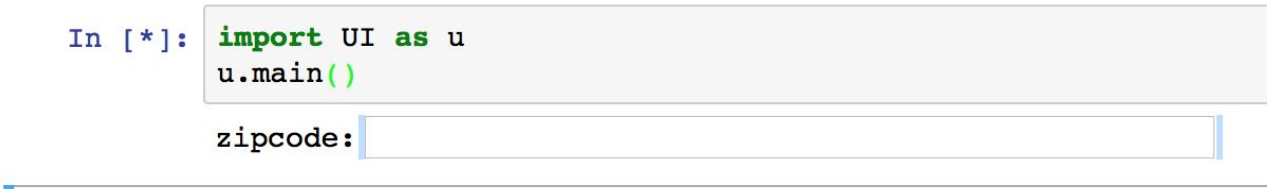
API: We first created a jupyter notebook (load and clean) and a clean_income_zipcode_data.py file for cleaning all raw datasets we obtained. Then we established a SQL database by populating the cleaned data into the database structure. Illustrating preliminary analysis on the data in another jupyter notebook (“visualization for review business”) and within the database jupyter notebook with queries, we verified that the database is accurate and up to date. Afterwards, we coded in two jupyter notebook, one for fake review detection model, the other is for sentiment analysis model (word cloud & feature extraction). Last but not the least, we connected these parts and present a user interface in the jupyter notebook with code in the UI.py file.

Since our raw datasets is very large, it takes at least an hour to clean all the datasets and store them properly in the SQL database. It will also takes up for hours to train our fake reviews detection machine learning model utilizing the fake review label dataset. As a result, we decided not to directly connect these parts in the automation to improve our user experience. Instead, we first cleaned data and created the database. Then we populated the cleaned data into the database and saved it to local files. Afterwards, we trained our fake review model and tested it on all our reviews. As a result, we outputed the test label outcomes with all the reviews as a csv file. That's all the preparation steps before we jumped into our UI design.

Taking in an user's inputs, we first query our database to give us the unique business id that is associated with restaurant's name and zip code. Loading the csv file with the predicted "deceptive vs. true" labels on the reviews, we only extract those reviews that are related to the unique business id. So we calculated the first output - fake review ratio - by counting all reviews that has a "deceptive" label and dividing it by the total number of this specific restaurant's reviews. As a final step, we connected the filtered reviews that takes out fake reviews with sentiment analysis machine learning model to fit data and output the word graph.

Interactive Application: The user interface takes in a restaurant's name and its postal code as the inputs, and displays a fake review ratio and a graph demonstrating the top 10 keywords that most positively or negatively influencing the reviews of the restaurant. Below is an example that illustrates the process and results:

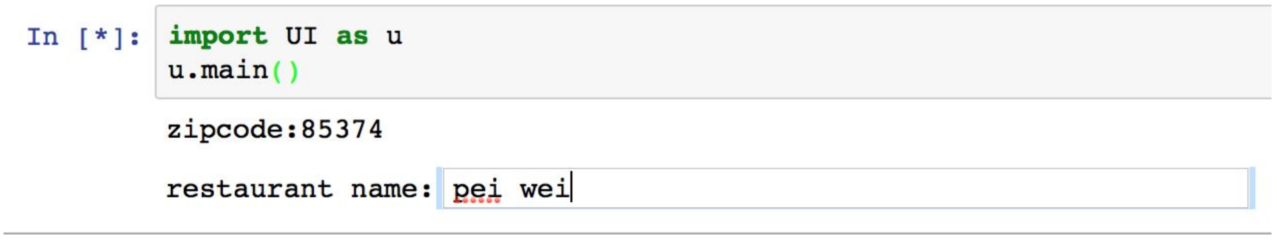
1. Open UI jupyter notebook



```
In [*]: import UI as u
        u.main()

        zipcode:
```

2. Type in the name and the zipcode of a restaurant that you would like to search
   - Here we typed in 85374 as the zip code and "pei wei" as the restaurant's name:

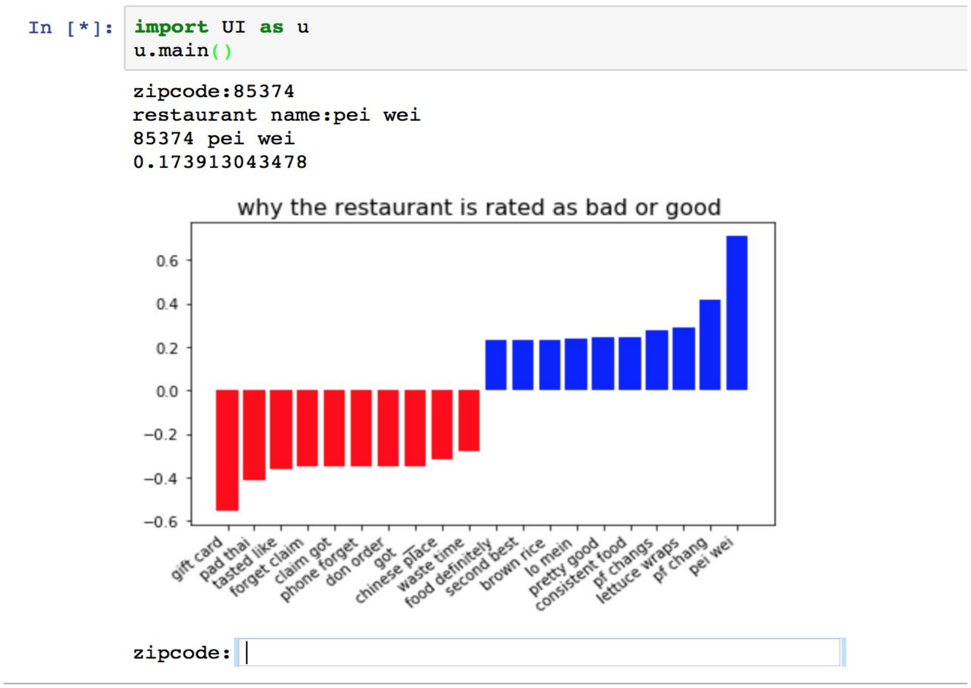


```
In [*]: import UI as u
        u.main()

        zipcode:85374

        restaurant name: pei wei
```

3. Waiting for the program to run for a couple of minutes, you will get the output:



```
In [*]: import UI as u
        u.main()
```

```
zipcode:85374
restaurant name:pei wei
85374 pei wei
0.173913043478
```

4. You can try another one when the output for the previous search is done.

Future Work:

1. User Interface - instead of using zip code, we will improve our user experience by making it more user friendly utilizing cities and states inputs with restaurant name.
2. CNN machine learning model - we currently have unbalanced data for fake reviews, and if we balance the data, CNN performed worse. For the next step, we will try to supply the CNN model with more logical data. In addition, because of the hardware limitations, we just added one Convolutional Neural Network layer and one Long Short-Term Memory layer. If we are able to try to add more neural network layers, our model could be more powerful. Moreover, deep Learning can be more efficient with large dataset. Our dataset is relatively small to fully evoke the power of the deep learning models. We are still exploring methods that are able to generate more training set such as semi-supervised learning, which is known by combining supervised learning and unsupervised learning models to produce more labeled text data.
3. Fake Review Detection - Intuitively, there will be certain patterns of fake reviews. For example, fake reviews may contain more positive words or tend to be long. If we spend more time on data exploration, we might be able to find such kind of patterns. If we extract these patterns and give them more weights, the model will be more efficient for training.

4. Sentiment analysis - we only worked with English, but the reviews include Japanese, Chinese, and other languages. It would be great if we are able to find ways to process these reviews as well. Potential ideas are translation into English or using distinct grammar rules to process the data. Moreover, we would like to improve our model by exploring more efficient feature engineering. In our current model, we only tried bigrams to feed it because of the computational limitations. By feeding larger grams into our model, the results may be more accurate.
5. Efficiency of the model needs to be optimized in order to handle even larger yelp review data in the future. For example, MapReduce implementation on Spark should reduce our training time dramatically. Beyond that, GPU processor should help improve our model efficiency significantly. After transforming to text data, the matrix representation of our data is pretty similar with the image data. GPU is known by efficiently processing of such matrix data. Another great option is Amazon Web Service (AWS), which provides cloud machines with high processing power, many RAMs and modern GPUs. In the future steps, we could apply this service to our project, which may lead to a more effective result.
6. Database - Exploring other database options such as MongoDB is possible. MongoDB is also a popular open-source database using dynamic schemas, which implies that we do not have to define the structure of the data before we store them into the database. One of the advantages of utilizing MongoDB is that we are able to change the records by simply deleting one column or adding one without re-defining a new schema. Since we currently is working with large datasets, MongoDB may be easier and more efficient for us to extract, store, and update our data.

Conclusion:

In this project, our goal is to help consumers and existing and new merchants to use Yelp more efficiently. From consumers' perspective, we enable them to realize the true restaurants' ratings by giving them the ratio of fake reviews. In the future works, we can also present the actual restaurants' ratings as one of the outputs as well. They also can have a better overview about the restaurants on features that the restaurants are famous for or needs to improve on by looking at the keywords, which helps them to choose places that they would mostly like to dine in.

From merchants' point of view, by filtering potencial fake reviews, they are able to find top keywords that indicate the advantages and weaknesses of their restaurants accurately. Combining pricing information and people's preference at different regions, business owners can modifies their corporate strategies on pricing, advertising, and dining food and services. In short, our complementary plug-in sample software can help Yelp retain more consumers and merchants through optimizing their searching and matching experience.

Reference:

1. https://www.yelp.com/dataset/challenge
2. https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_16_1YR_S1902&prodType=table
3. https://www.gaslampmedia.com/download-zip-code-latitude-longitude-city-state-county-csv/
4. http://www.ics.uci.edu/~vpsaini/
5. https://medium.com/tensorist/classifying-yelp-reviews-using-nltk-and-scikit-learn-c58e71e962d9
6. http://www3.cs.stonybrook.edu/~leman/pubs/15-kdd-collectiveopinionspam.pdf
7. https://plot.ly/python/scatter-plots-on-maps/