Northeastern University

Project Report

Individual Project
Yuchen Guo

Abstract

The main purpose of this report is to provide suggestion for US restaurant business owner, based on data mining technology. Data pre-processing is implemented to raw data first. Data exploration and visualization are done with exampled city, to give suggestion about restaurant location, favor and operation. K-means and GMM algorithm are used to cluster the restaurant favors. Word-cloud will help owner understand the customer's focus from their reviewing. To exclude fake reviewing rating, reviewing regrading is implemented with prediction model: SVM, Random Forest, logistic regression and Artificial Neural Networks.

Introduction & Background

Every business wants success. In this era of big data, 'data' can provide more information than consultant.

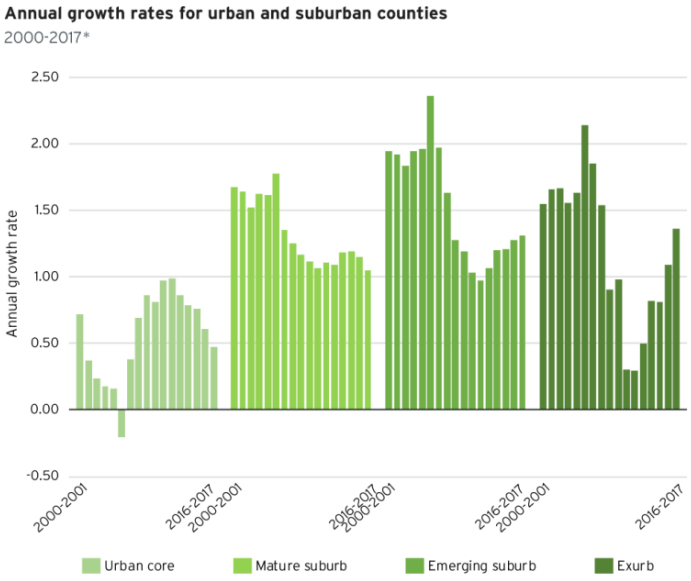
Businesses are always customer-oriented, especially for restaurants. As a typical retails, it is important for restaurant to look at their consumer behavior. For example, people with special favors may live in a compact communities. It may be a disaster that a Mexican Restaurant open in a town no one like Mexican.

Also sometimes restaurants would like to imitate the winner and get suggestion from their customers' review. But what if the review is fake?

Fake review is not illusory or small issue. Even yelp, who does a strict supervision, has a problem of fake review. In Sept. 25, 2017, Yelp estimated that as many as 25 percent of the reviews on Yelp may be fake, when it is interviewed by NBC4.

Besides fake review, more and more information is coming up. Without data mining, information will only disturb the decision of the restaurant owner. For example, suburbanization. By 2010,

85.3% American choose to live in suburbs and exurbs area[1]. And the growth rate[2] for suburban is increasing. It seems most of the business owner should open restaurant in rural area, because restaurant is customer-oriented. But actually from the following data analysis, the answer is no. Downtown is still the best place.



Methodology & Result & Discussion

Here I combined Method, Results and Discussion together for better explanation. Code part is ignored because all the code file has been attached.

I am trying to provide suggestion for restaurant using data mining. For example, performance of prediction model is compared, so the best one can be used to regrade the rating. By this way, it can decrease the influence of non-integirity of data. For example, some business may hire people to give high stars and low star to their competitors.

First, Data pre-processing is applied on the raw data. Then Data exploration, visualization and word-clouds will provide visual business suggestion. Two clustering algorithm, K-Means and GMM, are used for clustering. Prediction methods handle with re-grading the restaurant and predicting the review rating.

---

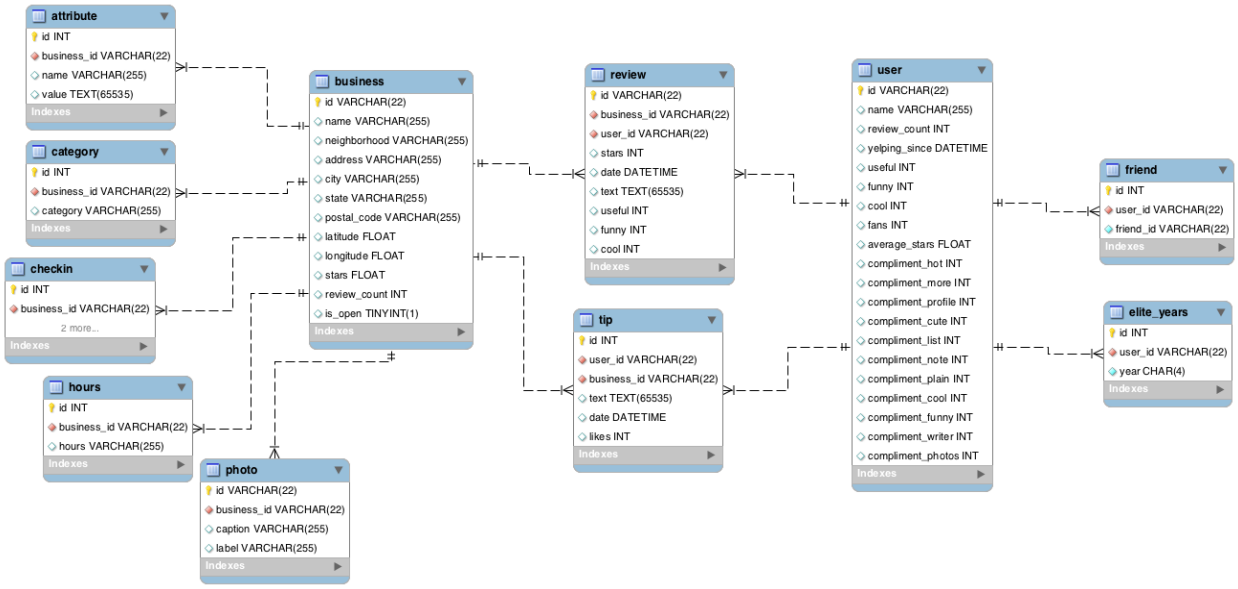
[1] The share of residents living in the suburbs and exurbs increased to 85.3 percent in 2010. https://opportunityurbanism.org/2017/12/suburbs-exurbs-grab-nearly-metropolitan-growth/

2Source: William H Frey analysis of Us Census Population Estimates, released March 22, 2018.

1. Data

1.1 Dataset introduce

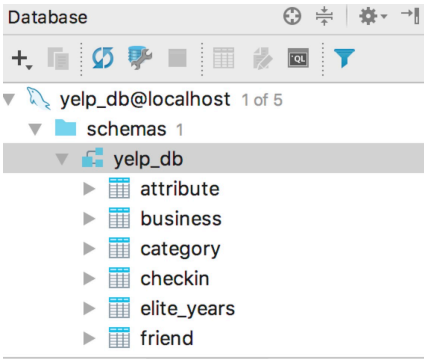
Here this report uses the official dataset from yelp[3]. The schema of this dataset is shown below. In this project, I mainly use table business, category, user and review. The attributes of each table can be found below. Every table is connected through foreign key.



[3] https://www.yelp.com/dataset/.

1.2 Database

MySQL database is selected to build a local database and then I import the yelp.sql file which can also be found on yelp public dataset[4][5]. All the data information in yelp.sql in now on MySQL database.



In [280]:

```
################################
## connect to database
## make sure the MySQL database is running
## please change the user and password to yours
## please establish new database yelp_db for the first time
################################

connection = mysql.connector.connect(
    user='root',
    password=[redacted],
    host='127.0.0.1',
    port=3306,
    database='yelp_db')
```

Pandas DataFrames, list and dictionary are the main data structure. Most of the query, filter and process will be implemented with them.

1.3 General overview

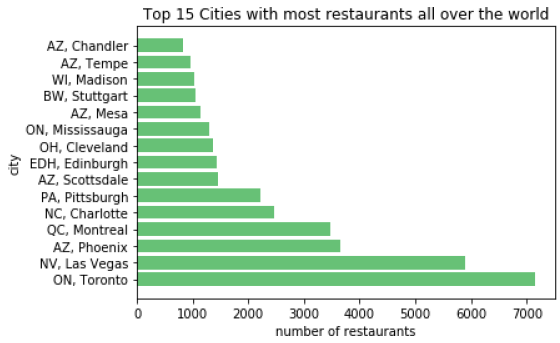
With sql queries, first of all it counts the numbers of restaurants for each states and find the city with most of the restaurant. Too much data will obscure the analysis and it is better to choose a city as sample.

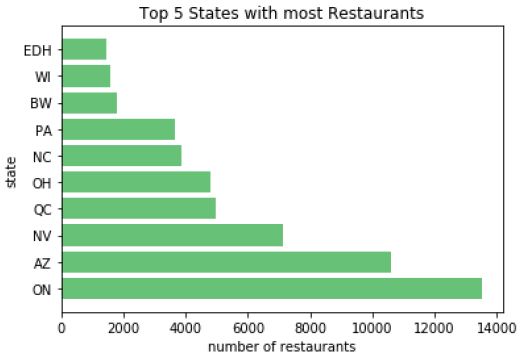
With 776 cities in record, the US city with most of the restaurant is Las Vegas. Top 1 is Toronto, a Canada city, it is inefficient to provide guidance for US restaurant owner. On second bar graph, NV also lists top 3rd, which means a flourishing restaurant industry here, which guarantee enough data to analyze.

[4] https://yelp-dataset.s3.amazonaws.com/YDC11/yelp_sql.tar.gz?Signature=eIu2Ghb%2BqNo1tU5KVUvPO3uI4ss%3D&Expires=1531785327&AWSAccessKeyId=AKIAJ3CYHOIAD6T2PGKA.

[5] note: when I first draft this report and download the sql.file, it is still available. But recently yelp only provide the json version file. I will try to upload compressed package of sql.file in my GitHub and attach the link if needed

total cities: 776





1.4 Data Pre-Processing

Data pre-processing goes through all the project. Handle with the data first, then model and analyze. For the following parts, simple example will be given for each part.

Transform raw data

The dataset file type from yelp dataset is sql. There are two ways to transform raw sql data to the data we can use.

The first way is including sql query language in i-python notebook, such as the picture below:

```
pd.read_sql("SELECT business.city, business.state, COUNT(*) "
            "AS 'Number of Restaurants for each City' "
            "FROM business INNER JOIN category on "
            "business.id = category.business_id "
            "WHERE category.category = 'Restaurants' "
            "GROUP BY business.state, business.city "
            "ORDER BY COUNT(*) DESC;", connection).values
```

Second way is transform MySQL database table to pandas data-frame as below. The any operation can be done on the data-frame.

```
df_review = pd.read_sql('SELECT * from review', connection)
df_user = pd.read_sql('SELECT * from user', connection)
df_business = pd.read_sql('SELECT * from business', connection)
```

B. Data cleaning

Sometime unexpected data appear and prohibit the analysis, for example, missing data.

When counting the number of restaurants in Las Vegas, there are some cuisine do not have any restaurants. The original value is None, which sorted() function cannot handle with. An data clean function was added to fill the None value with scalar value 0.

```
def dict_clean(dic):
    result = {}
    for key, value in dic.items():
        if value is None:
            value = 0
        result[key] = value
    return result
```

And also, the weird data is detected before further analysis. For example, before doing K-means clustering, all the neighborhoods is listed. Neighborhood with ‘’ is removed.

```
'Eastside',
'',
'Downtown',
'Sunrise',
'Northwest',
'Chinatown',
'Southwest',
'Summerlin',
'University',
'The Lakes',
'South Summerlin']
```

```
In [14]: cluster_table = cluster_table
         [cluster_table.neighborhood!='']
         neighborhoods = cluster_table.neighborhood.unique()
         list(neighborhoods)
```

C. Text pre-processing & Natural Language Toolkit

a. Tokenization

Reviewing is text, string type, and it is hard to be handled as a whole. In this project, Penn Treebank Tokenizer is used, which assumes that the text has already be segmented to sentence. This properties meet the attribute of reviewing. Even though customer write the review with full freedom, 99.99% of reviewing is segmented by sentence.

b. Stemming & Lemmatization

Inflectional forms and derivationally related forms of a word also affect the analysis badly.

Stemming is commonly used toolkit for Natural Language Processing (NLP) systems and Information Retrieval (IR) systems. It reduces the words to their word roots by removing suffixes and prefixes.

Lemmatization has different method and similarity function, which reduce the word to a basic form.

d. Filtering

When analyzing text, some words have high frequency but low importance, such as “to”. Stop Words Filtering is implemented through my input and stop-words provided.

D. Data Reduction

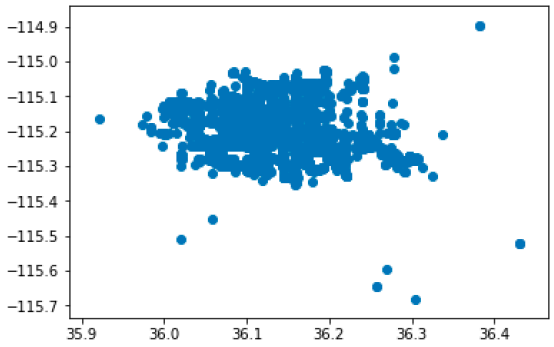
When dealing with reviews, with 47355021 views in database, data reduction is truly need. Too much data means time-consuming.

Here Systematic Data Sampling is used and detail method is Equal-probability method. Because the reviews do not have a regularity in it, it is safe to implement this easy and quick method. Here k is choose to 200 because $k = \frac{N}{n}$ according to the number of sample I want.

2. Data exploration and visualization

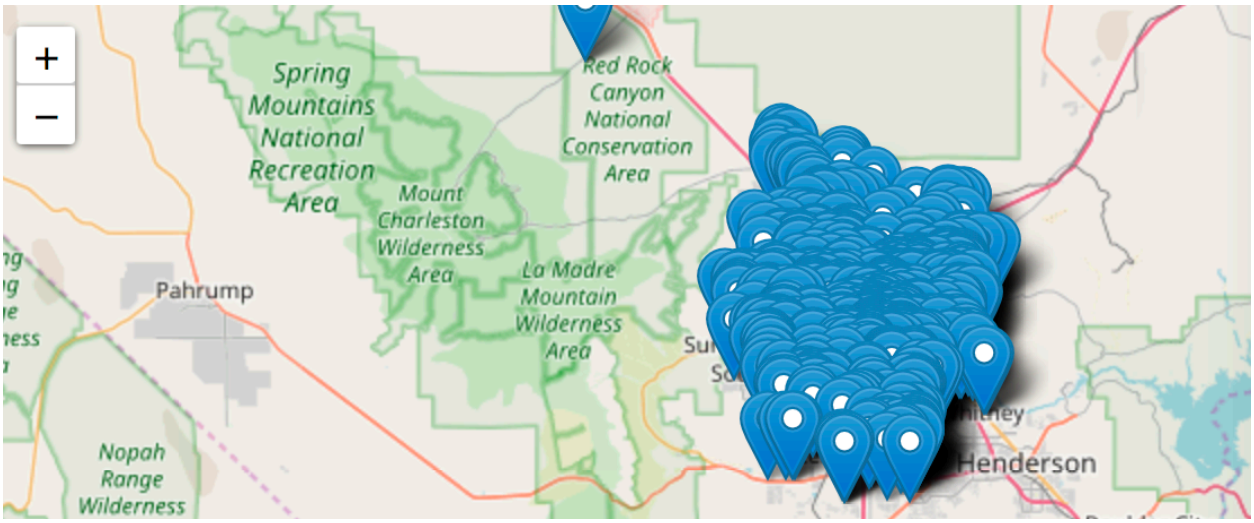
A. Location

Nowadays, more and more citizens are apt to live in suburb. Restaurant needs customers and for this kind of business, distance is important. It is easy to understand no one want to drive for 30 mins just for a single lunch. Does it mean the city center is no longer the best place to start a restaurant?

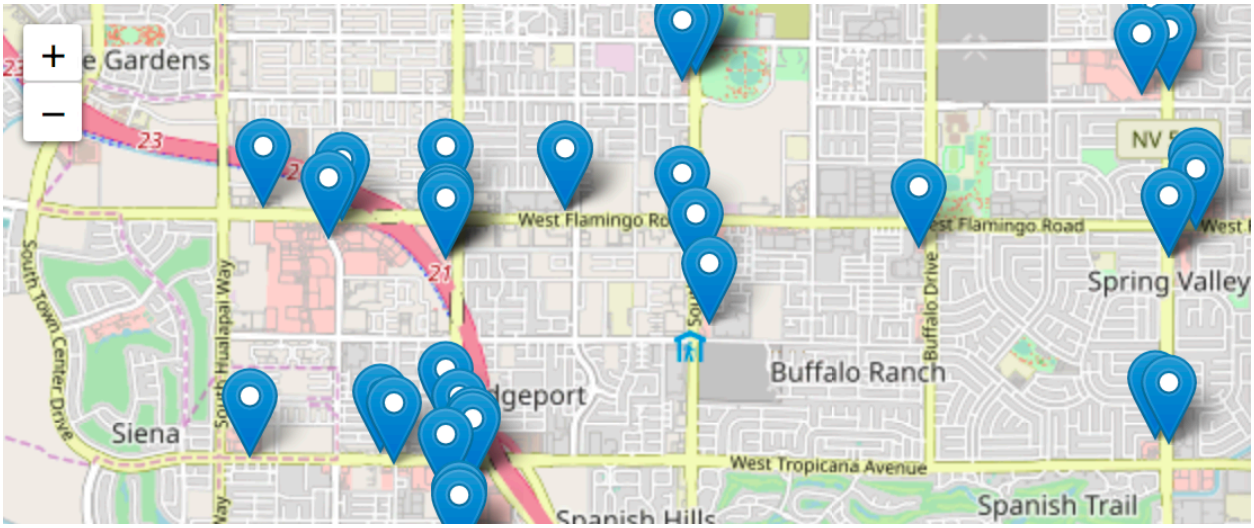


First I tried to scatter the distribution of restaurant as below. Even though longitude and latitude are used for axis, I still find it is really hard to read the location.

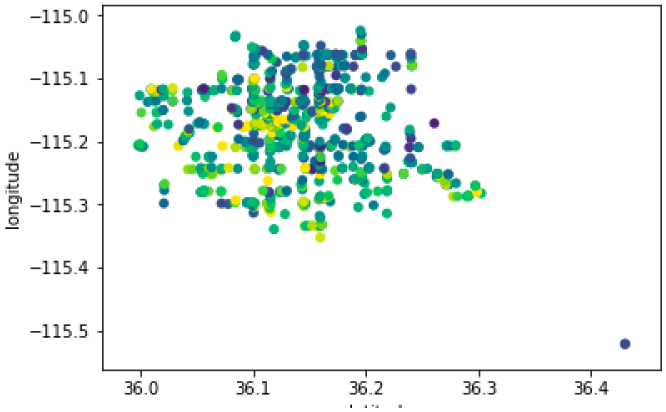
It will be better if a real-view map can be shown, just like the google.map. Here coms the folium library.



With the Zoom-in and Zoom-out button on the left, actually we can see on the level of street and neighborhood level.



Although there is a tendency for people to move away from the city center, most of the restaurant is still located on the center. Here folium library is used to plot the location of all the restaurant in Pittsburgh. Latitude and longitude are used to figure out where they are.

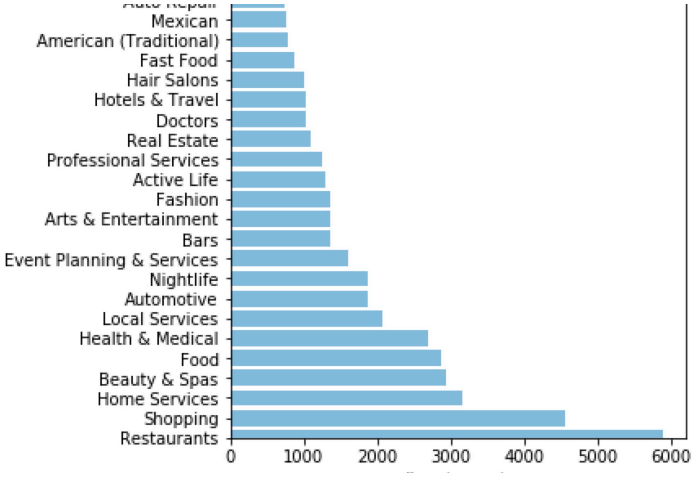


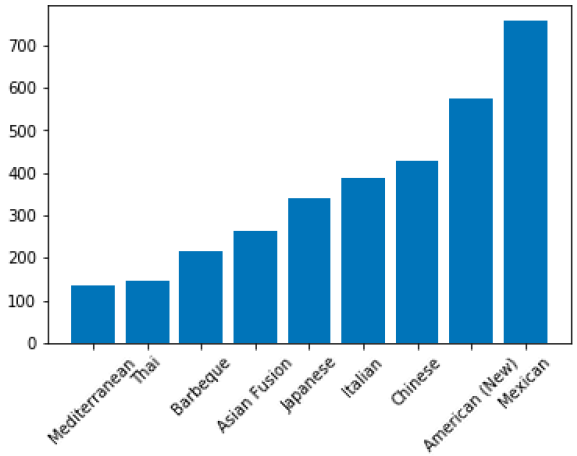
After we get the location of all the restaurant, the city center still has the most density. Is it better the avoid competition or join in for a constellation effect? Other figure may give the consultation. The following scatters show the location of the most popular restaurants.

The most popular restaurant highly overlap the number of restaurant above. Comparing running alone in a lonely place, restaurant groups seem give more benefits.

B. The type of cuisine

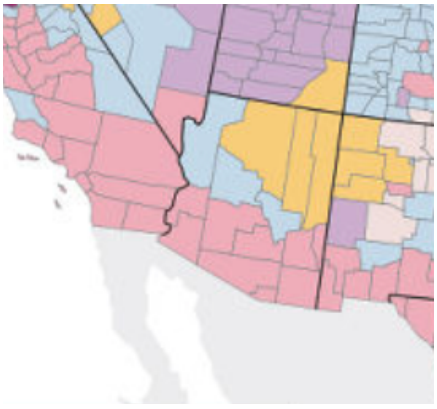
Next, the restaurant may want to know what kind of food is more welcome. The number of review for each cuisine is explored. One problem appear here is that, the restaurants have two category “restaurant” and its cuisine. As the result show below, “restaurant” is the most cuisine, which means nothing to us. To make it specification, a deep exploration is executed and finally I get the detail cuisine for each restaurant.





The total number of different cuisines in figure2 is 5902 statistically, which meet the number of restaurants in figure1.

From cuisine exploration, American(New) and Mexican dominate the restaurant of Las Vegas. American(New) is as expected, but why Mexican? Las Vegas is not the city close to Mexico. As mentioned in Lecture, the meaning of data mining is to find the thing unexpected and I would like to explore the reason behind it.



While the race and ethnicity distribution map can tell the reason: Where pink represent the distribution of Mexican, Las Vegas in great pink means majority of population here is Mexican.

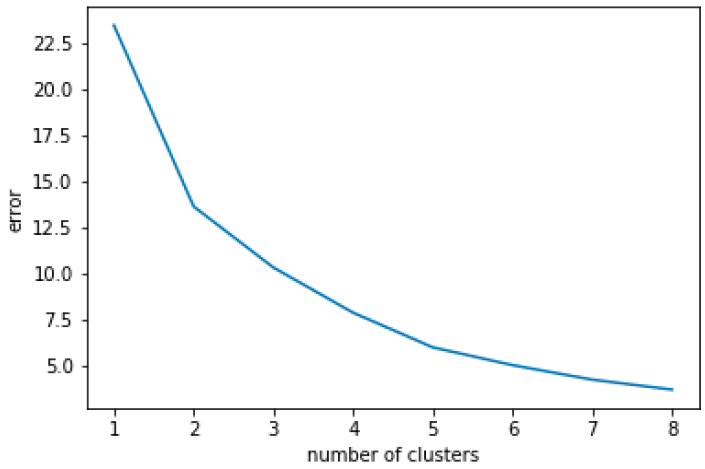
3. Clustering

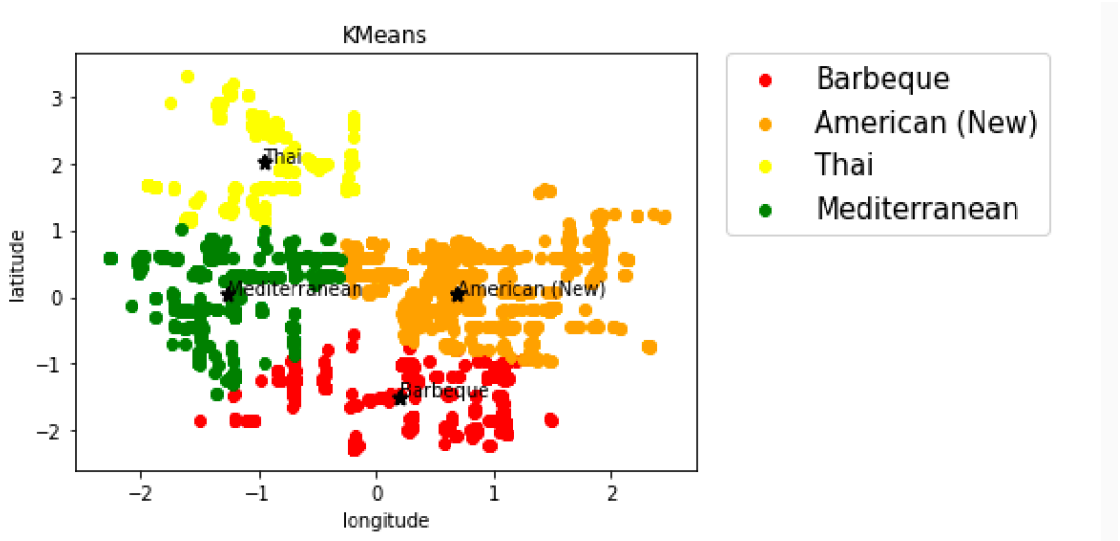
But what if the restaurant has already decide on their cuisine? Then they may want to find a neighborhood with most of the customers who has this kind of tones.
Clustering tech “K-Mean” and “GMM” are used for this and business like “cafe” which cannot be figure out cuisine has been dropped.

A. K-Means

When doing with K-means, the method of initialization is ‘K-means++’, which can speed up the convergence because of a gorgeous way to choose the cluster center.

The first task is to choosing the number of clusters. Sum of squared distances of samples to closest center is used to calculate the error below. The figure below shows a elbow between 2 and 5. 4 clusters is the final choice because the number of cluster must be the integer and more cluster is better for business suggestion.





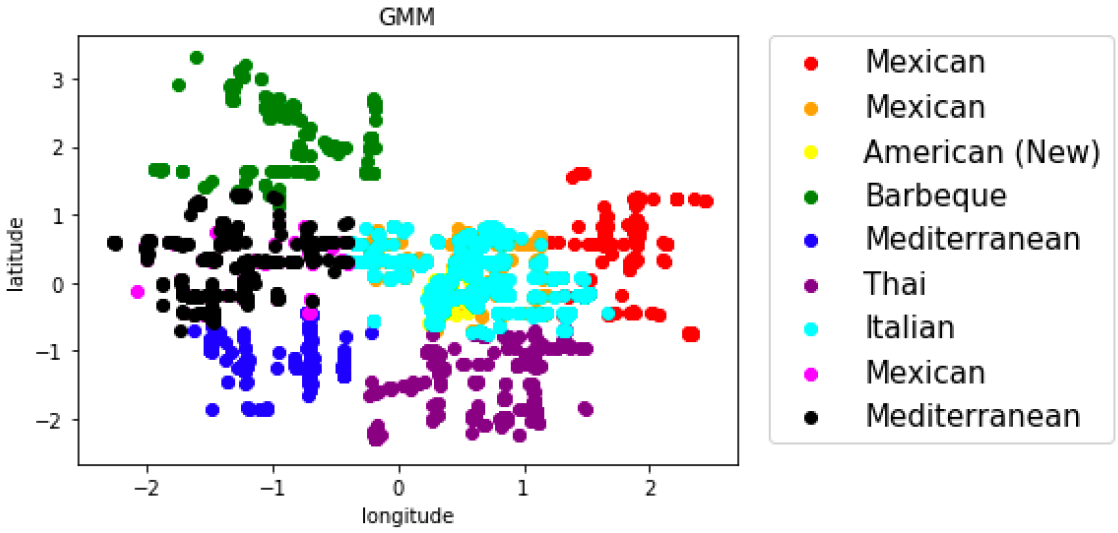
Another problem is encountered during clustering is that, some kind of cuisine enjoy its popularity in this city. It is likely to get only one cuisine. As we analysis before, American and Mexican are domesticated this city. To deal with it, ratio of each cuisine is used instead. (total number of cuisine and the number of cuisine in this area)

Which is a big surprise is that, Mexican does not show on this picture. With most of the restaurant on Las Vegas, Mexican even cannot dominate one cluster above.

The four label for the clustering is that: Thai in the north-west, Mediterranean in the west, Barbeque in the south and American (new) in the easter. The silhouette score is 0.27 and calinski-harabaz score is 844.

The business owner then can choose the clustering location if his cuisine is in these four labels. But what if his cuisine is not on the above. A better clustering picture to show is GMM.

B. GMM



GMM has more cuisine on its map and also the over-lapping. Because it tried to consider points to each cluster, which cause the over-lapping. But on the overview, GMM still shares the similar patter with K-means.

4. Word cloud

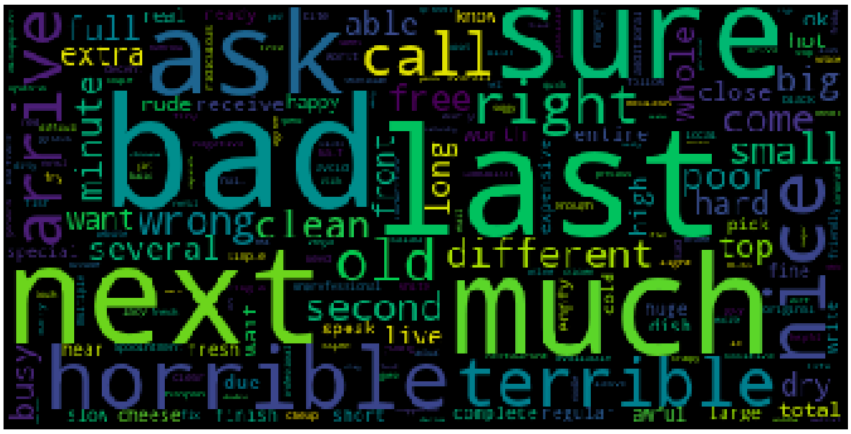
Before word-cloud is implemented but after token, a part-of-speed tagger was implied to this sequence of words and attaches the speech tag to them. Penn Treebank is also used for this. The word with tag “different” is what I interested in.

Review with a rating star 4 or 5 is regarded as good(positive) review, while the review with star 1 or 2 is negative review.

To remove the meaningless words, such as “how”, “to” and duplicate word, three methods are used. First is stop-words in nltk library, which removes the commonly useless words in English. Second is stem and Lem; Last is physical input array. After generating the word-cloud first, meaningless word is collected and put into the array.

The first graph is for positive rating and “delicious” appear most. As a restaurant, the main issue is still food. Some other words like “friendly” will provide some advises. It makes sense customer do not want restaurant with rude service.

Also the negative words show the aspect should be avoided.



5. Reviewing re-grade and prediction

For the fake reviewing question we mentioned on Introduction, machine learning prediction models are implemented to regrade the rating. By comparing the performance of each model, the best one can be used to generate the rating of review supposed to be.

Before any further modeling is implemented, the data weight should be considered into the calculation first. If each feature is multiply by the weight of their frequency, like $log(\frac{M}{a_i})$[6], and SVM is implemented, the result generated is useless.

TF-IDF is used here for scaling and normalizing, which use a weighing factor to weight term frequency against the frequency of the document.

With 47355021 views in database, the training size is set to 75%.

6 where M is the total number of reviewing and ai is the number of reviewing with the word

A. SVM

a. Model choose:

SVM has many different kernels, Linear, Polynomial, Gaussian and Sigmoid. In oder to train the model to “learn”, SVM is a good choice. Choice between each kernel is interesting to discuss.

RBF has a strong problem of overfitting. Considering the flexible on reviewing, it is discarded. Polynomial one consider the combination of feature and interactions. It increases the running time and does not work very well on the high dimensional data.

Each word can be regards as a feature. As a result, the number of feature is really large compared to the trining sample. In this aspect of high dimensions, Linear and sigmoid are good choices. The project tries both of the sigmoid and linear, then choose a better one to analyze.

And Linear model usually run faster than non-linear SVM model, considering logistical regression run almost 2 hours in my computer. Linear SVM is final choose.

b. Model Result

0.7468721683788222

	precision	recall	f1-score	support
1	0.79	0.84	0.82	146570
2	0.66	0.48	0.55	87409
4	0.65	0.58	0.61	244660
5	0.79	0.86	0.82	450599
avg / total	0.74	0.75	0.74	929238

The accuracy is 0.74, which is the average around all the model I tried. The time consuming is 15 mins, which is later than Perceptron but clearly fast than other models.

c. Model Analysis

```
[[123631 15733  3282  3924]
 [ 27987 41550 13616  4256]
 [  2614  4967 142875 94204]
 [  2140   943 61550 385966]]
```

The confusion matrix show good prediction on bottom and top ratings. Because almost all of the model performs good in this aspect, we can assume it is a easy task, which cannot be used to judge the performance between different model.

The prediction on neutral rating is similar to perceptron but a little better than it. The different between true prediction and wrong neutral-rating prediction around true is lower.

B. Regression

Reviewing prediction is typically categories classification, which is also an example of pattern recognition. There are two models I want to try - Random Forest and Logistic Regression.

a. Model choose

Random Forest is effective, which combines decisions from a sequence of base models (model ensemble). This arises my interest because in common sense, the more suggestion generated more reliable results.
In random forests, the base models are constructed with different subsample of the data.
Logistics regression is a statistical classification model. Using Sigmoid function, when classifier is performing, Logistics regression uses probabilities instead of classifier directly, which is better. That is also the reason why I include Logistics regression here.

b. Model Result

Random Forest perform worse unexpectedly. Its accuracy is only 48% and none of the three score in classification_report exceed 0.5. Unbalanced data (distribution of the rating) and hardness of reviewing prediction contribute to this result.

```
0:09:18.840304
0.48491236905937984
             precision    recall  f1-score   support

          1       0.00      0.00      0.00    146570
          2       0.00      0.00      0.00     87409
          4       0.00      0.00      0.00    244660
          5       0.48      1.00      0.65    450599

avg / total       0.24      0.48      0.32    929238
```

```
1:30:56.094526
0.7599517023625809
             precision    recall  f1-score   support

          1       0.80      0.85      0.82    146570
          2       0.70      0.46      0.56     87409
          4       0.67      0.59      0.63    244660
          5       0.79      0.88      0.83    450599

avg / total       0.75      0.76      0.75    929238
```

On the contrast, logistic regression generated best result with accuracy 75%. The following analysis will focus on logistic regression.

c. Model Analysis

```
[[125056  13435   3258   4821]
 [ 27392  40557  14582   4878]
 [  2503   3339 145461  93357]
 [  2175    522  52800 395102]]
```

Logistic Regression is truly time consuming and it runs half and a hours on my computer. The slight improvement is not worthy.

It has the similar pattern with Perceptron: does well on extremely different rating. Compared with Perceptron, it does well on netual analysis and rarely confuse on it, such as star 2 and star 3.

C. Artificial Neural Networks - Perceptron

a. Model choose

Artificial Neural Networks is last but important model in prediction. Here I choose the most basic type - Perceptron, which was outlined by Rosenblatt in 1957. All the inputs x are multiplied with their weights w.

Another reason is that, first, Perceptron can be easily trained with huge data; Second, theoretically Perceptron guarantee the maximum number of mistakes won't be too big.

b. Prediction Result

Perceptron only takes 2 mins to finish the prediction, even though millions of reviewing data was pull in. The accuracy is 71% and average precision, recall and f1-score are 0.71.

```
0:02:25.876966
0.712487005481911
             precision    recall  f1-score   support

          1       0.77      0.82      0.79    146570
          2       0.59      0.43      0.50     87409
          4       0.58      0.56      0.57    244660
          5       0.78      0.81      0.79    450599

avg / total       0.71      0.71      0.71    929238
```

c. Analysis

Compared with other model, Perceptron doesn't perform very well. But considering the little different on the performance score and the time it saves, Perceptron is very great prediction with huge dataset.

```
[[120180  16782   4870   4738]
 [ 29518  37783  15097   5011]
 [  3860   7090 137277  96433]
 [  3317   2159  78293 366830]]
```

With the help of confusion matrix, it is clear Perceptron performs good on negative and positive reviewing. The number on star 1 and star 5 is obviously bigger than others. It means Perceptron makes good prediction when the sentiment contained in reviewing is extreme.

Also, Perceptron gets confused between the adjacent rating, for example, 2 and 1, 5 and 4. The number sharply goes down on left-bottom corner and right-top corner. Perceptron may makes wrong prediction, but it usually isn't too far off.

Northeastern University

Reference

1. William H Frey, Us Census Population Estimates, Brookings: https://www.brookings.edu/blog/the-avenue/2018/03/26/us-population-disperses-to-suburbs-exurbs-rural-areas-and-middle-of-the-country-metros/
2. NBC4 I-Team: https://www.nbclosangeles.com/news/local/Fake-Reviews-on-Yelp-Facebook-Google-447796103.html
3. Yelp DataSet: https://www.yelp.com/dataset/challenge
4. The Fundamentals of Neural Networks: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53
5. CS6220: https://www.northeastern.edu/mscs_online/cs6220-37506-spring-2018/