Capstone Project

What Makes A Great Restaurant?

--Yelp Data Exploration

Prepared for Dr. Amir Jafari

Wei Hao Fei Wang

12/03/2018

Introduction

Nowadays technology makes people's access to information faster and easier. Two decades ago we needed to go out wandering around on the street to find a good restaurant to have dinner, or when we arrived at an unfamiliar place without knowing anyone, we needed to talk to people in the coffee shop to inquiry about surrounding good places, and how to go there. When Internet was worldwide popular several years later, we went online to search for well-known places or businesses around us, but we were still in the dilemma of making choices.

Nevertheless, things change dramatically in current era with smartphones. Yelp is one of the mobile apps on smartphones that makes our life convenient. It is a local-search service that publishes reviews and rates local businesses and services, which facilitate the process of people making decisions regarding their dining, shopping, and choosing services choices just by simply tabbing on the Yelp app on our phone, then it would provide us with the most related information based on our preferences. With more and more people are introduced to this app, it also provides Yelp a huge amount of market information, which could be used for local businesses to improve their performance based on ratings and reviews and local government to identify the most thriving business type in town. It is highly likely that this is one of the reasons that Yelp host this big data challenge for the public to analyze from different perspectives.

Dataset Description and Capstone Objectives

Yelp dataset contains its businesses, reviews and user data in Json file format and involves 5,996,996 reviews, 188,593 businesses, 1,185,348 tips by 1,518,169 users, and 1.4 million business attributes like hours, parking, availability, and ambience in 10 metropolitan areas of US.

Business file		Review file	
Attribute	**Meaning**	**Attribute**	**Meaning**
business_id	Business ID (index for datasets merging)	busindess_id	Business ID(index for datasets merging)
neighborhood	The neighborhood of the restaurant	date	Date of this review
is_open	Weather the restaurant is open	stars	Yelper rating levels
categories	Classify by cuisine, abstention, etc.	text	Review text
review_count	Number of reviews	cool	"cool" counts this review received
name	Name of the restaurant	funny	"funny" counts this review received
stars	Business/Restaurant rating levels	useful	"useful" counts this review received
postal_code	Restaurant's location		

Hopefully, through this capstone project, we could explore interesting business insights and improve our skillset in the area of natural language processing, machine learning and data analysis, and provide productive business recommendations for the current and incoming ventures on Yelp, and help Yelp to improve its services to provide people with a more accurate information as well.

Proposed Research Questions

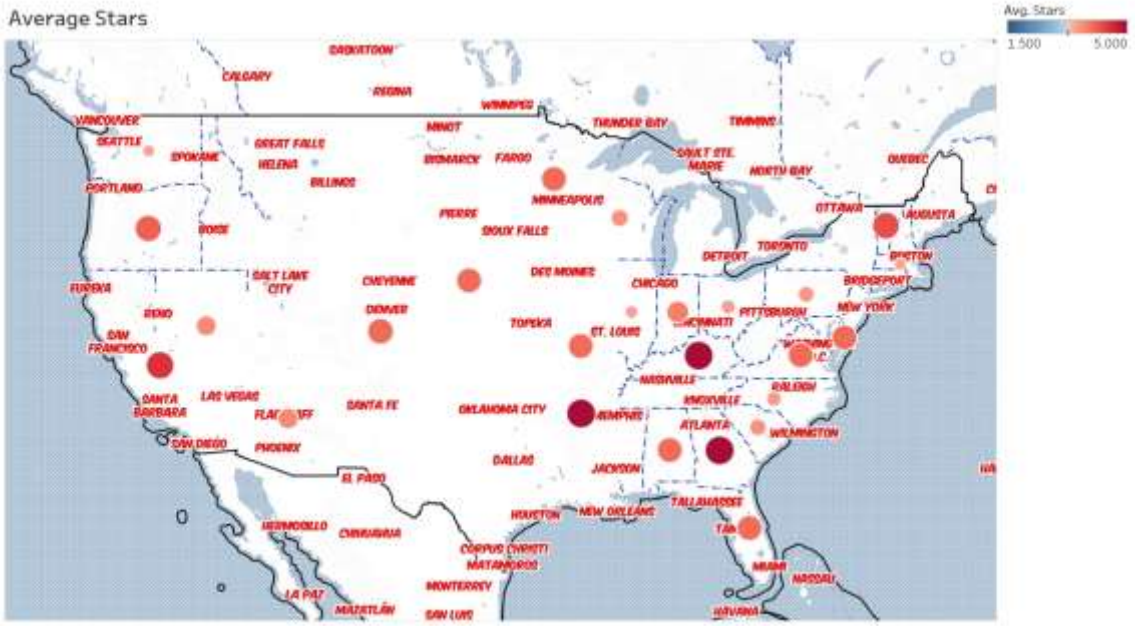
1. Identify top-rated geographic area for different business type.
2. Examine the most important attributes affect people's choice among identical restaurants.
3. Predict STAR ratings that consumer would score based on business attributes and their reviews.
4. Identify the common words that consumer tend to write in their reviews for certain businesses.

Exploratory Data Analysis

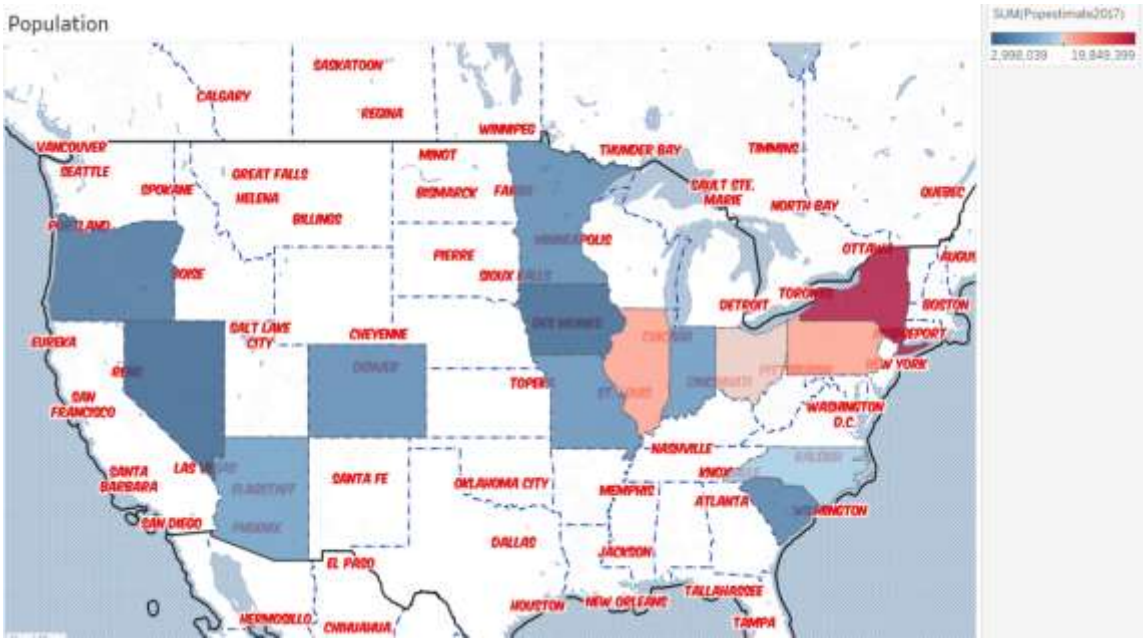
Below graph presents the major cities with their business amount stored in the Yelp dataset. Las Vegas comes on the top of the list with business amount 28,865, following by the Phoenix and Toronto. It is strange that some of big cities in the U.S like New York, Chicago, and Boston do not make into the top of the list. However, Canada has two cities bump into the top 10, and comparatively smaller cities like Pittsburgh, Phoenix also have a large amount of businesses using Yelp services.

Las Vegas	Phoenix	Toronto	Charlotte	Scottsdale	Calgary	Pittsburgh	Mesa	Montréal
28865	18633	18233	9204	8822	7384	6804	6239	6045
Henderson	Tempe	Chandler	Madison	Cleveland	Glendale	Gilbert	Mississauga	Peoria
4815	4492	4272	3509	3506	3469	3397	2954	1868
Markham	North Las Vegas	Champaign	Scarborough	North York	Surprise	Richmond Hill	Concord	Brampton
1699	1508	1243	1175	1140	1119	978	975	929
Vaughan	Goodyear	Etobicoke	Matthews	Oakville	Avondale	Fort Mill	Huntersville	Lakewood
853	827	760	726	699	663	620	608	509
Gastonia	Cornelius	Mentor	Cave Creek	Urbana	Monroeville	Westlake	Thornhill	Laval
494	467	442	402	400	390	379	376	373
North Olmsted	Strongsville	Whitby	Pineville	Middleton	Fountain Hills	Cuyahoga Falls	Aurora	Medina
373	362	357	352	347	341	340	336	327
Newmarket	Pickering	Boulder City	Indian Trail	Montreal	Kent	Ajax	Beachwood	Wexford
324	315	295	286	283	282	281	281	280
Parma	Monroe	Buckeye	Willoughby	Sun City	Rocky River	Sun Prairie	Woodbridge	Stow
274	273	265	248	247	246	244	226	221
Litchfield Park	Avon	Solon	Hudson	Cleveland Heights	Fitchburg	Bethel Park	Elyria	Verona
220	218	210	201	200	199	198	198	195
Airdrie	Chagrin Falls	Bridgeville	Brunswick	Coraopolis	Kannapolis	Canonsburg	Waxhaw	Brossard
191	186	181	178	175	173	171	167	166
Belmont	Fairlawn	East York	Verdun	Mayfield Heights	North Royalton	Tolleson	Independence	MESA
158	156	151	150	149	144	144	136	135
(Other)								
12572								

Built upon above information, we are interested at which area are being top rated regarding their overall business performance. Looking at below graph, the darker the red demonstrates presents a higher rated geographic area like California, Kentucky, Arizona, and Georgia. However, there are other factors that could influence certain average ratings aside from the among of businesses.



One of the factors that comes to the top of our head is each states' population. Take a look at the below graph, actually states with a higher average rating have a relative smaller population. On the other hand, areas like New York, Pennsylvania, and Illinois with a comparatively large population are having a comparatively lower average star rating.



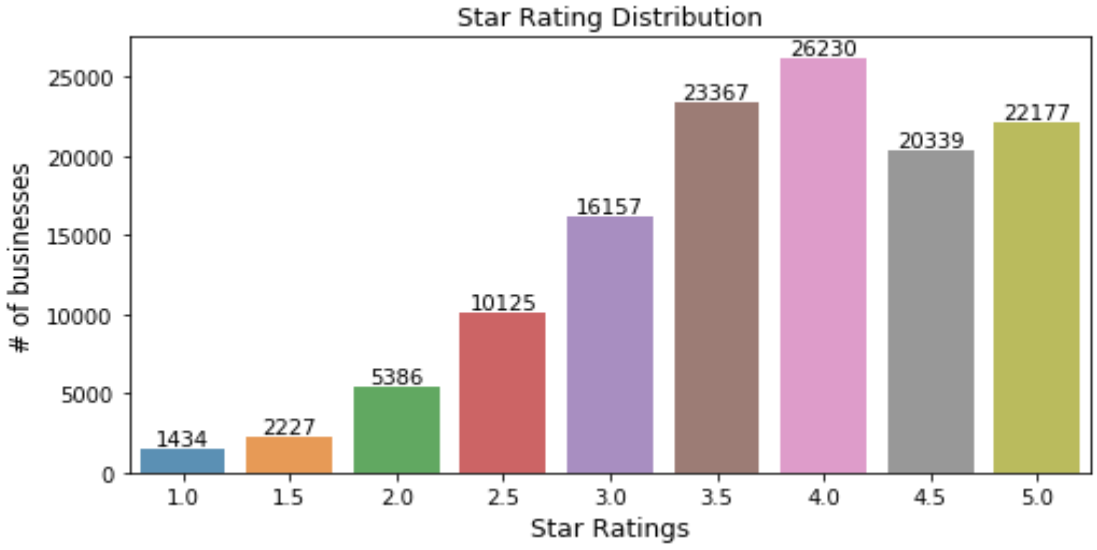
To dive deeper in the dataset, among all businesses, restaurants are the most common business. And among all restaurants' categories, Mexican, Chinese, Pizza restaurants occupy the largest share in major states on Yelp.

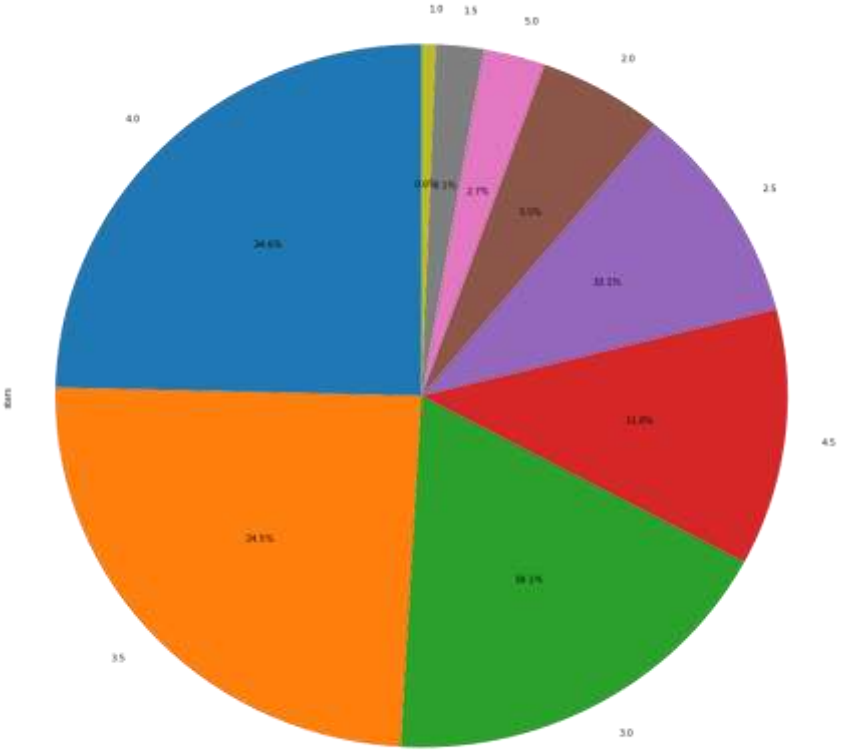
categories <chr>	n <int>
Restaurants, Pizza	1092
Pizza, Restaurants	1060
Coffee & Tea, Food	1036
Nail Salons, Beauty & Spas	1015
Beauty & Spas, Nail Salons	981
Food, Coffee & Tea	966
Restaurants, Mexican	932
Mexican, Restaurants	908
Beauty & Spas, Hair Salons	893
Restaurants, Chinese	889

state <chr>	categories <chr>	n <int>
AZ	Restaurants, Mexican	394
ON	Restaurants, Chinese	305
NV	Mexican, Restaurants	181
OH	Pizza, Restaurants	156
PA	Restaurants, Pizza	151
QC	Restaurants, Pizza	102
AB	Restaurants, Pizza	86
NC	Restaurants, Chinese	82
WI	Restaurants, Mexican	37
IL	Restaurants, Mexican	21

Then we look at the stars, based on the current data, we want to know the star distribution, which could give us a sense of consumers' tendency or preference of rating. People tend to give 3.5 to 4.0 to businesses, which is not extremely good or bad, but definitely those businesses received 3.5 or below are not providing satisfying services.

Therefore, as in later classification process, businesses fall into rating 4.0 to 5.0 are considered as good businesses, and businesses receive rating below 3.5 (including 3.5) are considered as less satisfying businesses.





Business Star Rating Predictive Modeling

1. Data Pre-processing

In the “yelp_academic_dataset_business.json” , it provides many features about the business's information, such as location, stars, business categories, business attributes. Because we’d like to predict the restaurants’ star by business attributes, then we deal with this dataset and get all the attributes of the restaurants.

Attributes we have :

```
Index(['AcceptsInsurance', 'AgesAllowed', 'Alcohol', 'Ambience casual',
       'Ambience classy', 'Ambience divey', 'Ambience hipster',
       'Ambience intimate', 'Ambience touristy', 'Ambience trendy',
       'Ambience upscale', 'Ambienceromantic', 'BYOB', 'BYOBCorkage',
       'BestNights', 'BikeParking', 'BusinessAcceptsBitcoin',
       'BusinessAcceptsCreditCards', 'BusinessParking lot',
       'BusinessParking street', 'BusinessParking valet',
       'BusinessParking validated', 'BusinessParkinggarage',
       'ByAppointmentOnly', 'Caters', 'CoatCheck', 'Corkage',
       'DietaryRestrictions', 'DogsAllowed', 'DriveThru', 'GoodForDancing',
       'GoodForKids', 'GoodForMeal breakfast', 'GoodForMeal brunch',
       'GoodForMeal dinner', 'GoodForMeal latenight', 'GoodForMeal lunch',
       'GoodForMealdessert', 'HairSpecializesIn', 'HappyHour', 'HasTV',
       'Music', 'NoiseLevel', 'Open24Hours', 'OutdoorSeating',
       'RestaurantsAttire', 'RestaurantsCounterService', 'RestaurantsDelivery',
       'RestaurantsGoodForGroups', 'RestaurantsPriceRange2',
       'RestaurantsReservations', 'RestaurantsTableService',
       'RestaurantsTakeOut', 'Smoking', 'WheelchairAccessible', 'WiFi',
       'stars'],dtype='object')
```

Most of these attributes are boolean variables, and also there are many NULL values in these attributes. At first, we delete the attributes whose values are higher than 30% percentage. such as 'AcceptsInsurance', 'AgesAllowed', 'GoodForDancing'. The total number of deleted attributes is 21. Then we used one hot encoding method to prepare the training dataset. After data pre-processing, we save the data as ‘dataset_business.csv’.

2. Classification

2.1 prepare

For the classification part, we want to predict the good star(ex: 5, 4.5) and bad star(ex: 1.0, 1.5, 2.0). We also want to explore the affection of star dividition and delete features. We used two ways of star division, the fist way we define star 5.0 as good and star 1.0 and

1.5 as bad, the second way we define star 5.0 and 4.5 as good and star 1.0, 1.5,2.0 and 2.5 as bad.

2.2 Training

In order to get the best model and best performance, we tried out many classification models, including linear models and non-linear models, such as GussianNB, Logistic Regression, linear SVC model, KNN model, Random Forest model. For each model, we training 70% of the data, then testing against the other 30%. Also, we used GridSearchCV do 5 fold cross validation to get the best parameters for each model. After that, we give the training result by training accuracy, testing accuracy, confusion matrix and so on.

2.3 Training Result

All the training result are saved as txt file in result file.

2.3.1

Bellow tables are shown the training results of different model.

Table 1. Delete attributes who has the percentage of Null value more than 30%

Model	Training Accuracy	Testing Accuracy	Recall-Good	Recall- Bad	Model Parameters
GussianNB	0.655	0.642	0.31	0.91	
Logistic Regression	0.713	0.707	0.60	0.79	LogisticRegression(C=70.01, class_weight=None, dual=False,fit_intercept=True,intercept_scaling=1, max_iter=100,multi_class='ovr', n_jobs=1, penalty='l2', random_state=None, solver='liblinear',tol=0.0001, verbose=0,warm_start=False)
linear SVC	0.711	0.708	0.59	0.80	LinearSVC(C=0.01, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='squared_hinge', max_iter=1000,multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,verbose=0)
Random Forest	0.733	0.705	0.49	0.88	RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',max_depth=8, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0,min_impurity_split=None,min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1,oob_score=False, random_state=42, verbose=0,warm_start=False)

KNN	0.770	0.690	0.54	0.81	KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=9, p=2,weights='uniform')

Table 2. Delete attributes who has the percentage of Null value more than 50%

Model	Training Accuracy	Testing Accuracy	Recall-Good	Recall-Bad	Model Parameter
GaussianNB	0.645	0.632	0.29	0.91	
Logistic Regression	0.711	0.706	0.59	0.80	LogisticRegression(C=10.01, class_weight=None, dual=False, fit_intercept=True,intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1, penalty='l2', random_state=None, solver='liblinear', tol=0.0001,verbose=0, warm_start=False)
linear SVC	0.707	0.703	0.58	0.80	LinearSVC(C=0.01, class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1, loss='squared_hinge', max_iter=1000,multi_class='ovr', penalty='l2', random_state=None, tol=0.0001, verbose=0)
Random Forest	0.738	0.705	0.50	0.87	RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=8, max_features='auto',max_leaf_nodes=None,min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1,min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1,oob_score=False, random_state=42, verbose=0, warm_start=False)
KNN	0.76	0.683	0.53	0.80	KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=9, p=2,weights='uniform')

Table 3. Define Label more extremely

Model	Training Accuracy	Testing Accuracy	Recall-Good	Recall-Bad	Model Parameter
GaussianNB	0.530	0.519	0.06	0.99	
Logistic Regression	0.734	0.705	0.69	0.72	LogisticRegression(C=10.01, class_weight=None,dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100,multi_class='ovr', n_jobs=1,

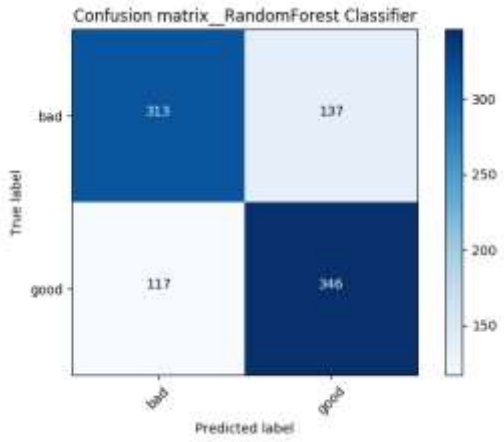
					penalty='l2', random_state=None, solver='liblinear', tol=0.0001,verbose=0, warm_start=False)
linear SVC	0.725	0.692	0.66	0.72	LinearSVC(C=0.01, class_weight=None, dual=True, fit_intercept=True,iintercept_scaling=1, loss='squared_hinge', max_iter=1000, multi_class='ovr', penalty='l2', random_state=None,tol=0.0001,verbose=0)
Random Forest	0.823	0.721	0.75	0.70	RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=8,max_features='auto', max_leaf_nodes=None,min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2,min_weight_fraction_leaf=0.0, n_estimators=500,n_jobs=1, oob_score=False, random_state=42, verbose=0,warm_start=False)
KNN	0.780	0.677	0.59	0.77	KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=5, p=2,weights='uniform')

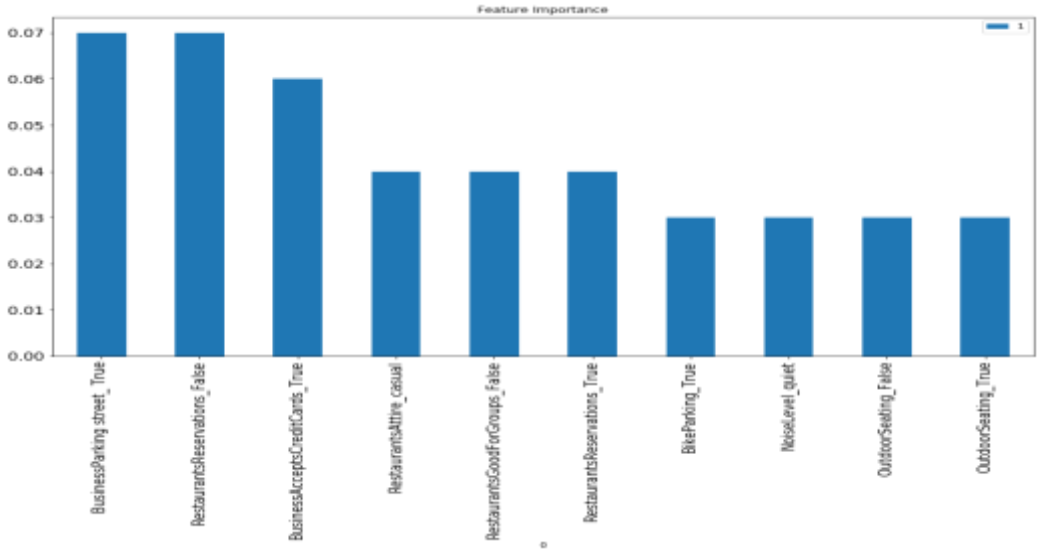
After compared these different variables, we found that:

1. Less attributes sometimes will improve the model's performance, because there are a lot of Null values in these attributes.
2. Because of 1, we think it is better for us to use less attributes.
3. If we define the bad and good more extremely, which means that only the stars 5 business is good, and star 1.0 and 1.5 is bad, we can get higher accuracy.
4. The average accuracy of these models is less than 0.8, we can't say it is a good performance. After did some paper reviews, we found that not only our job but also others who tried predict start by classification models also got the similar result.
5. In all of the models, we got our best model : Random Forest Model.
6. The confusion matrix and Top 10 important features of the best model are in bellow. From these graphs we can see in this model this ten features are very important to decide whether if a restaurant's is a good restaurants.

0	BusinessParkingstreet_True
1	RestaurantsReservations_False
2	BusinessAcceptsCreditCards_True
3	RestaurantsAttire_casual
4	RestaurantsGoodForGroups_False
5	RestaurantsReservations_True
6	BikeParking_True

7	NoiseLevel_quiet
8	OutdoorSeating_False
9	OutdoorSeating_True

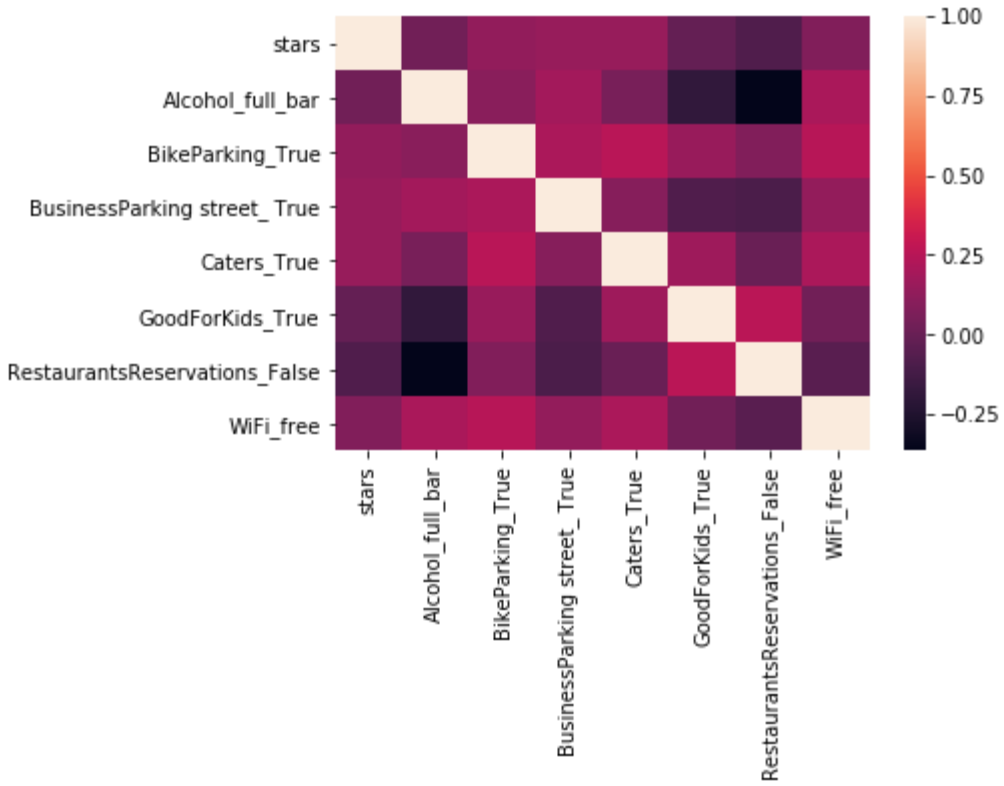




3.Regression

3.1 Correlation

We have 77 business attributes as we presented above, and we are wondering that which attributes are most correlated with star ratings. However, features are too many to present a matrix with all all of them. Looking at the below correlation matrix, which could gives us a gist that bike parking, parking on the street, and Wifi_free have a comparatively high correlation with star ratings.



In all business features, which ones could have a real impact on predicting star ratings? To find out prediction with the highest accuracy, we tried out several regression models including linear regression, decision tree regression, Ridge CV, and SVC regression. For each regression, we utilize 75% of the dataset as our training data, and 25% as our test dataset.

3.2 Linear Regression

We started from linear regression, and below chart listed the top 5 attributes that have a significant impact over star ratings. It is obvious that parking is fairly important when people make their decisions regarding certain restaurant. And the training mean square error (MSE) is 0.741, and the test MSE is 0.757.

Feature	Coefficients
BusinessParking street_ True	-4.931739e+12
BusinessParking valet_ True	4.047394e+12
Ambience classy_ True	3.362815e+12

GoodForMeal breakfast_ True	-3.256624e+12
Ambienceromantic_ True	-2.815944e+12
Training MSE: 0.7412492567199715 Test MSE: 0.7572142703112591	

We also cross-validate the above linear regression using K-fold method. We chose 10 folds to cross-validate the results, and MSE of the validation is the square of the avg MSEs which is 0.746. It is not too far from our training MSE or test MSE.

```
10-fold RMSEs:
[0.7376264896385099, 0.7534747627960025, 0.7410598543565764, 0.736387835472943, 0.7600223523964008, 0.7342258835840773, 0.7399
720340524605, 0.7492238293286, 0.7526578556698972, 0.7547683587490751]
CV RMSE:
0.745991924440831
Std of CV RMSE:
0.012895228941877555
3.4353085182042027
```

Could the above happened because of overfitting? We need try whether regularization uses a generalized cross-validation to prevent overfitting problem with ridge regularization. With ridge regression, we have an MSE of 0.746. After involving polynomial features, we have MSE of 0.729.

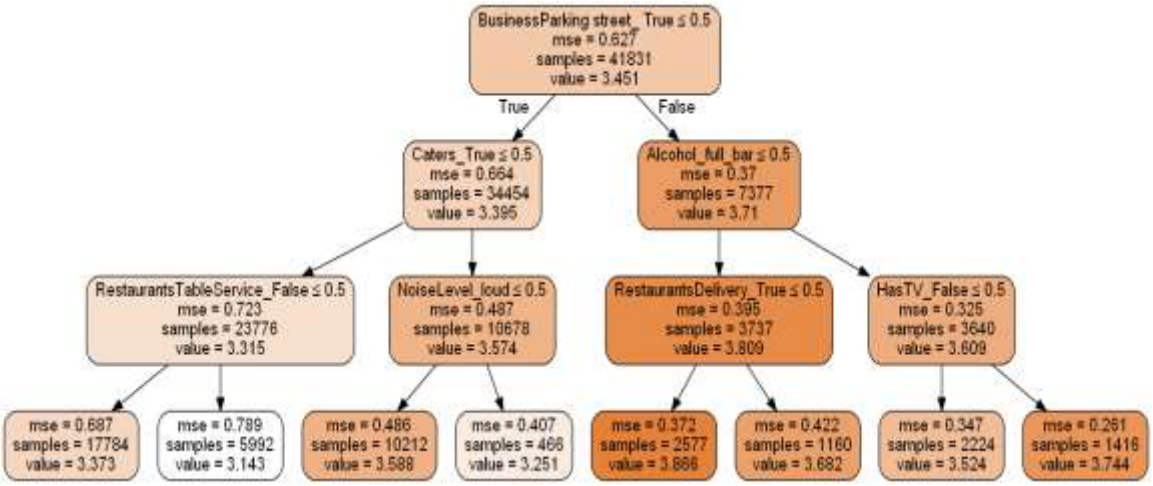
MSE: 0.7458908459319094 PF_MSE: 0.7289958883334875

3.3 Decision Tree Regression

Aside from linear models, we tried out with non-linear models as well. Starting from predicting star ratings using decision tree regression based on attributes like business_parking_street, caters, alchohol_full_bar, restaurant_table_service, noise level, delivery, and has_TV. A decision tree will track down the tree until it reaches a leaf. Each step splits the current subset into two. For a specific split, the contribution of the variable that determined the split is defined as the change in star ratings. We achieve this by limiting the maximum depth of the tree to 3 levels. The below table presents the most import features from decision tree regression model, and we have a training MSE: 0.780, and test MSE: 0.752.

Feature	Importance

BusinessParking street_ True	0.395462
Cater_True	0.322624
RestaurantsTableService_False	0.155524
Alchohol_full_bar	0.048029
NoiseLevel_loud	0.033251

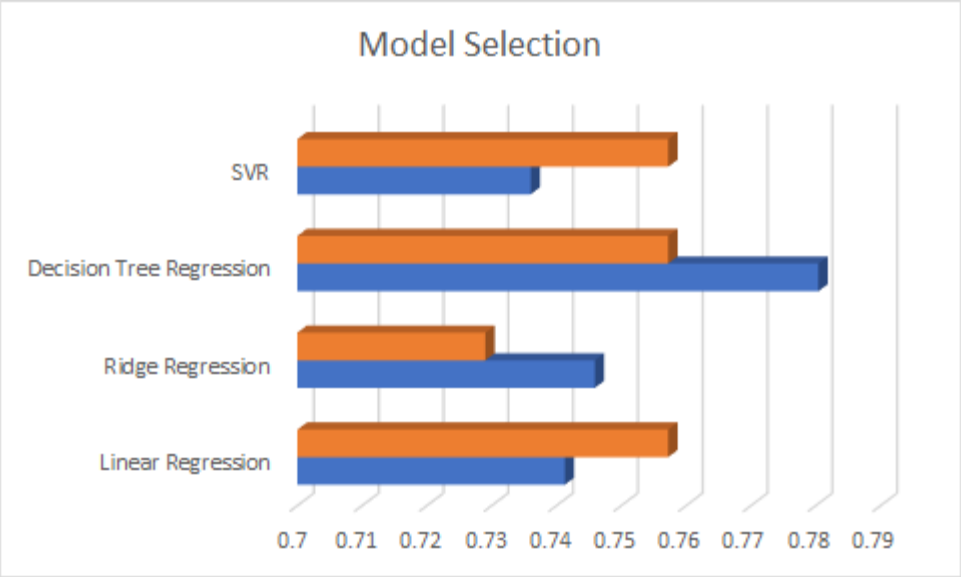


3.4 Support Vector Regression (SVR)

SVR performs regression models in a higher dimension space, which tries to fit as many instances as possible while limiting marginal violations. We choose the kernel “RBF”, which are useful for non-linear hyper-planes. For the support vector regression, we have a training MSE: 0.736, and test MSE: 0.757.

3.5 Conclusion

The best model for predicting star ratings of businesses based on attributes is support vector regression (SVR). However, if we are taking the polynomial features into consideration, then we should choose ridge regression.



Review Data Exploration

1. Data Processing

Attributes in review data include business id, full address, price range, business categories and etc, while attributes in review data include review text, rating, business id and etc. The attributes we used are business id, review text and rating. Specifically, review text is the corpus for our analysis; rating is our identifier for discriminate positive or negative sentiment; business id serves as the key for data munging.

The business dataset was first filtered by the attribute "category" so that only the reviews of restaurants are selected. We got 3115009 restaurants. Then the business dataset was merged with the reviews dataset by the attribute "business id".

Then all the reviews were filtered by the attribute "star", and then 50 thousand of reviews in star 1 are randomly chosen and 50 thousand of reviews in star 5 are random are randomly chosen. Then they are saved as new json files.

We assumed and labeled reviews with ratings equal to 5 as positive while ratings equal to 1 as "negative". This decision was made based on our observation of the distribution of ratings. The decision was made based on our observation of the distribution of ratings.

2. Hot Word Cloud

The words in each review were separated and the punctuations were removed so that a "bag of words" was generated for each review. Finally, we stemmed, lemmatized and filtered out the stop words in each bag of words using both the in-built list in Python's NLTK package.

After separating the vocabulary and remove the stop words, we got the word frequency of in positive reviews and negative reviews. We think the adjective words are more frequently when people want to express their mood. Graph 1 and Graph 2 show the top 100 highest frequency words in two kinds of reviews. The bigger the word is, the more often they appear.

Graph 2. Word Cloud about Positive Reviews

Graph 3. Word Cloud about Negative Reviews

From the two graphs, we can see, there are same high-frequency words in the two kinds of reviews. Such as "good"," service'' and "great". So it is hard to separate the negative reviews and positive reviews by the word frequency. Then we explore the word "good ", “service” and their context.

```
good
negative
Displaying 1 of 16376 matches:
e twice locat desert ridg first time good last time veri disappoint order pepp
positive
Displaying 1 of 23305 matches:
look good authent chines food care craft tri p
```

```
servic
negative
Displaying 1 of 24493 matches:
overs..youl happi incred disappoint servic mean realli realli bad place order
positive
Displaying 1 of 18014 matches:
ay come back food consist veri good servic server manag veri friendli veri pro
```

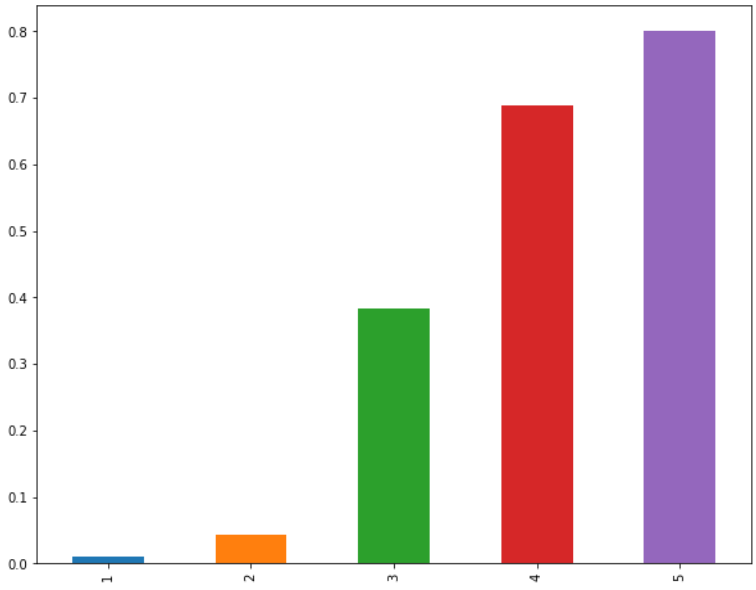
In this graph we can see in negative reviews, good is also used as "not good" in the text. So it further explanation we cannot predict mood by the frequency words. It means we need the machine learning model to get the most reasonable word can represent the different mood in the review text. But we can get an interesting information that when people review the restaurants, they take the service of restaurants as a very important elevation.

3.Machine learning model

3.1 Naive Bayes

At first, we got the word features by the words are in the text and add the label of each text. The label is "pos" and "neg". Then we built a dictionary and used the Naïve Bayes and SVQ model. The data were randomly separated into training and testing set according to ratio 7:3.

Then we got the word features by removing the words are in the text and add the label of each text. We built a dictionary and used the Naïve Bayes model. The data were randomly separated into training and testing set according to ratio 7:3.Bellow graph show the probability of each review to be positive review, the x axis is the True star. From the result, we can see, our model can predict in a high accuracy.



The listing shows that the words in the training set that has "appalled" are negative 40.5 times more often than they are positive. These ratios are known as likelihood ratios, and can be useful for comparing different feature-outcome relationships.

```
Most Informative Features
                appalled = True              neg : pos    =     40.5 : 1.0
                demanded = True              neg : pos    =     26.5 : 1.0
              unhelpful = True               neg : pos    =     24.5 : 1.0
                pathetic = True              neg : pos    =     24.2 : 1.0
                shrugged = True              neg : pos    =     21.9 : 1.0
              cockroach = True               neg : pos    =     21.5 : 1.0
                  rudest = True              neg : pos    =     21.3 : 1.0
                  argued = True              neg : pos    =     21.2 : 1.0
            deliciously = True               pos : neg    =     21.1 : 1.0
                  nerve = True               neg : pos    =     20.7 : 1.0
              defensive = True               neg : pos    =     20.5 : 1.0
                  rancid = True              neg : pos    =     20.5 : 1.0
                disgrace = True              neg : pos    =     19.5 : 1.0
                vomiting = True              neg : pos    =     19.5 : 1.0
                 arguing = True              neg : pos    =     19.5 : 1.0
               insulting = True              neg : pos    =     18.9 : 1.0
             incompetent = True              neg : pos    =     18.6 : 1.0
                    puke = True              neg : pos    =     18.5 : 1.0
              apathetic = True               neg : pos    =     17.5 : 1.0
                 hazard = True               neg : pos    =     17.5 : 1.0
```

From these features, we can see that most of these words are about the service，cleanliness and tasty of the food. We can have a basic conclusion that people put a greater emphasis on the service than other attribute of the restaurants. It’s a similar result with the one we got from word cloud part.

3.2 Word to Vector analysis

The goal of word vector embedding models, or word vector models for short, is to learn dense, numerical vector representations for each term in a corpus vocabulary. The vectors it learns about each term should encode some information about the meaning or concept the term represents, and the relationship between it and other terms in the vocabulary. Word vector models are also fully unsupervised — they learn all of these meanings and relationships solely by analyzing the text of the corpus, without any advance knowledge provided.

We'll train our word2vec model using the normalized sentences with remove the stop words. We set up our training process to run for twelve epochs. We try out some interesting things of our Word2Vec model, we can see that our model knows well that brunch is a combination of breakfast and lunch.

```
word_algebra(add=[u'breakfast', u'lunch'])
```

```
brunch
```

Then we explore the “service” again, we show the Top 10 words similar to service. It seem that there parts are also very important to the customers. If you want make your restaurant's to be a great one, you should carefully consider these.

```
get_related_terms(u'service')
```

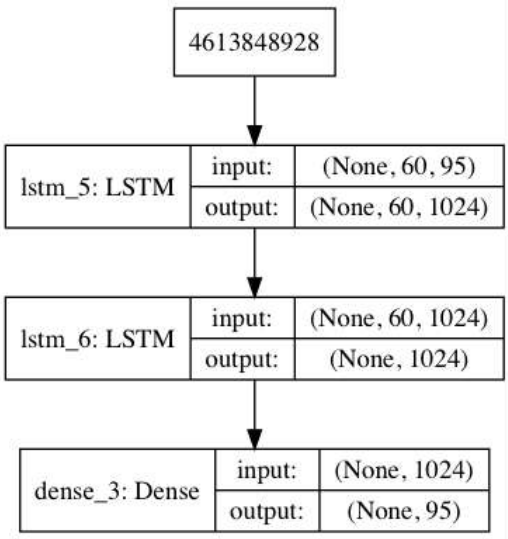
staff	0.666
food	0.594
customer	0.591
attentive	0.526
atmosphere	0.526
ambience	0.52
prices	0.517
friendly	0.513
prompt	0.501
polite	0.495

```
word_algebra(add=[u'service', u'delicious'], subtract=[u'brunch'])
```

```
excellent
```

3.3 Good reviews Generator

In this part, we built a character-level language model. Character level models are more computationally expensive to train than the word model, but it let us don't have to worry about the unknown library. We use LSTM model in RNN, and the data is from the reviews texts of star 5. The model is quite similar to the model here(https://github.com/keras-team/keras/blob/master/examples/lstm_text_generation.py). To return all timesteps' activation values, we set the return_sequences parameter to True. Each input sample is a one-hot representation of 60 characters,Each output is a list of 115 predicted probabilities for each character.



Due to that we have a large dataset, we just choose the reviews text whose length is less than 250. We tried to training our model in one epoche, but it will cost almost 3 hours on GPU, so we selected pre-trained model from Chicago university. Here is the example that we generated from the model. They seem very good and realistic.

"<SOR>So good and so fresh! Love this place for the first time a try. It was delicious and well portioned. Can't wait to go back!<EOR>"

"<SOR>This place is awesome!! I had the all you can eat. Fresh fish tuna salmon yellow curry and squid okonomiyaki. Was very pleased! Had takoyaki on a previous visit and that was good for the price.<EOR>

Conclusion

In this project, we mainly focus on the business data and review data. We conducted data analysis and built up many machine learning models, we tried to find attributes that makes a great restaurant.

We successfully built up our classification models and regression models, though we didn't get a very high accuracy of these models, we did obtain a bunch of interesting results and ideas that what can be achieved in the future. Through our classification and regression models, we retrieved the most important features that are associated with restaurant star ratings. We also taught ourselves some basic NLP. After reviewing the analysis, we received more information on how to make a great restaurant.

After this project, we could provide interesting business advices for the restaurants who want to get a 5-star rating on yelp. As a restaurant owner, you should pay much attention to how to improve the food and service, and attributes like parking, catering and reservation. Also, you might want to try advertising with NLP techniques to generate some "fake" 5- star reviews to improve grading on yelp, but this strategy comes the last until the general services and food quality are improved on our perspectives. After all, great food is the essence of a great restaurant.

Future Work

1. More models or more encoding methods can be tried to improve the accuracy of business.
2. We didn't go too in depth with each model when we trained, maybe there will be more parameters could be tuned in the future.
3. We could establish times series model to examine how previous reviews impact the future review.
4. For the review generator part, word sequence models can be tried.

Reference

Data:

https://www.yelp.com/dataset/challenge

link:

https://www.kaggle.com/yelp-dataset/yelp-dataset

https://rstudio-pubsstatic.s3.amazonaws.com/127262_5d4d64fbf72d4efa9e702869a25fb88b.html

https://minimaxir.com/2015/12/lets-code-1/3https://medium.com/@aariff.deen/creating-a-real-time-star-prediction-application-for-yelp-reviews-using-sentiment-analysis-9c7e94978bf6

https://medium.com/@Vishwacorp/nlp-analysis-of-yelp-restaurant-reviews-30b3d0e424a

https://medium.com/tensorist/classifying-yelp-reviews-using-nltk-and-scikit-learn-c58e71e962d9

http://cs229.stanford.edu/proj2011/BechonGrimaldiMerouchi-ImprovingYelpReviews.pdf

https://nycdatascience.com/blog/student-works/project-1-exploratory-visualizations-of-yelp-academic-datasetdraft/

Paper:

Asghar, N. (2016). Yelp Dataset Challenge: Review Rating Prediction. arXiv preprint arXiv:1605.05362. 5 6

Kevin Reschke, Adam Vogel, and Dan Jurafsky. Generating recommendation dialogs by extracting information from user reviews. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 499–504, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (Vol. 3). London: Pearson.