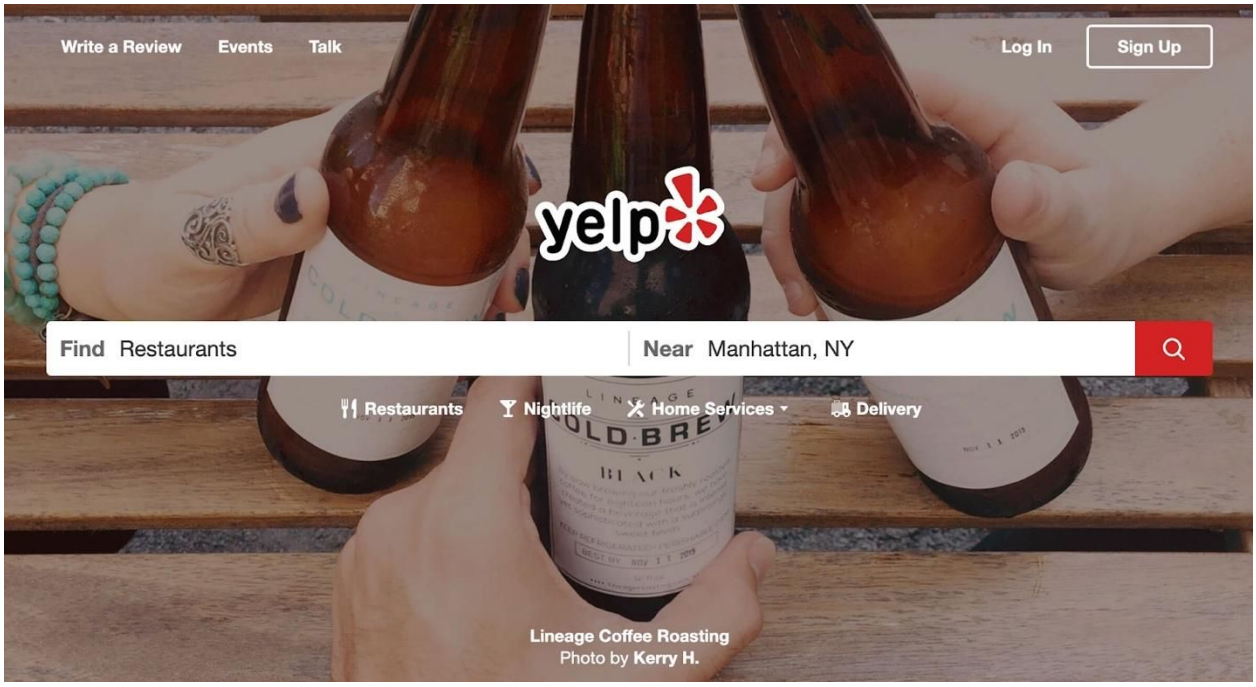


Manhattan Restaurants Reviews on Yelp

BIA 660 Final Project Report

Team Members:
Hongyi Chen
Junhan Zhou
Tingyi Lu
Xiaomin Yang

1. Motivation and objectives

Online reviews play an irreplaceable and indispensable role in customers' purchase decisions. In catering industry, the popularity of restaurants can be effectively evaluated through social medias, which could be used to provide suggestions for customers on restaurant selections as well as provide guidance and predictions for potential restaurant owners to make profits. In this web mining and analytics project, we tend to evaluate the popularity of restaurants in Manhattan area using reviews and ratings on Yelp.

2. Purpose

The purposes of our research are two-folded: Firstly, using multiple Classification models to generate the sentiment of each review based on the review text; Secondly, to generate the classification report of each model and compare the reports to see which model has the highest accuracy. After these two steps, we would be able to tell the restaurant owner what a new review really indicates regardless of what the score maybe, since some of the customers are just too "polite" to give low scores even if they had bad experiences.

3. Data Preparation

In this project, the original dataset comes from the following website: https://www.yelp.com/search?find_desc=Restaurants&find_loc=Manhattan%2C%20NY&ns=1&sortby=review_count. Initially, our group planned to use the BeautifulSoup to scrape the customer information as well as their reviews and scores for the restaurant they visited. However, due to the reason of lacking web scraping experience, we always reached the limit of numbers (10,000) of scraping per day, which caused the Status Code 503, "Service Unavailable" happened. Since the

dataset we planned to scrape should be larger than the limit while using BeautifulSoup, we first scraped about 3,400 pieces of combined data (Restaurant Name, Price Range of the Restaurant, Customer Name, Score, Review Date, Review Content) for initial analysis.

Due to this limitation, the result of one of our models, more specifically, the accuracy of Naive Bayes Model were extremely low(around 0.40), which makes the ROC-AUC analysis performed poorly. Now looking back, the occurrence of limitation should be caused by the lack of sleep time during each round of scraping, making the website detects the scraping.

Later on, when we learnt how to scrape data using Selenium, with the help of sleep(), the limit problem disappeared, which enlarged our dataset to more than 19,000 pieces of data, with 25 restaurants and 19720 pieces of review data. The new dataset significantly increase the performance of the models generally.

4. Data Description

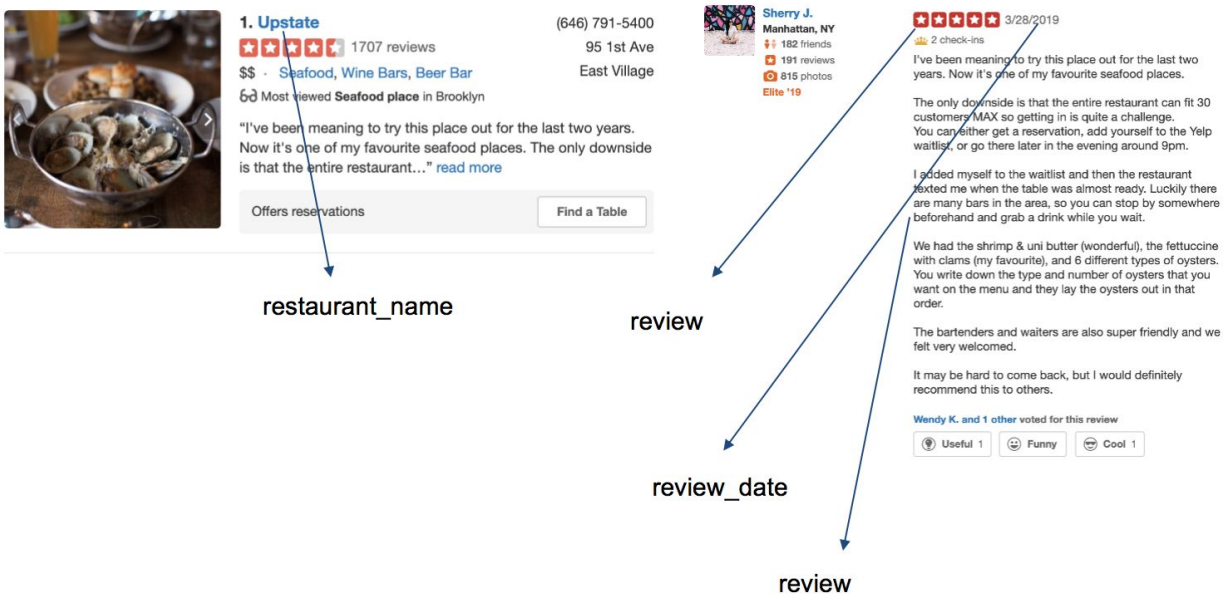
The dataset contains 19720 rows of data, each one representing a customer's review information, which includes the restaurant name, the review content, the score that the customer left for the restaurant and the review date.

Below is a sample of the data:

Restaurant_Name	Review	Score	Review_Date
Ippudo NY	This place is awesome.. i lo	5	4/23/2019
Ippudo NY	Ramen noodles are good, t	4	5/3/2019

This review system will give game company or developer an overall insight of how this running on this platform by collecting all the data from users' reviews.

On Yelp, the interface looks like this:



5. Data Cleansing

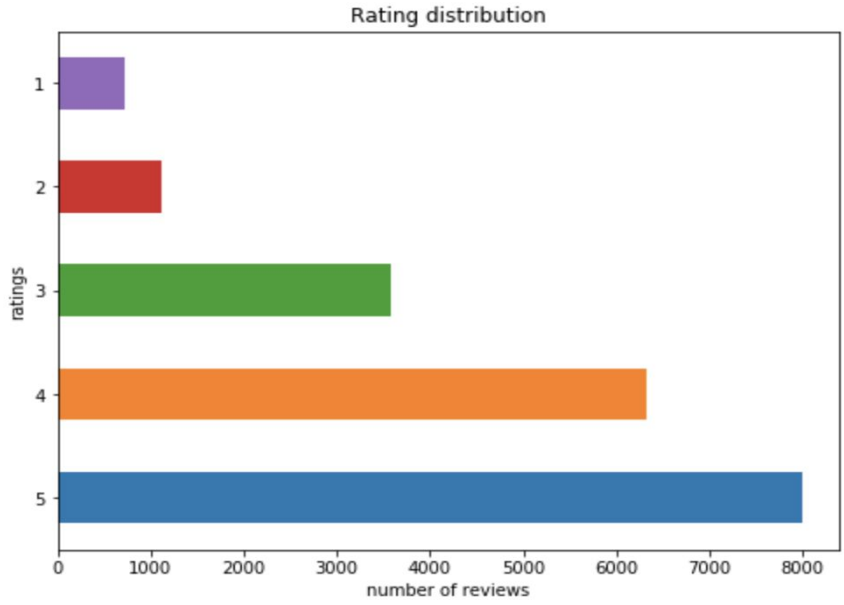
The data contains some symbols and stop words which cause problems for us to find meaningful information from the user reviews and thus we remove these noise and transfer all characters into lower case and split the reviews into words.

6. Exploratory Data Analysis

Without any missing value in the dataset, we embarked on the exploratory data analysis. Firstly, we generated a word cloud to visualize the reviews of all restaurants and to know what kinds of keywords are closely related to those restaurants.

Since the word cloud above is just an initial visualization to give us some ideas on the reviews, a deeper analysis on more specific features will be achieved in the following parts.

Next, we tried to make an analysis of rating distribution to see how popular those restaurants are.

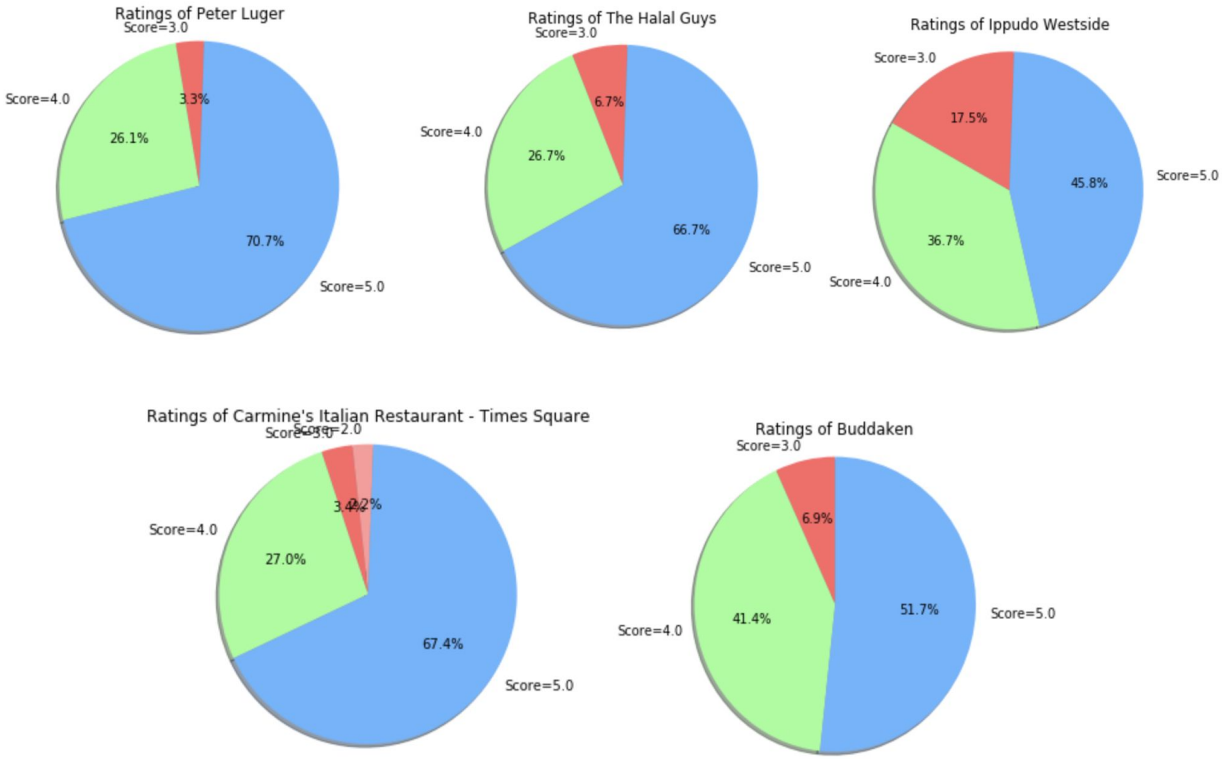


As the bar chart shows above, we find that most restaurants are rated with full score, which indicates the overall popularity of the restaurants are relatively high.

After having an overview of the ratings for all restaurants, we decided to go deep into the most popular ones. Therefore, we computed the average ratings for all restaurants and finally got the top 5 restaurants with the highest scores.

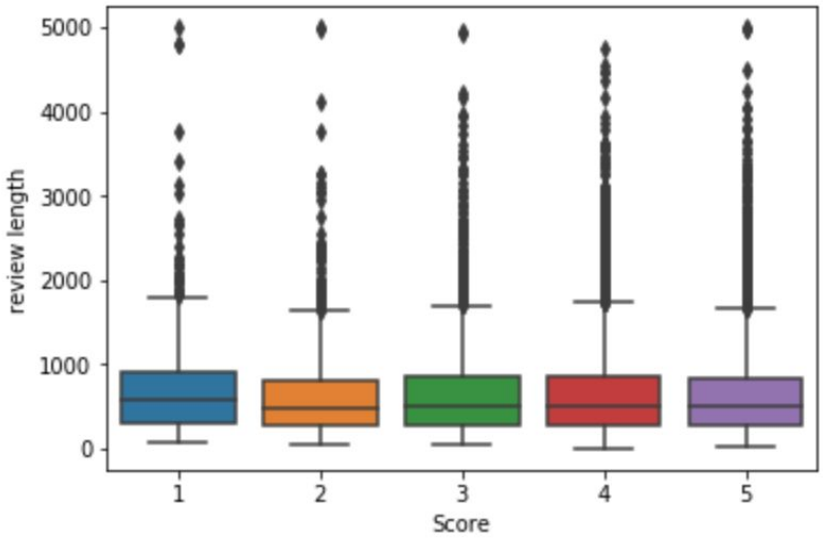
Restaurant_Name	Score	time, food, pretty	wait, good, experience	ordered, table, great	place, service, like
Peter Luger	4.60	1.075000	1.605882	1.052941	1.422059
The Halal Guys	4.50	1.776471	0.916176	0.316176	0.863235
Carmine's Italian Restaurant - Times Square	4.45	1.654412	1.335294	1.083824	1.126471
Ippudo Westside	4.45	1.126471	1.889706	0.970588	1.554412
Buddakan	4.35	1.554412	1.491176	1.400000	1.497059

Also, we utilized pie chart to present the rating distributions of these top 5 restaurants.



In order to know whether review length will be helpful for our analysis, we also generated the box plot as below. We can see the review length is quite uniform among different scores, all around

2000 words. But there are many outliers which can be seen as points above the boxes. Because of this, maybe review length won't be such a useful feature to consider after all.



All the steps done above helped us to understand the current situation of review sentiment distribution as well as the data condition, which leads to our next step, Topic Modeling.

7. Topic Modeling

7.1 LDA Model

In this part, the first thing needed to be done is to convert the "tokenized_sents" dataframe column into array by using ".values" function. Then. it was further converted into list using tolist() function. After that, the tf matrix could be generated from the list. Lastly, the original dataset was

separated into training and testing sets.

	Restaurant_Name	review	Score	review_date	lowerReview	indexedReview	tokenized_sents
0	Ippudo NY	This place is awesome.. i love the soup broth,...	5	4/23/2019	[place, awesome, love, soup, broth, tasty, fla...	[1208712, 1203140, 1208649, 1207272, 1207790, ...	[This, place, is, awesome.., i, love, the, sou...

For the next step, the Latent Dirichlet Allocation model was performed with 5 iterations. This would be helpful in the following step: the generation of word distribution in each topic created in the above step. Also the word cloud of each topic was also generated. From the word cloud, we manually select the top three meaningful words for the use of features.

Lastly, by creating four extra column for the topics' count, our new table of topic analysis is generated.

Restaurant_Name	review	Score	review_date	lowerReview	indexedReview	tokenized_sents	time, food, pretty	wait, good, experience	ordered, table, great	place, service, like
Ippudo NY	This place is awesome.. i love the soup broth,...	5	4/23/2019	['place', 'awesome', 'love', 'soup', 'broth', ...	[1208712, 1203140, 1208649, 1207272, 1207790, ...	['This', 'place', 'is', 'awesome..', 'i', 'lov...	0	1	1	2

8. Data Balancing

Before we put data into modeling, we need to balance the data. The number of negative reviews and positive reviews are different, which could significantly lower the accuracy of models. In order to solve this issue, a new column called Sentiment was created based on the score the customer left for the restaurant. The sentiment 0 means negative reviews, where the scores are 1 or 2. The number of records of negative reviews is 1836.

The sentiment column, 1 means positive reviews, with scores of 4 and 5. The number of records is 14314. By using the python code, we randomly select the same amount of positive reviews as

negative reviews.

review	Sentiment
Food: 2/5 I've been here a handful of times and I never understood what the hype was about. We ordered okonomiyaki, karrage, and three different ramens. I ordered the ichiraku because I'm	0
I have always heard Ippudo was the ramen spot to go in New York. My friends have raved about how good this place was a few years back. I guess it's a reason why I barely hear anyone mentic	0
Hmmm. 3.5 stars rounding to 4. I liked this place, but it's EXTREMELY overpriced. Between a group of 5, we all ordered 1 beer each, 1 bowl of ramen each and shared pork buns and fried chick	0
OMG!!! I don't think this place needs another review but their ramen is out of this world. I ordered the akamaru modern, with their suggested toppings. It comes with this perfectly boiled egg	0
Ugh, one of the best ramen I've had.... I came here expecting a long wait, but got seated within 10 minutes. I ordered the Akamaru Modern with boiled egg and pork buns. Akamaru broth was	0
Good ramen. Very good ambience. Very very good service. Expect a long wait unless you queue in 30 mins before opening time. I had the Shin tantan-men. Extremely spicy. it contained ground	0
Worth the wait so long as it isn't insane. I've been wanting to go to Ippudo for years and I finally got a shot - glad we did.	0
This place is a must if traveling to NYC. the atmosphere here is beyond amazing and the service is over the top. I come here as much as I can and whenever I have friends or family visiting me,	0
I love Ippudo!! I've been to the one in San Francisco but this one is MUCH better. We came on a Saturday night at 5:30 and had to wait an hour and a half, but it was definitely worth the wait :)	0
We were one of the first to get in at 5pm on a Monday. Luckily we arrived at 4:40pm because by 5, there was already a loooong line!! All employees there welcomed each party as we walked t	0
Had an amazing meal here! I was referred to this place multiple times. The pork buns are to die for!! We wound up getting three orders because they were so good. The sauce they put on the	0
SECOND TIME IS THE CHARM. The first time I came here, I had a terrible experience. This had nothing to do with the service or the food-- but I think I had high expectations for how everything	0
I heard so much about this place. I finally decided to see what was the fuss about. People were not wrong. The ramen is exceptionally good. The broth is rich, full of flavor and at the same time	0
Excellent ramen. I'm not sure I have full enough context to go to five, since we've only been once in the last five years and only got two ramen dishes on our last visit, but I did enjoy it a lot. The	0
A girl was straight up fiending for some ramen on her lunchbreak, and I found myself close to the elusive Ippudo I had been hearing about. I busted my puyot over a few blocks, and stepped up	0

This gives us a total number of 3672 reviews with their sentiments. Then we can put the data into multiple models.

9. Supervised Learning

9.1 Multinomial Naïve Bayes

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

The Classification Report and mean AUC score of the Naive Bayes Model is shown below:

```
=== Confusion Matrix ===
[[411 147]
 [ 61 483]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.87      0.74      0.80       558
           1       0.77      0.89      0.82       544

   micro avg       0.81      0.81      0.81      1102
   macro avg       0.82      0.81      0.81      1102
weighted avg       0.82      0.81      0.81      1102


=== All AUC Scores ===
[0.36371692 0.5424297  0.88132089 0.92253958 0.93058837 0.9977552
 0.92251187 0.95207382 0.94439965 0.92896175]


=== Mean AUC Score ===
Mean AUC Score - Naive Bayes:  0.8386297748728937
```

Besides that, we also plot the ROC curve for the Naive Bayes Model:

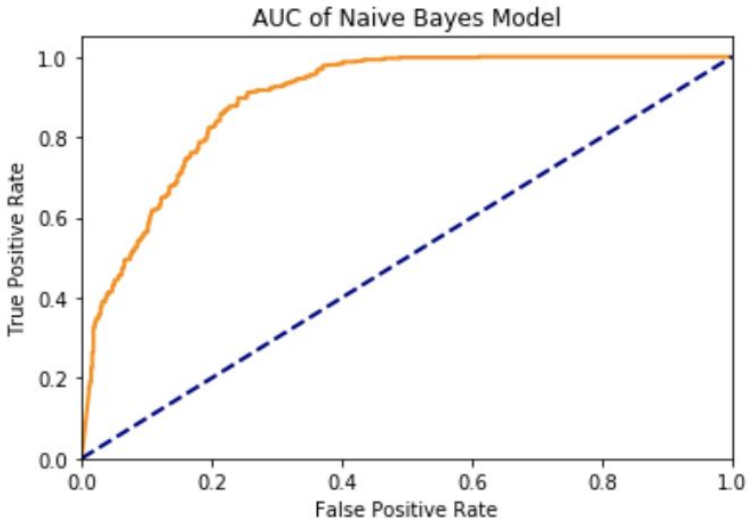


Fig.1: ROC Curve for Multinomial Naive Bayes Model

From the above ROC curve, we could clearly see that the accuracy has been improved significantly since the presentation.

9.2 Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

The Classification Report and mean AUC score of the Support Vector Machine Model is shown below:

```
=== Confusion Matrix ===
[[406 134]
 [133 429]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.75      0.75      0.75       540
           1       0.76      0.76      0.76       562

   micro avg       0.76      0.76      0.76      1102
   macro avg       0.76      0.76      0.76      1102
weighted avg       0.76      0.76      0.76      1102


=== All AUC Scores ===
[0.84699905 0.86002481 0.84729442 0.84726489 0.85216801 0.86011342
 0.84663621 0.86741915 0.79658993 0.85224999]


=== Mean AUC Score ===
Mean AUC Score - SVM:  0.8476759882071425
```

9.3 KNN

KNN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.

The Classification Report and mean AUC score of the KNN Model is shown below:

```
=== Confusion Matrix ===
[[373 167]
 [242 320]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.61      0.69      0.65       540
           1       0.66      0.57      0.61       562

   micro avg       0.63      0.63      0.63      1102
   macro avg       0.63      0.63      0.63      1102
weighted avg       0.63      0.63      0.63      1102


=== All AUC Scores ===
[0.71389119 0.74729738 0.71937027 0.75441576 0.70119034 0.71573724
 0.7236406  0.73530114 0.72756726 0.74121353]


=== Mean AUC Score ===
Mean AUC Score - KNN:  0.7279624709671053
```

9.4 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time

and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The Classification Report and mean AUC score of the Random Forest are shown below.

```
=== Confusion Matrix ===
[[430 110]
 [157 405]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.73      0.80      0.76       540
           1       0.79      0.72      0.75       562

   micro avg       0.76      0.76      0.76      1102
   macro avg       0.76      0.76      0.76      1102
weighted avg       0.76      0.76      0.76      1102


=== All AUC Scores ===
[0.84317403 0.86602079 0.83821184 0.83776879 0.8415495  0.85835598
 0.85726657 0.88791842 0.82762997 0.83485622]


=== Mean AUC Score ===
Mean AUC Score - Random Forest:  0.8492752106611793
```

9.5 Decision Tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

The Classification Report and mean AUC score of the Decision Tree Model are shown below.

```
=== Confusion Matrix ===
[[391 149]
 [115 447]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.77      0.72      0.75       540
           1       0.75      0.80      0.77       562

   micro avg       0.76      0.76      0.76      1102
   macro avg       0.76      0.76      0.76      1102
weighted avg       0.76      0.76      0.76      1102


=== All AUC Scores ===
[0.72826087 0.77173913 0.7798913  0.7826087  0.77445652 0.80163043
 0.79508197 0.7431694  0.75409836 0.74043716]


=== Mean AUC Score ===
Mean AUC Score - Decision Tree:  0.7671373841767641
```

9.6 Logistic Regression

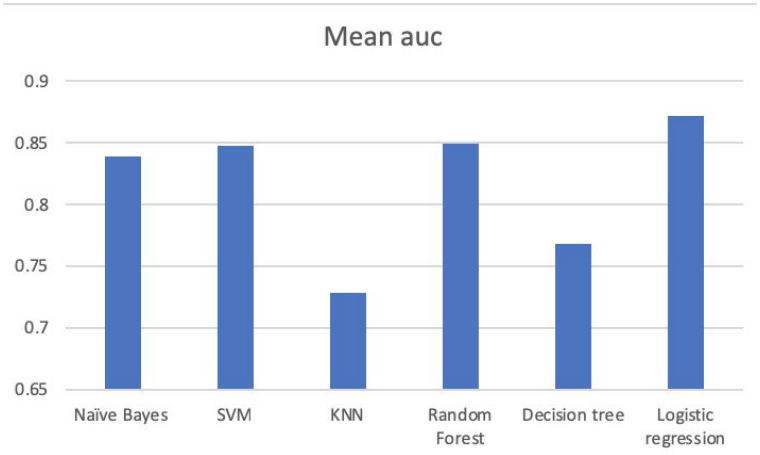
In statistics, the logistic model (or logit model) is a widely used statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

The Classification Report and AUC score of the Logistic Regression Model are shown below.

```
=== Confusion Matrix ===
[[420 120]
 [119 443]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.78      0.78      0.78       540
           1       0.79      0.79      0.79       562

   micro avg       0.78      0.78      0.78      1102
   macro avg       0.78      0.78      0.78      1102
weighted avg       0.78      0.78      0.78      1102


=== All AUC Scores ===
[0.87130789 0.89458294 0.86451441 0.86324433 0.88040525 0.87597472
 0.86535877 0.89020275 0.83591627 0.87757174]


=== Mean AUC Score ===
Mean AUC Score - Logistic Regression:  0.8719079075516488
```

11. Conclusion

From the above Classification Reports, we could easily compare the AUC score of the models. In all six of them, Logistic Regression Model achieved the best AUC score, which is the prediction accuracy of the model, which is about 87%, which is a generally good prediction of the accuracy of the model.



To sum up, from now on, the restaurant owners could start to put new review information into our model to generate the true ideas of their customers and better adjust their services to make more profits.

12. Future Work

According to the conclusion above, the Logistic Regression Model achieved the best performance and should be implemented for restaurants. In the future, we could focus more on the Logistic Regression Model, including but not limited to:

- Feature Analysis based on Logistic Regression Model;
    - Extract features that customers talk about the most for restaurant improvements
    - Detect potential "Fake Reviews" based on Feature Analysis
    - ……
- Text Clustering Based on Logistic Regression Analysis
    - Sentiment Analysis of review based on Multi-feature Fusion
    - …...

Reference & Related Work

1) http://cs229.stanford.edu/proj2017/final-reports/5244334.pdf
2) https://link.springer.com/article/10.1186/s40965-017-0020-9
3) https://en.wikipedia.org/wiki/Random_forest