Olarn Pornpitaksuk

1/28/17

AN ANALYSIS OF CUSTOMER SENTIMENTS ACROSS ALL

THAI RESTAURANTS

in New York City

**INTRODUCTION**

As a member of family that owns Thai restaurants in New York City, listening to customer feedback and understanding how they rate Thai food and service is valuable information for improving a business. Yelp has created a wonderful community where customers can share their experiences about restaurants by sharing ratings and reviews. This has created a treasure trove of data that can be used to assess customer sentiment for restaurants across New York City. Our goal is to create a predictive model that is able to accurately predict the rating a Yelp member gave a Thai restaurant by analyzing the words in their written review. We are very interested in the words in these reviews that possess the highest probability of being in a good (4 or 5) or a bad (1, 2 or 3) review.

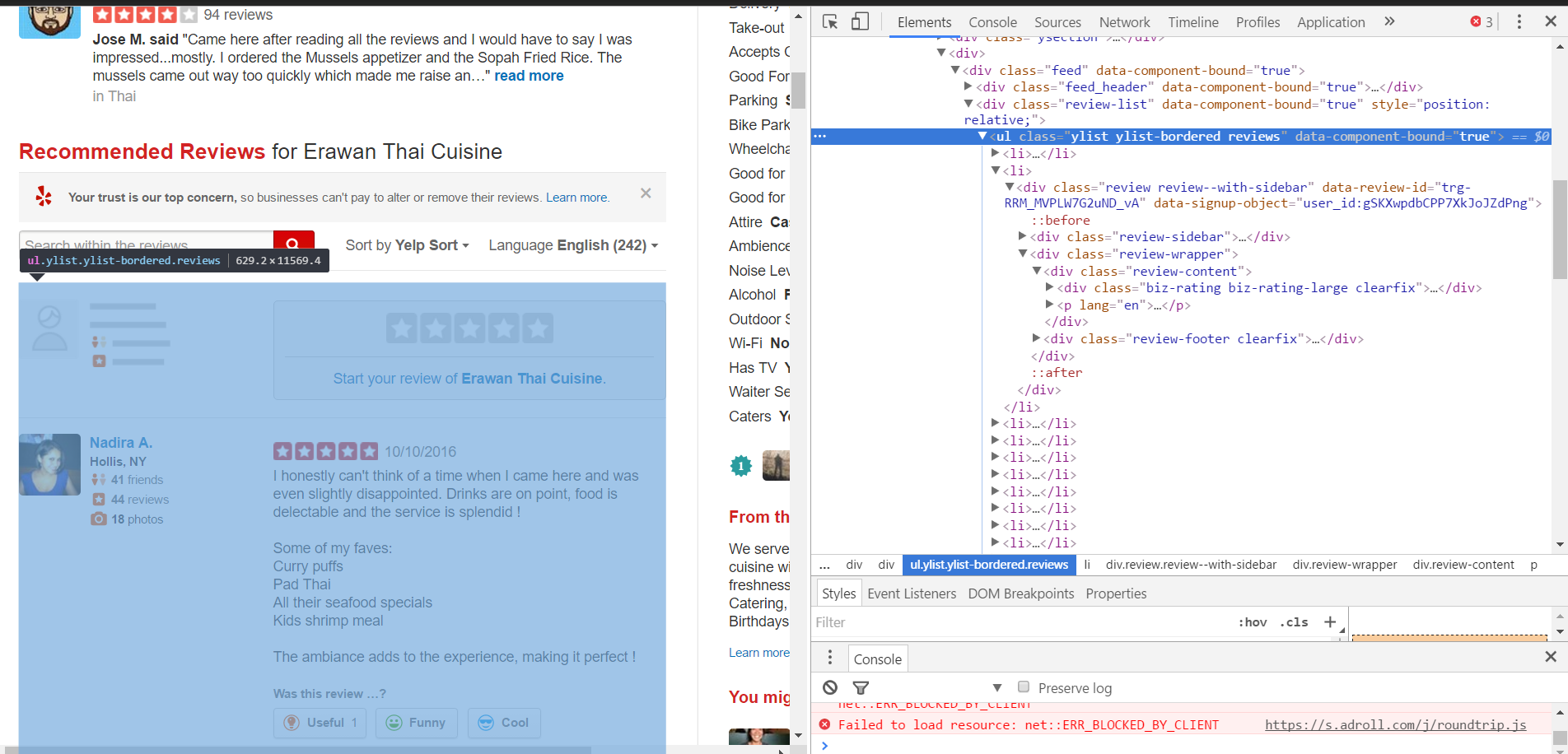
According to Yelp, there are approximately 531 Thai restaurants located within the 5 boroughs of New York. This translates to 83,349 Thai restaurant reviews available for analysis. At a high-level, this project is aimed at building a web-scraper to download the freely available data from the Yelp website and developing a model to predict restaurant rating based on the words within each review.

**METHODS**

**Web Scraper:**

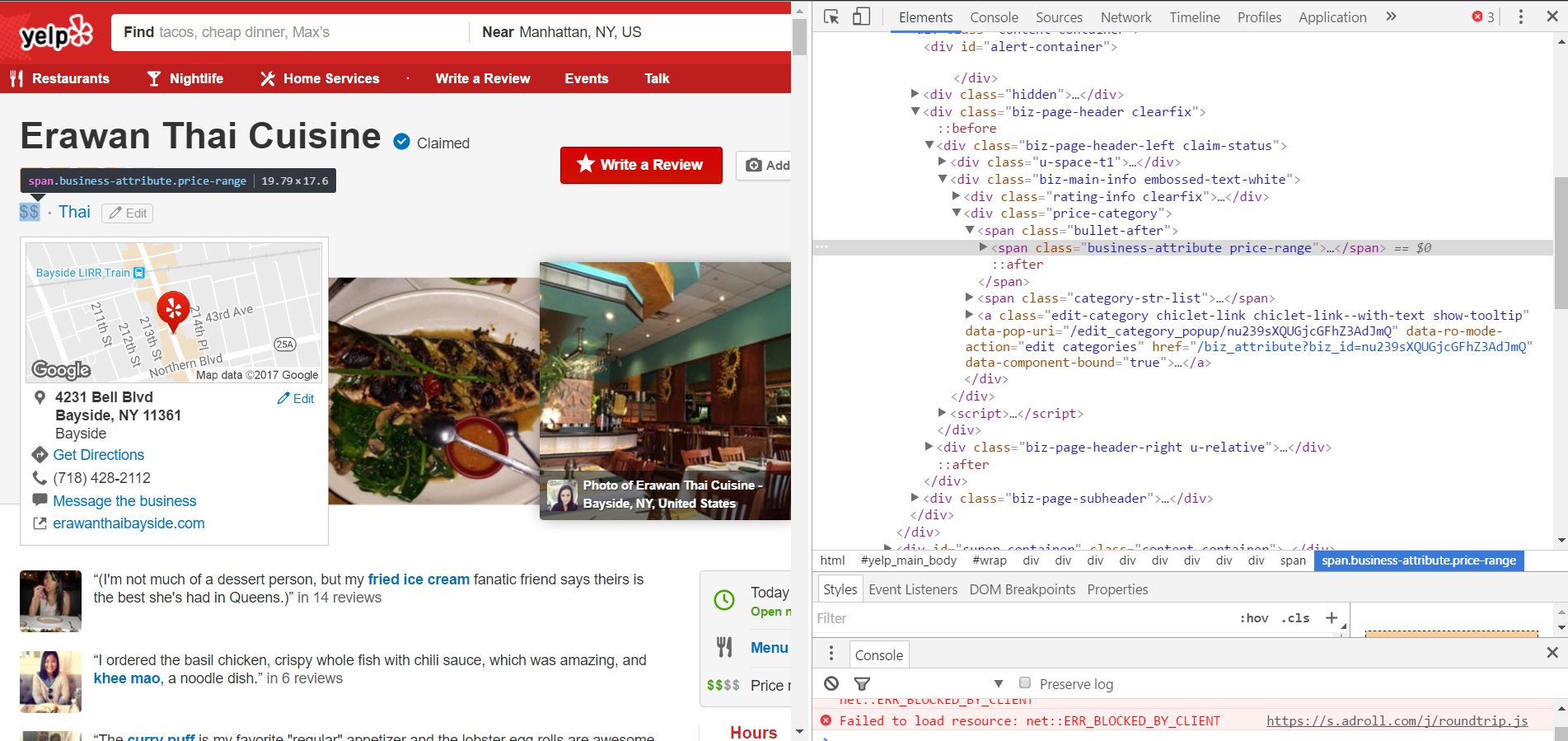
The packages Beautiful Soup and lxml were used to scrape each the Yelp webpage of each restaurant, which contains the review data written by the community and the restaurant’s metadata, or attributes self-reported by the restaurants. Processing was divided into two stages to collect each type of data (i.e. the meta-data and review data). Using the web browser developer tools, the appropriate DOM elements on each Yelp webpage was identified for extraction.

1. **Scraping Restaurant Review Data**



To collect data for each review, the crawler iterated over each page of reviews, with each page containing several reviews (e.g. a Thai restaurants with 20 pages of reviews, with each page containing 20 reviews). This allowed us to collect a total of 83,349 rating and reviews across 541 Thai restaurants.

1. **Scraping Restaurant Meta-Data**



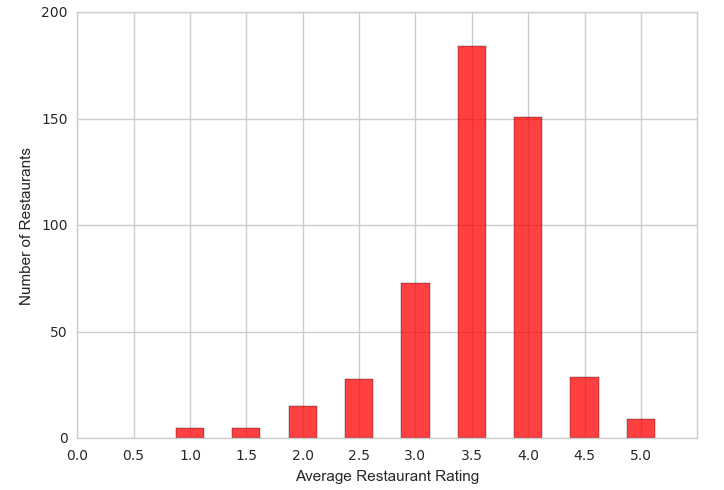
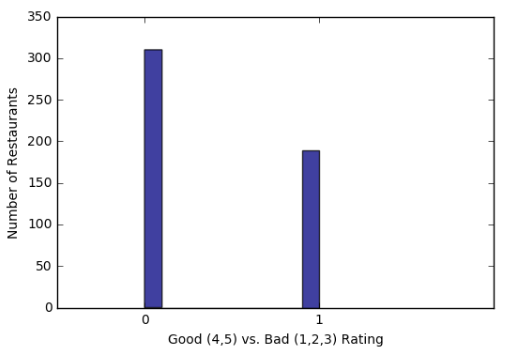
To collect the metadata for each restaurant, the crawler iterated across the restaurants that appeared in a New York City Thai Restaurant query in the Yelp search bar. Links that appeared in this query directly lead to the Yelp home page, where the metadata was stored. The resulting metadata set contained a total of 541 restaurants and 28 distinct attributes/features. Sample of the metadata features collected for each restaurant include:

* The aggregate rating
* The numbers of reviews received
* How expensive the restaurant is
* Availability of alcoholic beverages
* Availability of waiter service

**Exploratory Data Analysis:**

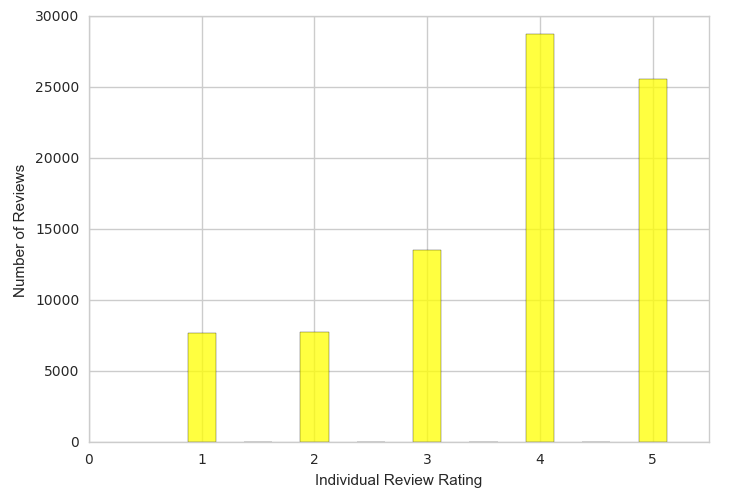
The landscape of Thai restaurants in New York City was analyzed using only the metadata, and was explored with the guidance of several key questions:

1. **What is the distribution of average rating across Thai restaurants in New York City?**

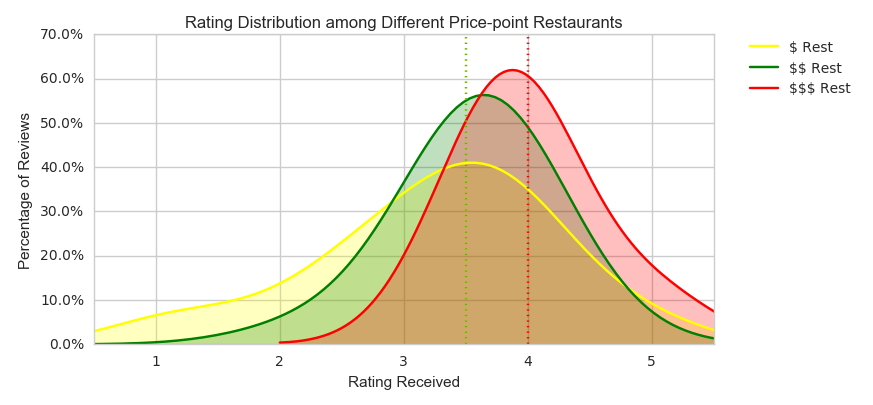
The average Thai restaurant in New York City received roughly a 3.5 rating, and the rating distribution roughly normally distributed with a subtle left skew. It would reasonable to expect that most Thai restaurants received above average reviews (or > 3.0) to be able to stay competitive in New York City. Customers have the ability to discriminate among a plethora of restaurants and cuisines outside of Thai food.

1. **What is distribution of individual ratings across all reviews of Thai Restaurants in New York City?**



The average review written on a Thai restaurant in NEW YORK CITY was roughly a 3.7 rating with a standard deviation of 1.3. Surprisingly, most Thai restaurant reviews written are for a 4.0 or 5.0 rating which can indicate that customers are fairly generous with their reviews or that Thai restaurants in NEW YORK CITY are fairly good.

1. **Do Thai restaurants that are more expensive generally receive higher ratings?**



The majority of restaurants cost between $11-$30 dollars. Within the $11-$30 segment, we notice that the average restaurant had a rating of 3.5 or higher. In the under $10 segment, there was noticeably more variance as the distribution of ratings was the widest. In the $31-$60 segment, we notice that there was significantly less restaurants. However, all restaurants that were within this segment were rated at least a 3.5 or higher. The median rating received for both the below $10 and $11-$30 segment was 3.5 stars, while the median rating received for $31-$60 was 4.0 stars. From analyzing the 3 segments, it is reasonable to state that restaurants that cost more will generally receive both higher and more consistent ratings.

1. **Do Thai restaurants that serve alcohol generally receive higher ratings?**



The majority of restaurants serve alcohol. Within the segment that does not serve alcohol, the average restaurant received an average rating of 3.49 stars, with a standard deviation of 0.63. Within the segment that does serve alcohol, the average restaurant received a rating of 3.56, with a standard deviation of 0.68. Because serving alcohol doesn’t clearly increase the average rating a restaurant receive, it is reasonable to conclude serving alcohol is not a likely predictor for the rating a Thai restaurant receives.

**Data Pre-processing**

The review and rating data were pre-processed using the bag of words model, a technique for approaching text mining problems. This was necessary to be able to feed our text into various machine learning algorithms.

The review data was cleaned and refined using several processes. First, each review written by a Yelp member was broken up to its individual words, including punctuation, using a process called tokenization. All punctuation was removed, and all words were converted into a lower-case form. Then, a word filtering process was used to remove stop words (or common words such as “an, or, the, it” that do not add any insightful value), and only keep meaningful descriptive words pertinent to a restaurant. This process was applied to all 83,349 reviews.

These reviews were then converted into a bag of words representation, which means any unique words/uni-grams (i.e. “delicious”) and pair of words/bi-grams (i.e “simply delicious”) that appeared in the 83,349 review became a feature. The resulting lexicon contained about 71,510 unique features.

Each feature created was tested for appearance across each individual review. If a particular feature (or word) appeared in a review, we would see that word represented as a 1, and 0 if it is not present. This rule was applied to all 83,349 reviews, each containing 71,510 words to be checked. Moreover, the frequency at which words that appeared were calculated to measure each word’s relevance to the study. The more frequently a word appeared, the more important it was to predicting the overall universe of Thai customer sentiment.

**Preprocessor Parameter Selection**

There are two parameters to determine: n-gram length and the minimum document frequency. The n-gram length is defined as the length of combination of words we’d like to include as features in our model, while the minimum document frequency is the number of times a feature must appear for it to be considered a relevant and predictive feature in our model. We chose n-gram to be 2 (or a pair of words) because it significantly increased the vocabulary in the bag of words, and would allow our model to resolve ambiguities encoded in the local positioning patterns in sentences. The minimum document frequency was chosen by using a grid function to search between a range of values (i.e. .00001 through 1) to find the value that maximized the F1 score. A value of .001 did the best job of maximizing the F1 metric.

**ROC AUC Metric Selection**

The ROC AUC was selected as our primary performance metric because it would optimize our model towards the most balanced composition of true positives and true negatives. When determining whether words should be associated with good (4 or 5) or bad (1, 2, 3) ratings, false negatives and false positive errors were viewed as equally bad. A false positive was represented as a review that the model predicted as a good review but is actually a bad review, while false negatives would be vice verse. In the context of improving businesses, both cases are equally harmful to a restaurant owner. A false positive would imply that a particular aspect of the restaurant is better than reality, resulting in failure to recognize and improve a restaurant’s pain points. On the other hand, a false negative would imply improving aspects of a restaurant that were already viewed favorably by customers. In either case, time and capital is misallocated because the strength and weaknesses of the restaurant were poorly characterized. Therefore, an ideal model equally weights the adverse impact of each error, and reduces their frequency with a balanced approach. The ROC AUC metric meets this need and is the best metric for evaluating our text classifier algorithms.

**Feature Reduction**

The chi-squared feature selection method was applied to reduce our lexicon of 71,510 words. The chi-squared method is used in text classification to determine whether the occurrence of a specific term and the occurrence of a specific class is independent. High scores on chi-squared indicate that the null hypothesis should be rejected, and therefore, the occurrence of the term and class are highly dependent. Words that possess a high chi-squared value are desired because they have the strongest likelihood of prediction to either the good or bad class, as opposed to be independent. A p-value of 0.05 was chosen as the threshold for alpha, meaning that words with a 5% probability of being associated with a class due to chance were included in our lexicon.

Values with a probability lower than 5% were kept, effectively reducing our lexicon to 24,915 features. This lexicon contained words with the highest levels of dependency with each class, in addition to reducing the calculation-time performance of our models.

**MODELING AND ANALYSIS**

**Model Selection**

A total of 3 machine learning algorithms were applied to classify and predict the rating of review data: Naïve bayes, Logistic regression, and Support Vector Classification.

1. The Naïve Bayes classifier was chosen because it was known as being one of the simplest and effective text classifying algorithms. It is computationally inexpensive, fast to train, and often times performs very close to more complicated and less efficient techniques.
2. The Logistic Regression was chosen because it’s ability assign weights to features for binary classification. In addition to being effective, it is also incredibly easy to implement, computationally inexpensive and the knowledge representation would be fairly easy to interpret.
3. Support Vector Classification performs incredibly well in high-dimensional input space. Text classification is known to have an incredible amount of features (>10,000) due to the number of words that existing in lexicon. Since SVMs can prevent over-fitting by using an (Gaussian) radial basis function kernel with a small scale factor (kernel parameter), they can be incredibly effective in large feature spaces. However, the one pitfall to note is that SVC does not directly provide probability estimates. They are instead approximated by applying logistic regression on the SVC per-class scores for each sample.

**Model Testing and Parameter Estimation**

All classifiers were trained using 80% of the review data, and tested using the remaining 20%. The training set was split into a total of 5 folds to find the optimal parameter values for each respective model. This was achieved by applying a search grid function, which effectively scans across a range of values for each parameter and then identifies the values that maximize the score of the F1 metric.

When applied to the Naïve Bayes classifier, the search grid function was used to identify the most effective parameters for alpha. By scanning through the values of alpha from .0001 through 10,000 by orders of magnitude, alpha = 1.0 was discovered as the highest performing parameters against the F1 scoring metric.

When applied to the SVM classifier, the search grid function was used to identify the most effective parameters for alpha. By scanning through the values of hyperparameter C from .01 through 100 by orders of magnitude, C = 100 was discovered as the highest performing parameters against the F1 scoring metric.

**Model Evaluation and Analysis**

The performance of each classifier was evaluated by assessing the F1 score, confusion matrix, ROC curve, and precision-recall curve.

* 1. **F1 Score**

Surprisingly, all three classifiers performed equally well on the F1 metric with a score of 0.88. This indicated that harmonic mean of precision and recall be equally maximized effectively as moderate performance on both would be favored over exceptional performance on any individual metric.

**Naïve Bayes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 Score** | **Support** |
| **Bad Review** | 0.84 | 0.81 | 0.83 | 8761 |
| **Good Review** | 0.90 | 0.92 | 0.91 | 16244 |
| **Average/Total** | 0.88 | 0.88 | 0.88 | 25005 |

**Logistic Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 Score** | **Support** |
| **Bad Review** | 0.85 | 0.80 | 0.82 | 8713 |
| **Good Review** | 0.89 | 0.92 | 0.91 | 16292 |
| **Average/Total** | 0.88 | 0.88 | 0.88 | 25005 |

**Support Vector Classification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 Score** | **Support** |
| **Bad Review** | 0.89 | 0.74 | 0.81 | 8660 |
| **Good Review** | 0.88 | 0.95 | 0.91 | 16345 |
| **Average/Total** | 0.88 | 0.88 | 0.88 | 25005 |

* 1. **Confusion Matrix**

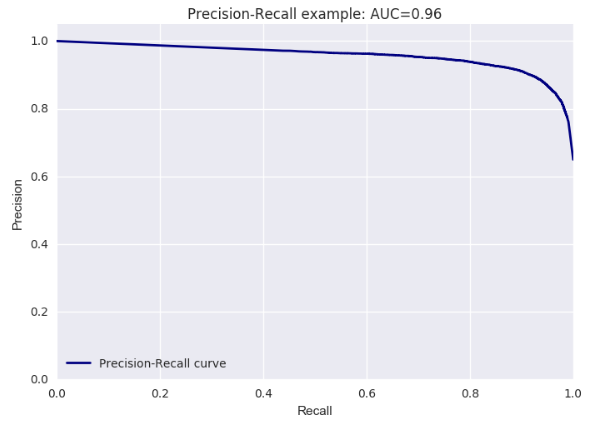
The confusion matrix provided a more granular perspective on the model's predictive performance. We were able to assess the proportion of values that were accurately or incorrectly predicted, while also identifying the type of incorrect predictions (i.e. false positive or false negatives) that were introduced into the model. The confusion matrices below allow us to identify that Logistic Regression classifiers performed strongest because it possessed the lowest amount of prediction errors and the most balanced ratio between false negatives and false positives. This was followed by the Naïve Bayes classifier which followed had more errors but a similar error ratio. The weakest performing was the SVM Classifier because it had the largest imbalance in the error ratio.

Naive Bayes 
Predicted 
Positive 
Negative 
Support Vector 
Classification 
Predicted 
Negative Positive 
Logistic Regression 
Predicted 
Positive 
Negative 
812 
2212 
5 

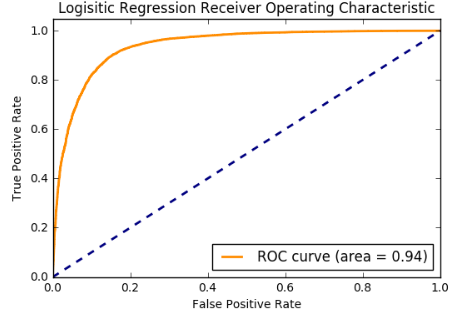
* 1. **ROC and Precision-Recall Curve**

Surprisingly, all three classifiers achieved a Precision-Recall AUC score of 0.96, with marginal differences in the ROC AUC score. The Logistic Regression achieved the highest ROC AUC score at 0.94, while the Naïve Bayes and SVC achieved 0.01 less at 0.93. This indicated that all classifiers performed incredibly well when predicting accurately, limiting errors and recalling a large portion of the dataset.

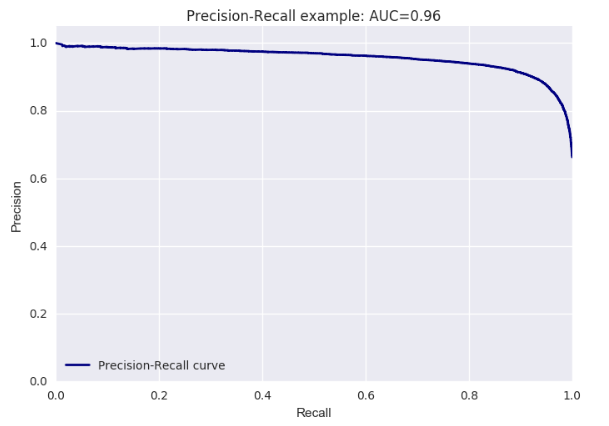
**Naïve Bayes**

Receiver Operating Characteristic 
ROC (area 0.93) 
Positive  

**Logistic Regression**

 Precision-Recall example: AUC=O.96 
Precision-Recall curve 

**Support Vector Classification**

Receiver Operating Characteristic 
Roc curve (area = 0.93) 
Faiso Positive Rato  

Because the F1 score and precision-recall performance of all three classifiers were incredibly similar, the only differentiator of performance was the confusion matrix and the ROC curve. The Logistic Regression was the highest performing classifier when evaluated on these remaining metrics. According to the confusion matrix, Logistic Regression predicted less errors than both Naïve Bayes and SVC. In addition, Logistic Regression also had the highest ROC AUC value at 0.94 when compared to 0.93 of the other classifiers. Therefore, the classifier chosen to evaluate our Thai restaurant was the model produced by Logistic Regression.

**Business Insights**

The Logistic Regression classifier determined the probability that the text in a review belonged to a good (4 or 5 stars) or bad (1, 2, or 3 stars) review category. The words with the highest and lowest probability were the most influential in determining the rating of a particular review. The presence of these words had the highest impact on the category of any review. Because words from reviews written for all Thai restaurants in New York City were used to train our classifier, a "universal" dictionary was created to describe what customers liked and disliked about their Thai restaurant experience.

The objective of this project was to uncover business insights for my restaurant’s and ultimately provide insights to support capital allocation decisions. By feeding my restaurant’s Yelp reviews into the Logistic Regression classifier, insights can be extracted in two ways. First, we can observe which highly influential words (or words with very high or low probabilities) appeared in my restaurant’s reviews, and at what frequency did these words appear. Second, for all words from my uncle’s reviews that appeared as part of the “universal” dictionary, we can compare the probabilities predicted by classifier with the local probability (or categorical probabilities calculated with only reviews within my restaurant) to observe how the restaurant is performing against the “universal” average. For example, if the universal classifier predicted “Pad Thai” appeared in a good review 60% of the time, and my restaurant’s Pad Thai appear in good reviews 80% of the time, then we might think that my restaurant’s Pad Thai is rated more superior than the average Pad Thai in New York City. The two methods allowed us to identify both strengths and weaknesses highlighted by customers who come to my restaurant.

1. **Presence and Frequency of Highly Influential Words**

The Logistic Regression output of my restaurant was analyzed for the appearance and frequency of highly influential words.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MOST INFLUENTIAL WORDS FOR  GOOD REVIEWS | | | MOST INFLUENTIAL WORDS FOR  BAD REVIEWS | | |
| Word/Feature | **Prob Good Review (Universal)** | **Frequency of Reviews** | **Word/Feature** | **Prob Good Review (Universal)** | **Frequency of Reviews** |
| delicious | 83% | 54 | **bit high** | 36% | 14 |
| happy | 82% | 10 | **bumping** | 37% | 1 |
| disappoint | 81% | 12 | **ridiculous** | 37% | 3 |
| felt bad | 80% | 11 | **disappointment** | 37% | 4 |
| fantastic | 79% | 11 | **food poisoning** | 37% | 2 |
| love | 77% | 39 | **tasted old** | 37% | 1 |
| really good | 77% | 14 | **underwhelming** | 37% | 1 |
| also yummy | 74% | 15 | **hype** | 37% | 1 |
| often | 74% | 12 | **miserable** | 37% | 1 |
| yum | 72% | 13 | **increased** | 38% | 1 |
| good food | 72% | 12 | **improve** | 38% | 1 |
| best | 72% | 48 | **mushy** | 38% | 2 |
| dipping sauces | 72% | 15 | **refused** | 38% | 1 |
| whole thing | 71% | 16 | **okay** | 38% | 12 |
| reasonable | 71% | 17 | **extremely rude** | 39% | 10 |
| romantic | 71% | 12 | **better thai** | 41% | 8 |
| great pad | 70% | 11 | **basic** | 43% | 18 |

For top 17 words highly influencing good reviews, we observe many common and expected words to be highly correlated with good review (e.g. delicious, happy, fantastic, love). At the same time, we also observe some nuggets of insights, such as the fact that “dipping sauces were” frequently mentioned as good, or being a “romantic” venue for dinner.

On the other hand, for top 17 words highly influencing bad reviews, we also observe common words associated with bad experiences (e.g. food poisoning, underwhelming, miserable, mushy). These words are expected because the quality and consistency of restaurant experiences inevitably fluctuate because managing the performance of staff can be difficult. Moreover, we also observe some insights into pain points of the restaurant as there may be a member of the staff that is “extremely rude” to customers, which raises a red flag. Several people have also voiced that the food is basic, or have had “better thai” food elsewhere, which may raise a concern on the restaurant’s ability to differentiate. Lastly, “bit high” may be referring to the price, which indicates that we could be pricing our menu items are too

The total lexicon contained over 2000 words to analyze. In addition to analyzing the most influential words, we could also analyze the words that were not highly influential and figure out why that may be the case. Perhaps we thought that my restaurant’s “spring rolls” were highly rated, but our data shows us that customers actually did not like them as much as we had expected. Insights such as these would provide another level of objectivity to evaluate our own standards and metrics for quality.

1. **Presence and Frequency of Highly Influential Words**

To assess what areas the restaurant were outperforming the universal average, we calculated the local probability of each feature and odds ratio between the local and universal probabilities. The odds ratio gave us a metric to assess exactly how much better various words/features were performing within our restaurant, when compared to the majority of Thai restaurants in New York City.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Word/Feature | Prob Good Review (Universal) | Prob Good Review (Local) | Odds Ratio | Frequency of Reviews |
| coconut | 58% | 90% | 6.517241379 | 12 |
| large | 62% | 88% | 4.494623656 | 8 |
| cocktails | 58% | 83% | 3.535496957 | 6 |
| wine | 65% | 86% | 3.307692308 | 7 |
| specials | 54% | 79% | 3.204585538 | 15 |
| sticky | 56% | 80% | 3.142857143 | 6 |
| excellent | 81% | 92% | 2.697530864 | 16 |
| pork | 61% | 80% | 2.557377049 | 5 |
| presented | 54% | 75% | 2.555555556 | 4 |
| red curry | 66% | 83% | 2.515151515 | 11 |
| mango | 66% | 83% | 2.515151515 | 9 |
| tofu | 62% | 80% | 2.451612903 | 7 |
| pla | 62% | 80% | 2.451612903 | 10 |
| curry dishes | 57% | 75% | 2.263157895 | 4 |
| fried ice | 59% | 76% | 2.200564972 | 19 |
| tom yum | 58% | 75% | 2.172413793 | 4 |

The results in this graph are outstanding as we can identify the restaurant favorites identified by both customers and competitively across restaurants. As an example, the alcoholic beverages (i.e. coconut beverages, cocktails, wine) and specific dishes (i.e sticky rice, pork, red curry, mango, tofu and fried fish) were some of the restaurant’s highly distinguished dishes.

**Recommendation**

Based on analyzing the lexicon, we were able to identify strength and weaknesses of the restaurant voiced by customers and ranked competitively across other Thai restaurants in New York City. By understanding the points of differentiation of the restaurant, the restaurant can strategically promote various dishes. For example, waiters can be told to recommend top favorites to customers or daily/promotional specials can be tailored around what customers enjoy most. By understanding the weaknesses of the restaurant, a manager can be more vigilant of any rude staff members and other points the restaurant has trouble being consistent with. It is clear that a text classifier possesses enormous utility for processing restaurant review data for business insights.

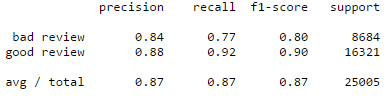
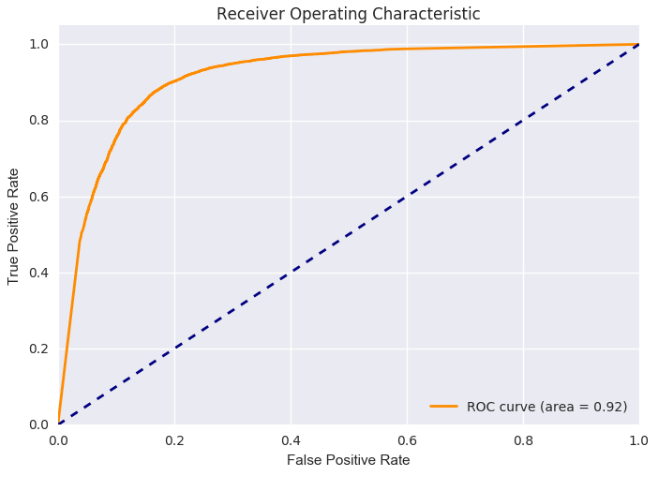
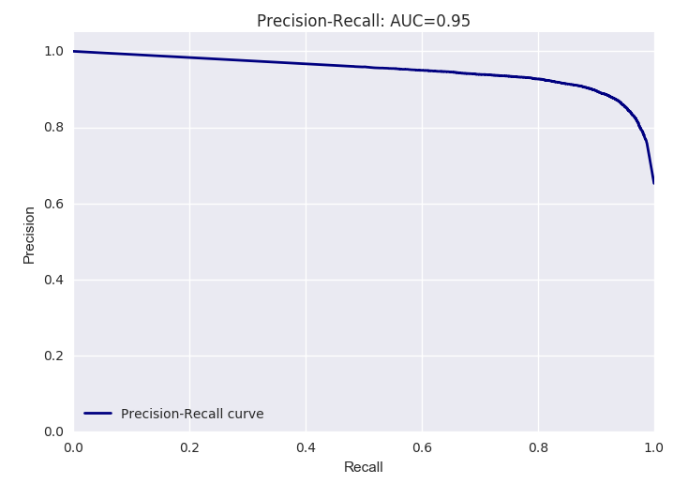
**Conclusions**

The review data for all Thai restaurants in New York City was scraped from the Yelp website, and trained and tested on three different models – Naïve Bayes, Logistic Regression and Support Vector Classifiers. The Logistic Regression classifier had the strongest performance based on lowest error rates and the higher AUC on the ROC curve. When the classifier was fed with my restaurant’s review data, business insights were extracted by observing the presence of highly influential words and assessing the performance of words against the universal probability. Business insights extracted help the business identify points of differentiation and weaknesses in the form of dishes, beverages or service. These insights can be to support business and investment decisions improving the quality and experience of my restaurant.

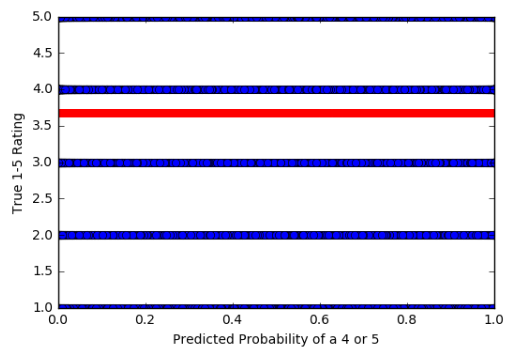
**Appendix**

Several other methods were tested to determine the most suitable and best performing processes for text classification.

1. The TFIDF bag of words representation was tested on our lexicon because the method was able to remove weighting biases from frequently appear features. In the context of text classification of restaurant reviews, shorter reviews with fewer words would be weighted more heavily than words that come from longer reviews. After testing this method on our lexicon, the TFIDF methodology resulted in poorer performance than CountVectorizer. The number of features that had a high probability (greater than 80%) of predicting a review to be a 4 or a 5 significantly dropped, which increased the number of false negative as observed in the resulting confusion matrix. We believe this is because TFIDF attempts to equalize between longer and shorter lengths. However, this is not valued in our context because longer reviews generally provide more detail and useful nuggets of information on the performance of the restaurant.
2. Adding the meta-feature to the lexicon (e.g. other cuisines, serving alcoholic beverages, price range) to observe if any of the meta-data features were strong predictors of good or bad ratings. We found that this wouldn’t effective for two primary reasons. Because several meta-features existed in isolation with a select few restaurants, the predictive probability of those features were determined by a very small sample of restaurants, which provided a very skewed and localized rating of the words. (i.e. there was one Cajun Thai restaurant that had high ratings, so the feature “Cajun\_cuisine” was rated at 4.0 despite only occurring once). To fix this, an option was to apply a minimum frequency to the meta-data features of all restaurants. However, because we only had 541 Thai restaurants in our data set, applying this limitation would reduce the relevance of our data set. Moreover, when observing the probabilities of meta-data features that did occur frequently across all restaurants, we found that none of the probabilities were very high or low. This indicated that meta-data features were not considered as the more influential features in determining a good vs. bad rating in the model. Therefore, adding the meta-data features would introduce faulty and non-discriminatory features to our lexicon, which would not be helpful to our model’s performance.
3. The effectiveness of the chi-squared feature reduction was assessed by comparing the performance of the Naïve Bayes when using feature reduction and without using feature reduction. Surprisingly, performance without feature reduction was marginally worse – the F1 score, ROC AUC score and Precision-recall AUC score was only 0.01 less than with feature reduction. However, there were other tangible benefits of feature reduction. It reduced the computational load (which was significantly more important for the SVC classifier) and helped to reduce the number of noisy features for analyzing feature performance for insights.



1. The possibility of mapping our model features from a good-bad rating back to the original 1,2,3,4 or 5 ratings was tested. We wanted to see if our feature’s probabilities intuitively mapped to a rating. For example, would a word with a classifier probability between 0.00 -0.20 would map to true rating of 1, 0.21-0.40 map to a true rating of a 2, etc. By plotting the true rating with the good-bad rating probabilities from the model in a scatter plot (in blue below) and then plotting a linear regression for the best fit line (in red below), we were able to test for positive correlation between feature probability and true ratings.



The graph observed indicated that there was not a positive relationship between probability and the true ratings. Although in theory, we do believe that is the case, the particular scatter plot and regression line did not show that because there is a noticeable imbalance between good and bad restaurant reviews within our data set. The imbalance in data resulted in skewing the regression line towards the average rating of all words.

1. A truncated version of Singular Value Decomposition (a form of PCA) was applied to the sparse matrix produced by the term-document matrix. Unlike PCA, estimator does not center data before computing SVD, which means it can work with sparse matrices more efficiently. When truncated SVD is applied to term-document matrices, this transformation is known as Latent Semantic Analysis or LSA and transforms a “semantic” space of low dimensionality. This gets rid of the effects of synonyms and polysemy - which means multiple meanings per word. However, because features in the SVD are semantic, it means that features lose their interpretable meaning in the lexicon. Therefore, this method was determined to not be as useful for text classification because we would like to retain the real-world meaning of the features.