Toni Hanrahan

Restaurant Reviews and Health Scores – Text Mining

**Introduction:**

The city of San Francisco has the largest number of restaurants per capita of any city in the United States – nearly one restaurant for every 280 people! To be fair, this is a slightly distorted number as the cost of housing in the city itself is tremendous, so the number of people living in the city proper has declined while the number of businesses including restaurants has increased. Yet, 300,000 people commute into the city every day, and an additional 25 million tourists visit San Francisco each year, and nearly every one of them needs to eat in a restaurant.

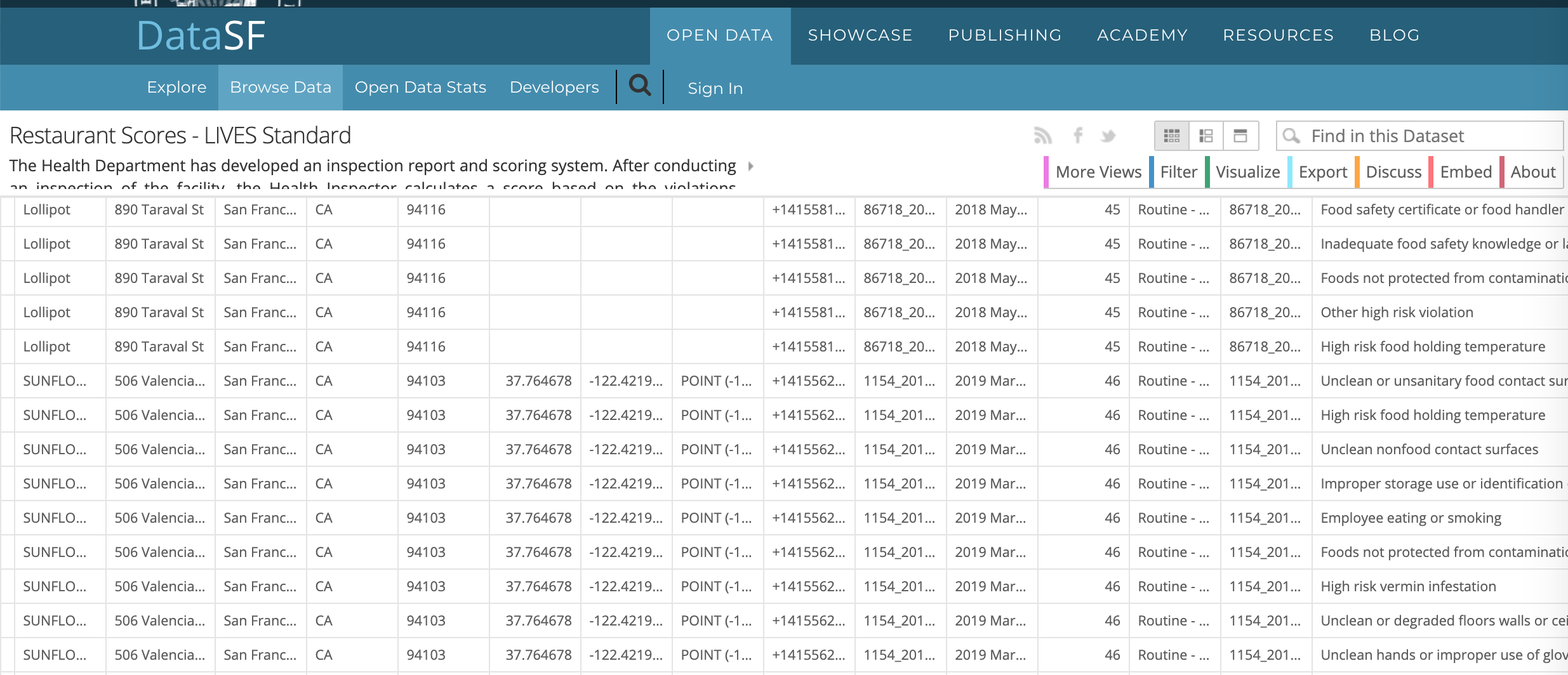
With the thousands of choices available, it can be challenging to find a good restaurant. “Star Ratings” are commonly used to rate good and services, but they often don’t tell the whole story. Many highly rated restaurants have low health scores and could be unsafe places to eat. So, what is a consumer to do? Read through hundreds of individual reviews and try to detect possible signs of an unsafe restaurant?

Well, not exactly. Instead, just a few people are needed to read through reviews and apply their world knowledge and human understanding to identify safe and unsafe restaurants. This information can then be read by a model, which can then create a safety prediction for other restaurants.

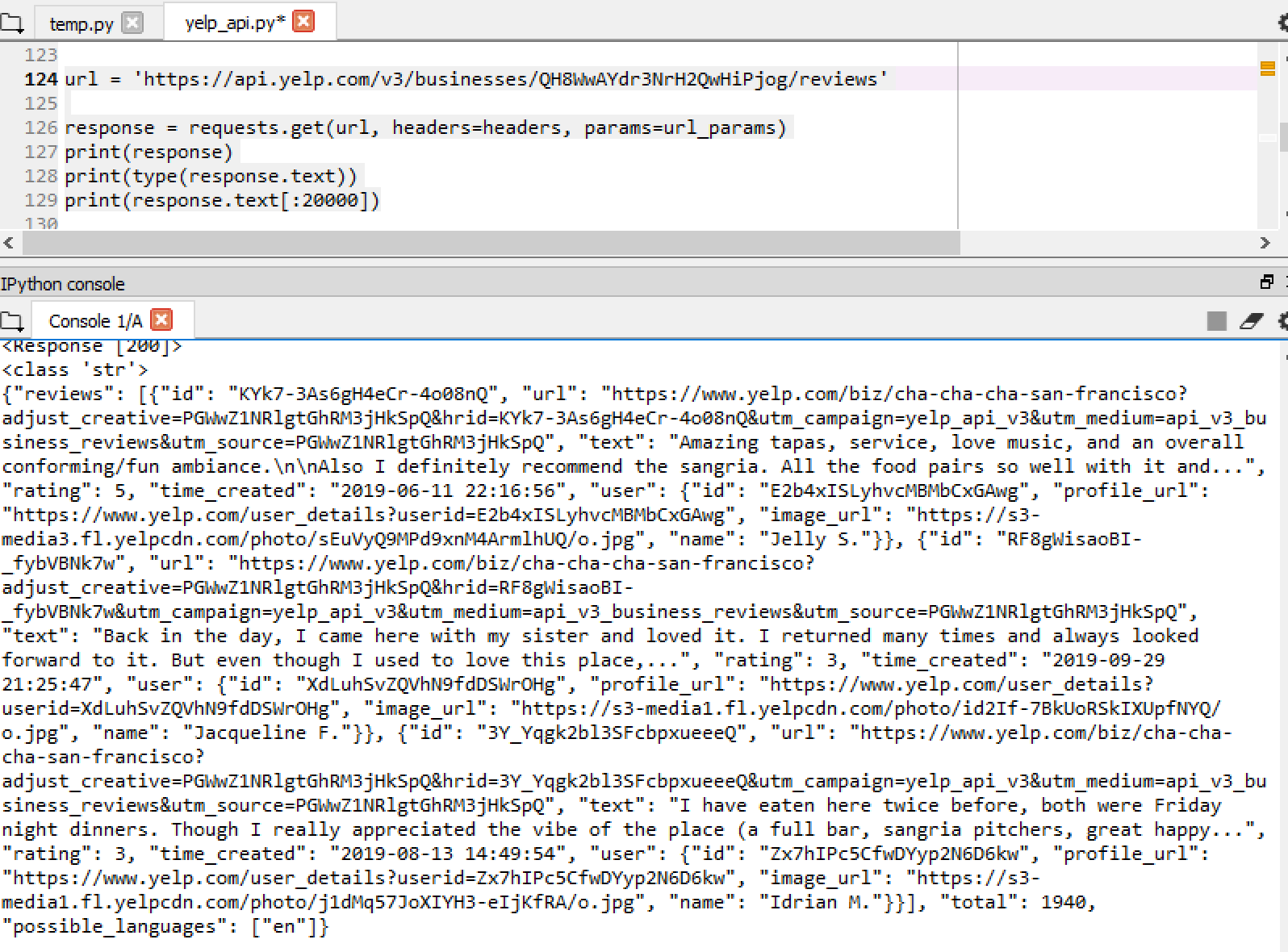
**Analysis and Models:**

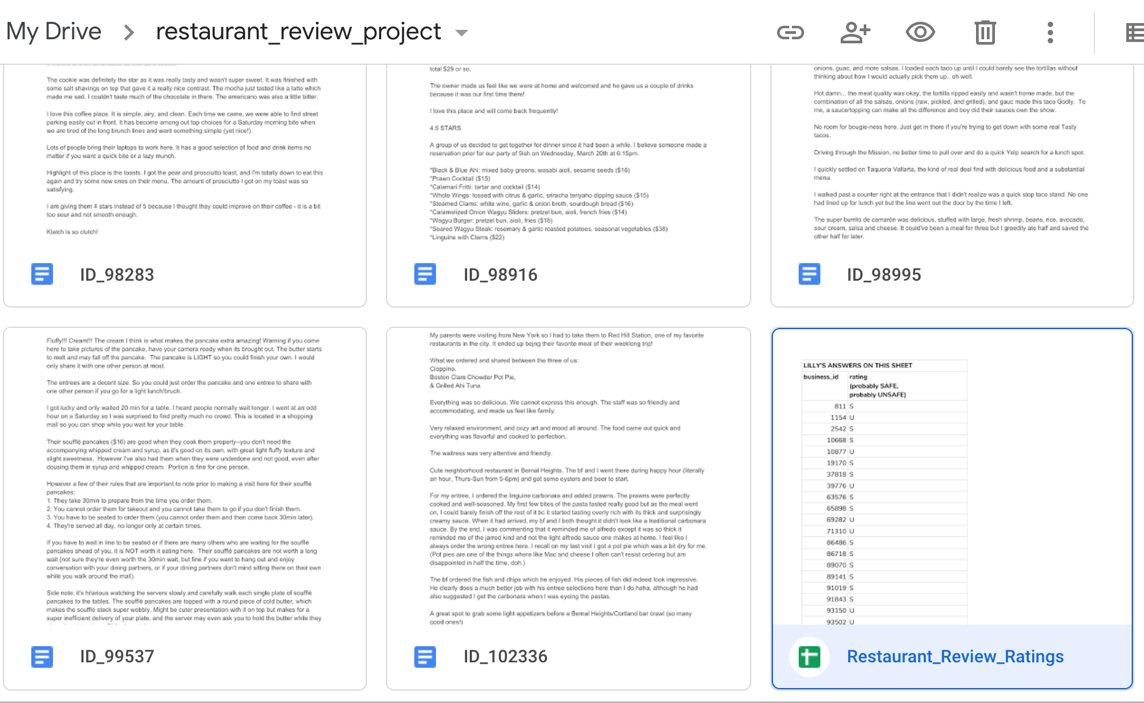
**About the Data:** Two main sources of data were used for this analysis.

Health Score Data: First, a data set was downloaded from the city of San Francisco that lists all of the restaurant health inspection scores from the last three years, with detailed information about the inspection. Each inspection was given a score, from 1 to 100, as well as a “high” “medium” or “low” risk category. Most restaurants have multiple records representing multiple inspections over time or multiple violations on the same day.



Yelp Review Data: Yelp restaurant reviews are available to query via API. However, the API only returns up to three review excerpts for a given business ordered by Yelp's default sort order. In order to create a large enough corpus to facilitate manual annotation, and serve as a base for training a model in the future, reviews were simply downloaded manually for a small number of restaurants. Only the first page of reviews was saved to make the work a manageable size, and the reviews were combined into one text file for each of the restaurants in the selected sample. A summary spreadsheet was also created and all files were posted to google docs.





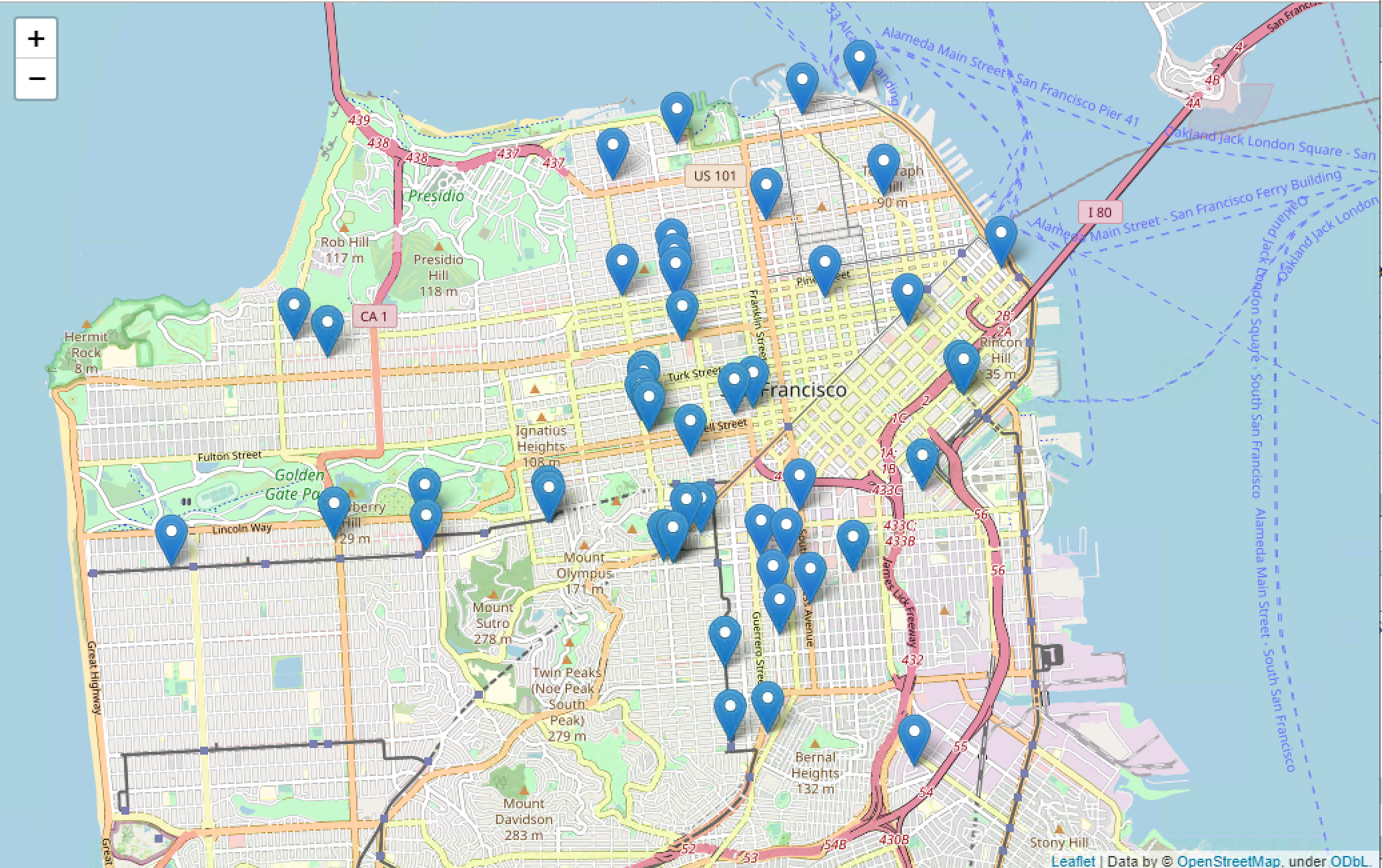
Wordclouds were used to explore the data and determine whether a clear pattern of “safe” versus “unsafe” restaurants could be identified. Unfortunately, trying to identify restaurant safety simply by looking at the most frequent words is a tremendous challenge.



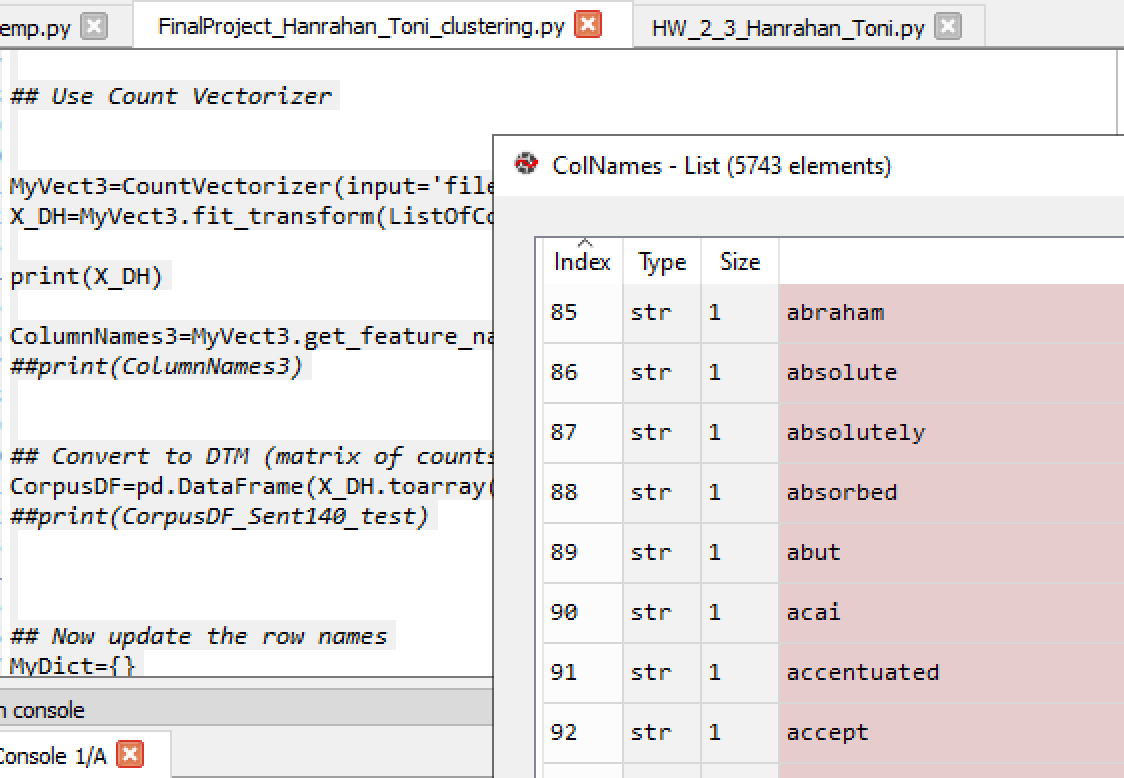
Both of these restaurants are 4 stars. Which one is safer?

Reading reviews is not always enough to determine possible safety issues.

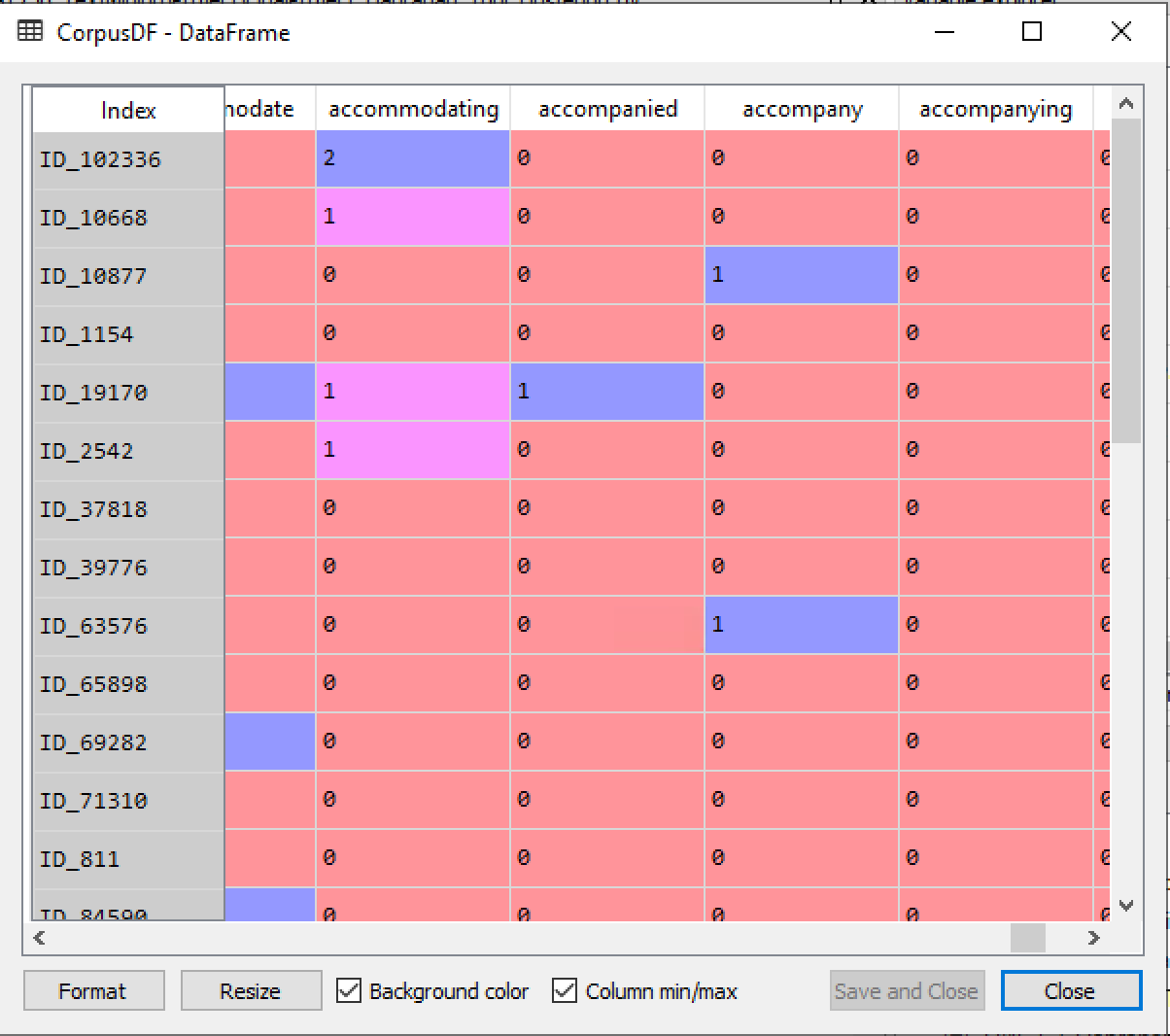
In addition, maps were created in order to explore whether there was any geographic connection between safe or unsafe restaurants. However, the distribution is quite dense and no clear pattern was detectable.



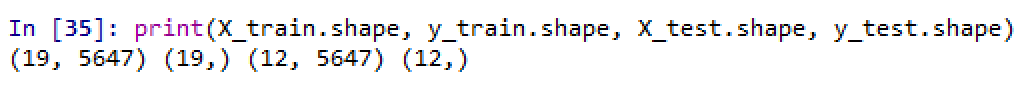
The review data was saved as individual text files, and stored in the same folder as a corpus. Next, the count vectorizer function was used which tokenizes the text and count the number of times each word appears.



Stop words were removed and then the dataframe was converted into a matrix.

****

The data was spilt into test and training datasets – 60% for training and 40% for testing.

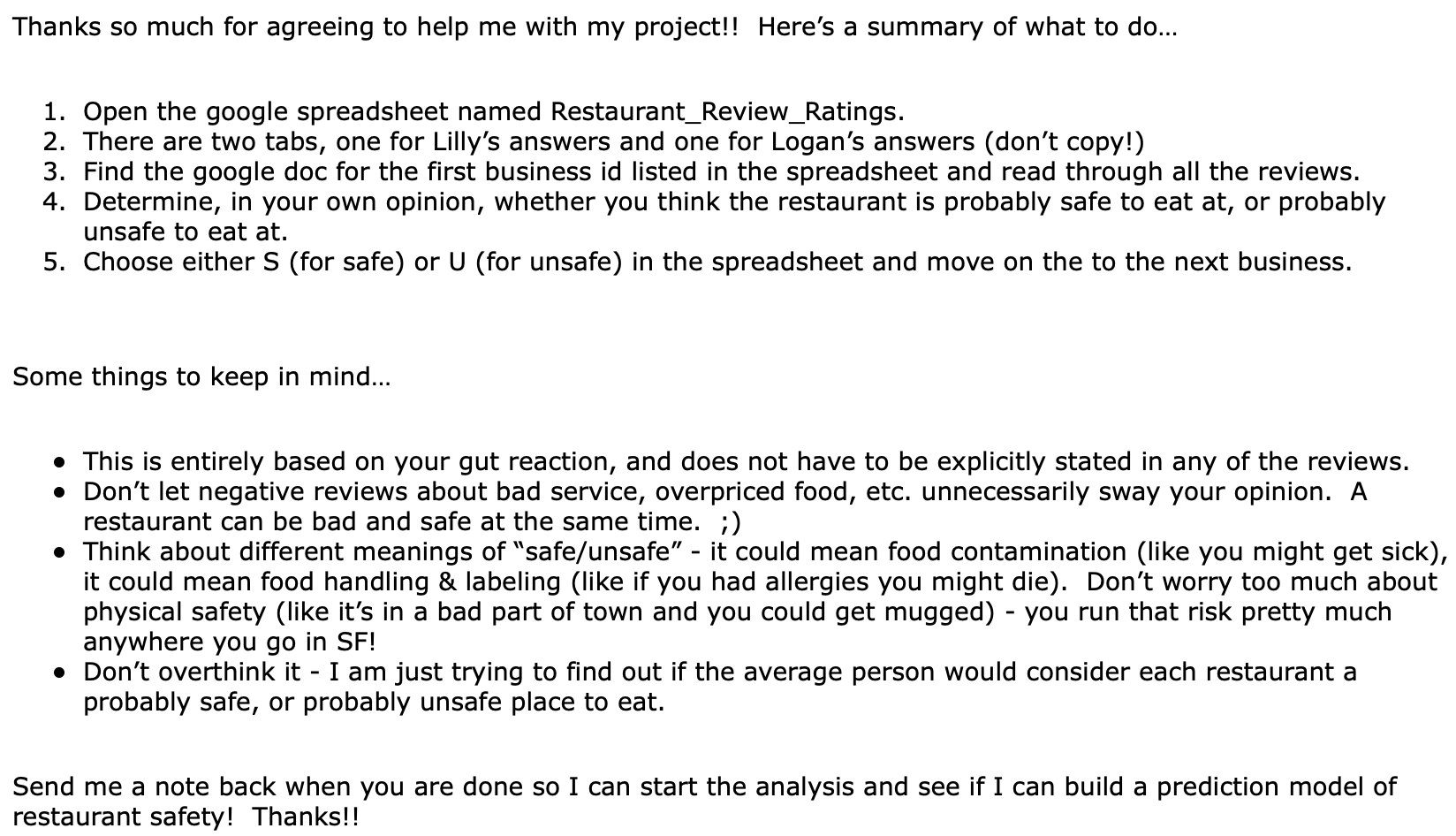
****

**About the Models:**

Health Score Data: In order to identify high risk and low risk restaurants, the scores of the individual inspection ratings were averaged. This is a slightly different methodology than the way the scores are displayed on Yelp. Yelp simply displays the most current inspection score. However, as the review data is being combined over time, not merely the most recent, it made more sense to apply a similar methodology to the score data and create an average score. Inspections with no score attached, such as a follow up when ownership changes, were removed from the data. Fifteen restaurants with very negative average scores, and sixteen restaurants with very positive scores were selected.

Yelp Review Data: Human workers were needed to read the set of reviews and use their world knowledge and human intelligence to determine whether the restaurant could be identified as “probably safe” or “probably unsafe”. The data set was relatively small, so rather than using the Amazon Mechanical Turker service, local resources were recruited to perform this work instead. (Namely my son and his boyfriend). This guaranteed that the workers had the skills needed to perform the task and ensured quality control. It was estimated that the entire project would take about 2 hours to complete, and payment of a $25 Steam gift card was negotiated as sufficient compensation for the task. Very clear instructions were provided, as well as a tracking spreadsheet to record the labels.

Instructions:



While both “turkers” appeared to be more positively biased, their bias seemed very similar. These results were not unexpected as it has already been established that despite poor health scores, customers still write positive reviews. Since the bias was similar across both workers, this was an acceptable result.



Kappa score = 0.162162162162162



*kMeans cluster algorithm*

Details: Per the sklearn documentation: “The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares (see below).” <https://scikit-learn.org/stable/modules/clustering.html#k-means>

Multinomial Naive Bayes model

Details: Per the sklearn documentation: “the multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification)”. <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>

Support Vector Machine classification model

Details: Per the sklearn documentation: “Similar to SVC with parameter kernel=’linear’, but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.” <http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC>

**Results:**

*Turkers*

The “turkers” results appeared, on the surface, to be very consistent and reliable. There seemed to be agreement between the workers responses as many of the reviews were labeled similarly. However, the Kappa score ended up being rather low (0.16). There are many opinions of what a “good” Kappa score should be. The most common recommendation is that values ≤ 0 indicate no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement. This could be due to the small sample size.

Overall, using locally sourced “turkers” to label the restaurant reviews was an effective approach. The results came back in a relatively short period of time, and the cost was reasonable. The results were as expected and this data is now prepped and ready to train a model.

*kMeans cluster algorithm:*

The corpus was analyzed using the kMeans cluster algorithm. Interestingly, the Kmeans did NOT cluster particularly well. The “Safe” files and the “Unsafe” files should have been clustered with each other, but it was only marginally accurate at 54.84%.

****

MNB *– using health inspection labels*:

****

SVM *– using health inspection labels*:

****

MNB *– using Turker labels*:

****

SVM *– using Turker labels*:

****

Comparison of all models:

|  |  |
| --- | --- |
| **MODEL** |  |
| KMEANS | 0.55 |
| MNB - using inspection labels | 0.75 |
| SVM - using inspection labels | 0.67 |
| MNB - using Turker labels | 0.42 |
| SVM - using Turker labels | 0.42 |

Both models using the health inspection labels had higher accuracy than the same models using the labels provided by the Turkers. Because the Turker labels introduced a third label – “N” (for no match), the samples that were used for the model may not have been large enough to provide sufficient data for the models to learn from. In general, the MNB model outperformed the SVM model, however both MNB and SVM (using inspection labels) outperformed kMeans Clustering.

**Conclusions:**

Reading hundreds and hundreds of restaurant reviews, in order to find a safe place to eat, is not a practical task for the millions of San Francisco restaurant-goers. However, machine learning techniques can assist. Once an initial set of reviews is read by several human workers, those ratings can be consumed by a machine learning process in order to predict restaurants that may be unsafe.

A restaurant safety prediction score could be a better approach than simply relying on a star rating system. Visitors and tourists who may not be familiar with the area, need an efficient and easy way to find a good and safe place to eat. Developing a model to predict unsafe restaurants can help consumers, as well as inform health inspectors on potential “hot spots” to prioritize their resources.