Yelp Restaurant Reviews Classification and Topics Generation

URL: https://www.kaggle.com/yelp-dataset/yelp-dataset
Dataset Documentation: https://www.yelp.com/dataset/documentation/main

Problem Statement

Background:

Customers usually leave feedback and reviews on food, service, buying experience and other aspects of the establishment after they visit restaurant(s). In order to stay competitive, it is important for the restaurants to understand and analyze factors that will help them to build a customer friendly image. Many restaurants spend considerable funds in promotional activities and understanding the factors that can improve their competitive edge. However, they can get most of these insights by leveraging the direct reviews and ratings left by prior customers. Most restaurants find it a challenge to derive insights from these reviews because of following 3 reasons. One, There are many reviews available for each restaurants. It would be time consuming and expensive to manually scan all these reviews. Two, Through the ratings in a scale 1-5 provide indication on sentiment of reviews, the real insights lie in the free-form text review. Without leveraging technology, it is challenging to make sense from numerous free-form text reviews. Three, As humans are subjective, their subjective judgements sometime get into their reviews, thus making it difficult to derive the sentiment and the ratings.

In this project, I intend to build a restaurant review classification model that can be used for predicting sentiment of a review. There will also be insights on important phrases that appear in positive as well as negative customer reviews. These insights can be used by restaurants to understand what they need to continue doing (takeaways from the positive reviews) and what they need to improve upon(takeaways from the negative reviews). In addition, I will present a framework that would help to find important topics from all reviews of a specific restaurant. My hope is that the insights from all restaurant reviews and the generated topics from individual restaurant reviews can help a restaurant to be liked by its customers on a continuous basis.

I used the Yelp dataset in Kaggle for the project. The dataset contains information about businesses across 11 metropolitan areas in four countries. For this project, I will consider only the restaurant reviews in scope.

Objective
I will extract and analyze the customer reviews of the restaurants with the following goals

- Build a classifier to correctly classify the sentiment (positive or negative) of customer reviews.
- Present the top phrases that appear in positive reviews and negative reviews respectively.
- Build a framework to analyze both positive and negative reviews of a restaurant and present important topics from them.

Customer/ Target Audience:
The project was presented for restaurants to derive insights from unstructured customer reviews. Nonetheless, the solution and the approach can be easily extended to any other business establishments with the following goals

- Improve customer experience/interaction based on their reviews
- Understand important phrases and key topics from the customers' text reviews.
- Predict whether a text review has positive or negative sentiment.

Data Wrangling and Exploratory Data Analysis

For my project, I used the datasets on business and reviews available in the Yelp dataset. The documentation on the dataset is available here: https://www.yelp.com/dataset/documentation/main. Data elements in both the business and review datasets were available in JSON format. I started with previewing few rows in both Business and Reviews dataset. I had the following insights regarding the fields of interest.

- Business Dataset: Each Business has one unique business_id. A category field mentions all categories (single or multiple values) associated with the business.
- Review Dataset: Each review has a unique review_id. Review has a business_id field that links the review to the business it is about.

Business Dataset

My first challenge was to identify what a restaurant really is. As per Wikipedia[1], *a restaurant or an eatery is a business which prepares and serves food and drinks to cusotmers for money. Meals are generally served and eaten on the premises, but many restaurants also offer take-out and food delivery services. Restaurants vary greatly in appearance and offerings, including a wide variety of cuisines and service models ranging from inexpensive fast food restaurants and cafeterias, to mid-priced family restaurants, to high-priced luxury establishments.* For my project, I decided to consider the restaurants where Food is generally served and eaten on the premises.

While previewing the business categories, I discovered there was a Restaurants category. In addition, there were also many other category that might qualify as restaurants. In order to find the most prevalent categories, I built a word cloud from the categories as shown in figure 1. It is evident from the word cloud that there are a good number of restaurants among the business establishments.

Figure 1: Word Cloud from all categories based on Frequency of occurrence.

While previewing the categories, I came across few establishments that are not associated with Restaurant category but are close (Example: Bakery, Food). In order to understand these alternate categories better, I visualized the categories of establishments that are not associated with restaurant but are associated with food as shown below. From the below visualization, I chose two additional categories: (1) Bakeries and (2) Juice Bars as I am focusing on Restaurants/establishments where Food is generally served and eaten on the premises. Customers usually expect a high level of service at these establishments. The establishments serving ice creams, yogurts or tea/coffee were not considered because they can be considered as destinations for quick eats or beverages.

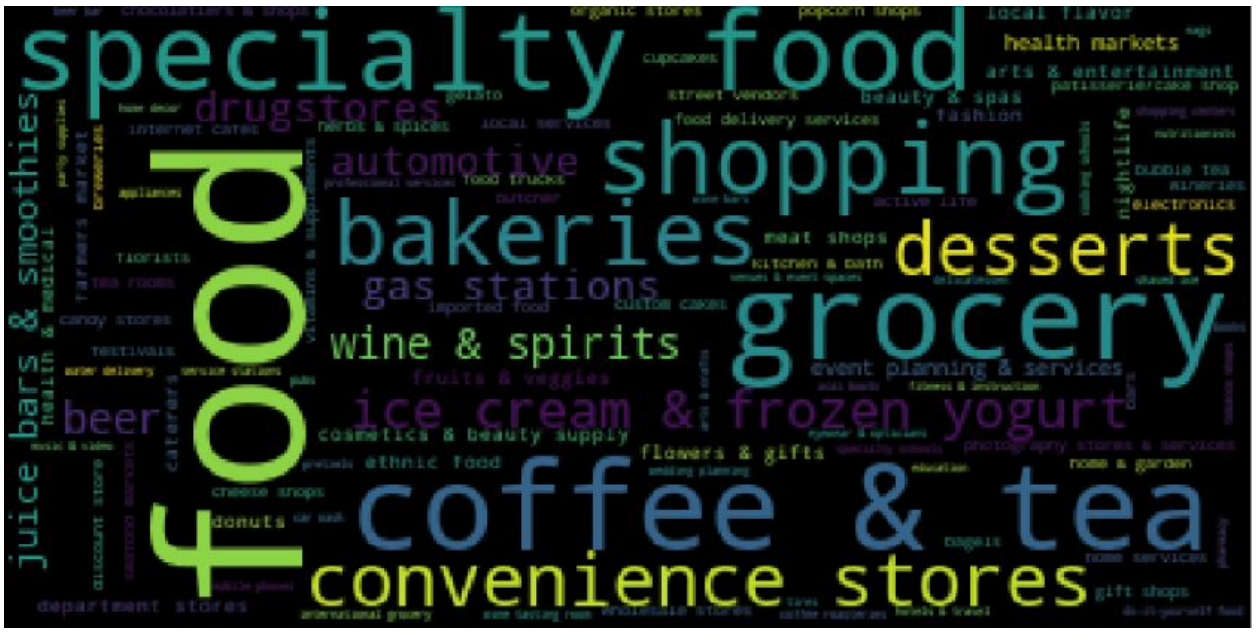

Figure 2: Word cloud of business categories (Condition: Restaurant not a category, Food is a Category)

Hence, I considered the establishments with one of the following categories in scope of this project. I collected all the business_ids associated with above 3 categories into a set and converted it to a frozenset as I don't anticipate the set to change.

Categories in Scope

- Restaurants,
- Bakeries or
- Juice Bars

Review Dataset

For our project, we considered the following two columns from the reviews dataset.

- Stars: 1-5 Range.
- Text: The Actual text review

I decided to classify the reviews based on their stars as follows.

Stars	Rating of review
1, 2	Negative
3	Neutral
4,5	Positive

Table 1: Reviews Stars/Ratings to Sentiment Mapping

I built the following two datasets

- **Dataset 1 for Classification (All Restaurants)**:
  File Name: yelp_restaurants_reviews_classification_rev.txt
  The first dataset was created from both positive and negative reviews (associated with ratings 1, 2, 4, 5). Neutral Reviews with rating 3 were excluded from this dataset. First 10, 000 reviews meeting the above criteria were collected to build the dataset. It was used for following
  - Building a classifier to accurately predict sentiment of a restaurant review
  - Generate top words from positive and negative reviews respectively.

- **Dataset 2 for Insights generation (Individual restaurant):**
  File Name: yelp_restaurants_reviews_topic_recognition_rev.txt
  Next, I built the 2nd dataset collecting the positive and negative reviews of a specific Seafood restaurant. The dataset was used for deriving personalized recommendation for the restaurant from its reviews through top words and topics generation.

The project showcased how insights could be generated at both the summary level (e.g. all restaurants) and individual level (e.g. one restaurant level).

Building the Restaurants Review Classifier

Exploratory Data Analysis

As a first step, I checked the frequency of the rating variable. As shown in below figure, the positive ratings account for around 76% of all the reviews and the negative ratings constitute the rest 24%. So, the principles associated imbalanced class variables would be applicable here.

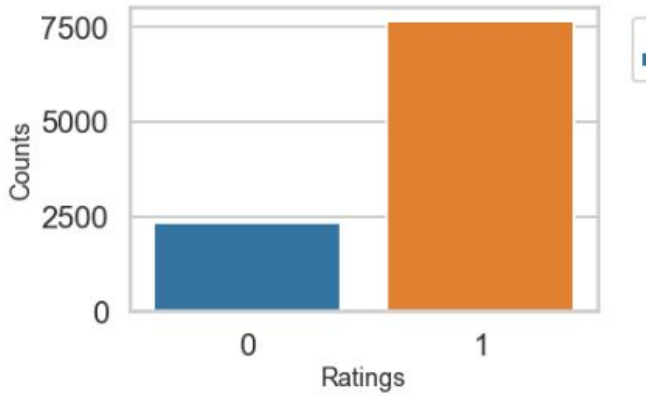

Figure 3: Counts by Ratings

As text data is usually free-form, it requires special techniques to process the data before it can be used with a machine learning algorithm. The usual steps while preparing the text data is as follows

- **Tokenization**: Parse the texts to collect the words (e.g. tokens)
- **Filtering**: Based on the language features, we may decide to remove specific words so that they do not influence the models (e.g. Stop-words or Frequently used words).
- **Stemming or Lemmatization**: By stemming or Lemmatization, the words are converted to their roots or stems. Lemmatization is a better choice for legibility because the lemmas are full words whereas stems may be part of a word (example: The stem of running is runn after removing the -ing phrase. Lemma of running is run).
- **Vectorization**: Map the tokens to integers and capture specific aggregations on the token. For example, Count Vectorizer captures the count of each token in a vector form. This step is necessary because many machine learning algorithms take data structures based on numerical data as input.

Baseline Classification Model

First, I built a baseline classifier using a count vectorizer on unigrams and naive bayes algorithm. I chose Naive Bayes for the base model because it is computationally fast, simple to implement and works well on high dimensional data (e.g. text). I followed the below steps to build and validate the base classifier.

- The text restaurant reviews were input into a count vectorizer to form the feature vectors.
- The feature vectors were fit using a naive bayes classifier. As there are many more positive reviews than negative reviews (imbalanced class problem), I used SMOTE to upsample the training dataset.
- The hyperparameters (Min_df for CountVectorizer and Alpha for Naive Bayes algorithm) were tuned through CVGridSearch based on best accuracy score.

- The, the final base model was built using the best hyperparameters on the complete training dataset.
- The performance of the best model on the unseen test dataset was found out.

The block diagram of the base classifier and the performance accuracy have been shown below.

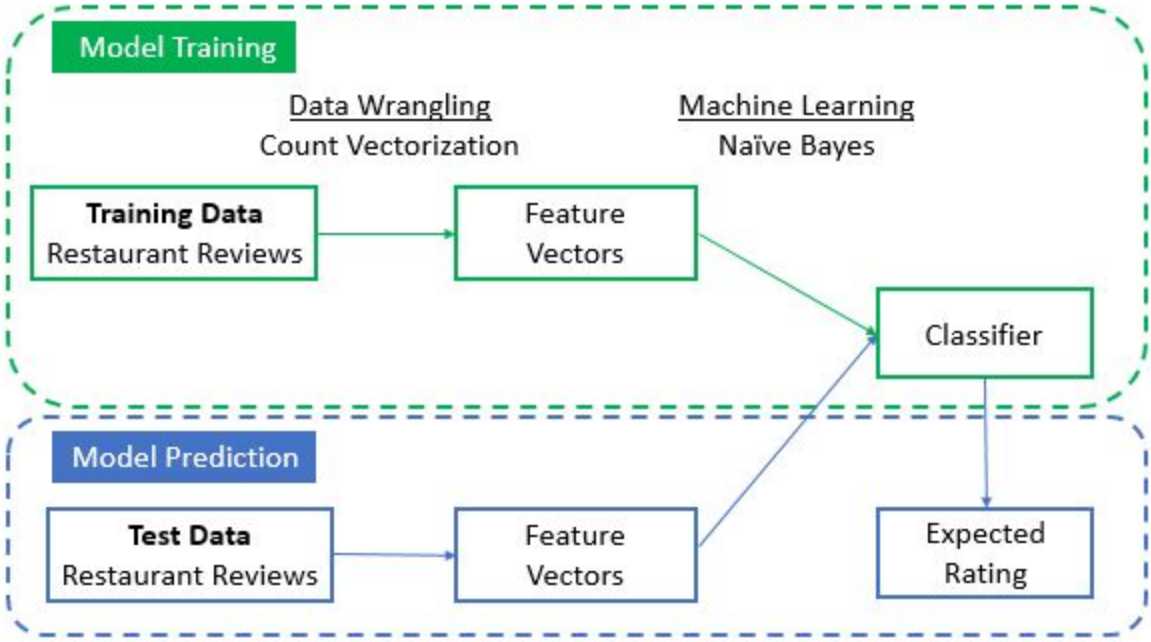

Figure 4: Block Diagram of the base model

Model	**Best Hyperparameters**	**CV Accuracy (Training Set)**	**Accuracy (Test Set)**
CV + NV (Base)	CV (Min_df: 0.0001) NV(Alpha: 1)	91.98%	90.77%

Table 2: Base Model Performance

Legend:

- Countvectorizer: CV
- Naive Bayes: NV

Model Results Analysis

- As noticed, the performance of the model on the training dataset and the test dataset were not very different.
- Based on the confusion matrix, 173 negative reviews were classified as positive reviews whereas 104 positive reviews were classified as negative.

		Predicted	
		0 (Negative)	1(Positive)
Actual	0 (Negative)	552	173
	1 (Positive)	104	2171

Figure 5: Confusion Matrix

Words with Positive Sentiment	
Words	% Positive
gem heaven donut feta delish	98
phenomenal refreshing art disappoint pleasure pleasantly	97

Words with Negative Sentiment	
Words	% Positive
unprofessional	02
downhill tasteless	03
horrible filthy	04
worse unfriendly rudely awful miserable worst disgusting Refund	05

Figure 6: Top Words with Positive and Negative Sentiments

- Next, I found the top words from both positive reviews. Most of the top words (gem, heaven, phenomenal, refreshing,pleasantly etc) were self-explanatory as these words were associated with high positive sentiments. However, there were also other frequently used words that are typically not associated with positive sentiments (example: donut, art, disappoint etc). In subsequent steps, I would **include bigrams and trigrams** so that the language context of the words won't be lost. I would also lemmatize the review text so that all derived words related to a word would be classified as a lemma (complete english word). Currently, I found stems (e.g. feta, delish) that did not make much sense.
- The top words from the negative reviews did include words with strong negative connotations. Considering bigrams and trigrams would also ensure that the language context of these words are preserved.
- I would also consider other classifiers (logistic regression and random forest) because Naive Bayes assumes inter-independence among features which is typically not true for text data. For example, my model considered the word *'disappoint'* as a word with a positive connotation,

whereas it is actually a word with negative connotations. It would be important to understand the language related contexts in the reviews.

- After looking at a few mispredicted reviews, I realized that the correct prediction of these reviews would require the model to understand the context of the language. Here are two examples of (i) A positive review predicted as negative and (ii) A negative review predicted as positive.

Example: Positive Review Predicted as Negative
Really good food, but beware the portions are huge!!! Was full off of the truffle fries alone with brisket and French dip to follow, didn't even finish half of the later 2 and ended up having to throw away leftovers due to hotel not having a refrigerator...only complaint was the long wait almost an hour just to find out they hadn't put our name down on the list or failed to call us when we were waiting right outside the entrance, luckily they got us a table immediately upon discovering the ...

Example: Negative Review Predicted as Positive
I got ramen noodle yesterday for lunch, but noodle became brown and got moldy.\nI like Japanese snack but they don't carry enough anymore. Only 1good thing they open till 11:00pm.

An improved Classifier on TF-IDF Vectorizer and Naive Bayes

In the next model, I incorporated the above mentioned improvements. The following steps were followed for data wrangling and modeling

- The input reviews were transformed through:
    - Tokenization: Each word in the review was converted to a token.
    - Stop-words Removal: The stop-words in english language as defined in nltk library were removed.
    - POS with Lemmatization: The tokens were lemmatized to their original lemmas so that words with various tense were considered as one.
    - Finally, the tokens were appended together to form the lemmatized review.

  Original Review:
  Went in for a lunch. Steak sandwich was delicious, and the Caesar salad had an absolutely delicious dressing, with a perfect amount of dressing, and distributed perfectly across each leaf. I know I'm going on about the salad ... But it was perfect.\n\nDrink prices were pretty good.\n\nThe Server, Dawn, was friendly and accommodating. Very happy with her.\n\nIn summation, a great pub experience. Would go again!

  Lemmatized Review
  go lunch steak sandwich delicious caesar salad absolutely delicious dress perfect amount dress distribute perfectly across leaf know go salad perfect drink price pretty good server dawn friendly accommodate happy summation great pub experience would go

- The lemmatized reviews were fed through a TF-IDF Vectorizer to form the feature vectors. TF-IDF is superior algorithm than the Count Vectorizer because the TF-IDF looks at both term frequency as well as document frequency of phrases. The algorithm weighs the unique words in documents high and weighs common words across documents low. Thus, it presents an improved representation of term importance in the context of a document.

- I used SMOTE to upsample the minority class.
- The feature vectors were fit using a naive bayes classifier.
- The hyperparameters (Min_df for CountVectorizer and Alpha for Naive Bayes algorithm) were tuned through CVGridSearch.
- The model with the best hyper parameters were drawn on the entire training set.
- The performance of the model was found out over the test dataset.

Classifiers using TF-IDF Vectorizer and Logistics Regression/ Random Forest

Afterwards, I built another classifier using (1) TF-IDF vectorizer + Logistics Regression and (2) TF-IDF Vectorizer + Random Forest. I chose these two machine learning algorithm alternatives because a Random forest is an ensemble of many decision trees that works well on correlated data. It is also easy to find importance of various features and the model has good explainability. Logistics Regression is good at presenting the relationship between the input and output, however it may suffer from overfitting.

A summary of performance of these models has been shown in below table

Model	**Best Hyperparameters**	**CV Accuracy (Training Set)**	**Accuracy (Test Set)**	**F1 Score (Test Set)**
CV + Naive Bayes (Base)	CV (Min_df: 0.0001) NV(Alpha: 1)	91.98%	90.77%	94%
TF-IDF + Naive Bayes	CV (Min_df: 0.001) NV(Alpha: 1)	91.15%	90.27%	93.44%
TF-IDF + Log Regression	C=100	93.1%	92.23%	94.87%
TF-IDF + Random Forest	Max_depth : 10, n_estimators: 100	83.73%	83.97%	89.76%

Analysis on Model Performance

		Predicted	
		0 (Negative)	1(Positive)
Actual	0 (Negative)	628	97
	1 (Positive)	195	2080

Figure 7: Confusion Matrix (TF-IDF Vectorizer + Naive Bayes)

Words with Positive Sentiment	
Words	% Positive
highly recommended	96
gem	95
delish phenomenal	94
friendly staff great food great	93
delicious perfectly hit spot	92

Words with Negative Sentiment	
Words	% Positive
horrible tasteless awful	03
disgust worst avoid place worse	04
horrible service worst service poor roach rude	05

Figure 8: Top Words with Positive and Negative Sentiments

- The top-words from the TF-IDF vectorizer (with trigrams) were more informative when compared with Countvectorizer Unigram features. Thus, the model was able to keep association between words (e.g. friendly staff, horrible service etc).
- The TF-IDF Vectorizer with Naive Bayes algorithm was better at classifying negative reviews.
- The performance of all the classifiers were close with the classifier built from Logistics Regression performed the best.
- The difference between the models will be more pronounced when we will consider more training and test data for the model.

Insights Generation from Reviews of one restaurant

The 2nd part of the project demonstrated top words and topics generation from positive and negative reviews of a restaurant. For this use case, I considered a seafood restaurant at random that had more than 1000 reviews. The techniques mentioned in this section can be used for any other restaurant of choice as well.

I chose all the business_ids with categories (1) Restaurants and (2) SeaFood. I visualized the businesses based on the number of associated reviews and selected one business_id at random (ID: yNPh5SO-7wr8HPpVCDPbXQ) that had significant number of reviews (> 1000 reviews). All positive (starts: 4,5) and negative (1,2) reviews associated with the above business_id were stored in a file for further consumption.

I visualized the frequency of the positive and negative reviews and found that this restaurant had 85% positive reviews.

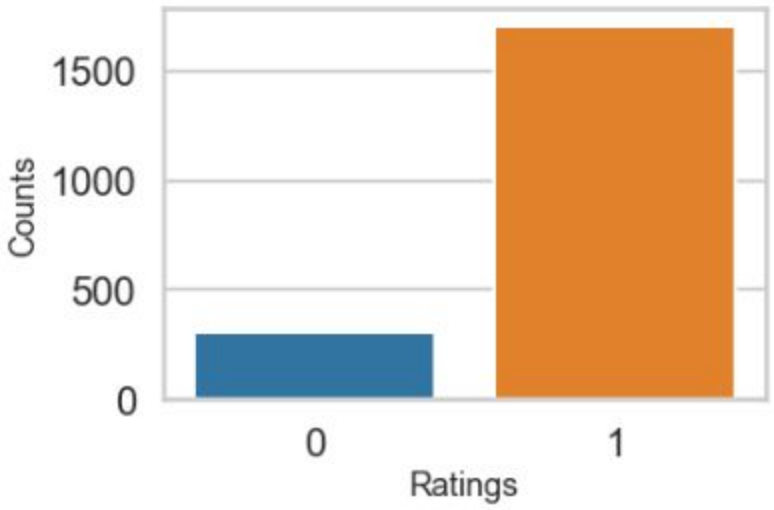

Figure 9: Rating vs Number of Reviews

In this section, I derived the topwords and the topics for positive reviews and negative reviews separately through two methods LDA (Latent Dirichlet Allocation) and NMF (Non-Negative Matrix Factorization). Both LDA as well as NMF are used for topic modelling, an unsupervised learning approach to clustering documents and to discover topics based on their contents. The NMF (Non-Negative Matrix Factorization) is a linear algebraic model and converts high dimensional vectors into low-dimensional representations. I used a TF-IDF vectorizer with NMF. The LDA is a probabilistic model and is predominantly used with Term Frequency Vectorizer.

First, I started with the topic modeling for the Positive Reviews based on the following steps

- I converted the reviews text to lemmatized text using stopwords removal and lemmatization with POS.

- I chose appropriate values of min_df, max_df, no_features. For this project, I chose 1000 features as the maximum limit.
- The lemmatized reviews associated with positive ratings were input to Vectorizer in order to create the feature vectors.
- The feature vectors were input to NMF and LDA models separately.
- I chose to display top 2 topics, 4 words per topic and 2 top reviews per topic.
- Then, the topics and the top reviews were displayed.

Insights from the Positive Reviews

The top topics recognized from the positive reviews were as follows

1. good order come really (NMF, Topic 1)
2. great food service place (NMF, Topic 2)
3. great cocktail orange make (LDA, Topic 1)
4. great good food place (LDA, Topic 2)

Here are a few top reviews contributing to these topics

Review 1

We have heard a lot about this place and were finally able to go.. Service was friendly and fast and food was good. We got the jambolaya and the fish and chips which were both really good. The best part of our meal was the dessert.. We got the salted carmel pudding and it was sooo good! We would go back just for that dessert! Everything tasted nice and fresh it was excellent. (NMF, Topic 1)

Review 2

Ask for Chip ... it's really like going to Disneyland, except it's restaurant. Great, knowledgable service, good food ... fun times (LDA, Topic 2)

Interpretation

From the above topics and reviews, the restaurant can get important insights on what customers are liking about it. For example, few emerging insights from the customer review are as follows.

- Good Food and Service
- Great Cocktail
- Good Place for food

Most customers find the good and service at the restaurant good. The restaurant should continuously thrive to maintain this high quality of food and service. People also praised great cocktails at the restaurant. Hence, the restaurant should continue to serve great cocktails in order to gain the goodwill of the customers.

Insights from the Negative Reviews

Following the similar methods as described above, I derived the insights from the negative reviews as well.

The top topics recognized from the negative reviews were as follows

1. food good place service (NMF, Topic 1)
2. Wait minutes table seat (NMF, Topic 2)

3. food good time service (LDA, Topic 1)
4. food good order place (LDA, Topic 2)

Here are a few top reviews contributing to these topics

Review 1
We visited this past weekend. It took on hour to receive our meal and two of the dishes were cold. The waiter was bad but gave us free dessert to make up for the food and service. The peanut butter cup gelato was inedible. Tiny scoop of gelato with peanut butter slop on top. Not sure if the restaurant just had a staffing issue or what but we'd never go back. (NMF, Topic 2)

Review 2
My friend told me you should never order seafood in the desert. Now I know why, because my salmon was so overcooked and dry that I had to ask for extra lemons just to make the taste bearable. My colleagues had burgers and a kale bowl and only 2/6 of us enjoyed our dinner.

Maybe it was an off night during our visit, but our experience was really disappointing. Although our server tried to be nice about the slow service, he did not really do anything to make up for it. (LDA, Topic 1)

Interpretation

From the above topics and reviews, the restaurant can get important insights on what customers did not like at the restaurant. For example, few emerging insights from the customer review are as follows.

- Good Food and Good Service, but Long Waiting time at tables, Cold Food
- Slow Service
- Expectation of Free Goodies in case of a bad experience

Based on the above insights, people typically like the food and the place but have complained about (1) long wait time at tables, (2) slow service. The restaurant should check whether they are adequately staffed to take care of the patrons. The management should consider hiring more staff in case the long waiting time is due to short-staffing.

Conclusion and Future Work

In this project, I demonstrated insights generation from restaurant reviews that can be highly beneficial for the restaurants to understand the customer feedback. I build text classifiers to accurately classify the sentiments of restaurant reviews. The classifier based on TF-IDF and Logistic Regression had the best performance (92.2% accuracy). Then, I modelled topics from positive and negative reviews of a restaurant and interpreted them so provide restaurant specific feedback.

In the future, I would implement the project using deep learning so that more reviews can be considered for insights generation and determination of the best classifier.