Landon Smith

Ayushi Singhal

**Dal Rae Restaurant**

**Customer Review Actionability Analysis**

**Introduction**

A common practice by restaurant management is analyzing customer reviews posted on websites like OpenTable and Yelp in order to gain insights into how to improve their business practices. Customer reviews have the potential to offer feedback to restaurant management on a wide variety of topics, such as quality of food and service, ambience, and countless others. According to the 2022 Local Consumer Review Survey, 98% of consumers look at restaurant/cafe reviews before deciding where to eat, meaning restaurants have a vested interest in improving their operations to improve the reviews they receive online, thereby increasing customer traffic. However, a large portion of the reviews restaurant management accesses online are far too vague to be actionable by restaurant management due to the reviews lacking specificity. The monotonous task of sifting through reviews that lack constructive content is time intensive for restaurant management and by extension, detracts from the overall productivity of the business. The goal of this project is to create a machine learning model that can parse customer reviews to automate the process of identifying actionable customer reviews, empowering restaurant management with a tool to analyze their customers' reviews with far more time efficiency.

**Source and Description of Datasets**

The dataset utilized in this project will be constructed using web-scraped aggregated reviews of the Dal Rae restaurant in Pico Rivera, California. The platforms we will obtain our data from are OpenTable and Yelp. At the time of writing, the current review counts for each website are 5,275 and 1,785 respectively, bringing our total number of reviews to 7,060. The tools used to web-scrape the reviews are open-source Python libraries called BeautifulSoup and Selenium, while we will be using a library called Pandas for data manipulation. The web-scraped reviews from both OpenTable and Yelp will be stored in a single Pandas dataframe.

Despite aggregating all of the Dal Rae restaurant’s reviews into a single dataframe to train our machine learning model on, we still need to accurately label each review as actionable or not actionable to restaurant management to create a fully-fledged labeled dataset suitable for supervised machine learning. Due to the size of our dataset, labeling all 7,060 reviews manually is not feasible for a group of two individuals. In order to solve this problem and expedite our dataset creation, we have decided to employ the OpenAI API in order to enable the gpt-3.5-turbo LLM to accurately label the restaurant reviews as actionable and not actionable by restaurant management. In order to ensure gpt-3.5-turbo’s efficacy and accuracy at performing such a task, we performed several tests as a group and determined that the LLM is indeed capable of decoding the language used in the reviews as actionable or not actionable. For example, the review “Had reservations at 8 pm, didn’t get seated until 9:30, ate dinner at 11:00 pm and had to sit in the noisy bar area” was able to be determined by gpt-3.5-turbo as actionable because “management can investigate why the reservation wasn't honored on time and consider measures to prevent such delays in the future. Conversely, when gpt-3.5-turbo was given the review “Terrible experience from start to finish outback quality,” the LLM was able to determine that the review is not actionable because “it does not provide any specific details about what aspects of the experience were problematic.” After confirming that gpt-3.5-turbo is capable of performing our desired task of classifying reviews based on their actionability, we created a Python script that feeds each review from our reviews dataframe to the OpenAI API for classification into a binary value of 0 for actionable and 1 for not actionable. The resulting dataset contains two columns, one containing the restaurant reviews themselves, and the other containing the binary classification of the reviews as actionable or not actionable.

While our initial tests on gpt-3.5-turbo’s capabilities yielded positive results, further progression toward the project goal revealed limitations of the OpenAI API as well as the gpt-3.5-turbo LLM model itself. The first limitation we discovered was the issue of the OpenAI API’s rate limit. Similar to Open AI’s native website for interfacing with gpt-3.5-turbo, interacting with the OpenAI API using an API key is free of cost, allowing us to utilize Python code to programmatically query the gpt-3.5-turbo LLM model at no cost. However, the benefit provided by OpenAI of not having to pay by the token to query the model was quickly diminished when we found that the maximum number of queries available to us on a free plan was three per minute. In the context of our project, the imposed rate limit meant that we could only feed the API three reviews per minute via loops, making a full cycle through our 7,060 review dataframe take over 39 hours. In order to bypass this issue, we decided to engineer a loop that instead processed reviews in batches of five, giving us the ability to process 15 reviews per minute and allowing us to fully loop through the entirety of our reviews in just under 8 hours. While the number of reviews per batch was selected arbitrarily and increasing the number of reviews processed per batch could yield a much shorter processing time, we decided not to raise the number of reviews per batch beyond five because of risks associated with gpt-3.5-turbo’s accuracy when evaluating too many reviews within a single query.

The second limitation we encountered during the creation of our dataset was related to the aforementioned topic of gpt-3.5-turbo’s classification accuracy. While our initial tests determined that gpt-3.5-turbo was capable of understanding nuanced forms of speech often used in human speech like sarcasm, the stochasticity of the model wasn’t revealed to us until we performed further experimentation post-review processing. Upon further inspection, when given a prompt to initialize the model and a review to analyze, gpt-3.5-turbo would often be able to categorize review as actionable or not actionable and cite specific topics mentioned within the content of the review that factored into the models decision. However, when looping through our reviews dataset a second time and comparing the two outputs as a measure of accuracy, we were able to see that gpt-3.5-turbo would at times come to different classification conclusions despite the prompt and supplied review being identical. To further investigate this issue, we ran the first 1000 reviews of our dataset through the OpenAI API five separate times and compared each review's classification predictions throughout all five of our trials. When evaluating our results, we observed that the percentage of reviews that were classified as actionable stayed relatively constant, denoted by the column **Actionability (%)** in the following table. However, the percentage of reviews that shared the same classification across all trials continued to decrease as more trials were added, as shown by the column titled **Identical Classification (Cumulative %).**

|  |  |  |
| --- | --- | --- |
| **Trial** | **Actionability (%)** | **Identical Classification (Cumulative %)** |
| Trial 1 | 60.6% | NA |
| Trial 2 | 61.1% | 91.3% |
| Trial 3 | 61% | 87.7% |
| Trial 4 | 61% | 87.7% |
| Trial 5 | 60.9% | 84.3% |

Once the stochasticity of the gpt-3.5-turbo model’s decision making was revealed to us in our identical classification experiment, we decided to create two different datasets for our models to learn from in an effort to improve our final chosen models performance. The first dataset we created was a simple dataframe of all 7,060 reviews accompanied by the actionabilty value assigned to the review by gpt-3.5-turbo, with a 0 in the column meaning that the review was actionable and a 1 in this column signifying that the review was flagged as non-actionable and should be filtered out. However, due to our concern about gpt-3.5-turbo’s stochasticity, we also decided to create a dataset containing the first 1000 reviews that gpt-3.5-turbo identified with the same actionability value five times in a row. As detailed on the above chart, the percentage of reviews where this occurred was 84.3%, meaning that the length of this dataset was 843 observations by extension. We felt that gpt-3.5-turbo’s consistency in classifying these reviews could potentially yield a dataset where we can be much more confident that the actionability labels were in fact correct. The process performed to create this dataset could not be replicated on the full 7,060 review dataset due to time constraints.

**Data Manipulation - Splitting & Encoding**

In order to be able to feed our data into our below-mentioned models of choice, we must first perform data manipulation in the form of splitting and encoding on each of our datasets. Using the sklearn Python library, we split our data into a training and test set, with 20% of each dataset being used to test the performance of our machine learning models. We then needed to encode our textual reviews data to be readable by our machine learning models. To do this, we used the keras Python library and tokenized each word found in each review, assigning each word a unique integer. As a final step, we converted these integers into vectors of 0 and 1, giving us data fully capable of being fed to our machine learning models of choice.

**Experimental Setup - Determination of Review Actionability**

Listed below are the machine learning models and evaluation metrics utilized:

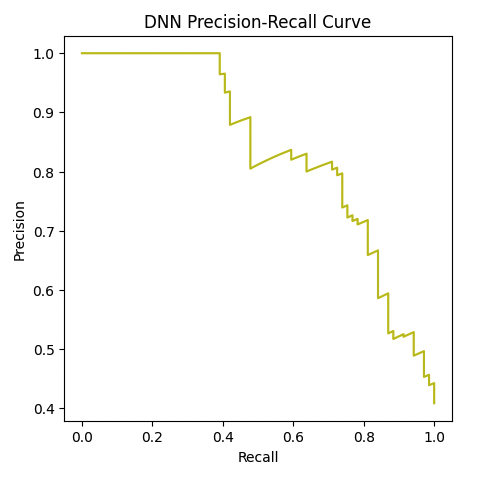
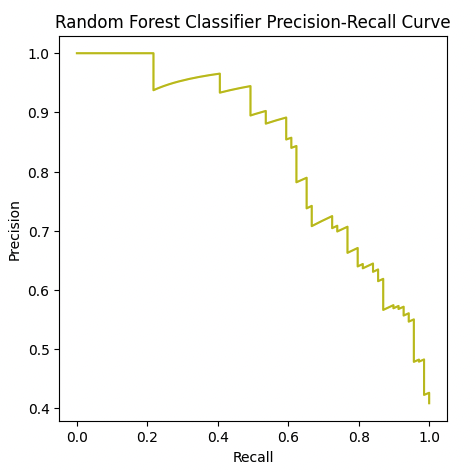
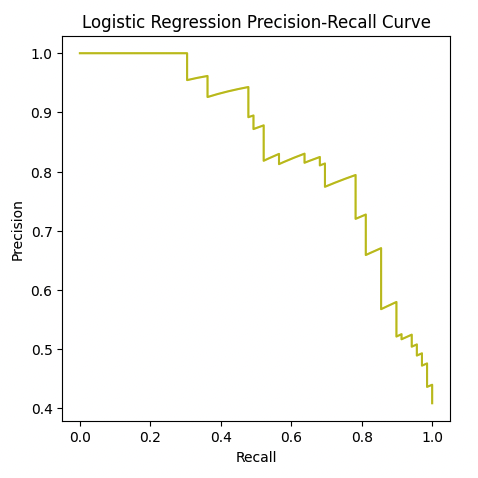
* Creation of Baseline Models: Logistic Regression & Random Forest Classifier
* Deep Neural Network Classification
* Metrics for Evaluation
  + Accuracy
  + Precision
  + Recall
  + F1 Score
  + Confusion Matrix
  + Precision-Recall Curve
  + ROC Curve
  + AUC

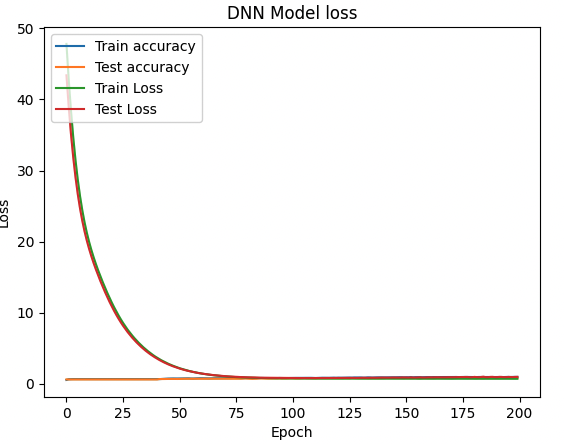
**Results - Determination of Review Actionability Summarization**

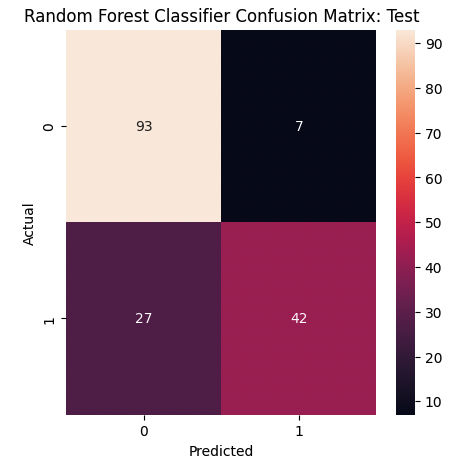
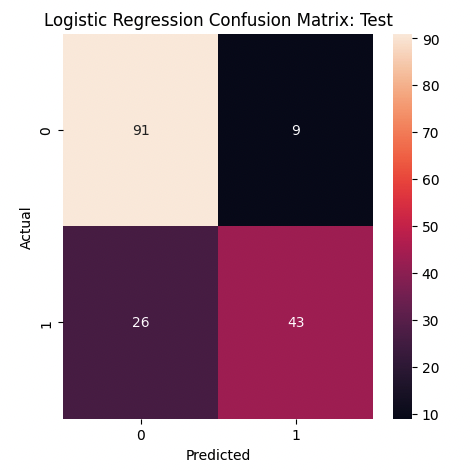
The results of our logistic regression, random forest classification, and deep neural network on our two datasets are listed in tabular format below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **All OpenTable + Yelp Reviews Dataset Summarization Table (n=7060)** | | | | | | |
| Model | Train Accuracy | Test Accuracy | Test Precision | Test Recall | Test F1 Score | Test AUC |
| Logistic Regression | 85.23% | 77.62% | 74.19% | 81.85% | 77.84% | 0.8470 |
| Random Forest Classifier | 87.92% | 77.2% | 74.72% | 79.35% | 76.97% | 0.8394 |
| Deep Neural Network | 99.75% | 84.67% | 48.01% | 100% | 64.88% | 0.8467 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identical Classification Dataset Summarization Table (n=843)** | | | | | | |
| Model | Train Accuracy | Test Accuracy | Test Precision | Test Recall | Test F1 Score | Test AUC |
| Logistic Regression | 90.8% | 79.29% | 82.69% | 62.32% | 71.07% | 0.8604 |
| Random Forest Classifier | 97.18% | 79.29% | 84% | 60.87% | 70.59% | 0.8549 |
| Deep Neural Network | 98.67% | 85.33% | 40.83% | 100% | 57.98% | 0.8526 |





****

**Model Parameters & Results**

The first model we used to determine the actionability of our reviews was a logistic regression model, which served as a baseline to compare the results of our more advanced neural network against. When utilizing this model, we incorporated an L2 Regularization penalty of 0.1 to reduce the model complexity and help us to avoid overfitting and achieve a test accuracy of 79.29%. Our logistic regression model gave us results that matched up almost perfectly with the results of our random forest classification model, which served as our second baseline model. Our random forest classification model allowed us to achieve an identical test accuracy of 79.29% while optimizing our hyperparameters at 1000 for our number of estimators and a maximum tree depth of 17. Finally, we tested the usage of a deep neural network against our baseline models. Our deep neural network consisted of an input layer, four hidden layers, and an output layer. Each hidden layer was composed of 200 neurons, utilized the ReLU activation function, and was subject to L2 regularization at a value of 0.04 and dropout regularization at a value of 0.3. Our output layer utilized a single neuron with an activation function of sigmoid. Our deep neural network was compiled using Adam as our optimizer and our loss defined as binary cross-entropy. Our optimized deep neural network was able to reach a test accuracy of 85.33%, higher than that of our baseline models. However, a major point of differentiation was our DNNs precision and recall metrics at 40.82% and 100% respectively. This precision and recall differentiation severely damaged the F1 Score of our DNN model.

When we consider the 79% accuracy achieved by our baseline models and the 85% accuracy of our fully optimized DNN, we came to the conclusion the most likely underlying cause for the inability to train a model which can achieve a higher accuracy is the aforementioned stochasticity of gpt-3.5-turbo’s review classifications. Despite utilizing our identical classification dataset, which attempted to account for this stochasticity by cross-referencing gpt-3.5-turbo’s outputs across 5 trials, we were likely not able to remove enough of the randomness in order to give the dataset labels that were 100% accurate. Evidence of this can be seen when referring back to the above table which tracks the cumulative percentage of observations that were classified identically. When we conducted two trials, the percentage of reviews that were classified identically was 91.3% of the original dataset, but when five trials were conducted, this percentage drops to 84.3%. We believe that had we continued to execute trials, this trend would have continued, diminishing our number of observations, but increasing the percentage of reviews in the dataset whose actionability labels were truly correct. As a result of this limitation, we view a 85% accuracy as satisfactory.

After creating a DNN model with higher accuracy than our baseline models, we decided that we should optimize the threshold used by the model to make classification decisions to better align with the context of how the model is being utilized by restaurant management. Despite our DNN outputting a 100% recall and 40.82% precision, we came to the conclusion that restaurant management would likely prefer higher precision over recall. In the context of the model usage, the restaurant would likely be more sensitive to incorrectly identifying a review as non-actionable when it is actually actionable than the converse, highlighting precision as the metric of choice. However, while this is our assumption of Dal Rae’s preference, further communication with restaurant management in the future is needed to determine this value.

**Experimental Setup - Review Topic Categorization**

Listed below is the machine learning model and evaluation metrics utilized:

* **Latent Dirichlet Allocation (LDA)**
* Evaluation Metrics
  + Coherence
  + Perplexity

The goal for preparing the topic modeling is to identify underlying themes or topics present in the dataset, which can provide valuable insights into the customers' opinions and preferences. The analysis includes data preprocessing, the application of Latent Dirichlet Allocation (LDA) models, and evaluation of the models based on perplexity and coherence scores.

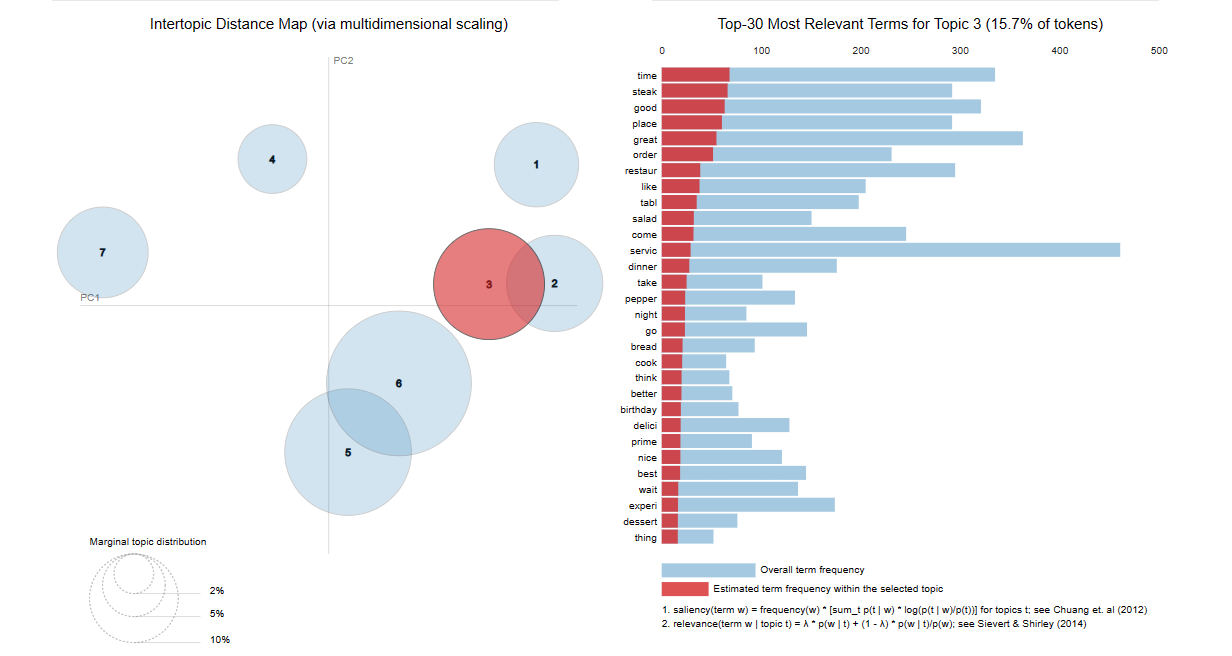
* Data Preprocessing: The dataset is loaded from a CSV file and unwanted columns are removed. Text preprocessing techniques are applied, including the removal of punctuation, conversion to lowercase, and the elimination of single quotes. These steps ensure a clean and standardized text format for further analysis.
* Exploratory Data Analysis: To gain a visual representation of the most frequently occurring words in the reviews, a word cloud and Intertopic distance plot are generated. This visualization helps us understand topics present in the reviews dataset.
* Topic Modeling(LDA): The dataset is transformed into a document-term matrix using the bag-of-words representation. An LDA model is then trained to discover latent topics within the reviews. Two approaches are explored: one using the bag-of-words representation and the other using the TF-IDF representation. The LDA models extract the most relevant keywords for each topic, providing insights into the different themes discussed in the reviews.
* Model Evaluation: Perplexity and coherence ratings are used to evaluate the trained LDA models. The model's capacity to predict unknown data is measured by perplexity, with lower values indicating higher performance. Coherence ratings evaluate the interpretability and semantic coherence of the models' generated themes. So the coherence of both the approaches were compared and the one with bag-of-words was finalized to use for topic modeling upon looking at the coherence value.

**Results - Latent Dirichlet Allocation (LDA)**

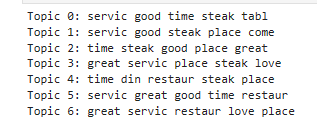
Within the review dataset, the LDA models indicate distinct topics. Each topic is labeled based on the major concept it portrays. The most essential terms related to each theme are highlighted in the top keywords for each topic. subject summaries are also constructed by extracting the top terms from each subject, offering an overview of the dataset's primary themes.



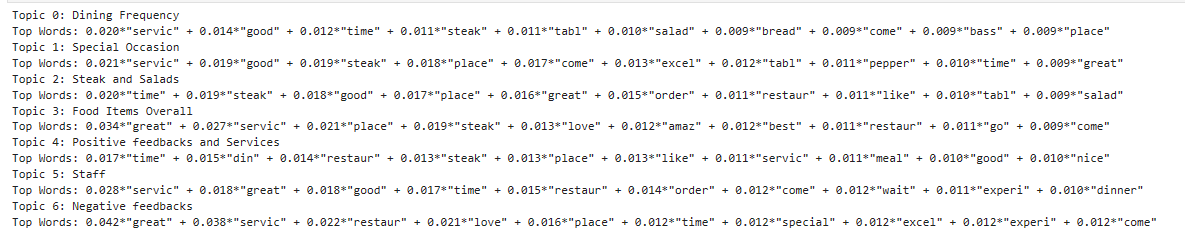
The figure shows the word cloud showing the most frequently used words.



The figure shows the Intertopic distance plot that helps understand the relation and the distance between topics.



Results from Summarizer

****

After applying the topic modeling to the dataset we successfully interpreted and labeled seven topics, where each topic indicates the theme along with their corresponding weights of occurrences. These findings help in understanding the content distribution in each of the analyzed reviews.

**Conclusion**

The goal of the project was to successfully create ML models that will help automate the process of determining actionable customer feedback for restaurant management. Through the web scraped data from multiple open review platforms, we were able to extract a good amount of reviews to train our models. The labeling of reviews to actionable and not actionable was successful with the help of OpenAI API.

Three models were used: Logistic Regression, Random Forest, and a Deep Neural Network (DNN). We used the Logistic Regression model as a baseline and got a test accuracy of 79.29% by using L2 regularization with a penalty of 0.1 to prevent overfitting.

Our second baseline, the Random Forest model, produced identical results with a test accuracy of 79.29%. The hyperparameters were tuned by increasing the number of estimators to 1000 and the maximum tree depth to 17.

We created an architecture for the DNN model that had an input layer, 4 hidden layers with 200 neurons each, and an output layer with a sigmoid activation function. To reduce overfitting, we used L2 regularization with a penalty of 0.02 and dropout regularization with a rate of 0.2. The Adam optimizer and binary cross-entropy loss were used to create the DNN model.

The optimized DNN model achieved a slightly higher accuracy of 85.33%, but its precision and recall metrics were notably different, with a precision of 40.82% and a recall of 100%. As a result, the F1 score of the DNN model was negatively impacted. Given that the baseline models consistently attain 79% accuracy and 85.33% of accuracy for DNN, we attribute the inability to achieve higher accuracy to the stochastic character of ChatGPT's random outputs.

For topic modeling, Latent Dirichlet Allocation (LDA) was used. The dataset underwent preprocessing, exploratory data analysis, and topic modeling using LDA. Two approaches, bag-of-words and TF-IDF, were explored. The LDA models extracted relevant keywords for each topic and provided insights into the themes present in the reviews. The models were evaluated using perplexity and coherence scores.

The addition of review topic categorization by LDA to the automated feedback process adds another degree of analysis and insight. Restaurant management can make more informed judgments to resolve particular customer problems and enhance overall customer happiness by evaluating both the actionability and the identified subjects of the reviews. Overall the project successfully developed ML models to automate actionable customer feedback determination, leveraging the data driven techniques and insights to improve decision making in restaurant management.

**Resources**:

Local Consumer Survey:

<https://www.brightlocal.com/research/local-consumer-review-survey/?SSAID=314743&SSCID=b1k5_uvia>

<https://www.brightlocal.com/research/local-consumer-review-survey-2022/>

Restaurant’s Reviews for Dataset:

OpenTable - <https://www.opentable.com/dal-rae?ref=1068>

Yelp - <https://www.yelp.com/biz/dal-rae-restaurant-pico-rivera>

OpenAI API:

<https://openai.com/blog/openai-api>

Class notes (May 27th, 2023) - NLP Classification - Encoding Structure

Latent Dirichlet Allocation(LDA):

<https://www.youtube.com/watch?v=1_jq_gWFUuQ>

<https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2>