

May 9, 2021

Quinn E Knudsen

IST 736

Syracuse University

The Spice is Right

Do Models Understand Sentiment or Deception more Easily?

**Table of Contents**

**Introduction**……………………………………………………………………………......pg 2

Areas of Focus …………………………………………………………….……......pg 2

Restaurant Reviews... ...……………………………………………………….........pg 2

Sentiment Analysis ………………………………………………………….…...... pg 6

Deception…………………………..………………………………………….…….pg 6

**Analysis**……………………………………………………………………………….….…pg 6

About the Data ...……………………………………………………………….…...pg 6

Analysis 1: Sentiment Classification.……………………………………………..... pg 6

Data Cleaning & Prep..……………………………………………….….…..pg 5

Analysis 2: Lie Detection………………………………..………………………....pg 10

Data Cleaning & Prep....………………………………………………...….pg 10

**Results**…....………………………………………………………………………..………..pg 11

Analysis 1: Sentiment Classification ……………………………………….…...….pg 11

*Research Question 1………*………………………………………………...……... pg 11

Analysis 2: Lie Detection………..…………………… ..……………………..…….pg 16

*Research Question 2………*………………………………………….……... pg 16

**Conclusions**…………………………………………………………………………….…...pg 19

**Introduction**

**Areas of Focus**

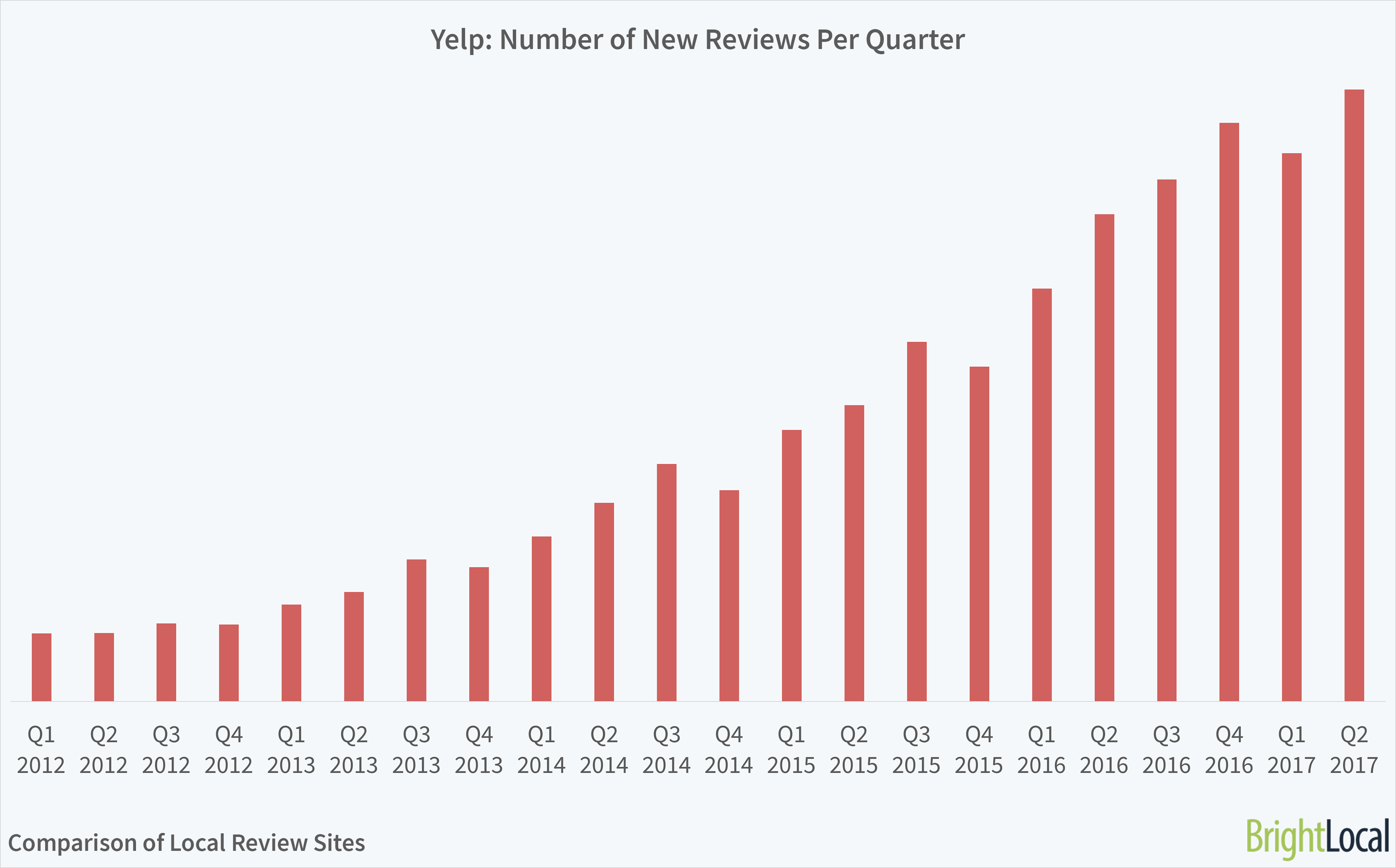
The scope of this paper is to explore classification problems utilizing a restaurant review dataset. Using a Multinomial Naïve Bayes Machine Learning (ML) algorithm, two models will be trained and tested to predict sentiment (positive or negative) and deception (truth or lie).

Through text analysis drawing from Python packages nltk. Finally, this paper will utilize Python packages such as sklearn and CountVectorizer to convert text corpuses into sparse matrixes for supervised machine learning classification. The objective of this research is to answer the question on whether or not machine learning algorithms can understand sentiment as well as they can understand deception. Implications and future considerations will be discussed.

**Restaurant Reviews**

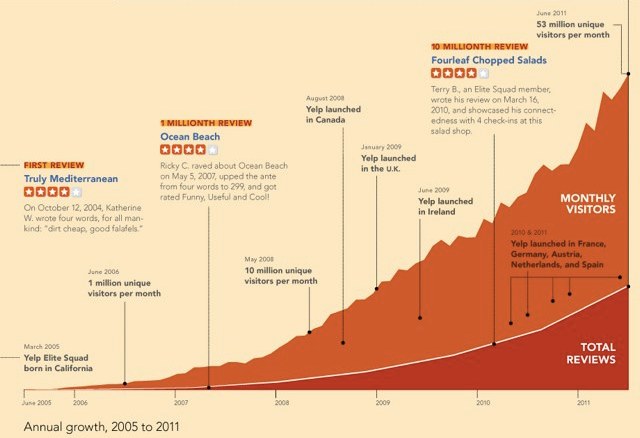
The world and the marketplace have never felt as virtual as they have post-pandemic. One of the hardest hit industries, restaurants, have struggled to cope with restrictions and health concerns over the virus. One saving grace for the industry has been take out and delivery. The manner in which consumers browse and select restaurants, traditionally strolling down a street smelling wafts of delicious food cooking, scenic ambiance or friendly staff have been replaced by online recommendations. Although certainly not a new concept, the need for consumers to select and discern quality restaurants from poor establishments have never before been so online driven. With that understanding, the relevance of online reviews has become increasingly important for customer growth.

One popular manner in which patrons rate restaurants is Yelp. Yelp asks consumers to provide a numeric rating and text-based review of the restaurant. As you can see in Figure 1 below, the growth of the platform has been exponential and has likely heightened it’s relevance in recent years.



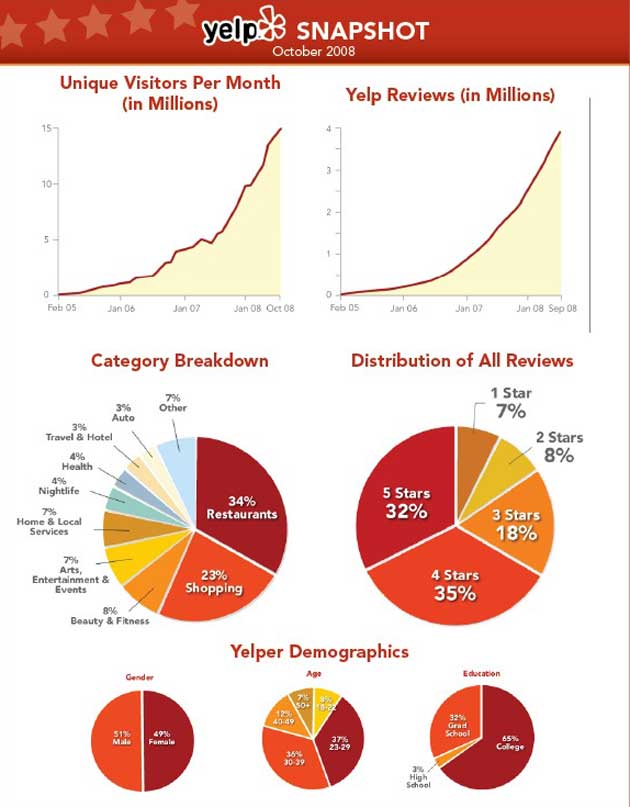
*Figure 1.* Growth of Yelp reviews, 2012-2017.

In addition to more reviews, the site has consistently attracted more monthly viewers with time. As the repository of information builds, so has the user base and the crowd sourced accuracy of the reviews. As of 2016, Yelp has 200 million unique visitors per months and about 200 million reviews worldwide (<https://review42.com/resources/yelp-statistics/>). One study found that 97% of people read reviews of local businesses (<https://review42.com/resources/online-reviews-statistics/>).



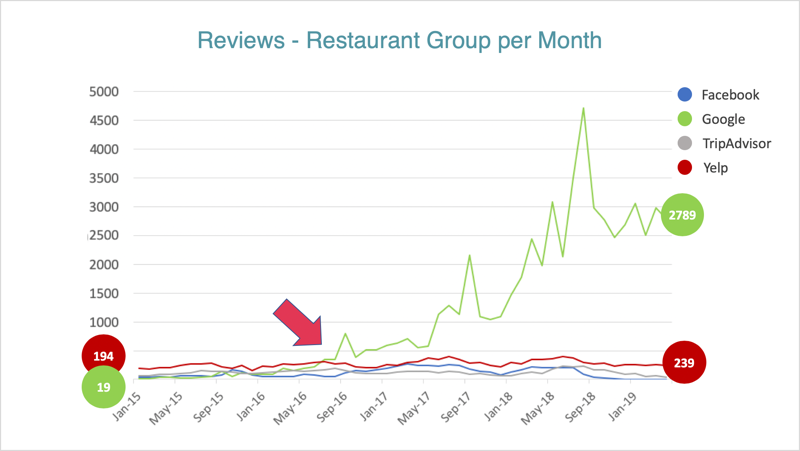
*Figure 2.* Growth of Yelp from 2015-2011.

Furthermore, the bread and butter (so to speak) of Yelp is restaurant reviews as seen in Figure 3. Key to this infographic is the groups most likely to use Yelp are college educated and younger. When thinking from a marketing vantage, college and young professionals are likely to go out to eat with friends and are perhaps more technologically inclined. If the trend for education continues in parallel with the rising proportion of Americans that receive college education and have proficiency and desire for technological crowdsourced tools such as Yelp, the demand for this service is likely to continue to grow in the foreseeable future.



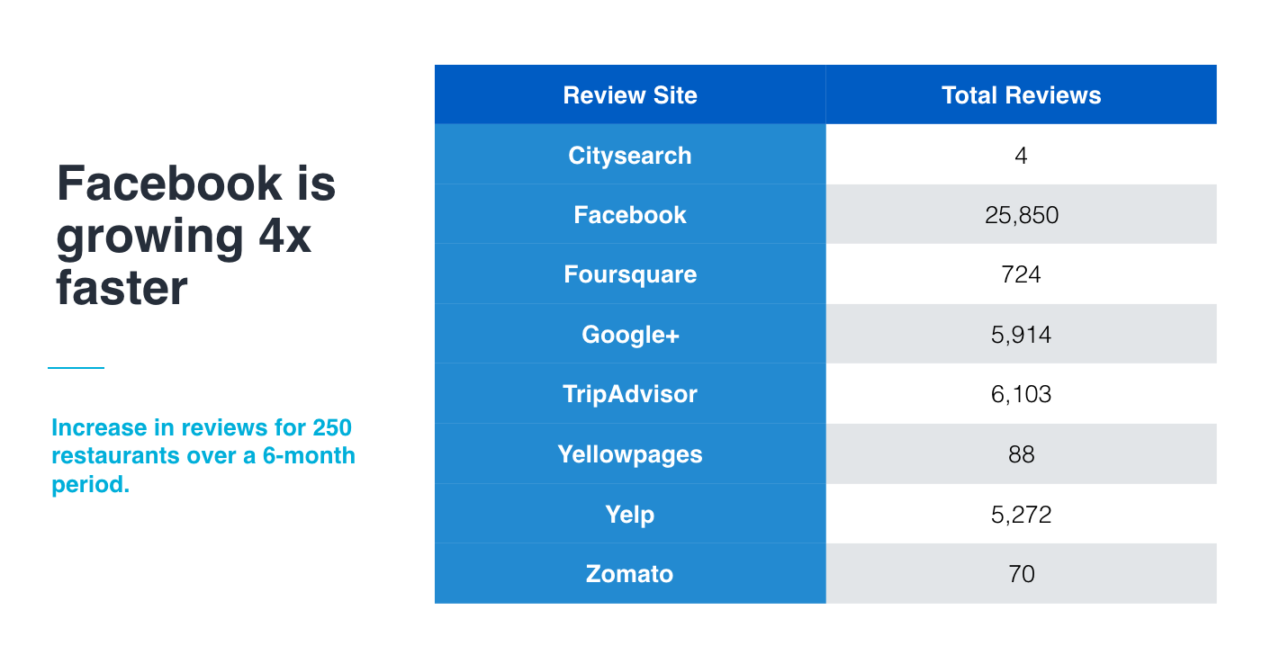
*Figure 3.* Restaurant reviews have been the most prevalent review type since the outset of Yelp.

Despite the market growth of Yelp, there are other key players in the restaurant review space. Crowdsourcing information is growing in a variety of channels and suggest scraping the web for the entirety of reviews might not be limited to one platform. In fact, Google review growth dominated the sector from 2016 to 2019.



*Figure 4.* Google reviews takes off from the pack in September 2016.

Other research has found recent trends in Facebook reviews spiking as seen in Table 1 below.



*Table 1.* Facebook dominates review growth in a recent study of restaurant reviews.

Ultimately, the future is uncertain for how people will choose to find their restaurant in the post-pandemic environment. However the data indicates that relying upon local word of mouth to find the hole in the wall may become a more and more outdated phenomena as consumers seek crowd sourced information to find the best eats.

**Sentiment Analysis**

One applicant of machine learning popularized in recent years is text analytics-or the process of converting large amounts of unstructured written human linguistics into quantitative data capable of uncovering insights meaningful for human interpretation. Sentiment analysis is a specific domain of natural language processing that tries to determine whether the written information is positive, negative or neutral based upon the polarity of the words used in the text. This can prove to be a tall order due to the complexities of language including harder to detect nuances such as sarcasm, slang, negation or amplification. The partial focus of the analysis portion of this paper will explore an applicant of sentiment analysis on commentary related to movie reviews.

**Deception**

A second focus of this paper is to explore whether or not the comment was a truthful statement or a lie. It is hypothesized that this task would be harder for both humans and machines alike to identify. Given a written statement, it is extremely hard to identify whether or not someone is trying to be deceitful. While traditional human methods relay in trained professionals or polygraphs (which can be very intense), so machine learning research has found support for autodetection (<https://www.frontiersin.org/articles/10.3389/frobt.2019.00064/full>). As lying is a complex cognitive process, real time detection can be useful as it is likely that humans will take longer to respond when lying and will speak more when being truthful. However, for a written restaurant review, these methodologies may be more challenging and this classification might be hard to decipher.

**Analysis**

**About the Data**

The dataset contains a series of 92 written statements about restaurants with a binary classification for sentiment and truthfulness. The 3 columns that exist are “lie”, “sentiment” and “review”. All statements provided are written in English and include plain text (no emojis or symbols).

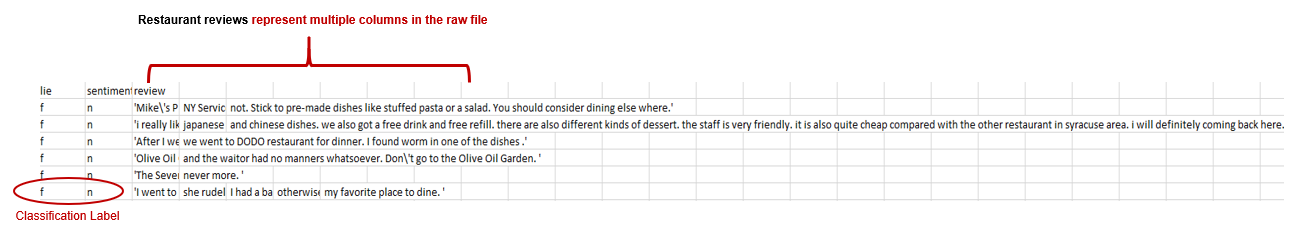
The structure of this dataset required separation into to distinct data frames. The first data frame will consist of sentiment and the restaurant review and the second will consist of the deception and the restaurant review. This was achieved by dropping the column “lie” or “sentiment” depending upon the analysis. 50% of the statements are positive, while the remainder are classified as negative. Detailed data preparation steps are outlined in the ensuing sections.

**Analysis 1: Sentiment Classification**

**Data Cleaning & Prep**

*Before Prep*

Prior to any preprocessing, as described, the dataset was not adequately prepared for conversion to a structured data frame. Before conversion to a document term matrix or tf-idf as will be detailed in the ensuing stages of preparation. An overview of the unaltered dataset can be observed in Figure 5 on the next page.



*Figure 5.* An overview of the raw data file.

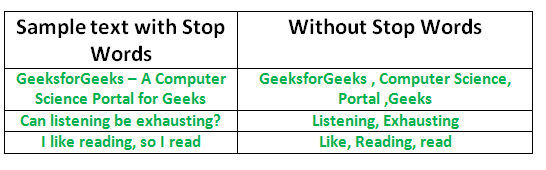
A clear complexity with the data is the alphanumeric symbols that appear throughout as well as the review being located in multiple columns within the datafile. This will need to be merged into one column, tokenized and normalized before modeling.

*Reading in the Data and Initial Preparation*

Data was imported into Python maintaining the header line containing the labels initially in order to segment the data into different data frames. Two lists were created with one containing the labels and the other the content with a split after the first comma to ensure the correct segmentation was achieved.

*Development of Word Clouds for Frequency*

Next, the review list was converted into a data frame with an index and a column of text strings labeled as “Review”. Leveraging the nltk stop words and custom words “go” and “going” the text was stripped of common English words and tokenized. An example of stop word removal can be found below in Figure 6. Each tokenized word was converted to lowercase for consistency. These words were then visualized in a word cloud to show frequency as demonstrated in Figure 7 on the next page.



*Figure 6.* Sample sentences demonstrating the effect of stop word removal.



*Figure 7.* The most commonly used words in the restaurant reviews after removing stop words.

An additional visualization was created in the shape of a slice of pizza which could carry additional domain specific significance.



*Figure 8.* Domain specific representation of the most commonly used words in the restaurant reviews after removing stop words.

*Document Term Matrix*

Python package sklearn CountVectorizer was instantiated to read in the 92 reviews and English stop words were removed. Stop words are common English language words such as “the”, “at”, “there” that generally provide little information gain due to the commonality of their utilization. Files were then vectorized and compiled into a document term matrix. This document term matrix was then converted into an array before finally becoming structured as a data frame that was essential for the analysis.

The final preparation for creating a document term matrix was to label each row in the data frame by their classified positivity or negativity. This was done by using the prepopulated sentiment label “’n’ or ‘p’. This resulted in a sparse, highly dimensional, term matrix with each word representing a variable and each text file representing an observation as depicted in Table 2 below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **10** | **100** | **15** | **16** | **20** | **25** | **2nd** | **30** | **50** | **5pm** | **... write** | **written** | **wrong** |
| 87 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 ... | 0 | 0 | 0 |
| 88 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 89 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 ... | 0 | 0 | 0 |
| 90 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 ... | 0 | 0 | 0 |
| 91 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 ... | 0 | 0 | 0 |

*Table 2.* A document term matrix exemplifying the conversion made.

*Normalized Document Term Matrix (tf-idf)*

An additional step was taken to convert the document term matrix into a normalized document term matrix using tf-idf transformation. This transformation takes the normalized value for which the word occurs in the text matrix (proportionality). This conversion can be found in Table 3 below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **10** | **100** | **15** | **16** | **20** | **25** | **2nd** | **30** | **50** | **5pm** | **... write** | **written** | **wrong** |
| 87 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 ... | 0.0 | 0.0 | 0.0 |
| 88 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.115373 | 0.0 | 0.0 | 0.0 |
| 89 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 ... | 0.0 | 0.0 | 0.0 |
| 90 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 ... | 0.0 | 0.0 | 0.0 |
| 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 ... | 0.0 | 0.0 | 0.0 |

*Table 3.* Conversion of document term matrix to normalized tf-idf matrix showing the normalized word frequency.

*Remove Columns with Numbers or Words Less than 3 Letters*

As indicated in Tables 1 and 2, there are still some undesirable columns included in the dataset even after stop words removal such as the numeric values In order to remove these from the data, a Boolean function was written to return a string for any numbers in a string. Then, using a simple if🡪then logical statement, all numeric data (labeled now as ‘True’) were dropped from the document term matric and tf-idf datasets. A secondary if🡪then statement was then utilized to drop all columns with less than three characters. This yielded the following modifications to the matrices. There are some limitations to this approach such as the removal of “ago” which may or may not provide value in the classification. The modifications are displayed in Tables 3 and 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Index** | **10** | **100** | **15** | **actually** | **add** | **additional** |
| 87 |  |  |  | 0 | 0 | 0 |
| 88 |  |  |  | 0 | 1 | 0 |
| 89 |  |  |  | 0 | 0 | 0 |
| 90 |  |  |  | 0 | 0 | 0 |
| 91 |  |  |  | 0 | 0 | 0 |

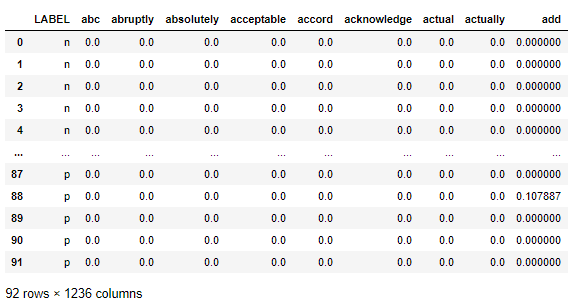
*Table 4.* Document term showing the normalized word frequency minus numeric columns.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Index** | **10** | **100** | **15** | **actually** | **add** | **additional** |
| 87 |  |  |  | 0.0 | 0.000000 | 0.0 |
| 88 |  |  |  | 0.0 | 0.107887 | 0.0 |
| 89 |  |  |  | 0.0 | 0.000000 | 0.0 |
| 90 |  |  |  | 0.0 | 0.000000 | 0.0 |
| 91 |  |  |  | 0.0 | 0.000000 | 0.0 |

*Table 5.* Normalized tf-idf matrix showing the normalized word frequency without numeric columns.

*Add Classification Labels*

The final stage was to then insert the sentiment label back into the dataset in the first column for both the document term matrix and the normalized tf-idf. The final clean data frame can be seen in Table 6 below.



*Table 6.* Cleaned tf-idf matrix ready for modeling.

**Analysis 2: Lie Detection**

**Data Cleaning & Prep**

In preparation for the second analysis, additional or differentiated preprocessing techniques when compared with analysis 1 were not needed. For data prep, this portion utilized the same initial dataset as was used to construct analysis 1 and therefore aside from dropping the sentiment category aforementioned steps can be extrapolated to this analysis. 50% of the statements are lies and 50% are truth.

**Results**

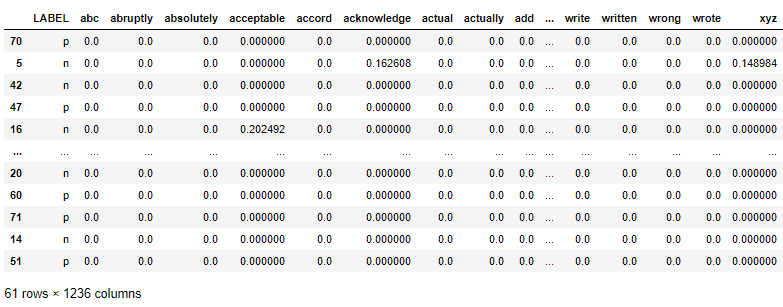
**Analysis 1: Sentiment Classification of Restaurant Reviews**

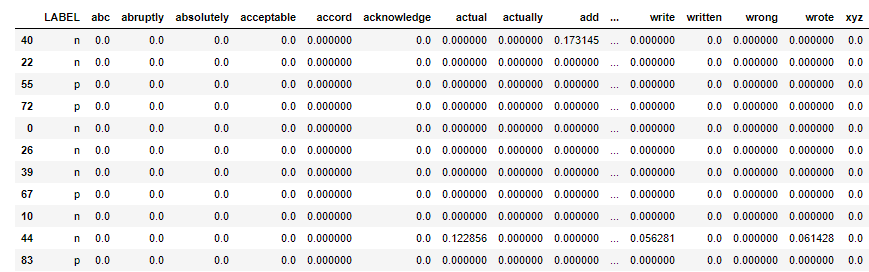
*Research Question 1:* Can a model be trained to correctly predict sentiment based upon given restaurant review text?

In answering this question, the dataset was partitioned into train-test splits with a 10-fold cross validation methodology.

*Train-Test Split*

For the classification modeling, the dataset was partitioned into training and testing. 66% of the data was used to train the model and 33% was held out to use for testing. As the sample was just 92 observations, this meant 61 observations were used to train and 31 were leftover to test. A visual representation of this can be observed in figure 9 below. Splitting the data into train test splits helps determine the accuracy of the model and can be used to fine tune the approach (at the risk of overfitting).



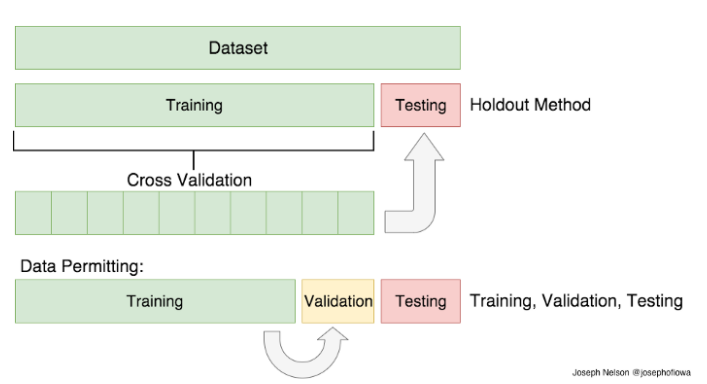




*Table 7.* Training and testing data visualized before modeling.

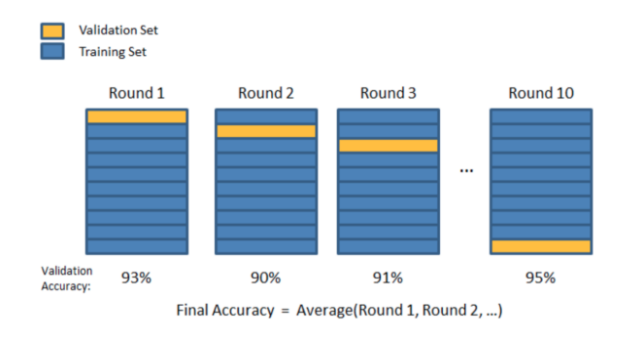
*10-fold Cross-validation*

The model was trained using a 10-fold cross validation procedure in order to improve generalizability and algorithmic learning. The 10-fold cross validation reduces the error in the estimate of the model performance by running multiple iterations of slices of training and testing data and then taking the average accuracy of the models. It is important that the testing data is held out to avoid “stacking the odds” in favor of the model. This would equate to taking the quiz with the answer key.



*Figure 9.* Visual example of a train-test split with cross validation and testing.

Each of the 10 slices have a held-out validation or testing set of the data and the model is trained on the remaining portion. This process is repeated 10 times and scored for accuracy each iteration.



*Figure 10.* Example of a 10-fold cross validation methodology.

**Models**

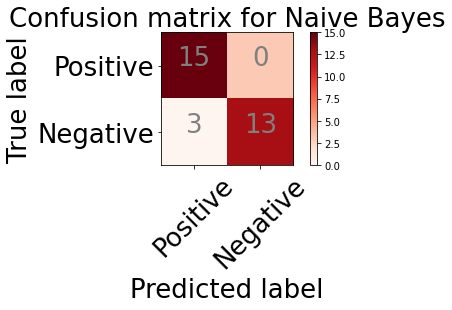
*Multinomial Naïve Bayes*

The Multinomial Naïve Bayes is a probabilistic algorithm often used in natural language processing based upon Bayes’ theorem with an assumption of independence in observations. In reality, it is understood that text violates this fundamental assumption, yet the usefulness of the algorithm is ascertained. This algorithm is useful for text analytics due to it’s ability to handle noise, missing values, can work quickly and scale on large datasets with parallel processing and it is straightforward in it’s interpretation. As this is a supervised method, it requires labeled data and in Python it requires numeric vectorized data inputs.

The multinomial naïve bayes algorithm was chosen for it’s ability to successfully process and classify sparse tdif (text-based) datasets. Using the train data, a model was developed a tested using 10 folds. The final accuracy for the normalized model was 72% with a standard deviation of 0.198 and 84% accuracy with a standard deviation of 15% for the non-normalized model. It was surprising to see the non-normalized model outperform the normalized model.

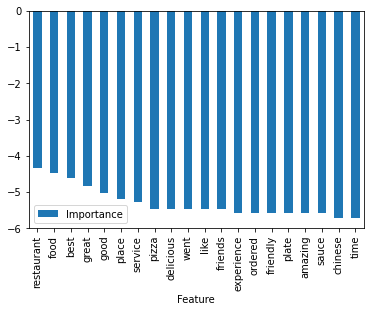
**Not Normalized Model**

*Confusion Matrix*

**

*Figure 11.*  Confusion matrix demonstrating model accuracy with sentiment classification.

*Most Important Features*

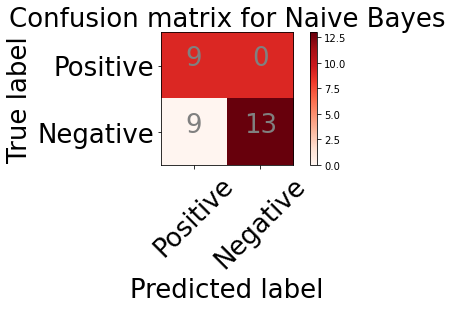
Beyond modeling for accuracy, it is often important to understand the drivers of the model. For this, model weights were extrapolated from the results and mean feature importance was calculated. Model coefficients and feature names were stored in variables, converted into a data frame and sorted in descending order. In this case, all coefficients were negative. The top 20 most influential features can be found in a bar graph in Figure 11.

*Figure 12.* Bar graph of the top tf-idf 20 features in descending order.

**Normalized Model**

*Confusion Matrix*

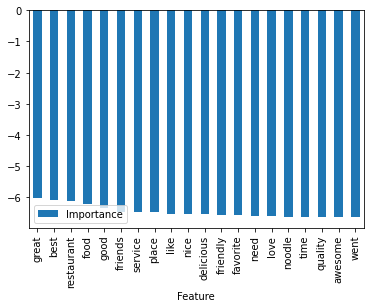
Below in Figure 11, a confusion matrix demonstrates the model was better at classifying negative sentiment than positive sentiment. In fact, positive classification was 50% accurate. These results are solid as this was a perfectly balanced classification task with 50% of the statements positive and 50% negative.



*Figure 13.*  Confusion matrix demonstrating model accuracy with sentiment classification.

*Most Important Features*

Using the same process as the non-normalized model, feature importance is graphed below. Table 7 neatly displays the feature rank, name and relative importance.



*Figure 14.* Bar graph of the top tf-idf 20 features in descending order.

|  |  |  |
| --- | --- | --- |
| **Rank** | **Feature** | **Importance** |
| 1 | great | -6.030968 |
| 2 | best | -6.089576 |
| 3 | restaurant | -6.128857 |
| 4 | food | -6.208176 |
| 5 | good | -6.344193 |
| 6 | friends | -6.46677 |
| 7 | service | -6.468136 |
| 8 | place | -6.468915 |
| 9 | like | -6.524669 |
| 10 | nice | -6.538585 |
| 11 | delicious | -6.545701 |
| 12 | friendly | -6.556972 |
| 13 | favorite | -6.587084 |
| 14 | need | -6.601202 |
| 15 | love | -6.616839 |
| 16 | noodle | -6.619211 |
| 17 | time | -6.620717 |
| 18 | quality | -6.620944 |
| 19 | awesome | -6.644236 |
| 20 | went | -6.64554 |

*Table 8.*  Rank order of top tf-idf 20 features in descending order with positive words in green.

The features in the model that scored with the highest degree of importance make intuitive sense with a classification problem on sentiment. As indicating by the green coloring in Table 7, it is clear to see that positive sentiment words were highly influential in the model. The words indicative of underlying emotion should predict sentiment and using n-grams might improve the model in the future. Ultimately, supportive evidence was found for machine classification of written sentiment in a sample of restaurant reviews.

**Analysis 2: Lie Detection using Restaurant Reviews**

*Research Question 2:* Can a model be trained to correctly detect lies based upon given restaurant review text?

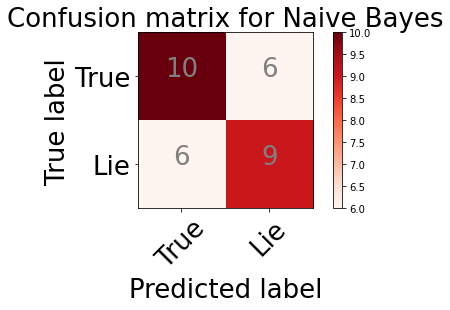
*Multinomial Naïve Bayes*

The multinomial naïve bayes algorithm was chosen for it’s ability to successfully process and classify sparse tdif (text-based) datasets. Using the train data, a model was developed a tested using 10 folds. The final accuracy was 72% with a standard deviation of 0.198.

**Normalized Model**

*Confusion Matrix*

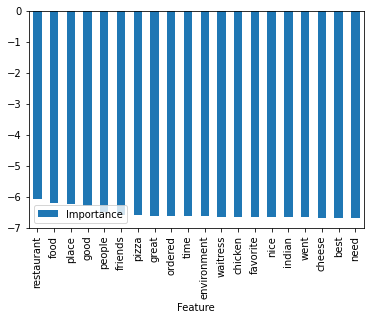
Below in Figure 13, a confusion matrix demonstrates the model was essentially guessing when classifying truth or lies. These results mirror random chance as this was a perfectly balanced classification task with 50% of the statements truthful and 50% deception. Models ranged from 38% accurate to 57% accurate through various cross validations.



*Figure 15.*  Confusion matrix for tf-idf model.

*Most Important Features*

The drivers of the model were also explored for lie detection. Following the same methodology as for the sentiment feature importance extraction, a data frame was created and plotted as illustrated in Figure 13.



*Figure 16.* Bar graph of the top 20 features in descending order.

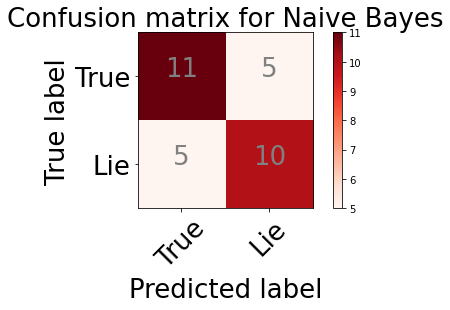
|  |  |  |
| --- | --- | --- |
| **Rank** | **Feature** | **Importance** |
| 1 | restaurant | -6.059453 |
| 2 | food | -6.206011 |
| 3 | place | -6.230061 |
| 4 | good | -6.301656 |
| 5 | people | -6.484593 |
| 6 | friends | -6.59138 |
| 7 | pizza | -6.594656 |
| 8 | great | -6.609941 |
| 9 | ordered | -6.610921 |
| 10 | time | -6.615974 |
| 11 | environment | -6.629507 |
| 12 | waitress | -6.637517 |
| 13 | chicken | -6.6457 |
| 14 | favorite | -6.652671 |
| 15 | nice | -6.659898 |
| 16 | indian | -6.662463 |
| 17 | went | -6.663573 |
| 18 | cheese | -6.667705 |
| 19 | best | -6.67671 |
| 20 | need | -6.682016 |

*Table 9*. Rank order of top 20 tf-idf features in descending order with positive words in green.

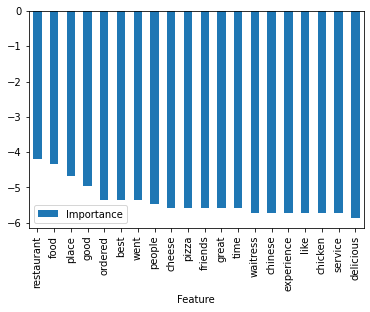
**Not Normalized Model**

*Confusion Matrix*

The non-normalized model faired just about the same as the normalized with accuracies ranging from 40%-60% and a mean of 51% mimicking guessing.



*Figure 17.*  Confusion matrix for non-normalized model.



*Figure 18.*  Rank order of the most important features of the non-normalized model.

Unlike with the sentiment classification, there are limited discernable conclusions that can be extrapolated from these key features. Coupled with the poor model results, these features should be interpreted with caution and practical implication should be restricted. Ultimately, support was not evident for machine classification of truthfulness in in a sample of written restaurant reviews.

**Conclusion**

Just as humans often struggle to understand whether someone is using humor, innuendos, or being deceitful, machines tend to have a hard time making this determination. It is a much less challenging task, however, to understand whether the words written are positive or negative (albeit not a perfect art). This paper found evidence that machines have greater capability in understanding human sentiment about restaurants than making a determination about the truthfulness of the statement.

Sentiment is often overt. For example, if a stranger comes up and states “I like apples”, one would naturally be inclined to infer this individual feels positive about apples. It would not, however, be easy to determine the truthfulness of the statement. Without any prior knowledge or assumptions, there is little to no indication from that statement that the words reflect deceit. Most people infer truthfulness as the default and keying in on lies, without prior knowledge, inflection or additional behavioral observations make it nearly impossible for the untrained eye to properly classify deceit. As such, the machine faired poorly in it’s quest to determine honesty.

In answering the overarching objective of this research paper, machines, like humans, are more capable of determining sentiment than they are deception.