**Lie to Me:**

**Are Customers Lying When They Leave a Review**

**By: Ben Harwood**

**Introduction**

The advent of the internet (by Al Gore or whoever) brought with it the ability for companies to self-promote themselves for free through word of mouth by online customer reviews. With no filters, customers can say as much or as little as they wish, along with a ranking (usually 1 to 5 stars). In order to maximize this opportunity, any time we buy just anything whether online, in person, a service, or even a doctor visit we are solicited to leave a review. Companies often ties incentives and/or penalties to these reviews.[[1]](#footnote-1)

A valid question one may ask while reading these reviews is, “How legit are these?” In other words, was the person writing the review being honest or were they lying for whatever reason. This could make a world of difference in how an “outside” perceives the company prior to doing business with them. Moreover, is it possible to detect sentiment? For example, the following phrase could be read one of two ways: “I couldn’t believe how much they took to help me.” On one hand, this could be meant for praise, acknowledging that the company or person took their time and made sure to assist the customer properly. On the other, it could be a subtle jab, indicating that it took too long. On its own this statement is difficult to interpret.

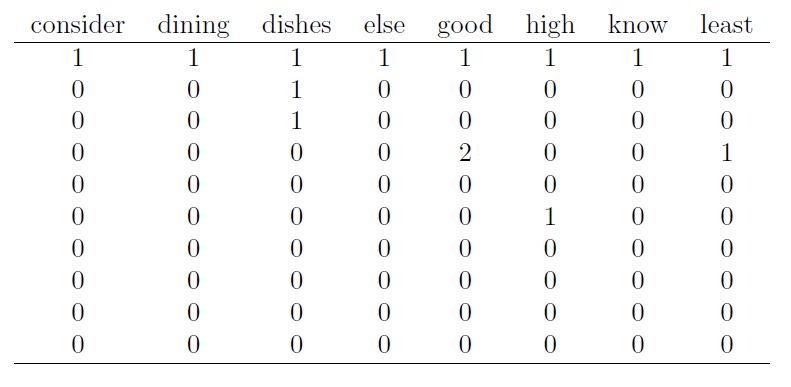
So how can a company decipher their reviews? Is it possible to detect fake bad reviews (for example left a competitor)? Could an outside determine if the company is hiring people to leave fake positive reviews (which happens, unfortunately[[2]](#footnote-2))? This paper is examination of the effectiveness of two techniques of doing so.

**Section 1: Analysis and Model**

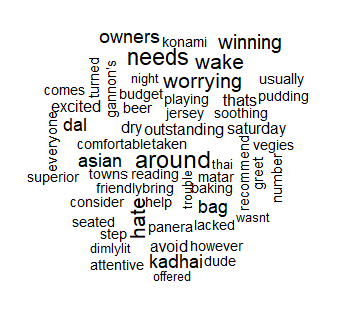
The general idea of this study is to apply Naïve Bayes classifiers and support vector machines to a collection of reviews that have been identified as either true or a lie, and either positive or negative.

**Section 1.1: About the Data**

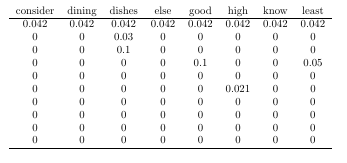
The data for this study is a collection of restaurant reviews. Each review was labeled ‘T’ or ‘F’ if was a lie or not, and “P” or “N” if the review was positive or negative, respectively. Two of the reviews provided had nothing in the comments section, so these were treated as missing values and excluded. The content of each review came broken into numerous pieces, which needed to be combined into a single data value. The reviews were then collected together into a digital corpus[[3]](#footnote-3) and then a so-called “document term matrix” (DTM going forward) is formed with the content of the reviews. The DTM is a matrix where each row is a different document, and the columns are the individual words from any of the reviews. The elements of the matrix are the frequencies in which the words appear in the documents. The below example shows the word counts for the some of the words that appear in 10 of the reviews[[4]](#footnote-4).



To get an idea for which words are more common than others, here is a “word cloud”, where the size of the word indicates its frequency amongst the papers (bigger is more frequent).



In normal circumstances, an important part of data cleaning is dealing with missing values. In this case, two of the provided reviews had only ‘?’ left as comments, so these were excluded. One thing that does need to be done, however, is to normalize the counts in each row, because each review is a different length. To this end, each word count is divided by the total number of words in the associated document, resulting in the following transformed matrix:

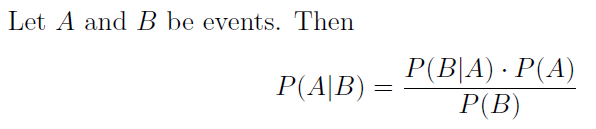


With this complete, two data sets were then created. One with the normalized data and the lie labels, and one with the normalized data and the sentiment labels.

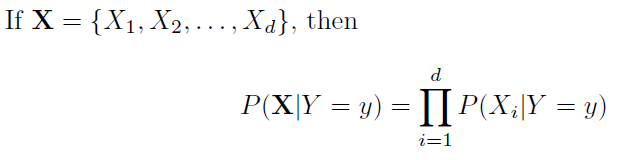
**Section 1.2: Analysis**

**1.2.1: Naïve Bayes**

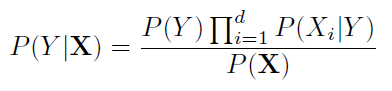
As a refresher, the Naïve Bayes classifier is based on Bayes’ Theorem from probability.



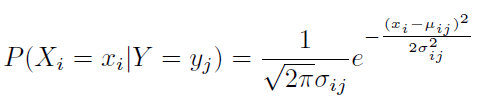
Bayes’ Theorem provides a way to calculate a conditional probability if the individual probabilities and the opposite conditional probability are known (or easily computable). In the case of a vector (like a dataset), a conditional probability is calculated by determining the conditional probability is done component-wise and then multiplying them all together:



Technically this only works if the components are independent, but independence is just assumed when using this technique[[5]](#footnote-5), and Bayes’ Theorem becomes



and because *P*(**X**) does not depend on *Y* the case (*Y*) that maximizes the numerator is likely “winner”. For categorical variables this all works very easily, however for continuous variables a distribution must be used to calculate the probabilities. Generally, the Gaussian distribution is used:



Now, with all of that groundwork laid, to determine the correct Naïve Bayes model to use, three separate models were developed. In each case, a random sample of 80% of the original data was used for training. Each model was then applied to the remaining portion of the dataset to see how well the model could predict if the review was a lie or not, and whether it was positive or not.

The following tables show predicted lie vs actual lie. The bold face entries show how many were correctly predicted for each.

|  |  |  |
| --- | --- | --- |
| Figure : NB lie prediction model 1 | Figure : NB lie prediction model 2 | Figure : NB lie prediction model 3 |

Similarly, these same training sets were used on the sentiment data. Here are the prediction tables for sentiment detection.

|  |  |  |
| --- | --- | --- |
| Figure : NB sentiment prediction model 1 | Figure : NB sentiment prediction model 2 | Figure : NB sentiment prediction model 3 |

Performance metrics for these models will be discussed in the next section.

**Section 1.2.2: Support Vector Machines**

As much as it sounds like some kind of self-help group, a support vector machine is a very useful technique for splitting data. It is very mathematical in nature, relying heavily on the wonder that is linear algebra. Without getting into the mathematics (which could comprise its own 1000 page book), the basic idea is it considers each data point as it would exist in physical space (whatever 785 dimensional space looks like, in the present example) and then finding a way to separate the data into two groups with as wide a gap as possible between the two groups. In the case of two-dimensional data, one might relate the separation as such:



It is not uncommon that a nonlinear model is needed, so certain kernel functions are used. In the case of the present data, a polynomial kernel function yielded the most useful results. Using the same training sets as with the Naïve Bayes models from the previous subsection, the following tables show the predictions using support vector machines for lie-detection and sentiment detection:

|  |  |  |
| --- | --- | --- |
| Figure : SVM lie prediction model 1 | Figure : SVM lie prediction model 2 | Figure : SVM lie prediction model 3 |
| Figure : SVM sentiment prediction model 1 | Figure : SVM sentiment prediction model 2 | Figure : SVM sentiment prediction model 3 |

What is that? A mythical 100% accurate prediction model??? And on the same data that was only 2/3 accurate with Naïve Bayes? Curious…

The following metrics were calculated for each model:

1. Accuracy: the ratio of correct predictions to total predictions
2. Recall/sensitivity: the ratio of true positives to total condition positives
3. Precision: ratio of true positives to total predicted condition positives
4. F1: harmonic mean of recall and precision
5. Receiver operating characteristic (ROC) curve – this shows several things, among them:
   1. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
   2. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
   3. The area under the curve is a measure of text accuracy.

So, ROC curves above the diagonal are good, but the further away they are from the diagonal, the better. These metrics will be discussed in the next section.

**Section 2: Results**

Numerous models were developed as discussed in the previous section. For each model, the various performance metrics were calculated. This table gives a synopsis of each lie prediction model’s performance:

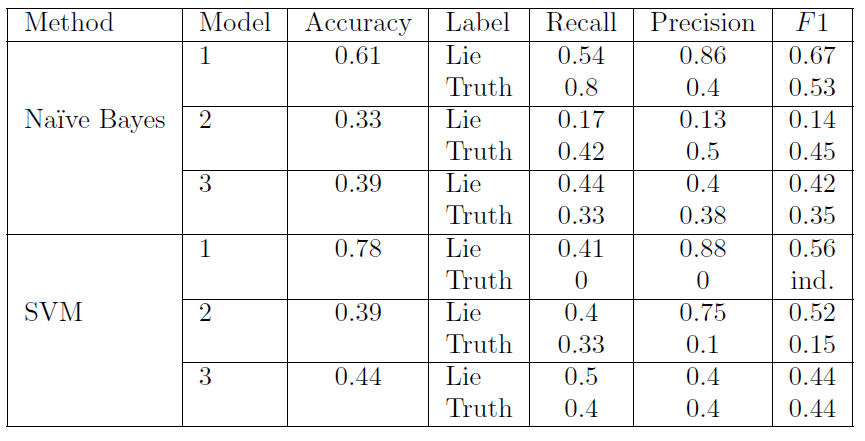


Figure : Lie prediction model performance metrics

An argument could be made for either Naïve Bayes model 1 or SVM model 1 to be the “best” due to the high accuracy of each. It is the belief of the present author that the Naïve Bayes model is better, because the AVM model did not correctly predict a single true review. The ROC curves for each support this claim:

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated  Figure : Naive Bayes ROC curve (lie) | A close up of a map  Description automatically generated  Figure : SVM ROC curve (lie) |

Remember, good models should have the ROC curve as far above the diagonal as possible.

Turning now to sentiment detection, here is the metric table:

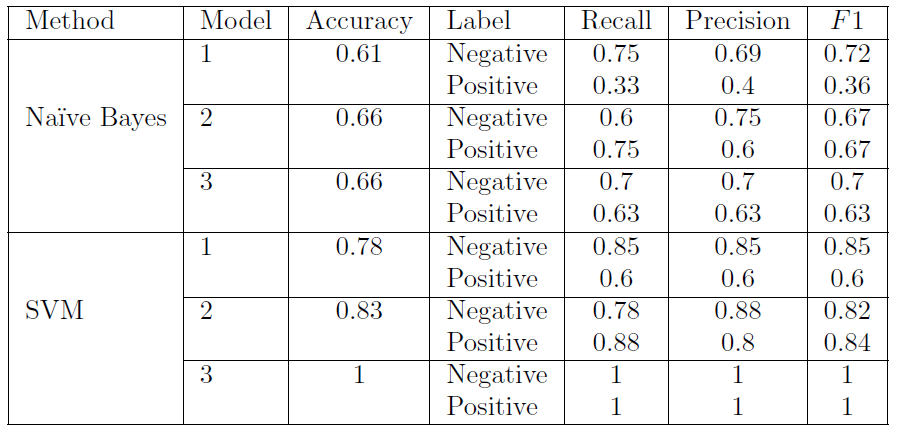


Figure : Sentiment prediction model performance metrics

Forgetting SVM model 3, the race is on between Naïve Bayes models 2 and 3 and SVM models 1 and 2. Here are ROC curves for all four:

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated  Figure : Naive Bayes model 2 ROC curve (sentiment) | A screenshot of a cell phone  Description automatically generated  Figure : Naive Bayes model 3 ROC curve (sentiment) |
| A screenshot of a cell phone  Description automatically generated  Figure : SVM model 1 ROC curve (sentiment) | A screenshot of a cell phone  Description automatically generated  Figure : SVM model 2 ROC curve (sentiment) |

From these, it is clear that the second support vector machine model was the best performer.

**Conclusion**

Regardless of how companies monitor their reviews, they’ll never be able to stop false reviews or negative reviews. Grant Cardone, a well-respected (and very over-the-top) sales coach, says, “If no one hates what you’re doing, no one knows what you’re doing,” meaning if you’re doing great things you will have haters. But this is a good thing, because it means you’re moving in the right direction.

Perhaps the benefits of a study like the one just described are better for the general public, in that a potential patron could be given a score for any posted review, like a truthfulness scale, so that they could help discern the legit reviews from the bogus ones. A similar thing could be done for positive and negative reviews as well. Of course, the respective companies might not be overly excited by such a mechanism.

In the end, people will be people. And if there is one thing that no model of this type can predict, it’s the idiosyncrasies of an individual human being.

1. The author gets a $250 penalty for every survey response less than 980… out of 1000. Since when did “very good” become unacceptable?!?!? [↑](#footnote-ref-1)
2. A certain automotive review website allows the dealership to, for an additional fee, filter out negative reviews [↑](#footnote-ref-2)
3. Corpus: a collection of written texts, especially the entire works of a particular author or a body of writing on a particular concept [↑](#footnote-ref-3)
4. After removing “stop words” such as ‘a’, ‘is’, ‘it’, etc., as well as excessively lengthy words. Also, all words are converted to lowercase, and numbers are removed [↑](#footnote-ref-4)
5. Hence the name Naïve Bayes [↑](#footnote-ref-5)