Darrell Nelson II

Text Mining (IST - 736)

08/23/2019

HW06: Benoulii and Multinomial Naïve-Bayes in Sci-kit Learn

Introduction

Lie detection is a major theme in "psychology and law," which is one of the main areas of applied psychology. It is not difficult to understand why it is important to know whether someone is lying or telling the truth in police investigations, court trials, border control interviews, intelligence interviews etc. To aid lie detection, psychologists and practitioners have developed numerous lie detection tools that range from observing behavior, analyzing speech, and measuring peripheral physiological responses to recording brain activity.

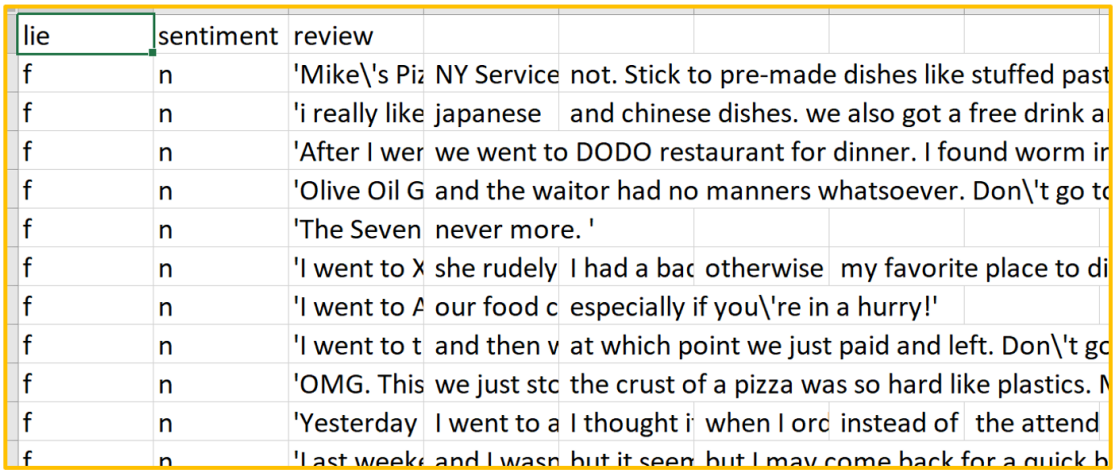
Humans ever since the beginning of time humans have tried to sharpen their own senses to detect when someone else is lying. Many claim to be experts and provide social clues and signs as to when someone is lying (e.g. if they look to the left) however, even the most self-established experts are still no better than random guessers based on probability. That means that if you toss a coin in the air you will be as likely to detect deception as the truth no matter how much you train. And while it is true that a very few people are better at detecting deception than others, they are barely above chance. In fact, those that are really good are only correct somewhere around 60% of the time; that means that 40% of the time they are wrong, and you would not like them on jury duty if you are on trial.

For humans there is no magic bullet. Detecting lies is still a mystery that has not been solved despite heavy study. Mainly, because the person detecting the lies does not have enough background information on the behaviors of the accused liar to make a valid deduction. In my estimation, the best person to detect if another is lying is either their mom or spouse. Can machines pick up the mantle that we can't possibly carry? Can machines solve deception and rid us of using juries and subsequently freeing all of humanity of the pain of jury duty?! Read on to find out.

Analysis and Models

About the Data

Restaurant reviews were pulled from the web and labeled based on their authenticity and sentiment. All 92 reviews were labeled as either being true (t) or false (f) and labeled negative (n) or positive (p) for their sentiment towards the product/company being reviewed. This dataset has a 50/50 split for authenticity and a 50/50 split for sentiment. The labeled reviews were stored in a comma separated values file (CSV) named "deception_data_converted_final" (*Figure 1*.).



lie	sentiment	review					
f	n	'Mike\'s Pi	NY Service	not. Stick to pre-made dishes like stuffed past			
f	n	'i really like	japanese	and chinese dishes. we also got a free drink a			
f	n	'After I wer	we went to DODO restaurant for dinner. I found worm ir				
f	n	'Olive Oil G	and the waitor had no manners whatsoever. Don\'t go to				
f	n	'The Seven	never more. '				
f	n	'I went to X	she rudely	I had a bac	otherwise	my favorite place to di	
f	n	'I went to A	our food c	especially if you\'re in a hurry!'			
f	n	'I went to t	and then v	at which point we just paid and left. Don\'t go			
f	n	'OMG. This	we just stc	the crust of a pizza was so hard like plastics. N			
f	n	'Yesterday	I went to a	I thought i	when I orc	instead of	the attend
f	n	'Last week	and I wasn	but it seem	but I may come back for a quick b		

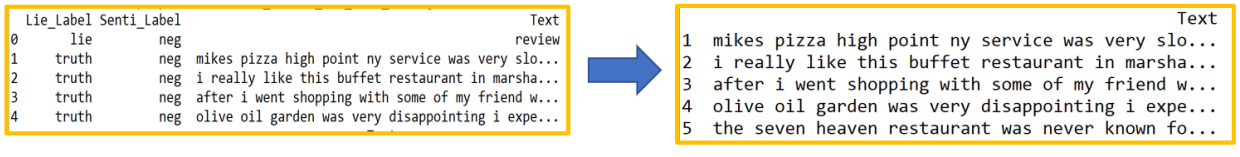
Figure 1. Raw CSV file

The raw CSV file reviews were bifurcated unevenly into multiple cells causing some reviews to be 10 cells/columns long and others to only take up one or two cells (*Figure 1.*). This issue was resolved by importing the rows into a new CSV. This method also allowed for preprocessing and text cleaning in series to porting to a new file. First, the text was cleaned by removing: unnecessary spaces at the beginning or end of the review, making all words lowercase, removing punctuations and other unwanted symbols (e.g. +, \, \t, etc.), and columns were renamed from “lie”, “sentiment”, and “review” to ”Lie_Label”, “Senti_Label”, and “Text” respectively.

Lie_Label	Senti_Labe	Text
lie	neg	review
truth	neg	mikes pizza high point ny service was very slow and the quality was low you would think they would
truth	neg	i really like this buffet restaurant in marshall street they have a lot of selection of american japanese
truth	neg	after i went shopping with some of my friend we went to dodo restaurant for dinner i found worm i
truth	neg	olive oil garden was very disappointing i expect good food and good service (at least!!) when i go ou
truth	neg	the seven heaven restaurant was never known for a superior service but what we experienced last v
truth	neg	i went to xyz restaurant and had a terrible experience i had a yelp free appetizer coupon which coul
truth	neg	i went to abc restaurant two days ago and i hated the food and the service we were kept waiting for
truth	neg	i went to the chilis on erie blvd and had the worst meal of my life we arrived and waited minutes for

Figure 2. Clean CSV

The clean CSV was then imported as a pandas dataframe where the label columns were removed, and the reviews were put into a list for vectorization.



	Lie_Label	Senti_Label	Text
0	lie	neg	review
1	truth	neg	mikes pizza high point ny service was very slo...
2	truth	neg	i really like this buffet restaurant in marsha...
3	truth	neg	after i went shopping with some of my friend w...
4	truth	neg	olive oil garden was very disappointing i expe...

	Text
1	mikes pizza high point ny service was very slo...
2	i really like this buffet restaurant in marsha...
3	after i went shopping with some of my friend w...
4	olive oil garden was very disappointing i expe...
5	the seven heaven restaurant was never known fo...

Figure 3. Label Removal

The vectorized list is stored as a sparse matrix where the rows represent each review and the columns represent all the words in the 92-review corpus. The values in each column represent how many times each word was used in each review. The data is then processed and divided into a 30/70 testing/training set. Multinomial Naïve-Bayes models were then run to predict review sentiment and truthfulness.

	abc	about	abruptly	absolutely	...	your	youre	yourself	yuenan
0	0	0	0	0	...	0	0	0	0
1	0	0	0	0	...	0	0	0	0
2	0	0	0	0	...	0	0	0	0
3	0	0	0	0	...	0	0	0	0
4	0	0	0	0	...	0	0	0	0
5	0	2	0	0	...	0	0	0	0
6	1	0	0	0	...	0	1	0	0
7	0	0	0	0	...	0	0	0	0
8	0	0	0	0	...	0	0	0	0
9	0	0	0	0	...	0	0	0	0

Figure 4. Sparse Matrix

Multinomial Naïve-Bayes

The multinomial naïve-bayes model is an application of supervised machine learning. This technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. It uses joint and conditional probabilities to draw relationships between the inputs and outputs. A Naïve-Bayes classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label. This assumption is violated in a lot of real-world cases, especially in text mining. Based off syntax and semantics in human language only certain words are followed after others in a sentence. Although there are a variety of different words one can use, the proceeding words are more or less determined by the language in which the texts are constructed.

Bernoulli Naïve – Bayes

Like the multinomial model, the Bernoulli algorithm assumes that the attributes are conditionally class independent. It is equivalent generates an indicator for each term of the vocabulary, either "1" indicating presence of the term in the document or "0" indicating absence. The different generation models imply different estimation strategies and different classification rules. The Bernoulli model estimates P(t|c) as the *fraction of documents* of class "c" that contain term "t". In contrast, the multinomial model estimates P(t|c) as the *fraction of tokens* or *fraction of positions* in documents of class "c" that contain term "t". When classifying a test document, the Bernoulli model uses binary occurrence information, ignoring the number of occurrences, whereas the multinomial model keeps track of multiple occurrences. As a result, the Bernoulli model typically makes many mistakes when classifying long documents. For example, it may assign an entire book to a particular class because of a single occurrence of that class-term in the book. (Manning, Raghavan, & Schutze, 2008)[1]

[1] (Manning, Raghavan, & Schutze, 2008)

Results

Both Naïve – Bayes models were used to predict sentiment and truthfulness from the restaurant review database using 3-fold cross validation. Three vectorization techniques were explored in each model to understand the differences in their performance. Overall, the lie detection accuracies were always below their sentiment counterparts by 14 to 23 percentage points. Therefore, it is reasonable to conclude that the Naïve – Bayes models are much better at predicting sentiment vs lies.

Multinomial Naïve-Bayes		
	Sentiment	Lie Detection
Baseline	76.88%	60.50%
Bigram	78.06%	63.69%
TF_IDF	78.06%	62.58%
Bernoulli Naïve-Bayes		
Baseline	84.66%	61.61%
Bigram	80.25%	58.21%
TF_IDF	79.14%	56.06%

Table 1. Naïve – Bayes Accuracy Performance

The 3 vectorizers used are described below:

- Baseline = Vectorizer that generates a column for each word in all the documents and counts how many times a word is used in each document.
- Bigram = Vectorizer that generates a column for each sequential pair of words in a document and counts how many times a word pair is used in each document.
- TF_IDF = Vectorizer that generates a column for each word in all the documents; the tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

For the multinomial model the bigram vectorizer achieved the highest accurayc, and the baseline vectorizer outperformed the rest in the Bernoulli model. Model performance not only depends on the vectorizer but also the type of data being analyzed. There are no silver bullets!

Naïve – Bayes models were 19% more accurate on average at predicting sentiment than figuring out the truth.

This could be due to the fact that in language there are a lot of key words that signify how people feel about a certain subject. For example, words like “bad, horrible, and worst” are generally associated with negative sentiment and words like “great, excellent, and happy” are generally positive leaning. However, with truth telling it is much less about what the reviewers write, and more about the context in which they write it. For example, if looking at reviews for books and a review is talking about how fast his new car goes it is obviously a fake review. These nuisances of context are lost on the model. The model has no way of understanding what the subject matter of the review should be about.

Conclusion

Using a multinomial naïve-bayes model to predict sentiment and authenticity is better than random guessing (50%). Therefore, both models have validity and can assist in predicting sentiment and authenticity.

The use of different vectorization methods can have a great effect on the accuracy of a model. Although, more complex/robust is not always better. For example, term frequency-inverse document matrix appears to be the best vectorizer because it accounts for word frequencies between documents. However, it didn’t perform better than the generic baseline vectorizer with the Bernoulli model. This phenomenon is what makes Data Science an art as much as a science.

In conclusion, the Bernoulli model proved to be the higher performing naïve-bayes model for text mining classification. Language is a lot more complicated than 1s and 0s and in order to build the best predictive models more effort needs to be put into transforming sentences and words into corresponding semantic values. Luckily, there has been a lot of work done in this area to transform our language into simpler categorical and numerical values that a computer can understand. Although the lie detection application for the model shown here leaves a lot to be desired, it does shed light on what is possible with ML.