Name: Morgan Gere

Course: IST 736- Text Mining

Date: 5/05/2022

**Multinomial Naïve Bayes**

**Introduction**

A data set was given that was comprised of 92 restaurant reviews. Each review had a sentiment classification being either positive or negative, and a lie classification either True or False. The goal was to create two separate Multinomial Naïve Bayes models to predict those classifications. A determination of how the model ran based on accuracy was conducted. Followed by an inspection to verify if the model learned what was intended. This was accomplished by looking at the 20 most important features (tokens/words) the model used for its predictions. This was analyzed to identify the model’s competency. We end with conclusions and thoughts on sentiment classification vs lie detection.

**Method**

Tokenization

The data preparation and modeling along with further analysis was all done in python. Only a brief overview of this is explained followed by what was used for each model and why. Several functions were created with the purpose of tokenization, removal of punctuation, removal of stop words, and stemming(Port Stemmer was used in the creation of the stemming function). These were later placed into a list and a function used this list of functions to transform the reviews for vectorization.

Text

Description automatically generated Text

Description automatically generated with medium confidence

Normalization

For sentiment classification the process of normalization involved making all tokens lowercase only. This was preformed to make less possible vocabulary. Since most punctuation does not express sentiment and the few punctuations that do are ambiguous, it was removed. An example of punctuation that expresses ambiguity would be “!”. “I hate you!”, vs “I love you!”. Next the stop words were removed. This was done because the predictor was found to use mostly stop words as its most important features. This was either a case the model learning something else or of overfitting to this data. With such a small data set if for instance “the” is predicted for negative. Then its possible these two example sentences get predicted as the same score. “The most amazing restaurant.”, “The most disgusting restaurant”. This is possible because the text available may not contain examples of amazing/disgusting. Lastly Stemming was attempted. It was found that stemming brought the accuracy of the model down. At first it was assumed that while the accuracy was being reduced either the precision or recall would be increased. After further investigation this was found not to be the case. The precision and recall were both found to have lowered. This is likely due to the small size of the data being used. So, for this model stemming was not performed.

For lie detection the process of normalization involved making all tokens lowercase only this was preformed to make less possible vocabulary. Since punctuation is used the same ways in a lie as in the truth they were removed as no information could be gleamed from them. The same reasoning was used with stop words. Stop words will be used regardless of if a statement is a lie. Although it is possible the extent of which they are used would differ. When tested the accuracy when the model included stop words was too low. Stemming was implemented on the data set. For each technique the lie detection model was created with it and without it. The accuracy of the model increases with each technique being added. More reasoning with the lie detection will be discussed in the conclusion section. Below is the code used to preform the pipelined tokenization using the functions shown previously.

Text

Description automatically generated Text

Description automatically generated

Vectorization

The lists of lists were transformed into a list of strings. Sklearn Tfidvectorizer was used on the tokenized strings. The vector was placed into two data frames.

Text

Description automatically generated

The labels obtained from the original data frame were converted to a list then transformed into numeric representations. These new lists of numeric representations were placed into the data frames of vectorized tokens.

Text

Description automatically generated

Multinomial Naïve Bayes

Sklearn was used to create a model for each label and accuracy was obtained using a 10-fold cross validation. The average of the 10 accuracies was found to represent the accuracy of each model.

Text

Description automatically generated Text

Description automatically generated

20 most indicative words

Permutation importance was used to find the 20 words that are most meaningful to the models. This gave a sence to what the model learned and how it was making its predictions.

**Text

Description automatically generated** **Text

Description automatically generated**

**Results**

The data set contained 92 examples perfectly split for both sentiment and lie detection. This means for sentiment classification there are 46 examples of positive reviews and 46 examples of negative reviews. While for lie detection there are 46 lies and 46 true statements. The raw agreement and baseline are both at 50%. The accuracy of the sentiment classification was found to be 84.78% and the accuracy of lie detection was found to be 60.67%. Both are predicting better than if they just gave back one classification. To understand if the models are preforming as intended. Meaning were the models predicting based on the correct information or were overfitting because of the small data size. The 20 most indicative words were acquired they are shown below.

Table

Description automatically generated with low confidenceA picture containing graphical user interface

Description automatically generated

**Conclusion**

To begin the conclusion looking at the accuracy of the models both are out preforming random guesses or simply predicting all examples as one label. While the lie detection is not preforming at a very high level, the sentiment classification, in terms of accuracy, was preforming well. This is promising but the words the model used needed to be examined.

For sentiment classification there are a few words that stand out as “not what was expected”, and are not something that convey sentiment. Examples are place, called, much, and staff. These are not good predictors and are examples of the model picking up on usage because of the small size of the data set. These are the model overfitting to the data. There are some solid words that indicate the model did learn something useful. Examples would be best, great, friendly, amazing, delicious, terrible, cold, fresh, and bad. Some of these are obvious to all sentiment and some like cold and fresh are specifically because the reviews are about restaurants and food. This model is doing an ok job. It is not great, and this is largely due to the amount of data that the models were trained with. A larger data set would have led to less overfitting.

As for the lie detection there didn’t seem to be any words that were convincing it was learning based on anything other than overfitting to the data. The use of lie detection with machine learning is a lot based on stop words such as no and why.(alexskopje/Depositphotos) In this case including stop words made the accuracy around 50% which is showing the model isn’t learning based compared to the raw agreement or the predicted baseline. Other words that indicate lying according to (alexskopje/Depositphotos) are always and never. While these words did in fact show up in the vectorized tokens they were not used in a highly effective way by the model (they did not appear in the most important features). This model should not be trusted to perform well on a new dataset.

The difficulty of creating each model is very different. With sentiment analysis it is easy to inspect the top indicative words and understand if the model is working as desired. While looking at lie detection there are few words that indicate lying and the ones that have been found to do so by others do not always indicate lies. This makes it near impossible to check if the model is working vs overfitting to the data. According to (alexskopje/Depositphotos) to detect lying the model they created had to use typing (micro-pauses) and the speed of the response which was not captured from this review list dataset. I believe that it is not possible to detect lies from words alone.

**Citations**

alexskopje/Depositphotos. “Online Polygraph Separates Truth from Lies Using Just Text-Based Cues.” *New Atlas*, 19 Mar. 2019, https://newatlas.com/online-polygraph-truth-lies-detector-text-communcation/58916/#:~:text=A%20machine%20learning%20algorithm%20trained%20on%20text-based%20communication,intelligent%20algorithm%20that%20can%20identify%20truth%20from%20lies.