Twitter Sentiment with NLP

—

By Eduardo Osorio

Objective:

The purpose of this study is to find consumer sentiment from their comments at the South by Southwest tech conference. The data we will be looking at today is in the form of tweets from users talking about products showcased at SXSW. We will be using said tweets tell us if the sentiment is positive or negative.

Data Used:

The data used for this for this study was twitter data in the form of individual tweets. The data was manually categorized meaning someone read the tweets and assigned the corresponding sentiment.

- There are 9,093 individual tweets in this dataset
- The three columns provided were Tweet, Subject of tweet, and Emotion (meaning sentiment). However, I only used the tweet and emotion data since subject of tweet didn't really help the model determine sentiment.
- From the emotion data, there are four classes. "Positive", "negative", "no emotion towards brand or product" and "I don't know"

Preliminary EDA:

The total vocabulary was 8848 unique words with 7274 tweets. The average number of non-zero elements was 16.26. Meaning 99.8% of the vectors are actually zero.

The top ten most frequent words are:

- 'sxsw', 7608
- 'mention', 5703
- 'rt', 2331
- 'google', 2059
- 'ipad', 1948
- 'apple', 1839
- 'quot', 1322
- 'iphone', 1230
- 'store', 1209
- "'s", 988

```
Average Number of Non-Zero Elements in Vectorized Articles: 16.261616717074514
Percentage of columns containing 0: 0.9981623215372274
```

Base Model:

For the base model I ran through most of the data without too much cleaning to see how well Random Forest and Naive Bayes would classify the sentiment. I decided to lemmatize the tweets during the tokenization step to help with the dimensionality of the data. Lemmentization basically reduces certain words to the root word. Both models had an average testing accuracy rate of about 65.5%

```
Random Forest
Training Accuracy: 0.9962          Testing Accuracy: 0.6773
```

```
Multinomial Naive Bayes
Training Accuracy: 0.7242          Testing Accuracy: 0.6427
```

EDA:

I used TF-IDF to vectorize. The total vocabulary was 5429 unique words with 2038 tweets.
The average number of non-zero elements was 16.63. Meaning 99.6% of the vectors are actually zero. This is a little higher than the preliminary average of non-zero elements 16.26.

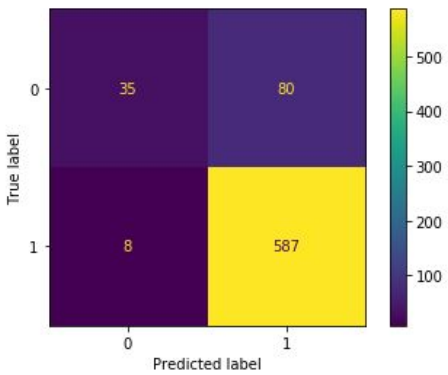
```
Average Number of Non-Zero Elements in Vectorized Articles: 16.634249471458773
Percentage of columns containing 0: 0.996936038041728
```

For the final model, I decided to dropped everything but the positive and negative rows. Out of the 9093 individual tweets, I ended up using only 3548. The reason for dropping more than half the data was because having multiple classes affected the accuracy of the model.

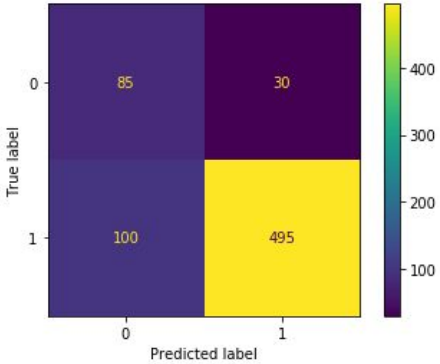
Final Model:

For the final model, I removed every tweet that wasn't positive or negative like was mentioned earlier. This cut about 60% of the tweets but at the same time greatly increased accuracy of our model. I also used SMOTE to address the class imbalance in the data set. The final testing accuracy results were 88%for Random Forest and 82% for Naive Bayes. Which is an average increase of about 20%.

Random Forest
Training Accuracy: 1.0
Testing Accuracy: 0.8761



Multinomial Naive Bayes
Training Accuracy: 0.9587
Testing Accuracy: 0.8169



Post EDA:

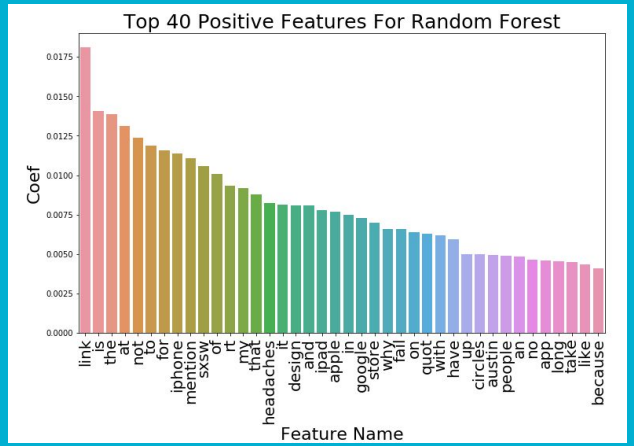
What I found interesting is that the most common negative words and positive words have similar tokens. By just looking at the tokens it seems there is a negative sentiment towards social media apps design like google+'s Circle feature and positive sentiment towards the new Apple products.

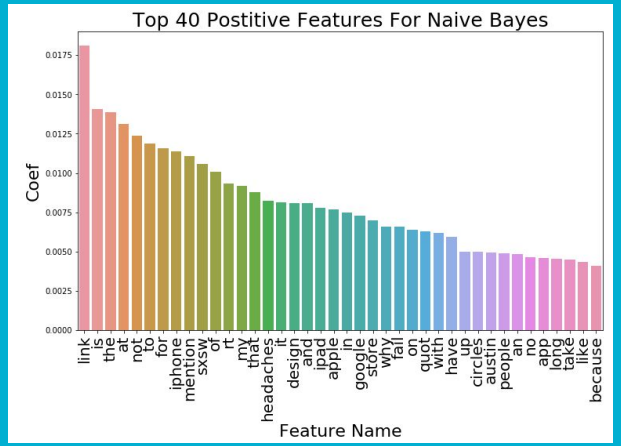
```
Most common Negative Sentiments Words

[('ipad', 188),
 ('quot', 175),
 ('iphone', 161),
 ('google', 145),
 ('apple', 120),
 ('2', 65),
 ('app', 60),
 ('store', 47),
 ('new', 43),
 ('like', 42),
 ('circle', 37),
 ('need', 35),
 ('social', 31),
 ('apps', 30),
 ('people', 29),
 ('design', 28),
 ('get', 25),
 ('android', 24),
 ('austin', 24),
 ('one', 23)]
```

```
Most common Positve Sentiments Words

[('ipad', 1002),
 ('apple', 925),
 ('google', 716),
 ('store', 554),
 ('iphone', 523),
 ('2', 503),
 ('quot', 464),
 ('app', 396),
 ('new', 359),
 ('austin', 294),
 ('amp', 211),
 ('ipad2', 209),
 ('android', 198),
 ('get', 180),
 ('launch', 174),
 ('pop-up', 151),
 ('one', 149),
 ('party', 148),
 ('line', 143),
 ('great', 137)]
```





Post EDA Continued:

Here's an example of positive tweets according to the model.

```
[['jessedee', 'know', 'fludapp', 'awesome', 'ipad/iphone', 'app', "'ll", 'likely', 'appreciate', 'design', 'also', "'re", 'givi
ng', 'free', 't'], ['swonderlin', 'wait', 'ipad', '2', 'also', 'sale']]
```

Here's and example of negative tweets according to the model.

```
[['wesley83', '3g', 'iphone', '3', 'hr', 'tweeting', 'rise_austin', 'dead', 'need', 'upgrade', 'plugin', 'station'], ['hope',
'year', 'festival', 'crashy', 'year', 'iphone', 'app']]
```

Summary/Conclusions:

- Consumers didn't respond so favorably to google's social media app feature called "Circles".
- The model suggests that consumers didn't like google+'s "Circles" because of its design.
- Consumers responded favorably towards the new apple products, Specifically the "IPad2".
- People seemed to like products with google's "Android operating system.

Recommendations:

- If google wishes to work on it’s social media apps, they should really pay attention to the design and functionality.
- Apple should just keep releasing new products in order to keep its fanbase excited.

Next steps:

- The model does not seem to recognize links and mentions correctly, so i would like to address that in the future.
- Train the model to be able to extract the subject of a tweet. That way we can tell where the sentiment is directed at.
- Use pipeline to streamline to modeling process

Thank you!

Thanks to everyone for watching my presentation

Sources used:

- www.geeksforgeeks.org
- https://medium.com/@acrosson/extract-subject-matter-of-documents-using-nlp-e284c1c61824
- www.stackoverflow.com
- https://machinelearningmastery.com/sparse-matrices-for-machine-learning/