

**DETECTING SARCASM WITH TEXT MINING**

IST 736

Professor Ami Gates

Mahsa Saadati, Kobi Wiseman, Daniel Pearl, Susan Rico, Gilbert Guyah

**Table of Contents**

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

**Analysis and Models**……………………………………….………..………………………..pg 3

**About the Data**………………………………………………………………………..pg 3

**Models**…………………………………………………………………………………pg 6

**Results**………………………………………………………………………………………....pg 7

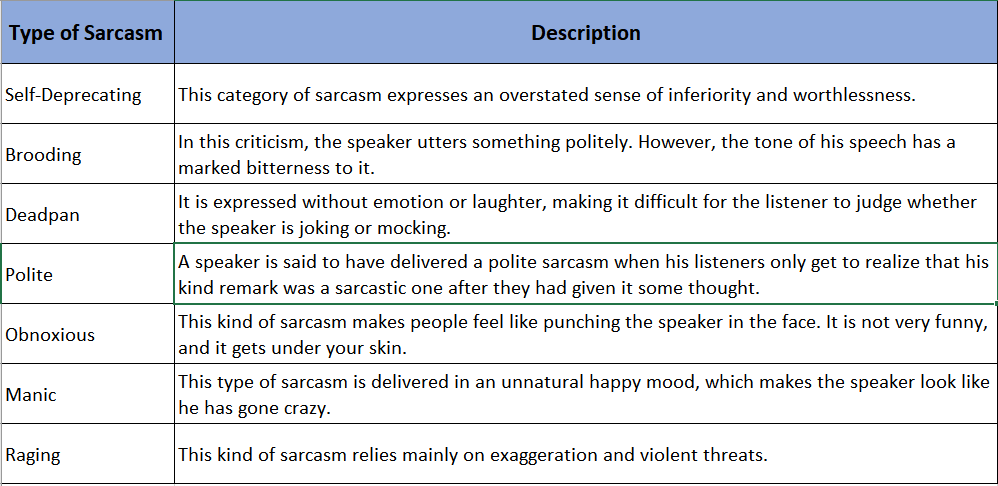
**Conclusion**…………..……………………………………………...………………………..pg 11

**References**…………………………………………………………………………..………..pg 12

**Introduction**

Cognitive psychologist and computer scientist, Geoffrey Hinton once said, “Computers will understand sarcasm before Americans do” (BrainyQuote, 2020). Merriam-Webster defines sarcasm as “a sharp and often satirical or ironic utterance designed to cut or give pain” and/or “a mode of satirical [wit](https://www.merriam-webster.com/dictionary/wit#h1) depending for its effect on bitter, [caustic](https://www.merriam-webster.com/dictionary/caustic#h1), and often ironic language that is usually directed against an individual” (Merriam-Webster, *Sarcasm* 2020).

Sarcasm is a good example of word use that means the opposite of what is being conveyed to express the feeling of frustration, anger, and or insult another individual and often depends upon the voice tone. The type of sarcasm displayed by humans can vary. Table 1 depicts 7 common types of sarcasm along with their description, ranging from self-deprecating, polite, obnoxious, and raging.



**Table 1**. Common Types of Sarcasm

Why does understanding sarcasm matter? Researchers believe that sarcasm can provide valuable insight into how the mind works. More specifically, they found that early exposure to sarcasm improves creative problem solving. On the contrary, an individual’s inability to detect sarcasm may prove to be an early warning sign of brain disease (Chin, 2011). But the benefits of detecting sarcasm are not only limited to the medical field. With an increase of social media platforms, e-commerce, and online customer reviews, many businesses and organizations are relying more on understanding peoples’ opinions on various topics and events to make better informed business decisions.

However, the English language can be very complicated and easily misinterpreted, due to the nature of words with multiple meanings, sentence structure, tone, and body language. Many of the studies performed on sarcasm detection relied solely on spoken words, giving researchers cues on true intention based on tone and other body language signs. In a time where 280- character Tweets and Emojis are interchangeably used to convey moods, emotions, and opinions, there has been an emphasis placed on improving sarcasm detection across the digital spectrum.

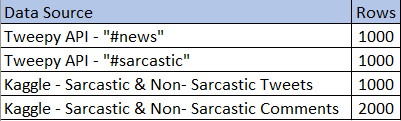
**Analysis and Models**

**About the Data**

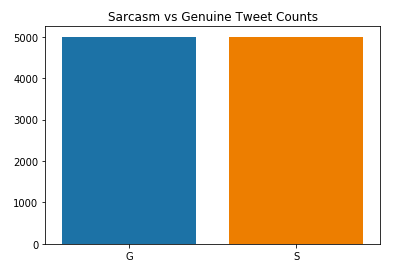
Two datasets were collected from Kaggle for this analysis. The first Kaggle dataset pertains to text data gathered from Reddit containing a “sarcasm” tag. This data contains over 32,000 rows and 10 columns including a “label” column (Ofer, 2018). The next Kaggle dataset was gathered using Twitter data and contains sarcastic headlines from The Onion, while the non-sarcastic text was collected from real news headlines from Huffpost.

To supplement additional data, tweets were extracted directly from the Twitter API through the Tweepy library. Six thousand tweets were collected using the search words “#sarcasm” and “#news” (3000 per search word). The tweets containing the hashtag “sarcasm” were labeled as such, while the “#news” tweets were labeled as non-sarcastic.

Ultimately, a subset of 5000 rows was selected due to computer processing power available to run the models selected for the analysis. Table 2 shows the breakdown of the number of rows selected from each data set. The data set was assessed to ensure the distribution of sarcastic and non-sarcastic comments were adequate for the analysis (Figure 1).



**Table 2**. Breakdown of Subset for Analysis

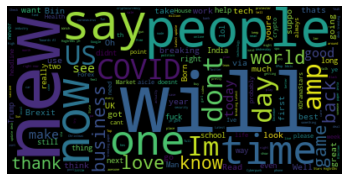
****

**Figure 1**. Text Count (Sarcastic or Genuine)

Figures 3 and 4 display WordClouds created from sarcastic and non-sarcastic tweets before data cleaning. It is worth mentioning that words such as “love,” “people” “occurred more often in sarcastic tweets than in non-sarcastic ones.

****

**Figure 2**. Sarcastic WordCloud

****

**Figure 3**. Non-Sarcastic WordCloud

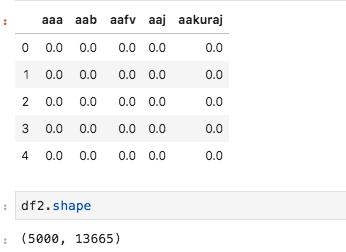
Data cleaning and transformation occurred concurrently. The data was read in Python. White spaces and stop words (words that do not add much meaning to the sentence), digits, non-letter words and words with length of equal and less than 3 were removed, and all the words were changed to lowercase. words such as news and sarcasm were removed from the data set. Each column was converted to a list. and as previously discussed, a subset of 5000 tweets were extracted from the original data set.

CountVectorizer and TfidfVectorizer from the Python library, sklearn, were used to tokenize the reviews. The value of gram\_range was set to (1, 3). WordNetLemmatizer function from the Python library was used to tokenize and lemmatize the words to reduce dimensionality of the data. The content of the text list was passed to CountVectorizer and TfidfVectorizer as content. Two data frames were created out of each vectorizer, each containing 5000 rows (number of tweets). CountVectorizer generated 1000 columns (number of words throughout the entire document) whereas TfidfVectorizer generated 13665.

Figures 4 and 5 depict each dataframe implementing either CountVectorizer or TfidfVectorizer:



**Figure 4**. Dataframe implementing CountVectorizer function

****

**Figure 5.** Dataframe implementing TfidVectorizer (Normalized) function

**Models**

The data set was split to train set and test set. The size of the test set was set to 30% and the remaining 70% was used as a training set.

**Naive Bayes**

The first model implemented a Multinomial Naive Bayes algorithm, which is a probabilistic learning method for classification. The Naïve Bayes classifier assumes that the occurrence of a particular feature is not related to any other feature and is commonly used for text classification. It is typically easy and can provide fast class feature extraction.

**Support Vector Machine**

Support Vector Machine (SVM) is another machine learning algorithm that is widely used for classification or regression problems. It performs this task by determining the best decision boundary, or hyperplane, between vectors. Within this method, there are several “kernels,” which recruit different mathematical functions to create the optimal hyperplane. Ultimately, SVM can be a great tool used to predict classes or sentiment.

For SVM, the kernel was set to linear and Cost to 1.

**Random Forest**

Random Forest is a supervised machine learning algorithm that builds and merges multiple decision trees to create an accurate and stable prediction. Random Forest is more highly efficient when performing analysis on a large dataset. It is adaptable, easy to use, efficient and accurate for both classification and regression. For this model, the n\_estimator was set to 40 for best result.

**Results**

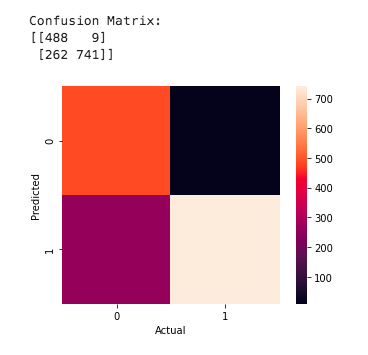
Running all the models on unnormalized data produced higher accuracy compared to normalized data from TfidfVectorizer. The following accuracy scores from different models pertains to the CountVectorizer data frame. Table 3 depicts the different variations of models implemented for this analysis.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Max ngrams** | **All lowercase** | **News included** | **Accuracy score** | **F1 score** |
| **Naïve Bayes** | 1 | no | yes | 0.733 | 0.725 |
| **SVM** | 1 | no | yes | 0.707 | 0.706 |
| **SVM** | 5 | no | yes | 0.701 | 0.7 |
| **SVM** | 3 | yes | yes | 0.701 | 0.698 |
| **Naïve Bayes** | 3 | yes | yes | 0.698 | 0.698 |
| **SVM** | 3 | no | yes | 0.693 | 0.692 |
| **Naïve Bayes** | 3 | no | yes | 0.684 | 0.684 |
| **Naïve Bayes** | 5 | no | yes | 0.676 | 0.677 |
| **Naïve Bayes** | 3 | no | no | 0.669 | 0.672 |
| **SVM** | 1 | no | no | 0.669 | 0.672 |

**Table 3.** Model Results

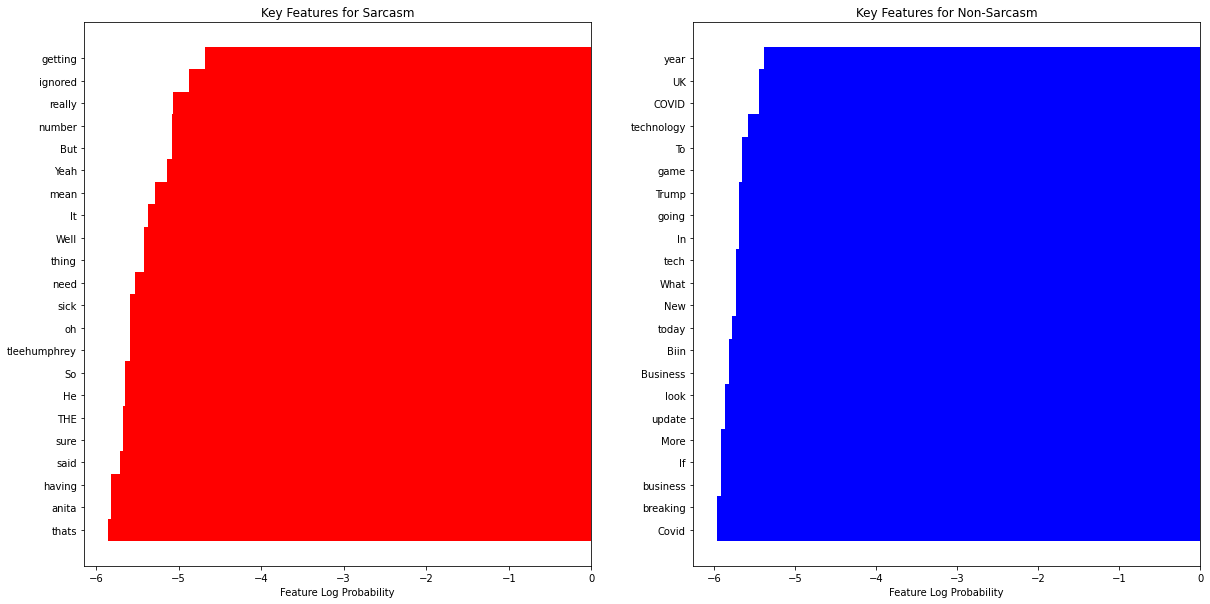
**Multinomial Naive Bayes**

Using this model on CountVectorizer data frame with 7-fold cross validation generated an accuracy score of 73.3%. Figure 6 represents the confusion matrix generated from the predicted and actual labels of the sample data set using Multinomial Naive Bayes.



**Figure 6.** Naive Bayes Confusion Matrix Heatmap

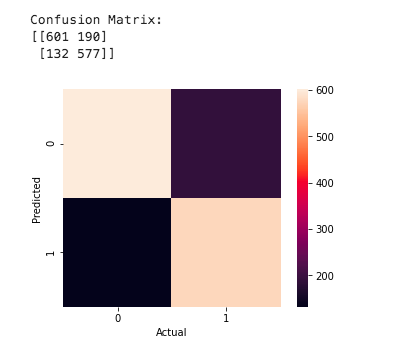
Figure 7 displays the top key features for the sarcastic and non-sarcastic category. Some of the words including “sure,” and “ignored” can be indicative of sarcastic text, however, the remaining features do not provide much insight. The key features for the non-sarcastic text seem to represent news topics based on the subset of data selected for the analysis. This may indicate that the model was better at grouping and defining certain key features for news than sarcastic text.

****

**Figure 7**. Naive Bayes - Top Features for Sarcasm and Non-Sarcasm

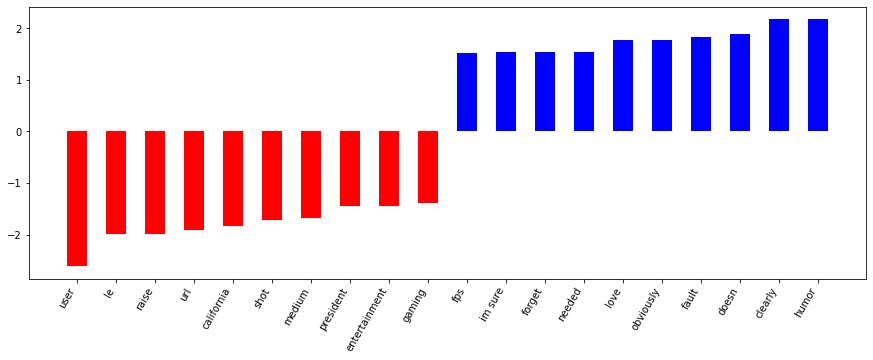
**Support Vector Machine**

Using SVM on CountVectorizer data frame with 7-fold cross validation generated an accuracy score of 70.7%. The Support Vector Machine algorithm performed much slower compared to all other models, which made running this specific model more time consuming in comparison to the others.



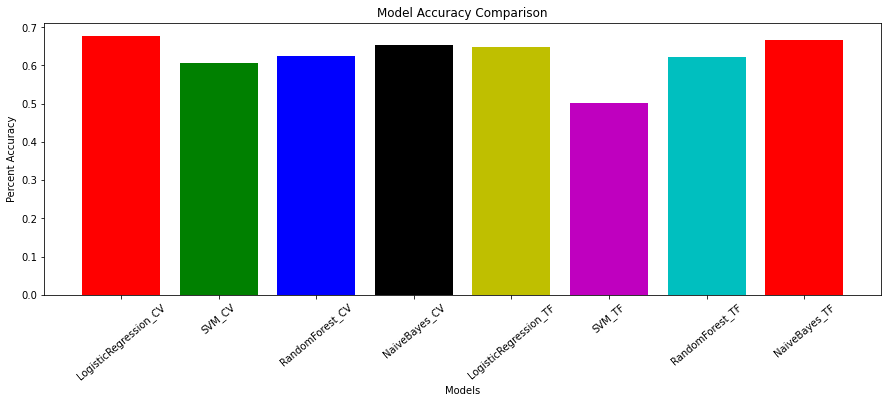
**Figure 8**. SVM Confusion Matrix Heatmap

Figure 9 depicts the top features generated from the SVM model. It appears that this model seemed to derive more insightful words for the sarcastic category based on the initial assumptions. Words such as “love,” “obviously,” and “clearly” are often used in sarcastic text. The non-sarcastic category, once again, depicts several news related words including “president” and “entertainment.”

****

**Figure 9**. Linear SVM - Top Features for Sarcasm (blue bars) and Non-Sarcasm (red bars)

To validate the results, another subset of the data with 5-fold cross validation and the same models generated the following accuracy scores, which can be seen in Figure 10.

****

**Figure 10.** Model Scores with 5-fold cross validation

Support Vector Machine with normalized data predicted labels with lowest accuracy, which was 50%, while all other models predicted labels with a very close range of accuracy between 60%-65%.

**Conclusion**

While detecting sarcasm can be a challenge for humans, it is an even bigger obstacle for computers. When humans speak, it can be easier to pick up on verbal cues, body language, and tones that convey sarcasm, which is a disadvantage of attempting to detect sarcasm through the use of computers. However, sarcasm can also be subjective in nature, which makes the overall task of detection increasingly more difficult.

In an effort to understand the complexity of detecting sarcasm through the use of computers, data was gathered from different sources including Twitter and Reddit and was analyzed. Both of these social media platforms are extremely popular with millions of users posting their opinions, reviews, and general information about any topic imaginable. The results solidify the notion that certain words are more frequently used and associated with sarcasm. For example, it was observed that words such as “love,” “obviously,” and “clearly” are usually associated with words used in sarcastic text.

Implementing the use of a computer to understand sarcasm is still considered a narrow field of research. However, it is believed that as technology continues to advance, future studies will focus on using larger, more diverse data to improve methods and accuracy of sarcasm detection.

**References**

Geoffrey Hinton Quotes. (2020). Retrieved December 09, 2020, from https://www.brainyquote.com/quotes/geoffrey\_hinton\_875276

Merriam-Webster. (n.d.). Sarcasm. In Merriam-Webster.com dictionary. Retrieved December 9, 2020, from <https://www.merriam-webster.com/dictionary/sarcasm>

Chin, R. (2011, November 14). The Science of Sarcasm? Yeah, Right. Retrieved December 09, 2020, from https://www.smithsonianmag.com/science-nature/the-science-of-sarcasm-yeah-right-25038/