**DETECTING SARCASM IN TEXT**

Calvin Choi

December 1, 2020

**Abstract**

The goal of this study is to develop a model that can accurately differentiate sarcastic text from non-sarcastic text. To obtain data for this model, user comments will be scraped from the popular social news website, Reddit. Natural Language Processing (NLP) techniques will be applied for the purpose of text preprocessing and topic modelling. This study will also train and test several supervised learning algorithms to classify sarcastic and non-sarcastic comments. The tools that will be used to accomplish this task including the popular NLP library, Spacy, as well as Scikit-Learn for classification tasks. Other analytical and visualization libraries such as Pandas and Matplotlib are utilized in this study.

**CONTENTS:**

1. **Introduction**
2. **Related Work**
3. **Data**
4. **Exploratory Data Analysis**

**5.1 N-grams of Sarcastic Comments**

**5.2 The Structure of Comments**

**5.3 Parts of Speech Tagging (POS) and Named Entity Recognition (NER)**

1. **Modelling and Evaluation**

**5.1 Text Preprocessing and Feature Engineering**

**5.2 Model Selection and Classification**

**5.3 Model Evaluation**

1. **Conclusion**
2. **References**

**1. INTRODUCTION**

The ability to understand sarcasm in published and other online content is a critical aspect in understanding the opinion and sentiment of its authors. Building a model that can automatically detect sarcasm contained in a large volume of data from websites and social media is of considerable interest to a diverse range of organizations. For example, political campaigns would utilize such models to understand the sentiment of voters, while businesses would apply these models to online reviews to analyze what customers think about their products or services. A machine learning model that can pick up on the linguistic nuances of sarcasm could allow for a more accurate analysis of text-based data. This study attempts to discern sarcastic from non-sarcastic text by transforming this problem into a classification task. Popular machine learning algorithms such as Naïve Bayes and Logistic Regression are employed to classify the text for the popular social media site, Reddit. Additional features are engineered from the raw text to provide the models with informative data.

**2. RELATED WORK**

Detecting sarcasm by way of classification is not a new concept and many studies already exist in this arena. One approach is the SCUBA (Sarcasm Classification Using a Behavioral Modeling Approach) framework which considers sarcasm as a behavioral response to certain situations, observation, or emotion, and includes all three in its analysis (Rajadesingan, Zafarani, and Liu, 2015). SCUBA was trained on tweets scraped from Twitter and includes looking into previous tweets made by the user to detect whether the user is a sarcastic person themselves.

A unique approach was also taken (Barbieri, Saggion, Ronzano, 2014), whereby the authors attempted to stay away from the vectorization of the entire text to reduce computational complexity of word vectors. Therefore, a word vector was not used for the classification task. Instead, the main approach centered around the engineered features which fall into seven categories: Frequency, Written-spoken, Intensity, Structure, Sentiments, Synonyms, and Ambiguity. Frequency features describe the gaps between common and rare words, and make the assumption that unexpectedness is a trait related to irony. Written-spoken features detect spoken style English within a mostly written style tweet, another trait which the authors associate with sarcastic comments. The Structure features encompass information such as the length of the tweet, count of various part of speech words, punctuations, etc. Intensity features look at the intensity of certain adverbs and adjectives, assuming that sarcastic comments typically exaggerate both of these items. The features in the Synonyms category attempt to capture whether a rare word is used rather than its more common synonyms, which the authors argue is a characteristic of sarcasm since the choice of word is very important for the user. Lastly, the Ambiguity features are associated with the ambiguity of the tweet. Here, the authors theorize that words with different meanings have a higher possibility of meaning something other than what is said.

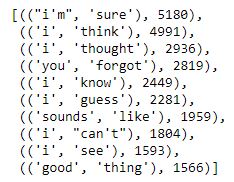
**3. DATA**

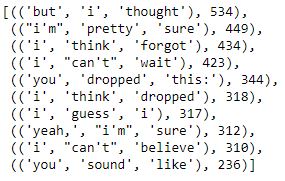
The data used in this study includes a pre-labelled dataset downloaded from Kaggle. As per the author, the dataset contains sarcastic and non-sarcastic comments, labelled as 1 and 0 respectively, that was scraped from Reddit. To accurately identify sarcasm in the comments, the author relied on the Reddit-specific tag /s which users typically utilize to designate sarcasm in their comment. The dataset includes 1,010,774 comments and attributes related to each comment, such as the number of likes, the date it was made, the parent comment, etc. In addition, the original dataset includes a perfect class balance between sarcastic and non-sarcastic comments. Because the models created in this study is intended to be used on text for any source, all Reddit specific data is removed, leaving only the comment. Note, in the modelling stage, the number of instanced is reduced to 50,000 (split evenly between classes) to reduce computation expense and reduce time.

**4. EXPLORATORY DATA ANALYSIS**

This section focuses on the exploratory data analysis (EDA) that was conducted to understand the underlying data in more detail. The insights gathered from this section also drove decisions in how to engineer additional features to feed into the predictive modelling. The EDA in this study can be categorized into three main areas: analyzing the N-grams generated for the sarcastic class, the structure of the comments, and analysis on the parts of speech (POS) and named entity recognition (NER) of the comments.

**4.1 N-grams of Sarcastic Comments**



A list of top ten bigrams and trigrams for sarcastic comments is generated to understand common components of sarcastic text. Figure 1 displays the most common occurring sarcastic bigrams and trigrams in the corpus. That is, any sarcastic comment in the corpus has a high chance of containing these two or three consecutive words. The most common consecutive two words for the sarcastic comments in the corpus are *“i’m sure”* which occurs 5,180 times. Similarly, the most common consecutive three words are “*but i thought*” which occurs 534 times. The other bigrams and trigrams presented in this list are also commonly associated with sarcastic phrases.

Because sarcastic comments have a high probably of starting with, or containing, these consecutive words it may therefore be worthwhile to train a model using bigrams or trigrams as an input in addition to the bag-of-words approach with unigrams.

Figure 1: Top 10 bigrams and trigrams for sarcastic comments.

**4.2 The Structure of Comments**

As in the study, *Modelling Sarcasm in Twitter, a Novel Approach (Rajadesingan, Zafarani, and Liu, 2015)*, the structure of the text can provide useful information that would help a model discern sarcasm from non-sarcasm. In this section, the focus was on the overall length of the comment (the number words), punctuations, special characters, repeating characters, as well as features associated with humour, i.e. ‘*lol’*, ‘*haha’*, and smiley emoticons.

When observing the length of each comment, there were a few notable differences between sarcastic and non-sarcastic comments. Short comments, between 1 to 5 words in length, tend to skew heavily toward being sarcastic. After that, comments between the length of 5 to 20 words skew more toward being non-sarcastic. Comments that are above 20 words in length appear to be equally distributed between sarcastic and non-sarcastic.

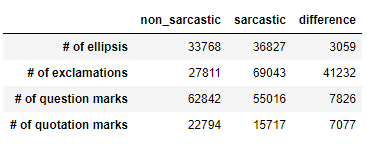
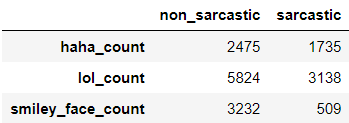


Figure 2: Number of comments that contain punctuation mark, split by sarcastic and non-sarcastic.

Four types of punctuations are of focus in this study: ellipsis, exclamation marks, question marks, and quotation marks. Figure 2 illustrates the distribution between sarcasm and non-sarcasm with respect to the count of punctuation marks in each class. The difference is also calculated for each. Both ellipsis and exclamation marks tend to associate more with sarcastic comments. Conversely, question and quotation marks are more inclined to be included in non-sarcastic comments. The difference in count between classes is significant for the number of exclamation marks, with a difference of 41,232. The difference in the number of ellipsis is relatively low, however it appeared that as the number of ellipsis increases from 1 to 2, and so on, the proportion of sarcasm to non-sarcasm also grew.

The number of repeating consecutive characters in a word was also analyzed for any difference in sarcasm and non-sarcasm. For example, consider the following comment taken from the corpus: *"Oooo you're so edgy putting that here".* Here, the count of repeating consecutive characters is 4. If a comment contains two words with repeating consecutive characters, only the word with the highest number of repeating characters was considered. As such, the study found that the distribution of comments that contained repeating characters skewed more toward sarcasm.



Lastly, analysis was also conducted on features that reflected humor, such as the words ‘*lol’* and *‘haha’* as well as the smiley face emoticon. In figure 3, it is apparent that there is only a notable difference in distribution for smiley face emoticons.

Figure 3: Number of comments containing features that describe humor, split by class.

**4.3 Parts of Speech Tagging (POS) and Named Entity Recognition (NER)**

Using the NLP library, spaCy, each comment in the corpus was individually parsed and its contents were annotated with POS and NER tags. The POS tags essentially indicate which parts of speech each word belongs to, i.e. noun, verb, adjective, interjection, etc. NER tags accomplish a similar task but with named entities, i.e. person, companies, political groups, product, etc.

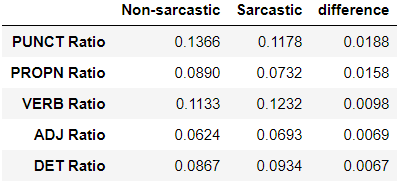
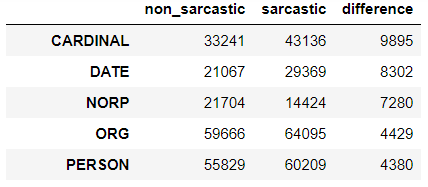
 After the text had been appropriately annotated with POS tags, each comment was observed by taking the ratio of each tag versus other tags within the comment. As per figure 4, the difference in ratios between both classes is analyzed and the top five POS tags with the highest difference are punctuations, pronouns, verbs, adjectives, and determiners. Sarcastic comments tend to contain a higher ratio of verbs, adjective and determiners. While non-sarcastic comments have a higher ratio of punctuations and pronouns.

Figure 4: Ratio of POS tag in each comment, filtered for top five in difference between class.

Figure 5: Count of NER tags, filtered for top five in difference between class.

For NER tags, this study simply computes the aggregation of each tag and groups them by class. The reason for this is, unlike POS tags, many comments contain only one NER tag so any ratios derived from these comments would be meaningless. The top five NER tags as shown in figure 5 are CARDINAL (numbers), DATE, NORP (nationalities, religious or political groups), ORG (companies, agencies, institutions, etc.) and PERSON (includes fictional). With the exception of NORP, all other tags fall under the sarcastic class.

**5. MODELLING AND EVALUATION**

To train the models in predicting sarcastic comments, the label column which labelled the comments as 1 for sarcastic and 0 for non-sarcastic was used as the target dependent variable. The comment column was used as a dependent variable. Using the insights gathered in the EDA stage, various text preprocessing and feature engineering steps was conducted using the ‘comments’ feature to prepare the text for modelling, as well as provide the model with informative data. Note, due to computational expense, the original dataset of around 1.1 million instances has been reduced to 50,000 for this section of the study.

**5.1 Text Preprocessing and Feature Engineering**

To classify the raw text, the bag-of-words modelling approach is used to transform the corpus into a word vector. Rather than using only unigrams, bigrams are also generated from the corpus to create a bag-of-ngrams, thus, preserving some of the context in the text. The text was also normalized prior to vectorization. That is, all comments were stripped of punctuations, converted to all lowercase, and numbers that were in integer format was converted to text numbers to standardize all comment to a level playing field. Lemmatization was then applied across the corpus. Lemmatization attempts to reduce words to a common base form. By making use of vocabulary (through the usage of POS tags) and morphological analysis, lemmatization not only reduces words, but ensures the context is not lost. For example, for the word “meeting”, lemmatization may return the word “meet” or “meeting” depending on the context. After text normalization and lemmatization, the preprocessed text is then transformed in a bag-of-unigrams and bag-of-bigrams. Tf-idf weights are also applied to each word, which determines its importance in the overall corpus.

Along with the bag-of-ngrams, additional features are also engineered enhance the predictability of the models. Based on the information uncovered in the EDA stage, the additional features engineered include the length of the comment (by words), the number of ellipsis, the number of exclamation marks, the number of question marks, the number of quotation marks, the number of smiley face emoticons, and the number of repeating characters. The text preprocessing and feature engineering steps are bound together in the pipeline which is then fit onto the data to create a sparse matrix at the time of training for each classifier. This allows the above steps to be completed consistently on the training and testing data, as well as new, unseen data.

**5.2 Model Selection and Classification**

In the model selection stage, five classifiers are compared: Multinomial Naïve Bayes, Logistic Regression, Decision Tree, Linear Support Vector Machines, and Gradient Boosting. These algorithms are 5-fold cross-validated using only the bag-of-ngrams as input to serve as benchmark models. The metrics used for evaluation at this stage is the f1-score for class ‘1’, i.e. the sarcastic comments. Accuracy is also assessed, but not considered as importantly as the f1 score since sarcastic comment are rarer in a real-world setting. The results are shown below in figure 6:

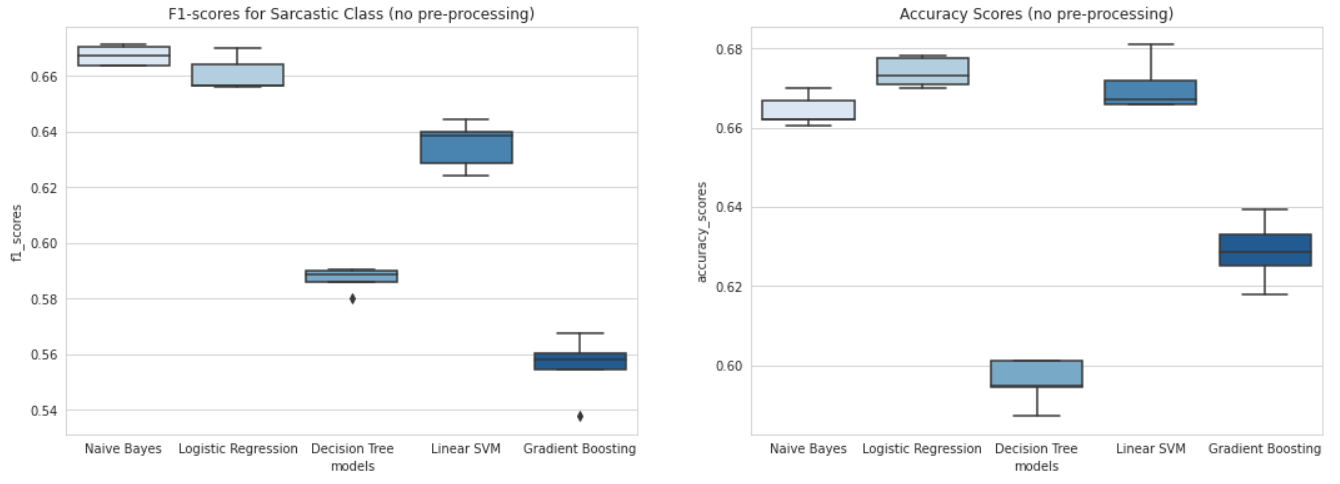
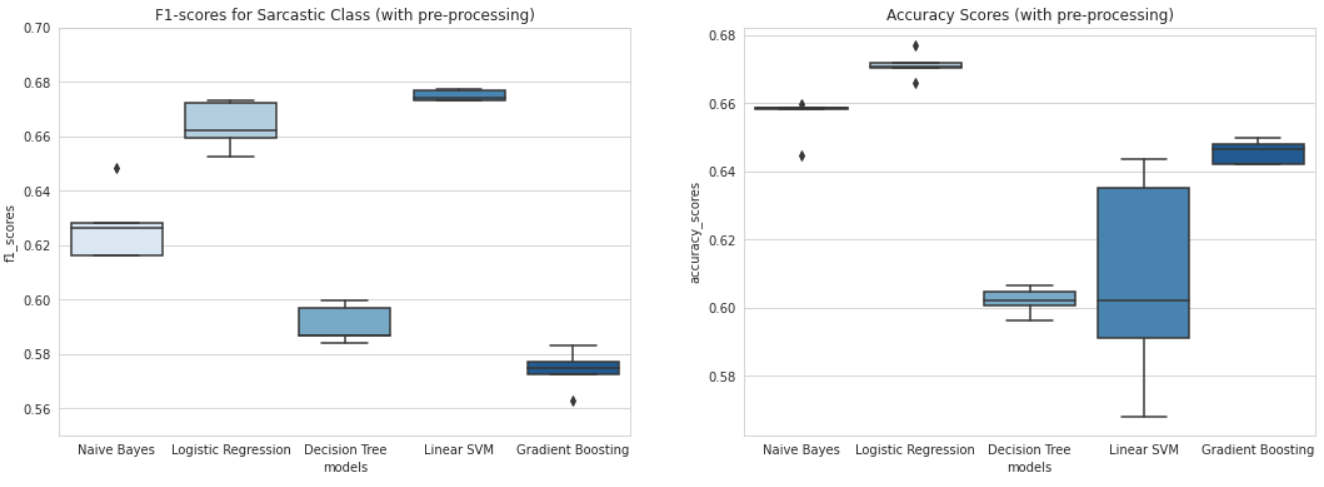


Figure 6: Boxplots for F1-scores (of sarcastic class) and Accuracy scores for each classifier.

Boxplots are used to visualize each the scores of each classifier. Based on f1-scores of the sarcastic class, Naïve Bayes and Logistic Regression appear the be the strongest models with scores around 66-68%. Linear SVM does not score as high on f1-scores, but its accuracy is comparable to Naïve Bayes and Logistic Regression. In fact, there is one instance where it scored the highest accuracy among all five models. Both Decision Trees and Gradient Boosting do not appear to perform well.

Next, the preprocessing and feature engineering steps mentioned in the previous section are bound with each of the classifiers in individual pipelines. The same 5-fold cross validation is conducted on each pipeline and results are shown in figure 7 below:

Figure 6: Boxplots for F1-scores and Accuracy scores for each classifier after preprocessing and feature engineering.

With the additional text processing and features, it can be observed that only the Logistic Regression model benefitted from these steps. The Linear SVM model scored much higher f1-scores, however the accuracy appears to have a large variance in performance, which may indicate that it is predicting only one of the classes accurately. Based on these observations, the Naïve Bayes and the Logistic Regression models are shortlisted for further investigation. A hold-out set was created for the purpose of testing the shortlisted models on unseen data. Figure 7 presents the results of those tests using the confusion matrix:

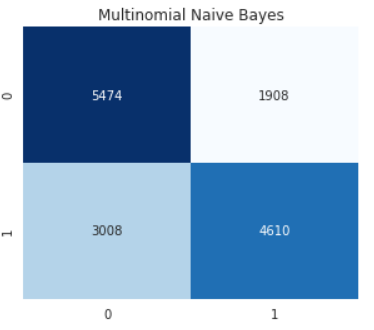


Figure 7: Confusion Matrix for predictions made by Naïve Bayes and Logistic Regression on testing set.

With respect to sarcastic comments, it can be observed that the Naïve Bayes model was able to predict 4,610 of the actual 7,618 sarcastic comments in the testing set, giving it a recall of 60.5%. In addition, out of the 6,518 predictions it made for the sarcastic class, it was correct 4,610 times, giving it a precision of 70%.

The Logistic Regression model predicted 5,075 of the 7,618 cancellations (66% recall), and out of the 7,220 prediction it made for the sarcastic class, it was correct 5,075 times (70% precision).

**5.3 Model Evaluation**

As a final evaluation measure, the ROC curve for the Multinomial Naïve Bayes and Logistic Regression model is graphed and the ROC-AUC scores for each are calculated. For a given set of predictions, the ROC for a model is the true positive rate against the false positive rate. The true positive rate is essentially the ratio of positive instances that were actually classified as positive, while the false positive rate is the ratio of negative instances wrongly classified as positive. The ROC-AUC score summarizes each curve by measuring the area under it - the higher the score, or the closer the curve is to the top-right corner of the graph, the stronger the

classifier.

In figure 8, the dotted line in the graph depicts a hypothetical purely random classifier with an AUC of 0.50. This line is used as a benchmark to visualize the strength of the other models, and how well it compares to the purely random model. It can be observed the both the Naïve Bayes and Logistic Regression models are similar in strength. Because the Logistic Regression model has a curve closer to the top-right corner, it is considered a slightly stronger model. In addition, the AUC score calculated for this mode is 0.731, while the Naïve Bayes model scored 0.731. Therefore, it is evident that Logistic Regression is the superior model for this task.

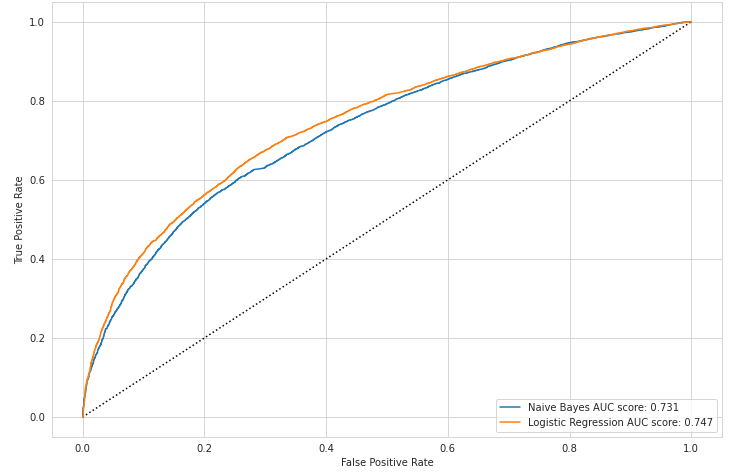
**6. CONCLUSION**

Figure 8: ROC curve and ROC-AUC scores for Naïve Bayes and Logistic Regression.

A corpus of comments from Reddit was used to help train and test a selection of classifiers to predict whether a given comment was sarcastic or not. Through exploratory data analysis, several traits belonging to sarcastic comments was uncovered. Sarcastic comments tend to be shorter in the number of words. They tend carry more ellipses and exclamation marks, but fewer question marks and quotation marks. Bigrams and trigrams were also extracted to view common two and three consecutive words associated with sarcasm, the most common being *“i’m sure”* and *“but i thought”.* Using these insights, additional features were created to accompany the bag-of-ngrams that was fed into five basic models. 5-fold cross-validation was utilized to shortlist two models; Multinomial Naïve Bayes and Logistic Regression. There models were then tested on a testing set and its recall and precision were evaluated. Logistic Regression performed slightly better with a recall of 66% and precision of 70% when predicting on the sarcastic class. The ROC-AUC score also suggest the Logistic Regression is the strongest model.

Despite Logistic Regression being the strongest model, its predictive accuracy is not strong enough to be utilized in a real-world setting. With a mere recall of 66%, it would not be feasible to employ this model to predict whether reviews of a product are sarcastic or can be taken literally, or whether the sentiment of voters in a political campaign truly have a positive outlook on the party.

**7. REFERENCES**

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu, (2015). Sarcasm Detection on Twitter, A Behavioral Modeling Approach

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano, (2014). Modelling Sarcasm in Twitter, a Novel Approach

Natalie Parde and Rodney D. Nielsen, (2018). Detecting Sarcasm is Extremely Easy ;-)

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang, (2013). Sarcasm as Contrast between a Positive Sentiment and Negative Situation

Tomas Ptacek, Ivan Haberbal, and Jun Hong, (2014). Sarcasm Detection on Czech and English Twitter