1. **Project Title and Team Members**

<https://github.com/dldowning/Fall2022-5222>

Analysis of Impact on F1 Score of Permutations of Tweet Regular Expressions for Sentiment Analysis

David L Downing

Justin Roberts

CSCE 5222 Feature Engineering - Dr. Sayed Shah

Fall 2022 University of North Texas

1. **Intro**

The goal of sentiment analysis is to identify and classify expressed opinions, usually in text form. Twitter, as a platform, encourages people to share their thoughts and opinions about any topic of interest to them. This framework means that Twitter can be an incredibly useful source for generating vast amounts of sentiment data. (TweetEval2020)

Sentiment analysis models are not new to the realm of machine learning. We have seen sentiment analysis used for marketing research, understanding brand reputation, alerting to potential security risks, and much more. As the field of sentiment analysis has expanded, we have seen a vast improvement in the accuracy and consistency of models. An element that has not received as much attention though, is what specific aspects/ features of input text should be considered when creating a sentiment analysis model.

In our project we aim to explore the significance of individual text features in a sentiment analysis machine learning model. We want to understand how text features that we engineered, can impact model predictions and other performance metrics. The results of this assessment can be useful in shedding light on what aspects of text may be most indicative of its sentiment classification. These results, in combination with more sophisticated NLP sentiment analysis models, may help to create even more robust, state of the art, models.

1. **Background**

The study of Natural Language Processing is quite intricate. There are a variety of language models and Machine learning algorithms that can be combined to create accurate text classification programs. Just in the past 5 years, we have seen the emergence of language models like BERT, GPT-3, XLNet, RoBERTa, and many others applied to word prediction, text classification, and even speech recognition. (Giachanou 2017)

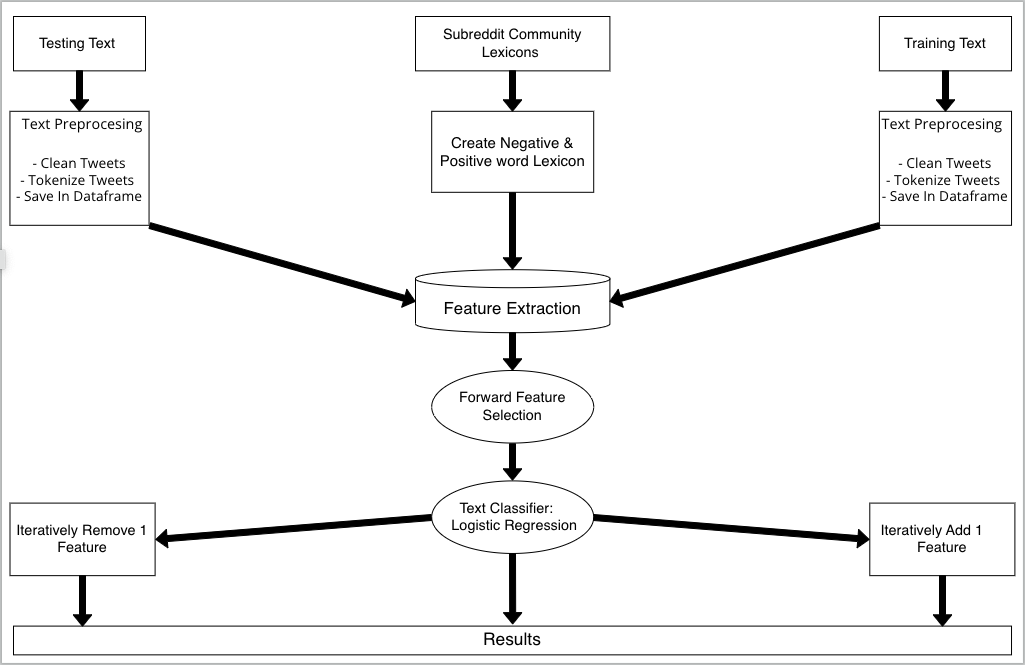
The inspiration for our project stems from a research initiative organized by the Cardiff University NLP Research Group. The initiative titled, “TweetEval: Unified Benchmark and Comparative Evaluation”, is a platform that allows for a unified research effort by providing standardized data for researchers to test their models on. This research platform also allows researchers to compare results without having to account for inconsistencies in data. In the comparative evaluations we see that two of the previously mentioned models, BERT and RoBERTa, were two of the best performing models when it comes to classification. (TweetEval 2020)

What is not provided though, is an explanation as to why these models performed better than others. We aim to explore how features will affect the prediction accuracy and attempt to explain why these changes occur. For the sake of simplicity, we decided to do this feature analysis on a logistic regression model because it gives us the greatest level of control in generating and selecting the features that will be used.

1. **Model**

Our classification model is based upon a logistic regression classifier. We selected this model, because of its consistency of output, ease of use, and it is an optimal algorithm for binary classification . This was done in effort to lower the number of variables to account for when considering the impact of individual features. A more complex model may have yielded better classification results but would have also complicated our main focus, feature analysis.

The following architecture diagram visualizes the execution pattern of all aspects of our model.



Our model starts off with 3 input sources, a training text dataset, multiple subreddit datasets, and a test text data set. After importing all of our data, the training and test data set are preprocessed into a format suitable for our machine learning algorithm. The subreddit data is aggregated into two lexicons, one of all positive words and one of all negative words. How these lexicons are created will be explained in greater depth in the next section. From here, our model performs the feature extraction. Here is where the input texts are analyzed for qualities like, the count of adjectives in a tweet or what type of punctuation a tweet contains. (Zhang 2011)

After extracting all of our features, we use the SkLearn, Forward Features selection method. This method is used to assign importance to each feature through a specific attribute. With this information we were able to identify our most important features.

Lastly we perform the feature analysis. We have 2 methods of analyzing our model features. The first is to take our entire list of features and iteratively remove exactly one feature and record how the removal of said feature affects the model when all other variables are held constant. The second method of feature analysis is to remove all of the best features identified by the forward feature selection and iteratively replace exactly one of them at a time. Here we will also record how this affected the model when all other variables are held constant.

1. **Dataset**

The data driving this project is sourced from two locations. The first being the Cardiff University NLP Research Group. This group provided the training and test dataset of tweets. The provided tweets are unedited, with varying content, dialects, and word choice. The sets provided included over 60,000 tweets in total. (TweetEval 2020)

The second data source is provided by a Stanford NLP research group. This data is a collection of community specific lexicon data from 250 of the largest subreddit communities on reddit.com. Each lexicon contains sentiment values for the top 5000 non-stop words in each community. The data also provides two sets of historically common adjectives from the years 1850-2000 and from 2000 onward. This data is what was used to put together our lexicons of positive and negative words which are used in feature extraction for our model.

The features of our model were engineered in a way that would allow us to see a wide range of feature importances and impacts on the model. Some of the features are designed with the intent to show that they have minor impact while others are anticipated to show as having a great impact on the model. The features are as follows:

List of Features

| 1. Average Sentiment Score of word #1 in tweet | 11. Log of Length of Longest Word in Tweet | 21. Ratio of capital letters to lowercase |
| --- | --- | --- |
| 1. Average Sentiment Score of word #2 in tweet | 12. Log of Count of Words With 5+ Characters | 22. Ratio of punctuation characters to total characters. |
| 1. Average Sentiment Score of word #3 in tweet | 13. Count of words in tweet in positive lexicon | 23. Does the tweet contain the word no |
| 1. Average Sentiment Score of word #4 in tweet | 14. Count of words In tweet in negative lexicon | 24. Mean length of words in tweet |
| 1. Average Sentiment Score of word #5 in tweet | 15. Count of Nouns in the tweet | 25. Log mean length of words in tweet |
| 1. Average Sentiment Score of word #6 in tweet | 16. Count of Adjectives in the tweet | 26. Value of highest sentiment score token in tweet |
| 1. Average Sentiment Score of word #7 in tweet | 17. Ratio: Count of unique words in tweet to total words in tweet | 27. Value of lowest sentiment score token in tweet |
| 1. Average Sentiment Score of word #8 in tweet | 18. Ratio: Count of stop words in tweet to total words in tweet | 28. Count of emojis in tweet |
| 1. Average Sentiment Score of word #9 in tweet | 19. Ratio: Count of nouns to total words in tweet |  |
| 1. Log of the word count of the tweet | 20. Ratio: Count of proper nouns to total words in tweet |  |

1. **Analysis of Data**

Of the most common data types used in machine learning models, text data is generally considered the most unstructured. We see this because writers are not required to work within any structural guidelines. Use of slang terms, incorrect grammar structure, misspelled words, and a myriad of other structural inconsistencies can make text difficult to work with.

This issue is more pronounced when looking at text data sourced from Twitter. The platform only allows posts to have a maximum of 140 total characters. This means that users oftentimes must condense and intentionally misspell words to convey their message in the limited character space. This poses a unique challenge to NLP models using twitter data because the models are trying to make sense of a unique amalgamation of complete and incomplete words.

Preprocessing is incredibly important for models like ours because we need to establish consistency among tweets to have a chance at yielding accurate results from our classifications. Our data processing goes as follows:

1. Clean Tweets:
   1. Make All Characters Lowercase
   2. Remove all Non-Alphanumeric Characters
   3. Remove Repeated Words
   4. Remove Single Character words (I,A, etc…)
   5. Remove Usernames from tweets
   6. Remove Emojis from tweets
2. Tokenize Tweets
   1. Most NLP tasks operate and create understanding of words using individual tokens rather than whole input strings

Text

Description automatically generated

1. **Implementation**

The algorithm powering our feature analysis investigation is logistic regression. Logistic regression is ideal in this use case because of its simplicity and consistent performance in classification tasks. Our Implementation of logistic regression is written from scratch. This design choice was made to give us, as users, more control in hyperparameter tuning. This also allows readers and us, the authors, to gain a more robust understanding of the algorithm, and how specific parts of it interact with the data to make our various predictions.

**A picture containing text

Description automatically generated**

Walkthrough of Implementation Steps

1. Import All Libraries and Datasets
2. Create data frames and dictionaries to store imported data
3. Clean Data
4. Perform Feature Extraction
5. Perform Forward Feature Selection and record which features are found to be most significant
6. Implement logistic regression from scratch and store results for baseline comparison
7. Add 1: Feature Analysis

Remove all of the most significant features found in step 5 from the model. One at a time, add back only one of the most significant features to the model, run the logistic regression, and collect results on how that metric alone changed prediction accuracy and other performance metrics.

1. Remove 1: Feature Analysis

One at a time, remove an individual feature, run the logistic regression, and collect results on changes in prediction accuracy and other performance metrics. We are interested in evaluating how the removal of one feature will affect the Sentiment analysis model when all other variables are held constant.

1. Gather and compare results
2. **Results**

Forward Feature Selection Results:

Forward feature selection found that our 12 most significant features would be features 11,12, 15, 16, 17, 18, 19, 20, 23, 24, 25 and 28. (SKlearn SelectFromModel)

Text

Description automatically generated

Add 1: Feature Analysis Results

The below output is repeated 12 times for each of the most significant features from our forward feature selection.

Text

Description automatically generated

Remove 1: Feature Analysis Results

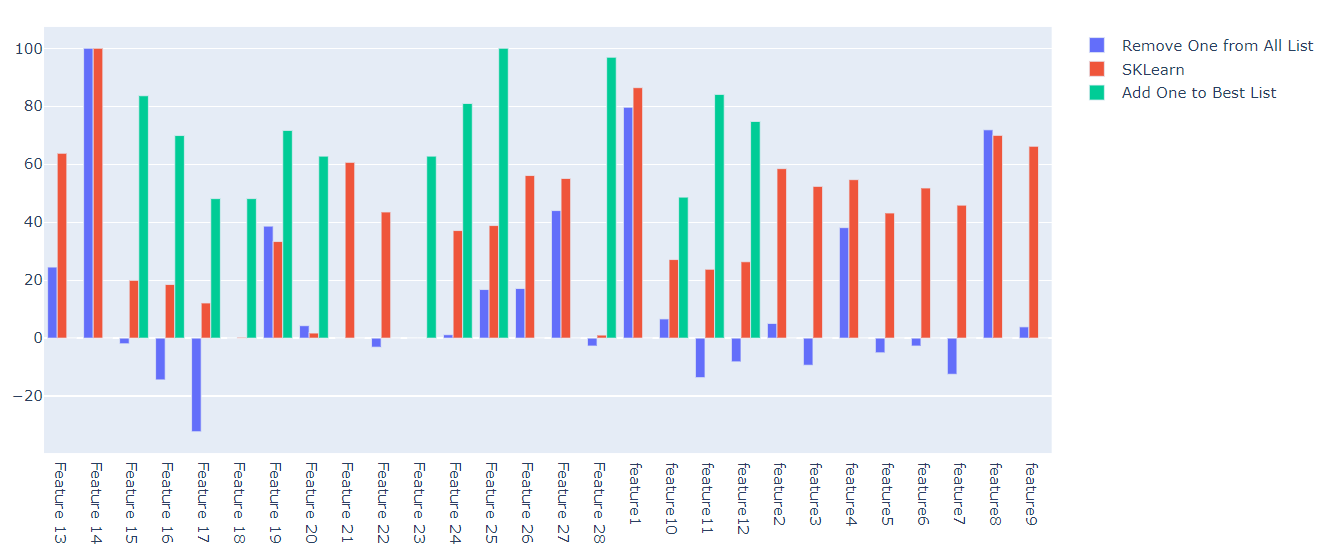
We have a similar output for each instance of one feature being removed and running the logistic regression.

Text

Description automatically generated

All Results Diagram:

Lastly, in the following figure, we have a bar graph representation of the calculated significance of each feature. Each bar cluster, (consisting of 2 or 3 bars) is a representation of which analysis the feature was used in and how great of an impact that feature had on the classification task.



Feature Ranking & Hypothesis

1. Feature 14
2. Feature 24
3. Feature 28
4. Feature 1
5. Feature 15
6. Feature 11
7. Feature 24
8. Feature 12
9. Feature 13
10. Feature 19
11. Feature 16
12. Feature 9
13. Feature 8
14. Feature 20
15. Feature 21
16. Feature 23
17. Feature 2
18. Feature 4
19. Feature 26
20. Feature 27
21. Feature 3
22. Feature 6
23. Feature 18
24. Feature 17
25. Feature 5
26. Feature 7
27. Feature 10
28. Feature 22

**Sorted by max relative importance across all metrics. Ranking is ordinal but increments may be minimal.**

Feature 14

Rank #1 - According to our model, feature 14, the “Count of words from tweet, in the negative lexicon” is seen as one of the most impactful features. From a linguistic point of view this makes perfect sense. If a sentence contains negative words it is very likely that the sentence will be negative.

Feature 24

Rank #2 - Feature 24, the ‘mean length of words in the tweet” is ranked 2nd in feature importance. We predict that this is indicative of a tweets classification because on twitter we tend to see serious tweets fully articulating words and spelling things fully while more light hearted, jovial tweets use shorter words and abbreviations.

Feature 28

Rank #3 - The “count of emojis in the tweet”. Ranked 4th in feature importance, emojis are more commonly used to communicate positive feelings, our model likely recognized this pattern and used it as a heavily weighted metric in classification.

Features 1-9\*

Rank #4 - (Rank 4, 12, 13, 17, 18, 21, 22, 25, 26) - Understanding the discrepancies in overall importance for features 1-9 is a challenging task. These features were all generated by averaging the Sentiment score of the ith(1-9) word in the tweet across a given subset of dataframes containing sentiment values. Some of these subsets may have been better suited to certain words than others and that is why we see such drastic differences in feature importance.

Feature 15

Rank #5 - The “count of nouns in the tweet”. Ranked 5th in feature importance, increased use of nouns, and proper nouns in particular may indicate that someone is attempting to communicate strong feelings about a particular topic.

Feature 11

Rank #6 - , the “Log value of the length of the longest word in the tweet”. From a language analysis standpoint it is not quite clear why the log value of the longest word of a tweet can be useful in classifying it as positive or negative. This is partially the purpose of machine learning though, to recognize patterns in data that humans can’t easily see.

Feature 24

Rank #7 - Similar to the previous feature, The “Mean length of words in the tweet” is not easily recognizable as something that would indicate a tweets sentiment, but from a mathematical standpoint this is clearly something quite important to a ML model.

Feature 12

Rank #8 - Next in feature importance the “Log value of the count of words with 5+ characters” is also not easily explained by humans but mathematical expressions powering our model can draw clear correlations between this feature and tweet sentiment.

Feature 13

Rank #9 - Similar to feature 14, this feature Counts the number of words in the tweet that we see in our lexicon of positive words. Much like negative words a direct correlation can be drawn between the amount of positive words used in a tweet and its likelihood to be classified as positive.

Feature 19

Rank 10 - We have the “Ratio of nouns to total words in the tweet”. We saw another noun related feature in the top 10 most impactful features so there may be a substantial correlation between noun usage and sentiment.

Feature 16

Rank #11 - Next in feature importance, “The Count of Adjectives in the tweet” could be indicative of a certain sentiment simply using more descriptive words means there is more emotion being communicated.

Feature 20

Rank #14 - The “Ratio of proper nouns to total words in the tweet” is 14th in feature importance. This is our second noun related feature and we saw earlier that there is indeed a correlation between noun usage and sentiment.

Feature 21

Rank #15 - The “Ratio of capital letters to lowercase letters”, is our 15th most impactful feature. The intent behind this feature was to try and understand if capitalization is an indicator of sentiment.

Feature 23

Rank #16 - Then, in model importance, “Does the tweet contain the word ‘no’?” We believed that there would be a stronger correlation between this feature and sentiment prediction but the word no is not as indicative of positive and negative emotions as we originally thought.

Feature 26

Rank #19 - The value of “Highest sentiment score in tweet”, was selected as a feature because we believe that the highest sentiment score of words is likely correlated to the tweets sentiment. If a tweet has a highly positive word we believe the tweet is likely to be considered positive, and the same vice versa.

Feature 27

Rank #20 - Much like the feature before it, the “Lowest Sentiment score in tweet” was believed to show a correlation between score and sentiment.

Feature 18

Rank #23 - Following in importance, the “Ratio of stop words to total words in the tweet” gives the model an understanding of the role stopwords play in conveying emotion. We believe that stopwords, while having a mathematical impact of sentiment prediction, in real life do not have much of an effect.

Feature 17

Rank #24 - The “Ratio of unique words to total words in the tweet”. Ranked 24th in feature importance, we think this is so because much like the previous feature there is a mathematical significance but not many other conclusions can be drawn by a human about this feature.

Feature 18

Rank #25 - The “Ratio of punctuation characters to total characters”, is our least impactful feature. The intent behind this feature was to try and understand if commonly used punctuation like exclamations and question marks are indicators of sentiment.

Feature 10

Rank #27 - The log of the word count of the tweet. This is apparently fairly random and appears like noise.

Feature 22

Rank #28 - Ratio of punctuation characters to total characters

Some of the rankings might be surprising and some might not, but using quantitative methods to evaluate features is superior to using qualitative methods.

1. **Project Management**

Work Completed:

* + Loading Data, creating initial Data frames, and creating lexicon dictionaries – 50/50
  + Data Preprocessing – Justin & David
  + Feature Extraction – Justin
  + Forward Feature Selection – David
  + Logistic Regression from Scratch – Justin & David
  + Add 1 Feature Analysis – David
  + Remove 1 Feature Analysis – David
  + Clean and comment codebase – Justin
  + Add Additional Visualizations of features - Justin & David

Spring 2023:

* + Extend and clarify the notebook
  + Author paper on the topic
  + Justin takes Directed Study with focus on feature paper
  + David takes Data Visualization and uses feature paper for project

1. **References**

1. [TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification](https://aclanthology.org/2020.findings-emnlp.148) (Barbieri et al., Findings 2020)

2. Giachanou, Anastasia and Fabio Crestani. 2016. Like It or Not: A Survey of Twitter Sentiment Analysis Methods. ACM Comput. Surv. 49, 2, Article 28 (June 2017), 41 pages.<https://doi.org/10.1145/2938640>

3. Kouloumpis, E., Wilson, T., & Moore, J. (2021). Twitter Sentiment Analysis: The Good the Bad and the OMG!. Proceedings of the International AAAI Conference on Web and Social Media, 5(1), 538-541. Retrieved from<https://ojs.aaai.org/index.php/ICWSM/article/view/14185>

4. Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). Combining lexicon-based and learning-based methods for Twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, *89*, 1-8.

<https://github.com/cardiffnlp/tweeteval/tree/main/datasets/sentiment>

<https://nlp.stanford.edu/projects/socialsent/>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html>

<https://computationalsocialnetworks.springeropen.com/articles/10.1186/s40649-020-00080-x>