FINAL PROJECT REPORT

On

Sentiment Analysis using Twitter Data

GROUP 2

Ben Wachtel

Choijamts Bataa

Kishan Rathor

Saikumarreddy Pochireddygari

Department of Information Studies, Syracuse University

IST707: Applied Machine Learning

Under Prof, K King

04/30/2023

We, Group 2, hereby declare that the project report submitted is entirely our own work and has not been plagiarized in any way. Any similarities to any material used for reference and study may exist, but we assure you that the project report has been completed together as a team.

Introduction, Research Problem, and Benefits:

Vaccines play a crucial role in preserving the public health of a large population, having successfully slowed or eliminated the spread of numerous communicable diseases such as smallpox, polio, measles, and malaria (World Health Organization [WHO], 2021). The Centers for Disease Control and Prevention (CDC) recommend that everyone should receive regular immunizations to prevent the spread of diseases (CDC, 2022). However, some countries have lower vaccination rates than others, either due to vaccine accessibility or misinformation. The "anti-vaxxer" movement has caused hesitancy towards the COVID-19 vaccine, and this misinformation existed before the vaccine was even available. While vaccine hesitancy beliefs have existed for some time, the use of social media and machine learning techniques such as natural language processing (NLP) allows for the categorization of public sentiment towards vaccines on social media. Analyzing this sentiment can be beneficial to public health organizations, as it can assist in identifying concerns or misconceptions about vaccines. Furthermore, this analysis can help these organizations improve their vaccine communication strategies so that negative sentiment is minimized as much as possible. Overall, this sentiment analysis can lead to better public health outcomes by helping public health organizations evaluate their communication strategies and make modifications if necessary.

Research Questions:

1. What type of NLP techniques should we perform to classify the tweets?
2. What type of ML algorithms should we choose to classify the tweets?
3. What type of evaluation metrics should we choose?

Data:

The data used for this study was obtained from ZINDI, a non-profit platform that hosts data science competitions. The dataset contained 10,001 observations with four features, including a unique tweet ID, the safe_tweet column containing the text of the tweet with redacted usernames and URLs, the label of the tweet, and the agreement. The label of the tweet was classified as either negative, neutral, or positive, with corresponding values of -1, 0, or 1, respectively. The agreement score was used as a sanity check, where tweets were scored by three individuals, and the average score was calculated.

Additionally, a second dataset was provided by Professor King and contained 20,000 tweets related to the COVID vaccine, collected on November 18th, 2020. This dataset had seven features, including the tweet ID, link to the tweet, screen name of the user, text of the tweet, and the type of

tweet (original, retweet, or reply). The first 10,000 records in this dataset were manually labeled by the professor and used as the training dataset. The remaining records were used as the test dataset and were also manually labeled by the professor.

Data Cleaning:

Prior to data preprocessing and model training, the Zindi data and Twitter data were cleaned. The label value was changed from a numeric -1, 0, 1 to a categorical 0, 1, 2 value, as certain classification models encounter difficulties or errors when negative values are present in the label. Two null values were observed and subsequently removed, resulting in a reduction of the dataset from 10,001 to 9,999 observations. In addition, 307 duplicate observations were found in the training data. To address this issue, the first instance of this duplicate record was retained, while the remaining 306 repeat records were removed. This resulted in the Zindi dataset containing 9,692 observations.

There were no null or duplicate values in the professor's data. However, sentiment was determined based on polarity scores provided in the labels. If the positive score was greater than the negative score, the tweet was categorized as positive; if the negative score was greater, the tweet was categorized as negative; and if the positive and negative scores were equal, the tweet was categorized as neutral.

Data Preprocessing and Feature Engineering:

Prior to creating and running models, the data went through several stages of preprocessing, including the removal of unnecessary text, feature engineering, and numerical vectorization. To remove certain text, example tweets were examined to identify characters and strings that needed to be removed. The data contained user and URL tags added by Zindi to keep the data private. Hashtag characters (#) were removed, emojis were deleted, contractions were split into two words, stop words were removed, observations with non-English characters were excluded, and text was converted to lowercase. These processes were carried out on both the Zindi data and data provided by Professor King. Below are two examples of tweets, one from each dataset, that illustrate the need for this cleaning process, as they contain usernames, user tags, and other unnecessary text.

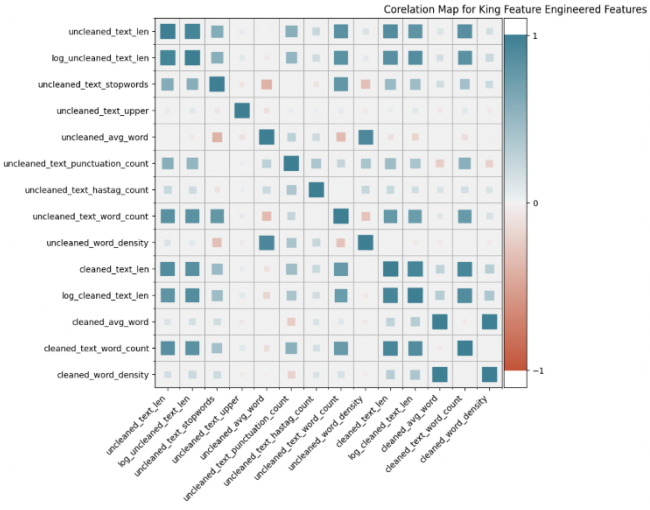
```
1 raw_train_data.safe_text.values[100]
```

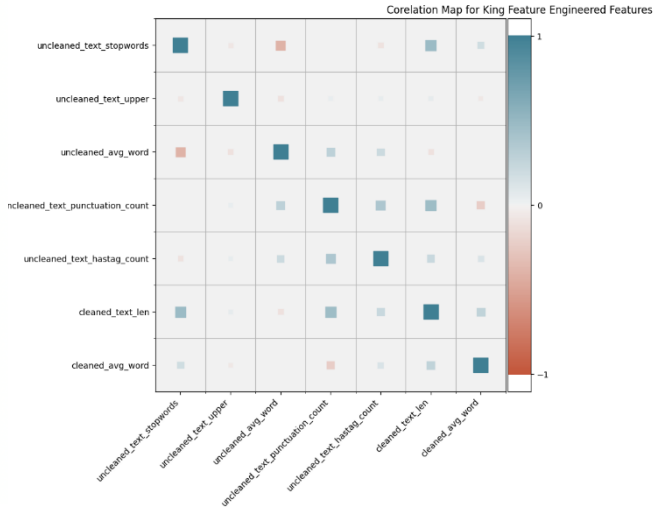
```
'"<user> Conservative Neurosurgeon Ben Carson Says Vaccines Are A Public Health Issue <url> 1 thing I agree with him on.'
```

```
1 raw_train_data_king.safe_text.values[2003]
```

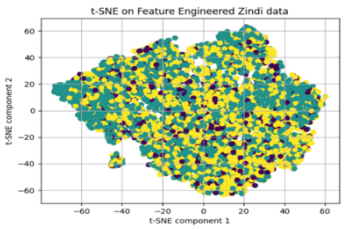
```
'RT @AP: BREAKING: Pfizer suggests its coronavirus vaccine is 95% effective and that the shots would protect older people most at risk of dyâ€¦'
```

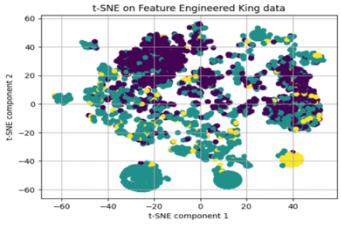
Next, feature engineering was conducted on both the cleaned and uncleaned datasets, along with hold out sets. On the cleaned data, length of the text, log of the length, a character count, count of numeric characters, and a word density are calculated. More features are created out of the original text file, given that it still has the hashtag character, uppercase characters, and stop words. The same metrics as cleaned data are calculated along with count of the stop words, count of uppercase letters, count of punctuation, and a count of hashtags. These features are checked for correlations, and we found that multiple of these feature engineered features were correlated in both datasets. To combat any potential issues, features were transformed or removed, then combined with the next step of this procedure, numerical vectorization. An example of this correlation is shown in the plots below, with correlated features being removed or combined in the plot on the right.





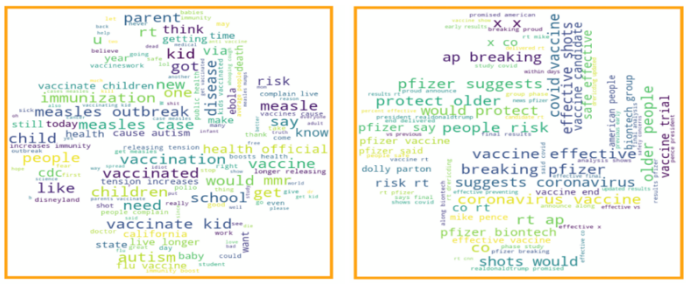
Following this, the Zindi data and Professor King's text data underwent cleaning, feature engineering, and numerical vectorization. The correlation between the feature engineered features was analyzed, and certain features were removed or transformed to avoid potential issues. Principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) were conducted on both datasets. PCA did not reveal any significant groupings in either dataset, while t-SNE partially identified groupings in Professor King's dataset, but not in the Zindi dataset. Below Two t-SNE plots were presented, with Zindi on the left and Professor King's on the right, visually demonstrating the effectiveness of t-SNE in the latter dataset.





From a visual comparison, it can be seen how t-SNE was more effective in Professor King's data compared to Zindi.

Following, The final comparison of the two datasets was done using word clouds, shown below. The word cloud on the left is for Zindi data and the one on the right is for Professor King's data.



Each word cloud examined the frequency of an individual word in the dataset. Interestingly, in Professor King' King's data, many words appear to have the same frequency, showing that they may be used within the same tweets.

Karabiber (2023) explained that term frequency-inverse document frequency (TF-IDF) is an NLP method that measures the importance of a term within the document relative to the entire corpus. This score is calculated by multiplying the term frequency, which is the number of times the word appears out of the total words in the entire document, by the inverse document frequency, which is the logarithm of the number of items divided by the number of items that contain the term. On the other hand, the bag of words method simply counts the frequency of the term's usage in the data (Great Learning Team, 2022). Word to vectors, a more advanced method, transforms text into a form that deep neural networks can understand, providing numerical representations of word features (pathmind, 2023). The vector average was computed by summing up the 100 vector values assigned to each word in each sentence and dividing it by the total words in that sentence. The ngram parameter specifies the number of words that are tokenized. With different combinations of these techniques and parameters, sixteen datasets were created, ten on the Zindi data and six on Professor King's data. From there, each dataset, except number one, was combined with the feature engineered datasets, and models were run to determine the best performance.

For Zindi data:

1. Bag of words, ngram=1,1, non-binary (done on raw text data)
2. Bag of words, ngram=1,1, appear in at least 5 documents, non-binary
3. Bag of words, ngram=1,2, appear in at least 5 documents, non-binary
4. TF-IDF, ngram=1,1, appear in at least 5 documents, non-binary
5. TF-IDF, ngram=1,2, appear in at least 5 documents, non-binary
6. Bag of words, ngram=1,2, appear in at least 5 documents, top 500 features, non-binary

7. Bag of words, ngram=1,2, appear in at least 5 documents, top 100 features, non-binary
8. Word2vec, minimum count of 5
9. TF-IDF multiplied by the word2vec
10. Bag of words, ngram=1,1, binary, appear in at least 5 documents ,top 100 features

For Professor King's data:

1. Bag of words, ngram=1,1, non-binary (done on raw text data)
2. Bag of words, ngram=1,2, appear in at least 15 documents, non-binary
3. TF-IDF, ngram=1,1 , appear in at least 15 documents, top 500 features, non-binary
4. TF-IDF, ngram=1,2, appear in at least 15 documents, top 100 features, non-binary
5. Bag of words, ngram=1,2 appear in at least 15 documents, top 500 features, non-binary
6. Bag of words, ngram=1,2, appear in at least 15 documents, top 100 features, non-binary

Experiment: Main Idea: - How are these features described above can be helpful with ML techniques in our objective of classification?

Evidence:-

Table 1:

Model	Dataset	Accuracy	F1 Score
Naïve Bayes (Baseline)	Zindi dataset #1	0.679	0.670
kNN (Baseline)	Zindi dataset #1	0.583	0.550
kNN	Zindi dataset #10	0.474	0.500
Decision Tree (Baseline)	Zindi dataset #1	0.678	0.660
Decision Tree	Zindi dataset #10	0.610	0.550

Table 2:

Model	Dataset	Accuracy	F1 Score
kNN (Baseline)	Professor King dataset #1	0.292	0.260
kNN (Baseline)	Professor King dataset #2	0.300	0.300

Table 3:

Model	Dataset	Accuracy	F1 Score
kNN	Professor King dataset #6	0.316	0.300

Table 4:

Model	Dataset	Accuracy	F1 Score
Decision Tree	Professor King dataset #6	0.292	0.260
Decision Tree	Professor King dataset #4	0.300	0.300
Random Forest	Professor King dataset #6	0.360	0.330
Random Forest	Professor King dataset #4	0.336	0.310
GBDT	Professor King dataset #4	0.360	0.330
SVM	Professor King dataset #6	0.464	0.450
SVM	Professor King dataset #4	0.456	0.440

Analysis:

Modeling - Training on Zindi Data Testing on Professor King's Twitter Data:

In the first stage of modeling, the tenth dataset created from the Zindi data was used in order to train a tuned kNN and decision tree model. These tuned models were compared against a baseline model created from the first Zindi dataset. Performance of these models, shown in the table 1 above, are used as benchmarks that additional models created using cleaner data. The chosen evaluation metrics for the models are accuracy and weighted f1 score, chosen due to its ability to handle class imbalance (sklearn, 2023). For the tuned kNN, hyperparameter testing was done to examine the log loss for each value of K. A plot comparing the log loss on the training and cross validation datasets was created in order to select an ideal K. Using visual analysis, the optimal point for K was determined to be 21 as the distance between the train and testing log loss appeared to prevent over or underfitting the best. When applying this model to the 500 manually labeled test observations, the observed weighted f1 score is about 0.50. For the first tuned decision tree model, a grid search cross validation technique was utilized. This grid search concluded that the best criterion to determine the quality of the split was the Gini impurity metric with the ideal maximum depth being seven. Additionally, the minimum sample leaf and split parameters were found to be 0.1. The resulting tree was tested on the same 500 manually labeled rows in the test dataset provided by Professor King. The weighted f1 score from this test analysis was 0.55. While results here are promising, neither tuned model outperformed its baseline counterpart.

Following this section, the procedure was changed models are trained and tested on Professor King's data.

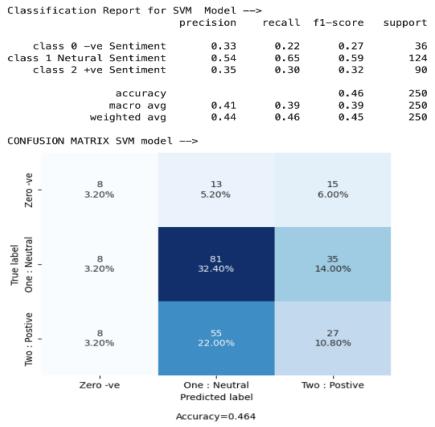
Modeling - Training and Testing on Professor King's Data:

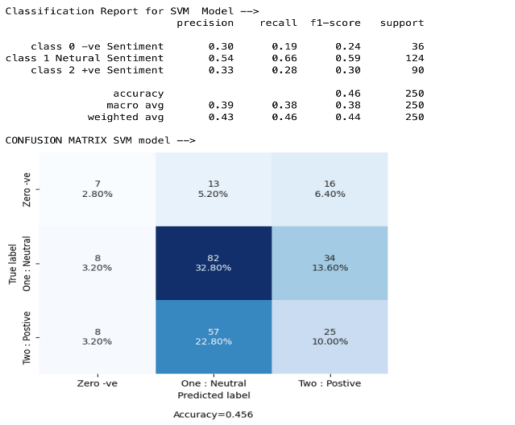
For this stage, training and testing was done on data provided by Professor King. The training set was the 10000 labeled rows and tested on 500 manually labeled observations. This process began by creating baseline models using bag of words with unigram and bag of words with bigram, both of which take all features into account. These initial models are kNN models, with results shown in table 2. Each of these kNN models have a K value of eleven, using the same visual analysis from the prior modeling process. From this initial modeling, using all the features led to poor performance as these models had an accuracy of roughly 29% and 30% respectively and a weighted f1 score of 0.26 and 0.30. Weighted f1 score was chosen as an evaluation metric here due to its ability to handle class imbalance seen in the data (sklearn, 2023). Additional models were judged on their ability to surpass this baseline set by the kNN models.

Next, in an attempt to see how viable a kNN model is for this data, another one is created on a more limited dataset. This model examines if including all the features was the cause of the poor baseline performance. The dataset used here was a bigram bag of words dataset limited to the top 100 features. As seen in the table 3 above, model results remain effectively unchanged, with accuracy improving marginally and f1 score not changing.

Following the kNN analysis, this next stage of the modeling applies different machine learning techniques to two datasets. Results can be observed in table 4 above. These selected datasets included a bag of words with bigram and top 100 features (Professor King dataset #6) and a TF-IDF dataset with bigram and the top 100 features (Professor King dataset #4). These models include decision trees, random forest, gradient-boosted decision trees (GBDT), and support vector machines (SVM). Optimal decision trees were found using a grid search cross validation technique. A GBDT model was only created on the TF-IDF dataset due to the time it took to run (over 12 hours), given it led to minimal improvements, it was not run for the bag of words dataset.

Examining these results as shown above in table 4, most of the machine learning models tested here provided little to no improvement upon the base kNN models that were created. However, out of these models, SVM provided the greatest improvement in the results, with accuracy increasing from 30% to 46%. Confusion matrices for the SVM models are shown below, results for the bag of words dataset are on the left TF-IDF results are on the right. A note about this modeling process is that models were tested with correlated feature engineered features as well as with these features being combined. Accounting for these correlated features through combination did not significantly affect the modeling results.





Advanced Modeling – BERT Vanilla:

The final model created in the analysis was a more advanced BERT Vanilla transformer model. This model produced an accuracy of roughly 50%, outperforming all of the machine learning

models created in this project. These BERT results show promise for future research of this topic using deep learning techniques.

```
Epoch 1/10
313/313 [==============================] - 39s 101ms/step - loss: 0.9091 - accuracy: 0.5856
Epoch 2/10
313/313 [==============================] - 11s 35ms/step - loss: 0.9045 - accuracy: 0.5856
Epoch 3/10
313/313 [==============================] - 10s 31ms/step - loss: 0.9046 - accuracy: 0.5856
Epoch 4/10
313/313 [==============================] - 8s 26ms/step - loss: 0.9039 - accuracy: 0.5856
Epoch 5/10
313/313 [==============================] - 7s 24ms/step - loss: 0.9043 - accuracy: 0.5856
Epoch 6/10
313/313 [==============================] - 8s 26ms/step - loss: 0.9034 - accuracy: 0.5856
Epoch 7/10
313/313 [==============================] - 9s 27ms/step - loss: 0.9033 - accuracy: 0.5856
Epoch 8/10
313/313 [==============================] - 7s 24ms/step - loss: 0.9033 - accuracy: 0.5856
Epoch 9/10
313/313 [==============================] - 9s 27ms/step - loss: 0.9036 - accuracy: 0.5856
Epoch 10/10
313/313 [==============================] - 7s 23ms/step - loss: 0.9030 - accuracy: 0.5856
8/8 [==============================] - 1s 21ms/step - loss: 1.2518 - accuracy: 0.4960
Test accuracy: 0.4959999918937683
```

Leadout-Conclusions and Future Research:

The research analysis concluded using of BOW, TFIDF, W2V vectorizations, Accuracy, F1 Score, Along with two modeling approaches, one using Zindi data and the other using Professor King's data. While the Zindi modeling did not surpass the benchmarks, it provided interesting results when tested on different data. The SVM models in the Professor King's data modeling showed some improvement, and a preliminary analysis using a deep learning approach showed potential for future research. The challenge faced was the nature of Twitter data, which includes internet slang, sarcasm, and emojis that were not accounted for in the analysis. Future research could explore using custom dictionaries and techniques like XGBoost and light GBM to analyze URLs.

References:

1. Project Data challenge Link: https://zindi.africa/competitions/to-vaccinate-or-not-to-vaccinate
2. Evaluation-metrics: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
3. Bag of Words definition: https://www.mygreatlearning.com/blog/bag-of-words/#sh1
4. Word2Vec definition: https://wiki.pathmind.com/word2vec
5. TF-IDF definition: https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/
6. F1-score metric: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html