IST 707 FINAL PROJECT UPDATE

ANALYZE PUBLIC SENTIMENT TOWARDS VACCINATIONS ON TWITTER

Presented by:

Ben Wachtel

Choijamts Bataa

Kishan Rathor

Saikumarreddy Pochireddygari

Under the guidance of:

Prof Kelvin King

BACKGROUND

- Vaccines are essential for maintaining public health by preventing the spread of communicable diseases.
- Public health organizations like the CDC recommend routine immunizations for everyone.
- The "anti-vaxxer" movement has caused hesitancy towards the COVID-19 vaccine and vaccines in general.
- Studying people's sentiment towards vaccinations is important for future vaccination efforts.
- Understanding public sentiment towards vaccines can help create better communication strategies to encourage more people to get vaccinated.

RESEARCH PROBLEM AND BENEFITS

- Analyzing sentiment towards vaccines on social media can help public health organizations identify concerns and misconceptions.
- Natural language processing (NLP) techniques can be used to classify tweets as positive, negative, or neutral towards vaccines.
- Improved communication strategies can be developed based on concerns identified through sentiment analysis.
- This sentiment analysis can lead to better public health outcomes by assisting public health organizations in evaluating and modifying their communication strategies.

Zindi Data

- Data were sourced from ZINDI, a non-profit platform for data science competitions.
- This dataset has 10,001 observations with four features: tweet ID, safe_tweet text, tweet label, and agreement score.
- Agreement score was not used in this analysis

Zindi Data

- The Zindi data was cleaned before preprocessing and model training.
- Label values were changed from numeric -1, 0, 1 to a categorical 0, 1, 2 to avoid errors with certain classification models.
- Two records contained null data and removed from the training set, reducing it from 10,001 to 9,999.
- 307 duplicate observations were found, the first record was kept and the rest were removed, leaving 9,692 observations.
- The agreement score and tweet ID were not considered in the modeling, leaving the dataset with two features.
- No nulls or duplicates were found in Professor King's data

```
In [102]: # check for null values
          raw_train_data.isnull().sum()

Out[102]: tweet_id     0
          safe_text    0
          label        1
          agreement    2
          dtype: int64
```

Professor King's Data

- Professor King provided a data set with 20,000 tweets related to the COVID vaccine, containing seven features including tweet ID, link to tweet, screen name of user, text of tweet, and type of tweet (reply, original, retweet).
    - 10000 rows labeled by Professor King
    - 1500 manually labelled rows
- We were only interested in the last four columns
    - dropped id through user_screen_name from the analysis

	id	tweet_url	created_at	parsed_created_at	user_screen_name	text	tweet_type	positive emotion	negative_emotion
0	1329066068720041989	https://twitter.com/minabelles/status/13290660...	Wed Nov 18 14:16:58 +0000 2020	44153.345116	minabelles	New Pfizer Results: Coronavirus Vaccine Is Saf...	original	0.0	6.67
1	1329066070854967296	https://twitter.com/meet_rayyan/status/1329066...	Wed Nov 18 14:16:58 +0000 2020	44153.345116	meet_rayyan	RT @DrEricDing: BREAKINGâ€"Updated results fro...	retweet	0.0	0.00
2	1329066076613828608	https://twitter.com/BiancaJagger/status/132906...	Wed Nov 18 14:17:00 +0000 2020	44153.345139	BiancaJagger	RT @BBCBreaking: Coronavirus vaccine by Pfizer...	retweet	0.0	4.35
3	1329066074424418311	https://twitter.com/Anony05690448/status/13290...	Wed Nov 18 14:16:59 +0000 2020	44153.345127	Anony05690448	This is an objectively written and factual thr...	original	0.0	2.27
4	1329066084180234241	https://twitter.com/Alysenve/status/1329066084...	Wed Nov 18 14:17:01 +0000 2020	44153.345150	Alysenve	RT @AP: BREAKING: Pfizer suggests its coronavi...	retweet	0.0	0.00

Professor King's Data

- We observed that positive emotion and negative emotion were continuous variables
- Used the logic shown in the for loop to the right
  - where positive emotion > negative emotion, the tweet is classified as positive
  - where negative emotion < positive emotion, the tweet is classified as negative
  - when the two are equal, the tweet is classified as neutral

tweet_type	positive emotion	negative_emotion
original	0.0	6.67
retweet	0.0	0.00
retweet	0.0	4.35
original	0.0	2.27
retweet	0.0	0.00

```
for (p,v) in zipped_list_pos_neg:
    if p > v:
        res_pos_neg.append(1)
    elif p < v:
        res_pos_neg.append(-1)
    else:
        res_pos_neg.append(0)
```

DATA PREPROCESSING

- Multiple preprocessing steps were carried out before running models:
  - Removal of unneeded text
  - Feature engineering
  - Numerical vectorization

DATA PREPROCESSING - REMOVING TEXT

- Observed hashtags, user/URL tags, and emojis in both datasets
  - Zindi shown on top
  - Professor King's on the bottom

- Created a series of functions to remove this unneeded text as well as convert contractions, remove stopwords, non-English characters and convert the data to lowercase.

```
#MAKING A GLOBAL COPY OF DATASET
raw_train_data_global_copy = raw_train_data.copy(deep=True)
```

```
# understanding few tweets
raw_train_data.safe_text.values[0]
```

```
'Me &amp;amp; The Big Homie meanboy3000 #MEANBOY #MB #MBS #MMR #STEGMANLIFE @ Stegman St. <url>'
```

```
raw_train_data.safe_text.values[10]
```

```
"<user> @ this point I have 2 text, butw/Bon Jovi cover playin @ Alibi's hope U can come out 2 MMR BBQ<user> will b there!"
```

```
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.'
```

```
raw_train_data.safe_text.values[1000]
```

```
'"<user> <user> Polio vaccine. It kept me from infecting and/or killing millions. #pathogenposse". //I still vote yoga pants.'
```

```
raw_train_data_king.safe_text.values[4444]
```

```
'RT @HillaryPix: #Vaccine #CovidRelief _x000D_\n@HillaryClinton at the Bloomberg New Economy Forum. _x000D_\n"Biden has a world class board of advisers to tâ€¦'
```

```
raw_train_data_king.safe_text.values[5555]
```

```
'"Covid Vaccine comes out"_x000D_\n._x000D_\n._x000D_\nMKBHD: Hey guys MKBHD here,_x000D_\nthis is a review video of Vaccine._x000D_\nSo I have been using this Vaccine for a couple of weeks now ðŸ˜,ðŸ˜,._x000D_\n#MKBHD @MKBHD'
```

```
raw_train_data_king.safe_text.values[6666]
```

```
'RT @ChrisVanHollen: Corporate executives shouldnâ€™t be able to game the system to profit off insider info like a COVID vaccine. Asked SEC Châ€¦'
```

```
raw_train_data_king.safe_text.values[7777]
```

```
'RT @Mike_Pence: President @realDonaldTrump promised the American people a safe and effective vaccine by the end of 2020 and he DELIVERED! Bâ€¦'
```

```
raw_train_data_king.safe_text.values[8888]
```

```
"RT @CNN: Dolly Parton's $1 million donation to coronavirus research was partly used to fund Moderna's promising Covid-19 vaccine â€” somethinâ€¦"
```

```
raw_train_data_king.safe_text.values[9999]
```

```
'Honestly the endgame to covid didnâ€™t have to be a vaccine. But thanks to the mishandling of the situation by our government + the careless/selfish nature of Americans, it really is the only foreseeable hope for this country.'
```

```
raw_train_data_king.safe_text.values[2988]
```

```
'The FDAâ€™s advisory panel is currently scheduled to meet December 8-10  to discuss #COVID19 vaccines.  If the @US_FDA authorizes the two-dose vaccine, @pfizer  said they could have up to 25 million doses available for the US by the end of the year. _x000D_\nhttps://t.co/lpvuhPUTOc'
```

DATA PREPROCESSING - FEATURE ENGINEERING

- Computed metrics on both the cleaned and uncleaned text data
- For the cleaned data, the following metrics were calculated:
  - count of total words
  - count of uppercase characters
  - word density
  - numeric character count
- For the uncleaned data, the following metrics were calculated:
  - same metrics as cleaned data
  - count of total stop words
  - count of uppercase characters
  - average word
  - punctuation count
  - hashtag count
- Combined metrics into one dataframe, then standardized
- This was done on the Zindi data, Professor King's data as well as any holdout sets that were created

DATA PREPROCESSING - NUMERICAL VECTORIZATION

- Sixteen separate training datasets created with different parameters:
  - binary/non-binary
  - different ngrams (amount of words to be tokenized at one time)
  - upsampling
  - feature limitation
  - different vectorization techniques
- Used three different vectorization techniques:
  - TF-IDF: measures the importance of a term within a document relative to the entire corpus.
  - Bag of words: counts frequency of a term's usage in the data.
  - words to vectors (words2vec): provides numerical representations of word features for deep neural networks
- Upsampling conducted due to class imbalance between tweets with negative and positive/neutral sentiment.
- First dataset created from raw text data, remaining seventeen from cleaned text data.

DATA PREPROCESSING - NUMERICAL VECTORIZATION

- To the right are the sixteen datasets that were derived from this procedure
  - 10 for Zindi Data
  - 6 for Professor King's data
- Training and hyperparameter tuning will be done on these datasets before applying to selected test data

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, maximum 500 features, non-binary
7. Bag of words, ngram=1,2, appear in at least 5 documents, maximum 100 features, non-binary
8. Word2vec, minimum count of 5
9. TF-IDF multiplied by the word2vec
10. Bag of words binary with ngram range 1,1 and top 100 Features

For Professor King's data:

11. Bag of words, ngram=1,1, non-binary (done on raw text data)
12. Bag of words, ngram=1,2, appear in at least 15 documents, non-binary
13. TF-IDF, ngram=1,1 , appear in at least 15 documents, non-binary
14. TF-IDF, ngram=1,2, appear in at least 15 documents, non-binary
15. Bag of words, ngram=1,2 appear in at least 15 documents, maximum 500 features, non-binary
16. Bag of words, ngram=1,2, appear in at least 15 documents, maximum 100 features, non-binary

Checkpoint Two Results Summary

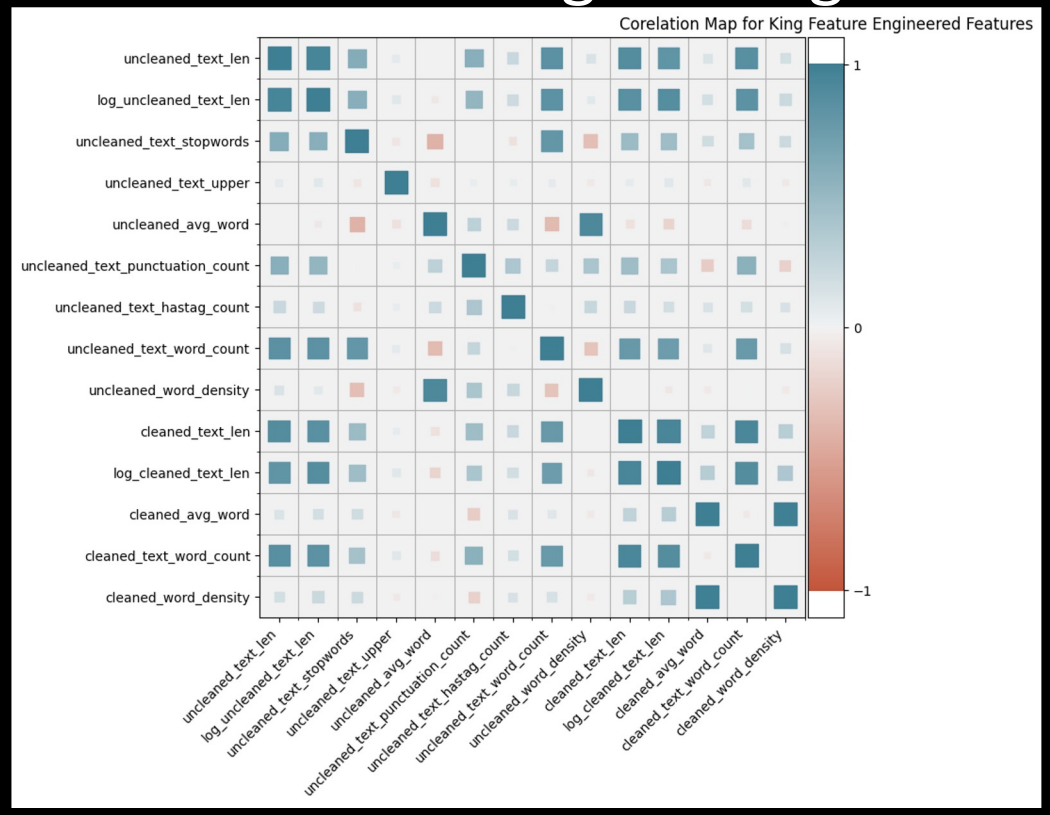
- Baseline models on unedited data
  - kNN weighted f1 score: 0.55
  - Decision tree weighted f1 score: 0.61
- Train on Zindi data, test on 500 manually labeled rows from Professor King's data
- Hypertuned kNN model had K=21
- kNN weighted f1 score: 0.50
- kNN log loss: 0.969
- Decision tree chosen using grid search
  - Tree uses Gini impurity, maximum depth of 7, minimum samples split of 0.1, and minimum samples leaf of 0.1
- Decision tree weighted f1 score: 0.55
- Decision tree log loss: 1.014

Checkpoint Three Procedural Changes

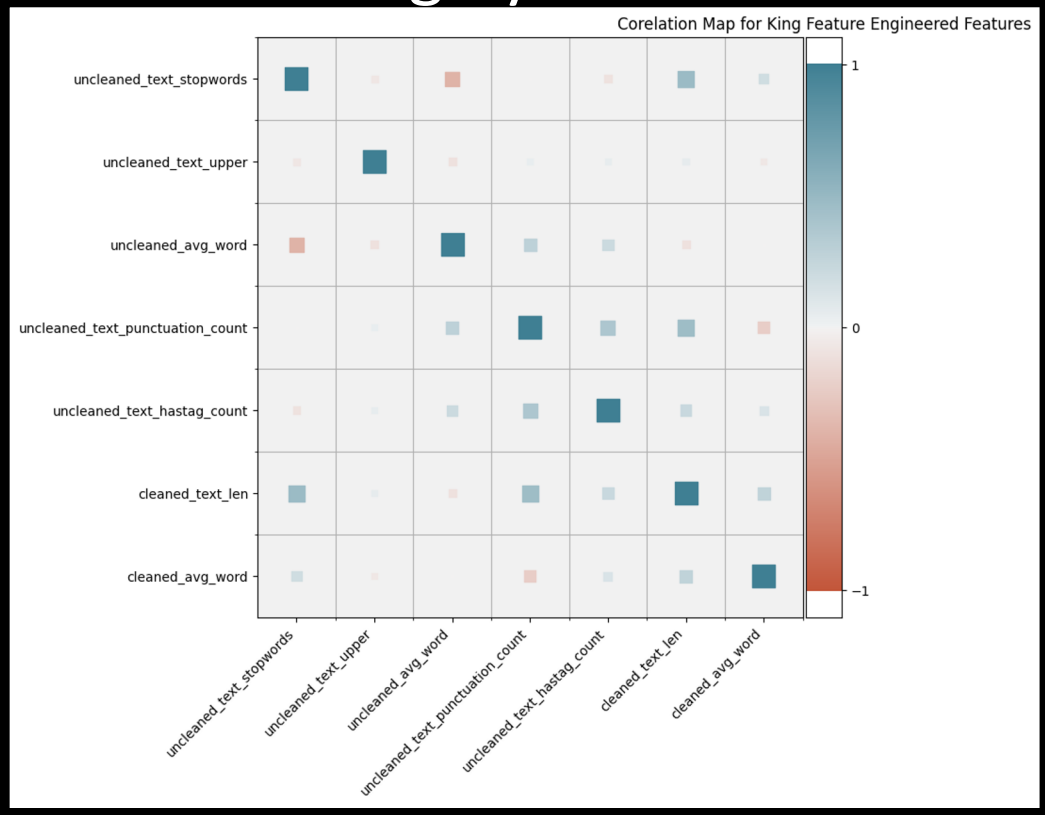
- Through checkpoint two we tained on Zindi data and evaluated results on manually labeled data from Professor King
- In checkpoint three, we train on Professor King's labeled Twitter data and test on different manually labeled records from this dataset.

Correlation Plot for Feature Engineered Variables

- Checked the correlations between the variables created through feature engineering and removed those that were highly correlated.

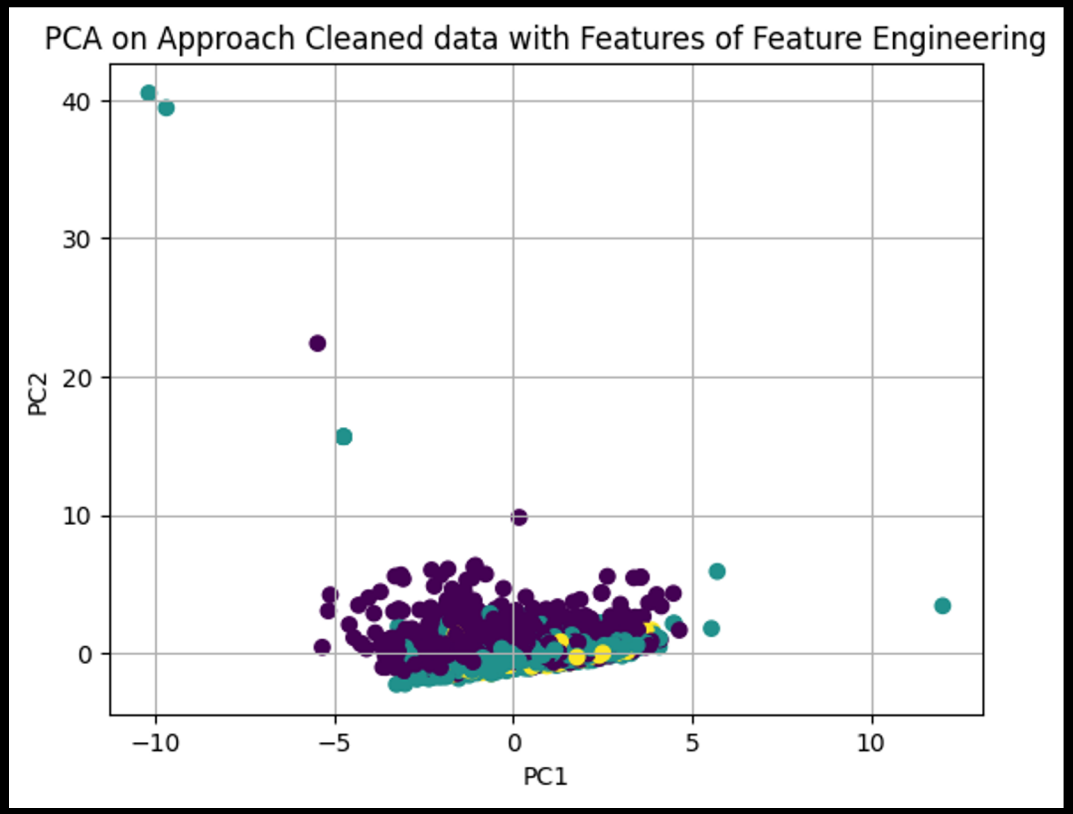


Post-removal

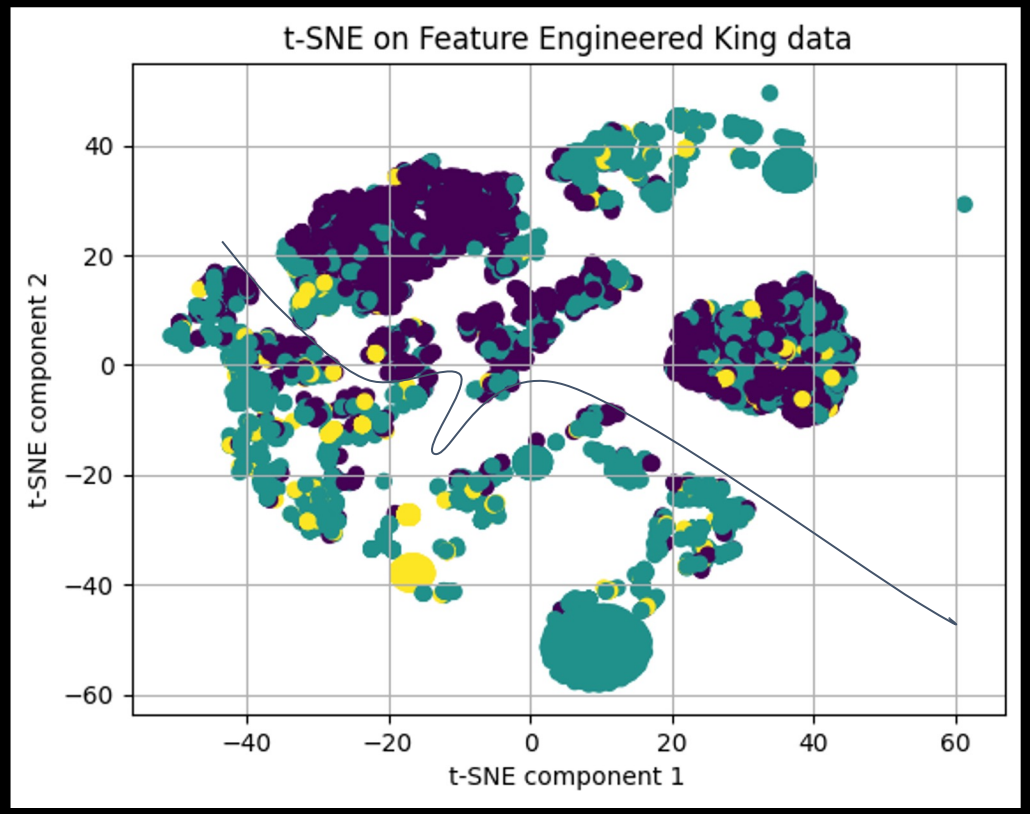


Examining the Usefulness of these Features

- Applied and plotted a simple PCA and t-SNE
- Observed that the t-SNE plot on the right was better at partially identifying groups in the data
  - Teal and violet points

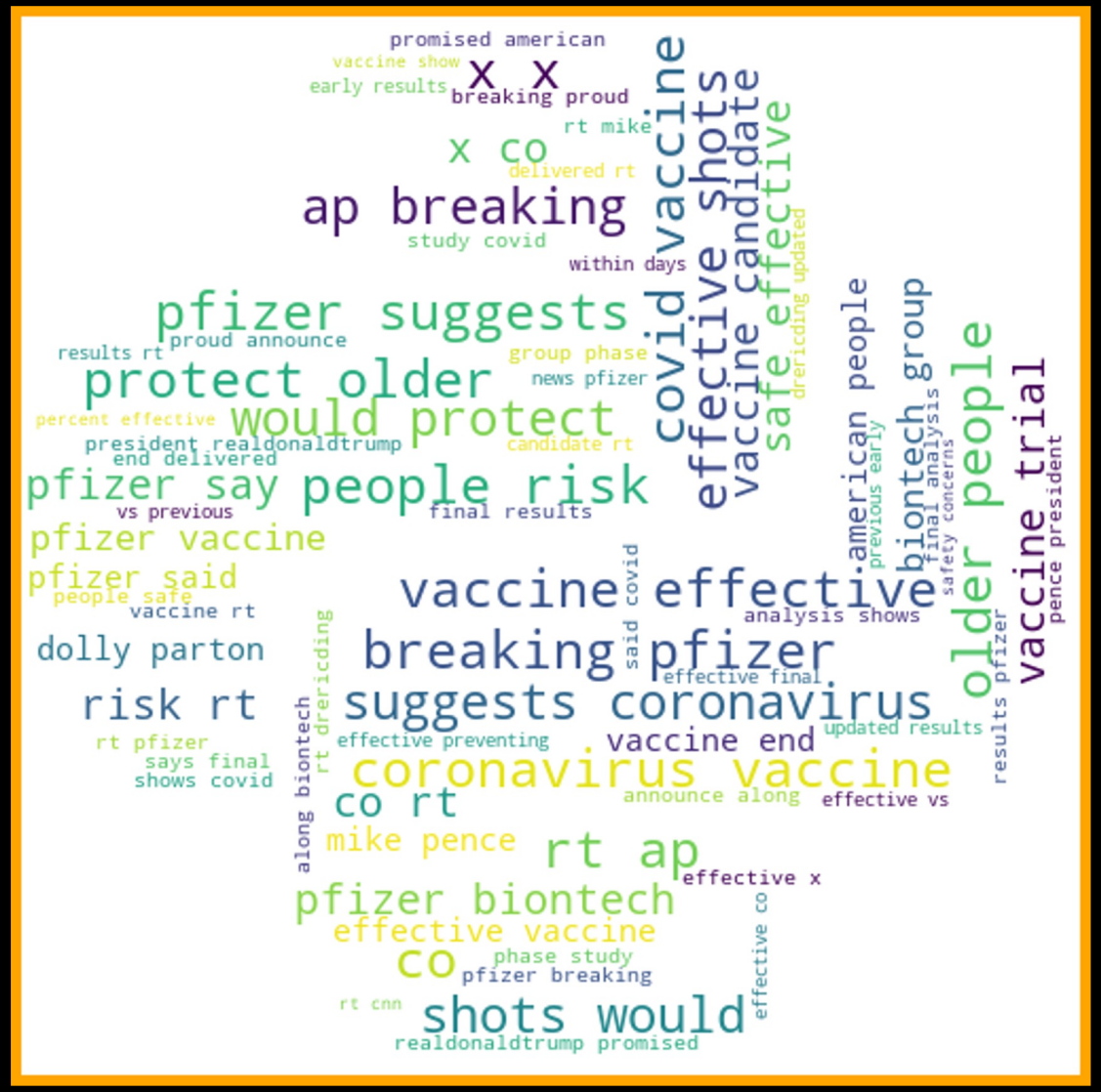


Somewhat t-SNE on the right hand side



Word Cloud

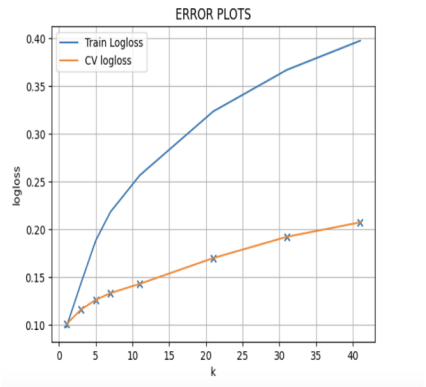
- Created a word cloud to observe the most frequent words in Professor King's data
- See many words with the same frequency
  - Likely used together with in the tweet
  - For example, "coronavirus" and "vaccine" are the same size, probably because they are used together in the tweets



Modeling - Baseline kNNs

- Started modeling with using bag of words with Unigram and taking all features at a time
- Also used bag of words with bigram (1,2) and taking all features at a time
- Used a kNN model, where the optimal amount of neighbors was found to be 11
- From this initial experiment we observe:
  - taking all the features leads to poor model performance
  - Unigram
    - 23% accuracy and 0.26 weighted f1 score
  - Bigram
    - 30% accuracy and weighted f1 score of 0.30

Tuning Result for BOW 1,1 with all features:

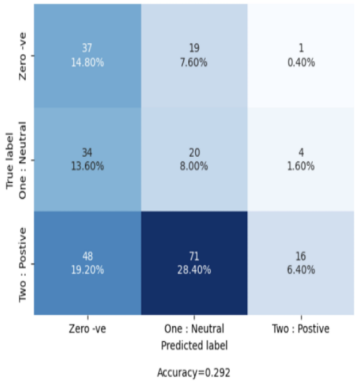


Confusion Matrix with Classification Report:

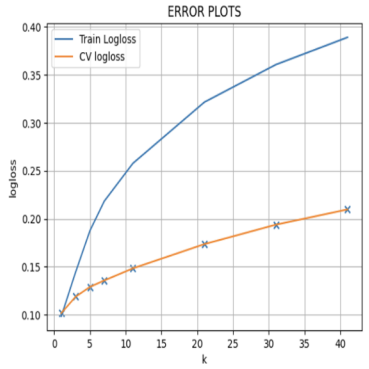
Classification Report for KNN Base Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.31	0.65	0.42	57
class 1 Netural Sentiment	0.18	0.34	0.24	58
class 2 +ve Sentiment	0.76	0.12	0.21	135
accuracy			0.29	250
macro avg	0.42	0.37	0.29	250
weighted avg	0.52	0.29	0.26	250

CONFUSION MATRIX KNN base model -->



Tuning Result for BOW 1,2 with all features:

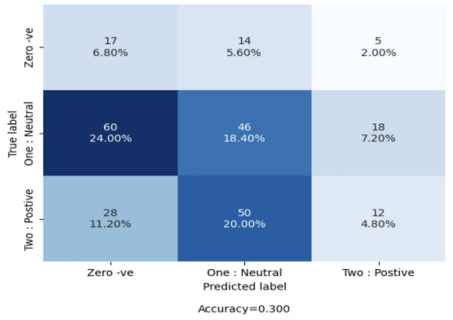


Confusion Matrix with Classification Report:

Classification Report for KNN Base Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.16	0.47	0.24	36
class 1 Netural Sentiment	0.42	0.37	0.39	124
class 2 +ve Sentiment	0.34	0.13	0.19	90
accuracy			0.30	250
macro avg	0.31	0.33	0.28	250
weighted avg	0.35	0.30	0.30	250

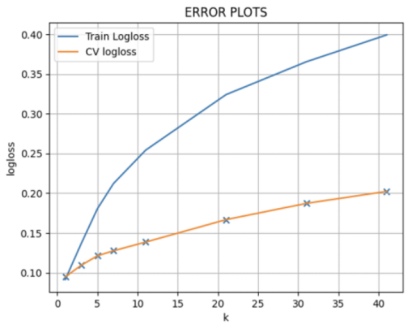
CONFUSION MATRIX KNN base model -->



Modeling - kNN with Different Dataset

- Here, bag of words with bigram (1,2) and taking top 100 features are used
- Optimal Neighbors was 11
- Taking the top 100 features did not provide any significant boost in the performance compared to the baseline models
- Next, we incorporate different modeling techniques

Tuning Result for BOW 1,1 with 100 features:

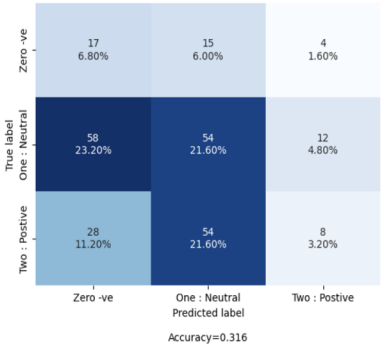


Confusion Matrix with Classification Report:

Classification Report for KNN Base Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.17	0.47	0.24	36
class 1 Netural Sentiment	0.44	0.44	0.44	124
class 2 +ve Sentiment	0.33	0.09	0.14	90
accuracy			0.32	250
macro avg	0.31	0.33	0.27	250
weighted avg	0.36	0.32	0.30	250

CONFUSION MATRIX KNN base model -->



Accuracy=0.316

Modeling - Datasets and Other Approaches

- Used different machine learning techniques on two datasets:
  - bag of words with bigram (1,2), taking the top 100 features
  - TF-IDF with bigram (1,2) taking top 100 features
- Decision Tree, Random Forest, Gradient-boosted decision trees (GBDT), and Support Vector Machines (SVM) models were used in this stage

Modeling - Decision Tree

- Decision tree results for each dataset are displayed to the right
  - bag of words
    - accuracy of 30% and weighted f1 score of 0.29
  - TF-IDF
    - accuracy of 35% and weighted f1 score of 0.31
- Minimal improvement compared to baseline kNN models

Tuning Res for BOW, TFIDF 1,2 with 100 features: Confusion Matrix with Classification Report:

Classification Report for DT Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.15	0.36	0.21	36
class 1 Netural Sentiment	0.44	0.44	0.44	124
class 2 +ve Sentiment	0.22	0.09	0.13	90
accuracy			0.30	250
macro avg	0.27	0.30	0.26	250
weighted avg	0.31	0.30	0.29	250

CONFUSION MATRIX DT model -->

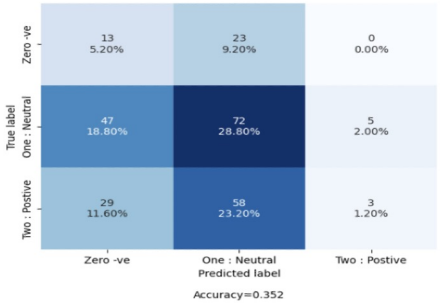

```
{'criterion': 'entropy',
 'max_depth': 7,
 'min_samples_leaf': 0.1,
 'min_samples_split': 0.1}
```

Bow Result Above

Classification Report for DT Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.15	0.36	0.21	36
class 1 Netural Sentiment	0.47	0.58	0.52	124
class 2 +ve Sentiment	0.38	0.03	0.06	90
accuracy			0.35	250
macro avg	0.33	0.33	0.26	250
weighted avg	0.39	0.35	0.31	250

CONFUSION MATRIX DT model -->




```
{'criterion': 'gini',
 'max_depth': 3,
 'min_samples_leaf': 0.1,
 'min_samples_split': 0.1}
```

Tf-Idf Result Above

Modeling - Random Forest

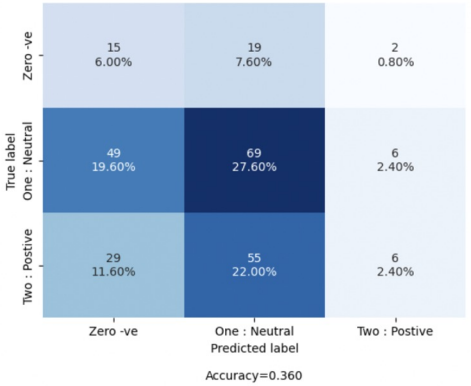
- Random forest results for each dataset are displayed to the right
  - bag of words
    - accuracy of 36% and weighted f1 score of 0.33
  - TF-IDF
    - accuracy of 34% and weighted f1 score of 0.31
- Slight improvements for bag of words, similar performance for TF-IDF

Tuning Res and Classification Report for BOW, TFIDF 1,2 with 100 features: Confusion Matrix:

```
Best parameters:
{'clf__max_depth': 11, 'clf__min_samples_split': 2, 'clf__n_estimators': 100}
Classification report:
                        precision    recall  f1-score   support

    class 0 -ve Sentiment       0.16      0.42      0.23        36
class 1 Netural Sentiment       0.48      0.56      0.52       124
    class 2 +ve Sentiment       0.43      0.07      0.12        90

                 accuracy                           0.36       250
                macro avg       0.36      0.35      0.29       250
             weighted avg       0.42      0.36      0.33       250
```

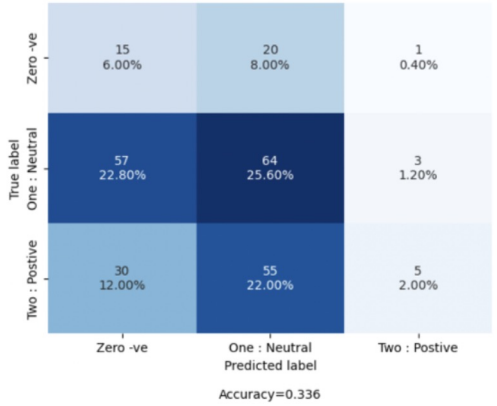


Above is Bow Result

```
Best parameters:
{'clf__max_depth': 11, 'clf__min_samples_split': 2, 'clf__n_estimators': 100}
Classification report:
                        precision    recall  f1-score   support

    class 0 -ve Sentiment       0.15      0.42      0.22        36
class 1 Netural Sentiment       0.46      0.52      0.49       124
    class 2 +ve Sentiment       0.56      0.06      0.10        90

                 accuracy                           0.34       250
                macro avg       0.39      0.33      0.27       250
             weighted avg       0.45      0.34      0.31       250
```



Above is TFIDF Result

Modeling - GBDT on TF-IDF dataset

- Random forest results for the TF-IDF dataset are displayed to the right
  - accuracy of 36% and weighted f1 score of 0.341
- Slight improvements compared to previous models for TF-IDF and baseline kNN
- This model was not done on the bag of words dataset due to how long it takes to run

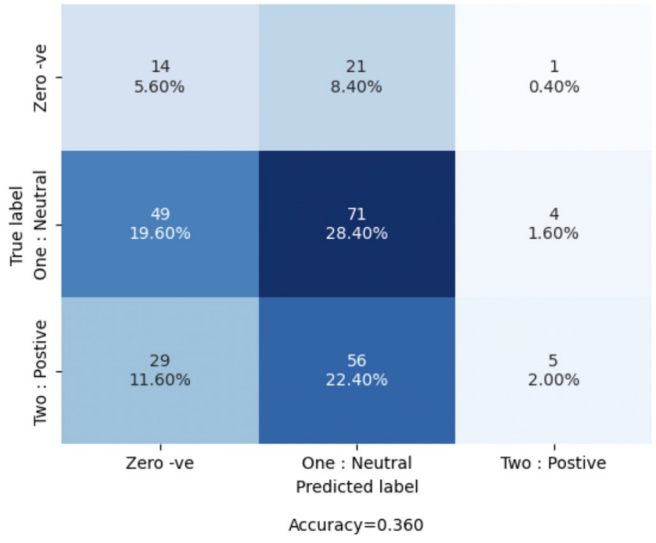
Tuning Res and Classification Report for TFIDF 1,2 with 100 features:

```
Best parameters:
{'clf__learning_rate': 0.1, 'clf__max_depth': 15, 'clf__min_samples_split': 5, 'clf__n_estimators': 200}
Classification report:
                          precision    recall  f1-score   support

    class 0 -ve Sentiment       0.15      0.39      0.22        36
class 1 Netural Sentiment       0.48      0.57      0.52       124
    class 2 +ve Sentiment       0.50      0.06      0.10        90

                 accuracy                           0.36       250
                macro avg       0.38      0.34      0.28       250
             weighted avg       0.44      0.36      0.33       250
```

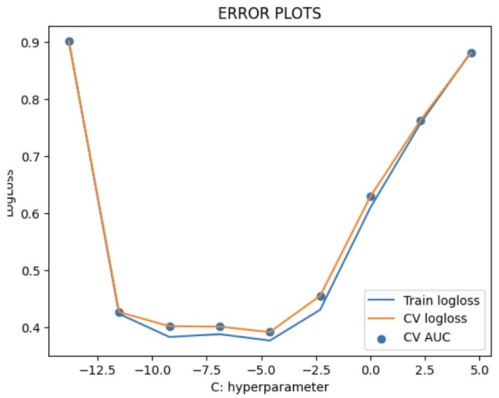
Confusion Matrix:



Modeling - SVM

- optimal C parameter was 0.01
- Results displayed on the right:
  - bag of words
    - accuracy of 46% and weighted f1 score of 0.45
  - TF-IDF
    - accuracy of 46% and weighted f1 score of 0.44
- SVM models provided a significant boost in model performance compared to baseline models and our other tests

Tuning Res for BOW, TFIDF 1,2 - 100 features:

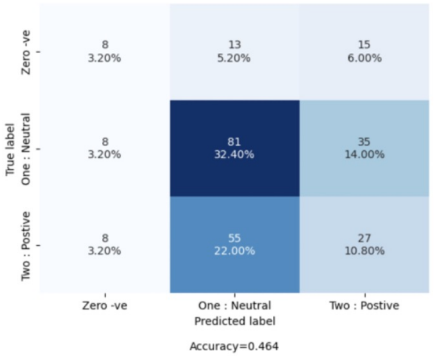


Confusion Matrix with Classification Report:

Classification Report for SVM Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.33	0.22	0.27	36
class 1 Netural Sentiment	0.54	0.65	0.59	124
class 2 +ve Sentiment	0.35	0.30	0.32	90
accuracy			0.46	250
macro avg	0.41	0.39	0.39	250
weighted avg	0.44	0.46	0.45	250

CONFUSION MATRIX SVM model -->



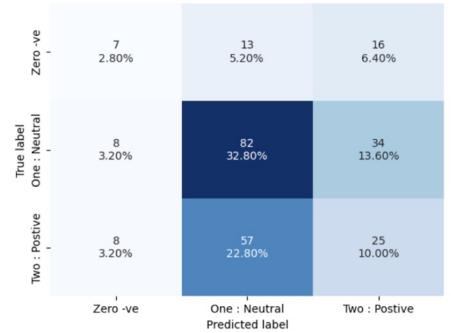
Accuracy=0.464

Above is for BOW result

Classification Report for SVM Model -->

	precision	recall	f1-score	support
class 0 -ve Sentiment	0.30	0.19	0.24	36
class 1 Netural Sentiment	0.54	0.66	0.59	124
class 2 +ve Sentiment	0.33	0.28	0.30	90
accuracy			0.46	250
macro avg	0.39	0.38	0.38	250
weighted avg	0.43	0.46	0.44	250

CONFUSION MATRIX SVM model -->



Accuracy=0.456

Above is for TFIDF Result, Skipped Tuning plot because we got same plot again

Modeling - Note

- Combined correlated feature engineered variables in case multicollinearity was a problem
  - Example below shows how this was done
- Modeling was done on these transformed features and results did not significantly change
- We also considered using XGBoost, GBDT, Light GBM techniques but they took too long to run
  - One GBDT took about 12 hours

```
raw_train_data_king['m_1'] = raw_train_data_king['uncleaned_text_len'] + raw_train_data_king['log_uncleaned_text_len'] + raw_train_data_king['uncleaned_text_word_count'] +\
raw_train_data_king['cleaned_text_len']+raw_train_data_king['log_cleaned_text_len']   +\
raw_train_data_king['cleaned_text_word_count'] + raw_train_data_king['uncleaned_text_word_count']
#keep uncleaned_tet_upper

raw_train_data_king['m_2']  = raw_train_data_king['uncleaned_avg_word'] + raw_train_data_king['uncleaned_word_density']
#keep uncleaned text punc count, uncleaned hastag count

raw_train_data_king['m_3'] = raw_train_data_king['cleaned_avg_word'] + raw_train_data_king['cleaned_word_density']
```

Conclusions

- With this new process, we found promising results from our models, primarily from the SVM model compared to our baseline kNN models
- Saw accuracy increase from 23% to 46%
- The machine learning models showed promise but did not perform well but there is more potential in deep learning approaches
  - Vanilla Bert model had a test accuracy of 49%

Future Research Considerations

- Custom dictionaries to handle Twitter corpus and words that cannot be interpreted. For example, someone on Twitter may use the word “gud” when they mean the word “good”, a custom dictionary would convert this internet lingo into correctly spelled English words that could then be classified based on its sentiment
- Applying XGBoost, GBDT, Light GBM techniques
- Taking content from the URLs into account
- Using Advanced Deep Learning Techniques with Tuning
- Using Named Entity Recognition (NER) to identify some company mentions, place mentions, Etc.

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. Brett Lantz (2019) Machine Learning with R (third edition).
4. Bag of Words definition: https://www.mygreatlearning.com/blog/bag-of-words/#sh1
5. Word2Vec definition: https://wiki.pathmind.com/word2vec
6. TF-IDF definition: learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/
7. F1-score metric: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

THANK YOU