Toxic Tweets: Examining characteristics of Donald Trump's tweets

Group 2, Jennifer Bochenek, Winter 2021 DSCI 631-900

[**Dataset**](#_v6ax0uw3qoey) **2**

[**Preprocessing**](#_37rm9y3i8j9w) **2**

[**Exploratory Data Analysis**](#_racdwcf8lv8o) **4**

[**Classification Supervised Methods**](#_v6eghdl7cwpy) **12**

[BERT](#_xiswprupiedq) 12

[SVM](#_z77a6fg3tdnn) 15

[Feature Selection](#_b9oa2ov4g2fm) 15

[SVC](#_6o6bvw8od9tb) 16

[Linear SVC](#_5m27chfct88e) 17

[Naive Bayes](#_xwr6jvpl2u4g) 18

[Multinomial](#_augt69peaskk) 18

[Gaussian](#_kr835nl99v97) 20

[Mixed](#_yzgsx3nxbl6x) 21

[**Clustering Unsupervised Methods**](#_o1n8x1btiu7d) **22**

[LDA](#_gcddbjdzlx65) 23

[Full Dataset](#_tj693ruo1ck7) 23

[Original Posts Only](#_5wmqzx2l0oq3) 25

[t-SNE](#_15d4zza86tnk) 27

[Text Vectors](#_6qk3ktos13cj) 27

[Numeric Variables](#_o3ff08t96xo8) 28

[K-Means](#_s8tcrqpvb714) 29

[**Regression Supervised Methods**](#_yyd94wezbtez) **32**

[Feature Selection - RFECV](#_eqooz5njwp32) 32

[Linear Regressions](#_eqh0se8asyg2) 33

[Linear Regression/Lasso/Ridge](#_20v5zi22n1aj) 33

[Random Forest Regression](#_vkeg4l1e9bgp) 34

[**Model Comparisons**](#_sdlafvrpiyii) **35**

[**Conclusions**](#_54iqxwsx2o06) **35**

# Dataset

The dataset was pulled from an archive as Donald Trump's tweets were suspended from Twitter and are no longer accessible. The dataset was downloaded as a .csv from <https://www.thetrumparchive.com/>. The dataset was pulled on 1/18/2021, but as Donald Trump was suspended from Twitter indefinitely he will no longer be contributing to the archive. The last data comes from 1/8/2021, which was the date the account was suspended. The data comes with 9 columns:

1. id (numeric) the tweet id
2. text (string/object) the full text of the tweet
3. isRetweet (string/object/categorical) a binary variable (t/f) marking if the tweet is a retweet from another account
4. isDeleted (string/object/categorical) a binary variable (t/f) marking if the tweet was later deleted by Trump (not affected by the Twitter ban).
5. device (string/object/categorical) from what platform the tweet was posted (such as computer or mobile)
6. favorites (numeric) the number of other twitter users that have favorited this tweet
7. retweets (numeric) the number of other users that have retweeted this tweet.
8. date (datetime) the date and time that the tweet was posted
9. isFlagged (string/object/categorical) a binary variable (t/f) marking if the tweet has been flagged by Twitter for misleading or blatantly false claims that go against the TOS.

Donald Trump was a prolific tweeter, so there are 56,571 total tweets. Some of these were image only tweets or link only tweets, which do not have text to use so it would further reduce the dataset to 55,330 tweets, removing just tweets that had mentions and urls (no original text), reducing it further to 54,664 tweets.

# Preprocessing

Preprocessing includes creating additional variables/metadata that can be used for model building. These include:

1. Turning the above binary variables from string t/f to 1/0 as bRetweet, bDeleted, and bFlagged
2. Counts of tweet characteristics such as:
   1. character count char\_count (numeric)
   2. word count word\_count (numeric)
   3. average word length mean\_word\_length (numeric)
   4. count of unique words count\_unique\_word (numeric)
   5. count of punctuation marks count\_punctuation (numeric)
   6. count of hashtags count\_hashtags (numeric)
   7. count of urls count\_url (numeric)
   8. count of @mentions count\_mentions (numeric)
3. Separating out hashtags as a possible keyword hashtag (string)
4. Determining the complexity of the tweet
   1. Flesch Reading Ease is a scale roughly from 0-100 (although over 100 and below 0 are possible) where lower scoring text means that it is difficult to understand. This was computed into flesch\_reading\_ease (numeric)
   2. Flesch-Kinkaid Grade level is the assumed literacy level required to read a text, so a score of 9.2 would mean that it could be understood by those who have passed 9th grade. The average text should aim to be 8th grade level if aimed at the general public. flesch\_kinkaid\_grade (numeric)
5. Coding texts for NRC sentiments - NRC is the National Research Council Canada and codes words for 10 sentiments - negative, anger, fear, sadness, disgust, positive, anticipation, trust, surprise, and joy. It has 27,000 words in the lexicon which makes it very flexible. The following variables were added based on the sentiment analysis:
   1. affect\_dict a dictionary of words and their semantic coding per tweet
   2. raw\_emotion a dictionary of semantic codings per tweet and their counts. From this the following is extracted:
      1. negative (numeric)
      2. positive (numeric)
      3. anger (numeric)
      4. fear (numeric)
      5. sadness (numeric)
      6. disgust (numeric)
      7. anticipation (numeric)
      8. trust (numeric)
      9. surprise (numeric)
      10. joy (numeric)
      11. all\_positive a sum of anticipation through joy (numeric)
      12. all\_negative a sum of anger through disgust (numeric)
      13. valence a comparison of the counts of all\_positive and all\_negative, with the possible levels of positive, negative, equal, and none, where positive is when the tweet has a net positive score and negative when the tweet have the net negative score, equal is given when the tweet has equal positive and negative scores, except when negative and positive are both zero, if both are zero it is given the level of none. (categorical/string)
   3. top\_emotions gives only the highest sentiment score (e.g., fear, positive, negative) with the proportion of sentiment as the value (string/list)
   4. affect\_frequencies nearly the same as raw emotion except proportion based instead of raw counts (dict)

As well as some data processing steps such as

1. Turning emoticons into their semantic meaning (string)
2. Fixing encoding errors (e.g., & instead of &amp;) (string)
3. Changing the tweet text to all lowercase (string)
4. Tokenizing the tweet text and removing punctuation (string/list)
5. Stopword removal (string/list)
6. Named Entity Recognition (string/list)
7. Part of Speech Tagging (string/list)
8. Lemmatization (string/list)

The goal was to extract as much meaning as possible from the tweets to use in potential models. Several variables could be used as target variables in a classification task such as valence, isDeleted, (or really bDeleted). While continuous variables such as flesch\_reading\_score or flesch\_kinkaid\_grade could be used for a regression task. Conceptually, another regression task could use favorites or retweets, both numeric counts of the tweet's likes and movement on twitter respectively, as a target variable (dependent variable) to determine its popularity. There are a lot of different ways to hash out this dataset, including by clustering and comparing the clusters, or by using deep learning.

Note that the dataset has yet to be split into train or test sets.

# Exploratory Data Analysis

Initial exploratory data analysis (EDA) suggests that the data is skewed, and in many cases binomial. This is partly because the data goes back to 2013 and between 2013 and 2021 Twitter increased the character length of tweets, so any variable using count of words or characters will be impacted by this.

The means, standard deviations, min, max, and IQR of the full (unshortened) dataset is as follows:

| **variable** | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 56571 | 7.99E+17 | 3.83E+17 | 1.70E+09 | 4.61E+17 | 7.47E+17 | 1.19E+18 | 1.35E+18 |
| favorites | 56571 | 2.83E+04 | 5.78E+04 | 0 | 1.00E+01 | 1.64E+02 | 4.39E+04 | 1.87E+06 |
| retweets | 56571 | 8618.987467 | 13306.13241 | 0 | 59 | 3450 | 13014.5 | 408866 |
| word\_count | 56571 | 19.587138 | 11.432801 | 1 | 12 | 19 | 23 | 60 |
| unique\_word\_count | 56571 | 18.386099 | 9.959306 | 1 | 12 | 18 | 22 | 52 |
| char\_count | 56571 | 127.783122 | 61.584192 | 2 | 92 | 132 | 142 | 328 |
| punctuation\_count | 56571 | 8.29524 | 4.428021 | 0 | 5 | 8 | 11 | 41 |
| hashtag\_count | 56571 | 0.170971 | 0.498546 | 0 | 0 | 0 | 0 | 10 |
| mention\_count | 56571 | 0.999098 | 1.100187 | 0 | 0 | 1 | 2 | 11 |
| flesch\_reading\_ease | 56571 | 60.779149 | 37.872037 | -472 | 46.44 | 62.68 | 76.22 | 206.84 |
| flesch\_kincaid\_grade | 56571 | 7.959667 | 5.766237 | -15.7 | 5.4 | 8 | 10.7 | 79.6 |
| joy | 56571 | 0.323045 | 0.622161 | 0 | 0 | 0 | 1 | 7 |
| positive | 56571 | 0.802266 | 1.032477 | 0 | 0 | 0 | 1 | 9 |
| anticipation | 56571 | 0.392516 | 0.680925 | 0 | 0 | 0 | 1 | 8 |
| surprise | 56571 | 0.197893 | 0.471001 | 0 | 0 | 0 | 0 | 6 |
| trust | 56571 | 0.510774 | 0.797271 | 0 | 0 | 0 | 1 | 7 |
| fear | 56571 | 0.28824 | 0.62288 | 0 | 0 | 0 | 0 | 8 |
| anger | 56571 | 0.285747 | 0.618378 | 0 | 0 | 0 | 0 | 7 |
| negative | 56571 | 0.542416 | 0.90651 | 0 | 0 | 0 | 1 | 8 |
| disgust | 56571 | 0.176698 | 0.472614 | 0 | 0 | 0 | 0 | 5 |
| sadness | 56571 | 0.268848 | 0.589611 | 0 | 0 | 0 | 0 | 7 |
| all\_negative | 56571 | 1.019533 | 1.935615 | 0 | 0 | 0 | 1 | 24 |
| all\_positive | 56571 | 1.424228 | 2.079896 | 0 | 0 | 1 | 2 | 23 |
| bRetweet | 56571 | 0.174595 | 0.379623 | 0 | 0 | 0 | 0 | 1 |
| bDeleted | 56571 | 0.019303 | 0.13759 | 0 | 0 | 0 | 0 | 1 |
| bFlagged | 56571 | 0.005374 | 0.073109 | 0 | 0 | 0 | 0 | 1 |

After removing rows with no real text (e.g., only @mentions and links):

| **variables** | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 54664 | 7.85E+17 | 3.80E+17 | 1.70E+09 | 4.51E+17 | 7.19E+17 | 1.18E+18 | 1.35E+18 |
| favorites | 54664 | 2.74E+04 | 5.71E+04 | 0 | 1.00E+01 | 1.36E+02 | 4.12E+04 | 1.87E+06 |
| retweets | 54664 | 8291.165813 | 12928.53364 | 0 | 53 | 2809.5 | 12599.5 | 408866 |
| word\_count | 54664 | 20.209644 | 11.122805 | 1 | 13 | 19 | 24 | 60 |
| unique\_word\_count | 54664 | 18.966779 | 9.622336 | 1 | 13 | 18 | 22 | 52 |
| char\_count | 54664 | 131.124707 | 59.871771 | 2 | 97 | 134 | 142 | 328 |
| punctuation\_count | 54664 | 8.373957 | 4.464803 | 0 | 5 | 8 | 11 | 41 |
| hashtag\_count | 54664 | 0.169856 | 0.494632 | 0 | 0 | 0 | 0 | 10 |
| mention\_count | 54664 | 1.022263 | 1.101327 | 0 | 0 | 1 | 2 | 11 |
| bRetweet | 54664 | 0.174466 | 0.379513 | 0 | 0 | 0 | 0 | 1 |
| bDeleted | 54664 | 0.018989 | 0.136486 | 0 | 0 | 0 | 0 | 1 |
| bFlagged | 54664 | 0.004647 | 0.068008 | 0 | 0 | 0 | 0 | 1 |
| flesch\_reading\_ease | 54664 | 69.046701 | 24.313353 | -301.79 | 57.61 | 71.51 | 83.36 | 206.84 |
| flesch\_kincaid\_grade | 54664 | 6.750146 | 4.038745 | -15.7 | 4.1 | 6.5 | 9.1 | 55.6 |
| joy | 54664 | 0.334809 | 0.630183 | 0 | 0 | 0 | 1 | 7 |
| positive | 54664 | 0.831992 | 1.040418 | 0 | 0 | 1 | 1 | 9 |
| anticipation | 54664 | 0.406904 | 0.689193 | 0 | 0 | 0 | 1 | 8 |
| surprise | 54664 | 0.205144 | 0.477998 | 0 | 0 | 0 | 0 | 6 |
| trust | 54664 | 0.52991 | 0.806159 | 0 | 0 | 0 | 1 | 7 |
| fear | 54664 | 0.299118 | 0.632208 | 0 | 0 | 0 | 0 | 8 |
| anger | 54664 | 0.296283 | 0.627143 | 0 | 0 | 0 | 0 | 7 |
| negative | 54664 | 0.562546 | 0.917186 | 0 | 0 | 0 | 1 | 8 |
| disgust | 54664 | 0.183155 | 0.479846 | 0 | 0 | 0 | 0 | 5 |
| sadness | 54664 | 0.278831 | 0.598158 | 0 | 0 | 0 | 0 | 7 |
| all\_negative | 54664 | 1.057387 | 1.960894 | 0 | 0 | 0 | 1 | 24 |
| all\_positive | 54664 | 1.476767 | 2.099663 | 0 | 0 | 1 | 2 | 23 |

Regarding the binomial variables, the counts are as follows:

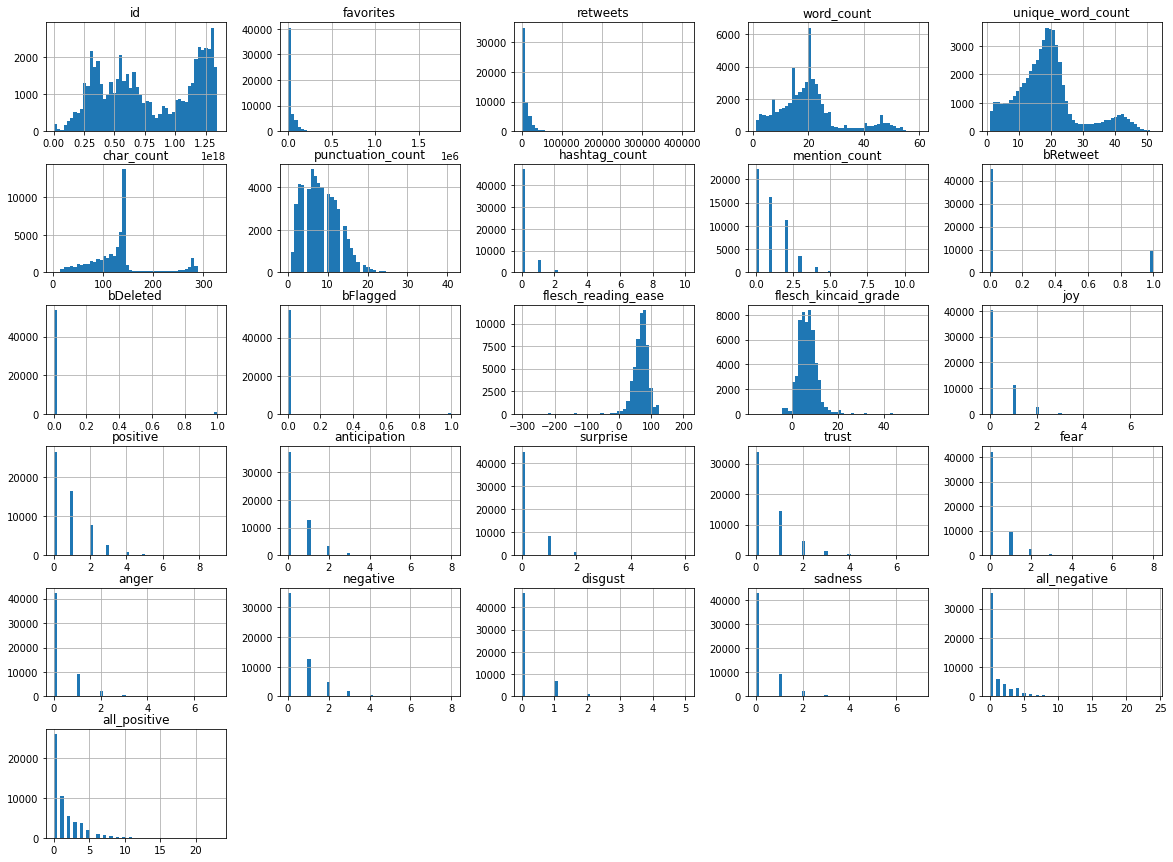
| **Level** | **isDeleted** | **isFlagged** | **isRetweet** |
| --- | --- | --- | --- |
| true | 1038 | 254 | 9537 |
| false | 53626 | 54410 | 45127 |

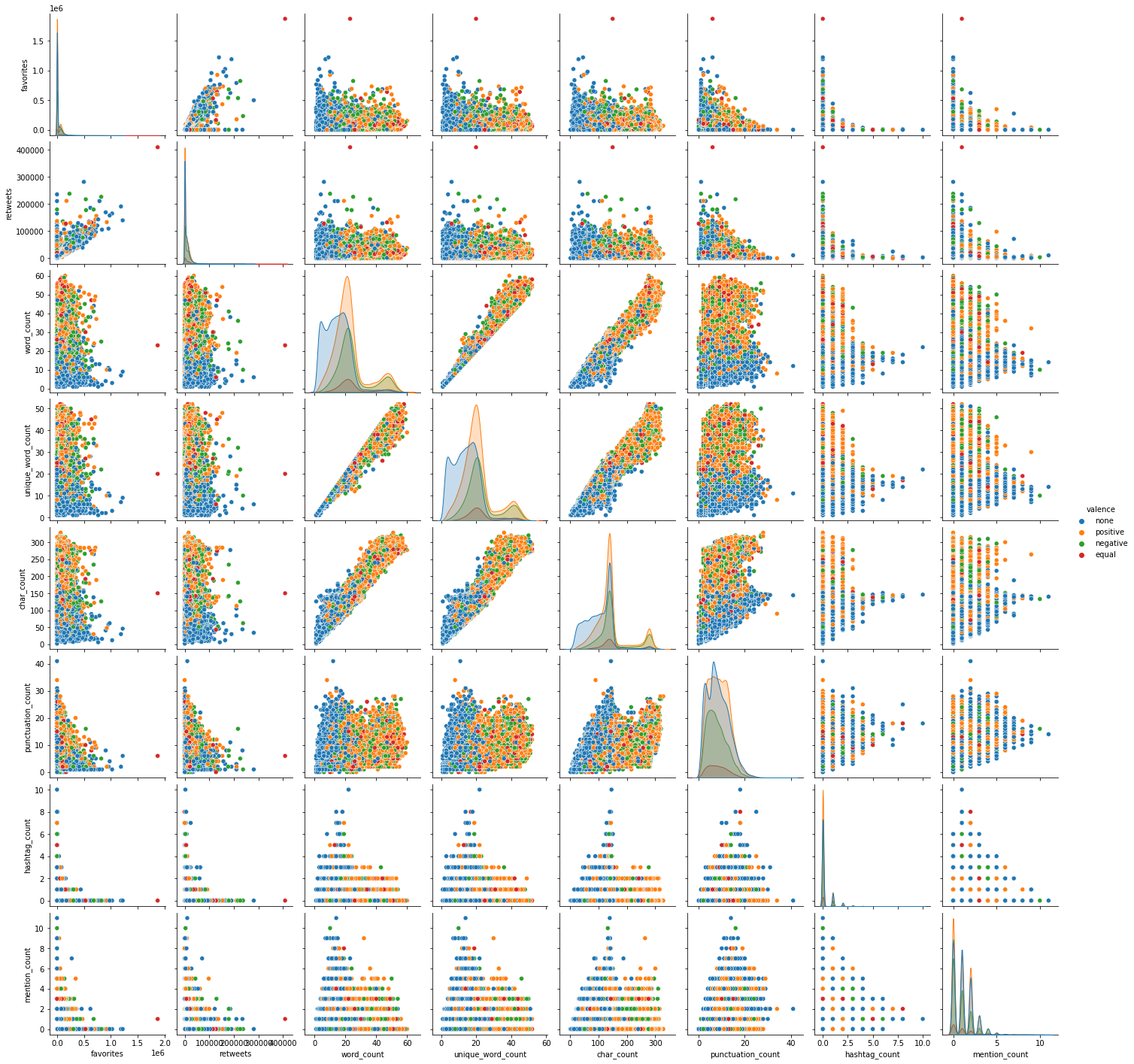
Finally, valence, the classification target has the following counts:

| **Level** | **valence** |
| --- | --- |
| positive | 20922 |
| none | 20150 |
| negative | 11267 |
| equal | 2325 |

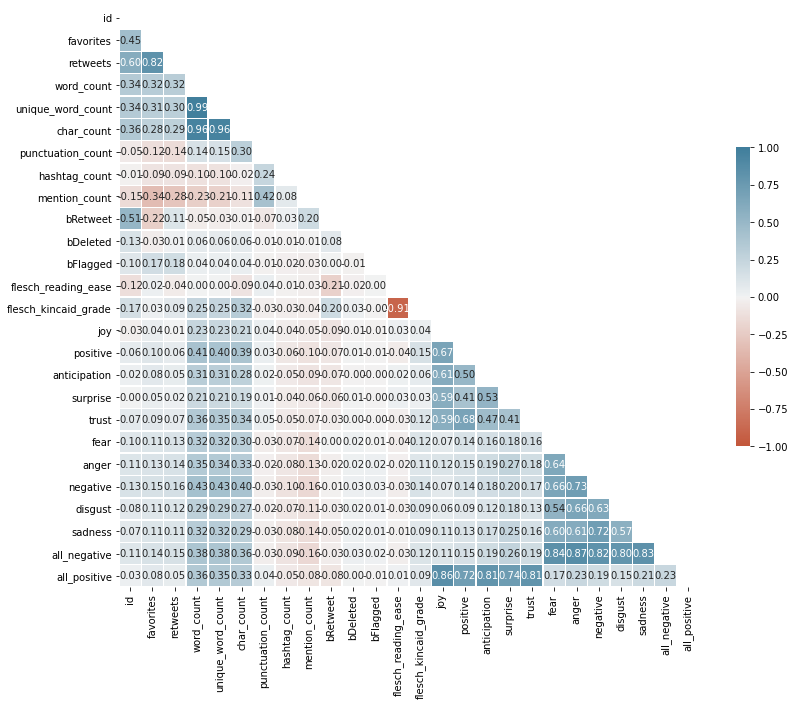
Graphs and correlations are provided below in the code to examine the data. Scaling of some method will be necessary, given the vastly different scales for the numeric, so long as it does not affect the binary variables.

Full size graphs are available in the file ‘Exploratory Data Analysis.ipynb’.

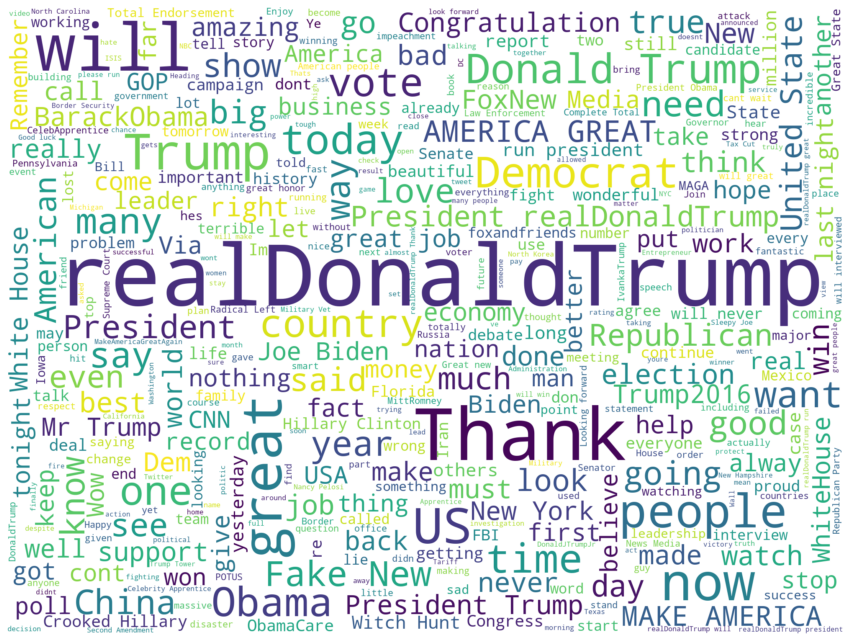




Above is an image of the numeric values related to tweets organized by valence, we can see that there are some characteristics different compared to different valences, this indicates that these might be good indicators alongside the tweet text.

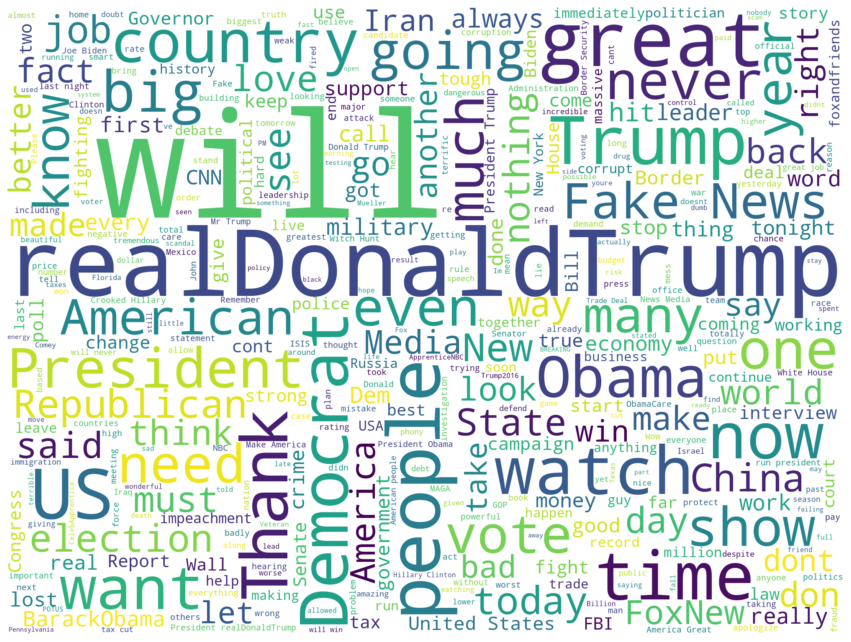
Below are the correlations of the numeric data, where we can see that the data is not highly correlated, except for the ones that are conceptually similar such as the reading ease and the grade that they should be at are negatively correlated. As well the word count and unique word counts are related as expected. Additionally we see a small relationship between the length of the tweet and how many words in that tweet have a sentiment meaning. That's because if a tweet is very long it has a more chance of having words that fall into this one of the sentiment categories. 

Below is the full dataset of tweets set in a word cloud. In this image the main word is ‘realDonaldTrump’ as it is a common one he seems to use that one a lot to tag himself so people can look follow that tag or people were tagging him with it. Additionally a few other trends become clear and pop out, but there is a lot of noise as it is the full dataset of tweets.



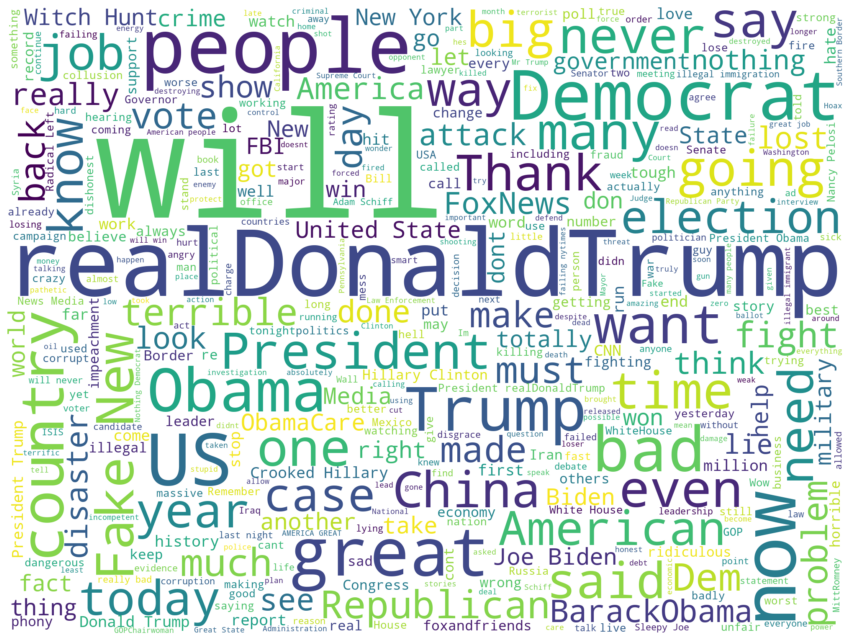
Next is the word clouds by their valence category. First is equal, and we can see that it is a mixed bag of positive and negative words as expected. Equal is made up of tweets that had positive and negative words in equal amounts.

Equal



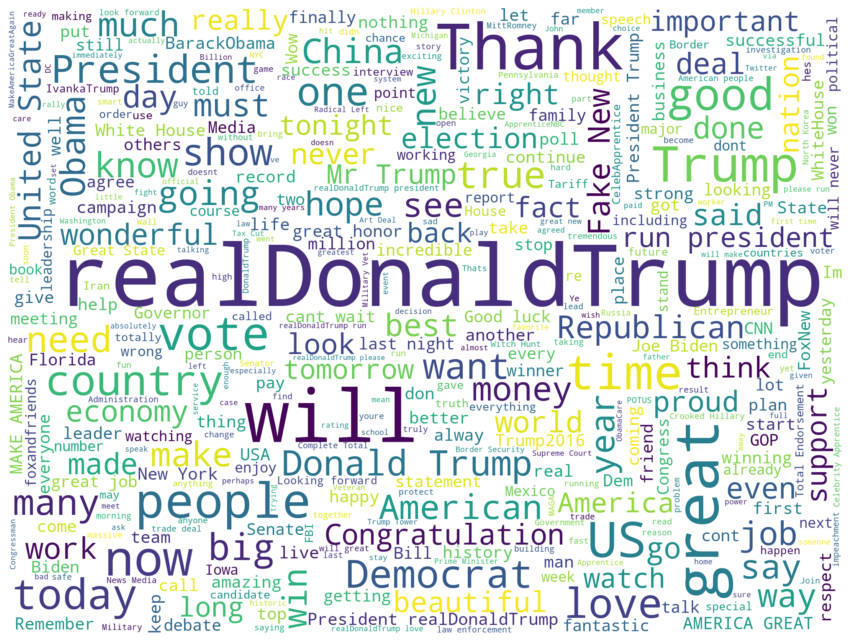
For the negative these are words that Donald Trump tends to use describing those he considers as ‘others’, which are his enemy or whoever he is in contention with at the time.

Negative



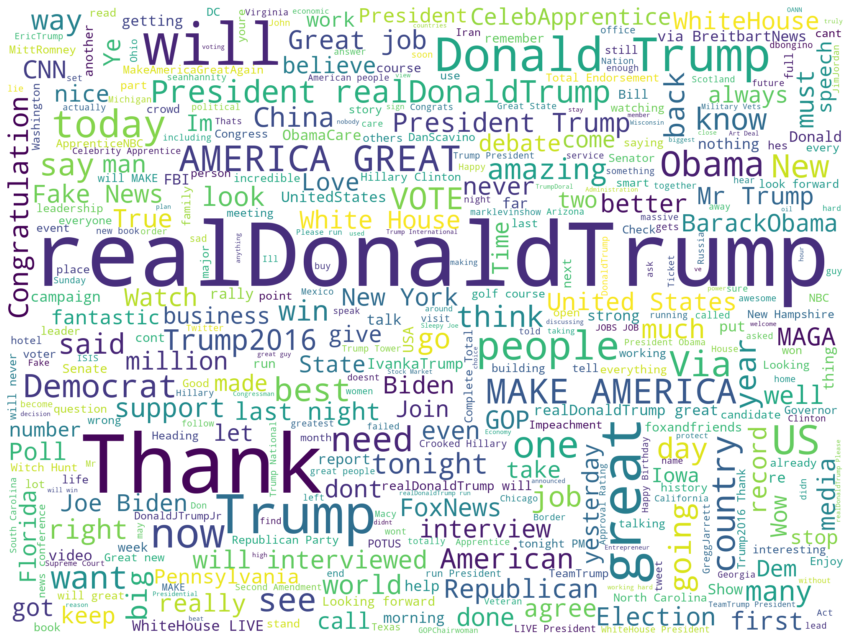
For the positive scored tweets we can determine that a lot more of the words he tends to use are those that are describing himself and/or thanking other people for contributions. These also seem to be aligned with his work and with his political career.

Positive



Finally then none, these tweets are a mixed bag of all the other tweets. Because the words did not exist in the NRC records, this is where ‘all the rest’ of the tweets end up.

None



# Classification Supervised Methods

Regarding these, the first will be attempting a classification model either binomial/logistic regression (negative vs positive), or multinomial/polynomial models (for all three to four categories, either done one v one or one v the rest). Issues with classification include class imbalance, given the numbers of the valence variable level are uneven. Actual algorithms include Support Vector Machines (SVM) or Naive Bayes, both will be attempted and compared. Additionally, an attempt will be made at using the BERT deep learning model. As the BERT model requires task-specific fine-tuning, it might be harder to implement. BERT would be run on positive/negative sentiment meaning as the target variable.

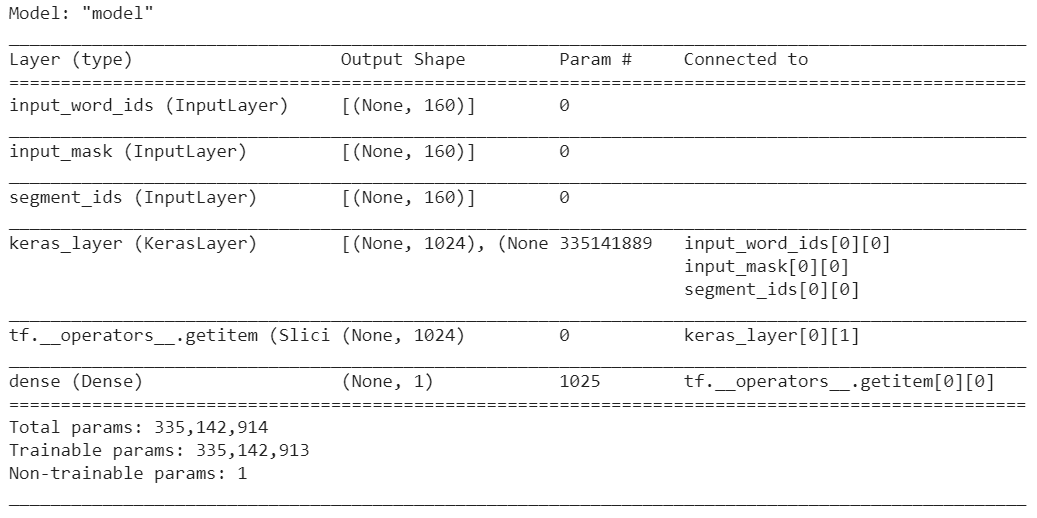
Full code can be found in ‘Supervised Methods.ipynb’.

## BERT

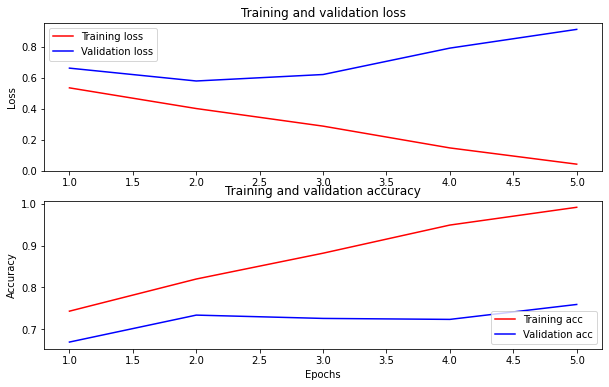
BERT stands for Bi-directional Encoder Representation from Transformers and is a deep learning model. It is a new method of using a model that has been pre-trained and then is now applied to the data of interest. How BERT works and why it is so good at NLP is best explained by articles dedicated to that purpose (see [Horev](https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270), [Wei](https://towardsdatascience.com/bert-why-its-been-revolutionizing-nlp-5d1bcae76a13), or [Rizvi](https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/) for some excellent explanations to better understand and program a BERT model).

I chose to use the BERT uncased english model. Since using it, the model has been updated, but this analysis uses the older model of BERT to match the initial results. The model uses 24 hidden layers, the hidden size is 1,024 and has 16 attention heads. This model was pre-trained for English on Wikipedia and BooksCorpus. The input type is uncased, meaning that the text has to be set to lowercase before tokenization and any accents have to be stripped in order to properly utilize this model. If the input has a mix of upper and lower case or if accent marks are important, then there are other pre-trained BERT models for use. While the text has also been tokenized using NLTK, it was important to use BERT’s tokenizer on the untokenized text, to better match the language in the pre-trained model.

Here is the model summary, it takes in three different input variables and then puts it into the keras layer. The keras layer taps into the BERT model itself and then it output the final prediction as a one-dimensional array.



The training was done over five epochs at batch size of fifteen. Having more epochs tended to overfit the training dataset whereas fewer epochs were too few to get a good performance for the test set. In total, training time was one hour on Google's TPU. Below is the accuracy versus loss for both the training and validation set.



The only input variable was the cleaned tweet text itself, no other variables were added. These prediction results were purely taking in text information so this is a very good performance relatively for using just text data. However, it is not performing as well as expected given the strength and power behind the BERT model. There clearly could be additional tweaks and fine-tuning that would need to be done to hone the model in order to get the best performance.

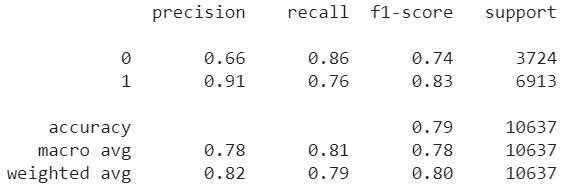
Overall the metrics were as follows:

Accuracy: 0.792

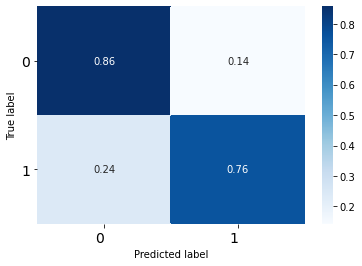
Recall: 0.757

F1: 0.826

Precision: 0.908



Performance was relatively good for classifying negative tweets, but not as good at predicting positive tweets. Below is the normalized confusion matrix to view the predicted vs true labels. Negative tweets are 0, and positive tweets are 1.



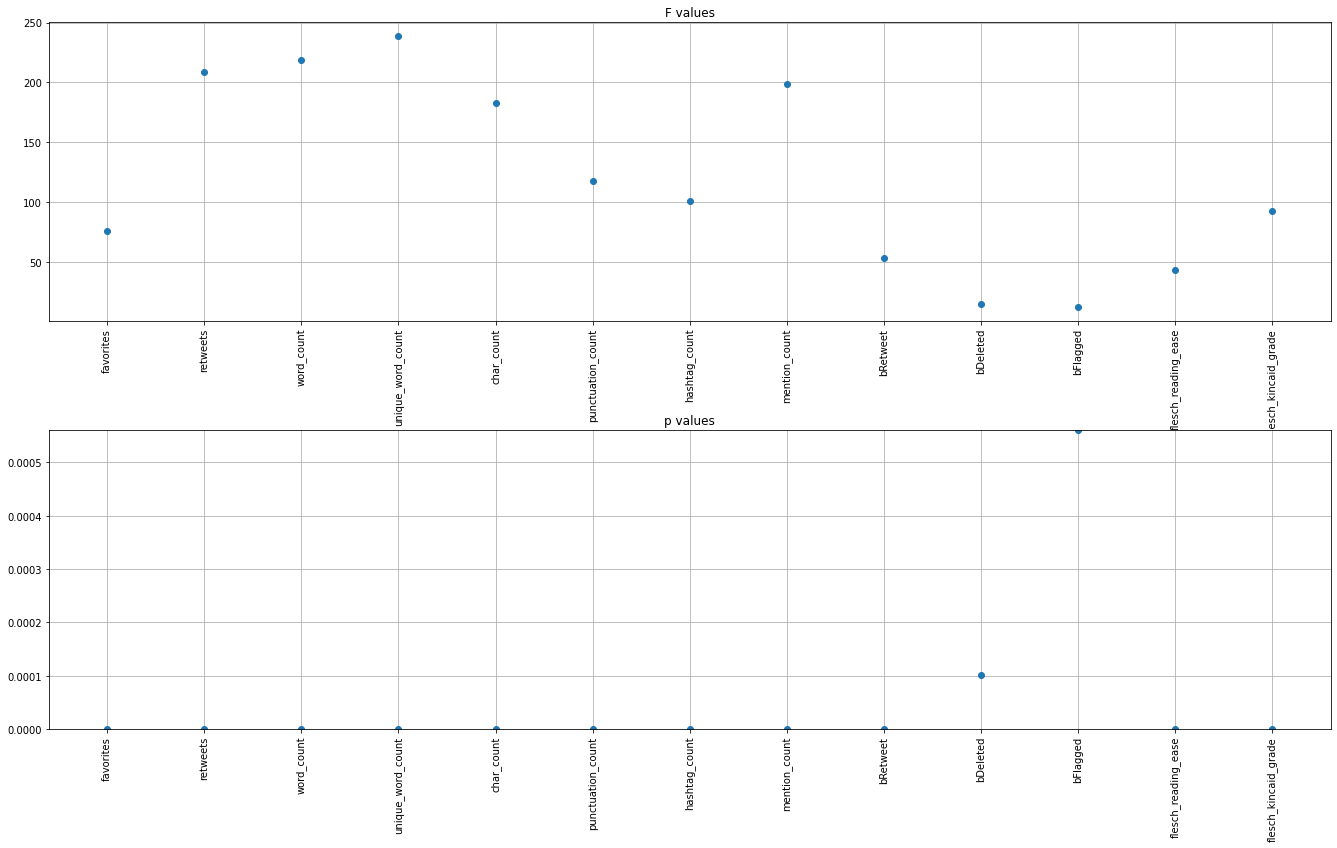
One potential way to increase the performance of BERT is to pass the numeric data alongside the text vectors. In BERT the data is padded, so that all vectors have the same length, it would be possible to append the numeric variables at the beginning of the vectors as special encodings (see [Freischlag](https://towardsdatascience.com/combining-numerical-and-text-features-in-deep-neural-networks-e91f0237eea4) for more information about this concept). Another method would be to use ensemble Deep Learning methods such as using Multilayer Perceptron for the numeric variables alongside the BERT for text and combining the results.

## SVM

SVM stands for Support Vector Machines, they are useful after text classifying because SVM utilize encoded text vectors to model the vectors and decide where to draw the best hyperplane. As well it also allows for the use of text vectors alongside numerical data without having to build separate models.

### Feature Selection

Before going too in depth, it was important to check that the numeric variables I wanted to use would be appropriate. To do so, one can look at F-Classification metrics. Below are the result of those analysis (which is based on Chi-Squared analysis), if a measure has a low correlation and a high F then the variable should be used. In this case, all the identified variables performed well and will be used in the SVM.



### 

### SVC

Initially, SVC was run with a variety of hyperparameters but it soon became clear that linear classifier was outperforming the rbf. So due to the long run times, this was abandoned and examined how a standard SVC would perform as compared to a LinearSVC.

Accuracy 0.680

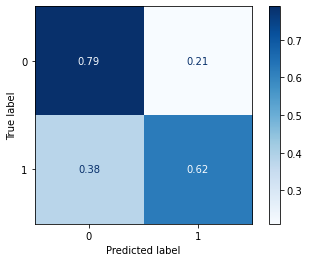
Recall 0.620

F1 0.715

Precision 0.845

In most ways, the results are poor, except for precision score which was 0.845.

In examining the normalized confusion matrix and classification report, it is clear it performs ‘okay-ish’ on classifying negative tweets but not well at all on positive tweets.



### Linear SVC

Using GridSearchCV, I examined the best hyperparameters for this model (hyperparameters in question are in the notebook). After the model ran, the best hyperparameters were

Best Score: 0.8962612788631157

Best Hyperparameters: {'clr\_\_C': 2, 'clr\_\_dual': False, 'clr\_\_fit\_intercept': False, 'clr\_\_penalty': 'l1', 'preprocessor\_\_tfidf\_\_tfidf\_\_analyzer': 'word', 'preprocessor\_\_tfidf\_\_tfidf\_\_ngram\_range': (1, 2), 'preprocessor\_\_tfidf\_\_tfidf\_\_smooth\_idf': False, 'preprocessor\_\_tfidf\_\_tfidf\_\_stop\_words': 'english', 'preprocessor\_\_tfidf\_\_tfidf\_\_use\_idf': True}

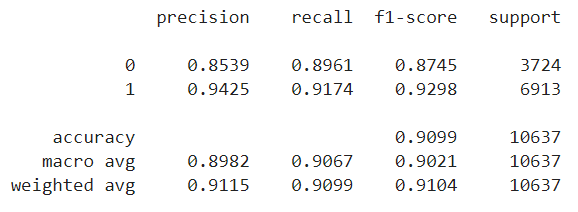
Against the test set, the model performed very well on the following metrics:

Accuracy: 0.910

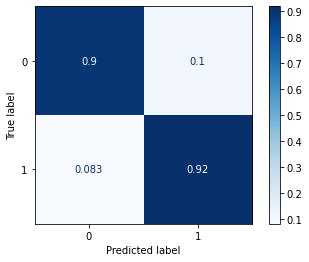
Recall: 0.917

F1: 0.930

Precision: 0.942



In examining the normalized confusion matrix below and the classification report above, it is clear that the ability to differentiate negative from positive tweets have greatly increased. In fact, the ability to determine positive tweets now slightly outperforms the negative tweets.



Either way, the model performs very well, and this gain seems to be a direct result of adding in the numeric data and recognizing that it is a linear relationship.

## Naive Bayes

Naive bayes was selected to see if it could outperform SVC. However, it is not expected that is will. This turns out to be true. Several methods will be examined, including multinomial for the text data, gaussian for the numeric, and then combining their probabilities to simulate the results of SVC.

### Multinomial

The multinomial Naive Bayes was performed only on the text column, similar to the BERT model. The methodology in this case was slightly different than others in that instead of using TF-IDF Vectorizer, the Count Vectorizer and TF-IDF Transformer were used. The TF-IDF Vectorizer is a combination of the two, but in this case it was important to make sure they were different named steps in the pipeline for better specificity. Technically the IDF should not be used in Naive Bayes (and this held up as it performed better without it), but I wanted to examine the effect it would have.

text\_clf = Pipeline([('vect', CountVectorizer()),

('tfidf', TfidfTransformer()),

('clf', MultinomialNB())])

tuned\_parameters = {

'vect\_\_ngram\_range': [(1, 1), (1, 2), (2, 2)],

'tfidf\_\_use\_idf': (True, False),

'tfidf\_\_smooth\_idf': [True, False],

'vect\_\_stop\_words': ['english', None],

'tfidf\_\_norm': ('l1', 'l2'),

'clf\_\_alpha': [1, 1e-1, 1e-2]

}

In the end the best parameters were as below:

Best set of parameters : {'clf\_\_alpha': 0.1, 'tfidf\_\_norm': 'l1', 'tfidf\_\_smooth\_idf': True, 'tfidf\_\_use\_idf': False, 'vect\_\_ngram\_range': (1, 1), 'vect\_\_stop\_words': 'english'}

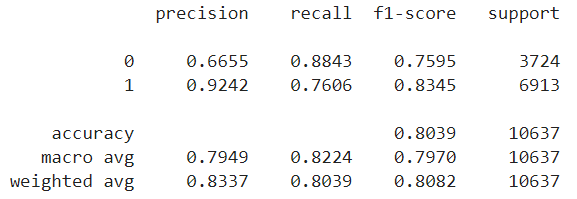
The scoring metrics for the testing dataset were moderate (for accuracy and F1), fine (for recall) and good (precision).

Accuracy 0.804

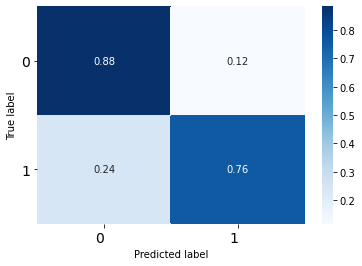
Recall 0.761

F1 0.834

Precision 0.924



In examining the normalized confusion matrix below and the classification report above, it is clear that the model can predict the negative tweets well but falters slightly on predicting the positive ones well.



As well, this is only done on the text based features and seems in line with the results of BERT. So it might be possible to boost the performance by adding in the numeric variables using a Gaussian model.

### Gaussian

The Gaussian model was run with the below numeric variables. These are the same as those used in the LinearSVC model. However, some of these are not Gaussian type variables, so that was also a concern with this model type.

numerical\_features = ['favorites', 'retweets','word\_count',

'unique\_word\_count', 'char\_count', 'punctuation\_count',

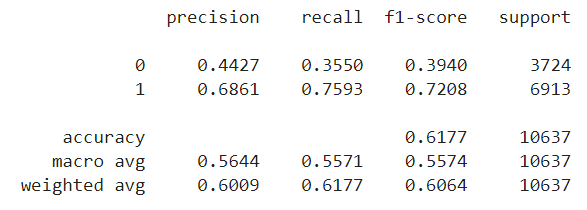
'hashtag\_count','mention\_count', 'bRetweet', 'bDeleted',

'bFlagged','flesch\_reading\_ease', 'flesch\_kincaid\_grade']

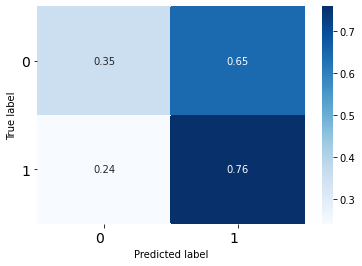
Using hyperparameter tuning in GridSearchCV, the best score was only 0.55, which was of concern. The performance metrics were not calculated as this was already assumed to have not worked, presumably because the variables lack a gaussian distribution.

Best set of parameters : {'clr\_\_var\_smoothing': 0.00019563983435170627}

Best score : 0.5551505023554398



As seen in the normalized confusion matrix below and the classification report above, the model is actually surprisingly good at determining the positive tweets, but very bad at determining the negative ones. This does give credence to the concept that loading for the positive tweets is on the numeric data, while negative loadings are found moreso in the textual data.



### Mixed

I combined the result of the probabilities for the multinomial probabilities and the gaussian probabilities to get an overall mixed method.

probabs = multinomial\_probas\*gaussian\_probas

log\_probabs = np.log(multinomial\_probas)+np.log(gaussian\_probas)

Then, because the methodology results in lower probabilities the items are adjusted based on the mean score. Those over the mean are turned into 1 and those below the mean are now 0.

ysb=0.25

def changing(input):

output = []

for i in input:

if i[1] >= ysb:

output.append(1)

else:

output.append(0)

return output

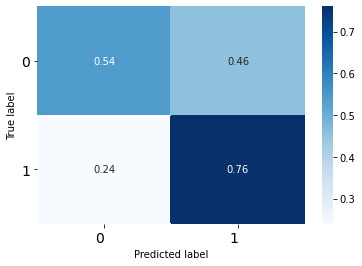
Accuracy 0.685

Recall 0.760

F1 0.758

Precision 0.756

The final result is an alright model but not as good as multinomial by itself. There is no improvement in positive prediction and only a decline in negative prediction.



# Clustering Unsupervised Methods

The second model will revolve around topic modeling (which is, in a way, a type of clustering, although it aims at discovering latent themes moreso than partitioning the data into coherent groups as in clustering). Both these methods require text vectorization (such as through Gensim, GloVe, or scikit's Vectorizer). That is why both topic modeling and clustering are presented at the same time as the intent is to attempt both, including potentially adding the results of the topic modeling as another feature. Clustering can be done with K-Means or with Hierarchical Clustering, but K-Means will be used here.

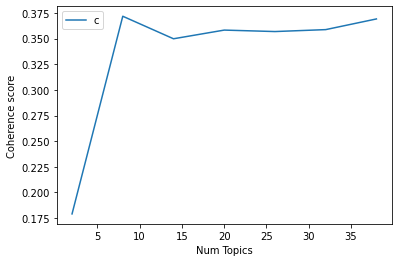
Full code can be found in “Unsupervised Methods.ipynb”.

## LDA

LDA stands for Latent Dirichlet Allocation, and it is a way of examining the tweet text for topics, themes and groups. LDA works by learning the representation of a fixed number of topics. When given this number it learns the topic distribution of each document in a collection. In this case, by passing the tweets as a collection, or corpora, to the model it is possible to determine which ones ‘stick’ conceptually together.

### Full Dataset

First the method was done using Gensim’s built in LDA, but attention was soon moved to the Java Mallet Wrapper and this improved the coherence score. Initially the score was 0.282, but with Mallet and tuning the number of topics following the graph below the score was raised to 0.372, which is a significant improvement.



Please see the attached “use\_lda.html” for the interactive LDA model. The table below shows the topic numbers, keywords, and the tweet with the highest contribution to that topic.

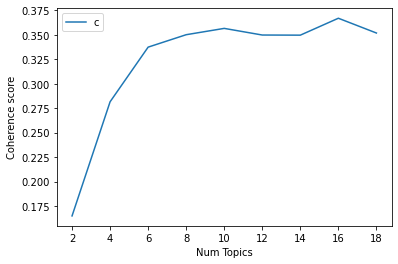
| Dominant\_  Topic | Topic\_Contrib | Keywords | Text |
| --- | --- | --- | --- |
| 0 | 0.2457 | realdonaldtrump, trump, run, president, love, true, foxandfriend, agree, hear, real | kenziestruder realDonaldTrump for pres 2016 raisinghands raisinghands raisinghands raisinghands raisinghands raisinghands |
| 1 | 0.3047 | people, vote, work, state, american, strong, support, hard, fight, border | Thank you Pennsylvania With your vote we will continue to support our Military Veterans and our Police Officers Biden and Harris would pass legislation to slash police funding all across America They stand with the rioters we stand with the HEROES of Law Enforcement MAGA |
| 2 | 0.2705 | bad, win, election, give, thing, happen, medium, campaign, put, poll | When I won the Election in 2016 the nytimes had to beg their fleeing subscribers for forgiveness in that they covered the Election and me so badly They didn’t have a clue it was pathetic They even apologized to me But now they are even worse really corrupt reporting |
| 3 | 0.2664 | big, watch, show, tonight, talk, start, interview, live, tomorrow, speak | Entrepreneurs Knowledge requires patience action requires courage Put patience and courage together and you’ll be a winner |
| 4 | 0.2893 | great, today, day, whitehouse, family, congratulation, amazing, life, honor, nation | Yesterday it was my great honor to recognize extraordinary Law Enforcement Officers and First Responders and to award them the the highest possible decoration for bravery by public safety officers the Medal of Valor AmericanHeroes |
| 5 | 0.3015 | great, make, country, good, deal, world, history, man, leave, leader | Tariffs will bring in FAR MORE wealth to our Country than even a phenomenal deal of the traditional kind Also much easier amp quicker to do Our Farmers will do better faster and starving nations can now be helped Waivers on some products will be granted or go to new source |
| 6 | 0.308 | call, dem, story, report, fact, lie, end, build, totally, read | These were Mueller prosecutors and the whole Mueller investigation was illegally set up based on a phony and now fully discredited Fake Dossier lying and forging documents to the FISA Court and many other things Everything having to do with this fraudulent investigation is |
| 7 | 0.3544 | job, time, year, amp, back, money, pay, economy, lose, record | On Taxes This is the biggest corporate rate cut ever going back to the corporate income tax rate of roughly 80 years ago This is a huge progrowth stimulus for the economy Every year the Obama WH overstated how the economy would grow Now real economics and jobs! WSJ Report |

As evident, there are 8 themes, with those that are negatively oriented as 6 and 2, while those with more positive tones are 3, 4, and 5. We can see that realDonaldTrump aligns very closely with zero (one in the graph on the html) but country aligns with 5 but also 0. On zero here we have realDonaldTrump, job, etc. While zero and 3 are about news but good news or news that's been kind to him such as fox news.

While 1 is more about people/americans and being nationalistic. While the negative ones are where almost everything bad is, so democrats, reports, lies, and it also includes impeachment, attacks, questions, collusions. While 6 bad but also includes win, election, campaign, so it is sort of about political and disasters. While the positive ones use words like great country, good, and white house, congratulations, amazing. So the themes revolve not only around concepts but also around these sentiment words.

### Original Posts Only

I also examined the data by looking at the tweets written by Donald Trump alone. In this case the optimal convergence is slightly lowered, but 16 topics arise from the dataset. The final convergence is 0.3671.



| Dominant\_Topic | Topic\_Contrib | Keywords | Text |
| --- | --- | --- | --- |
| 0 | 0.239 | vote, election, build, change, candidate, close, republican, presidential, remember, result | go to your Polling Place to see whether or not your Mail In Vote has been Tabulated Counted If it has you will not be able to Vote the Mail In System worked properly If it has not been Counted VOTE which is a citizenâ€™s right to do If your Mail In Ballot arrives |
| 1 | 0.1817 | great, make, love, day, realdonaldtrump, amazing, agree, yesterday, respect, fantastic | Back by popular demand the fabulous LilJon returns to the record setting 13th season of All Star CelebApprentice The fans love him |
| 2 | 0.2012 | state, money, lose, time, spend, attack, send, government, save, meet | As I predicted 16 states led mostly by Open Border Democrats and the Radical Left have filed a lawsuit in of course the 9th Circuit California the state that has wasted billions of dollars on their out of control Fast Train with no hope of completion seems in charge |
| 3 | 0.2546 | job, pay, high, number, record, year, hit, low, buy, trade | We are getting other countries to reduce and eliminate tariffs and trade barriers that have been unfairly used for years against our farmers workers and companies We are opening up closed markets and expanding our footprint They must play fair or they will pay tariffs |
| 4 | 0.1918 | realdonaldtrump, trump, run, president, true, real, leader, hope, office, politician | kenziestruder realDonaldTrump for pres 2016 raisinghands raisinghands raisinghands raisinghands raisinghands raisinghands |
| 5 | 0.2033 | watch, show, tonight, talk, foxandfriend, interview, tomorrow, live, foxnews, rating | A fantastic new book â€œAmerican Crusadeâ€ written by a Great American Patriot PeteHegseth â€” is available tomorrow Hear more about it as he joins SeanHannity tonight at 900pmE and foxandfriends tomorrow morning at 830amE Get your copy today |
| 6 | 0.2542 | medium, story, totally, report, fact, fail, book, read, bad, wrong | The Washington Post is far more fiction than fact Story after story is made up made garbage more like a poorly written novel than good reporting Always quoting sources not names many of which donâ€™t exist Story on John Kelly isnâ€™t truejust another hit job |
| 7 | 0.2051 | great, big, win, congratulation, poll, leave, debate, nice, play, guy | The biggest difference between now and 2016 is FoxNews They are a whole different deal Despite this our campaign is doing much better with bigger crowds and even more much enthusiasm than we had in 2016 Big Debate SCOTUS Win Real Polls have us winning everywhere |
| 8 | 0.1942 | amp, country, year, history, sign, plan, fast, destroy, fix, economy | I urge the Senate and House to pass the Paycheck Protection Program and Health Care Enhancement Act with additional funding for PPP Hospitals and Testing After I sign this Bill we will begin discussions on the next Legislative Initiative with fiscal relief |
| 9 | 0.2584 | strong, stop, support, job, fight, border, love, people, crime, military | Congressman LanceGooden has done a wonderful job for the people of Texas while supporting our MAGA Agenda He continues to protect your very important 2A Lance is Strong on Crime and the Border he Loves our Great Vets and Military Lance has my Complete amp Total Endorsement |
| 10 | 0.1893 | work, give, hard, problem, lot, smart, success, people, stay, focus | To everyone on the Gulf Coast As you make preparations to protect your homes amp loved ones from flooding amp the coming storm it is imperative that you heed the directions of FEMA State amp Local Officials We are working closely w them Please be prepared be careful amp be SAFE |
| 11 | 0.2302 | campaign, dem, lie, total, deal, find, start, political, turn, disaster | foxandfriends “New Bombshell in the Obama Spying Scandal Did other Agencies SPY on Trump Campaign” Even Clapper worlds dumbest former Intelligence Head who has the problem of lying a lot used the word SPY when describing the illegal activities |
| 12 | 0.192 | time, bad, long, speak, hear, fire, make, terrible, person, top | Watching the DodgersRed Sox final innings It is amazing how a manager takes out a pitcher who is loose amp dominating through almost 7 innings Rich Hill of Dodgers and brings in nervous relievers who get shellacked 4 run lead gone Managers do it all the time big mistake |
| 13 | 0.259 | great, today, man, wonderful, family, honor, friend, nation, beautiful, woman | The courage & sacrifice of our heroes is the reason our flag stands tall our hearts beat with pride and our Country remains one people one family and one NATION UNDER GOD Today we thank you we honor you amp we forever cherish the memory of our Fallen Men and Women in Blue |
| 14 | 0.2272 | good, make, country, thing, world, happen, business, deal, lead, case | A message from Kim Jong Un â€œNorth Korea will stop nuclear tests and launches of intercontinental ballistic missiles!Also will â€œShut down a nuclear test site in the country’s Northern Side to prove the vow to suspend nuclear tests! Progress being made for all |
| 15 | 0.2306 | people, back, call, american, put, bring, end, stand, question, home | Mark Esperanto Secretary of Defense â€œThe ceasefire is holding up very nicely There are some minor skirmishes that have ended quickly New areas being resettled with the Kurds! USA soldiers are not in combat or ceasefire zones We have secured the Oil Bringing soldiers home |

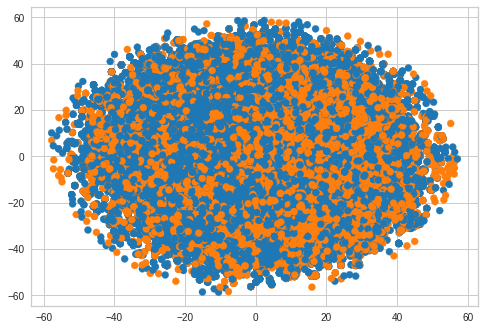
Since there are more topics, there is more to unpack but roughly it's more of a breakdown of Donald Trump’s life. For example there are topics that are more related to his work pre-white house. And others are more about general American life and support for causes. But there are also some familiar themes with a topic on fake news and evil democrats rising up. While some positive tweets are about civic duty such as voting and support for republicans. It is interesting to see this more as the strata of his life, with some being coded as positive or negative but more having a neutral overall code.

## t-SNE

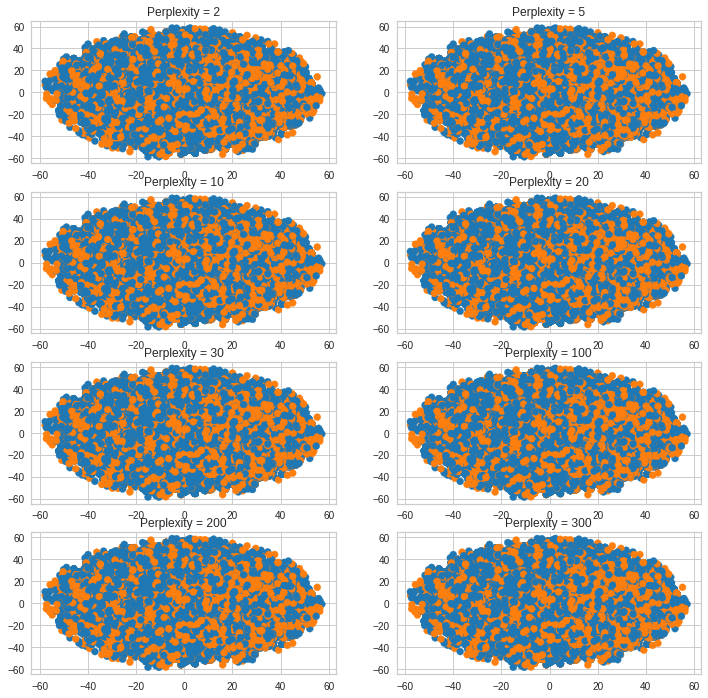
t-SNE stands for t-distributed stochastic neighbor embedding, which is a type of dimensionality reduction and neighbor clustering in one. In this case, the method was not applicable to the data (either for text vectors or for the numeric data), so nothing could be determined.

### Text Vectors

Below is the graph of the text vectors, where the color is determined by the valence coding. The end result is a blob with no defined clusters.



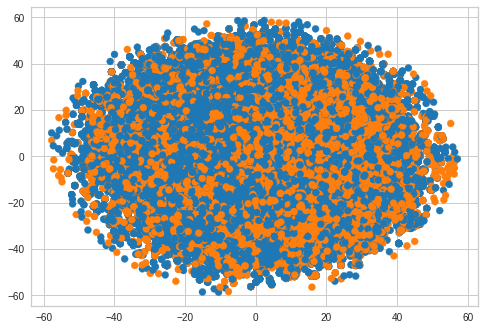
Even adjusting for perplexity did not resolve this issue as shown below.



Thinking that this was unique to the text vectors, I also examined the numeric data.

### Numeric Variables

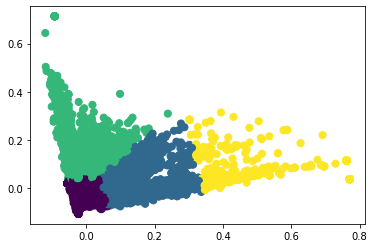
However, this issue remained even with just the numeric data. Thus this dataset does not seem to be a good candidate for t-SNE.



## K-Means

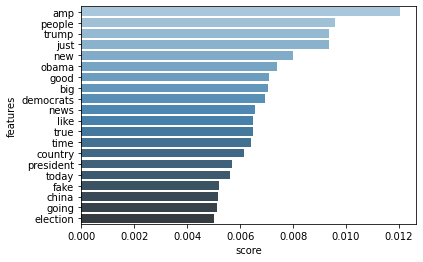
Instead, I turned to K-means to examine the clusters. For K-Means, I examined the tweet text for groups and then examined those clusters to view themes. K-Means works by placing centroids, then evaluating the cluster distances, and then re-centering the centroid until the centroid no longer moves. Given the data was text vectors, a slightly different elbow method was used below to determine optimal k. In this case, 4 groups were selected.

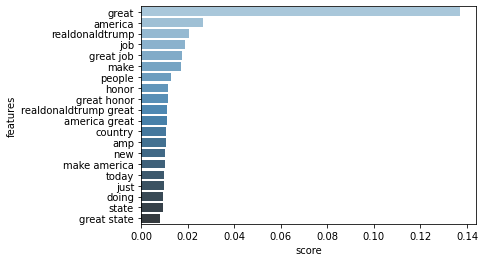


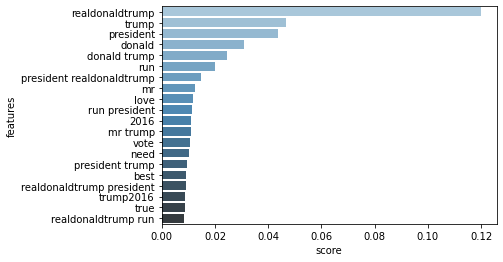


Above are the plotted text vectors with groups colored, while below show the text associated with each group.









Examining the top words in each cluster, one can get an idea of if these are actually making sense. As seen herem one of the clusters is very different from the other three, This was a concern as it could be that while the vectors seem to yield 4 clusters, three of the clusters seem similar textually. Those that are the one ones that are very skewed are a lot more similar to each other than the one that stands alone. So this method might not be the best method for teasing out different clusters as compared to using LDA when it comes to textual data.

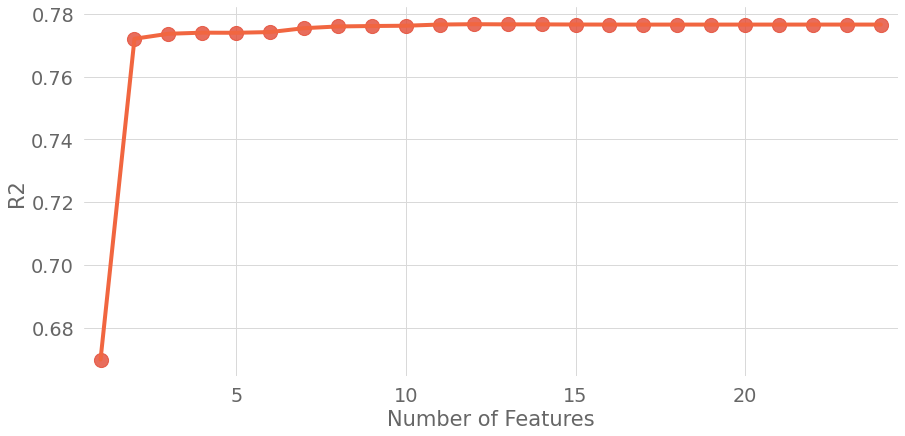
# Regression Supervised Methods

Although not required or proposed earlier, I also wanted to take a look at regression. For these analyses, the target variable was ‘favorites’. I wanted to determine if there was a pattern to the popularity. In this case I started with RFECV feature selection. I already know that Retweets will be a strong covariate.

## Feature Selection - RFECV

After running the RFECV, I kept the following variables. Of most interest to me was that of the sentiment variables, only negative was kept.

| retweets | TRUE |
| --- | --- |
| word\_count | TRUE |
| unique\_word\_count | TRUE |
| char\_count | TRUE |
| punctuation\_count | TRUE |
| hashtag\_count | FALSE |
| mention\_count | TRUE |
| bRetweet | TRUE |
| bDeleted | TRUE |
| bFlagged | TRUE |
| flesch\_reading\_ease | TRUE |
| flesch\_kincaid\_grade | TRUE |
| positive | FALSE |
| negative | TRUE |
| sadness | FALSE |
| anger | FALSE |
| fear | FALSE |
| disgust | FALSE |
| surprise | FALSE |
| trust | FALSE |
| anticipation | FALSE |
| joy | FALSE |
| all\_positive | FALSE |
| all\_negative | FALSE |



As we can see in the above graph, having ‘retweets’ is the strongest predictor while the others provide only a marginal gain. However, 11 variables were chosen in total as matching the above table.

## Linear Regressions

### Linear Regression/Lasso/Ridge

These were originally meant to be separate analyses but they all converged on the same answer, R2, and RMSE. For the Linear Regression:

Best set of parameters : {'clr\_\_fit\_intercept': True, 'clr\_\_normalize': False}

Best score : 0.7767293934098224

Informed Variance : 0.7924657202107629

R2 : 0.7924624615800115

RMSE : 26240.73419580512

For Lasso:

Best set of parameters : {'clr\_\_alpha': 1.5, 'clr\_\_fit\_intercept': True, 'clr\_\_max\_iter': 100000, 'clr\_\_normalize': False}

Best score : 0.7767295379190179

Informed Variance : 0.792468826081726

R2 : 0.7924655558786873

RMSE : 26240.538575840787

For Ridge:

Best set of parameters : {'clr\_\_alpha': 1.8, 'clr\_\_fit\_intercept': True, 'clr\_\_max\_iter': 100000, 'clr\_\_normalize': False, 'clr\_\_solver': 'sag'}

Best score : 0.7767506566697744

Informed Variance : 0.7924720190602903

R2 : 0.7924686821471676

RMSE : 26240.340933280906

In the end these all ended up with the same result, however I was not happy with the RMSE (although the R2 was in line with what was expected given the variables explained that much of the variance anyway). So I moved on to Random Forest Regression.

## Random Forest Regression

Random Forest Regression makes use of many decision trees to cut the data until the optimal solution is found.

Best set of parameters : {'clr\_\_max\_depth': 10, 'clr\_\_n\_estimators': 900}

Best score : 0.9139271225352501

Informed Variance : 0.9320700119153125

R2 : 0.9320700109511016

RMSE : 15012.686780963088

In this case, the Random Forest Regression model was a significant improvement over linear regression as it accounted for 93% of the variance with a much reduced RMSE. Thus these variables are a good way of determining if a tweet is popular, and one aspect of that is whether or not the tweet contains negative words. In fact, the more negative the tweet, the more likely it is to be popular.

# Model Comparisons

| Model | BERT | Linear SVC | Multinomial Naive Bayes |
| --- | --- | --- | --- |
| Accuracy | 0.792 | 0.910 | 0.804 |
| Recall | 0.757 | 0.917 | 0.761 |
| F1 | 0.826 | 0.930 | 0.834 |
| Precision | 0.907 | 0.942 | 0.924 |

Overall, LinearSVC performed best for classification and Random Forest Regression performed best for regression. When it came to clustering, LDA performed better at generating understandable topics than k-means, but it is designed to do that better as it makes use of more than just text vectors.

# Conclusions

In conclusion, the models worked fairly well, with LinearSVC as the best classifier and Random Forest Regression as the best regressor. LDA was useful in determining topic clusterings. The limitations included having additional variables from twitter that could help build additional data points, as well as computing power (sometimes the TPU or GPU were not available even though I am a paying colab customer). The next steps involve further tuning the BERT model given the two alternate improvements mentioned in the BERT section as it should be able to perform better. As well, other options include determining if there are other ways that the data can be examined to yield further insights.