How did the Everyone Feel During the Election?

James Nelson

MSDS696 Data Science Practicum II

Professor Busch, Michael

Regis University

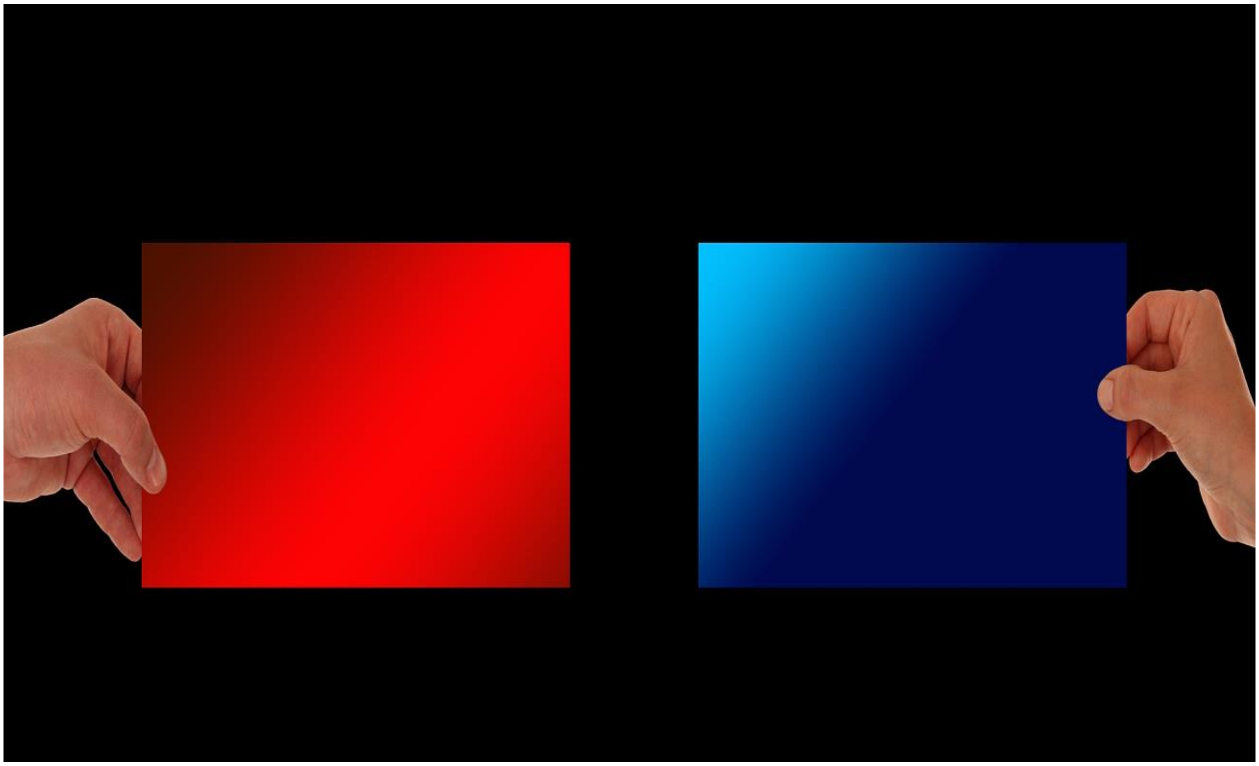


Photo by Geralt on Pixabay (geralt, 2019)

Introduction: This year's election is definitely one that everyone could agree upon is out of the norm. There is heightened tension on opposite sides and a lot of people who are worried about the outcome. The point of this research project is to gain a better understanding/perspective on the sentiment towards this historic event and model what seems to be the topic on mind.

What will this sentiment analysis and topic model answer? That is a question that I am trying to solve and answer myself. After watching the Netflix documentary "The Social Dilemma," and having suspicions of my own about how the general population is in a state of polarization, I want to see if both my sentiment analysis and topic model can provide a deeper understanding of the issue especially with this election.

Contents

(Image by qimono from pixabay, 2016)

1. Understanding The problem

1.1 The Problem

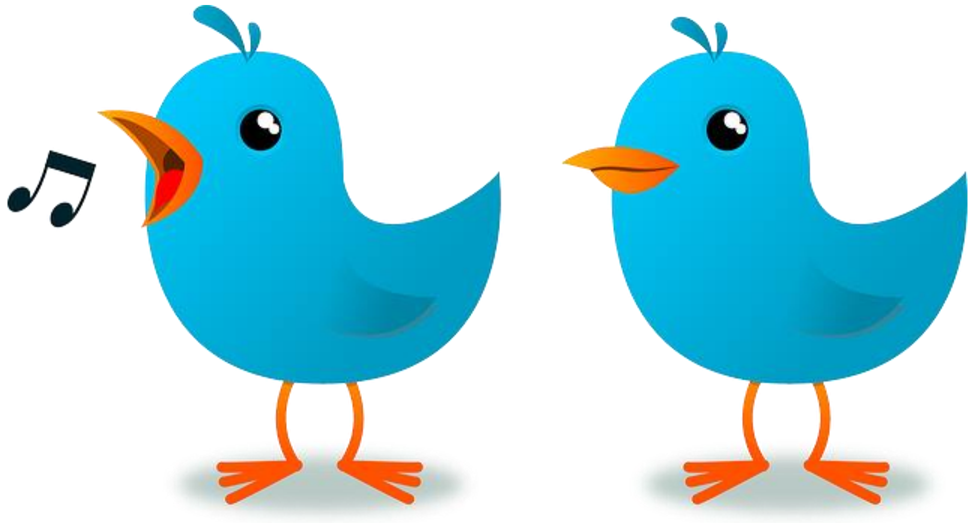
The problem right now that I feel that this country is now facing is the decreased amount of exposure for the opposite side's political opinion. After watching the Netflix documentary "The Social Dilemma," having confirmed my suspicions on social media, search engines, and ad targeting algorithms I wanted to discover how bad this problem could possibly be. However, researching this issue would take months or even years to collect the right amount of data and explore all the factors/issues that tie into this problem, I have decided to start out small with a sentiment analysis and topic modeling.

1.2 The Plan

With a twitter developer account, I plan on harvesting tweets in real time and storing them straight into a database of collections in MongoDB. After that I will begin with prepping, cleaning, exploring the data for sentiment analysis and topic modeling.

1.3 The Solution?

I honestly do not think a solution can be discovered with only running a sentiment analysis or a topic modeler for this. As described in 1.1 The Problem, a solution for the problem would require a lot of effort into this issue to produce a viable solution that does not take away the benefits from social media, search engines, and ad targeting algorithms.

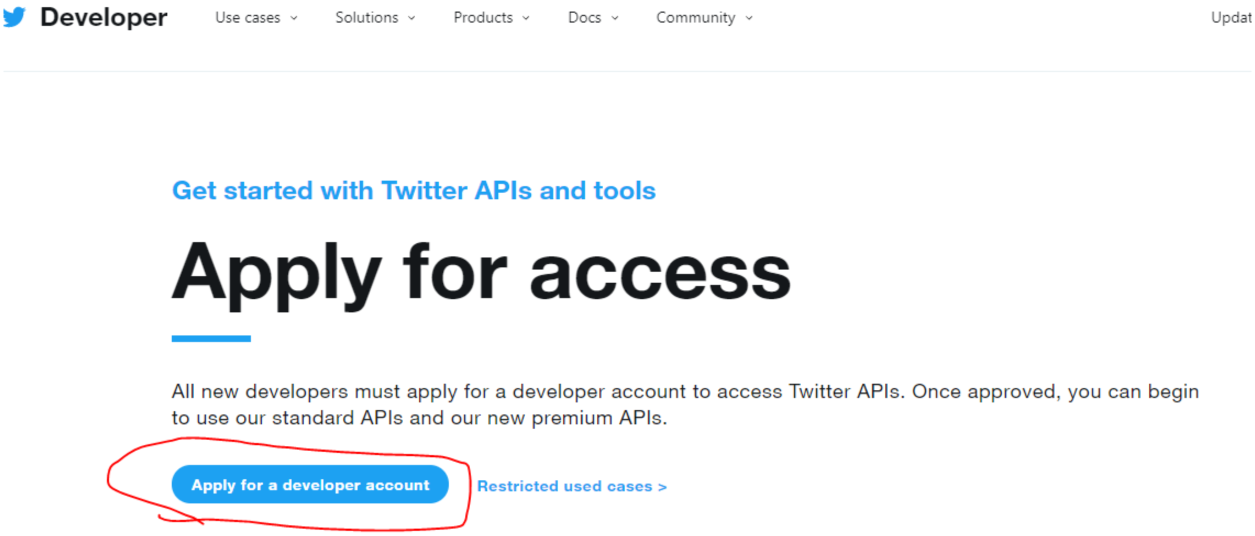


(Image by Clker-Free-Vector-Images from pixabay, 2012)

2. Mining the Data

2.1 Twitter Developer Account

To collect real-time tweets from twitter during this election a developer account is needed. The starting point for an application is at this URL application for twitter developer account ("Apply for access – Twitter Developers," 2020). Once on the page click the "Apply for a Developer Account" button.



Get started with Twitter APIs and tools

Apply for access

All new developers must apply for a developer account to access Twitter APIs. Once approved, you can begin to use our standard APIs and our new premium APIs.

(Image from the developer account application page)

The next page will prompt the user for login information, but if they do not have any, they can create a username to login in with to proceed to the next step of the application process. Once a user has completed the application for a developer account, they will then need to grab their consumer key, consumer secret key, access token, and access secret token.

2.2 Streaming the Tweets

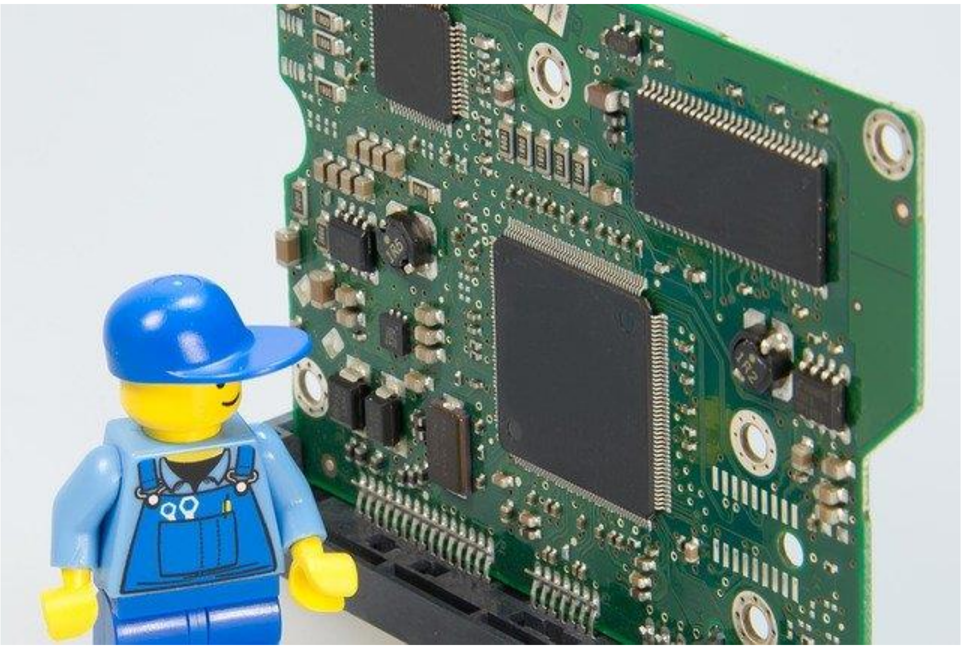
To stream the tweets in real time the provided credentials are needed to make a connection with Twitter's API. Python's third-party library Tweepy can be used to establish this connection and set up an easy-to-use stream listener to listen/collect the tweets based off the established filter provided in the Python script. However, even though these tweets can be stored as comma separated values it is best to store them in a database.

2.3 Streaming to MongoDB

Using MongoDB in conjunction with Python's third-party library PyMongo for easy local-host database connection set up will allow a user to utilize the stream listener created with Tweepy for storing real time tweets into a collection within a database inside MongoDB.

2.4 Streaming the Election

There were three topics at various times throughout the election I targeted for the streaming of tweets, "Election," "Trump," and "Biden." The times I chose were in the morning of election day, the afternoon, the evening, and poll closing at mountain/pacific time. The final stream collection was to be after the announcement of the presidential elect for all three topics. I did not account for the results for the presidential elect to be announced four days later, but I looked at the positive of this which was going to make the research project even more interesting and will capture different sentiments/topics over the four-day span. At the end of collection, I had 360,000 tweets stored into the tweets databased within MongoDB.



(Image by blickpixel from pixabay, 2014)

3. Processing the Data

3.1 Exporting the Data

The next step in the cycle would be to clean/process the data for modeling, but this cannot be done inside of MongoDB and must be done in Python. So, with Python I was able to use PyMongo to establish a connection with the local-host database collection and export the data into excel spreadsheets. This needs to be done since the code used for the sentiment analysis and topic modeler uses Python's library Panadas to create data frames for data manipulation and modeling. After exporting to excel spreadsheets I loaded the data files onto google drive to download on an Ubuntu Linux OS for faster code execution performance.

3.2 Downloading the Data

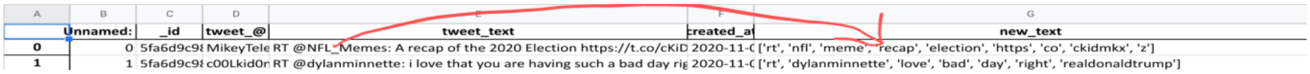
Once on the Ubuntu OS the data would be downloaded from google drive onto the OS platform. I created a subfolder within the datasets folder of my research project directory called "tweets." From here is where I will preprocess the data by cleaning/processing the text/tweets with text normalization code that I got from a book called "Text Analytics with Python," by Dipanjan Sarkar.

3.3 Prepping the Data

The preprocessing script starts out with defining a bunch of functions and objects to proceed to the next step of cleaning/processing the text/tweets. The first set of objects are instantiating stop words from the python library Natural Language Toolkit which are common words in the language that are overused and do not provide any real value for the natural language processing life cycle. After that creating a contractions map dictionary is needed to take contractions in the text and separating the word into the two that make up the contraction. Something like the contractions map dictionary is set up but for slang terms and then onto the functions for cleaning the text.

The first function for cleaning the text is stripping the hypertext markup language (html) from the tweets if there is any. Next would be to remove any characters that have accents above them except for apostrophes. After that would be to expand the contractions. Then lemmatizing the text which returns a token of the text to return only the base or root word of the text. Removing the stop words with the nltk stop words object would be next. Next would be to find the correct spelling of the word tokens. After would be to remove specialized characters in the text. Finally, tokenize the text for vectorization in the next part of the step for processing the data. Tokenization of the text is so that instead of giving the machine the entire sentence of text to analyze, taking tokens/words out of the string of text/sentence breaks it down for vectorization. In tweets a token is a word in a sentence where a sentence is a token in a paragraph ("NLP | How tokenizing text, sentence, words work - GeeksforGeeks," 2019).

After the text preprocessing/cleaning is complete a new feature is engineered and ready to be used for clustering:



A	B	C	D	E	F	G
	Unnamed:	_id	tweet_@	tweet_text	created_at	new_text
0	0	5fa6d9c9ŧ	MikeyTele	RT @NFL_Memes: A recap of the 2020 Election https://t.co/cKiD	2020-11-(	['rt', 'nfl', 'meme', 'recap', 'election', 'https', 'co', 'ckidmkx', 'z']
1	1	5fa6d9c9ŧ	c00Lkid0r	RT @dylanminnette: i love that you are having such a bad day rig	2020-11-(	['rt', 'dylanminnette', 'love', 'bad', 'day', 'right', 'realdonaldtrump']

(Image from election_results_clustered.xlsx)

(Image by Peggy_Marco from pixabay, 2016)

4. Feature Engineering / Exploring the Data

4.1 Vectorizing

Since the vectorization of the text is for the clustering, I stored the preprocessed tweet data in a separate folder from the original tweet data to use for the vectorization/clustering. Vectorization is also known to be feature engineering and is a tool used for creating features to be used when clustering the data. It is the process of taking meaningful feature or attributes from raw textual data and feed it into a statistical or machine learning algorithm (Dipanjan Sarkar, 2019). For the feature matrix creation, the textbook “Text Analytics with Python,” recommends using TF-IDF which stands for term frequency-inverse document frequency which gives a certain weight for each term used in the text when creating and using it for the feature matrix. TF-IDF is the combination of two metrics and can be represented like this equation tfidf = tf * idf. Term frequency is taken from a different text feature extraction concept known as a bag of words model and in which the frequency of a term in a document is established and can be represented as such equation:

$$tf(w,D) = f_{w_D}$$

“Where f_{wD} denoted frequency for word w in document D, which becomes the term frequency (tf)," (Dipanjan Sarkar, 2019). The inverse document frequency is then computed by taking the number of documents and dividing it by the document frequency of each term followed by the application of logarithmic scaling to the computed result. The mathematical equation of idf is represented like this:

$$idf(w,D) = 1 + \log \frac{N}{1 + df(w)}$$

“Where idf (w, D) represents the idf for the term/word w in document D, N represents

the total number of documents in our corpus, and df (t) represents the number of

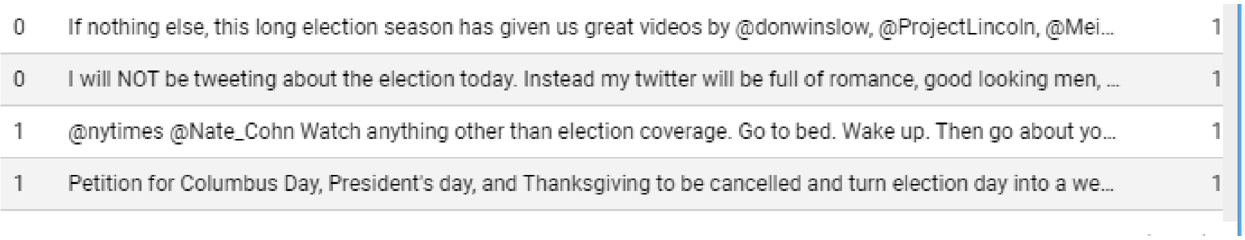
documents in which the term w is present,” (Dipanjan Sarkar, 2019). Multiplying both computations together and then normalizing tf-idf matrix of computations by dividing it by the Euclidian L2 norm.

4.2 K-Means Clustering

The feature extraction used in tf-idf vectorization was used to extract features for clustering. Clustering is an unsupervised form of machine learning where there is not any categorical data for the raw data to be associated with. The way K-Means works is by placing centroids randomly on in a dimension where the datapoints of the data will be measured based on its distance for similarity. The algorithm iterates over and over trying to move these centroids around the k clusters chosen to place themselves in the center, and when that happens the algorithm stops (Anas Al-Masri, 2019).

The clustering portion of this research project was not really to be used for the sentiment analysis or the topic modeling, but rather to be used to categorize the tweets with k-means clustering and recognize text similarity. It does not really fall into this research project for presenting other than the fact it will be shared for show and further exploration of the data. I chose to use K-Means clustering because I only wanted to work with two clusters since K-Means clustering is directed towards finding spherical shape based on similar distance with the specified number of clusters.

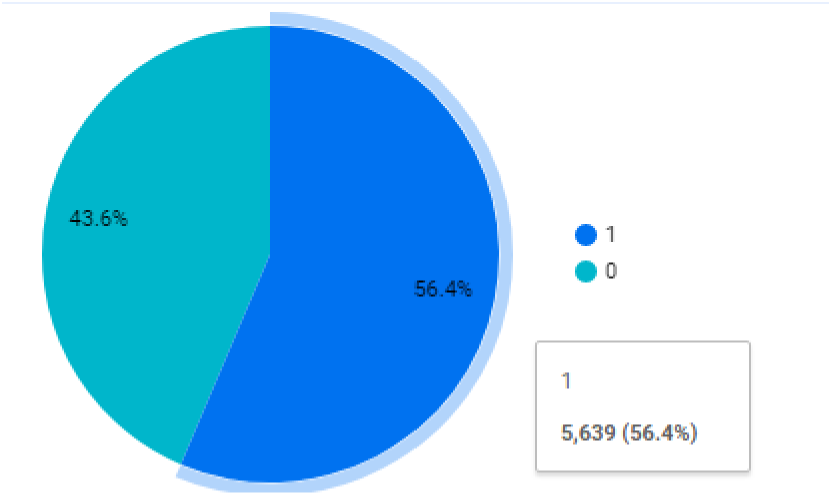
Doing a table for the dataset “election_am_clustered.xlsx and cluster fitted with its appropriate tweets:



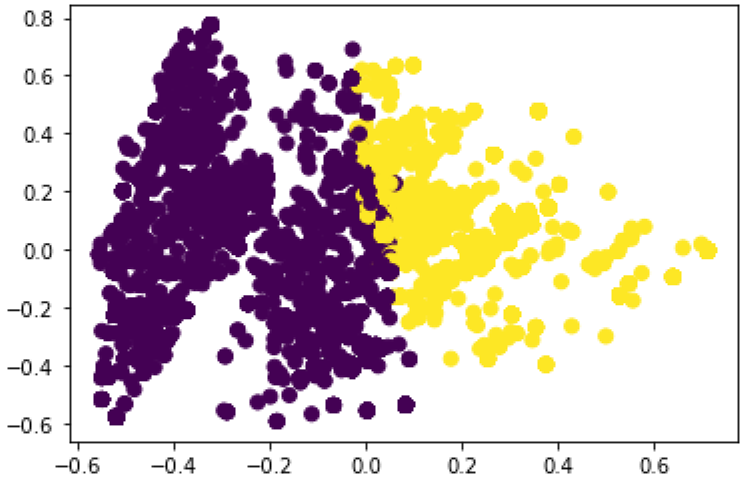
0	If nothing else, this long election season has given us great videos by @donwinslow, @ProjectLincoln, @Mei...	1
0	I will NOT be tweeting about the election today. Instead my twitter will be full of romance, good looking men, ...	1
1	@nytimes @Nate_Cohn Watch anything other than election coverage. Go to bed. Wake up. Then go about yo...	1
1	Petition for Columbus Day, President's day, and Thanksgiving to be cancelled and turn election day into a we...	1

(Four tweets from election_am_clustered.xlsx)

One can see that the clustering does not really work as a topic model. However, this is more for the exploration and visualization of how k-means would categorize the tweets. I do not think two clusters is the best choice for such a remarkably diverse and polarized election with so many mixed feelings the more optimal number of clusters would have been higher, but I went with the rule of thumb on this one. Cluster 0 is the highest cluster for the dataset with 6,5,639 documents out of 10,000 being selected for cluster 0 with K-Means:

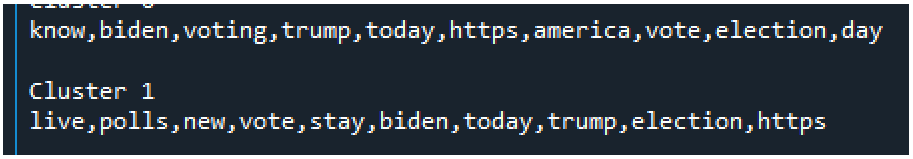


(Pie Chart from election_am_clustered.xlsx)



(Scatter plot of the categorized tweets from election_am_clustered.xlsx)

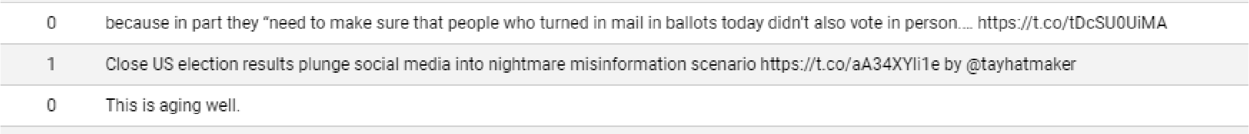
using tf-idf a python script can be made to snag the feature names of the terms of the clusters and these are the terms:



```
Cluster 0
know,biden,voting,trump,today,https,america,vote,election,day

Cluster 1
live,polls,new,vote,stay,biden,today,trump,election,https
```

(Image of tfidf being used to get the feature names for the clusters in election_am_clustered.xlsx)

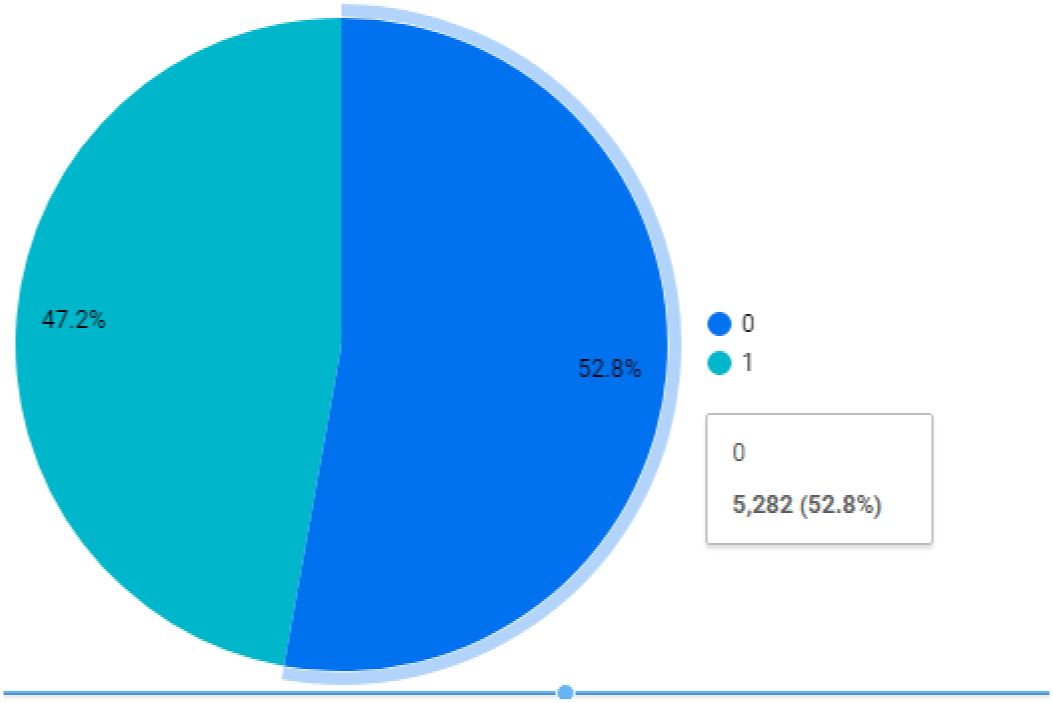
So even with this I am still not convinced that clustering the textual data will reveal any cateforical topics, but it does create some insight and understanding that the data is extremely diversified.



0	because in part they "need to make sure that people who turned in mail in ballots today didn't also vote in person.... https://t.co/tDcSU0UiMA
1	Close US election results plunge social media into nightmare misinformation scenario https://t.co/aA34XYli1e by @tayhatmaker
0	This is aging well.

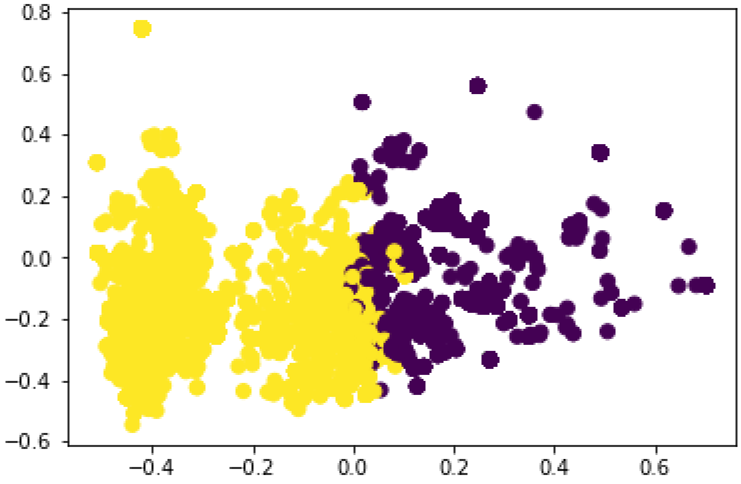
(Table of a few tweets from election_aft_2d_clustered.xlsx)

This is what a few tweets and their respective clusters looked like from the second day in the afternoon.



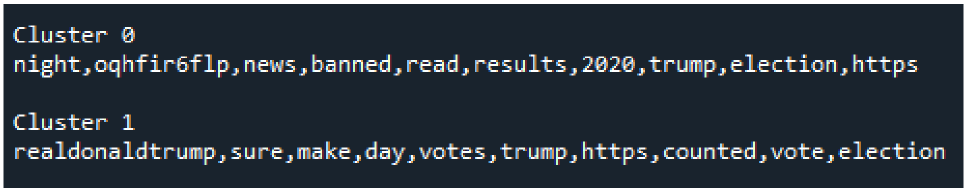
(Pie chart of the number of tweets per cluster for election_aft_2d_clustered.xlsx)

It is interesting to see though that both days had cluster 0 outmatching cluster 1.



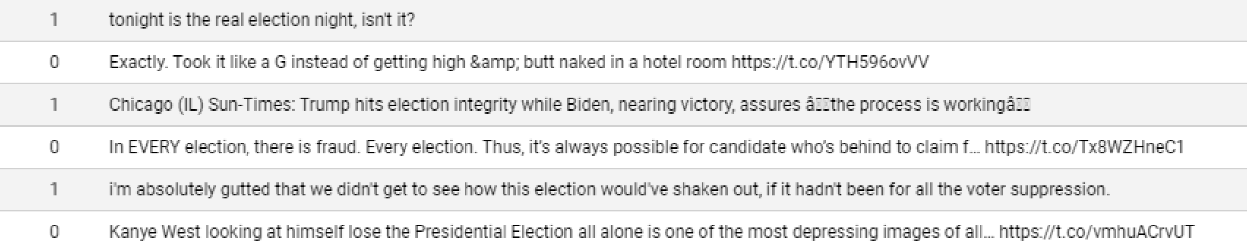
(Scatter plot of the categorized tweets from election_aft_2d_clustered.xlsx)

The scatter plot looks like the last one only the categorized clusters have traded places.



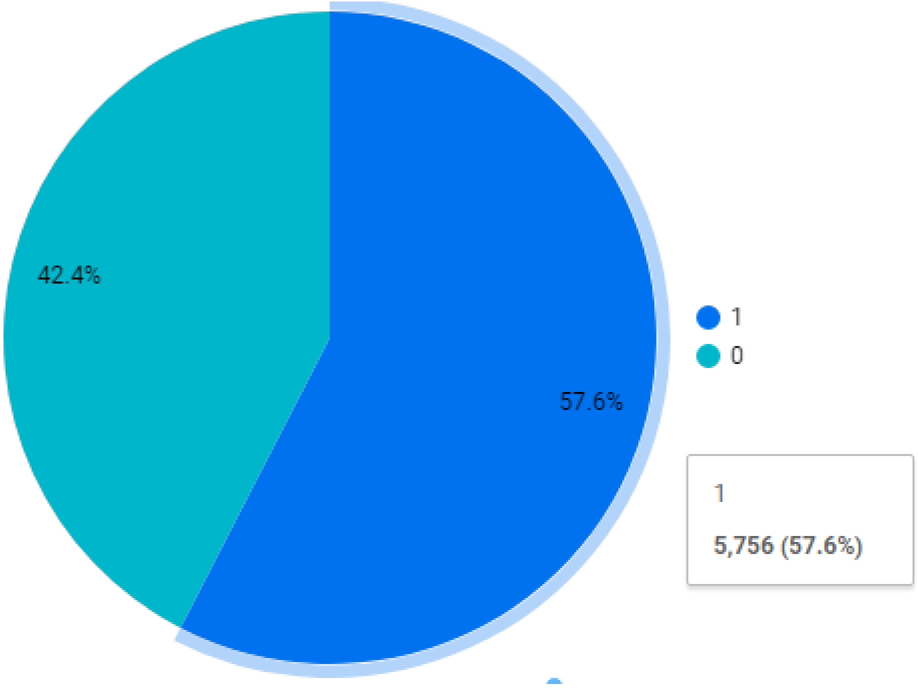
```
Cluster 0
night,oqhfir6flp,news,banned,read,results,2020,trump,election,https

Cluster 1
realdonaldtrump,sure,make,day,votes,trump,https,counted,vote,election
```

Once again, this feature name extraction and this exploration does not really provide me with anything new other than a lot of people have very mixed opinions on the election and even using the rule of thumb of two clusters you can see even with a polarized election trump is mentioned in both clusters.

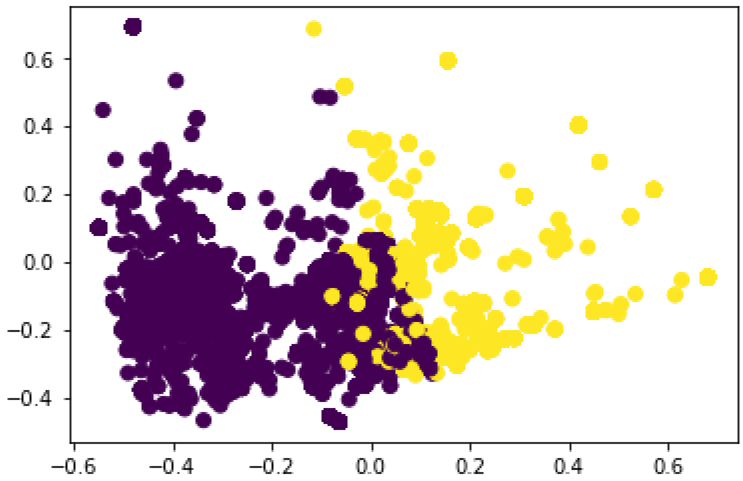


1	tonight is the real election night, isn't it?
0	Exactly. Took it like a G instead of getting high &amp; butt naked in a hotel room https://t.co/YTH596ovVV
1	Chicago (IL) Sun-Times: Trump hits election integrity while Biden, nearing victory, assures â□□the process is workingâ□□
0	In EVERY election, there is fraud. Every election. Thus, it's always possible for candidate who's behind to claim f… https://t.co/Tx8WZHneC1
1	i'm absolutely gutted that we didn't get to see how this election would've shaken out, if it hadn't been for all the voter suppression.
0	Kanye West looking at himself lose the Presidential Election all alone is one of the most depressing images of all… https://t.co/vmhuACrvUT

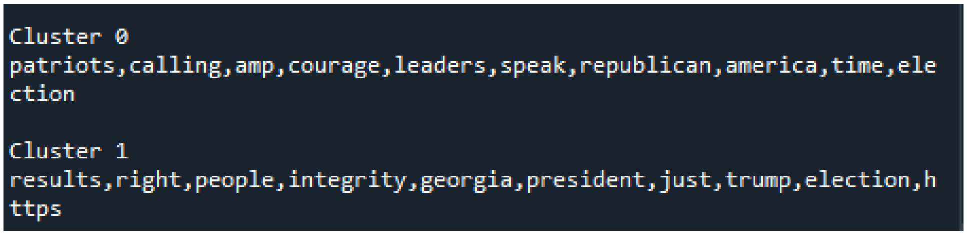
(Table of some tweets from election_even_3d_clustered. xslx)



(Pie chart of number of tweets per cluster in election_even_3d_clustered.xlsx)



(Scatter plot of the categorized tweets from election_even_3d_clustered.xlsx)



```
Cluster 0
patriots,calling,amp,courage,leaders,speak,republican,america,time,ele
ction

Cluster 1
results,right,people,integrity,georgia,president,just,trump,election,h
ttps
```

(Image of tfidf extracted features for the clusters in election_even_3d.xlsx)

I know that using more clusters would have had better results or using a different type of clustering method, but there were 360,000 tweets to go through spanned out on 36 datasets and with my local machine's performance I chose the quickest route. However, this does give probable cause and justification for myself to use as sentiment analysis and topic modeling to answer my question.

5. Modeling the Sentiment Analysis

5.1 Introduction to a Sentiment Analysis

A sentiment analysis is a tool of textual analytics to gain insight from the sentiment of a user's text. It is considered the more popular tool of natural language processing/text analytics (Dipanjan Sarkar, 2019) probably since it is easy after normalizing the text. With everything out there that is from a comment, email, tweet, post, review, etc. there is a certain level of value behind someone's point of view and this is where a sentiment analysis comes into play. How the text of a document is scored is based off the polarity score reflecting the words in the text. The type of sentiment analysis I will be doing is an unsupervised sentiment analysis. The way an unsupervised sentiment analysis works is by setting up the python scripts to give a sentimental score to the processed text within the corpus of text documents given an established lexicon like the two I will be using AFINN and VADER.

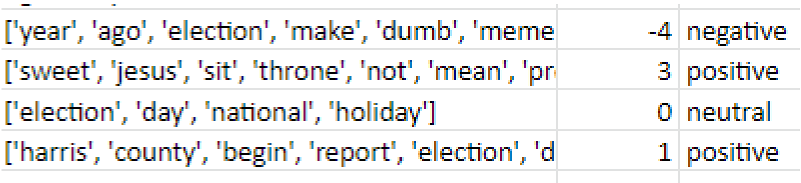
5.2 AFINN and VADER Lexicons

"The AFINN lexicon is perhaps one of the simplest and most popular lexicons and can be used extensively for sentiment analysis. Developed and curated by Finn Årup Nielsen," (Dipanjan Sarkar, 2019). As of now AFINN-en-165.txt is the most recent/update of Finn's lexicon which has over 3,300 words attached with polarity scores to each of them. The VADER lexicon which stands for Valence Aware Dictionary and sEntiment Reasoner which was developed by C.J. Hutto (Dipanjan Sarkar, 2019) is the one I will be basing most of my analysis off because it has 7520 lexical features for sentiment polarity scoring. I chose to do both as a side-by-side comparison for just to see if the scoring would change.

5.3 AFINN Sentiment

For the AFINN sentiment I chose to add the neutral threshold for the predicted sentiment. If the threshold of the predicted sentiment is equal to zero then the predicted sentiment is neutral, but if it is anything greater than zero it is positive. The same goes if the threshold is lesser than zero it will be a negative predicted sentiment.

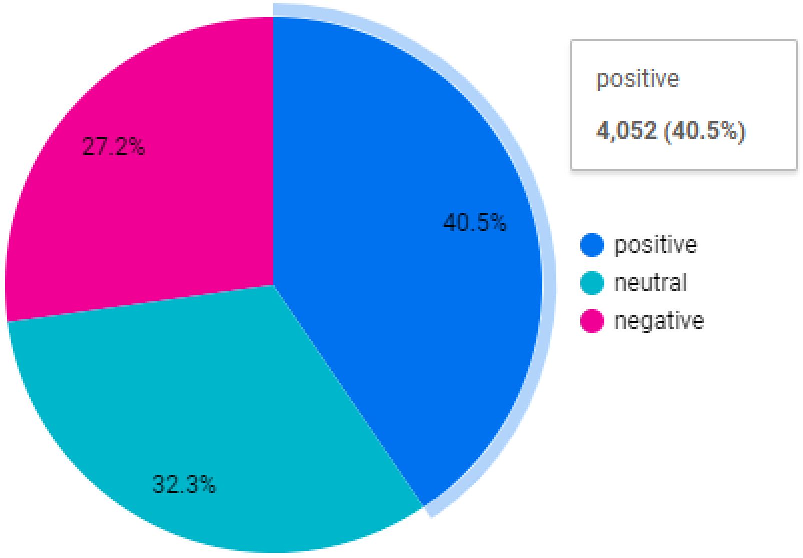
Before executing the python scripts, the AFINN python module must be installed and imported first. How the python script for it works is the clustered files are loaded in by a for loop turning each of the preprocessed text in the "new_text" column into a NumPy array. Then an object will be made tracking the predicted sentiment of the processed text/tweets in a for loop. Then a for loop will capture the threshold of the predicted sentimental polarity for the tweet and append the sentimental value to a list where it will then be returned to the data frame. After an iteration through the first file of the original loop is complete a new file is saved with a new dataset containing the appended AFINN predicted sentiment and the sentimental polarity to each tweet.



['year', 'ago', 'election', 'make', 'dumb', 'meme	-4	negative
['sweet', 'jesus', 'sit', 'throne', 'not', 'mean', 'pr	3	positive
['election', 'day', 'national', 'holiday']	0	neutral
['harris', 'county', 'begin', 'report', 'election', 'd	1	positive

(A snippet of the results of the AFINN sentiment analysis from the election_pollclose_mtn_afinn.xlsx dataset)

Above is just a sample of the sentiment analysis for AFINN and is an excellent example at how effective a sentiment analysis is for understanding text but bear in mind though these sentiment analyses are not meant to understand someone's personality and therefore cannot take into context someone's dark humor or sarcasm.



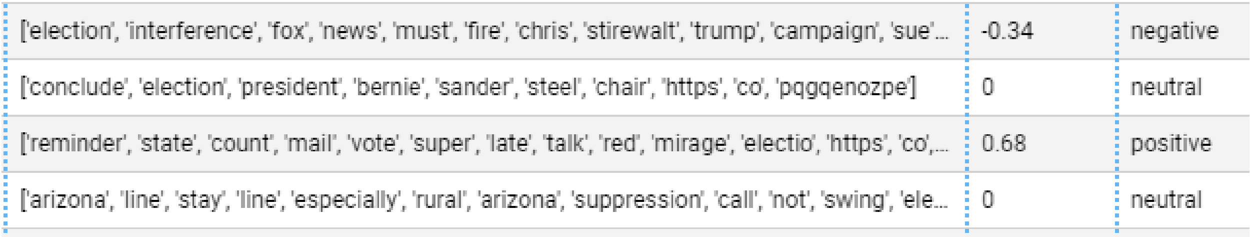
(A pie chart of the overall predicted sentiment of the AFINN analysis from the election_pollclose_mtn_afinn.xlsx dataset)

Even though it is an evenly distributed pie chart, it does bring comfort seeing that the positivity outweighs the neutral and negativity.

5.4 VADER Sentiment

For the Vader sentiment analysis, the library of the module Vader resides in NLTK's sentiment directory. Executing the python scrip to run the Vader sentiment analysis is fairly like the AFINN python script execution, but there needs to be some defined functions first. The first function is the analyze_sentiment_vader_lexicon() function which will be the main function to attach Vader polarity scores to the text/tweets in the corpus. Just like the AFINN analysis scripts, this function will have a similar threshold for the polarity declaring whether the predicted sentiment is positive, negative, or neutral. The next three functions analyze_sentiment_vader_neg(), analyze_sentiment_vader_pos(), and analyze_sentiment_vader_neu() will give a percentage to how much of the text/tweet is positive, neutral, or negative. These functions will be followed by analyze_sentiment_vader_fin() to establish a final polarity score for the text/tweet. All of these will be added to the dataset attached to the tweets.

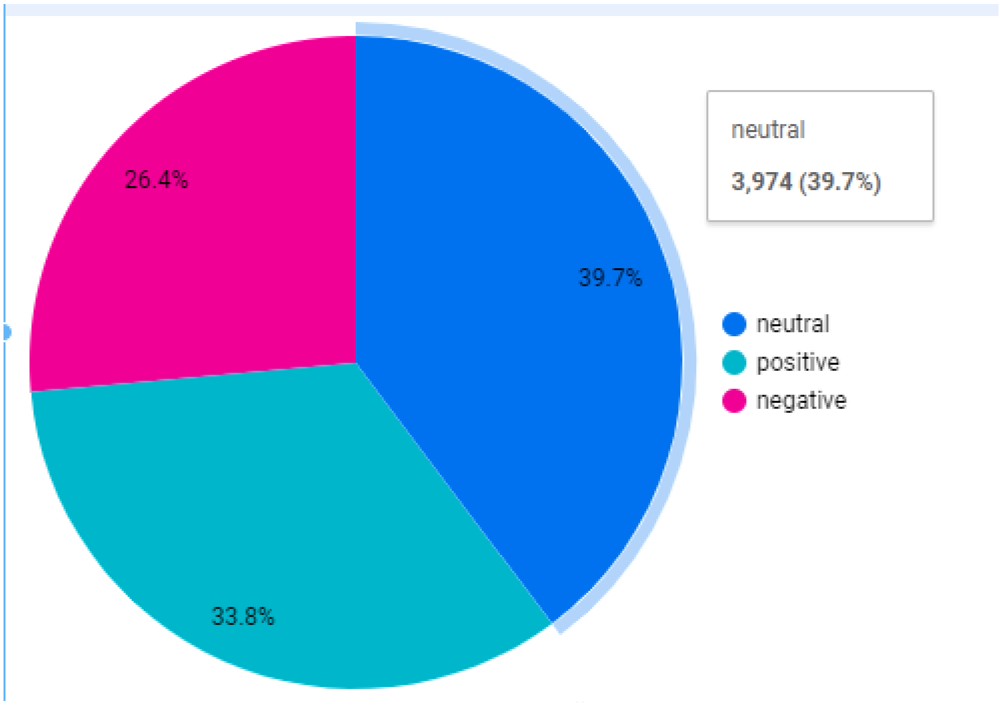
I added one more function that was not part of the recommended code from the textbook "Text Analytics with Python," which would analyze the three separate percentage values given a tweet/text's sentiment and declare the predicted sentiment based off the highest scoring value and not off the calculated polarity. I did this to see the difference between going off the highest scoring sentimental value (which often was neutral) vs the polarity score. However, I will be going off the polarity score for the actual analysis portion of this research project's lifecycle.



['election', 'interference', 'fox', 'news', 'must', 'fire', 'chris', 'stirewalt', 'trump', 'campaign', 'sue'...	-0.34	negative
['conclude', 'election', 'president', 'bernie', 'sander', 'steel', 'chair', 'https', 'co', 'pqgqenozpe']	0	neutral
['reminder', 'state', 'count', 'mail', 'vote', 'super', 'late', 'talk', 'red', 'mirage', 'electio', 'https', 'co',...	0.68	positive
['arizona', 'line', 'stay', 'line', 'especially', 'rural', 'arizona', 'suppression', 'call', 'not', 'swing', 'ele...	0	neutral

(A snippet of the results of the VADER sentiment analysis from the election_pollclose_west_vader.xlsx dataset)

The polarity scoring is done a little differently for the Vader sentiment analysis as far as the scale where AFINN only scores with integers scale and Vader scores on a float scale. Once again, I will not be comparing AFINN to Vader in my analysis for I do not want to choose one analysis over the other based off a bias conscious and there for will only stick to the Vader analysis.



(A pie chart of the overall predicted sentiment of the VADER analysis from the election_pollclose_west_vader.xlsx dataset)

6. Topic Modeling

6.1 Introduction to Topic Modeling

Topic modeling is basically what the title says which is modeling topics found in the corpus of documents. Topic modeling uses mathematical and statistics to snag multiple topics and themes from the text (Dipanjan Sarkar, 2019). I chose to use topic modeling in conjunction with a sentiment analysis to paint a better picture of the overall opinion and feelings of those tweeting during these times of streaming the tweets into the database. Using the sentiment and looking at every tweet individually would take forever and that is why using a topic model will help articulate the same thing but on a broader scale.

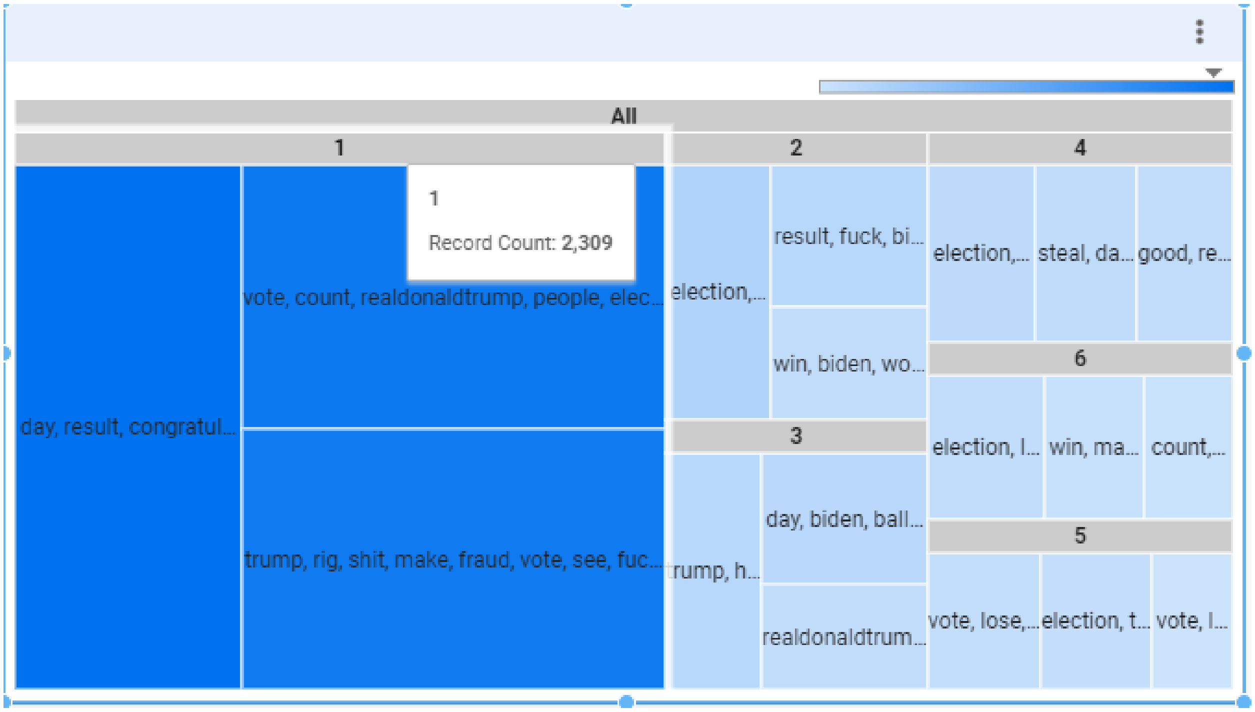
6.2 Topic Modeling with LDA genism and Mallet

The way that topic modeling works is by engineering the features necessary to set up the topics. How the features are obtained are through gathering bigrams or trigrams. Bigrams or trigrams are a type of ngram which can be termed as a phrase from a string of text. A bigram is two terms linked together as a phrase, and a trigram is three. The process of calculating and ngram is taking one term and moving it forward to the next and then taking that next term and moving that forward to the next term (Kavita Ganesan, 2018).

Once a bigram model is established with the extracted phrases the next part is to create a dictionary of those bigrams to be used for a bag of words model. The bag of words model is used later for using the lda genism mallet python library to iterate through the selected ngrams and calculate a topic based on the entire corpus of documents. The initial iteration using LDA genism with Mallet is a way to see the performance and the base topics of a corpus before going through and finalizing the model.

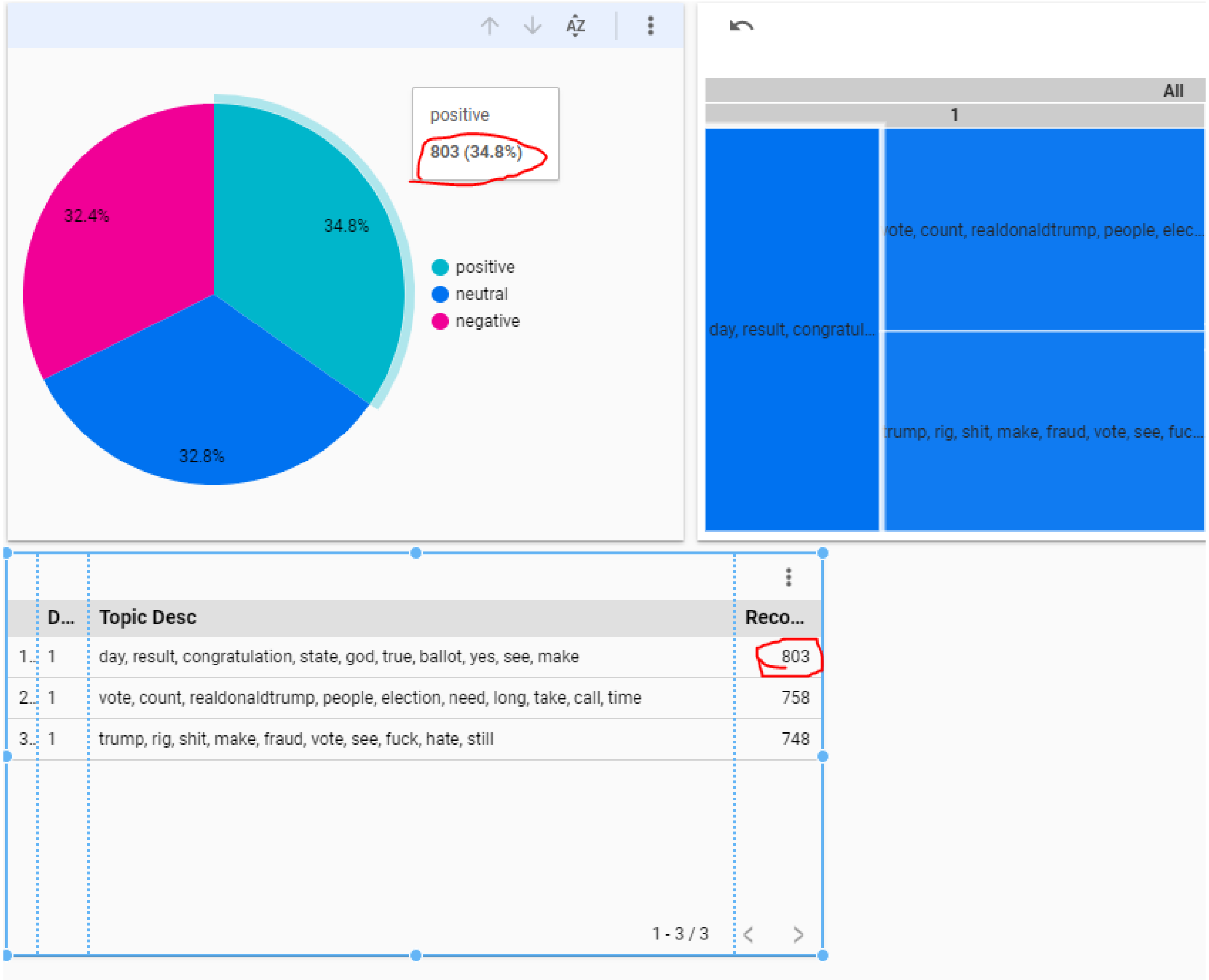
From the textbook "Text Analytics with Python," I made a function that would run a test and iterate the corpus through 30 topics. The function would then give a coherence score as to what the best number of topics to go with for this research project should be. However, I decided against using the number of topics recommended by the coherence score because the recommendation 20 or more topics and that would be too much to cover for this research project. I kept it simple though it probably isn't wise and used only 6 topics per corpus with 3 subtopics to reflect the sentiments. I added the topics with their respective sentiments as because I wanted to see how a topic and the sentiment that correlated with it would change from positive to negative.

6.3 Preview of the Topic Model



(A snippet of a tree map showing the 6 topics with their sentiment subtopics for election_day_two_analysis.xlsx)

When going through and looking at the topic model through a tree map and their descriptions alone it is hard to tell why those words were chosen for that topic and the subtopics relative to the sentiments with it. However, when combining the tree map with multiple visualizations it becomes clear which sentiments are related to which subtopic of the selected topic.


positive
803 (34.8%)
32.4%
34.8%
32.8%
● positive
● neutral
● negative
All
1
day, result, congratul...
ote, count, realdonaldtrump, people, elec...
rump, rig, shit, make, fraud, vote, see, fuc...

	D...	Topic Desc	Reco...
1...	1	day, result, congratulation, state, god, true, ballot, yes, see, make	803
2...	1	vote, count, realdonaldtrump, people, election, need, long, take, call, time	758
3...	1	trump, rig, shit, make, fraud, vote, see, fuck, hate, still	748

1 - 3 / 3

(A combo of a tree map for the topic model, a table, and a pie chart to visualize topic 1's highest subtopic sentiment for election_day_two_analysis.xslx)

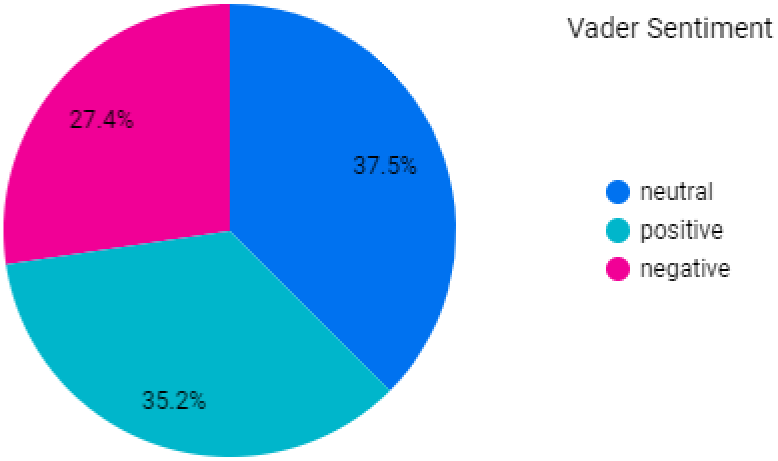
How I would interpret the sentiment analysis and model for this would be that for the tweet stream filter "election" combining all three stream times on day two, the most dominant topic one had a higher scoring sentiment of a subtopic for ['day', 'result', 'congratulations', 'state', 'god', 'true', 'ballot', 'yes', 'see', 'make']. I could see why this would score higher than the neutral subtopic or the negative subtopic because of the more positive use of words in this subtopic. Now onto the analysis.

7. Research Project Analysis

7.1 Introduction to the Analysis

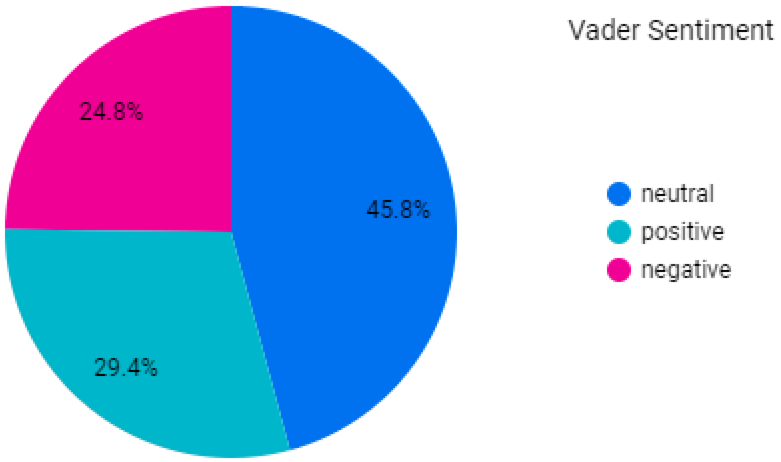
The analysis will look like the last image for the topic modeling, but I will incorporate and go over the highest scored sentiments with their topics and subtopic for all the "election" tweet filter streams over the three days until the announcement of the results. It is important to bear in mind that the streaming was done on a developer account meaning the algorithms of social media would not apply to the mining of data, and the lexicons or genism library used for the sentiment analysis and topic modeling hold no bias either. Therefore, this analysis should be viewed with an open mind to what the results may hold because the question that this research project is trying to answer is how politically polarized the United States is.

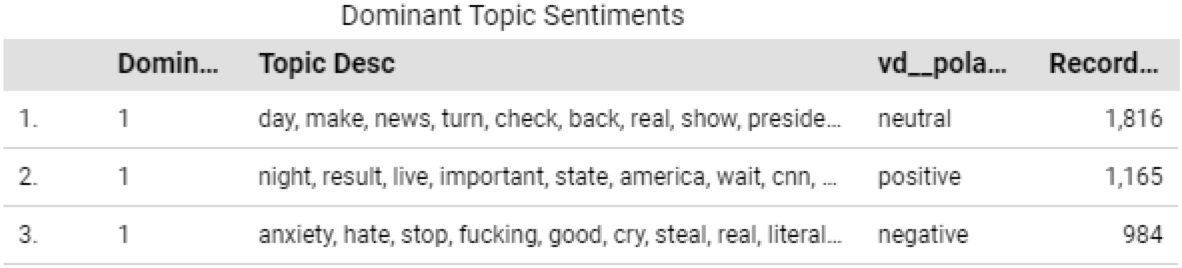
7.2 Election Analysis Day One



(A pie chart of the overall Vader Sentiment analysis for election_day_one_analysis.xlsx)

This right here is showing that all 9,092 tweets (not retweets) collected on day one for the stream filter “election” show an overall neutral sentiment. I do not necessarily want to cast my judgment onto the sentiment or the analyses about to be shown because I do not want to include my own bias. However though, judging by the score of the pie almost being completely symmetrical I would say that it is a good thing that there is a higher score for neutrality and positivity of the overall tweets for the election rather than having a higher negative score.



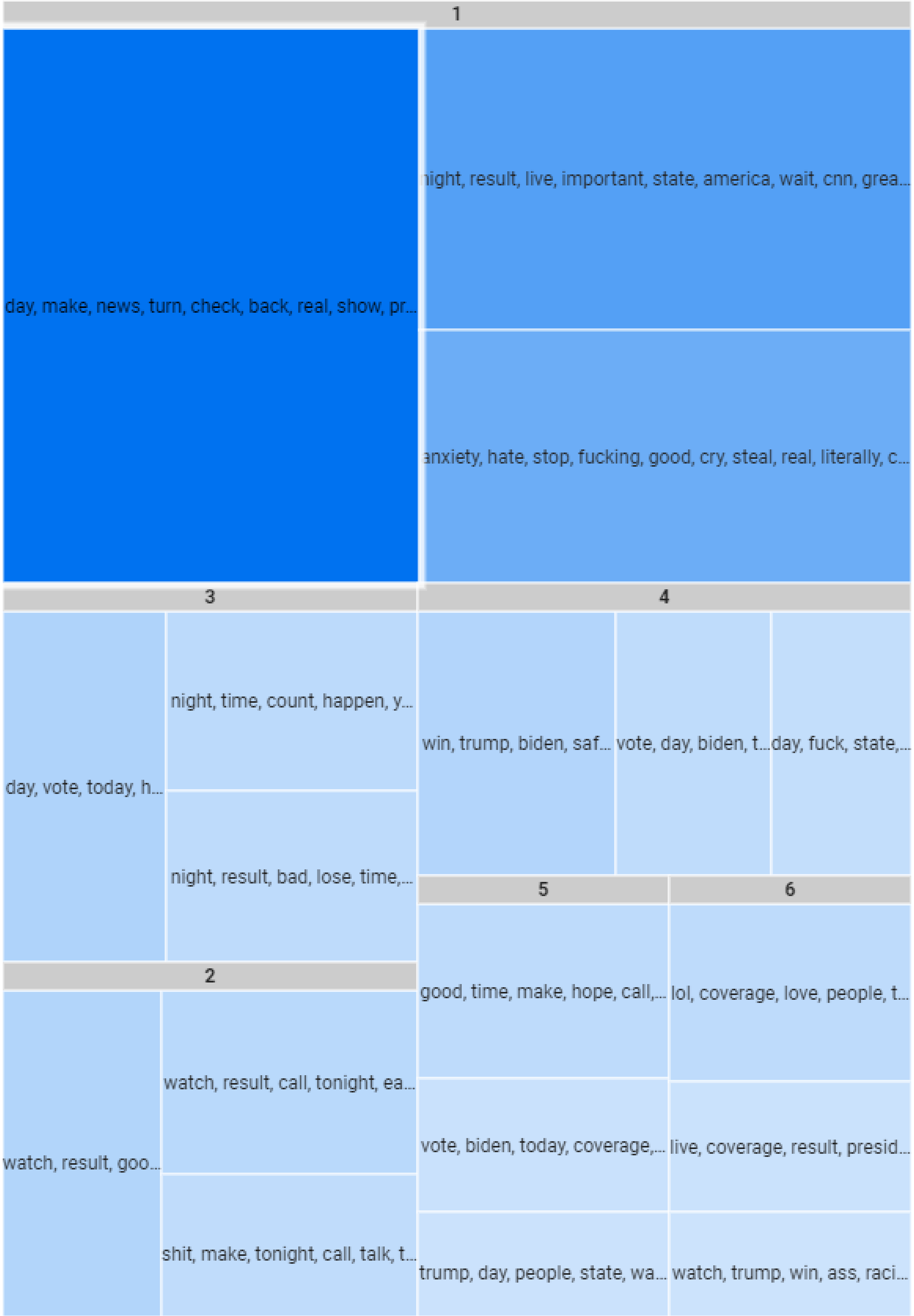

Dominant Topic Sentiments

	Domin...	Topic Desc	vd__pola...	Record...
1.	1	day, make, news, turn, check, back, real, show, preside...	neutral	1,816
2.	1	night, result, live, important, state, america, wait, cnn, ...	positive	1,165
3.	1	anxiety, hate, stop, fucking, good, cry, steal, real, literal...	negative	984

(A pie chart and table of topic 1 with the sentiments of the subtopics for election_day_one_analysis.xlsx)

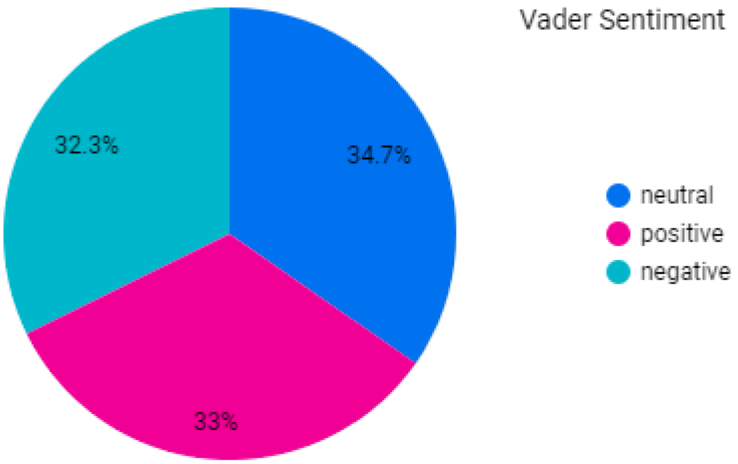
The highest-ranking topic for the first day had an exceedingly high neutral sentiment with 1,816 tweets contributing to that neutral sentiment and of that neutral subtopic the main ngrams for it are on the first row. With these ngrams I can see why there was a lot of neutrality for this first topic but am also noticing that even though joining the sentiment analysis and topic model together to try and gain more insight proves to be tougher understanding than I originally thought. When looking at the sentimental subtopics for topic one I want to say that the terms in row two are very neutral to me also besides the term “great.” However, I know that the algorithms are doing what they are supposed to do correctly because the terms for row three of the negative subtopic are a key indicator that the topics are being correlated with their respective sentiments correctly.

The results an analysis of day one will not be enough to let me know if the country is politically polarized as much as people are claiming. This is a visualization the 6 dominant topics with their sentimental subtopics



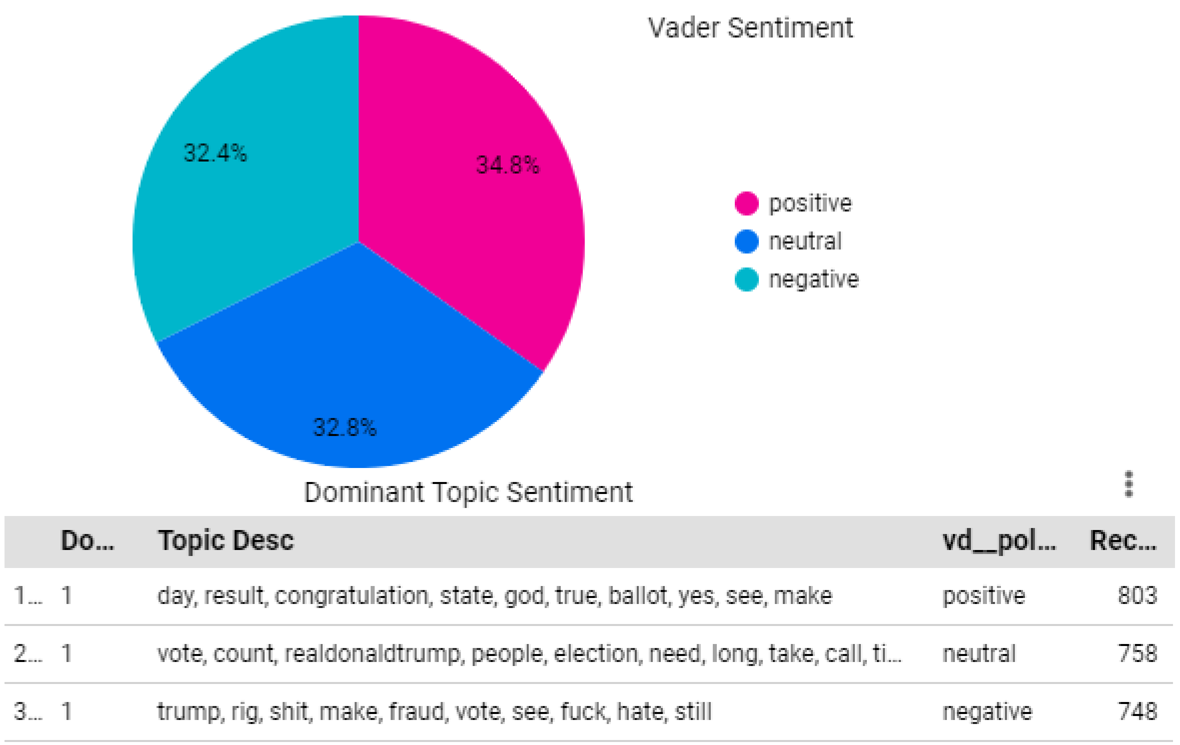
(A tree map of the 6 dominant topics and the three sentimental subtopics for them from election_day_one_analysis.xlsx)

7.3 Election Analysis Day Two



(A pie chart of the overall Vader Sentiment analysis for election_day_two_analysis.xlsx)

The next day of the election process, for 1,397 tweets the positive sentiment had yielded the highest score. The drop in tweets is because the rest of the 8,603 tweets out of the 10,000 collected at this time were retweets and that would distort the analysis. However, the one thing I see in both pie charts is the fact that they are symmetrical with their sentimental scores. It should be noted that these are aggregated counts of the Vader predicted sentiments based off the polarity score. The only question I have in mind is having a symmetrical sentiment of the overall tweets plausible for a nation being politically polarized?

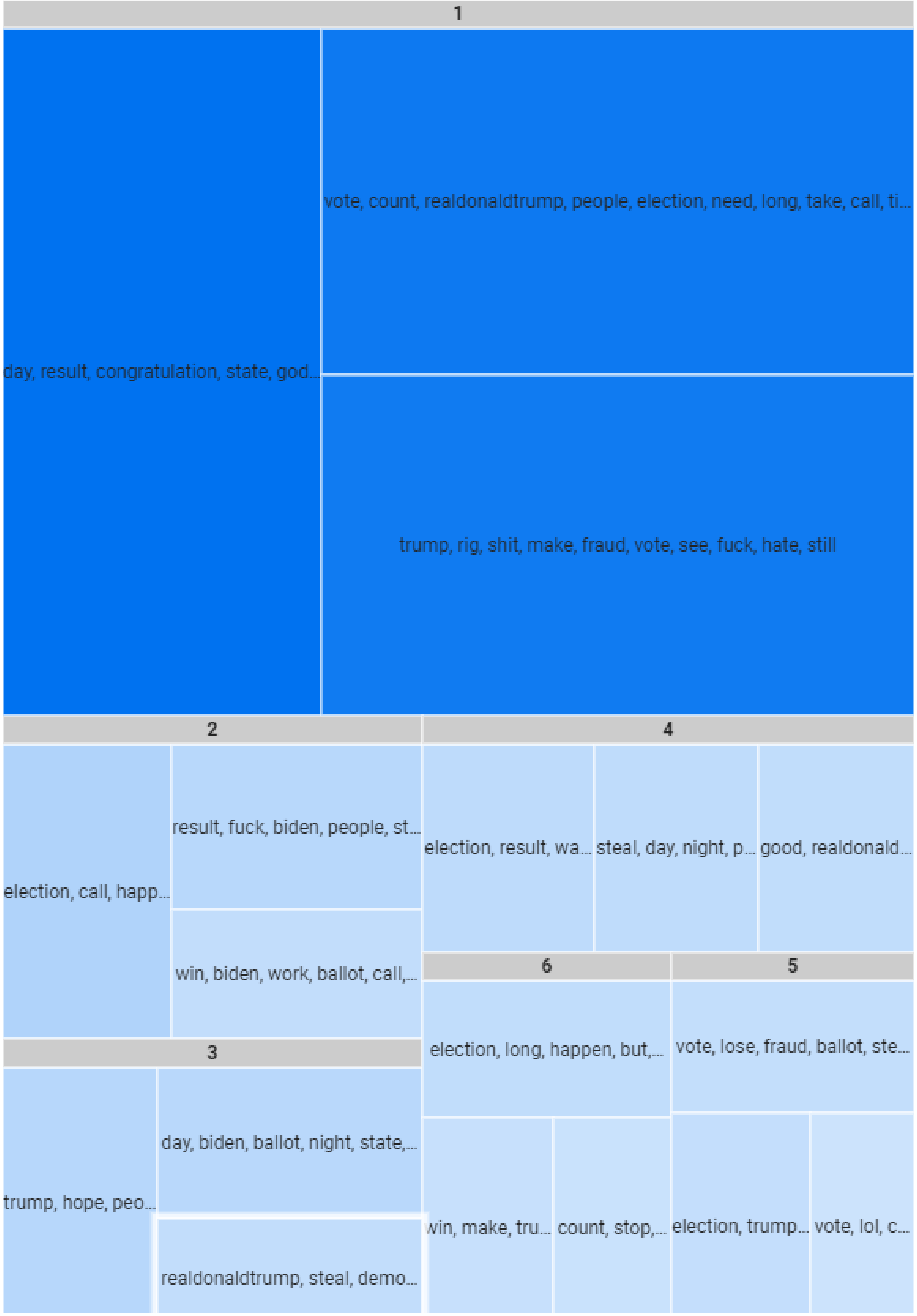


Dominant Topic Sentiment

	Do...	Topic Desc	vd_pol...	Rec...
1...	1	day, result, congratulation, state, god, true, ballot, yes, see, make	positive	803
2...	1	vote, count, realdonaldtrump, people, election, need, long, take, call, ti...	neutral	758
3...	1	trump, rig, shit, make, fraud, vote, see, fuck, hate, still	negative	748

(A pie chart and table of topic 1 with the sentiments of the subtopics for election_day_two_analysis.xlsx)

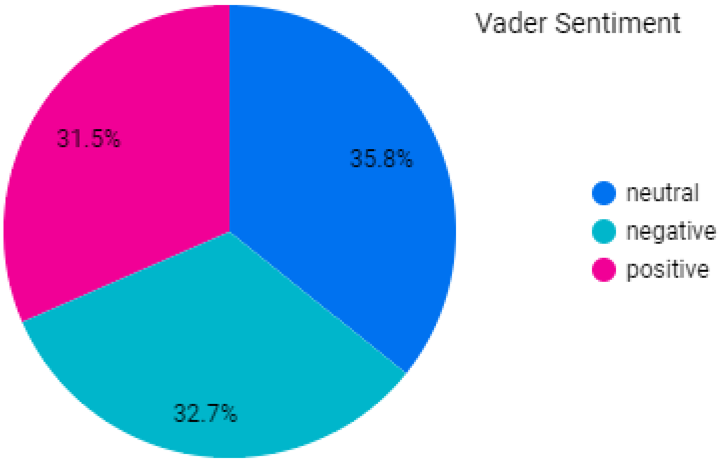
The two terms sticking out at me for the first row being "result" and "congratulations" throw me off since the announcement of the results were not made this day. I also find it odd that trump falls under the neutral subtopic sentiment because would have thought the mention of the president would be either positive or negative. However, the next subtopic sentiment leads me to believe that there was a lot of discussion about voter fraud.


1
day, result, congratulation, state, god...
vote, count, realdonaldtrump, people, election, need, long, take, call, ti...
trump, rig, shit, make, fraud, vote, see, fuck, hate, still
2
election, call, happ...
result, fuck, biden, people, st...
win, biden, work, ballot, call,...
4
election, result, wa...
steal, day, night, p...
good, realdonald...
6
election, long, happen, but,...
win, make, tru...
count, stop,...
5
vote, lose, fraud, ballot, ste...
election, trump...
vote, lol, c...
3
trump, hope, peo...
day, biden, ballot, night, state,...
realdonaldtrump, steal, demo...

(A tree map of the 6 dominant topics and the three sentimental subtopics for them from election_day_two_analysis.xlsx)

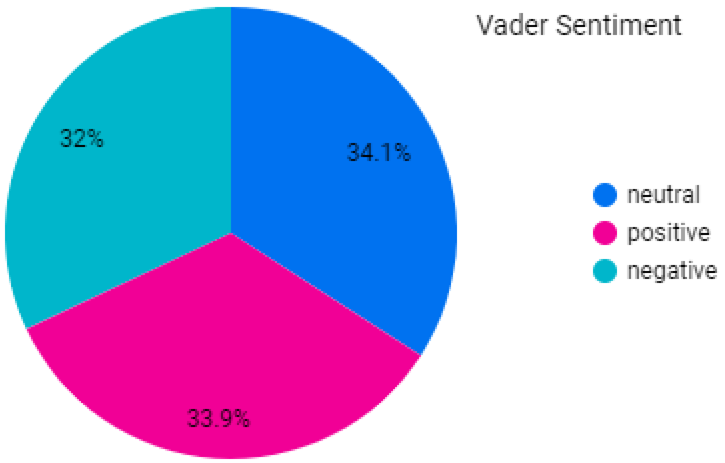
It still is tough to make out and decide what most of these topics/subtopics are categorized as because most of them are all talking about the same thing which may be because the streaming filter was "election" so the one word that can sum most all of these up is election.

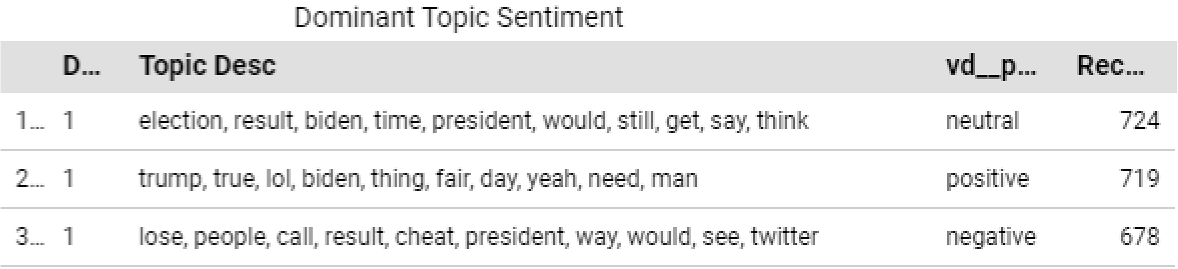
7.4 Election Analysis Day Three



(A pie chart of the overall Vader Sentiment analysis for election_day_three_analysis.xlsx)

Day three's highest-ranking sentiment for all 3,733 tweets was neutral. Although the pie chart is symmetrical like the others, I would say that having both neutral and positive outranking negative is a good thing. However, that does not mean that a neutral and positive sentiment outranking a negative sentiment score means its win because neutral and positive are not teammates in this analysis.



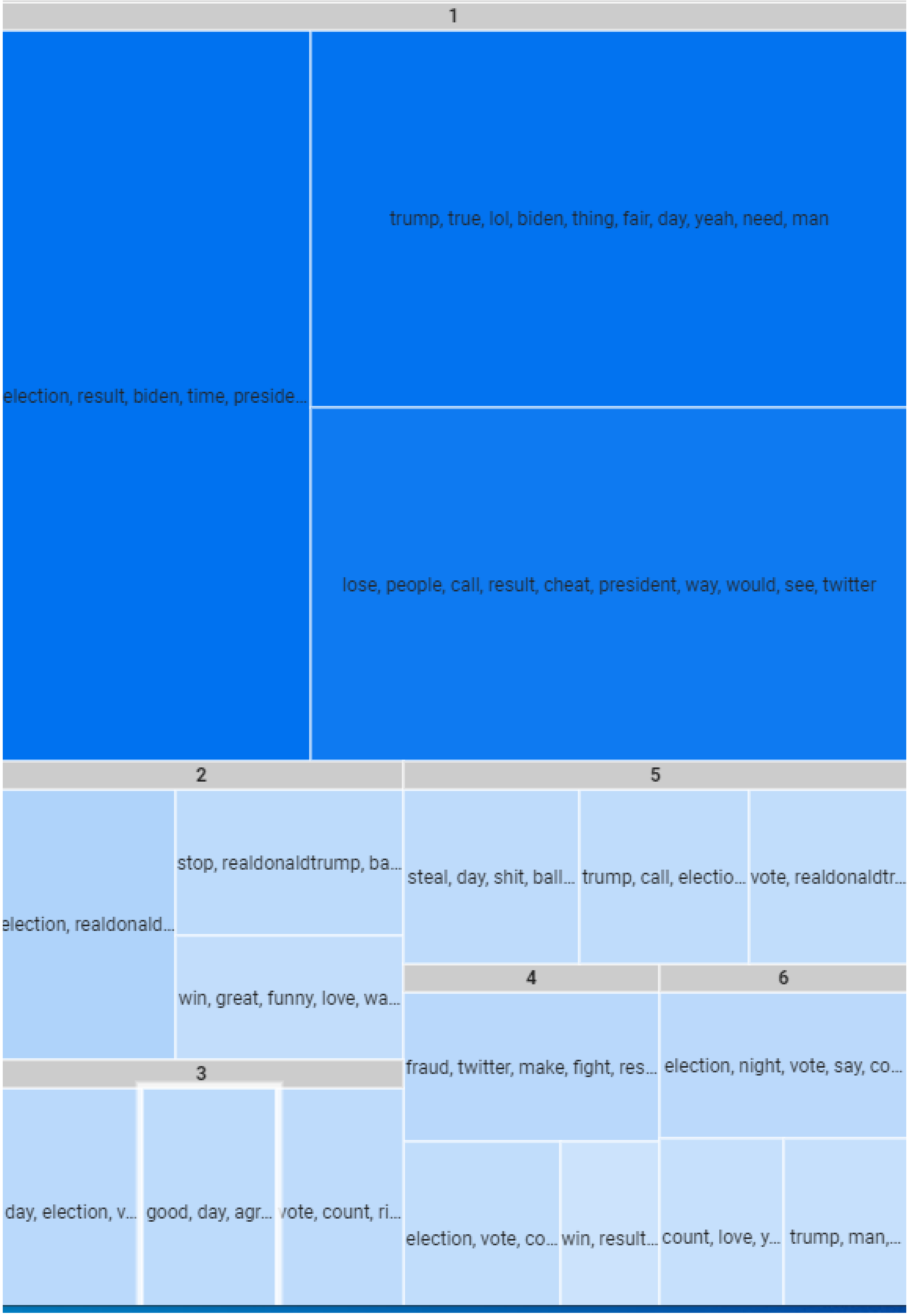


Dominant Topic Sentiment

	D...	Topic Desc	vd__p...	Rec...
1...	1	election, result, biden, time, president, would, still, get, say, think	neutral	724
2...	1	trump, true, lol, biden, thing, fair, day, yeah, need, man	positive	719
3...	1	lose, people, call, result, cheat, president, way, would, see, twitter	negative	678

(A pie chart and table of topic 1 with the sentiments of the subtopics for election_day_three_analysis.xlsx)

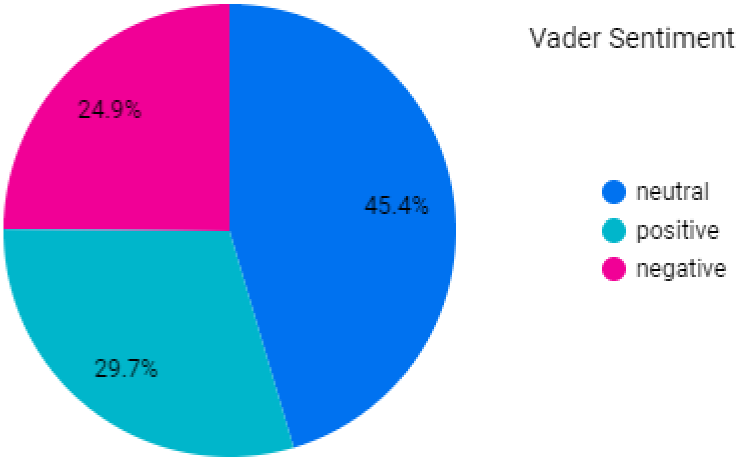
Just like day two’s topic one subtopic negative sentiment the correlated synonym to fraud is cheat, and that is one of the terms in the negative section. This time around Biden is shown for the first time in the positive subtopic and I am thinking this is because the results at the time (which is one of the terms) was in the lead so far. So, what I am thinking is the first topic is an absolute discussion about the results of the election real-time and the active discussion from people about it.


1
election, result, biden, time, preside...
trump, true, lol, biden, thing, fair, day, yeah, need, man
lose, people, call, result, cheat, president, way, would, see, twitter
2
election, realdonald...
stop, realdonaldtrump, ba...
win, great, funny, love, wa...
5
steal, day, shit, ball...
trump, call, electio...
vote, realdonaldtr...
4
fraud, twitter, make, fight, res...
election, vote, co...
win, result...
6
election, night, vote, say, co...
count, love, y...
trump, man,...
3
day, election, v...
good, day, agr...
vote, count, ri...

(A tree map of the 6 dominant topics and the three sentimental subtopics for them from election_day_three_analysis.xlsx)

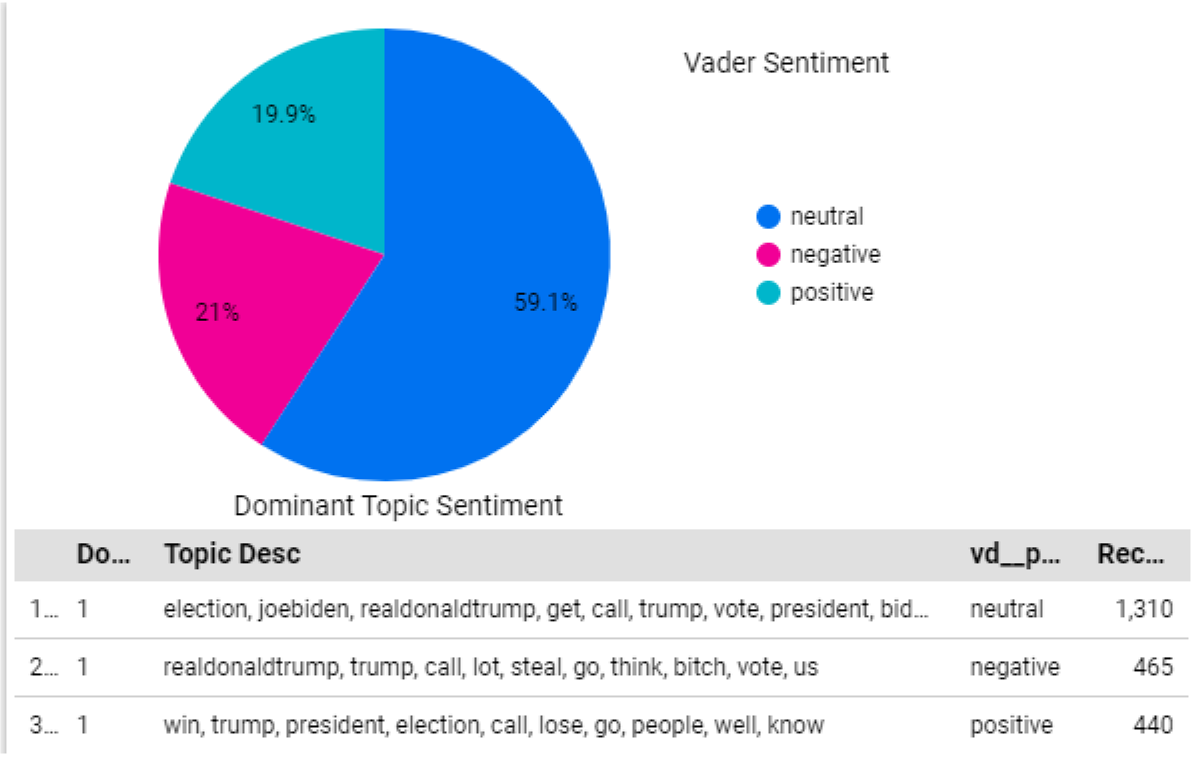
7.5 Election Analysis Elect Results

This analysis is from when after Biden was announced the presidential elect with 3,369 tweets.



(A pie chart of the overall Vader Sentiment analysis for election_elect_results_analysis.xlsx)

So, this is remarkably interesting. The pie chart symmetry completely shifted and now the overall ranking sentiment is neutral. I wonder what everyone was discussing on twitter for it to shift like that from almost symmetrical pie charts.

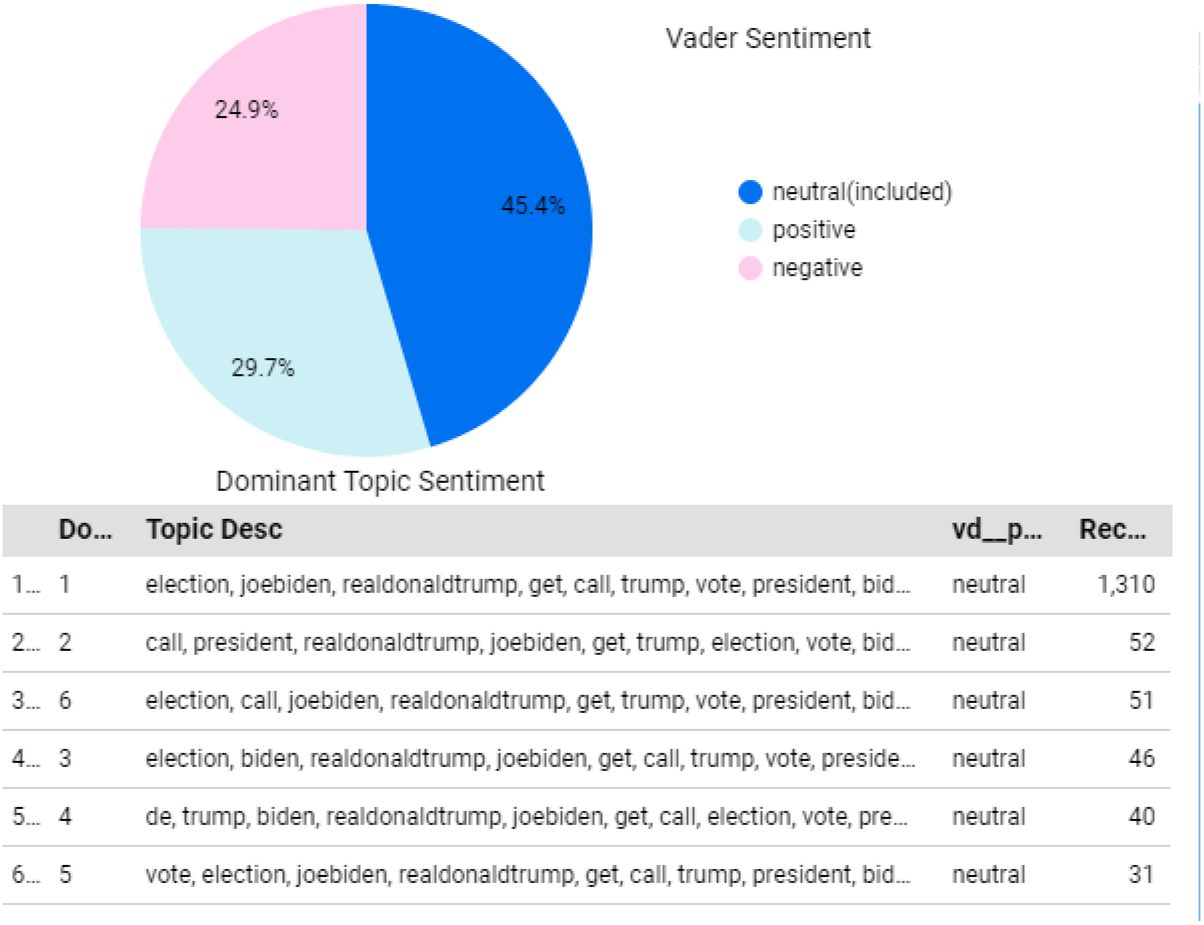


Dominant Topic Sentiment

	Do...	Topic Desc	vd_p...	Rec...
1...	1	election, joebiden, realdonaldtrump, get, call, trump, vote, president, bid...	neutral	1,310
2...	1	realdonaldtrump, trump, call, lot, steal, go, think, bitch, vote, us	negative	465
3...	1	win, trump, president, election, call, lose, go, people, well, know	positive	440

(A pie chart and table of topic 1 with the sentiments of the subtopics for election_elect_results_analysis.xlsx)

That is a huge portion for neutrality in the topic one. The neutral subtopic though is a discussion off both Trump, Biden, and the election which would make for a neutral fighting ground.

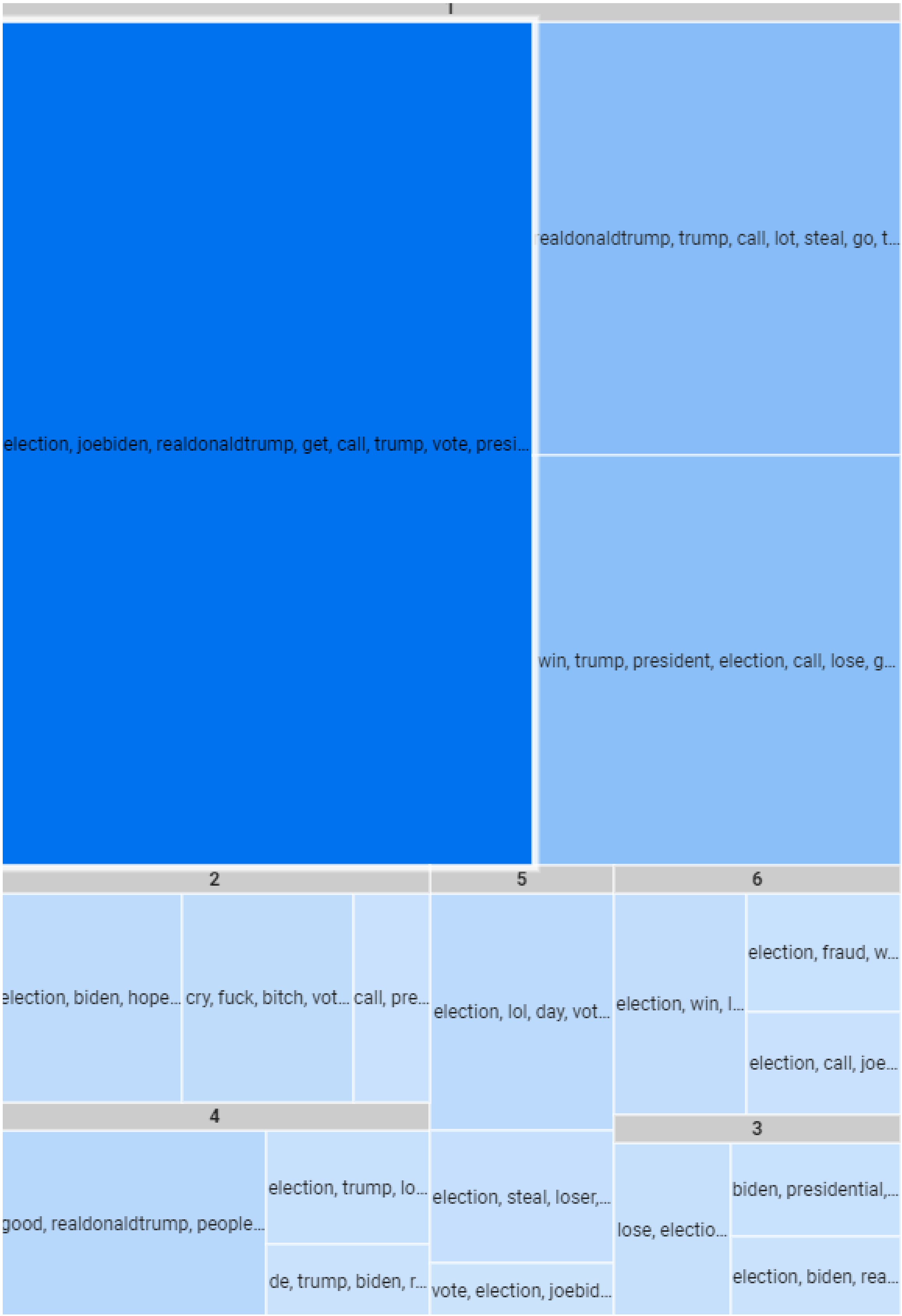


Dominant Topic Sentiment

	Do...	Topic Desc	vd__p...	Rec...
1...	1	election, joebiden, realdonaldtrump, get, call, trump, vote, president, bid...	neutral	1,310
2...	2	call, president, realdonaldtrump, joebiden, get, trump, election, vote, bid...	neutral	52
3...	6	election, call, joebiden, realdonaldtrump, get, trump, vote, president, bid...	neutral	51
4...	3	election, biden, realdonaldtrump, joebiden, get, call, trump, vote, preside...	neutral	46
5...	4	de, trump, biden, realdonaldtrump, joebiden, get, call, election, vote, pre...	neutral	40
6...	5	vote, election, joebiden, realdonaldtrump, get, call, trump, president, bid...	neutral	31

(A pie chart and table of all neutral sentiment subtopics for election_elect_results_analysis.xlsx)

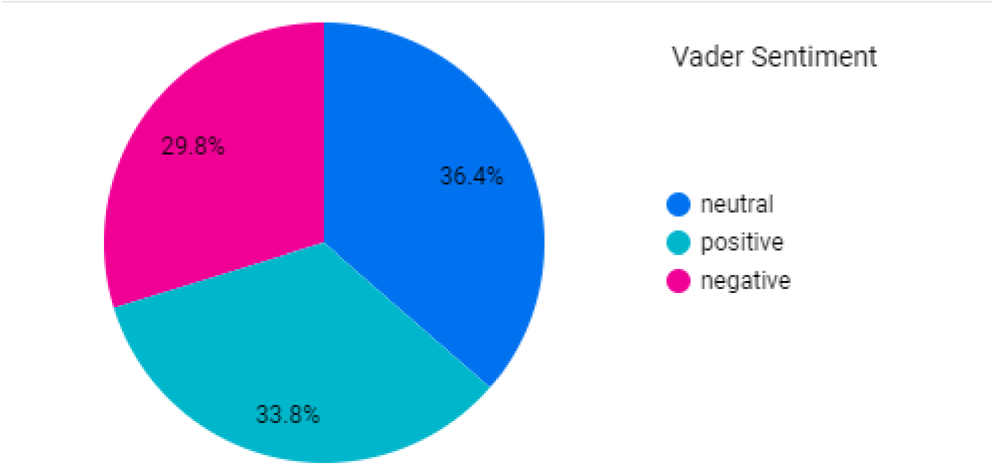
This is where the results of this major neutrality shift come from though in topic one's neutral subtopic. It is interesting to see this, but still does not really articulate whether the nation is polarized or not.


1
election, joebiden, realdonaldtrump, get, call, trump, vote, presi...
realdonaldtrump, trump, call, lot, steal, go, t...
win, trump, president, election, call, lose, g...
2
election, biden, hope...
cry, fuck, bitch, vot...
call, pre...
5
election, lol, day, vot...
election, steal, loser,...
vote, election, joebid...
6
election, win, l...
election, fraud, w...
election, call, joe...
4
good, realdonaldtrump, people...
election, trump, lo...
de, trump, biden, r...
3
lose, electio...
biden, presidential,...
election, biden, rea...

(A tree map of the 6 dominant topics and the three sentimental subtopics for them from election_elect_results_analysis.xlsx)

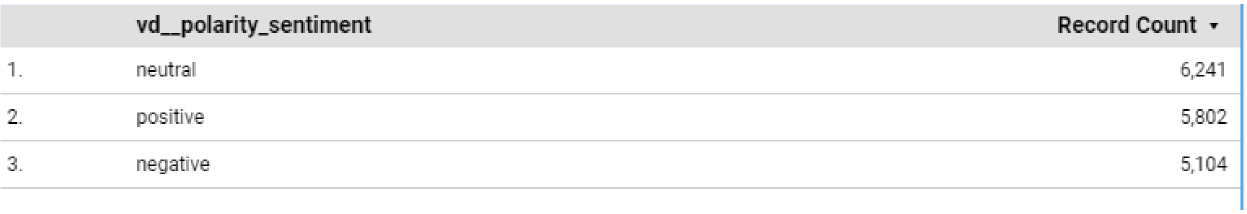
7.6 Election Overall Analysis

This analysis is the overall tweets (17,147) combined with 12 topics and 3 subtopics for the final overview. chose to combine all of them together to see if the topic model would change. The sentiment analysis would not change much since the sentimental polarity score is calculated individually for each tweet. However, the only change is seeing a combined aggregation of all the sentimental scores.



(A pie chart of the overall Vader Sentiment analysis for election_overall_analysis.xlsx)

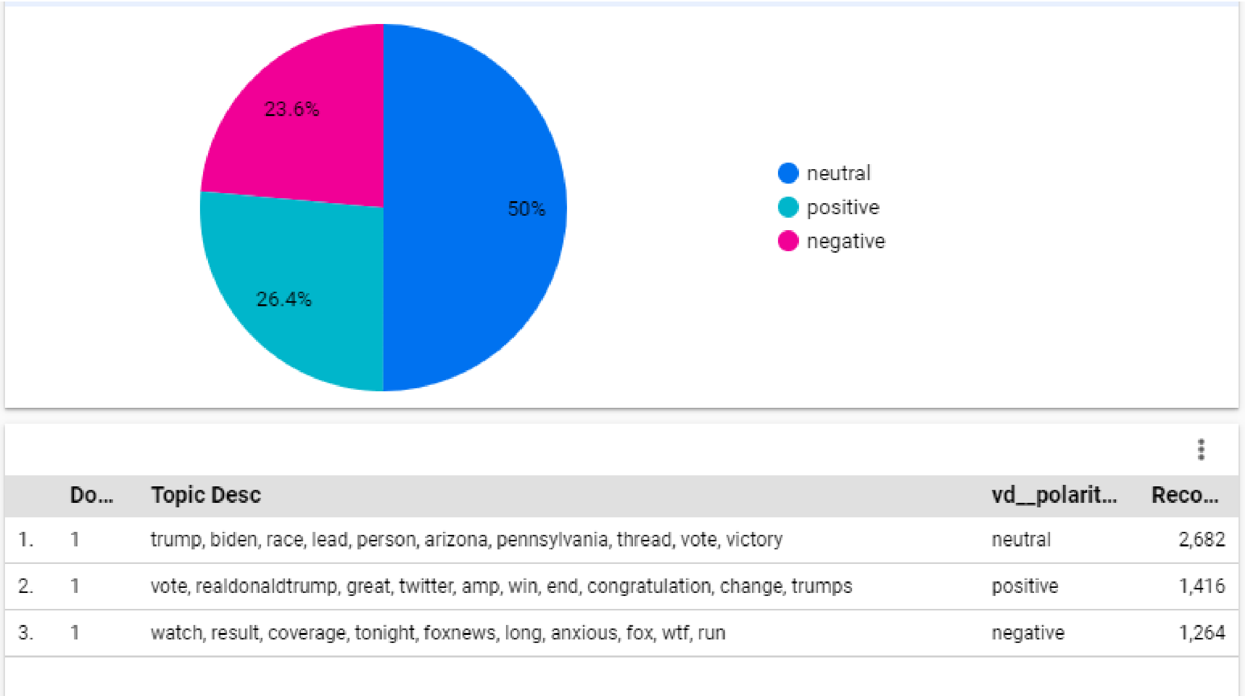
For all the days and various times that tweets were combined the neutral sentiment reigned supreme while the positive comes in second. This is still a close symmetrical pie chart where it was still a narrow escape between the positive and negative aggregated sentiment scores.



	vd__polarity_sentiment	Record Count ▾
1.	neutral	6,241
2.	positive	5,802
3.	negative	5,104

(A table of the overall Vader Sentiment analysis record count for election_overall_analysis.xlsx)

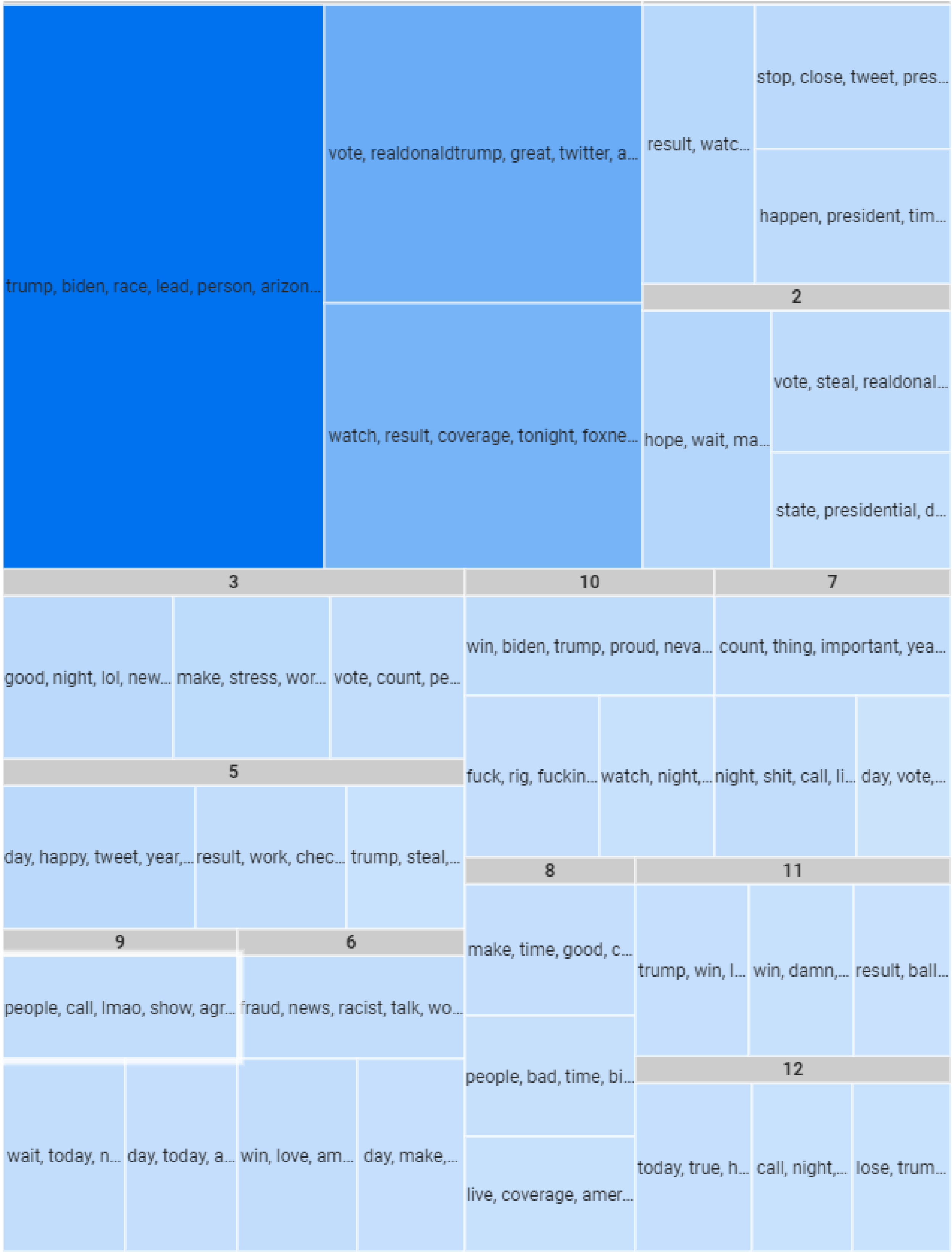
I would say all the sentiments from neutral, negative, and to positive are evenly distributed.



	Do...	Topic Desc	vd_polarit...	Reco...
1.	1	trump, biden, race, lead, person, arizona, pennsylvania, thread, vote, victory	neutral	2,682
2.	1	vote, realdonaldtrump, great, twitter, amp, win, end, congratulation, change, trumps	positive	1,416
3.	1	watch, result, coverage, tonight, foxnews, long, anxious, fox, wtf, run	negative	1,264

(A pie chart and table of topic 1 with the sentiments of the subtopics for election_overall_analysis.xlsx)

Looking at the first topic one could see this is where a third of the neutral records come from.



(A tree map of the 12 dominant topics and the three sentimental subtopics for them from election_overall_analysis.xlsx)

Examining the overall sentiment analysis and topic model though gives a better perspective as to if the nation is politically polarized or not. I do not like casting my judgement for such things like this and saying that the nation is polarized because one I am not qualified in political science or anything politics related for that fact. All I can say is that Looking at the sentiment analysis and trying to apply it to the entire general population is difficult to do.

8. Conclusion

8.1 Analysis conclusion

After looking at the analysis and conducting a research project aiming towards only using a sentiment analysis in conjunction with a topic model, I must conclude that this is not something for me to decide. I am no political science expert and the only complete set of qualifications I have are with data science, mathematics, and statistics. I cannot take a small sample such as this and apply it to the entire population nor can I say that everyone who's tweets were used in this analysis represents everyone in the U.S.A. Not everyone uses twitter and not everyone is tweeting at the times I collected tweets.

The way the sentiment analysis looks with the topic model suggests that the sentimental polarity is evenly distributed amongst the recorded tweets that are not retweets. The topic model was more for a way to see the overall sentiment of similar discussions and to give a closer look as to why such sentiments may have been more positive, negative, or neutral than others.

I do not think it is possible either to capture how the algorithms keep users on social media away from seeing each other's opinions. So therefore, I do not think I can conclude or say that this sentiment/topic modeling analysis captures the understanding of the U.S.A.'s political polarization, but only highlights that the sentiment and topic behind the election was pretty symmetrical with all the scores.

8.2 Moving forward

If this research were to be continued for further investigation or used as a standpoint for a political case, for research and political discourse, a lot more would need to go into improving the strength of the study. The duration of research would have to be extended, a larger dataset with different populations not just tweets, and multiple machine learning analytical approaches.

The duration for how long I collected tweets for this analysis was only four days. Even though I gathered a lot of tweets I do not think that they are enough to capture the entire picture. I think a yearlong research project with an appropriate data science research life cycle for application would be a great starting point. Defining the issue, mining the appropriate data from the right populations, cleaning the data, exploring the data, engineering the features, the appropriate analytical machine learning algorithms, and visualizing the issue for everyone to see. I do not think this type of research project will be how to solve the problem, but how do to highlight the issue.

The dataset will need to have multiple populations. Social media does not capture everyone's input. Even though it captures a huge portion of everybody's opinion it still does not capture those who do not use social media? An easier way to gather everyone’s opinion would be a government assisted poll/response survey be mailed to everyone with the intentions that it is all voluntary. There should be a response section, s multiple questionnaires asking those responding about their current livelihood to develop a demographic. Other questions would be if the respondent uses social media or not. It also should include multiple open-ended question sections for a person to fill out their open-ended response, and this will be useful for textual analytics portion of the research project.

The different machine learning methods for analyzing this forward moving research project will need to be diverse for capturing all angles of the data. Textual analytics will obviously be a big part of the research because of the multiple ways for analyzing textual responses. Some of the methods that could be utilized are the first obvious two which were used in this short project, but text similarity, semantic analysis, text summarization, and text classification are all great analytical tools of natural language processing within text analytics. Deep learning with transfer learning could be utilized to the benefit of text analytics since there are models already out there that are highly proficient when it comes to applying them to the data.

8.3 Possible Solution

The outcome of such a research project will only point towards the possible issues that everyone faces when it comes to political visibility and big tech company algorithms. The enlightenment of such an issue could pave the way into the future of how laws are made to ensure that big tech companies and their algorithms cannot condition everyone into opposite silos of each other based on political belief. Although everyone is entitled to their own opinion and free thought it is important that everyone is exposed to an opposing belief so democracy can still thrive.

References

Anas Al-Masri. (2019, May 14). How Does k-Means Clustering in Machine Learning Work? Retrieved December 1, 2020, from Medium website: https://towardsdatascience.com/how-does-k-means-clustering-in-machine-learning-work-fdaaaf5acfa0

Apply for access – Twitter Developers. (2020). Retrieved November 29, 2020, from Twitter.com website: https://developer.twitter.com/en/apply-for-access

blickpixel. (2014, October 25). Electrician Lego Repair - Free photo on Pixabay. Retrieved November 29, 2020, from Pixabay.com website: https://pixabay.com/photos/electrician-lego-repair-craftsmen-499799/

Clker-Free-Vector-Images. (2012, May 7). Mascot Blue Bird - Free vector graphic on Pixabay. Retrieved November 29, 2020, from Pixabay.com website: https://pixabay.com/vectors/mascot-blue-bird-twitter-tweet-48563/

Dipanjan Sarkar. (2019). *Text analytics with Python : a practitioner's guide to natural language processing*. New York: Apress Springer Naure. (Original work published 2019)

geralt. (2019, November 10). Mockup Hands List - Free image on Pixabay. Retrieved November 9, 2020, from Pixabay.com website: https://pixabay.com/illustrations/mockup-hands-list-stickies-4612927/

Kavita Ganesan. (2018, November 2). What are N-Grams? | Kavita Ganesan, Ph.D. Retrieved December 7, 2020, from Kavita Ganesan, Ph.D website: https://kavita-ganesan.com/what-are-n-grams/#.X85r8NhKiUk

Natural Language Toolkit — NLTK 3.5 documentation. (2020). Retrieved November 29, 2020, from Nltk.org website: https://www.nltk.org/

NLP | How tokenizing text, sentence, words works - GeeksforGeeks. (2019, January 28). Retrieved November 29, 2020, from GeeksforGeeks website: https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/

pandas - Python Data Analysis Library. (2020). Retrieved November 29, 2020, from Pydata.org website: https://pandas.pydata.org/

Peggy_Marco. (2016, November 18). Builders Master Builder - Free image on Pixabay. Retrieved November 29, 2020, from Pixabay.com website: https://pixabay.com/illustrations/builders-master-builders-builder-1825689/

PyMongo 3.11.1 Documentation — PyMongo 3.11.1 documentation. (2020). Retrieved November 29, 2020, from Readthedocs.io website: https://pymongo.readthedocs.io/en/stable/

qimono. (2016, October 10). Question Mark - Free image on Pixabay. Retrieved November 29, 2020, from Pixabay.com website: https://pixabay.com/illustrations/question-mark-question-mark-sign-1722862/

The most popular database for modern apps. (2020). Retrieved November 29, 2020, from MongoDB website: https://www.mongodb.com/

Tweepy. (2020). Retrieved November 29, 2020, from Tweepy.org website: https://www.tweepy.org/