John Voorhess

voor@umich.edu

SI370 Final Project

Chat.db Analysis

Motivation and Research Questions

When I learned that Apple's iMessage applications stores all text messages natively in a directory on my computer I became intrigued. I wondered what kind of insight I might be able to glean from analyzing my own communication with friends, but it wasn't until recently that I had the tools to do so. The open-ended nature of this assignment allows me to pursue an analysis of a data set with which I have been interested for several months.

Several questions come to mind regarding text analysis of my own text messages. Particularly, I am interested in whether my own suspicions about a friends declining mood will be reflected in our texts with each other.

1. Are there observable trends in sentiment?
2. Can a model be fit to a period of text message sentiment?
3. What characterizes a negative or positive text message?
4. Who are the most negative or positive senders?
5. What are the most common words used?
6. Can we characterize senders by the texts that they send?

By answering these questions I hope to learn more about myself and my friends and employ techniques learned in SI370 as well as some that I have developed on my own.

The Data

Apple OSx store all iMessages in a sqlite database called chat.db. By default, this database is store at the following filepath: /Users/<yourusername>/Library/Messages/chat.db. Sqlite3 is installed on Apple computers from the factory, so accessing the chat.db is as simple as running the following command in the terminal:

sqlite3 /Users/<yourusername>/Library/Messages/chat.db

Once connected to chat.db, it can be explored by using the command, ".tables". This will list all of the tables in the database. The command, ".schema [tablename]", where the tablename is replaced by the name of the table to be explored, will retrieve a CREATE statement that shows a complete picture of the structure of the table and all of the data types.

Once I had a handle on the structure of the database and its tables, I ran a few queries to determine which fields I would include as columns in my main dataframe. I created a single dataframe with a row for each text message in my computer and the following columns:

- 'time'
    - The time at which the message was sent.
- 'id'
    - A unique ID for each message.
- 'text'
    - A string representation of the message.
- 'sender'
    - An integer representing the contact with whom the message was conveyed.
- 'Is_from_me'
    - A boolean value
    - ‘0’ represents a text not sent by me.
    - ‘1’ represents a text sent by me.

Sentiment

Since the central question of this project is sentiment analysis, a tool is necessary to add sentiment scores to the dataframe. The Natural Language Tool Kit’s Vader package is particularly well suited to this task. The Vader package is designed for messy social media posts and accounts for a wide range of colloquial speech found in the type of terse text typically found in text messages and tweets. A call to Vader’s SentimentIntensityAnalyzer.polarity_scores() returns a dictionary of values that includes:

- 'Neu': a float ranging from 0 to 1.
    - The ‘neutrality’ score of the text.
- 'pos': a float ranging from 0 to 1.
    - The ‘positivity’ score of the text.
- 'neg':a float ranging from 0 to 1.
    - The ‘negativity’ score of the text.
- 'compound': a float ranging from -1 to 1.
    - The ‘compound’ score of the text that is an aggregate of the three previous scores.

These scores have been added to the dataframe as compound_polarity_score, Positivity_score, negativity_score, and neutrality_score for further analysis.

In their 2018 paper, In an Absolute State: Elevated Use of Absolutist Words Is a Marker Specific to Anxiety, Depression, and Suicidal Ideation, Mohammed Al-Mosaiwi and Tom Johnstone propose a list of 'absolutist' words present in greater proportions of communication by those who are depressed versus those who do not identify as being depressed by analyzing depression-related communication on the social news-aggregation website, Reddit. I have taken their list of ‘absolute’ words and added to quantity and proportion of these words to the dataframe.

Analysis and Results

Q1 - What characterizes a negative or positive text message?

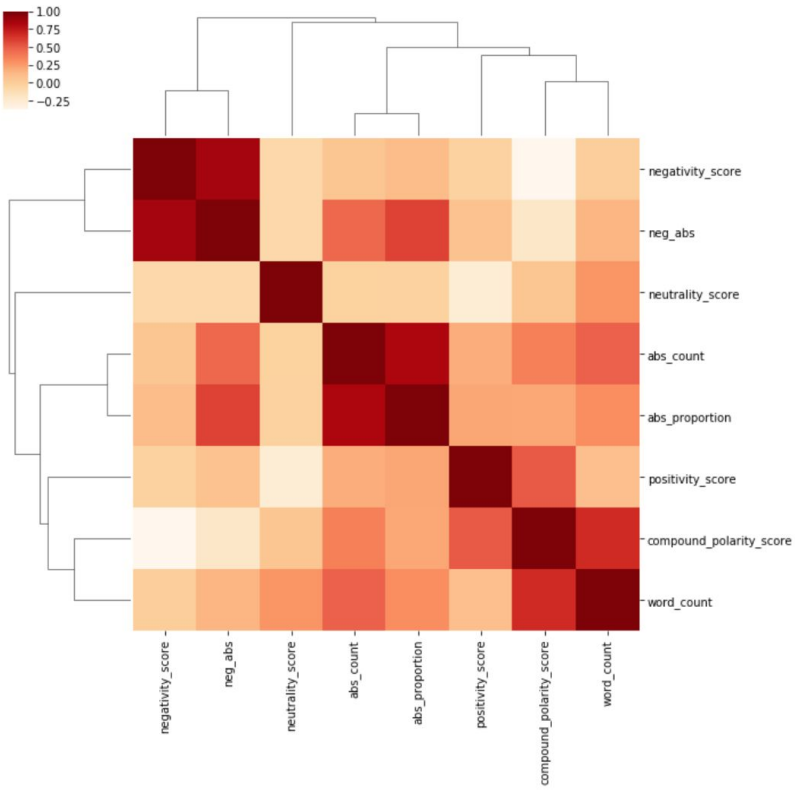
To characterize the difference between positive and negative texts, I set a threshold of positivity and negativity scores of greater than .5 for all texts in the dataframe and find the frequency of words used in those messages.

- For positive text messages, the most frequent words found in the texts are:
  - **['i', 'love', 'you', 'good', 'a', 'hope', "i'm", 'you!', 'so', 'have']**

- For negative text messages, the most frequent words found in the texts are:
  - **['no', 'i', "i'm", 'oh', 'that', 'a', 'fuck', 'sorry', 'is', 'so']**

- The most common ‘absolute’ words used in the corpus are:
  - **['all', 'totally', 'definitely', 'never', 'always', 'everything', 'whole', 'everyone', 'every', 'full']**

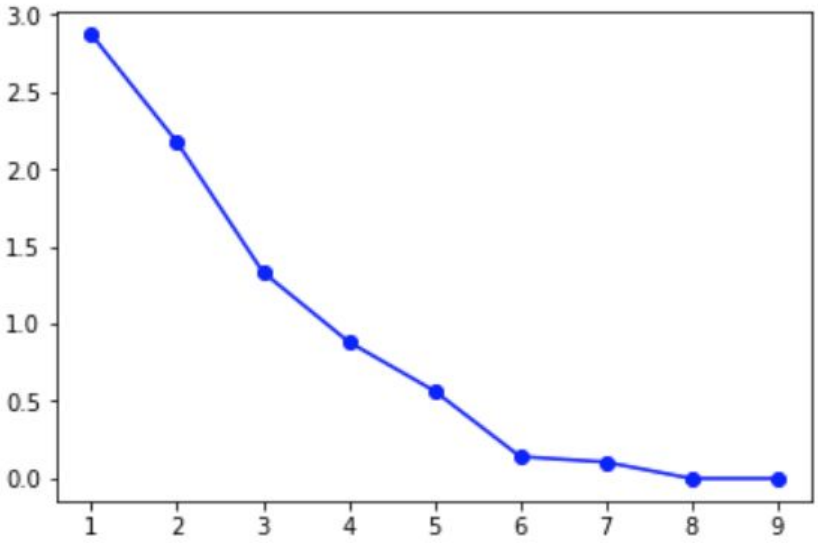
Furthermore, I have created a correlation matrix to determine if there are any unexpected correlations between any of the factors that we are examining:



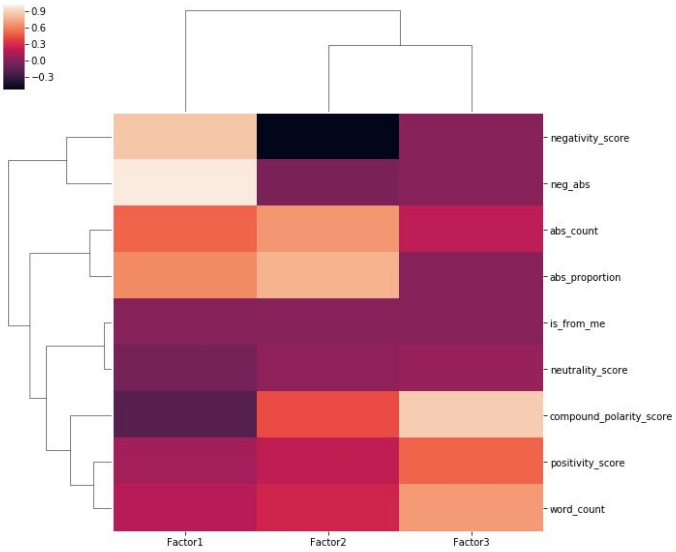
As expected, there are no strong correlations between fields in the dataframe that are not derived from or related to each other. For instance ‘neg_abs’ and ‘abs_proportion’ appear to be strongly correlated, but ‘neg_abs’ is a weighted average of the negativity score and the proportion of absolute words found in a text. I would expect these to be correlated since one is derived from the other.

Principle Component Analysis of the dataframe fields:

A scree plot reveals that we will need three principle components since the accepted threshold is to take components over 1.0.



We can visualize the factor loadings using a clustermap:



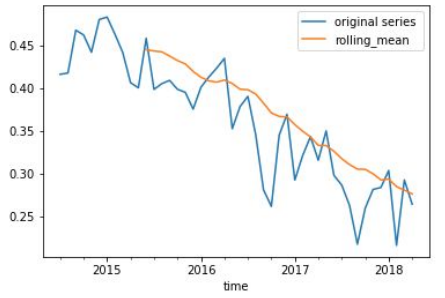
Factor One seems to be characterized by the negativity score, neg_abs compound score, and the proportion of absolute words.

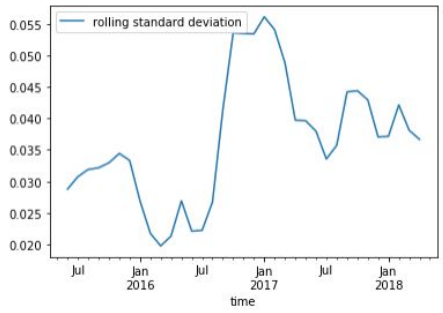
Factor Two seems to be characterized by the count of absolute words, the proportion of absolute words, and the compound polarity score.

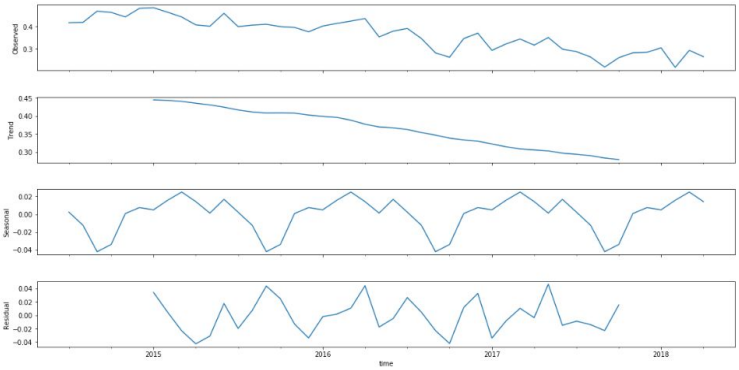
Factor Three seems to be characterized by the word count, positivity score, and compound polarity score.

Q2 - Are there observable trends in sentiment?

My hypothesis is that a particular contact will display a downward trend in sentiment over the several years that we have been communicating. By grouping this sender's messages by month and averaging their compound polarity score we can use time series decomposition to plot the trend, auto-correlation, and partial auto-correlation.







The plots confirm that there is a negative trend in this contacts compound polarity score over time as well as seasonality present. An attempt to fit an ARIMA model, however, was unsuccessful, producing a model with a mean absolute error of .04.

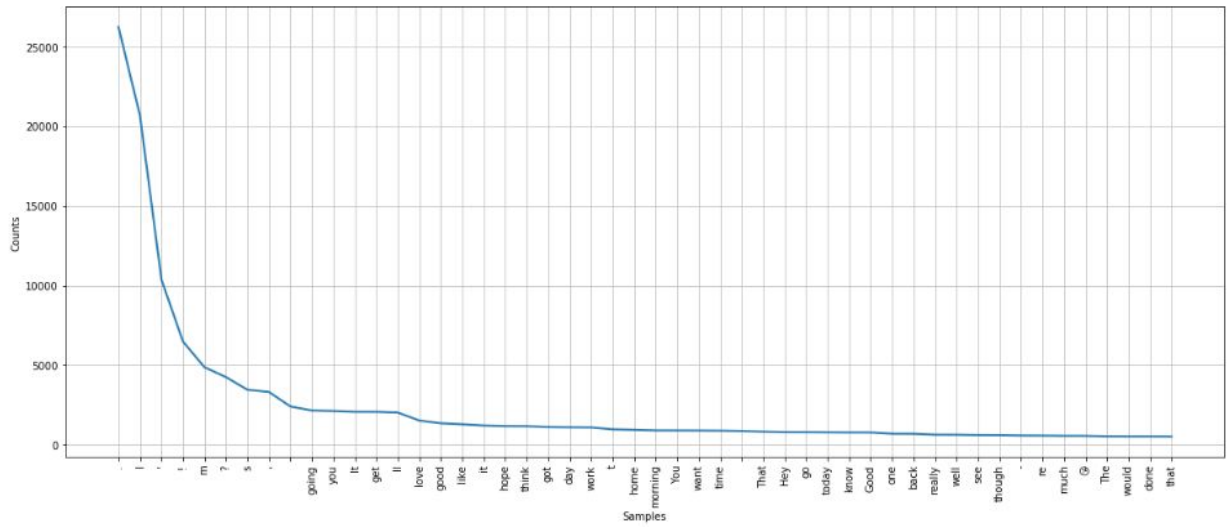
Q3 - Who are the most negative or positive senders?

By computing the highest average positive and negative sentiment scores in the dataframe grouped by sender, it was revealed that the most positive sender was one of my previous building managers from an apartment in Seattle, Kelsey. Oddly enough, the most negative sender was my next building manager in Seattle, Maggie. Perhaps her negative sentiment had something to do with the fact that I lived directly above her apartment.

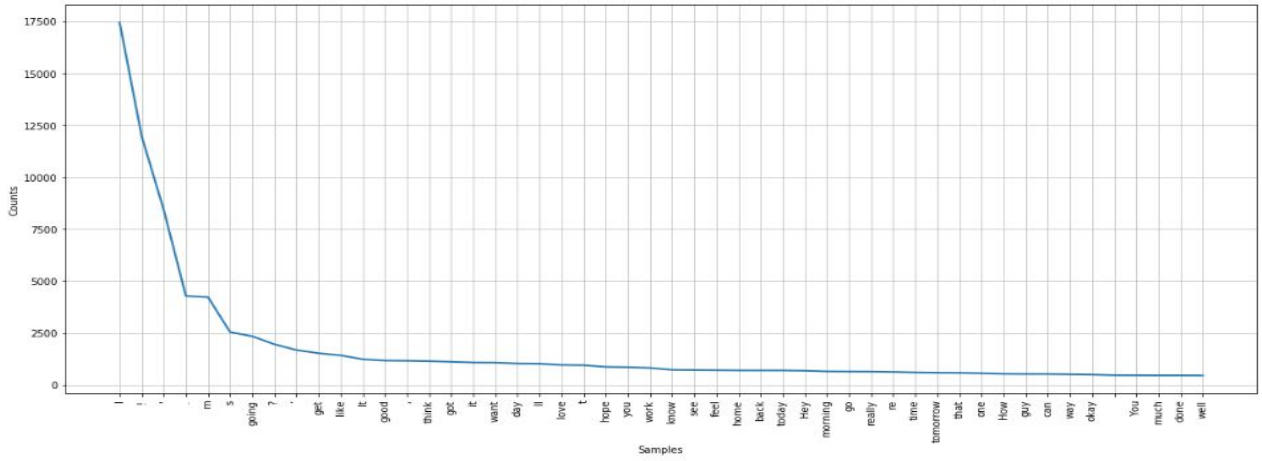
Q4 - What are the most common words used?

Using the Natural Language Tool Kit's tokenizer I was able to tokenize all of my own and my partner's text messages and plot the most frequent words that we use as a method of characterizing our communications.

My own word frequency when communicating with my partner:

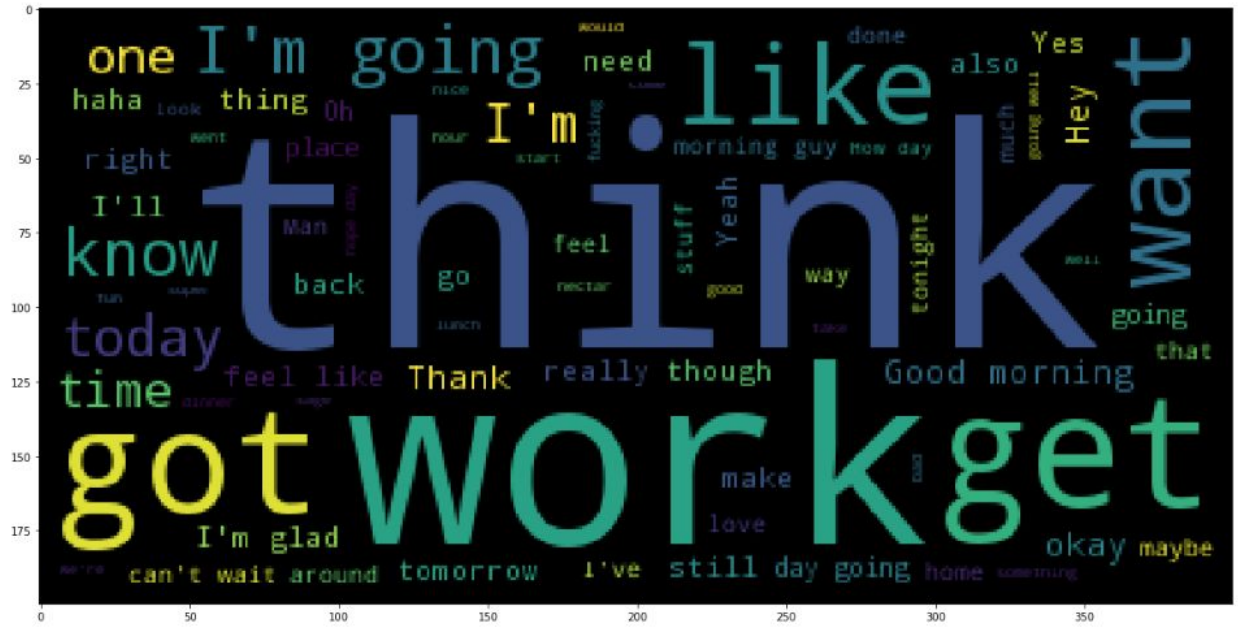


My partner's word frequency when communicating with me:



When I use a wordcloud to visualize the frequency of each, it reveals that we use very similar word frequencies with each other. I suspect that during our four years together, we have learned to communicate from each other such that our style of communication has become very similar.

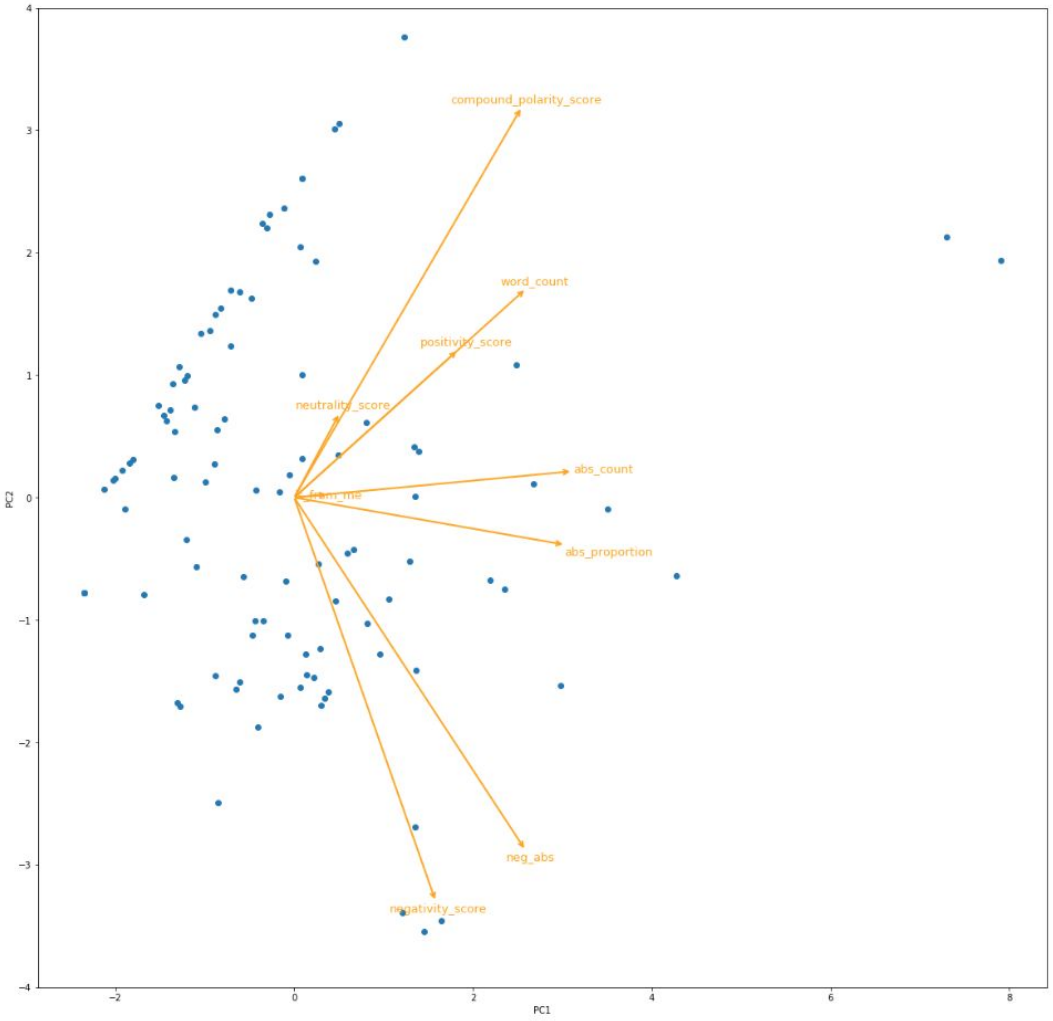
My partner's word cloud:



My own word cloud:

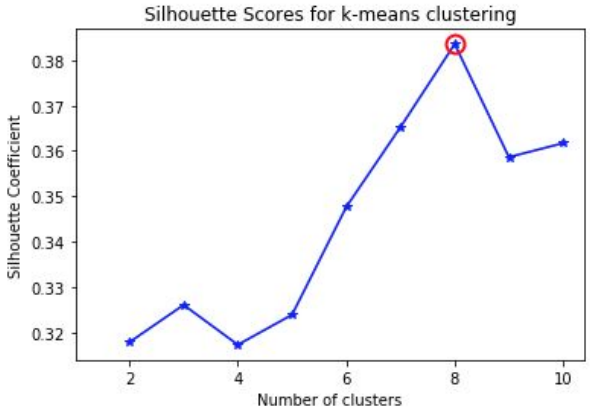
Q5 - Can we characterize senders by the texts that they send?

We can plot the dataframe grouped by sender as a PCA biplot to visualize the distribution and differences between senders:



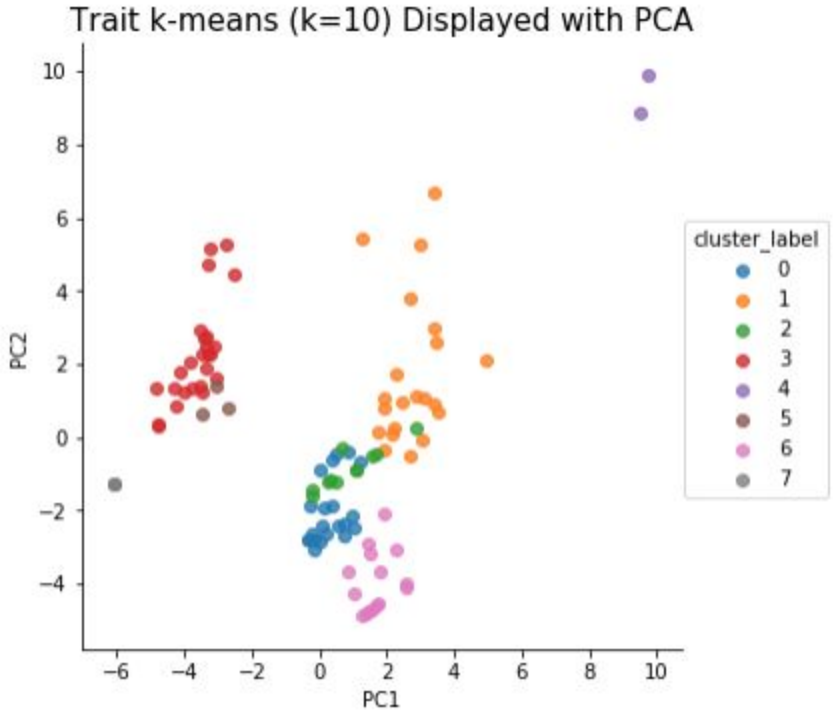
Of interest, we do find that word count is dissimilar to negativity score. This would suggest that shorter texts are, on average, less positive than longer texts.

To characterize senders by their texting habits, we use k-means clustering.

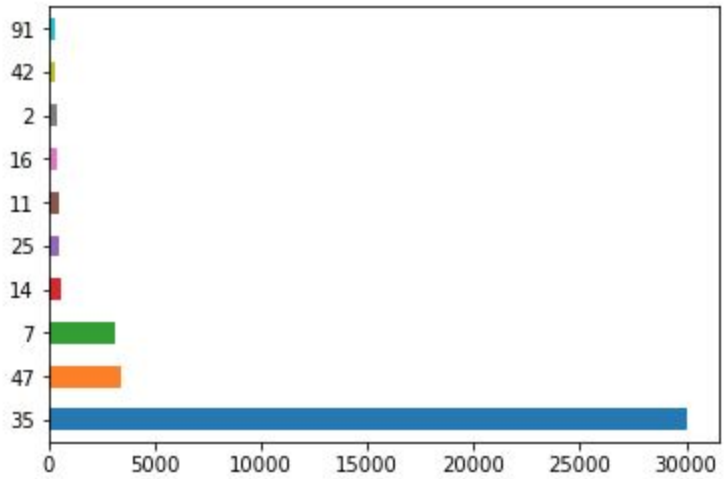


Silhouette scores suggest that the highest average silhouette coefficient results from eight clusters.

I have displayed the clusters of senders and color coded them. I was particularly interested in the senders that appeared at the top right of the frame. Upon investigation, it was revealed that these were senders who send very few, but positive messages. Opposite these in the bottom left of the frame are senders who sent empty or null-value messages.



Q6 - Can we accurately classify a text message by sender using machine learning techniques?



A significant challenge presented by this data set is the disproportionate quantity of text messages from just a few senders. For example, this graphic shows the number of text messages from the most frequent senders. The top sender, my partner, accounts for ten times more messages than the next most frequent sender. This means that the 70% accuracy that I get from the model is probably really zero because so many messages come from my partner.

To account for this mis-proportioned data set, I have used a resampling technique to randomly choose texts from each sender with replacement to create a dataframe with 1000 text messages from each sender.The result, after several hours spent computing the optimal max_df and min_df for the classifier, is a classifier with **85% accuracy**. A multinomial naive bayes classifier is used because a gaussian classifier would be inappropriate for vectorized text.

Next Steps

A logical next step in this operation would be finding more parameters with which to classify texts, e.g. time of day, various proportions of types of speech, etc. It would be interesting to compare classifiers built with different techniques to see which is more accurate. Additionally, I would like to figure out a better way to build a model and predict seasonality of sentiment. My own sentiment did not present any clear trends over time, but it appears that there is more to be learned from some of my other contacts.

References

Al-Mosaiwi, Mohammed & Johnstone, Tom, In an Absolute State: Elevated Use of Absolutist Words Is a Marker Specific to Anxiety, Depression, and Suicidal Ideation, *Clinical Psychological Science* (2017), retrieved from https://doi.org/10.1177/2167702617747074