**Fall 2020 Independent Study Report**

**Joe Soonthornsawad**

**Note:** Please also refer to the “readme” document as well as files referenced in the document in the folder.

**I. Introduction**

**Research Overview**

Current methodologies related to tracking COVID-related behaviors and sentiments include using survey methodologies, online search data, mobility data, and studying tweets about pandemic information. However no work has been done designed to identify, classify, and measure changes in relative frequency of tweets that describe COVID-related behaviors. Alongside analysis of data mentioned previously, this analysis of self-reported behavior on Twitter can provide a new source of granular information that could be useful in helping shape the response to future public health crises, and more broadly, furthers the state of the art as it relates to how Twitter data can be used in computational social science research.

**Machine Learning Step**

Building from research conducted this summer, this semester’s work aims to create two machine learning classifiers designed to classify tweets related to the COVID-19 pandemic. This project aims to detect tweets that represent self-reported behaviors that have increased that represent safer behaviors (referred to in this paper as “increased”) and behaviors that may have decreased that represent less safe behaviors (referred to as “decreased”). The goal is to be able to develop a technique to identify tweets that describe a behavior, and later, plotting the change in relative frequency of tweets that are related to the pandemic in a time-series graph.

**II. Early Work (June - August)**

**Selecting states**

This summer, tweets were pulled from a 10% sample of all tweets from 5 states, pre-selected because of differences in region, politics, and difference in each state governments’ approach to imposing public health-related restrictions. These states were Michigan, New York, Idaho, Alabama, and Florida.

**Creating samples by state**

From the 10% sample of all tweets from Decahose, tweets were collected based on users who have tweeted from that state. If a user tweeted from a state once, they were assigned to a state, and their entire timeline was pulled for January 1, 2020 to July 13, 2020 and added to a file. Drawing from this data housed on the server, tweets were then filtered for “increased” and “decreased” keywords and phrases for each state. Following this, a filter for keywords was applied to separate tweets into increased and decreased categories for two of these states: Idaho and Alabama. Note that further validation as to whether users in this dataset are tweeting from the state in question did not occur because of time constraints for the poster presentation.[[1]](#footnote-0)

**Labeling tweets, refining keyword and phrase selection**

Increased and decreased tweets were classified from Idaho and Alabama in order to develop a set of guidelines for labeling, as well as serving as a preliminary check on the relevance of keywords and phrases. This early labeling process led to subsequent rounds of adding keywords and phrases, and removing irrelevant ones.

**Visualization**

For the states of Alabama and Idaho, all tweets were gathered for a given day from January 1 2020 to July 13 2020. Then the proportion of tweets with a keyword or phrase describing a behavior to all tweets from that day was calculated, compared to the total amount of tweets from each state from that same time period. This proportion was then visualized in a time series plot, and compared to week-over-week case growth and timelines for shutdowns and reopenings.

Notably, these tweets were all assumed to be positive. The purpose of the machine learning classification process was to build a classifier that can help reduce this noise in this data, as well as introduce the possibility of classifying tweets as “increased” and “decreased” from across the country that may contain a keyword.

**III. Labeling data for Machine Learning**

**Provenance of the Data**

Beginning in September, labeling of 4568 “increased” behavior tweets and 5298 “decreased took place, randomly sampled from each state: Michigan, New York, Alabama, Idaho, and Florida.

This sampling process is described in section “ II: Early Work (creating samples by state)”. The files which samples were drawn have file names formatted: “<state name>\_<increased>.csv” or “<state\_name>\_<decreased>.csv”, and are located in the csv folder of this repository.

**Labeling**

Labeling took place based on the guidelines set out in the document activity\_tweet\_annotation\_guide.doc in the home folder of “covid\_twitter”. A tweet that described a behavior received a “1”, while a tweet that did not describe a behavior received a “0”.

**New filtering protocol emerges**

In addition to labels to prepare data for machine learning, this process yielded new tweet filtering guidelines that were not uncovered during earlier phases of this research this summer. Tweets that were replies or retweets (tweets that start with @<username>) were dropped. Additionally, tweets that were from official or brand accounts would need to be excluded. Excluding official or brand account tweets is particularly important because these tweets are posted as part of an official statement from a public personality company or institution, rather than that of a private citizen.

The process to clean and prepare data from each state for labeling is described in the next sub section.

**Preparation of data for labeling**

This process takes place in the file “convert\_drop\_rts\_rand\_sample.ipynb”.

The main processes:

1. dropping retweets and replies (based on a tweet starting with @)
2. converting tweet data (text, user name) to proper data types
3. Randomly sample 1500 tweets from each state
4. Export a dataframe of those tweets to a .csv

During the process of writing this report, it was discovered that New York was excluded from the sample for increased tweets. This was likely due to time constraints on the labeling process, or perhaps due to human error. The same can be said for parity in the amount of labels used for both increased and decreased training sets.

**IV. Excluding more keywords/phrases**

**Keywords and Phrases**

Note: The keywords and phrases used before further filtering are detailed in random\_decreased\_labeling/freq\_table/freq\_table\_dec.csv and random\_increased\_labeling/freq\_table/freq\_table\_inc.csv. This is excluded here due to the length of these tables.

**How Keywords and Phrases were Excluded**

Labeling provides an additional check on keywords and phrases; a label is another data point that helps clarify keyword/phrases.Though each keyword and phrase was designed to pick up on a particular behavior, additional checks are required to ensure these terms are both relevant (do not occur rarely in the sampled data) and actually describe a behavior (indicated by the proportion of positives).

Frequency tables were constructed to help make these determinations. For each keyword or phrase in the random sample, the number of tweets that contained the word or phrase were recorded, as well as how many of these tweets positive, negative, and the average rate of positivity was calculated. These tables can be viewed in more detail in the documents “freq\_table\_dec.csv” and “freq\_table\_inc.csv”.

As mentioned earlier, thresholds and decisions to remove were designed to root out words/phrases that did not appear often in the data, and ones that may not actually reflect the action/behavior they intended to capture. For example, tweets containing the word “haircut” appeared 44 times in the random sample of “decreased” tweets, but only 1 tweet was labeled positive; indeed, many of these tweets described people “needing a haircut” or talking about how long it has been since they have gotten a haircut, rather than going to get one.

**Keywords/phrases Eliminated**

Keywords and phrase were dropped from the final sample of classified tweets, based on the following rules:

**for "increased" words:** words that have a frequency larger than 20 and a positive rate lower than 10% were removed (social media, post, watches, deliver, on twitter, clean, read, gaming, stay in).

**For "decreased" words:** a threshold of frequency 20 and positive rate lower than 3%. (fly, concert, service, tennis, church, practice, haircut, work out, meeting).

**Irrelevant words/phrases**: these terms did not accurately fit the “decreased” or “increased” activity, but were hold-overs from earlier stages of the research process. For decreased: ('drink', 'drunk', 'drinking', 'drank', 'drinks'). For increased: ('barred', 'tripped')

**Dropping rejected keywords and phrases**

This process involved cleaning the decreased and increased .csvs from each of the 5 states. In “freqtables\_mlprep\_mlunlabeledprep.ipynb”, the following processes are carried out:

1. Upload of all labeled and unlabeled data for each of the five states.
2. Dropping a tweet that *only* contains one or more of the rejected keywords and phrases from labeled data.
3. For each behavior type (increased and decreased), concatenate all labeled data from each state into one increased labeled data frame and one decreased labeled data frame
4. Convert to inc\_ml\_ready.csv and dec\_ml\_ready.csv.
5. Create data frames of unlabeled data for each state by dropping tweets with tweet ids that are found in labeled data.
6. Convert unlabeled data to .csvs (inc\_unlabeled.csv, dec\_unlabeled.csv)

**V. Preparing for Machine Learning Classification**

**Data Preparation**

Classifiers were prepared for increased and decreased behaviors, using “ml/inc\_ml\_ready.csv” and “ml/dec\_ml\_ready.csv” for training, validation, and testing. Data was split 80% for training, 10% for validation, and 10% for testing. Further data cleaning and casting to correct data types was needed, as these .csvs had numeric labels encoded as int-like strings and float-like strings. There were also a very small number of labels that were simply a character (such as “d”) that were imputed as 0s.

**TF-IDF Vectorizer**

A TF-IDF vectorizer was chosen for this text classification task. TF-IDF stands for “term frequency-inverse document frequency”, meaning the weight assigned to each token not only depends on its frequency in a document but also is offset by the frequency at which the term trm occurs in the entire corpora.

**SVM Classifier**

An SVM classifier was selected for this process because of this classifier’s efficacy and efficiency in handling high dimensional data, especially data in which the amount of dimensions exceeds the number of samples. This is the case with text classification, in which texts may contain several unique tokens and feature counts can be adjusted to reach the tens of thousands.

For more information, see the “svm\_increased.ipynb” and “svm\_decreased.ipynb files”. These notebooks are fully annotated and describe all the steps taken during the machine learning process.

These two classifiers were designed to optimize for f1 score. More detail on this will be provided in the section “VI. Tuning Hyperparameters”.

**Challenges and lessons learned**

Many of the challenges in this part of the process were due to the technical intricacies of implementing and tuning a machine learning classifier for the first time.

At first, vectorizers were created without excluding stopwords, resulting in the inclusion of features that were not particularly useful to identifying a behavior-related tweet.

There were also problems that arose from an unfamiliarity with the machine learning process Additionally, poor variable naming sometimes resulted in confusion, as well as local edits to parts of the notebook not being reflected throughout the entire program. For example, the classifier was, in different iterations of the code, incorrectly built using the testing data rather than the training data. Also, there was some confusion related to creating a vectorizer, resulting in not consistently using the same vectorizer for each combination of hyperparameters.

In earlier versions, there was also an important flaw with the training data labels not completely being converted into integers, which dramatically affected the results of classifier creation and tuning. This error was caught through restarting the kernel and re-running code line by line to help catch bugs.

The lessons learned from these missteps are reflected in the final updated notebooks “ml/svm\_increased.ipynb” and “ml/svm\_decreased.ipynb”, which clearly detail the process for implementing an SVM classifier step-by-step, tuning using hyperparameters, and interpreting the results, with comparison to 3 naive classifiers. More detail on this process and these classifiers can be found below.

**VI. Tuning Hyperparameters**

**Why optimize for f1?**

Optimizing for precision (true positives/ true positives + false positives) can result in a model that is highly accurate when it predicts a tweet is positive, but at risk of a relatively conservative classifier that may miss positive tweets. Optimizing for recall (true positives/ true positives + false negatives) can result in a model that finds most of the positive tweets, but at risk of returning many irrelevant ones.

For this prediction task, the goal is to improve upon naively assuming that all tweets that contain a keyword or phrase are positive — the approach taken in the visualizations for the poster session presentation this past August. The f1 score is the harmonic mean of the precision and recall, which results in both metrics to be accounted for.

**Hyperparameter Tuning Function**

To optimize for the f1 score, a hyperparameter tuning function and a vectorization function was created using Sci-kit learn’s sklearn.feature\_extraction.text library and three nested for loops to iterate through combinations of ngram values (the length of the sequence of words in a single token), maximum feature numbers, and minimum document frequency (when building the vocabulary for the vectorizer, the fewest amount of times a term can show up). These hyperparameters are important because they control how these texts are understood by the classifier, and the way different attributes of the data are determined to be salient. The functions can be found in the “ml/svm\_increased.ipynb” and “ml/svm\_decreased.ipynb” documents in lines 20 (feature\_preprocessing) and 34 (try\_features).

The function try\_features creates a classifier with the training data, using MDF, ngram, and max features as parameters. This classifier is then tested on the validation data to determine its performance. The final fscore, mdf, ngram, and features used to achieve that score, were are added to a row of a data frame. When all combinations have been tested, the function returns this data frame, which was used to determine an optimal set of hyperparameters to optimize the f1 score.

**Evaluating performance and comparing with baselines**

The results were visualized in scatterplots, where each plot is a different ngram and mdf pair, x represents the number of features, and y represents the f1 score. This dataframe was also sorted by f1 score, descending, to determine the combination(s) that yielded the highest f1 score. A full visualization of these plots can be found in the “ml/svm\_increased.ipynb” and “ml/svm\_decreased.ipynb” notebooks.

Once the optimal combination of hyperparameters was selected based on F1 score, a classifier was trained using the corresponding hyperparameters, and tested on the test data. 3 baseline naive classifiers, classifying testing data according to predetermined rules were used to test the performance of the classifier. The results of these tests and detail on these dummy classifiers are detailed in fig 1.2 and fig 2.2.

The best performing hyperparameters for “increased” and “decreased” tweets are illustrated in the tables below alongside 3 naive classifier benchmarks. Additionally, top 20 features by weight (absolute value) are visualized as an additional sanity check for the classifier (see fig 1.3 and 2.3).

**Increased**

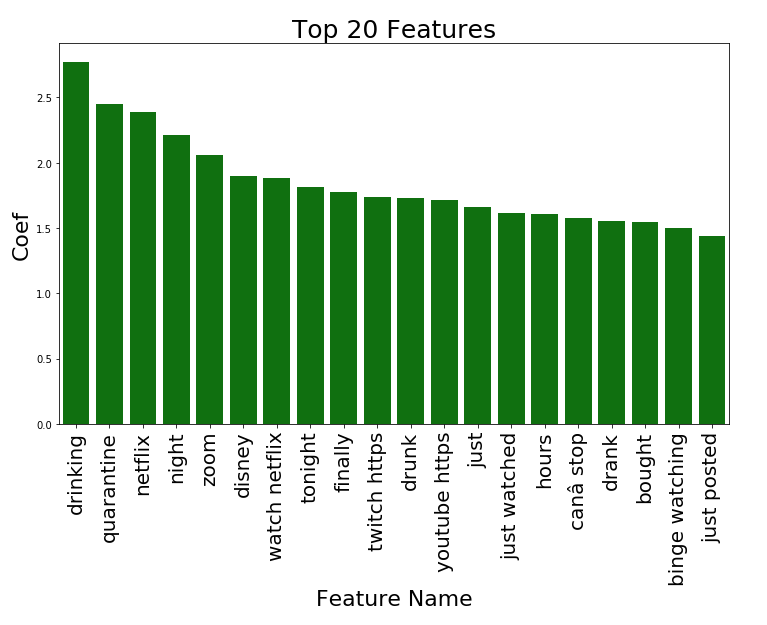
**Fig 1.1. Positives and Negatives in Labeled Increased Data**

| **Data type** | **Positives** | **Negatives** | **Total** | **Prop pos.** |
| --- | --- | --- | --- | --- |
| Training | 1056 | 2598 | 3654 | .2890 |
| Validation | 145 | 312 | 457 | .3173 |
| Testing | 130 | 327 | 457 | .2845 |
| Total | 1331 | 3237 | 4568 | .2914 |

**Fig 1.2. Classifier Performance on Testing data (Increased)**

| **Classifier** | **Notes** | **Precision** | **Recall** | **f1** |
| --- | --- | --- | --- | --- |
| SVM | Mdf: 1,  ngram: 2,  Max Features: 19000 | 0.6563 | 0.4846 | 0.5575 |
| Constant | all positive | 0.2845 | 1.0 | 0.4429 |
| Stratified | based on training set class distribution | 0.2917 | 0.3231 | 0.3066 |
| Uniform | generates predictions uniformly at random | 0.3084 | 0.5077 | 0.3837 |

**Fig 1.3. Top 20 Increased Vectorizer Features by Absolute Value** (Green indicates positive weight)

****

**Decreased**

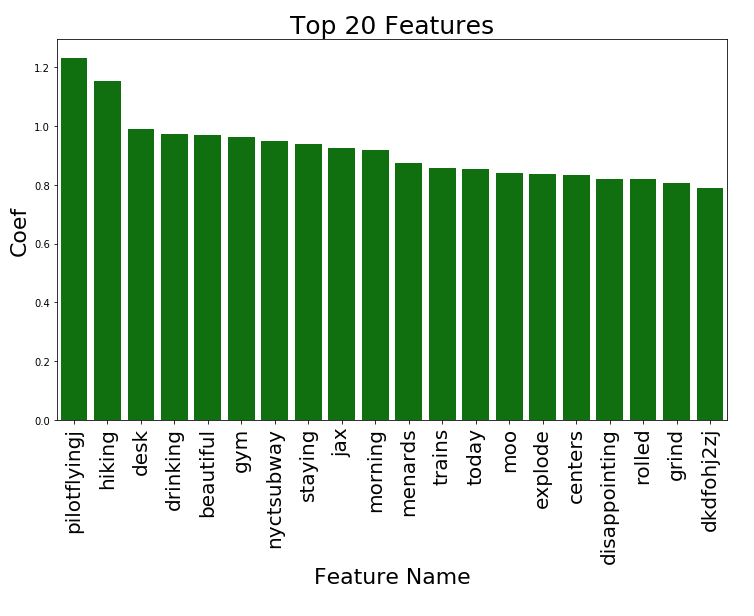
**Fig 2.1. Positives and Negatives in Labeled Decreased Data**

| **Data type** | **Positives** | **Negatives** | **Total** | **Prop. pos.** |
| --- | --- | --- | --- | --- |
| Training | 280 | 3958 | 4238 | .0661 |
| Validation | 40 | 490 | 530 | .0755 |
| Testing | 38 | 492 | 530 | .0717 |
| Total | 358 | 4940 | 5298 | .0676 |

**Fig 2.2. Classifier Performance on Testing Data (Decreased)**

| **Classifier** | **Description** | **Precision** | **Recall** | **f1** |
| --- | --- | --- | --- | --- |
| SVM | Mdf: 1,  ngram: 1, feat\_numbers: 17000 | 0.4 | 0.05 | 0.0889 |
| Constant | all positive | 0.0717 | 1.0 | 0.1338 |
| Stratified | based on training set class distribution | 0.0556 | 0.0526 | 0.0541 |
| Uniform | generates predictions uniformly at random | 0.0657 | 0.4737 | 0.1154 |

**Fig 2.3. Top 20 Decreased Vectorizer Features by Absolute Value** (Green indicates positive weight)

****

**VII. Analysis**

**Increased**

There is a marked improvement in f1 score for the SVM classifier for increased tweets versus the naive classifiers. Notably, in addition to the highest f1 score, this classifier has the highest precision when compared to all of the naive classifiers, although recall (.4846) is slightly worse than uniform random prediction (.5077). When viewing the top twenty feature weights, there is a remarkable level of correspondence with the list of keywords and phrases originally used to classify tweets as “increased.” Taken with the frequency tables from section IV, this is perhaps more evidence to suggest that the keywords and phrases are a strong indication that a tweet is a behavior tweet.

**Decreased**

For decreased tweets, however, the f1 score was lower than all three of the naive classifiers. Though there is a higher level of precision (.4) than the naive classifiers, the recall is significantly worse than uniform random prediction (.4737). However, this may be able to be attributed to the relatively low rate of positives within the entire data set, and thus, within each category of separated data (see Fig. 2.2). Future work to improve this classifier might involve using SMOTE (Synthetic Minority Over-sampling Technique) oversampling or random sampling of a negative class in the training data to remedy this problem, treating different parameters of these methods as hyperparameters for the classifier.

Some of the top 20 features are surprising (“dkdfohj2zj”, “jax”), and suggest the training data and/or the classifier may not generalize well to other data. Some features, like “drinking”, show the model has a perhaps undesirable overlap with the increased classifier, despite the fact “drink” and “drinking” were verified as being excluded from the decreased training data. This may be due to the fact that tweets related to going to bars, cafes, and restaurants were labeled as positive for decreased data; this may merit additional investigation.

Additionally, the term “hiking” raises some potential red flags, as it is perhaps not a good indication of a less safe behavior. Though in this research hiking was initially thought to be a less advisable activity, the safety of hiking could be argued to be highly dependent on where the person is (wilderness versus a park), what case numbers and community spread is in the area, and what ICU occupancy is in that area. Arguably, a term like this that is not “cut and dry” should be excluded from this classifier.

**VIII. Future work**

Once problems with the decreased classifier are remedied, unlabeled decreased and increased tweets from each state can be classified using the appropriate classifier. Next, additional filtering should take place in order to properly assign users to a state, using location data included in each tweet object.

Plotting tweet frequencies after these two processes are finished would yield a more accurate picture of self-reported user behaviors related to Covid-19. This analysis would be informed by an understanding of the performance of the classifier, the limitation of a small amount of labels done by a single annotator, and the fact that the tweets contain a limited amount of keywords and phrases that do not fully account for all possible safer and less safe behaviors.

Just as further improvements in the accuracy of tweet data can be made, there is also potential to increase the resolution of the data describing state-level restrictions in this research. The Oxford Covid Government Response Tracker seeks to rate restrictions not based on the binary state of being “opened” or “closed”, but based on scores rating economic support provided to citizens and stringency of lockdown policies (broken down by types of restrictions). These two data sources, examined together, may provide grounds for useful analysis, with a potential to uncover connections between policy making, safer and less safe self-reported behaviors, and growth in COVID-19 cases and deaths.

1. Note: The keywords and phrases used are detailed in random\_decreased\_labeling/freq\_table/freq\_table\_dec.csv and random\_increased\_labeling/freq\_table/freq\_table\_inc.csv. This is excluded here due to the length of these tables. [↑](#footnote-ref-0)