Using Machine Intelligence to Understand Human Emotional Trends during COVID-19

Randy Truong, Andrew Yin

December 7, 2020

Abstract

Social media engagement has increased as a result of Covid-19. Lockdowns, quarantine, and other enforced regulations have hampered the ability to meet in person and have led to more people expressing themselves and interacting with others online. Although many people present a facade online, the emotions behind the content are real and can be qualified, and can reveal insights into how people are actually faring under these unorthodox circumstances. This study looks at millions of tweets put out during the pandemic and the primary emotion associated with each respective tweet. Various tree-based machine learning models are created and optimized and their performances are compared to determine the best model for determining tweet emotions. Of the models evaluated, the model that performed the best was the random forest with 95.09% accuracy at determining the proper emotion label. This research provides insights into how people are faring during the pandemic as well as how we each perceive our respective conditions.

Introduction

In 2001, the World Health Organization published a report that stated that one in four people in the world will be affected by a mental or neurological disorder at some point in their lives [10]. Mental health is becoming an increasingly talked about issue as attitudes towards it improve, but the lingering stigma and misunderstanding still prevent individuals from seeking help from a professional. This dilemma can be further exacerbated by a global pandemic such as the coronavirus disease (COVID-19) currently afflicting the world. Regulations put in place intended to mitigate virus transmission have also led to unwanted outcomes such as unemployment, economic contractions, and general financial instability which disrupt the social ecosystem, cause psychological distress, and potentially trigger or exacerbate mental health issues [11]. Treatment exists, but nearly two-thirds of people with a known mental disorder never seek help from a professional, thus the cycle of neglect and misunderstanding self-perpetuates [10]. The question lies in how can people be identified as needing help, treatment in the absence of an admission, or desire to seek help, and an angle for study is to analyze the emotional patterns of people. However, the straightforward admissions are not always forthcoming, and must be parsed from one's actions and words.

One of the primary mediums for text is social media, and with the rise of platforms like Twitter and Facebook, it has become easier to broadcast thoughts and emotions to a large audience at any given moment. Covid-19 regulations such as quarantine or lockdown have driven more people online whether it be for work or entertainment; within the online sphere, social media engagement has also increased. The focus of this exploration is Twitter, which has documented a strong growth in new and returning users. Twitter's 2020 Q3 earnings report and shareholder letter revealed that there was a 29% increase in monetizable daily active users (mDAU) year over year and in Q2 experienced a 34% increase in mDAU [16]. This growth and overall increased engagement contribute to a diverse ecosystem of thoughts and emotions that can reveal key insights into how individuals, communities, and populations mentally cope with Covid-19.

The analysis of emotions falls under the larger field of sentiment analysis, whereby instead of categorizing the emotion, as its name suggests, the sentiment behind the text is

analyzed. Sentiment analysis (or opinion mining) is the practice of applying natural language processing, computational linguistics, and other text analysis techniques to identify and extract subjective information from text [9]. Emotion analysis is more specific than sentiment analysis, whose classification focuses on positive, negative, or neutral. Intensity of an emotion is a feature within a text that is loosely defined as the degree to which an emotion is experienced: the power of an emotion [15]. The definition varies from researcher to researcher and the consensus is that too little research has been conducted. The idea of intensity will be further elaborated upon in the discussion of our data.

The focus of machine learning on parsing text and text mining is the advantage of natural language processing techniques to process large quantities of data and convert it into structured data that can be used. Given the amount of unstructured text available, natural language processing can decipher ambiguities, extract relationships, or provide summaries in an unbiased manner. This study seeks to employ tree-based machine learning models to categorize tweets based on primary emotions. The novelty of our approach lies in implementing tree-based models over the typical text-based classifiers because the decision tree acts similarly to the human brain. The human brain's thought processes are highly correlated to the act of traversing a decision tree and arriving at a conclusion [3]. Thus we are able to slightly simulate how others perceive tweets with this approach, and not solely what a pure model would see. A variety of natural language processing techniques are used to process the tweets and models trained on the tweets to extract a number of features. The goal is to conclusively determine which optimized model or models from our set can most accurately predict the primary emotion of a tweet given a range of emotional intensity values. First, we examine related research performed on emotion and sentiment analysis (Background), then juxtapose it with our intended methodologies and experimental procedure (Methodologies), and then report our results (Experiment/Results). Finally, we summarize and interpret the key findings of our research (Discussion).

Background

Significant research in the field of sentiment and emotion analysis and tweet parsing already exists. Gupta et al. presents a large annotated dataset on public expressions related to the Covid-19 pandemic [7]. They utilized Latent Dirichlet Allocation (LDA) on millions of tweets to generate a series of subtopics related to Covid-19 and used five different machine learning algorithms and an in-house Emotion Intensity lexicon to extract intensity measures and assign a sentiment and emotion label.

Bali et al. utilized Convolutional Neural Networks (CNN) trained on word vectors for sentence classification [2]. They implemented three variants of CNN to classify the emotions of 5600 Arabic tweets and compared the performance of the CNN variants with three other "traditional" machine learning algorithms. They found that the best CNN variant was able to achieve 99% on both training and validation accuracies.

Li et al. performed similar research to Gupta et al. by creating an annotated Covid-19 tweet dataset and then training two models based on a multilingual BERT model [12]. Despite only testing English, they predict strong results for tweets in up to 104 languages. They then performed additional trend analysis on the present emotions over a period of two weeks in order to explore high correlations between the fear and sadness emotions and words and phrases.

Guntuku et al. characterized the contents of tweets to highlight the region-varying effects of Covid-19 [6]. They geolocated the extracted tweets and found relative frequencies of single words and phrases. Mental health estimates were then computed by applying four pre-trained

machine learning models and were compared to estimates from the same period in 2019. They quantified the change in estimates between the years using Cohen's d and graphed the changing frequencies.

Chakraborty et al. explored the increasing popularity of negatively classified tweets during the Covid-19 pandemic and how fact checkers should be applied to social media [5]. They collected the most popular tweets and the most popular retweets from late 2019 to mid 2020 and analyzed the sentiments of both groups. They used 9 different models on the feature vectors created from fuzzy logic and Gaussian membership functions to find that retweeted tweets are overwhelmingly negative in sentiment while regular tweets are positive.

Among the studies looked at, many utilize more complex and varied models, with the most popular being Naive Bayes. In our study we will utilize variations of trees to assist us in classifying the emotions.

Methods

The Methods section will talk about the implemented machine learning models for sentiment and emotional analysis. Extant research has used diverse textual classification methods to evaluate social media sentiment. The focus of this study is to demonstrate how commonly used ML methods can be applied and used to contribute to the classification of emotion rather than the development of contributions to new ML theory or algorithms. The three models tested to perform multilabel classification in this study are the basic decision tree, an AdaBoost decision tree, and a random forest.

The basic decision tree forms the base for the other two algorithms and while simple, it is intuitive and can reveal much about the data. No doubt in class we have covered it much, but the fact that it continuously demands revisiting warrants it a brief description and strengths and weaknesses. The decision tree is a supervised machine learning algorithm that can be used for classification and regression. It results in a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The decision tree is a relatively simple model that can model complex boundaries and does not require significant preprocessing. It is much like a brain in the sense that the brain accepts raw data and is able to make conclusions regardless if data is missing or scarce [3]. The "rules" of the decision tree form the splits along which the tree is constructed, and the best split is determined by either calculating Information Gain or the Gini Index.

Information Gain is based on Entropy, which can be thought of as the impurity or randomness within a dataset. Entropy can be found as follows:

$$Entropy(x) = -\sum (P(x=k) * log_2(P(x=k))$$

Where P(x=k) is the probability that a target feature takes a specific value, k. Information Gain, then is:

$$InformationGain(feature) = Entropy(Dataset) - Entropy(feature)$$

The feature with the largest information gain should be used as the root node to build the tree through recursive partitioning.

The alternate criterion for creating a tree, the Gini Index, is calculated as follows:

$$Gini\ Index = 1 - \sum (P(x=k))^2$$

The feature with the lower Gini index should be chosen for the split. The Gini Index favors larger partitions whereas Information gain favors smaller partitions with distinct values. For this experiment, we test and compare both criteria for potential performance advantages.

As mentioned earlier, decision trees also provide the foundation for more advanced ensemble methods such as bagging, random forests and gradient boosting. AdaBoost is one such algorithm implemented intended to improve the strength of weak classifiers (basic decision trees) by combining them into a single strong classifier. AdaBoost "learns" from misclassification errors of the previous model at each iteration. AdaBoost works as follows:
Given a dataset containing n points, where

$$x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}.$$

First, initialize the weights for each data point as equal weights:

$$w(x_i, y_i) = \frac{1}{n}, i = 1, ..., n.$$

Then, for iterations m = 1… M:

Fit weak classifiers to the data set and select the one with the lowest weighted classification error:

$$\epsilon_m = E_{w_m}[1_{y \neq f(x)}]$$

Calculate the weight of the mth weak classifier:

$$\theta_m = \frac{1}{2} ln(\frac{1 - \epsilon_m}{\epsilon_m}).$$

Update the weight for each data point:

$$w_{m+1}(x_i, y_i) = \frac{w_m(x_i, y_i) exp[-\theta_m y_i f_m(x_i)]}{Z_m},$$

Where Z sub m is a normalization factor. The final equation for classification is:

$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x)),$$

Where f_m represents the mth weak classifier and θ_m is the corresponding weight.

Another ensemble algorithm we tested was bagging in the form of the random forest. The random forest improves on regular decision trees by having less bias and moderate variance through creating trees via bootstrapped data. The final predictions for unseen samples x' can be made by averaging the predictions from all the individual trees on x', denoted below:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

Experiment/Results

The dataset we used for the experiment was the result of work done by Gupta et al as listed in the References section [8]. They collected raw tweets from Twitter's standard search API using keywords "corona", "wuhan", and "nCov". They preprocessed each raw tweet by converting them to ASCII characters, removing accented characters, forming bigrams and trigrams, filtering out stop words, and performing text tokenization. After forming a bag-of-words corpus with the 75.6 million processed tweets, they sampled 1% and trained a latent dirichlet allocation (LDA) model to extract the top 10 most representative topic clusters [4]. Using their trained LDA model, they tagged tweets with corresponding topic clusters depending on which topics the tweet content was most relevant to (tweets can be relevant to multiple topics).

Example tweet text	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
Remember when doja cat said corona was just the flu	1	0	0	0	0	0	0	0	0	0

Figure 1: snapshot example of the derived binarization of related topics from the dataset [8].

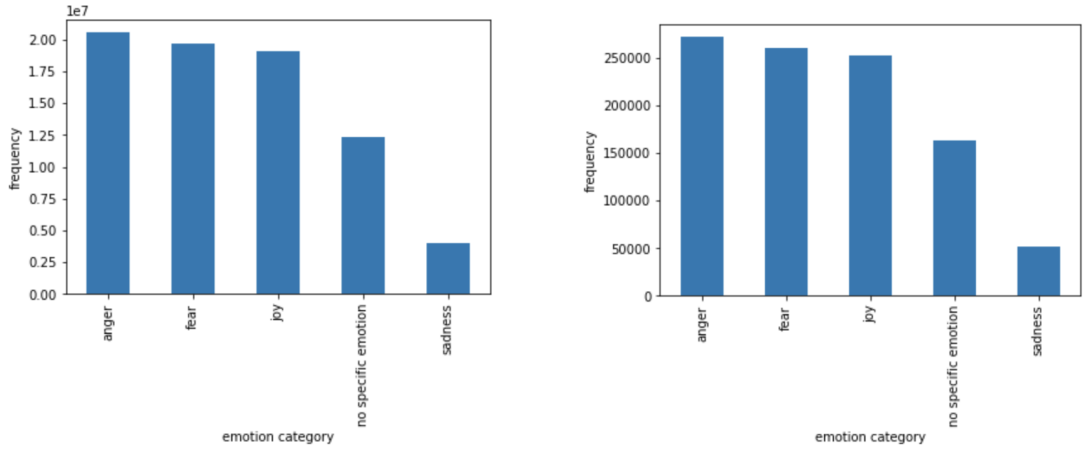
Another set of features were parsed using CrystalFeel, a collection of five machine learning algorithms to calculate emotion intensity from tweets [9]. The CrystalFeel algorithm extracts the intensities associated with each emotion using a proprietary emotional intensity lexicon and assigns a value typically between 0 and 1 [9]. Some scores were greater than 1, representing extreme cases.

Example tweet text	valence_ intensity	anger_ intensity	fear_ intensity	sadness_ intensity	joy_ intensity
Community hospital Bright Vision transfers all patients to make room for stable COVID-19 cases	0.505	0.391	0.444	0.423	0.334

Figure 2: snapshot example of the emotional intensity feature derivation from the dataset [8].

Finally, utilizing the values calculated for each intensity, they assigned sentiment and emotion labels (sentiment_category, emotion_category, respectively) to tweets depending on the greatest intensity values. The resulting feature set contained 18 features: *'tweet_ID*', '*t1*', '*t2*', *'t3*', '*t4*', '*t5*', '*t6*', '*t7*', '*t8*', '*t9*', '*t10*', '*valence_intensity*', '*anger_intensity*', '*fear_intensity*', '*sadness_intensity*', '*joy_intensity*', *'keyword_used*', and '*user_ID*'. The dataset contained two label classes: ''*sentiment_category*' and '*emotion_category*'. Our project focused on the *emotion_category* as our primary label.

Before preprocessing our data, we reduced our large dataset to a smaller, more manageable size. Since our original dataset contained 75.6 million records of processed tweet data, we took a random sample of one million records. We decided that this was an appropriate number because we would still have a large amount of data to infer and predict the emotion labels from the features. To assure that the sample represented the original data, we plotted frequency distributions for both datasets as follows:



(a) Original dataset (b) Sample

Figure 3: Emotion label distribution for the original dataset (a, left) and our sample (b, right).

From the right figure, we found that the overall label distribution aligned closely with the original dataset's distribution in the left figure, thus validating our use of this sample.

Next, we performed feature selection to remove unnecessary and redundant features. We dropped the 'tweet_ID', 'keyword_used', and 'user_ID' features because they did not provide any important information aside from identification purposes. Using the Pearson correlation coefficient metric, we computed pairwise correlation across the remaining features and plotted the following heatmap.

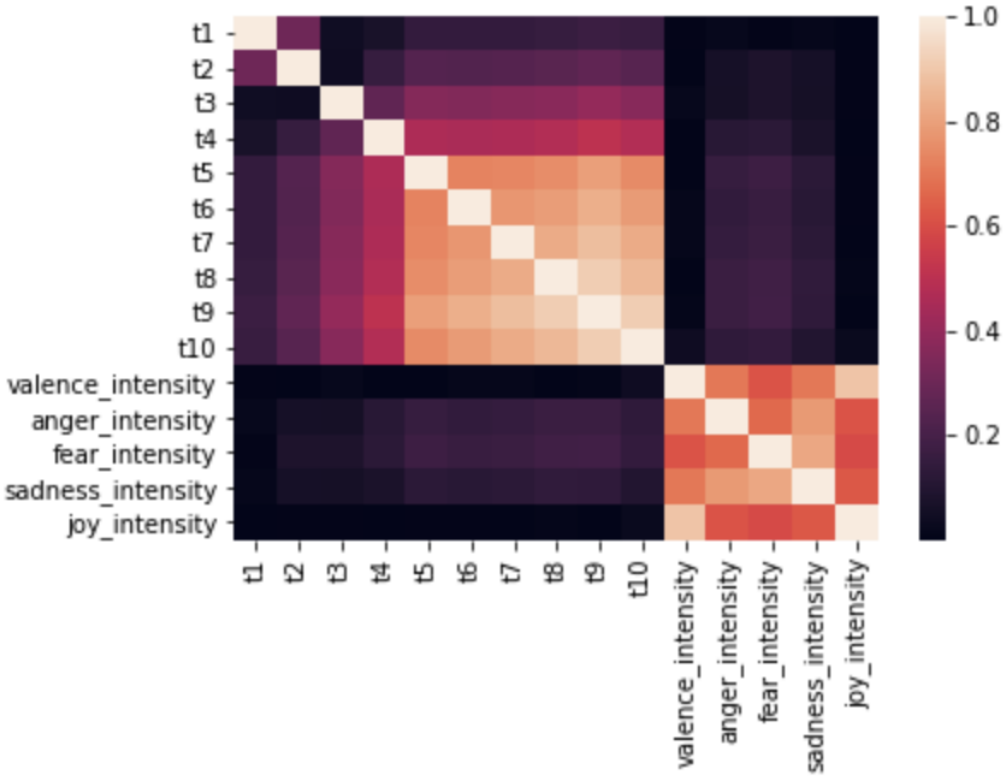


Figure 4: Pearson correlation coefficient heatmap

As highly correlated features introduce redundancy and noise, we dropped the features with correlation coefficients greater than a 0.80 threshold. This selection process yielded our final feature set: *'t1', 't2', 't3', 't4', 't5', 't6', 't7', 'valence_intensity', 'anger_intensity', 'fear_intensity'*. We reduced the total number of features from 18 to 10.

Further, we separated the dataset into respective feature and label data. After shuffling the data accordingly, we performed a simple hold-out strategy to further split the data into training and testing sets, using a 70-30 train-test size ratio. With StandardScaler() from the sklearn.preprocessing package [1], the training data was transformed and standardized which was also applied to the testing data. The training data was split again using a 0.80-0.20 train-test ratio hold-out method to create our validation sets for later hyperparameter tuning. By this standard, we strictly reserved different sets for their intended purposes: training, validation, and testing. From the same sklearn package [1], we used LabelEncoder() to encode our multi-class string labels ('anger' 'fear' 'joy' 'no specific emotion' 'sadness') into respective integers (0, 1, 2, 3, 4).

Given the nature of our labelled data, we focused on evaluating decision tree models and respective ensemble methods, such as Random Forest and AdaBoosted Decision Trees. We hypertuned our parameters using GridSearchCV() from the sklearn.model_selection package [1]. Although we took a smaller sample from the original dataset, our dataset was still large. We attempted to use a 5-fold or a 10-fold cross validation approach, but it was very computationally expensive and unfeasible given our already expensive learning models, thus we opted to use simple hold-out cross validation strategies. As such, we used StratifiedShuffleSplit(), modified to do hold-out, as the cross-validation generator in GridSearchCV() because it used a 5-fold CV by default. Since GridSearch follows an exhaustive approach, we first ran a wide search given broad, large intervals and then narrowed the parameter intervals as optimality approximately converged. The optimal number of trees for the Random Forest and AdaBoost ensemble was determined by plotting the classification error against the number of estimators in each ensemble and selecting its earliest convergence point.

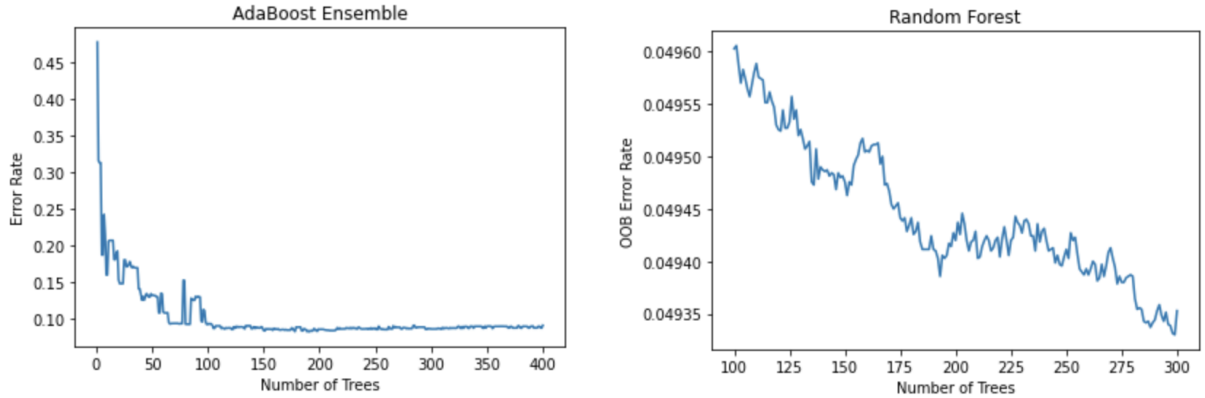


Figure 5: Error Rate with respect to Number of Trees in Ensemble Methods

We found that the models converged to an error rate before the max number of trees was reached. This early stopping helped reduce overall training time and memory usage for the completed model. The tuned hyperparameters for our standard Decision Tree model were *max_depth* = 13, *min_samples_leaf* = 26, and *criterion* = 'entropy'. The tuned hyperparameters for the Random Forest ensemble were *max_depth* = 15, *min_samples_leaf* = 23, *criterion* = 'entropy',

max_features = 5, *n_estimators* = 207. The tuned hyperparameters for AdaBoost ensemble were *max_depth* = 1, *learning_rate* = 1.75, and *n_estimators* = 150, *criterion* = 'entropy'. This hypertuning process was done to balance model complexity and bias-variance trade offs.

Using these hypertuned parameters, we trained each model accordingly. Then, we calculated the accuracies for each model against the testing data using the sklearn.metrics package [1]. In the following figures, we compared the overall accuracies and errors for each model, as well as the individual accuracies for each label.

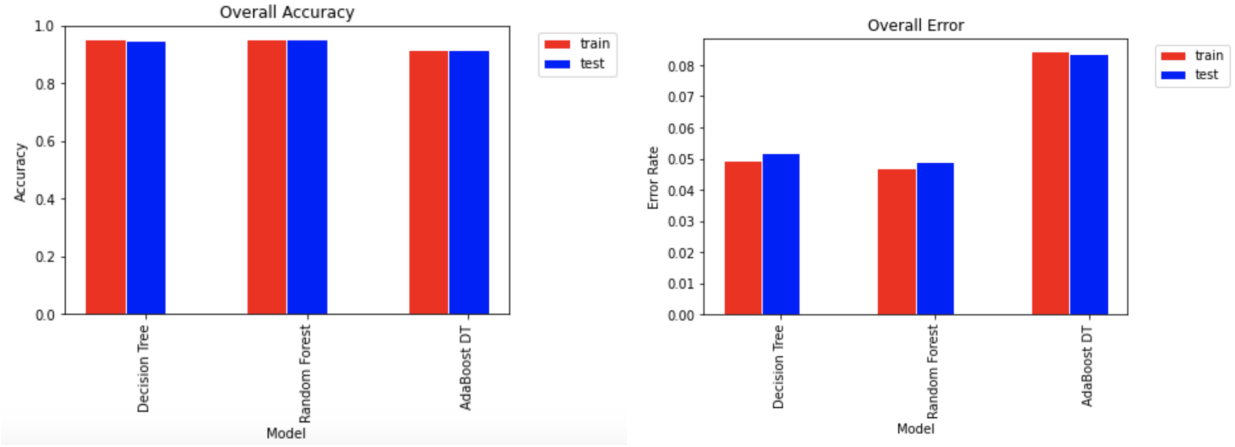


Figure 6: Train and Test Accuracy Scores with respect to Model (DT, RF, AdaBoost)

From the left figure, the Random Forest model had the highest score with a 0.9509 test accuracy and a training time of 295.11 seconds. The Decision Tree model had a 0.9483 test accuracy and a training time of 1.96 seconds. The AdaBoost model had a 0.9162 test accuracy, the lowest overall, and a training time of 103.10 seconds. The error rates were calculated from 1 - accuracy score. From the right figure, the AdaBoost model had the highest test error rate at 0.0838.

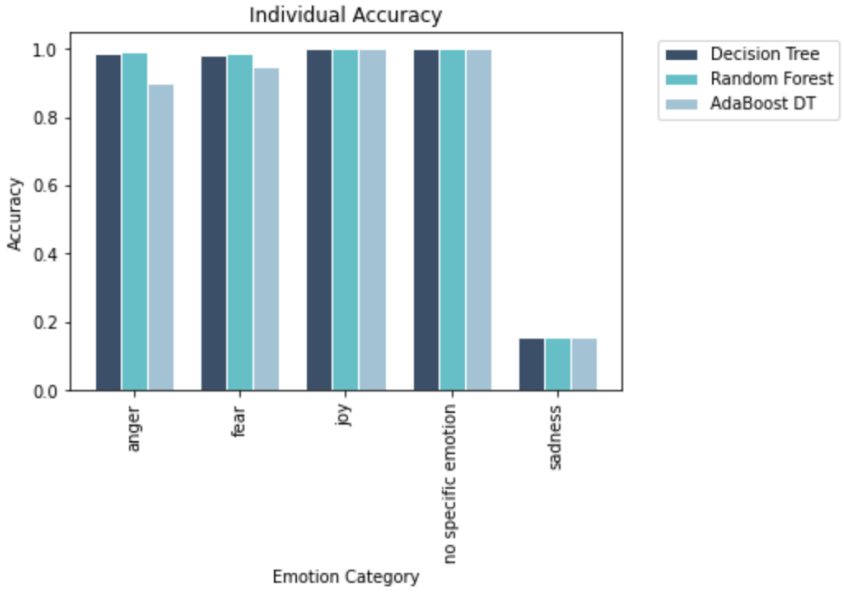
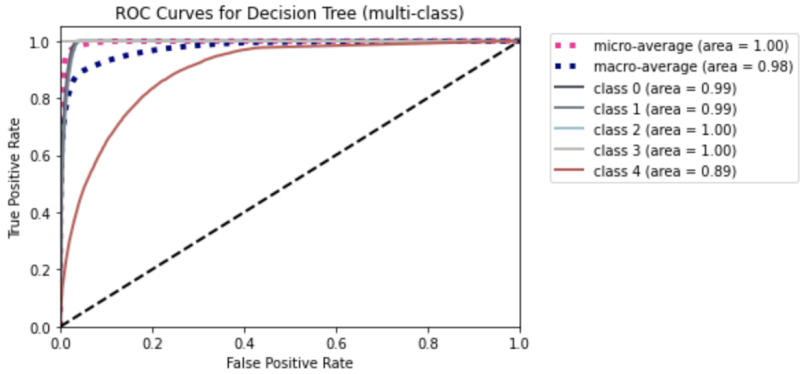


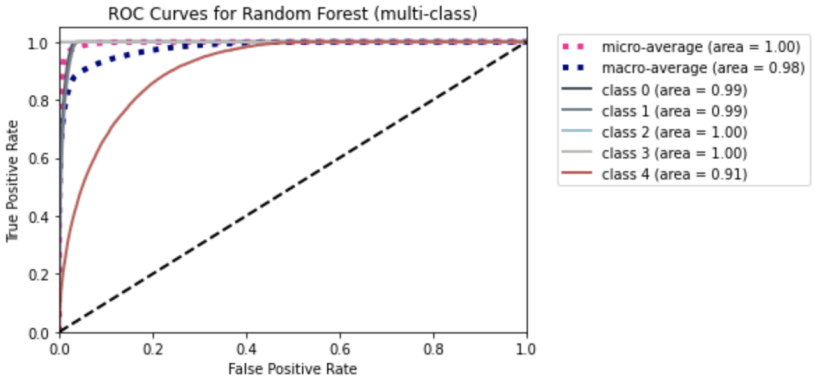
Figure 7: Test Accuracy Scores with respect to Individual Class Label and Model.

For the individual test accuracies for each label in the previous figure, the '*joy*' and '*no specific emotion*' classifications notably had a 1.0 accuracy for all of the models. Contrarily, the '*sadness*' classification had the lowest accuracy (Decision Tree: 0.1539, Random Forest: 0.1543, AdaBoost: 0.1584) out of all of the classes.

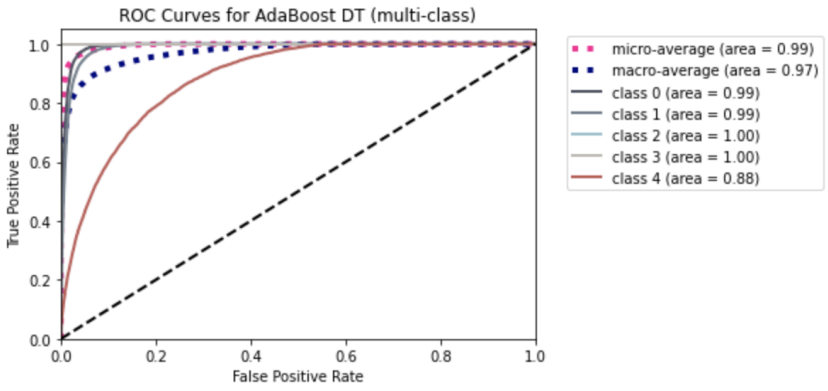
Further, we plotted multi-class ROC curves to evaluate the performance of each model in the following figures. The curves were plotted using the predicted probabilities of each label. Since our data utilized multi-class labels, we used a One-vs-the-rest strategy for mapping appropriate ROC curves for each class; this involved binarizing the classes and comparing each class with the rest [1]. As introduced previously, the classes ('anger' 'fear' 'joy' 'no specific emotion' 'sadness') are encoded (0, 1, 2, 3, 4, respectively).



(a)



(b)



(c)

Figure 8: Receiver operating characteristic (ROC) Curves with respect to each Model and Class

Overall, the ROC curves for our models had area-under-curves (AUCs) greater than 0.85, thus our models effectively classified more true positives and less false positives. The 'anger', 'fear', 'joy', and 'no specific emotion' classes had the highest AUCs. The 'joy' and 'no specific emotion' classes had 1.0 The 'sadness' class had the lowest AUC but still had a high overall AUC.

Discussion

Overall, our models yielded high accuracies for classifying the emotion labels given the feature set and human annotated labelling of the tweets. All of the models had marginal differences between train and test accuracies, thus they fitted and generalized the data well. Although the Random Forest model had the highest accuracy at 0.9509, it took the longest to train and predict at 265.34 seconds. Compared to the Decision Tree, which had a 0.9483 accuracy and a run time of 1.96 seconds, we found that the tradeoffs between higher accuracy and higher time complexity were significantly different because the Random Forest model had a marginal improvement in accuracy but a much longer run time. Thus the Decision Tree was the most optimal model empirically; however, given that Decision Trees overfitted to the data, this simple, individual model would not fare well when fed other samples of data, such as extending it to the size of the original dataset. Further, the AdaBoost ensemble had the lowest test accuracy of 0.9162 but a middleground run time at 103.10 seconds. When focusing on the individual label test accuracies, the AdaBoost model had lower accuracies for anger and fear classifications compared to the Random Forest and Decision Tree models, but it had the highest accuracy for the sad label classification. AdaBoost's attempt to balance and maximize the accuracies for the labels was notable and may require further research. Although the Random Forest model took the longest, the model offered low bias, moderate variance, and reduced overfitting by considering the results of multiple individual trees. Given our large dataset, this high run time is expected and feasible, thus our Random Forest model had the overall best performance. In general, these ensembling strategies fare well in real-world applications, especially in the context of emotion classification where different people express a wide range of different emotions.

Additionally, Gupta et al.'s research utilized the CrystalFeels API for determining the emotional intensity (anger, fear, joy, etc.) features. This API had predicted accuracies of 0.814, 0.775, 0.766, 0.791, and 0.860 for the anger, fear, sadness, joy, and valence (no specific emotion) intensities, respectively [9]. Notably, the sadness intensity had the lowest accuracy out of all of

their predictions. This paralleled our results when focusing on individual test accuracies because the sad label test accuracy was around 0.155 for all models; however, as illustrated in our ROC curve plots, the sad label classification fared well when compared to the other labels in a one-vs-the-rest manner with AUCs of 0.89, 0.91, and 0.88 (DT, RF, AB, respectively). The classifiers effectively predicted more true positives. Even when considering the class imbalance, the micro-average AUC remained high between 0.99 and 1.00. During our Preprocessing steps, we interestingly dropped the '*sadness_intensity*' and '*joy_intensity*' features due to information redundancy and noise. We inferred that we did not need to capture the whole range of emotions to make effective classifications. For instance, if a tweet did not have a highly negative sentiment, then it may have a positive or neutral sentiment in contrast. Given the results of our models and the CrystalFeels API's moderate accuracies, the API provides notable thresholds for helping classify emotions from tweet-derived data, such that high intensity accuracies may not be required for high label classification accuracy.

Some degree of error was expected given the nature of our project's theme. Even in understanding human dynamics, humans are fallible and many times cannot fully recognize or understand their own emotions, let alone another person's emotions. As such, this natural error leaks into machine intelligence. One limitation to our project that may have heavily influenced the previous sad accuracy measurement was our initial data sampling. Although our sample followed a similar label distribution as the original dataset, both the sample and original dataset had significantly fewer records with sad-annotated labels, compared to the other labels. This skew may have played a role in outputting a lower test accuracy because the models had overall less data to learn from for the sad labels. Since we only took a one million sample out of the 75.6 million tweets, a future exploration may entail oversampling the sad label to create a uniform class distribution. Further, we considered removing entries by duplicate users to minimize potential spam cases and limit the model from fitting to the noise of a user's text and expression style, but we could not find a feasible way; if we limited users to only one tweet in the dataset, then a user would only have one emotion represented for the entirety of the COVID-19 pandemic, which is unrealistic and inorganic. However, our sample contained tweets by unique users in over three-fourths of the data, thus it was acceptably distributed. Further, given our large dataset and use of computationally expensive classifiers, we had to be cognizant of inefficient computation and tradeoffs. As we increased the number of trees in both ensemble methods, the computational complexity increased significantly. As such, hyperparameter tuning was a strenuous portion of our project. We utilized a top-down approach of determining the general interval for parameters before finetuning them. Although this facilitated our process, we could not feasibly exhaust through a wide range of parameters with depth, thus potentially missing more optimal parameters; however, our parameters were ones that approximately converged on relatively low error rates. Another limitation was that our project was shaped by our inability to fetch related tweet data due to Twitter API call limits, thus we could not perform our own text processing and sentiment analysis.

Our project was interesting because of the novelty of and growing interest in affective computing and sentiment analysis. Given the multiplicity and variety of human expression (culturally, socially, physically), it is difficult to establish a general framework for how emotions evolve, thus machine learning tools are important for analyzing patterns and making predictions. Much feature engineering is required for determining how to represent more abstract concepts, such as emotions, into concrete data that a machine can understand and learn from. Instead of relying only on emotional intensities derived from different text patterns and keywords, the data

that we used included associated topics with each tweet. This is important because context matters in interpreting emotions. In further dissecting our research, other explorations include analyzing the limits of 280-character tweets as efficient and sufficient mediums for expressing emotion, or predicting the most prominent emotion in the COVID-19 time frame (i.e. is anger the most profound in the beginning, middle, or end of the pandemic?). This can be applied to future pandemic cases and used to support epidemiological works in exercising appropriate public health interventions and preventative measures.

Contributions

Andrew wrote the Abstract, Introduction, Background, and Methodologies. He also wrote part of the data section under Experiment/Results. He helped prepare the presentations as well and presented the spotlight in class.

Randy outlined much of the introduction in the project proposal and helped prepare the presentations. He mostly worked on the Experiment/Results, Discussion, and Code sections of the paper. He proposed doing this project's theme/topic.

Code

Our code can be found here:
https://drive.google.com/drive/folders/1YzCI7uOqHgfOmZ0pwl-S80LyQgM0Iq7z?usp=sharing
The dataset referenced [6] can be found here: https://doi.org/10.3886/E120321

References

[1] Sklearn. https://scikit-learn.org.

[2] M. Baali and N. Ghneim, "Emotion analysis of Arabic tweets using deep learning approach," *J. Big Data*, 2019, doi: 10.1186/s40537-019-0252-x.

[3] N. S. Bastos, D. F. Adamatti, and C. Z. Billa, "Discovering Patterns in Brain Signals Using Decision Trees," *Comput. Intell. Neurosci.*, 2016, doi: 10.1155/2016/6391807.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, 2003, doi: 10.1016/b978-0-12-411519-4.00006-9.

[5] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, "Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers—A study to show how popularity is affecting accuracy in social media," *Appl. Soft Comput. J.*, 2020, doi: 10.1016/j.asoc.2020.106754.

[6] S. C. Guntuku *et al.*, "Tracking Mental Health and Symptom Mentions on Twitter During COVID-19," *Journal of General Internal Medicine*. 2020, doi: 10.1007/s11606-020-05988-8.

[7] R.K. Gupta, A. Vishwanath, and Y. Yang. “COVID-19 Twitter Dataset with Latent Topics, Sentiments and Emotions Attributes,” *arXiv*. 2020.

[8] R.K. Gupta, A. Vishwanath, and Y. Yang. “COVID-19 Twitter Dataset with Latent Topics, Sentiments and Emotions Attributes,” 2020, doi:10.3886/E120321.

[9] R. K. Gupta and Y. Yang, “CrystalFeel at SemEval-2018 Task 1: Understanding and Detecting Emotion Intensity using Affective Lexicons,” 2018, doi: 10.18653/v1/s18-1038.

[10] G. Hartl, “Mental disorders affect one in four people,” *World Health Organization*, 2001.

[11] G. Hartl, “Mental health in emergencies,” *World Health Organization*, 2019.

[12] I. Li, Y. Li, T. Li, S. Alvarez-Napagao, and D. Garcia, “What are we depressed about when we talk about covid19: mental health analysis on tweets using natural language processing,” *arXiv*. 2020.

[13] I. Nadjenkoska, F. Stojanovska, S. Gievska, “Detecting emotions in tweets based on hybrid approach,” *CiiT*., 2018.

[14] J. Samuel, G. G. M. N. Ali, M. M. Rahman, E. Esawi, and Y. Samuel, “COVID-19 public sentiment insights and machine learning for tweets classification,” *Inf.*, 2020, doi: 10.3390/info11060314.

[15] V. Shuman, D. Sander, and K. R. Scherer, “Levels of valence,” *Front. Psychol.*, 2013, doi: 10.3389/fpsyg.2013.00261.

[16] C. Valuenzela, F. Giovanna. “Q3 2020 Shareholder Letter”. *Twitter*, 2020.