Machine Learning for Mental Health Screening

A Major Qualifying Project submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degrees of Bachelor of Science in Computer Science and Data Science.

Submitted By:

Connor Bruneau

Hunter Caouette

Rimsha Kayastha

Veronica Melican

Miranda Reisch

Advised By:

Professor Elke Rundensteiner

April 6, 2021

Abstract

Mental health disorders such as depression are prevalent in both the United States and the world. Left untreated, such conditions can greatly decrease the quality of one's life and even lead to suicide. Therefore, accurate screening methods for mental health are a necessity. Surveys are commonly used but can be biased and perceived as intrusive, so there is a need for passive screening methods. This project builds on three previous years of MQP research that aimed to develop passive mental health screening methods. We made improvements to the Android and website surveys developed by previous teams. In addition, we collected two new datasets: one to investigate how students are affected by depression and another that aimed to answer remaining research questions about the mobile application survey in order to improve it. We refined the existing machine learning pipeline to increase efficiency and usability. Finally, we investigated the potential of using time series constructed from text and call logs to predict depression. Overall, this work contributed to the development of non-intrusive passive mental health screening methods that will facilitate faster diagnosis and treatment for those affected.

Acknowledgements

There are a number of people without whom this project would not have been possible. We would first and foremost like to thank our advisor, Professor Rundensteiner. Over the course of this year she has been a guiding force for us, shaping the development of the project while allowing us the individual freedom to make our own decisions.

ML Tlachac provided invaluable oversight, coordination, and assistance throughout the entire duration of the project. They helped to not only continue the progress of previous MQP teams, but also to define and refine our specific goals for this project. They also helped coordinate team members to ensure those goals were met. ML was able to provide helpful direction during all stages of the project, from survey development to data collection and data analysis. This team would not have been able to accomplish nearly as much without them.

Additionally, Ermal Toto helped provide our team access to the data collection servers, as well as provided many hours of assistance for data collection, python programs, SQL and Postgres queries, feature extraction, and server guidance and maintenance. James Kingsley at the Academic Technology Center helped us get set up with Ace, WPI's high performance computing cluster, and troubleshoot any problems that we encountered.

We would furthermore like to thank the university, Worcester Polytechnic Institute, with special thanks to the Departments of Computer Science and Data Science. They have provided us with ample support and resources for our work over the course of this year.

We would also like to thank the professors that provided additional insight and guidance for our research. Professor Emdad helped provide the team with information regarding statistical significance and how to calculate how many participants were needed for our studies to be statistically significant. Professor Harrison shared our survey with many students, both graduates and undergrads, helped with stereotype threat research and provided statistical methods for analyzing AB testing.

Finally, we would like to thank the previous MQP and research teams who have contributed to this project, Ricardo Flores, Tabassum Kakar, and all the professors who shared our surveys with their students.

Contents

List of Figures

List of Tables

1 Introduction

Depression is a mood disorder characterized by symptoms such as a persistent sad mood, lack of interest in activities, and a decrease in energy. In 2019, it was estimated that 20.6% of adults in the United States experienced some form of mental illness such as depression (*Key Substance Use and Mental Health Indicators in the United States: Results from the 2019 National Survey on Drug Use and Health*, 2020). One group that is particularly affected by depression is college students. College students face many stressors that can lead to mental illness such as increased academic pressure, being away from home for the first time, and increased adult-like responsibilities (Pedrelli, Nyer, Yeung, Zulauf, & Wilens, 2015). A 2015 study estimated that between 7% and 9% of college students are affected by depression (Pedrelli et al., 2015). Some college students deny that they are experiencing depressive symptoms or do not have access to adequate treatment, which may exacerbate the problem.

Left untreated, depression can significantly lower one's quality of life and even lead to suicide. Due to this high impact, accurate screening metrics for depression are a necessity. Commonly, depression is screened for via the Patient Health Questionnaire (PHQ-9), a nine-question survey that asks people to rank how frequently they experience some common symptoms of depression. However, surveys such as the PHQ-9 can be intrusive and biased, as patients may not want to answer the questions honestly. Therefore, there is a need for accurate passive screening methods for mental illness.

This project builds on the work of previous teams, which trained machine learning models to detect depression in text and audio inputs. This involved the development and deployment of a survey application to collect data from users. The surveys included text and audio prompts, demographic information, and the user's PHQ-9 responses, all of which were stored in databases. The data was later passed into a machine learning pipeline, which extracted features from the data that were then used to train machine learning models. From this previous work our team wished to not only refine and expand the data collection process but also add new features to the machine learning pipeline. Additional studies and data analysis were also conducted.

This project expanded on previous MQP teams' work in the following ways:

- Updated the survey to increase completed modalities.
- Created multiple survey applications for various data collections.
- Refined the machine learning pipeline to improve usability.
- Collected a new dataset from college students.
- Conducted experiments to see if time series constructed from text and call logs can predict depression.
- Updated the Android mobile application to experiment with triggering Stereotype Threat.

2 Background

2.1 Related Works and Datasets

Previously conducted studies and collected datasets were analyzed to prepare for this project. Additionally, prior research on mental health was studied.

Modma Dataset ModMa is an open source depression screening dataset that includes EEG and audio data from clinically depressed patients. This data has been used to train machine learning models to detect depression using data from depressed patients as a training platform (Cai et al., 2019).

Student Life Dataset from Dartmouth This was a similar dataset collected by Dartmouth College. It contains 53 GB of continuously collected data, 32,000 self-reports, and pre and post surveys from 48 undergraduate students. The categories and some modalities can be seen below (not all data labels are included) (*StudentLife Survey*, n.d.).

Table 1: Student life dataset.

Objective Sensing Data	1. sleep (bedtime, duration, wake up) 2. conversation duration 3. conversation frequency 4. physical activity (stationary, walk, run)
Location-Based Data	1. location 2. co-location 3. indoor and outdoor mobility
Other Phone Data	1. Bluetooth, Wi-Fi, and audio 2. light and screen lock/unlock 3. phone charge and app usage

Self Reports	1. stress, mood and loneliness 2. behavior 3. class opinions and cancelled classes 4. events 5. exercise 6. social and study spaces
Pre-Post Surveys	1. PHQ-9 depression scale 2. UCLA loneliness scale 3. positive and negative affect schedule (PANAS) 4. perceived stress scale (PSS) 5. big five personality 6. flourishing scale 7. Pittsburgh sleep quality index
Academic Performance	1. class information 2. deadlines 3. grades (grades, term GPA, cumulative GPA)
Dining Data	1. meals data 2. location and time 3. seating data

Detecting depression and mental illness on social media: an integrative review This study found that symptoms of mental illnesses can be observed on social media, such as Twitter and Instagram, via different features. This study identified mentally ill social media users using screening surveys, their public sharing of a diagnosis on Twitter, or memberships in online forums. Automated detection methods were also used to distinguish depressed users from control users by their language patterns and online activity. (Guntuku,

Yaden, Kern, Ungar, & Eichstaedt, 2017)

2.2 Previous MQPs

Previous MQP teams deployed mobile applications that were used to collect user data including texts, vocal samples, social media information, and geographical data. The data would then be processed through a machine learning pipeline in order to evaluate the state of the user's mental health. These applications have allowed for medical professionals to gather information from patients in a quick and efficient manner. Through crowd sourcing and later the surveying of students the application was tested to determine the effectiveness of not only gathering data but also the analysis of data. An ultimate goal of this data analysis was to train machine learning models to predict a given participant's score in the PHQ-9, a depression module of a diagnostic instrument used to detect common mental disorders. With an average test set root-mean square error (RMSE) of 5.67 in predicting the PHQ-9 score, this machine learning was concluded to be an intuitive way to diagnose depression (Ball, Dogrucu, Isaro, & Perucic, 2018).

Initially, prior teams ran a study to determine how willing participants would be in sharing various data with a medical staff member. This study concluded that the majority of participants would be willing to share microphone data, such as a recording of their voice, as well as images of their face. Meanwhile, only about 40%-50% of participants stated that they would be willing to share other types of data such as text, GPS information, call logs, browser history, etc. As a result most of this information, while being collected by the application, was not initially factored into the machine learning pipeline. From here an Android application was developed to gather this information from participants phones, with text and audio samples being collected along with participants phone numbers, social media information, and more. From this data multiple features were extracted, including the number of syllables from text messages and speaking rates from audio files.

In addition to the collection of this information the application also required users to fill out a questionnaire. This questionnaire, known as the PHQ-9, is a multipurpose tool used in the medical field for screening, monitoring, and measuring the severity of depression. Users of the application were asked to answer each question of the PHQ-9, and the scores of all questions were combined to form a PHQ-9 score for each participant. Cutoff points for mild, moderate, moderately severe, and severe depression were formed at 5, 10, 15, and 20 respectively. Through this PHQ-9 form information can be gathered on the individual symptoms of each participant. This data could then be used to train and test the accuracy of the machine learning systems. The 2018 team, for instance, used 85% of the data as a training set for the model, while 15% was reserved to check the accuracy of the system overall (Ball et al., 2018).

DAIC-WOZ The name DAIC-WOZ stands for Distress Analysis Interview Corpus, Wizard of Oz. The DAIC is a series of clinical interviews which served to

diagnose depression, anxiety, PTSD, and other disorders. A virtual interviewer named Ellie interviewed participants and the data was stored in the DAIC-WOZ database. The dataset contains 189 interviews consisting of audio recordings, transcripts, and facial data from the interviews.

The previous MQP team used the F1 scores and results of studies utilizing the DAIC-WOZ dataset to validate the results of their study and make comparisons based on machine learning algorithms used on the DAIC-WOZ dataset (Caltaiano et al., 2020).

TwitterPul This independent dataset is utilized for analyzing the Twitter feature extraction data. The dataset contains data directly pulled from Twitter and will be used to analyze activity on Twitter relative to mental health.

2.2.1 Datasets

The datasets described below were created by previous MQP teams. This project expanded on the EMU and EMU Summer datasets, which were used for analysis and future changes to the survey applications throughout our team's project.

Moodable Moodable is a dataset that was started by the 2018 MQP team. They distributed the Moodable application on the Amazon Mechanical Turk platform. The app collected user data such as texts, social media content, geospatial data, and voice samples up to two weeks from the data gathering date. There are over 300 entries from crowd-sourced participants (Assam, Flannery, Resom, & Wu, 2019).

EMU This dataset was gathered by the 2019 MQP team. The new application, EMU, was deployed on the Amazon Mechanical Turk platform once again. Similar to the Moodable dataset, the EMU dataset includes audio, text messages, social media and GPS data from the user. It also includes survey data, which collects the GAD-7 (Anxiety) and PHQ-9 (Depression) scores of the crowd-sourced users. There are over 60 valid entries in this dataset (Assam et al., 2019).

EMU Summer The EMU Summer dataset is a subset of the EMU dataset and includes survey data from the Summer 2020 gathering period.

There were a few specific areas of the MQP targeted for improvement by the previous teams. These included the number of complete data sets acquired through the applications, as well as the machine learning systems themselves. Previous teams expressed the need for more data to be gathered in order to properly train the machine learning systems (Assam et al., 2019), with great emphasis put toward the potential for future studies to utilize the collection of more modalities, including call logs, GPS information, and more (Caltaiano et al., 2020).

2.3 Screening Surveys

2.3.1 PHQ-9 (Patient Health Questionnaire - 9)

Many tools and scales have been developed over the years to help with the diagnosis and analysis of depression. Among them, the PHQ-9 is one of the most commonly used tools. The PHQ=9 helps clinicians monitor the severity of patients' depression and their response to treatment. Specifically, the study is using the Major Depressive Disorder (MDD) module of the full PHQ survey. Figure 1 shows the PHQ-9 questionnaire.

		Not at all	Several days	More than half the days	Nearly every day
1.	Little interest or pleasure in doing things	0	1	2	3
2.	Feeling down, depressed, or hopeless	0	1	2	3
3.	Trouble falling or staying asleep, or sleeping too much	0	1	2	3
4.	Feeling tired or having little energy	0	1	2	3
5.	Poor appetite or overeating	0	1	2	3
6.	Feeling bad about yourself — or that you are a failure or have let yourself or your family down	0	1	2	3
7.	Trouble concentrating on things, such as reading the newspaper or watching television	0	1	2	3
8.	Moving or speaking so slowly that other people could have noticed? Or the opposite — being so fidgety or restless that you have been moving around a lot more than usual	0	1	2	3
9.	Thoughts that you would be better off dead or of hurting yourself in some way	0	1	2	3

Figure 1: PHQ-9 questionnaire (*Depression — PHQ-9*, n.d.).

2.3.2 GAD-7 (General Anxiety Disorder - 7)

Similarly to the PHQ-9 questionnaire, the GAD-7 is used to help diagnose patients. However, the GAD-7 is used to help diagnose anxiety instead of depression. Figure 2 shows the GAD-7 questionnaire with questions that are to be answered based on the patient's experiences over the past two weeks.

		Not at all	Several days	More than half the days	Nearly every day
1.	Feeling nervous, anxious or on edge	0	1	2	3
2.	Not being able to stop or control worrying	0	1	2	3
3.	Worrying too much about different things	0	1	2	3
4.	Trouble relaxing	0	1	2	3
5.	Being so restless that it is hard to sit still	0	1	2	3
6.	Becoming easily annoyed or irritable	0	1	2	3
7.	Feeling afraid as if something awful might happen	0	1	2	3

Figure 2: GAD-7 questionnaire.

2.4 Machine Learning

Machine learning is a branch of artificial intelligence involving algorithms that search for patterns in data in order to be able to make predictions (*Machine Learning*, n.d.). Machine learning algorithms are trained on data examples, or instances, so that they assign the correct output to each instance. The desired output is also called the target. The data instances are made up of features that describe the data. For example, in a model predicting house price, the features may be the age of the house, its size, location, etc. In what is called supervised learning, the target labels are known. The machine learning model can improve its performance and "learn" based on whether or not it predicts the correct label.

Two main types of machine learning problems are regression and classification. In regression problems, a numerical value is the target. For example, the model may be predicting the price of a house or a person's age. In classification problems, the target consists of categorical labels, such as "cat", "dog", "bird" or "true", "false". In binary classification, there are two target labels.

When training machine learning models, it is common to split the data into a training set and a testing set. The model is trained on the training set and evaluated on the testing set. The goal of this split is to see how the model generalizes to new or unseen data.

2.4.1 Data Balancing and Dimensionality Reduction

Downsampling and Upsampling A dataset is imbalanced if there is significantly more of one target class than another. In binary classification, the label for which there are more examples is called the majority class, and the other is called the minority class. If a dataset is imbalanced, the model may be biased and predict the majority class more often. Therefore, it is often necessary to

balance the training data before using it to train a classifier (*Imbalanced Data*, 2020).

There are two common ways to balance data: downsampling and upsampling. In downsampling, data samples from the majority class are randomly dropped so that the number of instances in the majority class matches the number in the minority class. In upsampling, data samples from the minority class are resampled (duplicated) so that there are the same number of instance in the majority class as in the minority class.

Principal Component Analysis Principal Component Analysis (PCA) is a dimensionality-reduction technique which reduces the number of features in a dataset while still retaining information from the original set of features. PCA reduces the number of features by identifying features that are highly correlated and may have redundant information. PCA creates principal components that are linear combinations of the original features (Jaadi, 2019).

2.4.2 Machine Learning Methods

k-Nearest Neighbors k-Nearest Neighbors (kNN) is a method that predicts the value of a new instance based on the values of the k closest instances, or neighbors, in the training set. For classification problems, kNN will assign the most common class across the k nearest neighbors to the new instance. For regression problems, kNN will assign the average value of the k nearest neighbors to the new instance. K is a user-defined parameter that determines how many closest neighbors to consider when making a prediction (Brownlee, 2016a).

Figure 3 illustrates how kNN works. In this example, when k = 3, the model will assign a class to the new instance by picking the most common class across the three closest neighbors, which is Class B.

Support Vector Machines Support Vector Machine (SVM) is a method that searches for an N-dimensional hyperplane, where N is the number of features, that separates instances by class. This method aims to find the hyperplane with the maximum margin, meaning that the maximum distance between instances of different classes is achieved (Gandhi, 2018).

Figure 4 shows hyperplanes for different values of N. When N = 2, the hyperplane is a line and when N = 3 the hyperplane is a plane.

Logistic Regression Logistic regression is commonly used for classification tasks. Logistic regression uses the logistic (or sigmoid) function to predict the probability that an instance belongs to a given class. Figure 5 shows the sigmoid function (Brownlee, 2016b). A higher value of the sigmoid function indicates that an instance is more likely to belong to a given class.

Decision Trees and Random Forests Decision tree algorithms work by iteratively splitting the training set into separate sets on the feature that best

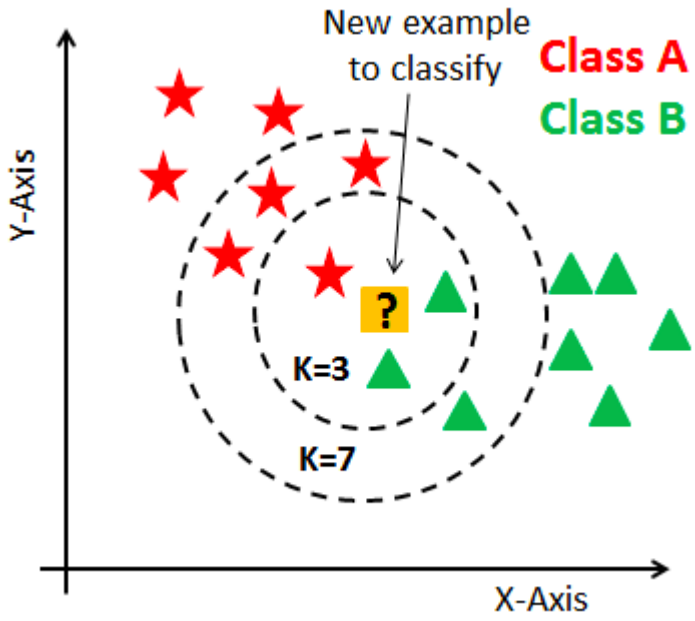

Figure 3: Example demonstrating kNN algorithm (Rohit Madan, 2019).

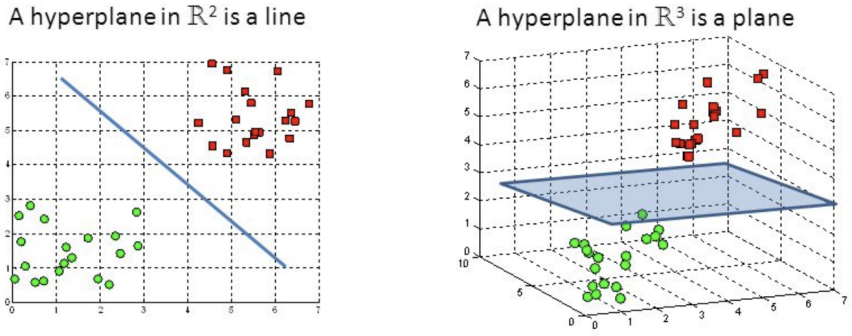

Figure 4: Example showing hyperplanes calculated by SVM algorithm for different numbers of dimensions (Gandhi, 2018).

separates instances of different classes. The algorithm will stop splitting a set either when it is pure (meaning that all instances in that set belong to the same class) or when there are no remaining features to split on (Gupta, 2017).

A sample decision tree is shown in Figure 6. The boxes labelled "Has feathers?", "Can fly?", and "Has finns?" are called nodes, which is where the algo-

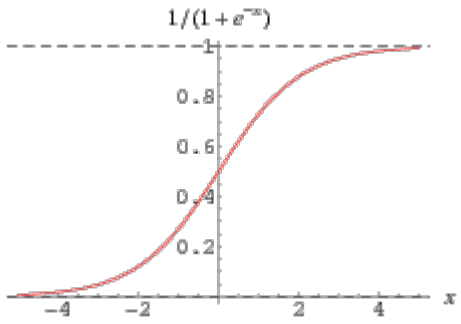

Figure 5: The sigmoid function (Weisstein, n.d.).

rithm splits the sets of instances on a feature. Branches extend from each node; each branch represents a different value of the feature that is being split on.

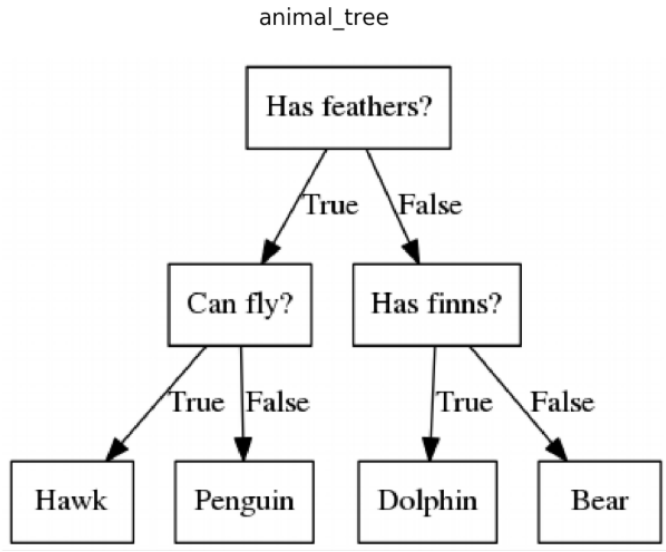

Figure 6: A decision tree (Chowdary, 2020).

Random forests are composed of multiple decision trees. Each decision tree in the random forest is built with a different subset of features. Random forests work by selecting the prediction that is most common across all the individual decision trees for classifiers, or averaging the individual predictions for regressors (Breiman & Cutler, n.d.).

Neural Networks Neural networks are loosely based on brain function and consist of interconnected nodes (modelled after interconnected neurons in the

brain). The nodes in a neural network are organized into layers. Connections between nodes in successive layers are called edges, and each edge is assigned a numerical value called a weight. Figure 7 shows a neural network (Hardesty, 2017).

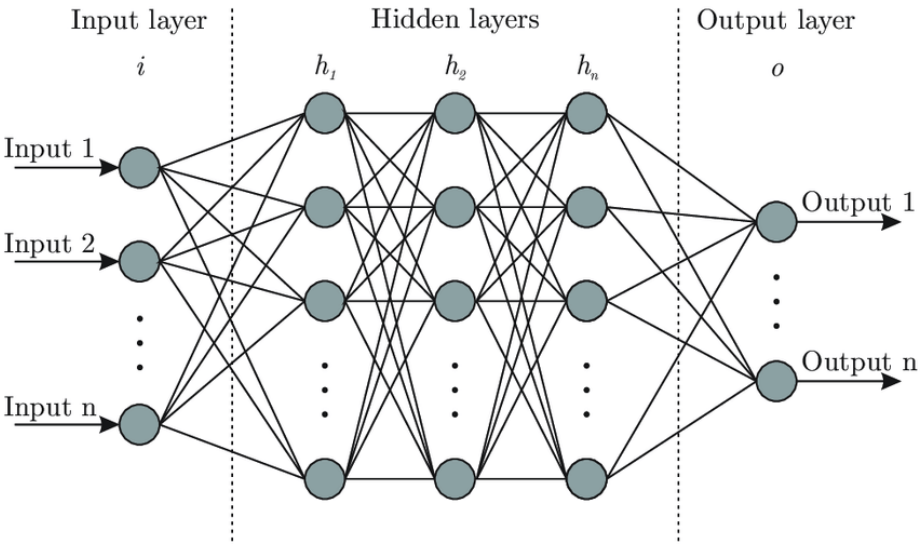

Figure 7: A neural network (Shukla, 2019).

The value of each node in the network (except those in the input layer) is determined by an input function, which is based on some calculation that involves the outputs of the previous layers and the weights that connect the previous layer to to current one. Nodes can also have an activation function, which modifies the values that were calculated by the input function before outputting them. One example of an activation function is the sigmoid function (which is used in logistic regression).

Neural networks iteratively update the weights in the network so the network can "learn" the correct outputs. First, input values are fed through the network to create predictions. Then, some loss function is used to calculate the error between the correct value and the predictions. Next the weights in the network are updated using this loss function in a process called back-propagation. These steps are repeated either for a fixed number of iterations or until the error below a certain value (*A Basic Introduction To Neural Networks*, n.d.).

Boosting Algorithms Boosting algorithms convert weak learners into strong learners, where weak learners are those that perform only slightly better than random chance. Boosting algorithms train a weak learner and identify which instances the learner performed poorly on. The algorithm then trains another weak learner that focuses more on the instances that the previous learner classified incorrectly. This process is repeated iteratively. Two common boosting

algorithms are AdaBoost and XGBoost (Ray, 2015).

2.4.3 Performance Evaluation of Classification Methods

Confusion Matrix A confusion matrix compares a model's predictions to the actual values. Each row and column is labelled with the classes in the dataset. One axis represents the true values, and the other represents the predicted values. For example, in Figure 8, 69 examples were predicted to be 4 (vertical axis) that were actually 4 (horizontal axis). However, 68 examples were predicted to be 4 (vertical axis), but were actually 9 (horizontal axis).

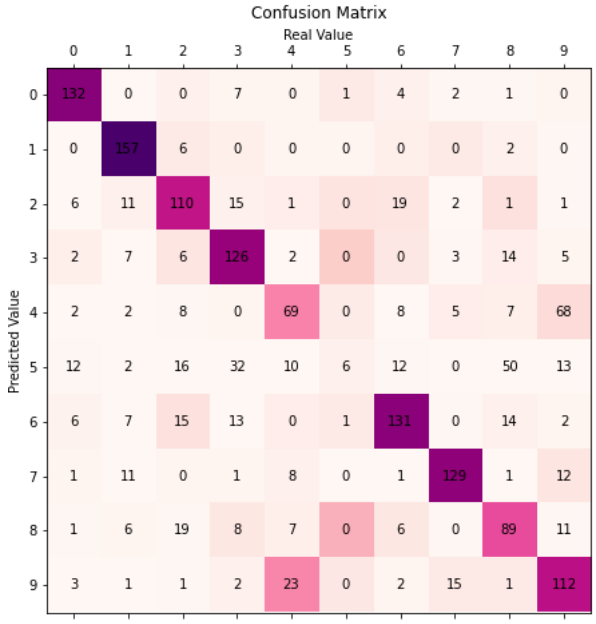

Figure 8: A 10x10 Confusion Matrix for a model trained on the MNIST dataset, which consists of handwritten digits.

In a binary classification, a confusion matrix will take the form of a 2x2 table representing True Positives, False Positives, False Negatives and False Positives. True Positives are values for which a model correctly predicts True, and True Negatives are values that are correctly predicted as False. False Positives are values that are incorrectly predicted as True when they are actually False. False Negatives False values that are incorrectly predicted as True. Figure 9 shows the relationship between these values, where the horizontal axis represents the actual values and the vertical axis represents the predicted values.

Accuracy Accuracy is one of the simplest ways to measure a model's performance. It is the fraction of True Positives and True Negatives over the total

	TRUE	FALSE
TRUE	TP	FP
FALSE	FN	TN

Figure 9: A binary confusion matrix.

number of instances, or the percentage of observations that were correct (Kartik Nighania, 2018). Figure 10 shows the equation to calculate accuracy.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TotalSampleSize}$$

Figure 10: Equation for accuracy.

Precision and Recall Precision measures the percentage of True predictions that were correct. Recall measures the percentage of True predictions that were correctly predicted to be True. Figure 11 shows the equations for these metrics. For example, in Figure 13, 60 examples were predicted as True but only 40 of them were actually True, so the precision is 67%. 40 out of the 50 True examples were correctly predicted as True, so the recall is 80% (Wood, n.d.-b).

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Figure 11: Equation for precision and recall.

Specificity and Sensitivity Sensitivity is equivalent to recall. Specificity measures the percentage of False values that were correctly predicted to be

False. In the binary confusion matrix shown in Figure 13, 30 out of the 50 False examples were correctly classified, so the specificity is 60% (Wikipedia contributors, 2020).

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives}$$

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Figure 12: Equations for specificity and sensitivity.

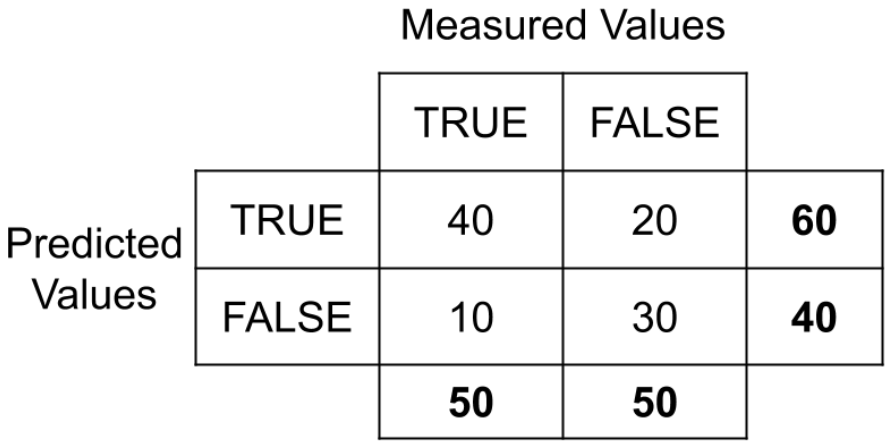

Figure 13: An example confusion matrix of a binary classification.

F1 Score The F1 score of a model is an example of a harmonic mean, which is typically used to calculate the average of a set of rates. As seen in Figure 14, it is expressed as a ratio of precision and recall. A perfect model, with both a precision and recall equal to 1, will have an F1 score of 1.00. Rather than being just balanced between precision and recall, the F1 score can also be adjusted to favor one of the values and is expressed as the F_β score, where the value of β determines the weight of the recall value in the calculation (Wood, n.d.-a).

ROC and AUC A Receiver Operating Characteristic (ROC) curve is a graph that plots a binary classifier model's recall, or True Positive Rate (y-axis), against its False Positive rate (x-axis), at different thresholds. The ROC curve of different models can be compared. After creating an ROC curve, you can

$$F1 = (\frac{precision^{-1} * recall^{-1}}{2})^{-1} = \frac{2}{\frac{1}{precision} * \frac{1}{recall}}$$

$$= 2(\frac{precision * recall}{precision + recall}) = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

$$F_\beta = (1 + \beta^2)\frac{precision * recall}{(\beta^2 * precision) + recall} = \frac{(1 + \beta^2)TP}{(1 + \beta^2)TP + \beta^2 FN + FP}$$

Figure 14: The F1 and F_β Formulas, where TP is the count of True Positives, FP is False Positives, FN is False Negatives and β is the weight of Recall (Wood, n.d.-a).

calculate the AUC value to summarize the model's skill (Brownlee, 2018). The AUC value is based off of the ROC curve, and stands for "**A**rea **U**nder ROC **C**urve (*Classification: ROC Curve and AUC*, n.d.). The AUC value, like the F1 score, is measured from 0 to 1 where a model that is always wrong will measure 0 and a model that is always correct will measure 1

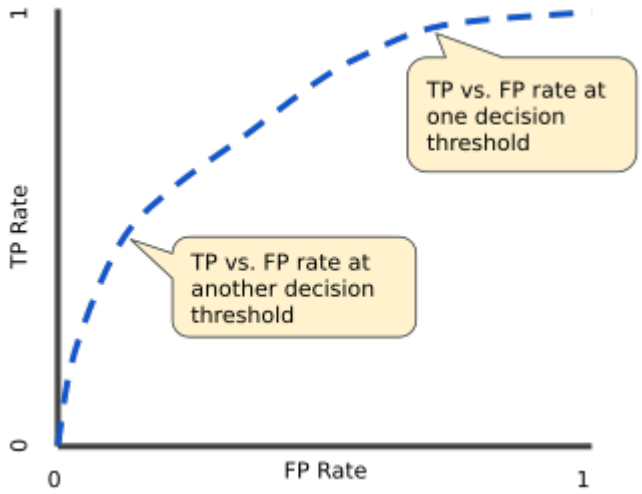

Figure 15: TP versus FP rate at different classification thresholds (*Classification: ROC Curve and AUC*, n.d.).

Precision-Recall Curve The Precision-Recall Curve is a plot of a model's Precision (y-axis) against its Recall (x-axis) at different thresholds. This method of plotting the relation between two related proportions at different thresholds is similar to the ROC Curve (Brownlee, 2018). While the ROC Curve (and

accompanying AUC value) are among the most commonly used metrics for evaluating classification methods, the Precision-Recall Curve can be just as important in certain situations. An article from *PLOS One* in 2015 proposed the idea that the Precision-Recall Curve should be used instead of the ROC curve when using imbalanced datasets (Saito & Rehmsmeier, 2015). Because both Precision and Recall don't rely on True Negatives to get their value, the P-R Curve can be a more useful indicator in situations with a high number of True Negatives.

2.5 Stereotype Threat

Stereotype threat is the possibility of affecting a person's performance by reminding them of a well-known stigma or stereotype. Many studies have been performed to test how people perform on intellectual tests when presented with information that they associate with, such as gender. The main purpose of stereotype threat studies is to determine what mediators trigger stereotype threat, identify which target groups are most effected by stereotype threat and how to run a good priming study with stereotype threat (Pennington et al., 2016).

2.6 Statistical Significance

Overview Statistical Significance is the relevance and accuracy of test results. When gathering data from a population, in order for the results to be used to be able to represent a population, enough data must be collected in order for conclusions to be made. While there are many factors that go into statistical significance, such as diversity in the data and collecting information from different sub-populations, the core of statistical significance is based on Normal Distribution, which is based on the mean and where the data falls underneath the bell curve as can be seen in the graph below (Koehrsen, 2018).

Z-Score We can determine how close the mean a data point is based on how many standard deviations it is from the mean. This becomes useful for statistical significance when we convert the mean and standard deviations to z-score as indicated below. (Koehrsen, 2018).

$$Z = \frac{X - \mu}{\sigma} (Koehrsen, 2018) \tag{1}$$

Sample Size One way to achieve statistical significance is obtaining the right sample size for the target population. The main elements of calculating sample size are confidence level, target population, and margin of error. (*Population Proportion – Sample Size*, n.d.).

$$n = \frac{N * X}{X + N - 1} \tag{2}$$

Table 2

Summary of stereotype threat literature examining mediational variables with key methodologies and findings.

Authors	Hypothesized Mediator	Mediator Method	Dependent Variable	Population	Conditions
Steele & Aronson [3], Experiment 2	Anxiety	State-trait anxiety index	Verbal GRE	20 black and 20 white females	2 conditions: 1 stereotype threa control
Spencer et al. [11], Experiment 3	Evaluation apprehension; Anxiety; Self-efficacy	State-trait anxiety index, evaluation apprehension questionnaire,	Math portion of Graduate Management Test (GMAT)	67 undergraduates (31 male)	2 conditions; 1 stereotype threa 2), control
Aronson et al. [14], Experiment 1	Anxiety; Effort	State-trait anxiety inventory and effort questionnaire	18 (GRE) math questions	23 male undergraduates	2 conditions: 1 stereotype threa 2), control
Aronson et al. [14], Experiment 2	Anxiety; Effort; Evaluation apprehension	State-trait anxiety inventory, Effort and performance expectancies questionnaire	15 GRE math questions	75 white male undergraduates	2 conditions: 1 stereotype threa 2), control

Figure 16: Sample of mediator research (Pennington et al., 2016).

$$where$$

$$X = \frac{Z^2_{\alpha/2} * p * (1-p)}{MOE^2} (PopulationProportion – SampleSize, n.d.)$$

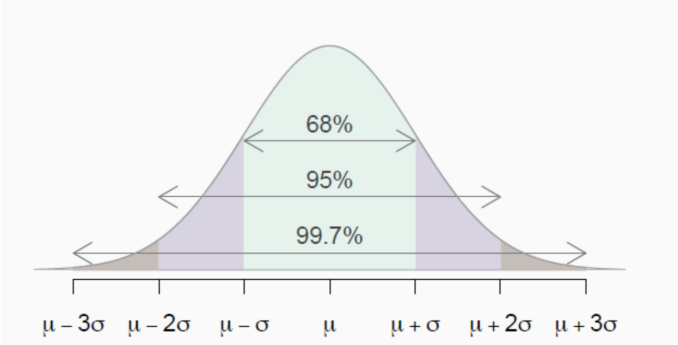

Figure 17: Normal Distribution with mean μ and standard deviation σ. (Koehrsen, 2018)

3 Methodology

3.1 Survey Platforms

3.1.1 Android App

In order to distribute our survey and collect data, an Android application was utilized. This application first asked students to allow the application to take snapshot of their phone's data, including call logs, SMS metadata, and calendar details. The application then required students to answer several survey questions, the responses of which were written to a database for later analysis. Users were then prompted to write a short response after being asked to describe their favorite place. This section was eventually appended to include questions relating to the COVID-19 pandemic, asking users not only if they were working remotely but also if they had previously contracted COVID-19. Finally, users were asked to record audio responses to one open ended question, and one static sentence. This information was also stored and used to train our machine learning models to recognize signs of depression in both text and speech. Lastly the application asked users to allow the application to use Google Maps to access users locations and also prompted users to enter their social media information.

Although an Android application had already been created in order to gather information from students with which to train machine learning models, this application needed to be updated to not only reflect the changes to the data being collected but also to increase the number of fully completed surveys. A new page was added to incorporate the PHQ-9 questions into the survey, in order to analyze how depressed a given user of the application is. This information, like the other answers in the survey, are written to a database to be viewed and analyzed later. While previous iterations of the application wrote all of the information to the EMU database, the current version only saves raffle data to this location, instead having migrated all of its other results to the newly created EMUTIVO database.

Other changes made to the student version of the application were made in an attempt to increase the proportional number of users who completed the entire survey. After analyzing a small amount of preliminary data, it was observed that very few users made it past the recording modalities. As a result, the demographics page was left empty and no data was written to the student server. In order to correct this, the demographics page was moved up in the survey to before the recording modalities, but after the PHQ-9 questions. Additionally, a progress bar was added to inform users of their current progress in the application, in an attempt to increase the number of users who made it through all pages. Finally, a raffle was also organized during data collection in order to prompt more users to take the survey. For every page of the application the users complete, more tickets will be entered into the raffle under their name.

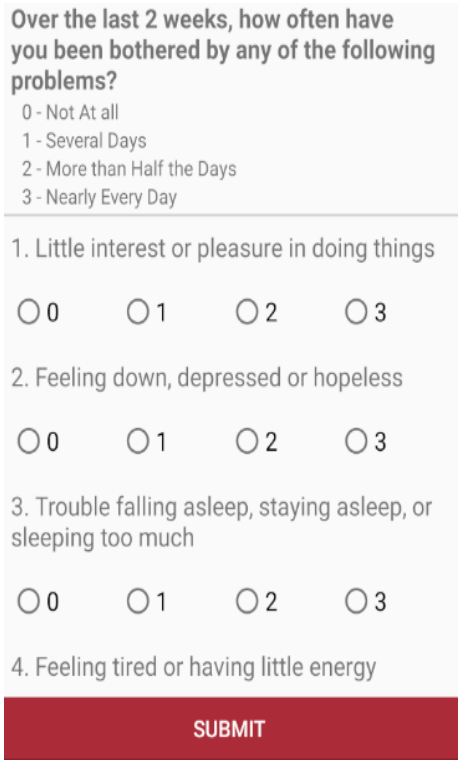

Figure 18: PHQ-9 page in student survey application.

Figure 19: Progress bar in student survey application.

Android App Performance Analysis SQL was used to extract data from the datasets. Preliminary analysis was performed on the EMU dataset in order to determine the effectiveness of the current application format. Queries were used to filter the data by survey question to understand why some users did not complete the survey.

Excel graphing functions were then used to graph the application performance data as shown in the graphs below. From these graphs, we determined that many users left the survey at the open audio question, as it had a nearly 50 percent drop-off rate.

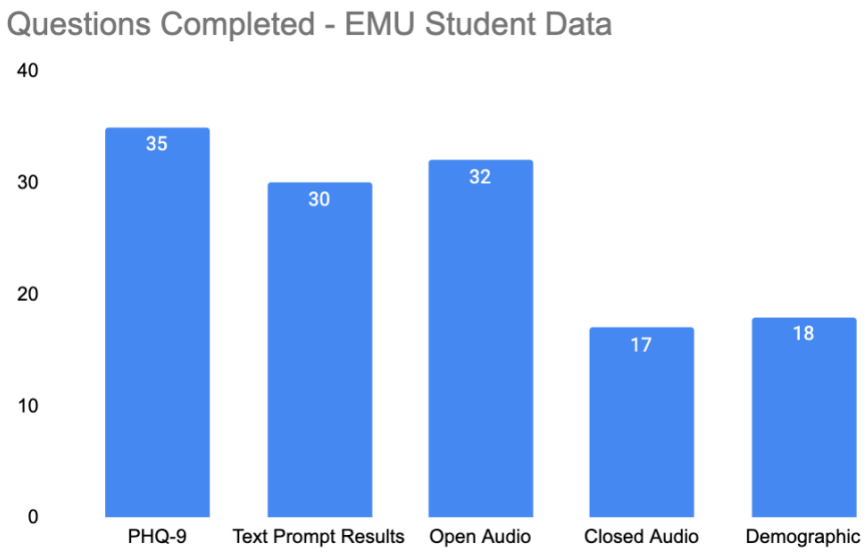

Figure 20: Number of questions completed on EMU student survey.

3.1.2 Stereotype Threat Changes

General Structure Due to the nature of this experiment and the fact that we want to compare how people who see the fact fill out the PHQ-9 and GAD-7 questionnaires compared to those that do not, this study implemented control and a test version that was assigned to each participant at random using a random integer variable.

Common Changes Some common changes to both versions of the app from our previous student version are the removal of the text prompt, the addition of the GAD-7 questionnaire, changes to the audio prompts, a Prolific survey completion code at the end of the survey so participants can be compensated, and inclusive options in the student and Covid-19 demographic questions since not all participants were students and we wanted to make sure there was an option that would apply to everyone. We also ran experiments with different versions of the text message sharing page.

Prolific ID Entry After the first study and determining that the prolific ID should be entered earlier in the survey in order to ensure there were matches with the participants and the entries in the database, the Prolific ID submission was moved from the completion page to the text data preference page as can be seen below in the Text Message Collection section.

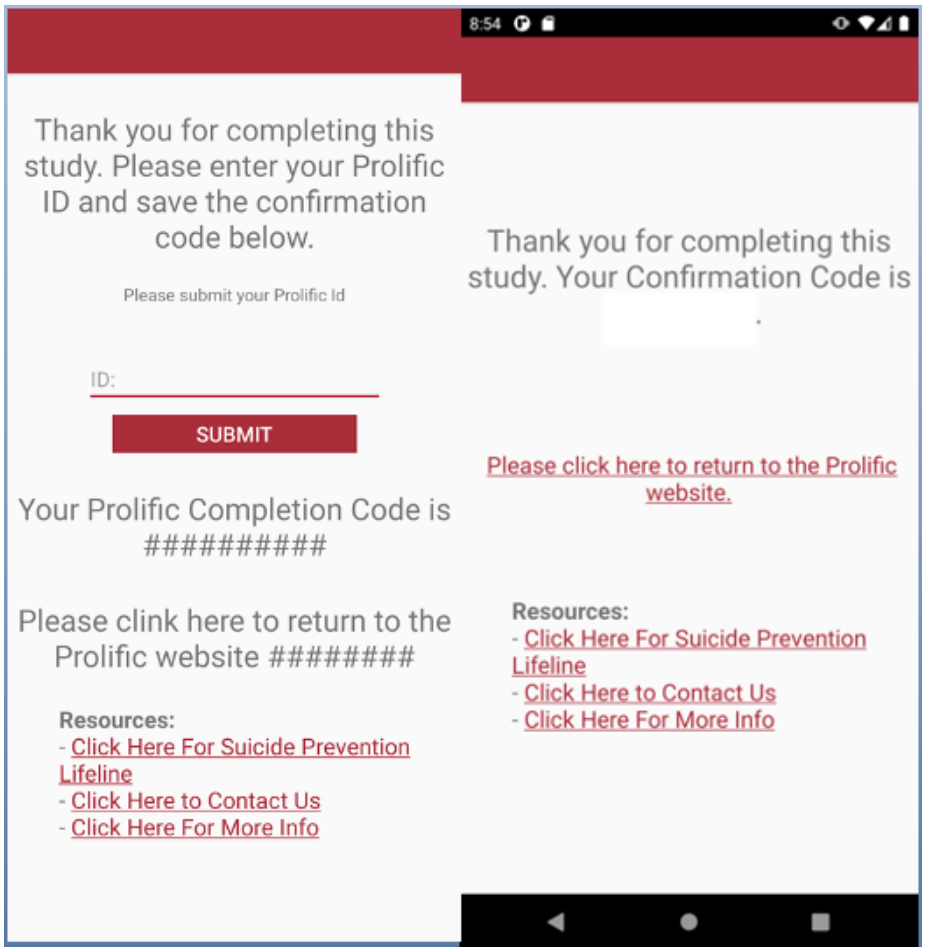

Figure 21: Completion pages with and without Prolific ID entry.

Control Version The control version of the Android application had a structure most similar to the original student version of the app with the changes listed above. The structure for this version of the application was as follows.

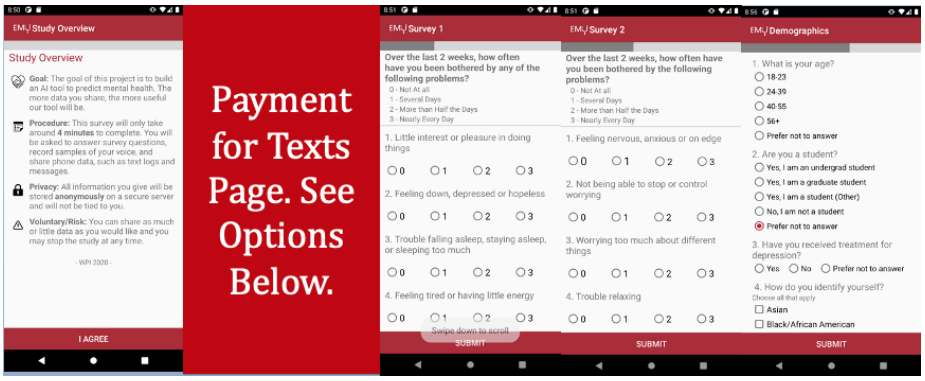

Figure 22: Control version of app: pages 1 - 5.

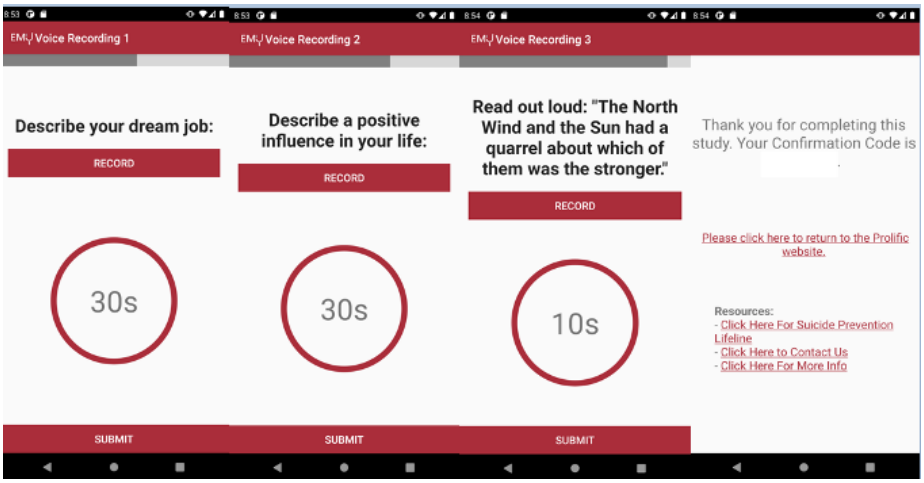

Figure 23: Control version of app: pages 6 - 9.

Test (Stereotype Threat) Version The test version had the most changes, as we introduced the fact page. We also moved the gender question from the demographics page to the end of the fact page. We decided on this approach in order to have the gender question in-between the fact and PHQ-9 questionnaire - namely to keep it fresh in the participants mind and encourage them to think about the fact and how they fit into the stereotype. We then presented the participants with the PHQ-9 and GAD-7 questionnaires, followed by the modified demographics page (without the gender question) and finally the other modality pages of the survey. The pages from this version of the app can be found below.

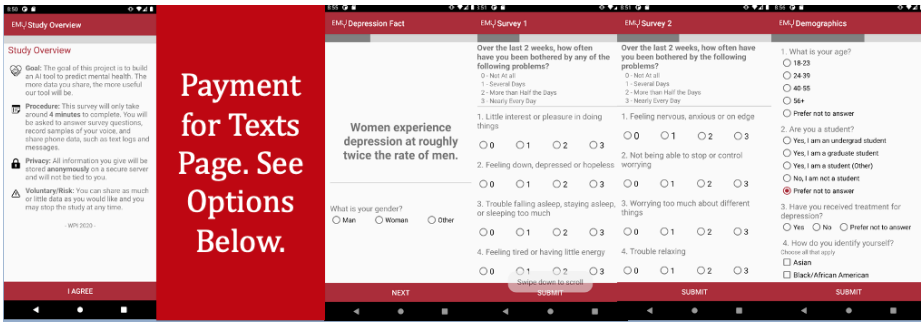

Figure 24: Stereotype threat version of app: pages 1 - 6.

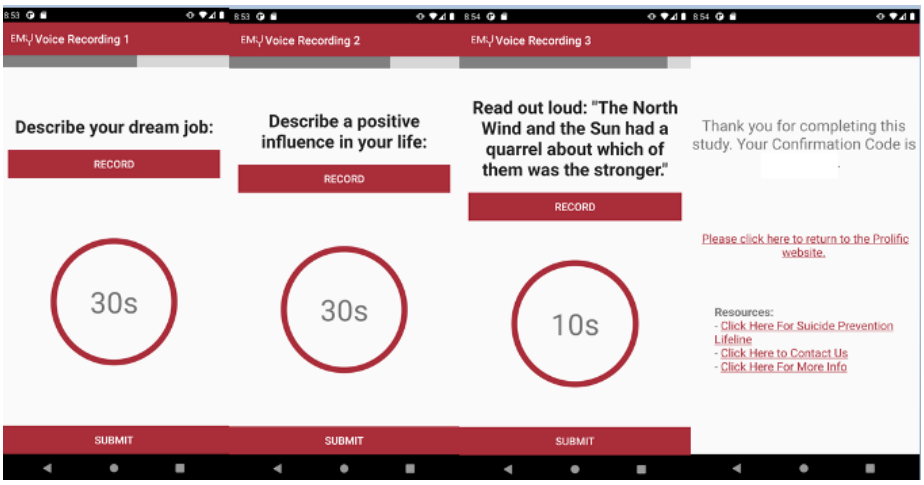

Figure 25: Stereotype threat version of app: pages 7 - 10.

Text Message Collection The initial test runs prompted participants to select to share no text data, text logs, or text content. After determining to only offer incentives for text content and wanting to only use the permissions to track whether participants shared content to reduce confusion between the content page and the Android permissions page, we changed the page format. The new page had less text and increased the font size of the payment options. Participants were given one of the payment options at random via a random number generator. The different pages can be seen below.

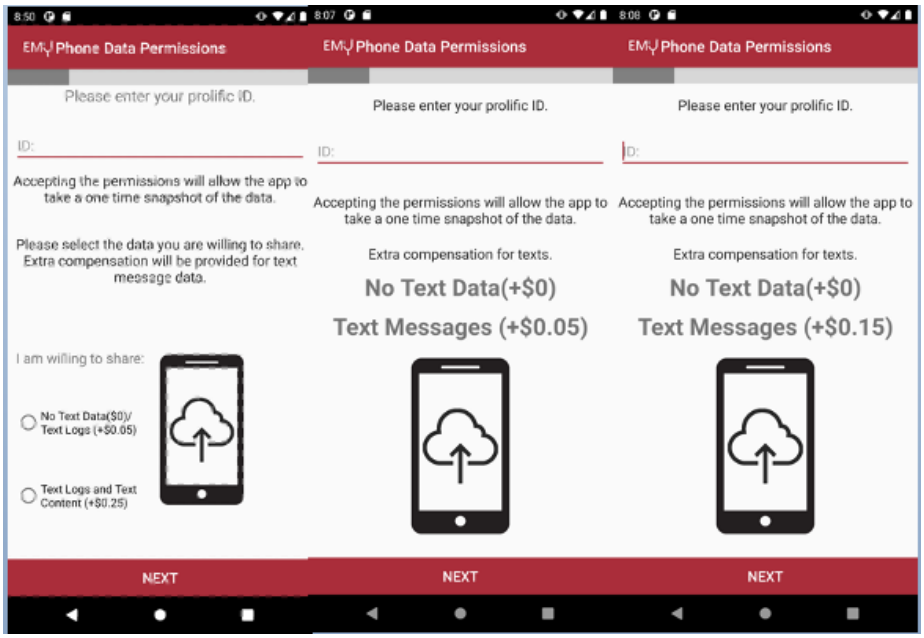

Figure 26: Text message incentive: original, $0.05, $0.15.

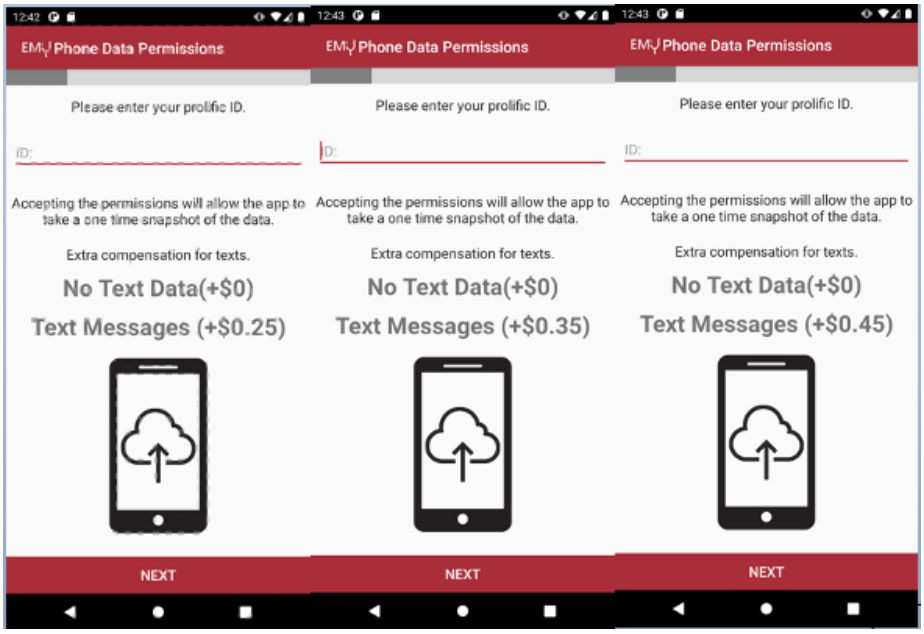

Figure 27: Text message incentive: $0.25, $0.35, $0.45.

3.1.3 Website Survey

The demographics' results from Dartmouth's StudentLife (Wang, Chen, et al., 2017), a research study with the targeted participants similar to ours, showed a 60/40 split of iPhone versus Android users in their study group. Mobile applications were the only platform originally used to distribute our survey and, additionally, there was the possibility of students not wanting to download the app. This limited the amount of data we were able to collect and, hence, led us to the creation of a new survey, similar to the mobile survey, on the EMUTIVO website over the Summer.

The website survey mimics most of the EMU application's modalities, which were originally selected after running an investigative study that explored which data modalities people would willingly share with us for a previous MQP group's study. This refers to the collection of the participant's data from a PHQ-9 survey, demographics, text and audio prompts, and Twitter tweets.

Web App Similar to of the android app survey, we also created four versions of the website survey: Test version, Student data version, and two longitudinal versions. While the test version is available to everyone and completely void of incentives for completion, the student version is accessible to students and give them the option of entering into a raffle system and the longitudinal version is accessible to patients enrolled in an affiliated longitudinal program. We made changes, such as adding appropriate error and help messages, to the problems observed during a minor data collection to better the user experience.

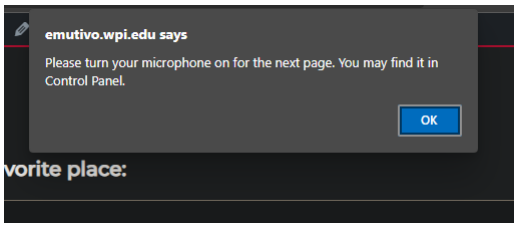

Figure 28: A pop up box for a reminder to turn the microphone on for the audio modality.

Website Changes Since our website app is made to be accessed through our Emutivo website, we decided it was important to make the website more accessible and easy to use as a whole. This involved beginning to apply Nielsen's Heuristics of User Experience design onto our website appropriately and making sure it showcases our research well and is user-friendly at the same time. To achieve this, we enlarged the font size for all text content, refined the menu tabs, and added a color theme that we followed throughout the website. Additionally, we added an EMU Tools page that showcases descriptions and screenshots of

both our mobile application and website survey, with a direct link to the website survey prominent at the end.

3.1.4 Student Survey Changes

A study at Indiana University shows that the presence of a progress bar in a survey generally results in increases the survey completion rate (Yentes, Toaddy, Thompson, Gissel, & Stoughton, 2012). With that information and the analysis of completion rates from our past data collections, we changed the survey to be a simplistic window located within a web page with a progress bar on top that shows the number of modalities the participant has gone past. Fig. 29 and Fig. 30 shows the survey with and without the progress bar changes.

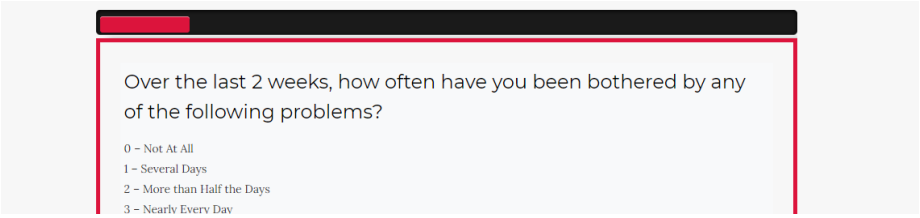

Figure 29: Old progress bar (without labels).

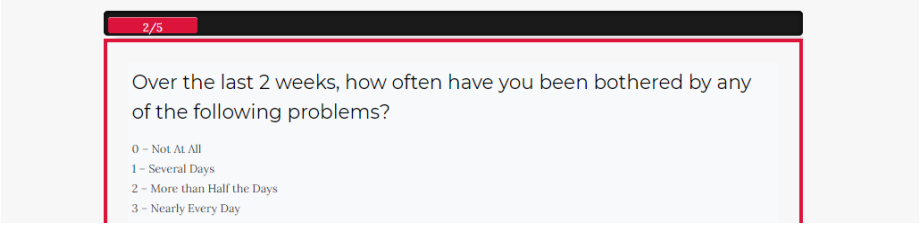

Figure 30: New progress bar (with labels).

We conducted Think Aloud sessions, a method used to gather data in usability testing in product design and development, to test how the surveyors comprehend the survey modalities. Another change we made to increase the completion rate of the survey is, we changed the order of the modalities so that the demographics page followed immediately after the PHQ-9 page as those didn't include thought provoking questions. Aside from that, in order to distinguish our data from the app and that from the website, a column in the student IDs database, 'paid', was re-purposed so that the value for paid is changed to 10 if the website was used.

3.1.5 Longitudinal Survey Changes

Several changes were also made to the student website and application to create a longitudinal version of the survey for passive data collection. In order to

accommodate multiple versions of the application, a new database entry was created to store the version number. Most changes were made in response to requirements suggested by our medical collaborator. For instance, users will be able to select whether or not they wish to share all text messages with the application, or simply text message metadata.

Additionally, the longitudinal survey does not ask users to answer questions regarding their age, gender, or other demographic information, as this information will be provided by medical collaborators. Users will, however, be asked to answer a new question, a depression scale provided by our medical collaborator. This information along with users' other answers and data will be written to the medical server to store the information for later use.

Figure 31: The depression scale in the longitudinal survey.

In addition to these more minor changes the longitudinal survey was also modified to accommodate different functionalities. The survey is meant to be taken three times, with the first time merely requiring users to enter their phone number and consent to basic data collection. The subsequent two takes have users answering survey questions, including the PHQ-9 scale as well as the depression scale, waiting two weeks between each take. In order to ensure that users complete the application on these specific times, a notification system was implemented that would allow the application to send single and repeating reminders to users depending on their level of completion with the survey.

After a successful completion of the survey, a notification is be sent to the users' phone reminding them to retake the survey in two weeks. The current time and date is captured by the application and used to create this reminder, which can be scheduled to repeat in set intervals, reminding users each week,

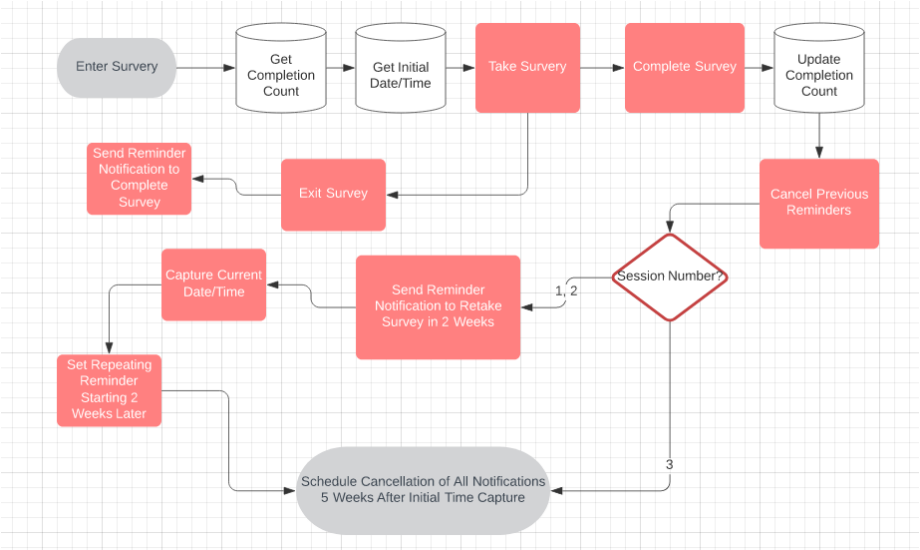

Figure 32: Sequence diagram of longitudinal survey reminders.

day, or hour once this deadline has expired. Once the user completes the survey one additional time, having made it to and completing the final page, this reminder is canceled. If users have not yet completed the survey a third time, another reminder is scheduled to repeat until this final session is finished. This system can also be modified to send notifications if users exit the survey before completing a given session.

3.2 Student Data Collection

Student Data The survey results are stored in four databases: Summer I, Summer II, Fall, and Winter. These included data from students in our college clubs and organizations, courses related to Computer Science and Data Science, and college students reached on social media.

Schema Changes As we started collecting data, we realized that we needed a more flexible data logging system in order to manage our data. Our existing database tables were not storing the timestamps of each data entry or any log tracing data for each session, both of which would be very helpful for data analysis. These was added to the database for the winter collection and logging system. While any email entries to our raffle were still getting logged in the previous database, we made sure the new collection entries for both the app and the website survey were logging into the winter database.

Data Cleaning Once the student data collection was complete, the data entries were cleaned prior to conducting analysis and running experiments with

the machine learning pipeline. This required the team to manually inspect the entries to determine what cleaning criteria were necessary. First, entries for which "not a student" was included in the demographic section were removed. Then, entries for which any version of the word "test" was written in the text prompt were removed.

Next, we decided how to handle duplicate entries. In the two summer and fall databases, we used the paid status in the database to identify which IDS to keep. For surveys completed on the phone application, a paid status of 0 means the survey was incomplete, a paid status of 2 means the survey was complete, and a paid status of 1 means the survey was a duplicate completion. For surveys completed on the website application, a paid status of 0 means the survey was incomplete and a paid status of 10 means the survey was complete. We first decided to keep complete submissions (paid status 2 or 10). We also kept incomplete submission as long as they did not also have a complete submission (paid status 0 without paid status 2 or 10).

In the winter database, we had information about IP address and completion time so we were able to conduct more in depth analysis. There were some participant IDs on the same IP address for which the time between the completions was short. We determined that these completions all belonged to the same participant, most likely someone who repeated the survey one or more times after encountering an error. Based on manual inspections of the data, we decided that five minutes would be the cutoff for multiple completions on the same IP address. For entries within five minutes of each other, the one with the most modalities was kept and the others were dropped. Entries for the same IP address that were submitted more than five minutes apart were treated as separate participants.

We used these criterion to write data cleaning code in Python using pandas and SQLite. The code returns a list of valid IDs for each database that can be fed into the machine learning pipeline.

3.3 Feature Extraction and Machine Learning Pipeline

The concept of a pipeline is important for data processing and machine learning. A pipeline should take in data and in each step of the pipeline perform an operation on that data. After a number of steps through the pipeline, the data will have been transformed into a state that is ready for consumption by the destination program. The feature extraction and machine learning pipeline of this project is designed to take in database files from different data collections and transform them to prepare them as inputs for a machine learning program.

3.3.1 Data Preparedness

SQL SQL was used to collect the PHQ-9 scores from surveys from the EMU data collected to determine if we have enough data to begin applying it to our machine learning pipeline.

Excel and Tableau After extracting the data from the SQL Data file, the PHQ-9 values were loaded into an excel spreadsheet which was then loaded into Tableau to create a dashboard on the current PHQ-9 data. Each element of the dashboard helps to conclude whether or not we have enough data to run on our machine learning pipeline. The important thing to note is that to have adequate training, we need to have different results in order to predict depression. The pie-chart in the top left shows the total number of results and how many of those results have depressed or not depressed scores. Depressed scores are determined by an overall PHQ-9 score greater than 10 and/or a score greater than 1 on the suicide question of our survey (question 9). The "Depressed v. Not Depressed" table displays how many survey users that were depressed by PHQ-9 score were also depressed by Question 9 and vice-versa. From the table, we can see that all participants who were depressed by Question 9 also had depressed PHQ-9 scores. Those that were not depressed given their Question 9 response was nearly evenly split. The "Avg. Sum" and "Question Comparison" gives the average overall scores and average individual question responses for those with depression compared to those not depressed. These elements show us that we have dispersed data and the average scores so that we can predictions based on individual results.

3.3.2 Changes to Pipeline Project Architecture

One of the major goals for the pipeline project was to try to improve the organization of the code and resources in the `Student2020` directory, where the most recent iteration of the project files were stored. As Figure 35 illustrates, there were many folders and uncategorized python files at the root level of the project. With the goal of structuring the project so that it would be easier to find results, datasets and code the directory was restructured.

Updated File System Architecture The file structure was reorganized so that it would contain 4 folders at the root level: `data_sources`, `output`, `pipeline_code` and `support_code`. Subsequent testing of file dependencies showed that it would be simpler and easier to have the `support_code` directory moved to `pipeline_code`, resulting in the current configuration of the "New Architecure", as seen in Figure 34.

Top Level Scripts vs Modules One discrepancy encountered in the code was in the ways `run_models.py` and `feature_extraction.py` were run. `run_models.py`, located in the `pipeline_code` sub-directory, was run by the file `machine_learning.bat`. Being run by a `.bat` file made it so that `run_models.py` used the `__main__` namespace, and had a top-level package of `./pipeline_code`. By comparison, `feature_extraction.py` was imported as a module by `run_feature_extraction.py`, meaning that `feature_extraction.py` had the namespace `__feature_extraction__`, and a top-level package of `./`, the directory root, since `run_feature_extraction.py` was being run by the terminal instead of `feature_extraction.py`.

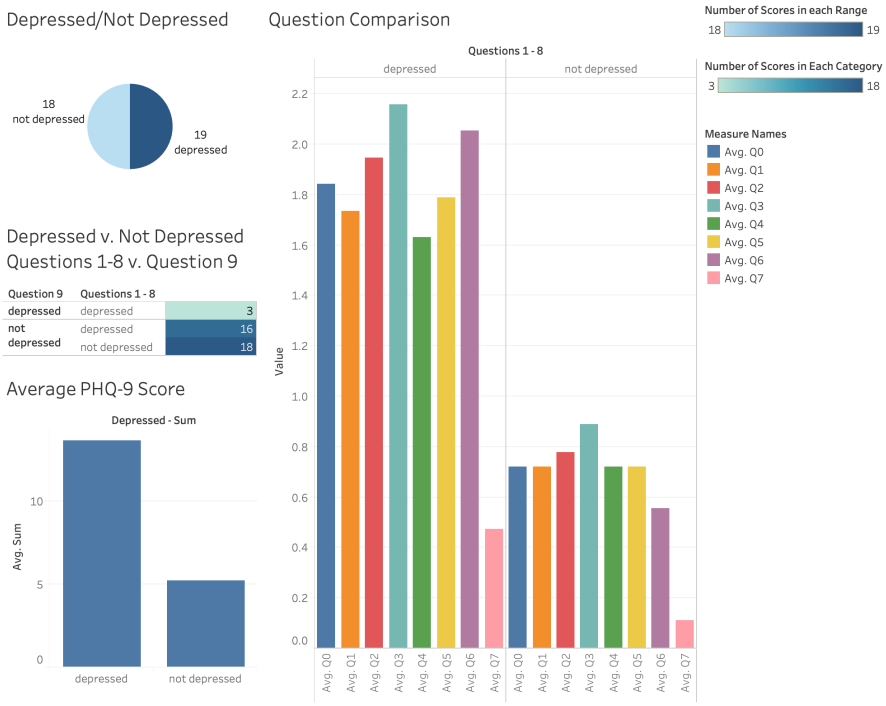

Figure 33: PHQ-9 dashboard for student EMU database.

The Top Level Package Error In Python when a program is run from a terminal, eg `python run_machine_learning.py`, then the `__name__` property of that instance is set to `__main__`. When a script's `__name__ == "__main__"`, then the location the script was run from is considered the absolute root of the environment for the script's runtime. This means that if there is a file in the `pipeline_code` folder that needs to import modules from a parallel or upper directory, then those files will not be accessible and will be met with the error `ValueError: attempted relative import beyond top-level package`. This error was the reason that the `support_code` directory was moved to become a sub-directory of `pipeline_code`, so as to prevent top level package errors when importing code from the `support_code` folder.

"Runner" Files To reconcile the differences between these two implementations, a `.bat` file for both machine learning code and for feature extraction code was created at the root level, along with a "runner" file. The purpose of these runner files was threefold. First, it would import the respective "functional code" file from `/pipeline_code`, either `run_models.py` or

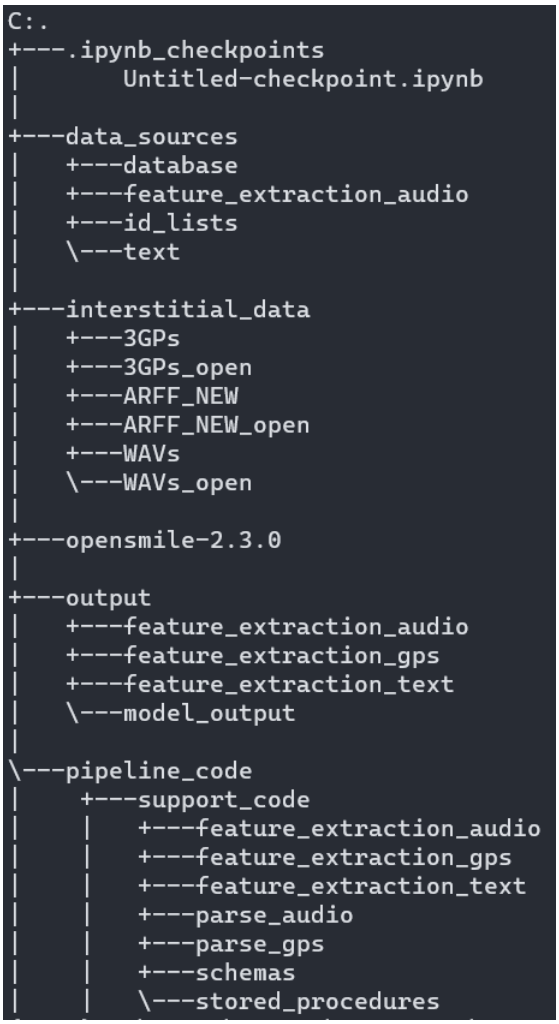

Figure 34: Tree structure of the folder hierarchy in the new layout of the directory.

`feature_extraction.py`. Second, it parses the arguments passed in from either the terminal, or a `.bat/.sh` file with preconfigured parameters. Third, it runs the `main` method of the imported python module. The major benefit to this method of a `.bat` file running an argument parser which runs the main code is that, with the `.bat` file and the argument parser "runner" at the root directory, top level package errors can be avoided. Another bonus is that the relative directory path is the root, eliminating the need for directory changes during execution. Using the `.bat/.sh` approach makes it easy to set up multiple runs of the feature extraction and and machine learning code and to repeat

those runs, and the use of the "runner" functions allow for extra logic to be introduced in between parsing the terminal arguments and the invocation of the `main` method.

Interstitial_data Directory Partway through the life of the project another folder was added to the root of the directory, the `interstitial_data` folder. The idea for adding this folder came from a desire to simplify the process of writing different stages of data processing to csv files and then reading those files in later instances. The `/data_sources` folder already existed for starting data needed by processes and the `/output` folder existed to hold the end results of feature extraction and machine learning. A place was needed where those files created while feature extraction and machine learning were still in motion. At the time processes would either write files in the directory from which they were run, or they would attempt to write those files in the directory of the next segment of the pipeline which would need the data. This approach caused data to be scattered around different levels of the project's hierarchy, causing confusion for people trying to find what they needed. It was hard to tell, for instance, when looking in a folder whether the csv file in the folder was a product of the python file in the folder, or a data source that the file would be consuming. By putting all temporary data into a common directory that ambiguity was eliminated.

3.3.3 Generalization of the Feature Extraction Process

Over the course of multiple iterations of data collection and survey designs, the database schemas used for storing data have changed as well. Prior to this year's project the resolution was to use `if/else` statements based on a text argument that was passed through the terminal. This logic would then determine whether to execute code corresponding to the schema of either the 2018 project, Moodable, or the schema of the 2019 project, EMU. With the addition of this year's data to the mix of databases the pipeline would need to support, along with the goal of publishing the codebase for others to use alongside the research, it was decided that the feature extraction code needed to be generalized.

Stored Procedures One of the primary reasons that the previous feature extraction implementation relied on knowing which database was in use was because of how the `sqlite3` package was used. The feature extraction program first connected to a local database (`.db`) file and established a connection via a *cursor*. The cursor was then used to execute SQL queries against the database. The problem was that there were an abundance of hardcoded SQL queries written in the code that would be selected based on the database in use. To scale the feature extraction program more `if/else` conditions would need to be added for each new database that could be used. The solution to this problem that was implemented was to have users supply a set of preconstructed SQL queries that could be used like stored procedures in a real database environment. The

```
10/13/2020  09:38 PM    <DIR>          .
10/13/2020  09:38 PM    <DIR>          ..
10/13/2020  09:38 PM    <DIR>          .ipynb_checkpoints
09/06/2020  06:40 PM             1,668 add_demographic_to_features.py
09/06/2020  06:40 PM             1,801 add_phq9_gad7_to_features.py
09/06/2020  06:40 PM             3,634 all_modalities_summary.py
09/06/2020  06:40 PM             2,308 calculate_phq_gad.py
09/06/2020  06:40 PM    <DIR>          correlation-plots
09/06/2020  06:40 PM             7,120 data.csv
09/06/2020  06:41 PM    <DIR>          database
09/06/2020  06:41 PM    <DIR>          demographic-distribution
09/06/2020  06:41 PM             1,792 demographic_features_only.py
09/06/2020  06:41 PM            15,142 edit_modality_data.py
09/06/2020  06:41 PM            64,935 emu-workflow-diagram.jpg
09/07/2020  02:38 PM    <DIR>          feature_extraction
09/07/2020  02:38 PM    <DIR>          feature_extraction_audio
09/06/2020  06:41 PM    <DIR>          feature_extraction_gps
10/13/2020  09:42 PM    <DIR>          feature_extraction_text
09/06/2020  06:42 PM    <DIR>          final-data
09/06/2020  06:42 PM             3,050 get_demographic.py
09/06/2020  06:42 PM               916 get_gad7.py
09/06/2020  06:42 PM             1,110 get_phq9.py
09/06/2020  06:42 PM    <DIR>          machine_learning
09/06/2020  06:42 PM    <DIR>          modality-distribution-histograms
09/06/2020  06:42 PM    <DIR>          modality-lengths
09/06/2020  06:42 PM    <DIR>          modality-summary
09/06/2020  06:42 PM    <DIR>          parse_audio
09/06/2020  06:42 PM    <DIR>          parse_gps
09/06/2020  06:42 PM            13,238 plotting_modalities.py
09/06/2020  06:42 PM             9,337 plot_correlations.py
09/06/2020  06:42 PM             2,053 plot_distributions.py
09/06/2020  06:42 PM             1,830 plot_q9_distribution.py
09/06/2020  06:42 PM    <DIR>          raw-data
09/06/2020  06:40 PM             6,766 README.md
09/06/2020  06:42 PM               187 requirements.txt
09/06/2020  06:42 PM             1,437 run_feature_extraction.py
10/13/2020  09:34 PM            23,230 run_fe_main_from_root.ipynb
09/06/2020  06:42 PM               848 run_machine_learning.bat
09/06/2020  06:42 PM               257 run_machine_learning.sh
09/06/2020  06:42 PM               400 run_machine_learning_turing.sh
09/06/2020  06:42 PM             1,780 run_plot_features.bat
09/06/2020  06:42 PM    <DIR>          timeline
09/06/2020  06:42 PM    <DIR>          total-distributions
```

Figure 35: Screen capture of the 'dir' command used in the directory of this summer's published repository.

python program loads the queries from another file and executes them, achieving the same results in far fewer lines.

List of Cleaned IDs The stored procedure innovation was directly tied to the next change that was made, loading the results of external data cleaning instead of performing the data cleaning in the feature extraction code. The data cleaning elements of the feature extraction code were just as complex and messy as the other SQL query implementations. Since data cleaning was already being performed as a part of the data collection process, the list of good IDs that were gathered are supplied from a file to the feature extraction program, and can be used in the same way that the SQL-gathered IDs were used in the previous iteration.

3.3.4 Tweets Feature Extraction

Our preexisting feature engineering code took into account all the aspects of a simple text message. However, when expanding the data analysis to tweets, we wanted to add tweet-specific features. Prior to incorporating these features into our analysis, we strengthened our text feature engineering code so that a more relevant and revealing word set is used to analyze the contents of the tweets. The TweetsPUL dataset was used in this analysis.

Lexicon Library: Empath Empath, created by Ethan Fast and his team at Stanford University, is a high quality lexicon used to analyze text. Up to par with the gold standard lexicon, LIWC, in terms of efficiency, it can generate and validate new lexical categories with a few seed terms. It then compares the wordsets generated against the dataset and returns the percentage of the categories in the data. Its pre-existing categories are also validated by human confirmation on a crowdsourcing platform, Mechanical Turk. Empath's contents are currently mined from three models, Fiction (through an e-book community called Wattpad), Reddit, and New York Times.

Previously, our code used the Fiction model, the model that first existed. As we anticipated a more modern vocabulary from our target population, we conducted an experiment to compare the results between the Fiction and the Reddit model, where users are more likely would include uncensored, more daily used vocabulary than the other two. Additionally, we also plan to generate and add categories related to the present.

Category Wordlists Experiment To ensure that we generated categories with the correct seed words, we ran a small naive experiment. First, we generated categories from the default (Fiction) model and the Reddit model using every other three consecutive words from the default categories wordsets as seed words before comparing every two sets of wordsets to find the common words between them.

Hashtags and Mentions In order to make our feature extraction pipeline more robust and adaptable to tweets, we wanted to start adding features that could only be gained through tweets. A study shows a good predictability of behavioral health discussions through tweet features, mainly popular mental health related hashtags (McClellan et al., n.d.). Following this, we wanted to explore more on the increase of accuracy the inclusion of hashtags and mentions (references to other Twitter users) might bring. We decided on adding the count of hashtags and mentions as new features to our pipeline. Additionally, we attached the hashtags of the tweets to their tweet content itself in order to have Empath perform an analysis on them as a whole.

3.4 Time Series Experiments

The goal of these experiments was to identify whether time series constructed from text and call logs can be used to predict if an individual is depressed. Time series are represented by a list of quantities, where each quantity in the series represents data from a given time interval. Quantities in time series are ordered chronologically (*Introduction to the Fundamentals of Time Series Data and Analysis*, 2019). First, time series were constructed using the text and call logs of participants in the Moodable and EMU datasets. Next, machine learning experiments were run using the time series and features extracted from the time series.

3.4.1 Data Selection and Time Series Construction

Time series were constructed using the text and call logs of participants in the Moodable and EMU datasets. All participants from the Moodable dataset and a cleaned subset of participants from the EMU dataset were used. To be able to identify which dataset participants came from, an 'm' or an 'e' was appended to the front of the participant ID to represent the Moodable or EMU dataset respectively.

Data required for the construction of the time series was manually extracted from the Moodable and EMU databases and saved to .csv files using SQLiteStudio. Four .csv files were created, one storing text logs, one storing call logs, one storing PHQ-9 survey responses, and one storing survey completion dates. These .csv files were then cleaned to contain only participant IDs that were going to be used in the construction of the time series.

Python code was developed using the NumPy and pandas libraries to create the time series. The code requires three input .csv files: one with text or call logs, another with PHQ-9 survey responses, and a third with survey completion dates.

First the code handles duplicate log entries. If a participant has duplicate log entries, only one entry is saved. If multiple participants share the same log entries, it is interpreted that one person completed the survey more than once and only the most recent completion is saved.

Next, the time series are constructed according to three variables:

- Days: the length of the period before the survey completion date to use data from
- Interval: the aggregation interval (in hours) to use for time series quantities
- Direction: the direction of the texts or calls (either all of them, only incoming, or only outgoing)

All survey completion dates are rounded down to midnight of the same day to ensure that the time series data start and end at the same time of day for each participant. Then, logs within the specified direction and number of days of this completion date are taken. Before constructing time series with these logs, participants that have fewer than a specified number of texts or fewer than a total duration of calls (in minutes) are dropped so that the time series are not too sparse. Then the time series are constructed.

Time series are created for each participant with the following variables:

- Counts: the number of texts or the total duration of calls (in minutes)
- Average length: average length of texts (in characters) or average duration of non-zero duration calls (in minutes)
- Unique contacts: the number of unique contacts

For calls, it was decided that the total duration would be used instead of a raw count because people tend to have fewer calls, and total call duration would give more information than just a count.

After the time series are constructed, the PHQ-9 survey responses are used to create a PHQ-9 summary score for each participant. A specified value determines whether a participant is labeled depressed based on this summary score; participants with a summary score equal to or above this value are labeled depressed, and those below are not. A true/false label is attached to the time series for each participant to indicate whether they are labeled as depressed.

Finally, the time series are output to a .csv file with a name that describes which parameters the data were constructed from. Figure 36 shows how the files are named.

Table 2 summarizes the command-line arguments required for the time series construction code. The time series construction code can take in lists of inputs for the following variables: modalities, days, intervals, and directions. The code will construct time series with all combinations of these variables.

3.4.2 Time Series Visualization

After constructing the time series, visualizations were created with Python and Matplotlib to see if there were any visual trends in the time series. Pandas was used to parse the values from a .csv file with ID numbers, time on the phone counts, and text counts. After parsing the data, the code loops through the

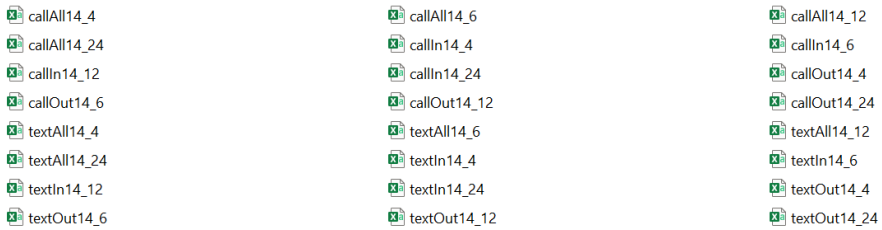

Figure 36: .csv files for time series data. Files are named like modalityDirectionDay_Interval.csv.

VARIABLE	DESCRIPTION
metadatapath	Path to .csv containing logs
completionpath	Path to .csv containing survey completion dates
phqpath	Path to .csv containing PHQ-9 survey responses
modalities	Either "text" or "call"
directions	Either "All", "In", or "Out"
days	Number of days before survey completion to use data from
intervals	Aggregation interval (in hours)
cutoff	PHQ-9 depression label cutoff
drop	Cutoff for dropping participants
output	Name of folder to output time series to

Table 2: Command-line arguments for time series construction code.

data values for each ID and graphs either the total duration of calls or number of texts over 4, 6, 12, or 24 hour aggregation intervals. Each graph displays the incoming, outgoing, and total calls or texts.

Color Schemes After experimenting with different colors, such as blue, black and different shades of purple, we decided to use blue for incoming, dark orange for outgoing and black for total. This decision was made in order to avoid using red and green for color blindness, pale colors that do not show up well on graphs, and colors associated with certain emotions. By doing two 'colorful' colors for the individual data recordings (incoming and outgoing) it shows their relationship to the total recordings, which is in black.

Line Styles One of the first experiments was to use different line styles for each line (incoming, outgoing, total). While we liked this design, the alternative was solid lines with different shaped markers (circle, square, diamond). While we liked both line styles, we decided on solid lines with a square marker for total, diamond for incoming, and circle for outgoing, because it showed up the best with a gray-scale filter in case we have to use the graphs in black and white. We also made each line different widths so we can see each line even when they

overlap.

Graph Sizing In order to make space for all points on the plot, without it being cluttered, we made the graph wider to allow for all points to be seen.

Title For analytical purposes, we included the ID number, time period, and PHQ-9 score associated with that ID. These are important for comparing data for the same user, determining which aggregation intervals are best for visualizing and for Time Series Machine Learning, and comparing results for similar PHQ-9 scores.

X-Axis The X-Axis shows which day the recordings are for out of a 14 day data collection. Each graph (no matter the recording segment), will be marked for each of 14 days. Graphs for 4 hour aggregation intervals would have six data points in between each day mark and graphs for 6 hour aggregation intervals would have 4 data points in between each day mark.

Y-Axis The Y-Axis simply plots the value associated with each data recording, which would be either call duration (in seconds) or number of texts depending on the graph.

3.4.3 Experiments with Time Series

Next, machine learning experiments with k-Nearest Neighbor (kNN) classifiers with $k = 3$ were conducted using the time series. Multiple train/test splits (with test size 0.33) were run to account for variability in performance depending on the split. Each train/test split was stratified so that the training set and the testing set had the same distribution of target labels as the full dataset. For each split, the training data was balanced before being used to train the kNN classifier. The following performance metrics were calculated for each train/test split: accuracy, precision, sensitivity, specificity, F1, and AUC.

The kNN classifiers use dynamic time warping (DTW) as a distance metric between time series. DTW is used to compare two time series that are different in length or cannot be matched linearly to one another. DTW differs from Euclidean distance metrics, where the distance between quantities in the same point in the time series are summed. Figure 37 compares Euclidean and dynamic time warping distance metrics. DTW was used instead of Euclidean distance because even though the time series are the same lengths for each participant, participants had logs from different date ranges and may be in different time zones, so it would not make sense to compare their times series linearly (Zhang, 2020).

The machine learning experiments were conducted using Python code and Scikit-learn and were run on WPI's high performance computing cluster, Ace. Since DTW can be slow, the Python library fastdtw (Salvador & Chan, 2007), which approximates DTW, was used. The command-line arguments required

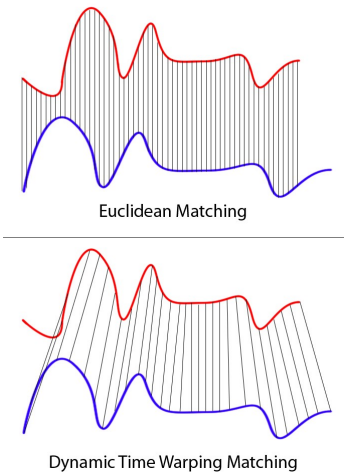

Figure 37: Comparison of Euclidean and dynamic time warping distance metrics (Zhang, 2020).

VARIABLE	DESCRIPTION
modalities	Either "text" or "call"
directions	Either "All", "In", or "Out"
days	Number of days before survey completion to use data from
intervals	Aggregation interval (in hours)
variables	Either "counts", "average_length", or "unique_contacts"
samplings	Either "up" or "down"
numsplits	Number of splits to run for each parameter combination
inputpath	Path to folder with input time series
outputFile	Name to use in results files

Table 3: Command-line arguments for time series machine learning code.

by the code are summarized in Table 3. Similar to the time series construction code, the machine learning code can take in lists of parameters for some of the inputs (modalities, directions, days, intervals, variables, and samplings). The code will run experiments with all combinations of these inputs.

The machine learning code creates two output .csv files. One contains the parameters and evaluation metrics for each train/test split. The other calculates the mean and standard deviation for each evaluation metric across all the train/test splits for each combination of parameters.

3.4.4 Experiments with Time Series Features

Features were extracted from the time series using the Time Series Feature Extraction Library (TSFEL), which extracts temporal, statistical, and spec-

tral features from time series data (Barandas et al., 2020). TSFEL feature were extracted using code provided by the TSFEL's developers that extracts all available features for a time series.

Features were extracted for each time series variable (count, average_length, unique_contacts). Each variables' feature were placed in a separate .csv file. In addition a .csv file was created that contained the features for all three variables combined.

Machine learning experiments were conducted using the time series features. Logistic regression (LR), k-Nearest Neighbor (KNN) with k = 3, support vector (SVC) and random forest (RFC) classifiers were trained. Similar to the experiments with the raw time series, multiple train/test splits (test size = 0.33) were run to account for variability. The train/test splits were stratified, and the training set was balanced. Prior to training the classifier, principal component analysis (PCA) was run to decrease the dimensionality of the feature set. The following performance metrics were reported for each experiment: accuracy, precision, sensitivity, specificity, F1, and AUC.

Python code was developed with Scikit-learn to run these experiments, and the experiments were run on WPI's high performance computer cluster, Ace. The code takes in the same arguments as in Table 3 with the addition of which machine learning method to be used (LR, KNN, SVC, or RFC) and the maximum number of principal components to use. The code iterates from one to this maximum number of principal components. The code can take in lists for the parameters: modalities, directions, days, intervals, variables, methods, and samplings, and will run experiments with all combinations of them.

The same output files are created as in the previous machine learning code: one with performance evaluation metrics for each train/test split, and another that calculates the mean and standard deviation of these performance metrics across each combination of parameters.

3.5 Stereotype Threat Priming Study

The purpose of this experiment is to see if we can trigger stereotype threat with our Android mobile app. More specifically, we wanted to see if a fact about depression and gender would influence how participants complete the PHQ-9 and GAD-7 questionnaires within the Android application. There were also two secondary goals - payment for text content and audio prompts on a mobile application.

3.5.1 Crowd Sourcing

Prolific In order to collect enough data for our study, we used a crowdsourcing platform called Prolific. This site allowed us to filter our participants based on our requirements to represent our target population. We were able to share the link to download our android app and the survey was shared with active Prolific participants within the filters. Participants on Prolific were compensated for taking the survey. To encourage people to take our survey, we

provided the following description "Participants will be asked to download an application to their Android phone, answer survey questions, and record samples of their voice. Additionally, participants will be given the option to share phone data, such as text messages." We performed small collections in order to perform experiments, and ensure everything in application was working correctly. We worked to obtain 400 participants for the bulk of the survey.

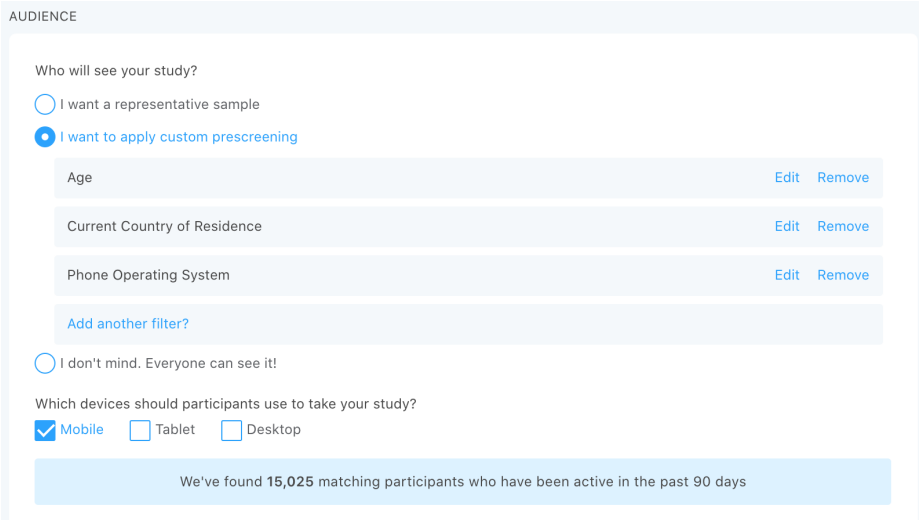

Figure 38: Participant filtering on Prolific.

3.5.2 Sample Size

In order to make sure our results were statistically significant, we calculated sample size so that we had enough data to make conclusions that was representative of our population.

Target Population For this study, we targeted US citizens 18 and older that have an android. Based on United States Demographic Statistics, there are approximately 209 million US citizens over the age of 18 (*United States Demographic Statistics*, n.d.). In addition, 91% of the Adult Population owns a cell phone and 28% of cell phone owners own an Android (Smith, n.d.). This means our target population was around 53 million people.

Confidence Level and Margin of Error This study used the default confidence levels and margin of error - 95% confidence level and 0.05 margin of error. It is important to note that the confidence level does not have to be in line with the margin of error. The main difference in the two is that the confidence level is how confident we are that our data and conclusions represent our target population and the margin of error is the percent of inaccuracy in our

results. With a target population of 53 million people, a 95% confidence level and 0.05 margin of error, we needed around 385 participants to be statistically significant.

3.5.3 Fact Selection

Fact Qualifications Previous studies determined that more simple and well-known facts are most effective for triggering stereotype threat and have had the most effect on the participants whether they were in a group setting, completing a survey, or performing a task (Pennington et al., 2016).

Fact Specifications For this study, our target groups were men and women. The chosen fact is "Women experience depression at roughly twice the rate of men" (*Depression and Women*, n.d.). This fact fits into both suggestions for triggering stereotype with a simple fact - it is straightforward and to the point, as well as a well-known stigma. It is important to note that the fact isn't necessarily true. It is not required for the fact to be true in order to trigger stereotype threat, it simply has to be believed. When it comes to mental health and many of the expected symptoms of depression and anxiety, they are mostly in line with the more common symptoms experienced by women. The questionnaires, such as PHQ-9 don't address other symptoms, such as aggression or anger (*Men and Depression*, n.d.). In addition, there are many social stigmas related to men and women and how to address and discuss mental health.

3.5.4 Text Message Collection

The purpose of this secondary goal was to determine if offering participants additional bonuses for their text messages would encourage them to share their text messages.

Additional Bonuses The initial test-run offered participants no additional pay for not sharing any text data, 5 cents to share text logs and 25 cents to share text message content. The initial 25 cent pay incentive was chosen, since the base pay to take our survey was 75 cents and we wanted the pay per participant to be approximately $1.00. After noting the complexity of collecting either text logs or text content and analyzing that we could collect enough text content without offering the log option, we decided to only offer two options - not sharing any text data and sharing text content. Due to the fact that we initially offered 25 cents and wanted to evaluate how much we would need to pay participants to increase the percentage that would be willing to share data, we tested 5 pay incentives - 5 cents, 15 cents, 25 cents, 35 cents, and 45 cents.

3.5.5 Audio Prompt

This secondary goal was based on interview questions that have been used in in-person studies. The purpose was to determine if participants answer the

interview questions the same way on a mobile application as they would during an in person study.

Prompt Selection Participants were asked to answer two open audio prompts and one closed audio-prompt. In addition to these prompts being used for in-person surveys and interviews, we wanted the open audio prompts to be questions that may offer insight to mental health. The first question is to describe their dream job. This question gives insight into the participants outlook on the future and some of their inspirations. The second question asks participants to describe a positive influence in their life. This will encourage participants to discuss someone important to them and possibly include the significance to their life. Finally, they are asked to repeat the phrase "The North Wind and the Sun had a quarrel about which one of them was the stronger."

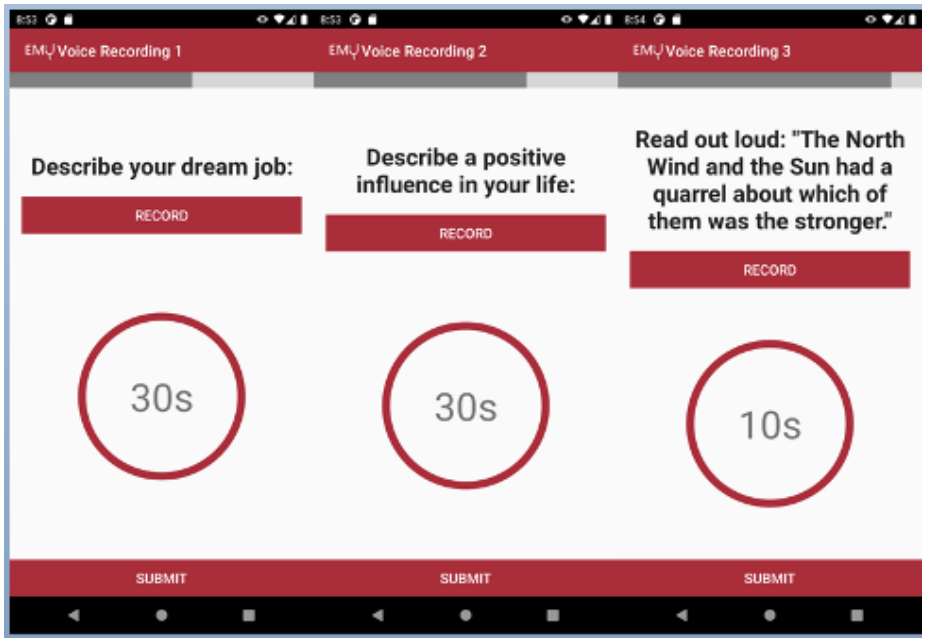

Figure 39: Audio prompts on Prolific application.

3.5.6 Data Cleaning

Python was used to run PostgreSQL queries on the data and export it into a spreadsheet. The queries utilized the prolific IDs to match participant users to the data in the tables. Session IDs were then matched to the prolific IDs to search the data to provide what versions of the app the participant completed, PHQ-9 and GAD-7 scores, demographics, and whether or not the participant shared text messages.

4 Results

4.1 Student Data Analysis

After cleaning the data as described in section 3.2, there were 317 good IDs. Out of these IDs, 281 completed the website survey and 36 completed the phone survey. The full list of IDs for each database can be found in Appendix C. This is larger than other related datasets. Table 4 compares the size of our student dataset to other datasets related to student mental health.

Dataset	N
Student Data	317
StudentLife (Huckins, et al.)	178
LifeRhythm (Ware, et al.)	104
LifeRhythm (Fahran, et al.)	79
DemonicSalmon (Boukhechba, et al.)	72
Reality Commons (Madan, et al.)	70
StudentLife (Wang, et al.)	48

Table 4: Size of datasets related to student mental health.

Out of the 317 participants in the cleaned set, only 310 had demographic information. This can be explained by the fact that the demographics page was at the end of earlier survey versions, so participants who completed the survey early on in the collection may not have gotten to it. Figure 40 shows the distribution of PHQ-9 scores for participants in the cleaned subset of IDs. In addition, Table 5 shows responses to the ninth question of the PHQ-9, which relates to suicidal ideation.

Response	N	Mean PHQ-9
0	236	7.67
1	49	14.49
2	17	18.00
3	15	22.35

Table 5: Responses to the ninth question of PHQ-9, which asks about frequency of suicidal ideation (0 = not al all, 1 = several days, 2 = more than half the days, 3 = nearly every day).

The student data has a similar distribution of PHQ-9 scores as the StudentLife study (Wang et al., 2017). In both datasets, more students have PHQ-9 scores on the lower range while there are fewer that exhibit higher scores. We also found that the response to the ninth question of PHQ-9 is positively correlated with the overall PHQ-9 score. Participants with a higher response to the suicidal idealation question tended to have a higher overall PHQ-9 score.

Table 6 shows the number of students who completed each modality, along with the mean PHQ-9 score for each subset of participants. The large drop

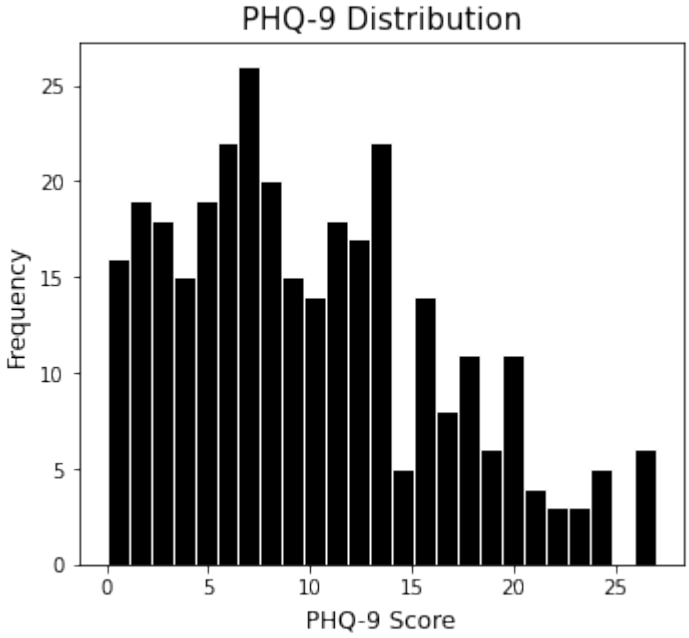

Figure 40: PHQ-9 distribution for cleaned subset of IDs.

in participants between the text prompt and open audio modalities can be explained by an error in the data collection process. Some participants on the website survey reported not being able to submit and continue after completing the open audio prompt. In addition, some participants reported not feeling comfortable completing this modality. There is no way to tell if participants dropped out at this point due to the error or unwillingness to share.

Willingness to Share Analysis was conducted to determine which modalities participants were willing to share. Every participant who completed the PHQ-9 questionnaire also completed the text prompt (317 total). In addition, nearly every participant who completed the PHQ-9 questionnaire also answered the demographic questions (310 total). It is possible that the 7 participants who did not answer the demographic questions took an earlier version of the survey, where the demographic questions were at the end. Out of the 40 participants who had Twitter, 36 (or 90%) shared their username.

It did not make sense to report willingness to share information for the audio prompts with the full set of cleaned IDs. Since the majority of the completions were on the website survey, some participants probably did not complete the audio prompts not because they were unwilling, but due to the error in the survey. Instead, the willingness to share analysis was only conducted on the subset of participants who completed the phone survey, because the error was

Modality	Count	Mean PHQ-9
Text prompt	317	10.00
Open audio	203	9.67
Closed audio	197	9.77
Has Twitter	40	9.88
Twitter username	36	9.69
GPS	22	7.41
Call logs	10	7.70
Text logs	10	7.70
Tweets	6	6.00
Calendar	11	7.18
Contacts	11	7.18

Table 6: Number of participants who completed/shared each modality and mean PHQ-9 for each subset.

not observed there.

Table 7 shows the percentages of participants who completed audio prompts and shared phone-specific modalities out of those who completed the phone survey.

Modality	Shared	Total	%
Open audio	30	36	83.3
Closed audio	30	36	83.3
GPS	22	36	61.1
Call logs	10	36	27.8
Text logs	10	36	27.8
Calendar	11	36	30.6
Contacts	11	36	30.6
Tweets	6	14	60.0

Table 7: Willingness to share for audio prompts and phone-specific modalities for participants who completed the phone survey. The tweets modality is out of 14 because that is the number of participants who completed the phone survey that reported having twitter.

Demographics Tables 8 through 11 summarize the demographics of the 310 participants who provided demographic information. Undergraduates students had higher mean PHQ-9 scores than graduate students. Women had higher mean PHQ-9 scores than men, which matches the findings of a 2015 study (Albert, 2015). Younger students (18-23) had higher mean PHQ-9 scores than older students (24-39). The mean PHQ-9 scores in the race/ethnicity distribution were not compared because the sample sizes of the most categories are small.

Student status	N	%	Mean PHQ-9
Undergraduate	248	80.0	10.25
Graduate	62	20.0	9.02

Table 8: Distribution of student statuses and mean PHQ-9 for each subset.

Gender	N	%	Mean PHQ-9
Woman	182	58.7	10.41
Man	115	37.1	8.85
Other	13	4.2	14.38

Table 9: Distribution of genders and mean PHQ-9 for each subset.

Age	N	%	Mean PHQ-9
18-23	252	81.3	10.17
24-39	55	17.7	8.89
40-55	3	1.0	16.33

Table 10: Distribution of ages and mean PHQ-9 for each subset.

Race/Ethnicity	N	%	Mean PHQ-9
White	201	64.8	9.77
Asian	59	19.0	9.42
Black	10	3.2	10.00
Other	10	3.2	14.43
Hispanic/Latino	9	2.9	9.22
Hispanic/Latino, White	8	2.6	11.25
Asian, White	4	1.3	9.75
White, Other	3	1.0	18.33
Black, White	2	0.6	11.00
Asian, Native/Pacific Islander, White	1	0.3	12.00
Asian and Hispanic/Latino	1	0.3	24.00
Asian and Black	1	0.3	11.00
Prefer not to answer	1	0.3	0

Table 11: Distribution of race/ethnicity and mean PHQ-9 for each subset. Participants were asked to check all races that they identified with. To save room in the table, White/Caucasian is abbreviated as White and Black/African American is abbreviated as Black.

We compared the demographic distributions from the student data to those from WPI, since we assumed that the majority of participants were WPI students. All the demographic data for WPI was found on the university's data dashboards (*Data Dashboards*, n.d.). We used the demographic data for both full-time and part-time WPI students. Tables 12 through 14 compare the de-

mographic distributions in the student data to those at WPI.

In both the student data and at WPI, there are more undergraduate students than graduate students. However, the student data is about 80% undergraduate students while the overall WPI population is about 70% undergraduate students. This difference makes sense because we sent the survey out to more undergraduate classes, many of which were introductory classes with a large enrollment, than graduate classes. In addition we sent the survey out to several student body clubs at WPI, which tend to have more undergraduate participants. We did not compare the age distributions directly since the data on the WPI dashboard uses slightly different age ranges. However, we observed that the majority of undergraduate students at WPI are aged 18-22, while graduate student ages cover a larger range. In the student data, we had 248 undergraduate students and 252 students in the 18-23 age group. It is likely that most of the undergraduate students fell into this age group.

We found that while more women completed our survey than men, the WPI population consists of more men than women. We did not know if this is because women were more willing to complete the survey or because the survey was sent out to more women. We also observed that the student data contains similar percentages of White and Black/African American students to the WPI population. However, the student data contains more Asian Students and fewer Hispanic/Latino students than the WPI population.

Student status	Student %	WPI %
Undergraduate	80.0	70.7
Graduate	20.0	29.3

Table 12: Distribution of student statuses in student data versus the WPI population.

Gender	Student %	WPI %
Woman	58.7	37.1
Man	37.1	62.9
Other	4.2	N/A

Table 13: Distribution of genders in student data versus the WPI population.

Correlation Analysis Correlation plots and tables were created for the summer and fall data collections. We selected features that had variance across different entries in the data collection to analyze how correlated the features were. While we took all data given by a participant, in the future taking data from a smaller time period, such as two weeks, could help standardize the features. Figures 41, 42 and 43 show the correlation plots and tables for the summer and fall data collections. Tables containing the correlation values can be found in Appendix D.

Gender	Student %	WPI %
White	64.8	58.6
Asian	19.0	7.3
Black/African American	3.2	3.0
Hispanic/Latino	2.9	7.9
Two or more races	6.5	2.8
Other/Prefer not to answer	3.5	5.7

Table 14: Distribution of race/ethnicity in student data versus the WPI population. We only compare against the race/ethnicity options presented in our survey, so the WPI percentages may not total to 100.

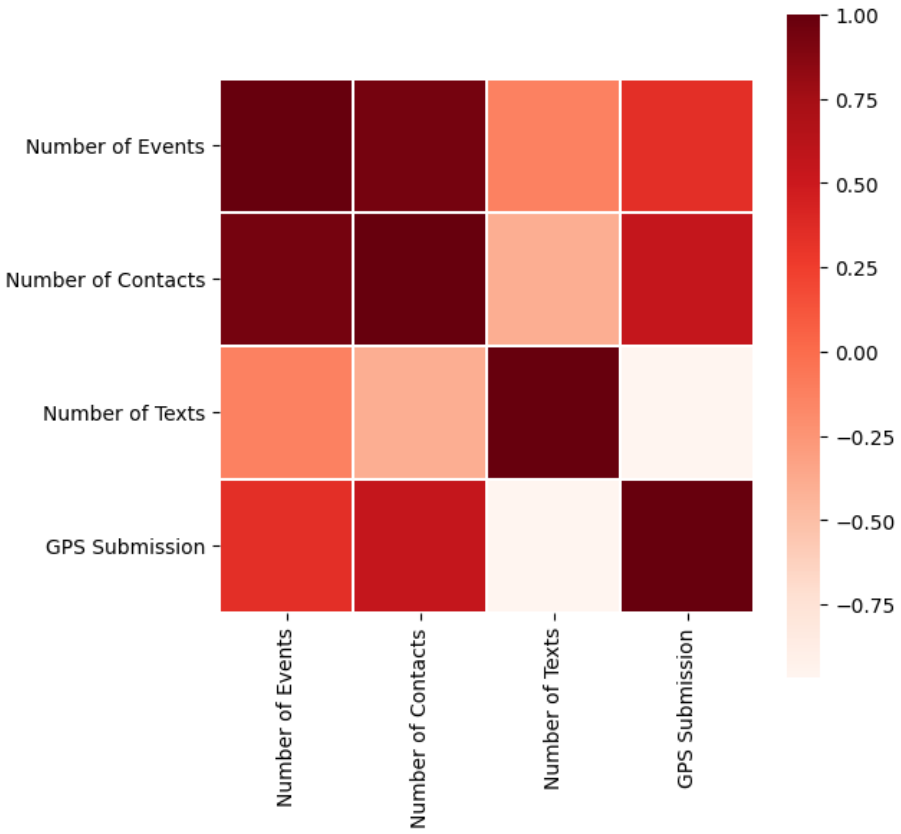

Figure 41: Correlation plot for the summer I collection.

There were smaller correlations between the PHQ-9 scores and other features than expected. The PHQ-9 scores were most highly correlated to the ninth question of the PHQ-9, which is related to suicidal ideation. This is expected as

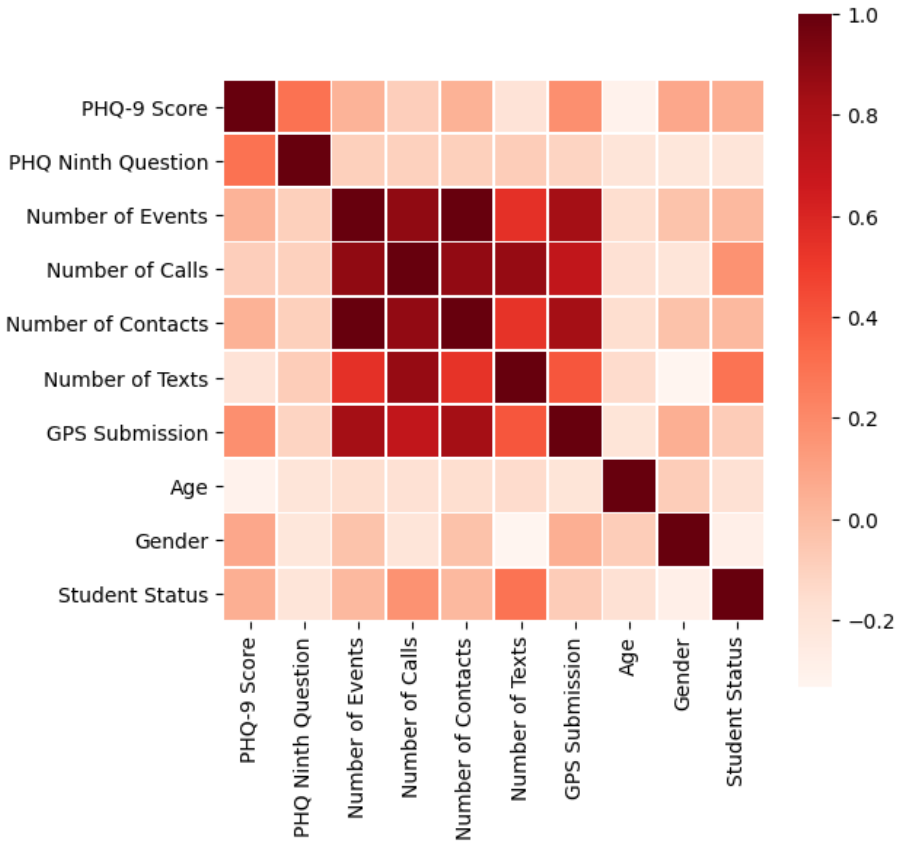

Figure 42: Correlation plot for the summer II collection.

people experiencing more severe depression are more likely to experience suicidal thoughts. In addition, we observed a higher correlation between the number of calls in call logs, the number of calendar events, and the number of contacts per person. It makes sense that these features are correlated because they all indicate that a person is more active and social.

4.2 Machine Learning Pipeline Results

Two major data collections were performed, one in the fall and one in the winter. For each of those data collections, two primary modality groups (audio and text) were collected. Of the four combinations of data collections and modality groups, only the fall audio group was able to be completely run through the pipeline. The machine learning pipeline, comprising of the feature extraction and machine learning components, is not complete for all modalities and datasets. The other three data/modality groups remain a work in progress. It

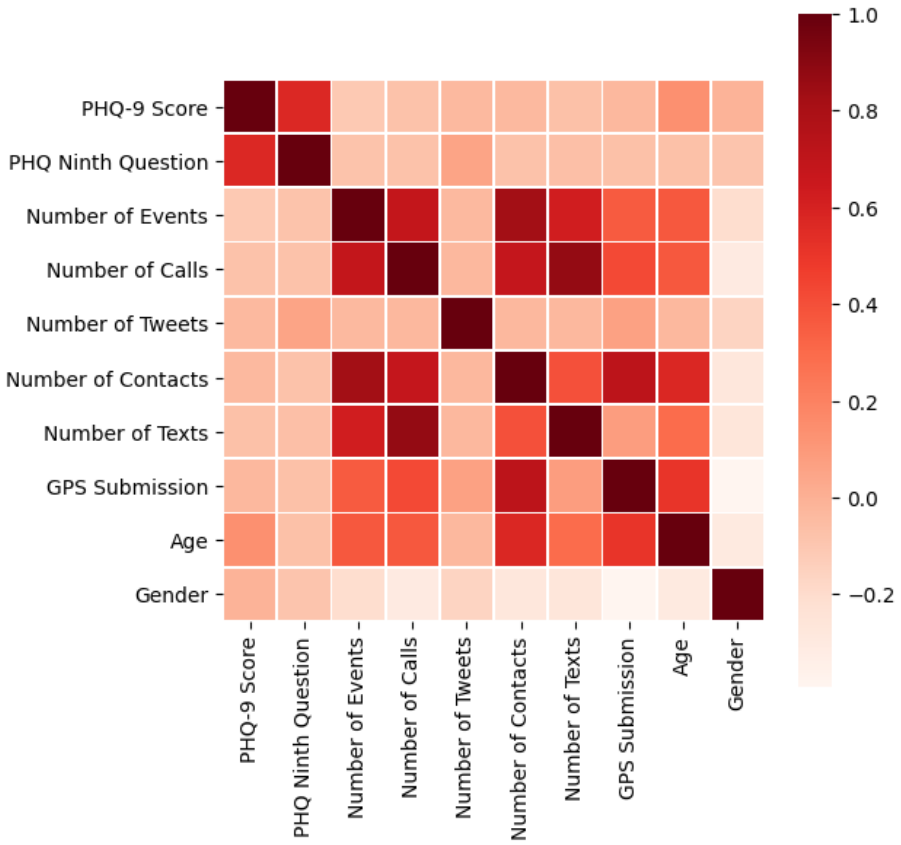

Figure 43: Correlation plot for the fall collection.

is also important to mention, however, that the pipeline continues to work normally on previous years' data, proving that the efforts made to generalize the pipeline's processing ability have been successful.

Running the Model In Appendix I the sequence of commands used to generate the results are displayed. The model was run using a train-test split approach and regular oversampling, and feature selection was done using Principal Component Analysis. The opposite target was set to Null because there was no GAD-7 data collected to use as an opposite target.

The Results Table 15, below, highlights a particularly high scoring result from each of the open and closed audio modalities. Of note is that despite the models differing, both the open and closed audio had their best scores when using low numbers of principal components. In appendix G there are charts which map the changes in AUC, F1 and Accuracy for all models as the number of

principal components increases. For both open and closed audio as the number of principle components increases, the quality of the model's predictions tends to go down. In the end, both open and closed audio performed rather similarly to one another across models and principal component counts, with open audio holding an edge over closed audio in its best run. The open audio's F1 score of 80% is very good, and the closed audio's F1 score of 76.76% is good as well. One more thing to consider is that due to difficulties in getting the initial round of results, the results were not able to be fully experimented upon with different types of upsampling, feature selection methods or evaluation strategies.

Modality	Model	Num. Features	AUC	F1	Accuracy
audio_closed	LR	2	82.57%	76.76%	90.04%
audio_open	kNN3	1	83.33%	80%	85.71%

Table 15: A sample of some of the higher scoring results from the fall Student-Data collection's audio data

4.3 Time Series Experiments Results

Time series were constructed using the parameters shown in Figure 44.

```
--modalities call text ^
--directions All In Out ^
--days 14 ^
--intervals 4 6 12 24 ^
--drop 2 ^
--cutoff 10 ^
```

Figure 44: Parameters used to construct the times series.

Table 16 shows the number of participants that had at least two texts or two minutes of calls in a given direction within 2 weeks of the survey completion date. There were 312 participants total. Figure 45 shows the PHQ-9 distribution for participants used to make text time series versus participants used to make call time series.

Modality	All	Incoming	Outgoing
Text	295	290	99
Call	212	182	197

Table 16: Number of participants in time series.

Next, visualizations for the time series were created. After creating the graphs and organizing them into folders, we determined that graphs showing 24 hour aggregation interval were the best for visualization, as they were less cluttered and showed trends better over time. Refer to Appendix E to see the

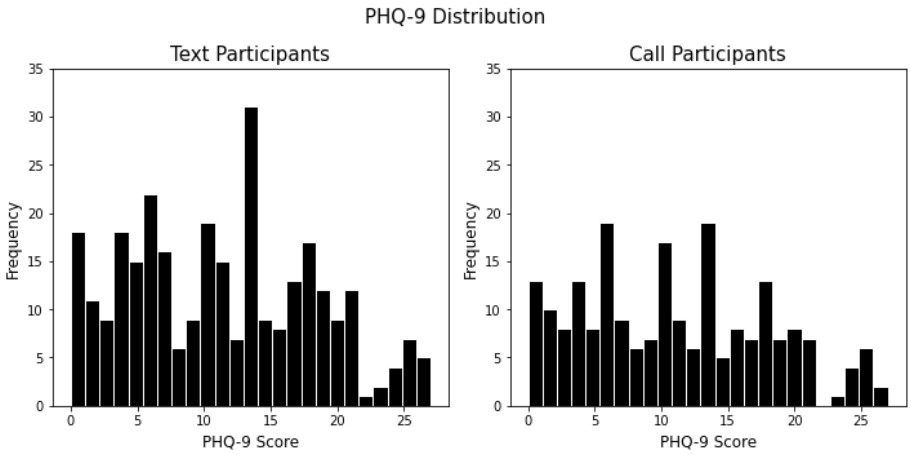

Figure 45: PHQ-9 distributions for participants whose data was used to make text and call time series, respectively.

visualizations. After developing and organizing the text and call graphs, we tested the usability of the graphs. We found it challenging to visually identify is someone was depressed from the graphs, so we concluded that the graphs are better for visualizing predictions while machine learning models will be better for classification.

It is important to utilize the colors and line styles for simplicity and clarity since the graphs will be used to visualize results from the machine learning models. While more data points may be more helpful for the machine learning model, less busy graphs, such as those with 24 hour aggregation intervals will be easier to interpret. Overall, the time series visualizations are a useful tool for data visualization and will be a helpful supplement to the outcomes of the time series machine learning models.

The first machine learning experiments that were run on the time series used kNN classifiers with DTW as a distance metric. These experiments are referred to as the time series experiments. Figure 46 shows the parameters that were used in these experiments.

```
--modalities call text ^
--directions All In Out  ^
--days 14 ^
--intervals 4 6 12 24 ^
--variables counts average_length unique_contacts ^
--samplings up down ^
--numsplits 100 ^
```

Figure 46: Parameters used in the time series machine learning experiments.

Figures 47, 48 and 49 show the result of these experiments for the average

length, counts, and unique contacts variables, respectively. The plots show the highest performing model for each modality/direction/interval pair across the balancing techniques that were used (upsampling or downsampling). Tables showing the other performance metrics for these experiments can be found in Appendix F.

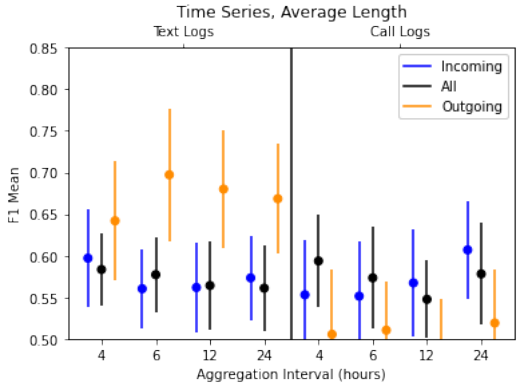

Figure 47: Mean F1 score for experiments with the average length variable. Error bars are standard deviation.

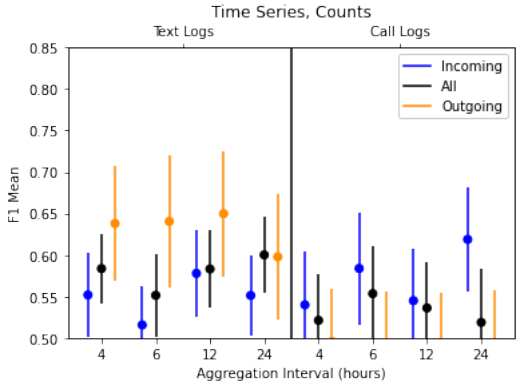

Figure 48: Mean F1 score for experiments with the counts variable. Error bars are standard deviation.

It can be inferred from these visualizations that outgoing text time series performed the best in these experiments. There do not seem to be any obvious trends among the other modality/direction pairs; the best one varies depending on the variable and aggregation interval being tested. In addition, there does not seem to be an aggregation interval that consistently performs better than the others.

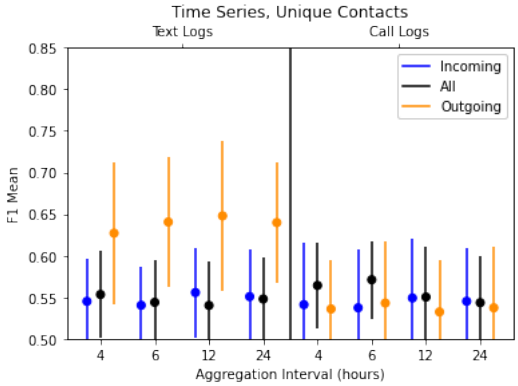

Figure 49: Mean F1 score for experiments with the unique contacts variable. Error bars are standard deviation.

In order to test whether the observed difference between outgoing texts and the other modality/direction pairs was significant, two-tailed t-tests were conducted using the scipy library. For each variable and aggregation interval, the results from the outgoing text time series were tested against all the other modality/direction pairs from the same variable and aggregation interval. Every t-test except for one (variable: counts, modality: text, direction: all, interval: 24, t-statistic = -0.27, p = 0.79) was significant at the 5% level, indicating that there was a significant difference between outgoing text time series and other modality/direction pair time series with the same aggregation interval and variable. Tables containing the results of these t-tests can be found in Appendix F.

Having established that outgoing text time series performed the best in these experiments, additional two-tailed t-tests were conducted to compare the outgoing text results across the three variables. The result from each variable was compared to the results from the other two variables with the same aggregation interval.

When comparing the average length variable against the other two variables, there was a significant difference at the 5% level for the 6, 12, and 24 hour aggregation intervals, but not for 4 hours. When comparing the counts and unique contacts variables, there was only a significant difference at the 5% level for the 24 hour aggregation interval. The results of these t-tests can be found in Appendix F. Based on the plots and the results of the t-tests, we can conclude that, experiments using the average length variable generally performed better than experiments using the counts or unique contacts variables, and the counts and unique contacts variables performed comparably.

Next, machine learning experiments were conducted using the TSFEL features that were extracted from the time series. These experiments are referred to as the TSFEL experiments. Figure 50 shows the parameters that were used in these experiments.

Figures 51, 52, 53 and 54 show the result of these experiments for the average

```
--modalities call text ^
--directions All In Out ^
--days 14 ^
--intervals 4 6 12 24 ^
--variables average_length counts unique_contacts combined ^
--methods LR SVC KNN RFC ^
--samplings up down ^
--numsplits 100 ^
--maxpc 15 ^
```

Figure 50: Parameters used in the time series feature machine learning experiments.

length, counts, unique contacts, and combined variables, respectively. The plots show the highest performing model for each modality/direction/interval pair across the balancing techniques, methods, and number of principal components that were used.

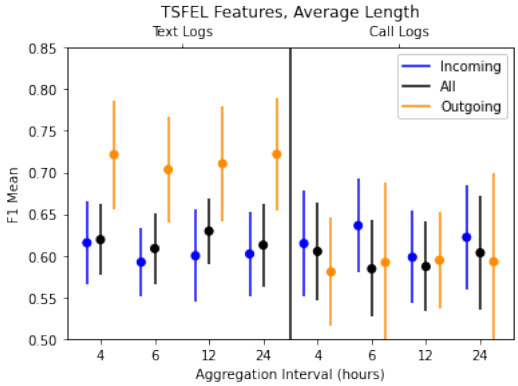

Figure 51: Mean F1 score for experiments with the average length variable. Error bars are standard deviation.

Similar to the previous experiments, outgoing text time series appear to perform the best using the TSFEL features. For the other modality/direction pairs, the best performing one varies depending on the variable and aggregation interval. There does not seem to be an aggregation interval that consistently performs higher than the others.

Again, two-tailed t-tests were conducted in scipy to determine whether the observed difference between the outgoing text time series and the other modality/direction pairs was significant. For these t-tests, all results were found to be significant at the 5% level. A table containing the results of these t-tests can be found in Appendix F.

More two-tailed t-tests were conducted to compare the outgoing text results across the variables tested. Each outgoing text result was compared to outgoing

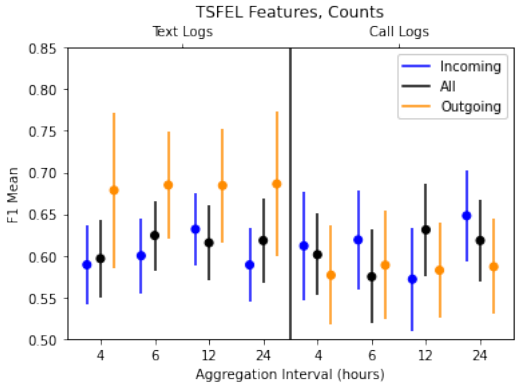

Figure 52: Mean F1 score for experiments with the counts variable. Error bars are standard deviation.

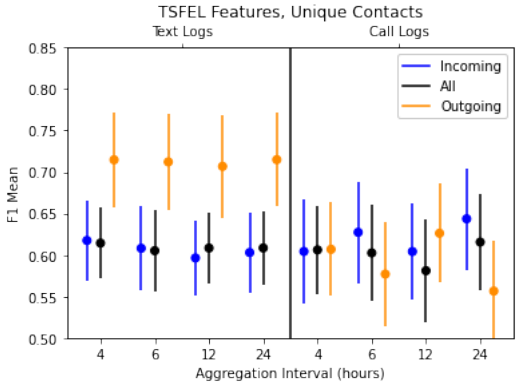

Figure 53: Mean F1 score for experiments with the unique contacts variable. Error bars are standard deviation.

text results from the other variables within the same aggregation interval. There was not a significant difference at the 5% level for any aggregation interval between the average length, unique contacts, or combined variables. There was a significant difference at the 5% level for all aggregation intervals between the counts variable and the other variables. A table containing the results of these experiments can be found in Appendix F.

Finally, two-tailed t-tests were conducted to compare the results across the two sets of experiments. For each variable, the time series experiments and TSFEL experiments from the same aggregation interval were compared. The combined variable was not used in these tests because it was only used in the TSFEL experiments. Every test was significant at the 5% level except for the average length variable with the 6 hour aggregation interval (t-statistic = -0.61,

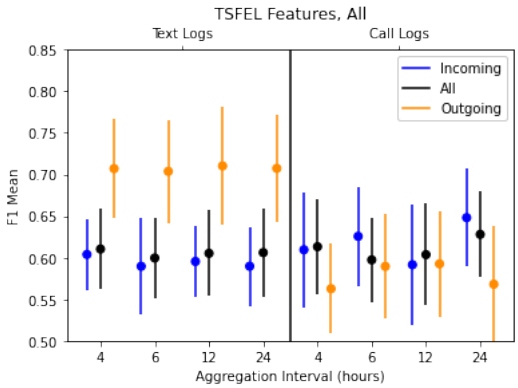

Figure 54: Mean F1 score for experiments with the three variables combined. Error bars are standard deviation.

p = .54). A table containing the results of these experiments can be found in Appendix F. Based on the results of these tests and the visualizations, it can be concluded that the TSFEL experiments generally performed better than the time series experiments.

In summary, the results of these experiments showed that time series constructed from outgoing text logs are promising in predicting depression. Machine learning experiments using feature extracted from the time series performed better than experiments using the raw time series. In addition, time series constructed from outgoing text logs with information about the average length of the texts and the number of unique contacts perform better than a simple count of the texts.

4.4 Stereotype Threat Priming Study Results

4.4.1 Stereotype Data Collection

The data collection and analysis for the Stereotype Threat study is still in progress. We collected 237 total entries so far. 131 participants completed the Control Version and 106 participants completed the Test Version. 139 Men 84 Women completed our survey. 7 participants identified as Other.

4.4.2 Text Message Collection

The results of this secondary study can be seen in the table below. Not much can be concluded from these results, as there was not extreme variation in any pay incentive

Count of prolific	Column Labels		
Row Labels	extra	no pay	Grand Total
0.05	51	11	62
0.15	35	5	40
0.25	31	5	36
0.35	30	3	33
0.45	27	4	31
$0.25 - Old	23	12	35
Grand Total	**197**	**40**	**237**

Figure 55: Table of Text Results by Pay Incentive.

5 Conclusion

Throughout this project we were able to analyze data collected in previous projects, as well as data collected throughout our project through visualization, analytical queries, and time series experiments. This analysis guided us to conclusions on what data will be best suited for training our machine learning models. In addition, we modified the website and mobile application based on data collected, pages that had technical difficulties, and modalities that participants didn't complete. We have continued to do research on how to prevent bias in our application with the Stereotype Threat Priming Study. All of these updates have allowed us to build upon our machine learning pipeline, which includes the tweet pipeline. Future projects will be able to use our data collection platforms, machine learning pipeline, and conclusions to achieve the long-term goal of this study - evaluating mental health through text prompts, phone data, audio prompts, and social media usage. This work can hopefully be expanded upon and improved to further develop methods for detecting mental health conditions such as depression, and make intervention and treatment more responsive.

5.1 Data Collection

Over the course of this project we were able to develop efficient web and mobile surveys with which to survey students. Several iterations were developed to survey both students as well as medical patients, with varying modalities and capabilities between the two. With these multiple survey versions data from over 400 participants have been collected and stored for analysis.

Future Work In addition to developing Android applications for both student and longitudinal data collections, other avenues for mobile data collection were also explored. This not only included the potential advantages and modalities of an IoS application, but the capabilities and feasibility of a research platform known as Beiwe to develop it along with an updated Android application. Both of these options were compared and contrasted with the capabilities of the existing Android applications in order to determine if either was a useful and viable addition to data collection. Beginning with the possibility of an IoS application, the various permissions and data accessible to developers mostly mirror that of Android. With user permission, the application would be able to collect text messages, message metadata, location, contacts, and calendar data. Additional modalities that could be implemented into an IoS application along with an Android application include motion, bluetooth peripherals, as well as access to media files. The only major limitation with IoS is that one would not be able to access and pull the content of text messages from users' phones, instead being limited to the message logs themselves. With this exception, the potential application would be capable of easily replicating all of the vital functions of the current Android application. Where the two platforms begin to differ is with the additional data collection tools that are offered to IoS developers. The Sirikit

package, for instance, could potentially provide the data collection application with recent search requests and interactions with Siri, with data being stored for up to two weeks (*SiriKit - Apple Developer Documentation*, n.d.).

Another tool with great potential for data collection is Apple Health. Apple Health is an application that uses machine learning to help users monitor and improve their health. It tracks users exercise, vitals, and other health information which is presented to them through interactive charts. The application can link to several health-related devices, most notably Fitbit to further collect and track this information. This helps further supply information on the users' specific activities, including walking, running, and sleeping as well as more advanced tracking of users' heart rate and body measurements. With users' permission, other applications can access the Health application and the data stored within it for their own purposes (*Reading Data From Healthkit - Apple Developer*, n.d.). Once this permission is granted, these applications can access a detailed account of the users daily activity and step count. This data could potentially be used to examine the correlation between the users' PHQ-9 scores and their daily step counts, as well as be used for a number of other studies.

In addition to looking into the feasibility of an IoS application, the Beiwe Research Platform was also investigated as a potential option for future data iterations of the data collection applications. The Beiwe Research Platform "consists of three cloud-based components for collecting data, managing studies and performing data analysis"(*Beiwe Research Platform*, n.d.). It allows partners to schedule, run, and analyze data collections from users of its applications. The open-source code provides the structure to create and schedule studies using not only questionnaires but also by utilizing the many sensors of the phone, such as the accelerometer and gyroscope. The back-end makes use of Amazon Web Services (AWS) cloud computing infrastructure and is used to manage studies and collect data. AWS Elastic Beanstalk is used to automatically handle the details of capacity provisioning and load balancing, making the application easily scalable. The data analysis pipeline performs data pre-processing, checks data quality, transforms data, carries out imputation, and computes summary statistics of interest(*Beiwe Research Platform*, n.d.).

Unfortunately, the current code on the open-source repository is somewhat lacking, containing only the already exported Android Studio and IoS projects, with no direct indication of a staging platform with which developers can create a base project that could then be exported to both operating systems. Additionally, several classes and modalities are in apparent need of further testing, as several classes are duplicated with many functions being commented out as inoperable or potentially leading to crashes(*onnela-lab*, n.d.). Another cause for concern is the potential cost, as the AWS services utilized can have a considerable price tag. While no exact breakdown of the pricing is given an example study of 6 months of data collected from 25 study participants each for a 12 month study ended up costing $12,088 (*Beiwe Research Platform*, n.d.).

In conclusion, an IoS application would not only increase the amount of potential users of the data collection application, but would also allow for the

potential collection of new data types. As such an IoS application would certainly be a good route to pursue in future data collections. As for the Beiwe platform, the potentially cost and current lack of information regarding a proper staging platform with which to simultaneously develop both a modified Android application and a corresponding IoS application, simply looking into recreating the current application in the IoS platform would seem to be the ideal route to take at this point.

5.2 Stereotype Threat Priming Study

At this time, we can not determine whether we were able to trigger stereotype threat by showing some participants a fact regarding gender and depression. However, we have collected data and text messages from nearly 400 participants that can be used for future analysis and to train machine learning models.

5.2.1 Stereotype Threat

Because data is still being collected for this study, conclusions will be drawn after the completion of data collection.

Future Analysis After collecting data, analysis can be done on the audio prompts and text messages, as well as visualization on the stereotype threat and demographic data. Future visualization research opportunities may be on how humans versus computers analyze data and what features are most important. This data collection will continue to provide data that will enhance machine learning models by providing modalities associated with depression and anxiety scores.

5.2.2 Text Message Collection

There was not much to gather from the results to make a statistically significant conclusion about how much incentive people need to share their text messages. However, we did determine that all women who were offered $0.35 shared text messages. In future collections, where we only collect data from women+, we will offer an additional $0.35 to share text messages. For future collections where data is being collected from both men and women, we will offer $0.15, because overall, it offered a balance between cost and enticing participants to share their text messages.

5.3 Feature Extraction and Machine Learning Pipeline

5.3.1 Feature Extraction

Future Work Future work for the feature extraction program should center around two areas. The first of these is completing the feature extraction pipeline generalization process. We faced difficulties in properly extracting data with different formats from the databases, specifically the winter collection's database

which used JSON formatting for the records' content. The second area of focus should be expanding the supported modalities. Specifically, work can continue on fully integrating tweet analysis and performing experiments to analyze the usability of different tweet-specific features. Additionally the category of 'GPS features' may benefit from specific efforts to find ways to use the GPS data that has been collected.

5.3.2 Machine Learning

Future Work Future work for the machine learning code will continue where this year left off, with incorporating new methods into the code and experimenting further on which of those methods improve prediction quality. One specific area to focus on is the addition of max voting and combining modalities, to see whether or not a max voting strategy can improve the quality of predictions.

5.4 MQP Experience

We believe that this project was not only a success but also a good learning experience. Each of the members of the team consistently and effectively contributed to their individual parts of the project while also lending help to other team members when necessary. The MQP was completed with no real conflicts between team members or major issues. Additionally, the team deemed that the progress made throughout the 3 terms was more than satisfactory, and not only left us with useful data from which we could draw many conclusions, but also a more solid framework which future MQP groups can use to conduct future data collections. With multiple versions of our survey and a more refined database schema, the project can be continued and retooled for a variety of studies. Ultimately, this project provided us with experience in app development, machine learning, data analysis, and several other areas while allowing us to contribute to a meaningful project.

6 Appendices

6.1 Appendix A - Tables of Accomplishments

6.1.1 A Term

Table 17: A Term Accomplishments

Bruneau, Connor	1. Worked in Android Studio on Data Collection Applications 2. Modified the Student Version of the EMU Application for Data Collection 3. Worked to develop Longitudinal Versions of the EMU Application with functionalities such as reminders, new questions, and increased data collection.
Caouette, Hunter	1. Served as team leader for meetings, responsible for administrative duties 2. Designed new file system structure Feature Extraction/Machine Learning Pipeline code. 3. Updated existing pipeline code to make it compatible with new file structure
Kayastha, Rimsha	1. Updated website with important project information and deployed a redesigned website survey 2. Developed the front-end and the server-end for four versions of the website survey: test version, student version, and two longitudinal versions 3. Conducted Think-Aloud sessions for usability testing of both the mobile app and website surveys 4. Transformed feature extraction code to perform on tweets

Melican, Veronica	1. Served as team scribe, took minutes at team meetings. 2. Created code to construct time series data from call and text logs. 3. Created code to run machine learning experiments on the time series data with kNN classifiers. 4. Ran machine learning experiments on WPI's HPC cluster, Ace. Made guide to assist team members/future MQP teams with running jobs on Ace.
Reisch, Miranda	1. Performed application analytics for the EMU Student dataset using SQL and Excel to increase completion rate of surveys 2. Used Python to develop and improve time series graphs for internal visualization and machine learning 3. Used SQL and Tableau to make a PHQ-9 data dashboard on the EMU Student dataset to determine amount of data for the machine learning pipeline

6.1.2 B Term

Table 18: B Term Accomplishments

Bruneau, Connor	1. Modified the Student Version of the EMU Application for Data Collection 2. Worked to develop Longitudinal Versions of the EMU Application with functionalities such as reminders, new questions, and increased data collection. 3. Helped develop data cleaning criteria

Caouette, Hunter	1. Completed A-term changes to the feature extraction pipeline code 2. Worked to make feature extraction code compatible with this year's data schema 3. Began work on GPS feature incorporation 4. In-progress Jupyter Notebook interfaces for pipeline steps 5. In-progress documentation of the pipeline
Kayastha, Rimsha	1. Team Leader for the term 2. Debugged surveys errors and helped move the website survey to a new back-end logging system 3. Worked on the feature engineering code for TweetsPUL data; Performed a few experiments with Empath; Added hashtags analysis and aim to add mentions analysis 4. Actively participated in the data collection 5. Regularly checked on the website, the qualtrics feedback survey, and survey code 6. Made some updates to the website, such as adding a new page, to refine it
Melican, Veronica	1. Added command line arguments to code from A term to make code easier to run. 2. Continued running experiments with time series data and kNN classifiers on Ace. 3. Created visualizations for machine learning results. 4. Researched Python libraries to extract features from time series. 5. Used the TSFEL library to extract features and ran machine learning experiments with these features.

Reisch, Miranda	1. Determined usefulness of time series visualizations for visual use and machine learning purposes. 2. Researched Stereotype Threat and how to trigger it in our Android Application 3. Set up Prolific and researched filtering and execution for our purpose and Android app 4. Researched and determined sample size in order to be statistically significant in our results 5. Made changes to the Android application in order to have a control and test version

6.1.3 C Term

Table 19: C Term Accomplishments

Bruneau, Connor	1. Modified the Student Version of the EMU Application to implement new data schema 2. Researched the capabilities and feasibility of developing new data collection applications for the IoS platform, as well as doing so through the research platform Beiwe 3. Served as team scribe in B term, taking minutes during meetings
Caouette, Hunter	1. Worked to complete generalization of the feature extraction code for compatibility across databases 2. Conducted machine learning experiments fall data collection audio 3. Laid groundwork for experimentation on winter audio, fall and winter text data

Kayastha, Rimsha	1. Implemented data cleaning for the student data collections 2. Added counts of hashtags and mentions, and hashtag analysis to tweet feature extraction process 3. Analyzed similar websites and updated EMUTIVO website to make it more accessible
Melican, Veronica	1. Updated cleaning strategy for time series data. 2. Re-ran previous experiments with updated cleaning strategy and ran additional experiments on Ace. 3. Analyzed time series machine learning results by creating visualizations and conducting t-tests. 4. Cleaned student data. 5. Conducted analysis on cleaned student dataset.
Reisch, Miranda	1. Ran test runs on Prolific to collect data for Stereotype Threat Priming Study 2. Utilized test runs to determine whether different incentives encourage participants to share text messages 3. Modified Stereotype Threat versions of mobile app based on general changes, results of test runs, and text message payment experiments 4. Developed code in Python to clean sort the prolific data

6.2 Appendix B: Table of Authorship

Section	Primary Author
1 Introduction	Team
2.1 Related Works	Team
2.2 Previous MQPs	Connor Bruneau
2.2.1 Datasets	Miranda Reisch
2.3 Screening Surveys	Rimsha Kayastha
2.4.1 Data Balancing and Dimensionality Reduction	Veronica Melican
2.4.2 Machine Learning Methods	Veronica Melican
2.4.3 Performance Evaluation of Classification Methods	Hunter Caouette
2.5 Stereotype Threat	Miranda Reisch
2.6 Statistical Significance	Miranda Reisch
3.1.1 Android App	Connor Bruneau
3.1.2 Stereotype Threat Changes	Miranda Reisch
3.1.3 Website Survey	Rimsha Kayastha
3.1.4 Student Survey Changes	Rimsha Kayastha
3.1.5 Longitudinal Survey	Connor Bruneau
3.2 Student Data Collection	Team
3.3 Feature Extraction and Machine Learning Pipeline	Hunter Caouette
3.3.4 Tweets Feature Extraction	Rimsha Kayastha
3.4 Time Series Experiments	Veronica Melican
3.4.2 Time Series Visualizations	Miranda Reisch
3.5 Stereotype Threat Priming Experiment	Miranda Reisch
4.1 Student Data Analysis	Veronica Melican
4.1 Student Data Analysis - Correlation Analysis	Rimsha Kayastha
4.2 Machine Learning Pipeline Results	Hunter Caouette
4.3 Time Series Experiment Results	Veronica Melican
4.4 Stereotype Threat Priming Study	Miranda Reisch
5 Conclusion	Team

Table 20: Table of Authorship

6.3 Appendix C: Student Data Cleaned IDs

6.3.1 Summer collection I

8181, 8170, 8516, 4041

6.3.2 Summer collection II

2613, 2128, 3227, 381, 7256, 836, 8650, 4782, 7505, 1811, 6510, 3920, 8663, 4353, 2623, 3064, 6658, 4598, 7912, 6548, 8279, 2843, 5229, 3830, 5245, 7569

6.3.3 Fall collection

8791, 3523, 4859, 3933, 9034, 2430, 6336, 7007, 7564, 319, 7279, 8640, 3685, 409, 6179, 1953, 5571, 7711, 396, 8479, 646, 4521, 2837, 4441, 1552, 8472, 5881, 4442, 1244, 7516, 1876, 191, 3041, 4769, 1269, 5028, 9745, 4698, 5047, 60, 3278, 1879, 3302, 8018, 7452, 4098, 7755, 1716, 552, 705, 5948, 6868, 4973, 8550, 3473, 103, 1846, 2478, 6831, 4001, 2627, 6706, 3102, 2496, 7370, 7547, 8085, 5330, 3985, 9986, 4707, 4755, 415, 6580, 7159, 3273, 518, 6390, 896, 1471, 8180, 3323, 528, 292, 850, 9934, 3662, 2121, 3250, 8918, 4549, 2222, 4879, 2525, 7974, 9754, 1056, 7612, 3517

6.3.4 Winter collection

08id9tgqbb59hkebrnnorpqau7_1609983150, 0htiuf40bp96rbpuckt4kvrcp8_1607642639, 0kat7i7l0fc05bivc4j35jkl06_1610640355, 0obffpan1ijpil0k5u3nbbb2oj_1607051003, 0tddacnt6flj7o013ceh28ju09_1607339125, 0vkpo73bvfncl2rni64br5o0gl_1607510768, 1581r9l83i654r5bd93j2ierhv_1608003341, 180oenlkf47u5kpof6b77egm11_1609899907, 1q53bujfiuo3r9dt0fdsqgncq9_1608188073, 1rpakrqs25olmvdee63ucsruj5_1607019351, 285ek0t5e1ioddol8qsieaqufb_1607440988, 29cm7rvd60vhcnugonb5abdmj8_1607135820, 2mtge62klaqeplapg5a597j4u0_1607494599, 2q1emqkbf8q3p7k5bcccnadqbu_1607269923, 2re0daffdihisih8s7b78l4am3_1608492986, 2t0avt8jdogppjbaaclo8f8imf_1608850996, 2tgoj38jgav9rbksj8ek0k8h2h_1608707232, 322qhk8vjh5rnv8ggm3f27emdc_1608683584, 322qhk8vjh5rnv8ggm3f27emdc_1608683738, 3346ksbf63fhg72biffbc9pccl_1611424664, 3f8ckdhdt7upcrpepm2vr9mhfs_1607087749, 3g4lfra6i99lkfs5gaav8t5rhs_1607495239, 3pnmi1ks3r1ccg9dkppf5uh8qh_1608663032, 42r5nr1ckfknf3a0r27daippk3_1608849324, 4eu4k80dirpvlvf662sdv79fef_1607133044, 4ijigm6cc73s81c8lvf1ql28t2_1608850448, 4jko6et25s1fqdgntd4fpmabsn_1611575361, 4lop4f8pqk445a2ruq8monf20m_1610110670, 4roa0gr360v19rtloc89dkme2k_1608582258, 4sahqqd9tkgkosadj8ti96nkca_1607510222, 4u5670ufsqocvtqa1gbcqedk6j_1607691623, 5172giagak00430pnqsn8on6k2_1608053349, 5avslpn84o64e5ruguqmg1o5h0_1607536408, 5fae4labo7381ij71sf6f141ir_1608051417, 5kjjto4rpeou61q0b4rmt1ab0v_1607929944, 5q177dig1v233j4nn44lae8gci_1608586814, 5s1eeegmeigg04l8d5npgi2ohd_1607124379, 5su3bopnaq8647rsdfc2llmhev_1607293726, 5v9m2l0lqr8ia0bkqg9ih2nj2o_1607560754, 648pl1m34k895ao1tf4ssd2tj6_1608359052, 64u4atmjkbo4nvuik8okj4md1o_1607807159, 6g9o5651jbmgfepl7pol7fvhj5_1607413039, 6hfias0r3sihbn7re8181kq9ne_1609941585, 6hqfinb78a6hdlq7imiamu4l1b_1608564004, 6v3321ti091r9vet0rvv5mkpl0_1607498554, 7305o0qcer7tabdoa4j93tu015_1608589576,

7bt089ljon8ccuol1nctte3fh6_1607674683, 7iueicd979gj7ovjed7icvvp3d_1609142183,
7iueicd979gj7ovjed7icvvp3d_1609166843, 7kinfg9ea5qoheboviuq7af0rd_1608595561,
7p5tsmfoeq1f7v2enguu081ndc_1608741452, 7t5hl043t35083mqvhapincjl9_1608059746,
81vpe8mhh2kdsst3oekp5vei89_1609887404, 86d5etju6mcl6v1dgqbpmf5emf_1607262842,
8und1lft6oamd8fiipi8k9p18m_1609771771, 8vdjssg10h8601ltri42rk26uv_1608200317,
912l96o76lj47lh5gest3hslhl_1607350992, 98p77pr7sb0cael5n287v6ej58_1607659758,
a1mosvebn5j0vv47tcbjgds4n1_1607738757, a381ftku6q3ao05vmj4gi5995h_1607712682,
a5h5783s2iujp6kftqk26lq9q5_1608588581, a6or2fcmb721ll439hj8jq1dck_1609027319,
aa4cdcs2f02e6fo67mv6kts9fk_1607266081, af9q4cgult6d60g416l7rnkmt8_1608201510,
agauc8l3e03a4lmsmib77lrogd_1611517276, ahdpndo70fcqh6151g0g83bqvc_1608624428,
akls0l83m6rvl002d6uajbij86_1609049435, and7p199rrh4vsebb1hgh62l0i_1607397061,
aqes11nt8d5492t871ure08eo1_1607131299, bff62uqk1qu97dg3he49elf27b_1607928177,
bgqatj7a6k6b7mcb3igp3oetap_1607939718, bgqatj7a6k6b7mcb3igp3oetap_1607939838,
bht0hbelv3avlceqdd4qhn1ofn_1607217921, bslvvi5n73smbia4o209gjcs0o_1607712777,
c1jgg5m678d9p0f8p3ca3dq0eb_1608631410, c2iqmer8gqq966k2ic87q87t24_1607129580,
c9neuh1t2t039mk02vspdicub9_1608586953, cddhkvfeqom717k13j4sbqimkd_1609889389,
cdjb5pos4c4nnhhih6lf41v1en_1608702785, cjitfh4ilnppisca6l5ek6hlt5_1610818662,
cngrt56g8vffoq8l9fmlbu41f9_1608168856, coba3e420or6q8dj11pphjgvi5_1607088730,
cq6k78t0h5e5uen59460kaeers_1607289708, d1gob06l188nfcqb8gqsm5g40a_1607129044,
dgeuv3obto2n1f2gh83862rd6m_1609890222, dhqmdgd9l1bfedc365u7i0rcfh_1607206195,
dmpqnkftli31nkju449rt84772_1607712793, e27odqb1g9g58g6qht52fpphv6_1607193886,
e46dtotn5a4snmc38122u1bllb_1608506424, e88t2uvhlut48900bak3scdaom_1608086804,
e8g3klmd7btoc1o95iagvaov55_1608537399, eo7e8atgqtr4audlr4j4467sfo_1609353016,
err8ov7hha4dqmdnu6b40l4980_1607315703, f2srdqam2cb2ejnlkdr4rufgla_1607810287,
f33uh8negkivgfok4ahqnoqv19_1607291670, f8b7v6pprejtcnfbi2inoengss_1609887167,
f8vdqjgd22tv26041t4fishiav_1607921306, fd3sopd4fi91cp34090mqcp58l_1608416516,
fkrvumqapihr5bncvhs2hv2ceu_1607772081, g0k80qcoe5alf5b643h6c2aqdq_1608672132,
gchnkhv04nsggthdcrhqllisp1_1607022963, givndq01n4r8o0m5luiavovd4p_1607572897,
gmo2ro4i3s6fopc9rfsi8fqils_1607696074, gnp62mld0qmgfl24smublmpu2r_1609082904,
gns1sc99ll47iqijso5fk44svj_1607927243, h8bdmn8icjrnckn6dfvmvnhh7e_1607133218,
h9mpbk9h8aktcvklqjq779f9b0_1608586899, hiajkbdagvb26ikc8qni9fp3mt_1608587203,
hq0bm08sdphuthouvhds84ta8m_1610791060, hq4hs2m3qbplvihfa1mfn0qmm6_1608770486,
hqe79v4ddp522tk7a0et5hrb4v_1608992344, ht1jap6vsrc6jcojh61cgmlif3_1607159307,
i2392ugm6g5dncgev2cjuujglo_1607315333, iel1r3nvchgd631pfch5tmjdo5_1608200497,
ioum24edk2e8f6l3ve5r10so6p_1609887249, j5mj357307fnoti0at3f6sttbg_1609893292,
j7g9714u36q7l29plslusf4sm1_1607010270, ji150e0sul134vsj8vd34mkl2d_1607097951,
k2kq15mbpheh0na0rsc4f4rtj6_1607807806, k483fi1reaj5mept92i54f09q0_1608048050,
k49hfnj5lgcmllu2593dncs22u_1607764720, k4j4lfg3rv6en0obkekhlr780e_1607802179,
kn3c0c74fvb2f1h650f2lhk2td_1608470962, l0qi1tniekfuuk5drk6vmti3t0_1607719324,
l5e8i8qhqmfaa4443tfdv7aekq_1609903202, l713j7fikt5eud46tb56abooujq_1609890530,
lk83k64c8r8p2s5te3b3ptu9q7_1608591490, lvpmdv4qferbkmfaobq5iuv1q1_1607273026,
mm1n3f960npi7vp399crgelgi0_1611704179, mnpqerrstsligobdribsarb9jc_1607104225,
n2ivvb1mhmvpp3o2las0cas26s_1610630377, nbfgs522tb8imirk31bo2cr8vf_1607270186,
ndcocragfd9cm461as5eb5uig2_1607257348, nr91ql0kmguvq1u0pa8sv6jsk2_1607040811,
nv979ivg79k8479ct0ak620mba_1607734901, o5g12rjjatpcmb37n81sicvcdb_1607968838,
oa9n8hl8pi3gehe11n9i1a0kv1_1609175623, ocog5algsbr0osql9degqbtcb7_1607410780,

ofj2acvboj8e629ggjo3r375o0_1608335906, oig59lfv3j3oa1nik1re4aebip_1607051040,
omrn9g5053b66vm90jvbosrgdm_1608596696, ool88gn72r48n3no5q2kg31usu_1607795480,
opcsotum0c5v02kg3jotni4ono_1608853150, or3fc7fcbvj5n86ntuh366np7g_1607799213,
p139pdddaiqlt5d1mt1e72ua9g_1607295286, p23hcrbrktnv5ut7qilcrk6btl_1607639340,
p2ov6s003jvehqm775fg4fr2hh_1610109929, p518sefhnpctp9l0iq8dehufda_1607134906,
pdrhme5r449edme3172ullht4d_1608495626, pg9d2v1078tcu6vqhhbqmst2ju_1608920128,
phlafssihhs1canuglsf8mokdu_1607276888, po3bpjfaejfvnputg64u2pr8oj_1610381937,
ppngh6l68s6o2s6cng63uetk1u_1609174124, psne873i6195fvl1lg7vmnl015_1608242917,
q3m8ds9lhlfdaqcukr8tpb91gj_1608335387, qavgqlm9o861u68i5fasjdljac_1609154262,
qfrbik13hrsmd3earo7odjfc97_1610380419, qhbf2jgv59fabhgpsnm1jlqten_1608917024,
r3lacs313e8ubtag5jocjfo5t6_1609111416, r4jh3rr7ju4u2jr6fu9hphg9kj_1607636681,
rll9jv0k5j0rsknr82n8k7is2g_1607785646, rnr60gfhqrduauhmtvcl1oqnma_1607365865,
rvrc1pie1e3mlk9j3oep3rdihm_1607809058, s026s1v37bf1b7igsjgki4t6e6_1609473849,
s4cnu8r49vrp63r28i2ugi67p9_1607118643, s91rr6ql113vcmhs5qr3oovrdo_1608588103,
s98sjpcmd55co1nrhicnpkj9e3_1607453496, s98sjpcmd55co1nrhicnpkj9e3_1607497867,
sjcpj6dnm7u87mabo71dm4du8i_1608062276, sr3ekeqe5ebjpdhg16qorbs2cq_1608061691,
t6ootfp2i1jdvnq8ioa5bdaalo_1608572299, tdkrdtnu1p5bajbg0n6p643uia_1607045076,
tj3eafd1jph5tfie4ag3t7tbt6_1607555727, u3f4e0j9if6dq3qvj57uvjhhnu_1607046006,
u9ui0e8ib0mkj5eim3j99qfdji_1608487726, uae07i8lmg7iqbr95uasfs9b6t_1607712704,
ucukpnpqp6jg9fi0id80bknva9_1607891972, uepsdf3bmjiupa4jgtgausoh0q_1609052616,
ujsoh3pq8jjsnh822d65b2ij9u_1608371327, uqs58dk27qcjttln51dmk4k53l_1609256130,
uri315441h3lvud9dljpphnvj4_1607559849, uuhs0seod3cmd7pesitpje6h9f_1609888813,
v0bnb2n9o00e0nmq58v0is95c7_1607357022, v4nf3cm51vntenvuu4bcifvv6s_1608587385,
vf5dti9m8h0egobor79uedalt0_1607314413, vj4c2j0tka91mnccr7t1p7vvdm_1607712784,
vncgagfgmnb3mc49trh8hlrhv8_1608607986, vpui7spt60ibticj20cj44433r_1607368510,

6.4 Appendix D: Student Data Correlation Analysis Tables

	Number of Events	Number of Contacts	Number of Texts	GPS Submission
Number of Events	1.0	0.9392	-0.1234	0.3402
Number of Contacts	0.9392	1.0	-0.3935	0.553
Number of Texts	-0.1234	-0.3935	1.0	-0.9656
GPS Submission	0.3402	0.553	-0.9656	1.0

Figure 56: Correlation table for the summer I collection.

PHQ-9 Score	PHQ-9 Score	PHQ Ninth Question	Number of Events	Number of Calls	Number of Contacts
PHQ Ninth Question	0.3007	1.0	-0.0891	-0.0974	-0.0887
Number of Events	0.0362	-0.0891	1.0	0.8884	0.9999
Number of Calls	-0.0839	-0.0974	0.8884	1.0	0.883
Number of Contacts	0.0392	-0.0887	0.9999	0.883	1.0
Number of Texts	-0.1914	-0.0822	0.5491	0.8715	0.5392
GPS Submission	0.1806	-0.112	0.8318	0.7092	0.8325
Age	-0.3092	-0.2052	-0.16	-0.1749	-0.1592
Gender	0.0788	-0.216	-0.0344	-0.203	-0.03
Student Status	0.0492	-0.2052	0.0098	0.1679	0.0057

PHQ-9 Score	Number of Texts	GPS Submission	Age	Gender	Student Status
PHQ Ninth Question	-0.1914	0.1806	-0.3092	0.0788	0.0492
Number of Events	-0.0822	-0.112	-0.2052	-0.216	-0.2052
Number of Calls	0.5491	0.8318	-0.16	-0.0344	0.0098
Number of Contacts	0.8715	0.7092	-0.1749	-0.203	0.1679
Number of Texts	0.5392	0.8325	-0.1592	-0.03	0.0057
GPS Submission	1.0	0.4025	-0.1475	-0.3329	0.2951
Age	0.4025	1.0	-0.2011	0.0504	-0.0766
Gender	-0.1475	-0.2011	1.0	-0.0792	-0.1729
Student Status	-0.3329	0.0504	-0.0792	1.0	-0.285
	0.2951	-0.0766	-0.1729	-0.285	1.0

Figure 57: Correlation tables for the summer II collection.

PHQ-9 Score	PHQ-9 Score	PHQ Ninth Question	Number of Events	Number of Calls	Number of Tweets
PHQ Ninth Question	0.571	1.0	-0.1077	-0.0741	-0.0377
Number of Events	-0.1077	-0.0792	1.0	-0.0717	0.0589
Number of Calls	-0.0741	-0.0717	0.6852	1.0	-0.0341
Number of Tweets	-0.0377	0.0589	-0.0341	-0.0309	1.0
Number of Contacts	-0.0346	-0.0712	0.8274	0.6815	-0.0307
Number of Texts	-0.066	-0.0653	0.6266	0.8717	-0.0281
GPS Submission	-0.0316	-0.0688	0.3591	0.4257	0.0705
Age	0.1369	-0.0654	0.3708	0.3714	-0.0282
Gender	-0.0089	-0.0849	-0.2035	-0.3042	-0.161

PHQ-9 Score	Number of Contacts	Number of Texts	GPS Submission	Age	Gender
PHQ Ninth Question	-0.0346	-0.066	-0.0316	0.1369	-0.0089
Number of Events	-0.0712	-0.0653	-0.0688	-0.0654	-0.0849
Number of Calls	0.8274	0.6266	0.3591	0.3708	-0.2035
Number of Tweets	0.6815	0.8717	0.4257	0.3714	-0.3042
Number of Contacts	-0.0307	-0.0281	0.0705	-0.0282	-0.161
Number of Texts	1.0	0.4012	0.7216	0.5789	-0.2748
GPS Submission	0.4012	1.0	0.0823	0.2961	-0.2652
Age	0.7216	0.0823	1.0	0.5121	-0.3915
Gender	0.5789	0.2961	0.5121	1.0	-0.2966
	-0.2748	-0.2652	-0.3915	-0.2966	1.0

Figure 58: Correlation tables for the fall collection.

6.5 Appendix E: Time Series Visualizations

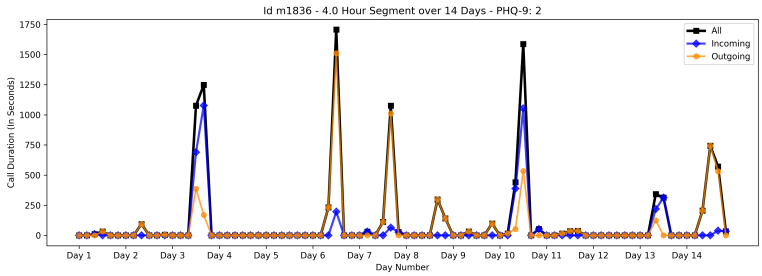

Figure 59: Time Series Visualization - Calls - Aggregation Interval: 4 Hours

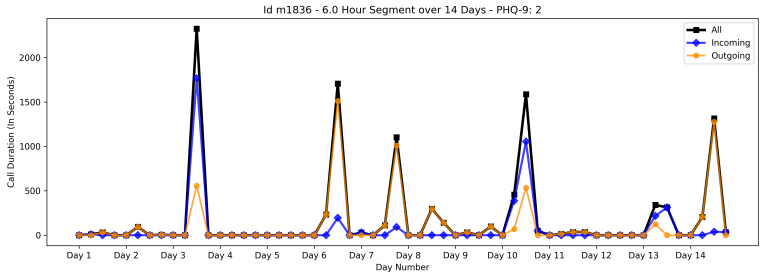

Figure 60: Time Series Visualization - Calls - Aggregation Interval: 6 Hours

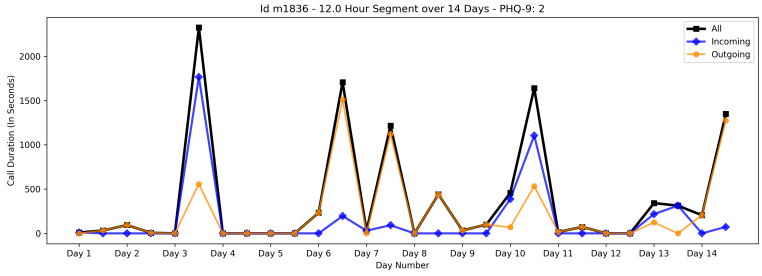

Figure 61: Time Series Visualization - Calls - Aggregation Interval: 12 Hours

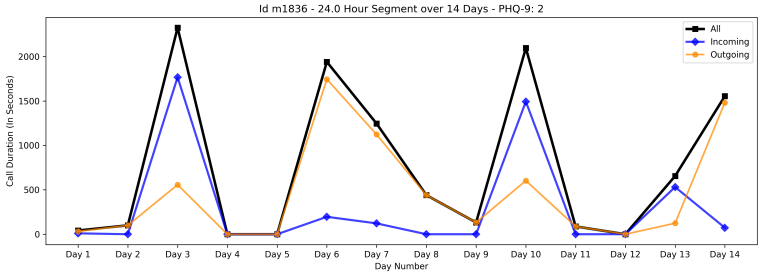

Figure 62: Time Series Visualization - Calls - Aggregation Interval: 24 Hours

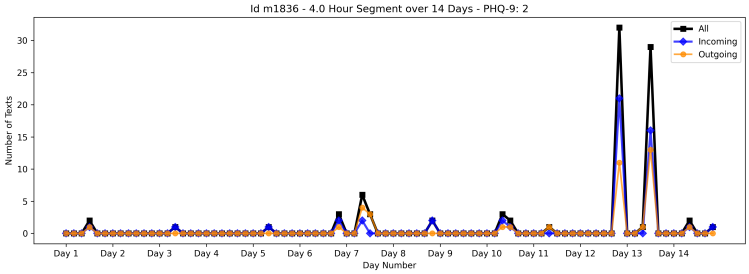

Figure 63: Time Series Visualization - Texts - Aggregation Interval: 4 Hours

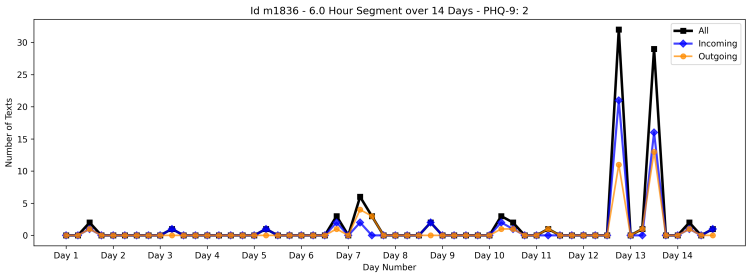

Figure 64: Time Series Visualization - Texts - Aggregation Interval: 6 Hours

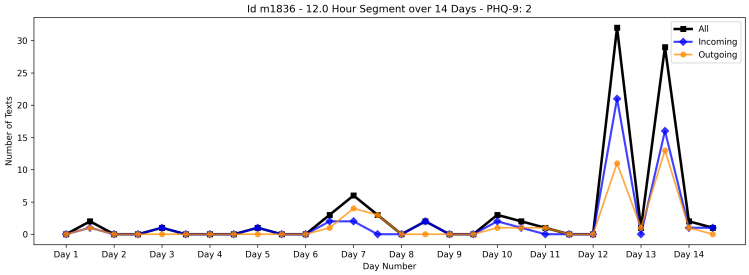

Figure 65: Time Series Visualization - Texts - Aggregation Interval: 12 Hours

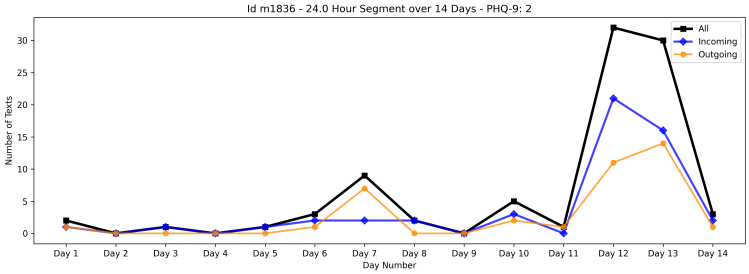

Figure 66: Time Series Visualization - Texts - Aggregation Interval: 24 Hours

6.6 Appendix F: Time Series Experiment Results

modality	direction	day	interval	variable	sampling	f1_mean	accuracy_mean	precision_mean	sensitivity_mean	specificity_mean	auc_mean
text	Out	14	6	average_length	down	0.697 +/- 0.08	0.638 +/- 0.082	0.704 +/- 0.069	0.698 +/- 0.111	0.545 +/- 0.136	0.622 +/- 0.083
text	Out	14	12	counts	up	0.65 +/- 0.076	0.587 +/- 0.082	0.67 +/- 0.077	0.639 +/- 0.103	0.506 +/- 0.153	0.573 +/- 0.088
text	Out	14	12	unique_contacts	up	0.648 +/- 0.09	0.587 +/- 0.086	0.667 +/- 0.076	0.64 +/- 0.129	0.506 +/- 0.14	0.573 +/- 0.085
call	In	14	24	counts	up	0.619 +/- 0.063	0.584 +/- 0.056	0.65 +/- 0.053	0.595 +/- 0.083	0.569 +/- 0.085	0.582 +/- 0.056
call	In	14	24	average_length	up	0.607 +/- 0.058	0.552 +/- 0.059	0.612 +/- 0.053	0.607 +/- 0.083	0.479 +/- 0.105	0.543 +/- 0.061
text	All	14	24	counts	up	0.6 +/- 0.046	0.545 +/- 0.044	0.614 +/- 0.04	0.59 +/- 0.066	0.484 +/- 0.075	0.537 +/- 0.044
text	In	14	4	average_length	up	0.597 +/- 0.059	0.551 +/- 0.054	0.613 +/- 0.05	0.586 +/- 0.083	0.504 +/- 0.086	0.545 +/- 0.053
call	All	14	4	average_length	up	0.594 +/- 0.055	0.566 +/- 0.052	0.621 +/- 0.052	0.572 +/- 0.074	0.558 +/- 0.087	0.565 +/- 0.053
text	All	14	4	average_length	up	0.583 +/- 0.044	0.528 +/- 0.049	0.6 +/- 0.047	0.57 +/- 0.057	0.469 +/- 0.091	0.519 +/- 0.052
text	In	14	12	counts	up	0.578 +/- 0.052	0.52 +/- 0.048	0.581 +/- 0.04	0.578 +/- 0.073	0.442 +/- 0.072	0.51 +/- 0.047
call	All	14	6	unique_contacts	up	0.571 +/- 0.047	0.528 +/- 0.044	0.578 +/- 0.041	0.567 +/- 0.065	0.479 +/- 0.079	0.523 +/- 0.045
text	In	14	12	unique_contacts	up	0.556 +/- 0.054	0.505 +/- 0.047	0.571 +/- 0.042	0.544 +/- 0.072	0.454 +/- 0.071	0.499 +/- 0.046
call	All	14	6	counts	up	0.553 +/- 0.057	0.533 +/- 0.048	0.592 +/- 0.051	0.524 +/- 0.075	0.545 +/- 0.083	0.534 +/- 0.048
text	All	14	4	unique_contacts	up	0.553 +/- 0.052	0.508 +/- 0.046	0.586 +/- 0.043	0.527 +/- 0.067	0.483 +/- 0.073	0.505 +/- 0.046
call	In	14	12	unique_contacts	up	0.549 +/- 0.072	0.541 +/- 0.061	0.63 +/- 0.069	0.492 +/- 0.087	0.608 +/- 0.099	0.55 +/- 0.061
call	Out	14	6	unique_contacts	up	0.543 +/- 0.074	0.507 +/- 0.065	0.547 +/- 0.057	0.544 +/- 0.1	0.462 +/- 0.085	0.503 +/- 0.063
call	Out	14	24	average_length	down	0.519 +/- 0.065	0.511 +/- 0.054	0.561 +/- 0.058	0.489 +/- 0.086	0.539 +/- 0.096	0.514 +/- 0.054
call	Out	14	4	counts	up	0.496 +/- 0.064	0.489 +/- 0.049	0.536 +/- 0.052	0.466 +/- 0.083	0.517 +/- 0.089	0.491 +/- 0.049

Figure 67: Results from the time series experiments. Only results from the best performing aggregation interval are shown for each set of parameters.

modality	direction	day	interval	variable	pc	method	sampling	f1_mean	accuracy_mean	precision_mean	sensitivity_mean	specificity_mean	auc_mean
text	Out	14	24	average_length	1	LR	down	0.722 +/- 0.068	0.673 +/- 0.073	0.746 +/- 0.071	0.708 +/- 0.1	0.619 +/- 0.135	0.664 +/- 0.076
text	Out	14	24	unique_contacts	1	LR	down	0.715 +/- 0.057	0.652 +/- 0.064	0.712 +/- 0.058	0.724 +/- 0.083	0.543 +/- 0.121	0.633 +/- 0.068
text	Out	14	12	combined	1	LR	up	0.71 +/- 0.07	0.655 +/- 0.074	0.723 +/- 0.068	0.705 +/- 0.099	0.578 +/- 0.125	0.642 +/- 0.076
text	Out	14	24	counts	1	LR	down	0.686 +/- 0.087	0.614 +/- 0.075	0.67 +/- 0.068	0.712 +/- 0.123	0.465 +/- 0.129	0.588 +/- 0.073
call	In	14	24	combined	12	RFC	up	0.648 +/- 0.059	0.595 +/- 0.054	0.645 +/- 0.047	0.658 +/- 0.096	0.509 +/- 0.105	0.584 +/- 0.053
call	In	14	24	counts	15	RFC	up	0.648 +/- 0.055	0.588 +/- 0.054	0.636 +/- 0.047	0.666 +/- 0.086	0.484 +/- 0.102	0.575 +/- 0.055
call	In	14	24	unique_contacts	8	RFC	up	0.644 +/- 0.061	0.587 +/- 0.063	0.637 +/- 0.056	0.655 +/- 0.089	0.494 +/- 0.113	0.575 +/- 0.065
call	In	14	6	average_length	15	RFC	up	0.636 +/- 0.056	0.57 +/- 0.057	0.618 +/- 0.047	0.661 +/- 0.086	0.448 +/- 0.1	0.554 +/- 0.057
text	In	14	12	counts	13	RFC	up	0.632 +/- 0.044	0.557 +/- 0.046	0.604 +/- 0.038	0.666 +/- 0.069	0.411 +/- 0.089	0.539 +/- 0.047
call	All	14	12	counts	10	RFC	up	0.631 +/- 0.056	0.578 +/- 0.054	0.616 +/- 0.048	0.652 +/- 0.086	0.485 +/- 0.098	0.569 +/- 0.055
text	All	14	12	average_length	12	RFC	up	0.63 +/- 0.04	0.549 +/- 0.039	0.602 +/- 0.03	0.662 +/- 0.066	0.392 +/- 0.073	0.527 +/- 0.039
call	All	14	24	combined	15	RFC	up	0.628 +/- 0.051	0.564 +/- 0.052	0.598 +/- 0.042	0.665 +/- 0.083	0.437 +/- 0.09	0.551 +/- 0.052
call	Out	14	12	unique_contacts	3	KNN	up	0.626 +/- 0.059	0.592 +/- 0.054	0.625 +/- 0.049	0.633 +/- 0.091	0.543 +/- 0.094	0.588 +/- 0.054
text	All	14	6	counts	15	RFC	up	0.624 +/- 0.042	0.538 +/- 0.04	0.592 +/- 0.029	0.664 +/- 0.07	0.363 +/- 0.072	0.513 +/- 0.039
text	In	14	4	unique_contacts	8	RFC	up	0.618 +/- 0.048	0.544 +/- 0.044	0.593 +/- 0.033	0.648 +/- 0.077	0.404 +/- 0.072	0.526 +/- 0.043
call	All	14	24	unique_contacts	11	RFC	up	0.616 +/- 0.058	0.559 +/- 0.052	0.597 +/- 0.045	0.64 +/- 0.087	0.456 +/- 0.091	0.548 +/- 0.052
text	In	14	4	average_length	15	RFC	up	0.615 +/- 0.05	0.533 +/- 0.048	0.582 +/- 0.037	0.657 +/- 0.079	0.366 +/- 0.086	0.511 +/- 0.048
text	All	14	4	unique_contacts	14	RFC	up	0.614 +/- 0.043	0.532 +/- 0.042	0.589 +/- 0.031	0.644 +/- 0.065	0.377 +/- 0.066	0.51 +/- 0.041
text	All	14	4	combined	14	RFC	up	0.61 +/- 0.048	0.531 +/- 0.043	0.589 +/- 0.032	0.637 +/- 0.075	0.383 +/- 0.066	0.51 +/- 0.041
call	All	14	4	average_length	14	RFC	up	0.605 +/- 0.058	0.545 +/- 0.052	0.584 +/- 0.043	0.632 +/- 0.091	0.435 +/- 0.086	0.534 +/- 0.051
text	In	14	4	combined	14	RFC	up	0.604 +/- 0.043	0.524 +/- 0.043	0.577 +/- 0.033	0.636 +/- 0.067	0.375 +/- 0.072	0.506 +/- 0.043
call	Out	14	12	average_length	2	KNN	up	0.595 +/- 0.058	0.567 +/- 0.056	0.609 +/- 0.056	0.586 +/- 0.081	0.545 +/- 0.098	0.565 +/- 0.057
call	Out	14	12	combined	3	LR	up	0.593 +/- 0.063	0.57 +/- 0.054	0.615 +/- 0.058	0.58 +/- 0.095	0.559 +/- 0.114	0.569 +/- 0.055
call	Out	14	6	counts	3	RFC	up	0.589 +/- 0.065	0.548 +/- 0.06	0.583 +/- 0.053	0.599 +/- 0.091	0.487 +/- 0.09	0.543 +/- 0.06

Figure 68: Results from the TSFEL experiments. Only results from the best performing aggregation interval are shown for each set of parameters.

variable	modality	direction	interval	t_statistc	p_value
average_length	call	All	4	5.358407174	2.32E-07
average_length	call	In	4	9.149153235	7.10E-17
average_length	call	Out	4	12.96509881	3.12E-28
average_length	text	All	4	6.988437522	4.14E-11
average_length	text	In	4	4.842646853	2.58E-06
average_length	call	All	6	12.36910876	2.07E-26
average_length	call	In	6	14.11369183	9.31E-32
average_length	call	Out	6	18.8309975	5.23E-46
average_length	text	All	6	13.12328781	1.02E-28
average_length	text	In	6	14.78231899	8.27E-34
average_length	call	All	12	15.57412183	3.13E-36
average_length	call	In	12	11.73540639	1.74E-24
average_length	call	Out	12	20.39867081	1.48E-50
average_length	text	All	12	13.0957365	1.24E-28
average_length	text	In	12	13.283	3.31E-29
average_length	call	All	24	9.999712846	2.57E-19
average_length	call	In	24	6.963184153	4.79E-11
average_length	call	Out	24	16.07709794	9.18E-38
average_length	text	All	24	12.77665049	1.18E-27
average_length	text	In	24	11.42017102	1.56E-23

Figure 69: T-test results for the time series experiment average length variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable	modality	direction	interval	t_statistc	p_value
counts	call	All	4	13.13224048	9.58E-29
counts	call	In	4	10.38060362	1.96E-20
counts	call	Out	4	15.08345131	9.88E-35
counts	text	All	4	6.71150186	1.98E-10
counts	text	In	4	10.0025059	2.52E-19
counts	call	All	6	8.866307803	4.42E-16
counts	call	In	6	5.405207815	1.85E-07
counts	call	Out	6	14.40463111	1.19E-32
counts	text	All	6	9.440054445	1.06E-17
counts	text	In	6	13.44011487	1.09E-29
counts	call	All	12	12.11179244	1.26E-25
counts	call	In	12	10.63976507	3.37E-21
counts	call	Out	12	16.04067935	1.18E-37
counts	text	All	12	7.507412512	2.01E-12
counts	text	In	12	7.859905375	2.41E-13
counts	call	All	24	7.902742372	1.86E-13
counts	call	In	24	-2.1388121	0.03367579
counts	call	Out	24	10.43729434	1.34E-20
counts	text	All	24	-0.27093908	0.78672013
counts	text	In	24	5.179175124	5.46E-07

Figure 70: T-test results for the time series experiment count variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable	modality	direction	interval	t_statistc	p_value
unique_contacts	call	All	4	6.292094379	1.97E-09
unique_contacts	call	In	4	7.580694239	1.30E-12
unique_contacts	call	Out	4	8.834679335	5.42E-16
unique_contacts	text	All	4	7.383011684	4.20E-12
unique_contacts	text	In	4	8.217126633	2.68E-14
unique_contacts	call	All	6	7.640421607	9.08E-13
unique_contacts	call	In	6	9.856226665	6.70E-19
unique_contacts	call	Out	6	9.061964	1.25E-16
unique_contacts	text	All	6	10.43254148	1.38E-20
unique_contacts	text	In	6	10.9998235	2.85E-22
unique_contacts	call	All	12	8.975071331	2.20E-16
unique_contacts	call	In	12	8.566203549	3.00E-15
unique_contacts	call	Out	12	10.52298777	7.46E-21
unique_contacts	text	All	12	10.28360467	3.79E-20
unique_contacts	text	In	12	8.760954807	8.69E-16
unique_contacts	call	All	24	10.46017916	1.14E-20
unique_contacts	call	In	24	9.773077678	1.17E-18
unique_contacts	call	Out	24	9.950207604	3.58E-19
unique_contacts	text	All	24	10.44484126	1.27E-20
unique_contacts	text	In	24	9.681621524	2.14E-18

Figure 71: T-test results for the time series experiment unique contacts variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable1	variable2	interval	t_statistc	p_value
average_length	counts	4	0.398389	0.690773
average_length	counts	6	5.029457	1.10E-06
average_length	counts	12	2.915587	0.003959
average_length	counts	24	7.021968	3.42E-11
average_length	unique_contacts	4	1.336515	0.182915
average_length	unique_contacts	6	5.092947	8.19E-07
average_length	unique_contacts	12	2.804017	0.00555
average_length	unique_contacts	24	2.912785	0.003994
counts	unique_contacts	4	0.993267	0.321792
counts	unique_contacts	6	-0.00344	0.997257
counts	unique_contacts	12	0.168882	0.866062
counts	unique_contacts	24	-3.99805	9.01E-05

Figure 72: T-test results comparing different variables for time series experiments.

variable	modality	direction	interval	t_statistc	p_value
average_length	call	All	4	13.31407359	2.65E-29
average_length	call	In	4	11.75157985	1.56E-24
average_length	call	Out	4	15.26627212	2.73E-35
average_length	text	All	4	13.08747668	1.31E-28
average_length	text	In	4	12.94542927	3.58E-28
average_length	call	All	6	13.92552726	3.52E-31
average_length	call	In	6	7.95104147	1.38E-13
average_length	call	Out	6	9.692852386	1.99E-18
average_length	text	All	6	12.46449437	1.06E-26
average_length	text	In	6	14.71835071	1.30E-33
average_length	call	All	12	14.17030631	6.24E-32
average_length	call	In	12	12.73503831	1.58E-27
average_length	call	Out	12	12.8864095	5.43E-28
average_length	text	All	12	10.23309452	5.33E-20
average_length	text	In	12	12.5243024	6.95E-27
average_length	call	All	24	12.24156264	5.06E-26
average_length	call	In	24	10.76402954	1.44E-21
average_length	call	Out	24	10.26514061	4.29E-20
average_length	text	All	24	13.001144	2.42E-28
average_length	text	In	24	14.14109557	7.67E-32

Figure 73: T-test results for the TSFEL experiment average length variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable	modality	direction	interval	t_statistc	p_value
counts	call	All	4	7.366487499	4.63E-12
counts	call	In	4	5.883930998	1.69E-08
counts	call	Out	4	9.238900769	3.96E-17
counts	text	All	4	7.901681366	1.87E-13
counts	text	In	4	8.568444546	2.96E-15
counts	call	All	6	12.87693084	5.80E-28
counts	call	In	6	7.513314539	1.94E-12
counts	call	Out	6	10.52754995	7.24E-21
counts	text	All	6	7.900649165	1.88E-13
counts	text	In	6	10.80900934	1.06E-21
counts	call	All	12	6.038746579	7.55E-09
counts	call	In	12	12.22371309	5.74E-26
counts	call	Out	12	11.43000832	1.46E-23
counts	text	All	12	8.401612374	8.47E-15
counts	text	In	12	6.492716134	6.64E-10
counts	call	All	24	6.789206688	1.28E-10
counts	call	In	24	3.708684956	0.00027051
counts	call	Out	24	9.565547486	4.63E-18
counts	text	All	24	6.760597357	1.51E-10
counts	text	In	24	9.986931942	2.80E-19

Figure 74: T-test results for the TSFEL experiment count variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable	modality	direction	interval	t_statistc	p_value
unique_contacts	call	All	4	13.89303065	4.43E-31
unique_contacts	call	In	4	13.03530361	1.90E-28
unique_contacts	call	Out	4	13.43627446	1.12E-29
unique_contacts	text	All	4	14.09179992	1.09E-31
unique_contacts	text	In	4	13.05034735	1.71E-28
unique_contacts	call	All	6	13.44735054	1.04E-29
unique_contacts	call	In	6	10.11633245	1.17E-19
unique_contacts	call	Out	6	15.90091289	3.15E-37
unique_contacts	text	All	6	14.24222649	3.75E-32
unique_contacts	text	In	6	13.59775123	3.58E-30
unique_contacts	call	All	12	14.41413855	1.11E-32
unique_contacts	call	In	12	12.16446482	8.69E-26
unique_contacts	call	Out	12	9.404480581	1.34E-17
unique_contacts	text	All	12	13.03809567	1.86E-28
unique_contacts	text	In	12	14.45087916	8.59E-33
unique_contacts	call	All	24	12.30729441	3.19E-26
unique_contacts	call	In	24	8.556701251	3.19E-15
unique_contacts	call	Out	24	19.040062	1.28E-46
unique_contacts	text	All	24	14.73554843	1.15E-33
unique_contacts	text	In	24	15.01039749	1.65E-34

Figure 75: T-test results for the TSFEL experiment unique contacts variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable	modality	direction	interval	t_statistc	p_value
combined	call	All	4	11.4547871	1.23E-23
combined	call	In	4	10.71113871	2.07E-21
combined	call	Out	4	18.02967886	1.22E-43
combined	text	All	4	12.67165453	2.46E-27
combined	text	In	4	14.08732856	1.12E-31
combined	call	All	6	13.37522755	1.72E-29
combined	call	In	6	9.112327418	9.02E-17
combined	call	Out	6	12.93046323	3.98E-28
combined	text	All	6	13.35025452	2.06E-29
combined	text	In	6	13.48716461	7.81E-30
combined	call	All	12	11.43521286	1.41E-23
combined	call	In	12	11.80843651	1.05E-24
combined	call	Out	12	12.42178746	1.43E-26
combined	text	All	12	12.04276729	2.04E-25
combined	text	In	12	13.87678032	4.97E-31
combined	call	All	24	9.603635237	3.60E-18
combined	call	In	24	6.774482205	1.39E-10
combined	call	Out	24	14.52030723	5.26E-33
combined	text	All	24	12.08794163	1.49E-25
combined	text	In	24	14.64426817	2.19E-33

Figure 76: T-test results for the TSFEL experiment combined variable. For each row, the modality/direction pair was compared against outgoing text results for the same aggregation interval.

variable1	variable2	interval	t_statistc	p_value
average_length	counts	4	3.771318	0.000214
average_length	counts	6	2.045863	0.042092
average_length	counts	12	2.722203	0.007063
average_length	counts	24	3.237337	0.001415
average_length	unique_contacts	4	0.774602	0.439499
average_length	unique_contacts	6	-1.03931	0.29993
average_length	unique_contacts	12	0.413203	0.679905
average_length	unique_contacts	24	0.79678	0.426533
average_length	combined	4	1.636397	0.103345
average_length	combined	6	-0.0432	0.965589
average_length	combined	12	0.035928	0.971376
average_length	combined	24	1.555407	0.121446
counts	unique_contacts	4	-3.30888	0.001113
counts	unique_contacts	6	-3.16407	0.001801
counts	unique_contacts	12	-2.44656	0.015295
counts	unique_contacts	24	-2.76292	0.006269
counts	combined	4	-2.57675	0.010701
counts	combined	6	-2.10844	0.036251
counts	combined	12	-2.64913	0.008721
counts	combined	24	-1.94852	0.052766
unique_contacts	combined	4	0.935578	0.35063
unique_contacts	combined	6	1.005421	0.315922
unique_contacts	combined	12	-0.36935	0.712261
unique_contacts	combined	24	0.87754	0.381257

Figure 77: T-test results comparing different variables for TSFEL experiments.

variable	interval	t_statistc	p_value
counts	4	-3.511582271	0.000551649
counts	6	-4.326617824	2.40E-05
counts	12	-3.389693605	0.000844421
counts	24	-7.666982893	7.75E-13
average_length	4	-8.241340184	2.31E-14
average_length	6	-0.606632212	0.544789944
average_length	12	-3.096534848	0.002241972
average_length	24	-5.637807902	5.88E-08
unique_contacts	4	-8.547609993	3.37E-15
unique_contacts	6	-7.414125596	3.49E-12
unique_contacts	12	-5.388717694	2.00E-07
unique_contacts	24	-8.145047474	4.19E-14

Figure 78: T-test results comparing different variables between the time series and TSFEL expeirments.

6.7 Appendix G: Machine Learning Results Charts

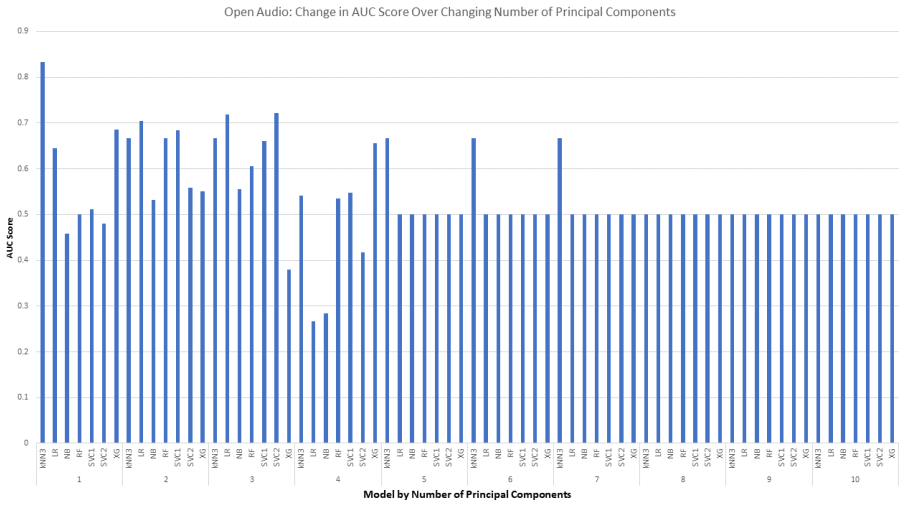

Figure 79: The changes in AUC score for Open Audio across models as the number of principal components increases.

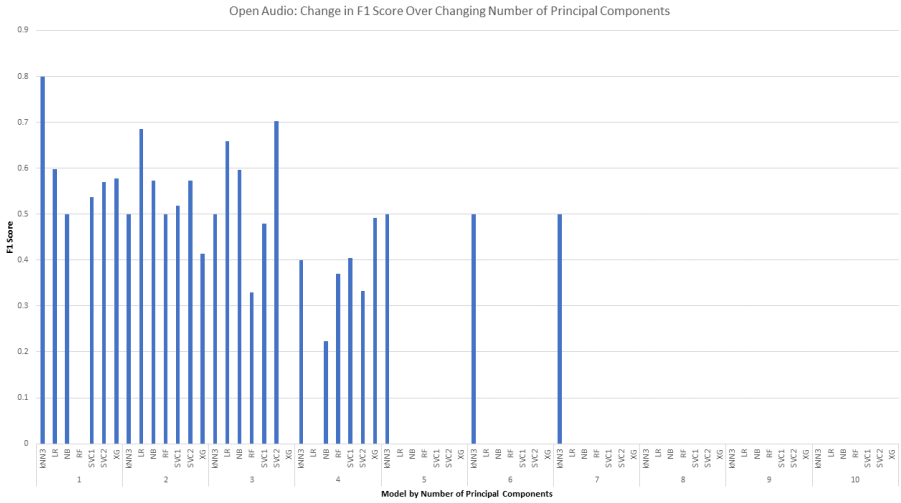

Figure 80: The changes in F1 score for Open Audio across models as the number of principal components increases.

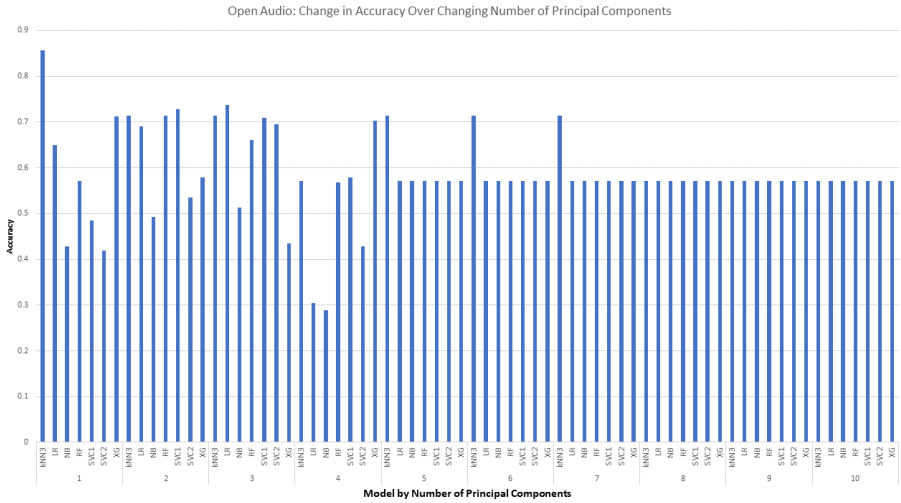

Figure 81: The changes in Accuracy for Open Audio across models as the number of principal components increases.

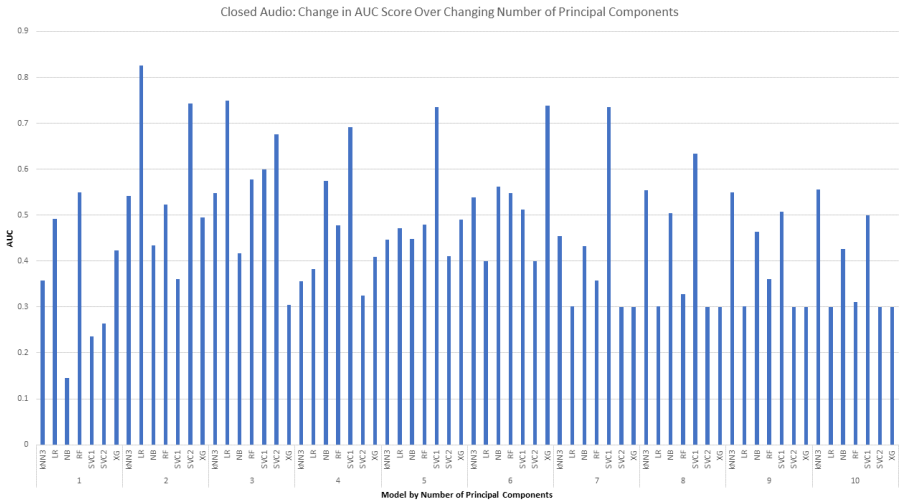

Figure 82: The changes in AUC score for Closed Audio across models as the number of principal components increases.

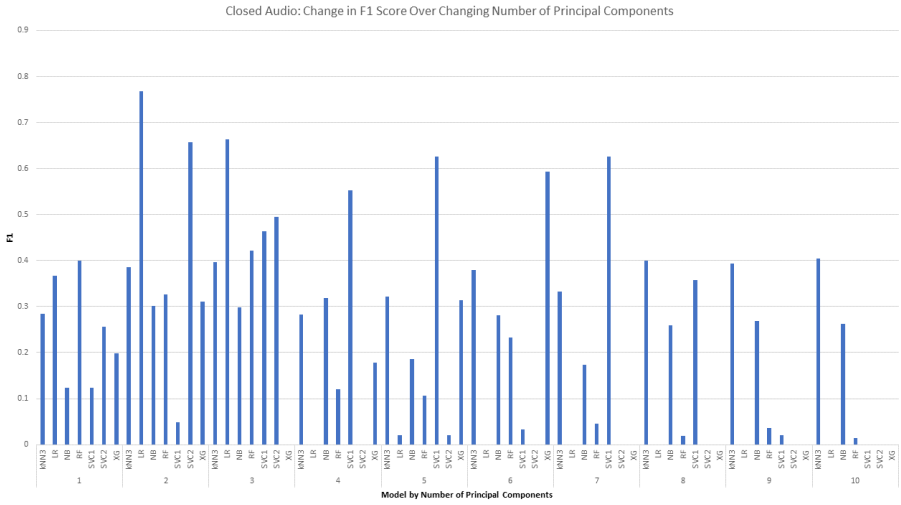

Figure 83: The changes in F1 score for Closed Audio across models as the number of principal components increases.

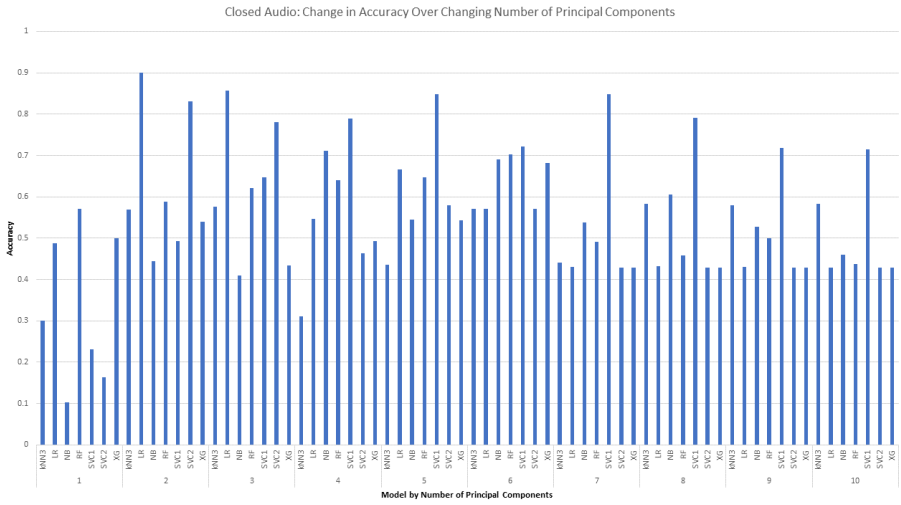

Figure 84: The changes in Accuracy for Closed Audio across models as the number of principal components increases.

6.8 Appendix H: Machine Learning Results Data

Table 21: Fall StudentData Open Audio Results.

num_features	model	auc	f1	accuracy
1	NB	46%	50%	43%
1	LR	64%	60%	65%
1	SVC1	51%	54%	48%
1	SVC2	48%	57%	42%
1	XG	69%	58%	71%
1	kNN3	83%	80%	86%
1	RF	50%	0%	57%
2	NB	53%	57%	49%
2	LR	70%	68%	69%
2	SVC1	68%	52%	73%
2	SVC2	56%	57%	54%
2	XG	55%	41%	58%
2	kNN3	67%	50%	71%
2	RF	67%	50%	71%
3	NB	56%	60%	51%
3	LR	72%	66%	74%
3	SVC1	66%	48%	71%
3	SVC2	72%	70%	70%
3	XG	38%	0%	43%
3	kNN3	67%	50%	71%
3	RF	61%	33%	66%
4	NB	28%	22%	29%
4	LR	27%	0%	30%
4	SVC1	55%	41%	58%
4	SVC2	42%	33%	43%
4	XG	66%	49%	70%
4	kNN3	54%	40%	57%
4	RF	54%	37%	57%
5	NB	50%	0%	57%
5	LR	50%	0%	57%
5	SVC1	50%	0%	57%
5	SVC2	50%	0%	57%
5	XG	50%	0%	57%
5	kNN3	67%	50%	71%
5	RF	50%	0%	57%
6	NB	50%	0%	57%
6	LR	50%	0%	57%
6	SVC1	50%	0%	57%
6	SVC2	50%	0%	57%
6	XG	50%	0%	57%

6	kNN3	67%	50%	71%
6	RF	50%	0%	57%
7	NB	50%	0%	57%
7	LR	50%	0%	57%
7	SVC1	50%	0%	57%
7	SVC2	50%	0%	57%
7	XG	50%	0%	57%
7	kNN3	67%	50%	71%
7	RF	50%	0%	57%
8	NB	50%	0%	57%
8	LR	50%	0%	57%
8	SVC1	50%	0%	57%
8	SVC2	50%	0%	57%
8	XG	50%	0%	57%
8	kNN3	50%	0%	57%
8	RF	50%	0%	57%
9	NB	50%	0%	57%
9	LR	50%	0%	57%
9	SVC1	50%	0%	57%
9	SVC2	50%	0%	57%
9	XG	50%	0%	57%
9	kNN3	50%	0%	57%
9	RF	50%	0%	57%
10	NB	50%	0%	57%
10	LR	50%	0%	57%
10	SVC1	50%	0%	57%
10	SVC2	50%	0%	57%
10	XG	50%	0%	57%
10	kNN3	50%	0%	57%
10	RF	50%	0%	57%

Table 22: Fall StudentData Closed Audio Results.

num_features	model	auc	f1	accuracy
1	NB	15%	12%	10%
1	LR	49%	37%	49%
1	SVC1	24%	12%	23%
1	SVC2	26%	26%	16%
1	XG	42%	20%	50%
1	kNN3	36%	28%	30%
1	RF	55%	40%	57%
2	NB	43%	30%	44%
2	LR	83%	77%	90%
2	SVC1	36%	5%	49%
2	SVC2	74%	66%	83%

2	XG	49%	31%	54%
2	kNN3	54%	39%	57%
2	RF	52%	33%	59%
3	NB	42%	30%	41%
3	LR	75%	66%	86%
3	SVC1	60%	46%	65%
3	SVC2	68%	49%	78%
3	XG	30%	0%	43%
3	kNN3	55%	40%	58%
3	RF	58%	42%	62%
4	NB	58%	32%	71%
4	LR	38%	0%	55%
4	SVC1	69%	55%	79%
4	SVC2	32%	0%	46%
4	XG	41%	18%	49%
4	kNN3	36%	28%	31%
4	RF	48%	12%	64%
5	NB	45%	19%	54%
5	LR	47%	2%	67%
5	SVC1	73%	63%	85%
5	SVC2	41%	2%	58%
5	XG	49%	31%	54%
5	kNN3	45%	32%	44%
5	RF	48%	11%	65%
6	NB	56%	28%	69%
6	LR	40%	0%	57%
6	SVC1	51%	3%	72%
6	SVC2	40%	0%	57%
6	XG	74%	59%	68%
6	kNN3	54%	38%	57%
6	RF	55%	23%	70%
7	NB	43%	17%	54%
7	LR	30%	0%	43%
7	SVC1	73%	63%	85%
7	SVC2	30%	0%	43%
7	XG	30%	0%	43%
7	kNN3	45%	33%	44%
7	RF	36%	4%	49%
8	NB	50%	26%	60%
8	LR	30%	0%	43%
8	SVC1	63%	36%	79%
8	SVC2	30%	0%	43%
8	XG	30%	0%	43%
8	kNN3	55%	40%	58%
8	RF	33%	2%	46%
9	NB	46%	27%	53%

9	LR	30%	0%	43%
9	SVC1	51%	2%	72%
9	SVC2	30%	0%	43%
9	XG	30%	0%	43%
9	kNN3	55%	39%	58%
9	RF	36%	4%	50%
10	NB	43%	26%	46%
10	LR	30%	0%	43%
10	SVC1	50%	0%	71%
10	SVC2	30%	0%	43%
10	XG	30%	0%	43%
10	kNN3	56%	41%	58%
10	RF	31%	1%	44%

6.9 Appendix I: Machine Learning Terminal Commands

```
python -W ignore run_machine_learning.py ^
--data data_sources/feature_extraction_audio/open_smile_features_open_phq_gad.csv ^
--modality audio_open ^
--evaluation_strategy tts ^
--sampling regular_oversampling ^
--feature_selection pca ^
--oppo_target ^
--target_type phq
```

Figure 85: The terminal commands used to run the machine learning code on Open Audio. The results are seen in Table 21 in Appendix H

```
python -W ignore run_machine_learning.py ^
--data data_sources/feature_extraction_audio/open_smile_features_closed_phq_gad.csv ^
--modality audio_closed ^
--evaluation_strategy tts ^
--sampling regular_oversampling ^
--feature_selection pca ^
--oppo_target ^
--target_type phq
```

Figure 86: The terminal commands used to run the machine learning code on Closed Audio. The results are seen in Table 22 in Appendix H

References

Albert, P. R. (2015). Why is depression more prevalent in women? *Journal of psychiatry and neuroscience*, *40*, 219-221.

Assam, J., Flannery, M., Resom, Y. G. A., & Wu, Y. (2019). *Machine learning for mental health detection* (Tech. Rep.). Worcester Polytechnic Institute.

Ball, D., Dogrucu, A., Isaro, A., & Perucic, A. (2018). *Sensing depression* (Tech. Rep.). Worcester Polytechnic Institute.

Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., ... Gamboa, H. (2020). Tsfel: Time series feature extraction library. *SoftwareX*, *11*, 100456. doi: https://doi.org/10.1016/j.softx.2020.100456

A basic introduction to neural networks. (n.d.). University of Wisconsin Madison. Retrieved from `http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html`

Beiwe research platform. (n.d.). Onnela Lab. Retrieved from `https://www.beiwe.org/`

Breiman, L., & Cutler, A. (n.d.). *Random forests.* Berkeley Statistics. Retrieved from `https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`

Brownlee, J. (2016a). *K-nearest neighbors for machine learning.* Machine Learning Mastery. Retrieved from `https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning`

Brownlee, J. (2016b). *Logistic regression for machine learning.* Machine Learning Mastery. Retrieved from `https://machinelearningmastery.com/logistic-regression-for-machine-learning/`

Brownlee, J. (2018). *How to use roc curves and precision-recall curves for classification in python.* Machine Learning Mastery. Retrieved from `https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/`

Cai, H., Gao, Y., Sun, S., Li, N., Tian, F., Xiao, H., ... Hu, B. (2019). *Modma dataset: a multi-modal open dataset for mental disorder analysis* (Tech. Rep.). Retrieved from `https://arxiv.org/ftp/arxiv/papers/2002/2002.09283.pdf`

Caltaiano, J., Pingal, N., Thant, M. M., Taye, Y., Sargent, A., & Seifu, Y. (2020). *Mental health sensing using machine learning* (Tech. Rep.). Worcester Polytechnic Institute.

Chowdary, D. H. (2020). *Decision trees explained with a practical example.* Medium. Retrieved from `https://medium.com/towards-artificial-intelligence/decision-trees-explained-with-a-practical-example-fe47872d3b53`

Classification: Roc curve and auc. (n.d.). Google Developers. Retrieved from `https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc`

Data dashboards. (n.d.). Worcester Polytechnic Institute. Retrieved from `https://www.wpi.edu/offices/institutional-research/data-dashboards`

Depression and women. (n.d.). mentalhealth.net. Retrieved from https://www.mentalhelp.net/depression/women/

Depression — phq-9. (n.d.). greenspace. Retrieved from https://help.greenspacehealth.com/article/85-depression-phq-9

Gandhi, R. (2018). *Support vector machine — introduction to machine learning algorithms.* Towards Data Science. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Guntuku, S. C., Yaden, D. B., Kern, M. L., Ungar, L. H., & Eichstaedt, J. C. (2017). Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, *18*, 43-49. doi: http://dx.doi.org/10.1016/j.cobeha.2017.07.005

Gupta, P. (2017). *Decision trees in machine learning.* Towards Data Science. Retrieved from https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

Hardesty, L. (2017). *Explained: Neural networks.* MIT. Retrieved from https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

Imbalanced data. (2020). Google Developers. Retrieved from https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

Introduction to the fundamentals of time series data and analysis. (2019). Aptech. Retrieved from https://www.aptech.com/blog/introduction-to-the-fundamentals-of-time-series-data-and-analysis/

Jaadi, Z. (2019). *A step by step explanation of principal component analysis.* Built In. Retrieved from https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Kartik Nighania. (2018). *Various ways to evaluate a machine learning model's performance.* Towards Data Science. Retrieved from https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15

Key substance use and mental health indicators in the united states: Results from the 2019 national survey on drug use and health (Tech. Rep.). (2020). Retrieved from https://www.samhsa.gov/data/sites/default/files/reports/rpt29393/2019NSDUHFFRPDFWHTML/2019NSDUHFFR1PDFW090120.pdf

Koehrsen, W. (2018). *Statistical significance explained.* Towards Data Science. Retrieved from https://towardsdatascience.com/statistical-significance-hypothesis-testing-the-normal-curve-and-p-values-93274fa32687

Machine learning. (n.d.). IBM. Retrieved from https://www.ibm.com/cloud/learn/machine-learning

McClellan, C., Ali, M. M., Mutter, R., Kroutil, L., , & Landwehr, J. (n.d.). Using social media to monitor mental health discussions 2 evidence from twitter. *AMIA*.

Men and depression. (n.d.). National Institute of Mental Health.

Retrieved from https://www.nimh.nih.gov/health/publications/men-and-depression/index.shtml

onnela-lab. (n.d.). Onnela Lab. Retrieved from https://github.com/onnela-lab

Pedrelli, P., Nyer, M., Yeung, A., Zulauf, C., & Wilens, T. (2015). College students: Mental health problems and treatment considerations. *Academic psychiatry : the journal of the American Association of Directors of Psychiatric Residency Training and the Association for Academic Psychiatry*, *29*, 503-511. doi: 10.1007/s40596-014-0205-9

Pennington, C. R., Heim, D., Levy, A. R., & Larkin, D. T. (2016). Twenty years of stereotype threat research: A review of psychological mediators. *PLOS ONE*. Retrieved from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0146487

Population proportion – sample size. (n.d.). Select Statistical Services Limited. Retrieved from https://select-statistics.co.uk/calculators/sample-size-calculator-population-proportion

Ray, S. (2015). *Quick introduction to boosting algorithms in machine learning.* Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/

Reading data from healthkit - apple developer. (n.d.). Apple Inc. Retrieved from https://developer.apple.com/documentation/healthkit/reading_data_from_healthkit

Rohit Madan. (2019). *K nearest neighbour for classification on breast cancer data , results with preprocessing and w/o normalising.* Medium. Retrieved from https://medium.com/@madanflies/k-nearest-neighbour-for-classification-on-breast-cancer-data-results-with-preprocessing-and-w-o-e21b0cc98a2f

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, *10*(3), 1-21. doi: 10.1371/journal.pone.0118432

Salvador, S., & Chan, P. (2007). Fastdtw: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*. https://dl.acm.org/doi/10.5555/1367985.1367993, doi = 10.5555/1367985.1367993.

Shukla, L. (2019). *Designing your neural networks.* KDnuggets. Retrieved from https://www.kdnuggets.com/2019/11/designing-neural-networks.html

Sirikit - apple developer documentation. (n.d.). Apple Inc. Retrieved from https://developer.apple.com/documentation/sirikit

Smith, A. (n.d.). *Demographics of mobile device ownership and adoption in the united states (2019).* Retrieved from https://www.renalis.health/research/2019/9/13/demographics-of-mobile-device-ownership-and-adoption-in-the-united-states-2019

Studentlife survey. (n.d.). Dartmouth College StudentLife Team. Retrieved from https://studentlife.cs.dartmouth.edu/

United states demographic statistics. (n.d.). InfoPlease. Retrieved from https://www.infoplease.com/us/census/demographic-statistics

Wang, R., Chen, F., et al. (2017). Studentlife: Using smartphones to assess mental health and academic performance of college students. *Mobile Health: Sensors, Analytic Methods, and Applications*, 7-33. doi: 10.1007/978-3-319-51394-2_2

Weisstein, E. W. (n.d.). *Sigmoid function.* Wolfram MathWorld. Retrieved from https://mathworld.wolfram.com/SigmoidFunction.html

Wikipedia contributors. (2020). *Sensitivity and specificity — Wikipedia, the free encyclopedia.* https://en.wikipedia.org/w/index.php?title=Sensitivity_and_specificity&oldid=980031477. ([Online; accessed 30-September-2020])

Wood, T. (n.d.-a). *F-score.* DeepAI. Retrieved from https://deepai.org/machine-learning-glossary-and-terms/f-score

Wood, T. (n.d.-b). *Precision and recall.* DeepAI. Retrieved from https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall

Yentes, R., Toaddy, S., Thompson, L., Gissel, A., & Stoughton, J. (2012). Effects of survey progress bars on data quality and enjoyment..

Zhang, J. (2020). *Dynamic time warping.* Towards Data Science. Retrieved from https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd