Mindtis: Online Emotion Recognition using EEG Signal

E6765_2019Spring_LXYZ_report

Yuhan Liu (yl4020), Shiyun Yang (sy2797), Tianyu Xie (tx2180)
Columbia University

Abstract

To help improve mental health awareness and evaluate overall emotional healthiness, we developed an emotion recognition system which integrates machine learning techniques and the use of a web application to classify positive and negative emotions based on EEG brainwave data and provide visualizations of results and emotion-based recommendations. Our system achieved over 80% accuracy in predicting emotions, and our goal to provide users the ability to monitor, analyze, and visualize their emotions was accomplished.

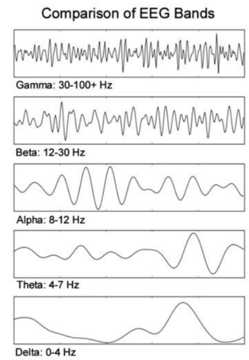
1. Overview

1.1 Problem in a Nutshell

Mental health, which encompasses our emotional, psychological, and social well-being, is a vital part of our lives and impacts how we think, feel, and act. According to the National Institute of Mental Health, approximately 1 in 5 adults in the U.S. (46.6 million) experiences mental illness every year, and suicide is the 2nd leading cause of death for people between age 10 and 34. [2] Mood Disorders, such as depression, is one of the most common mental health conditions affecting nearly 10% of adults each year, and can contribute to higher medical expenses, poorer performance at school and work, and increased risk of suicide if untreated.

Ability to recognize emotions and evaluate overall emotional healthiness can greatly improve mental health awareness and self-assessment, therefore, we propose an emotion recognition system that utilizes machine learning techniques to classify positive and negative emotions and provides result visualizations and recommendations through a web application.

The approach we use to classify emotion is by collecting and analyzing electroencephalogram (EEG) brainwave data. EEG is a noninvasive monitoring method to record electrical activity of the brain and the signal recorded is typically divided into the following 5 frequency bands: delta, theta, alpha, beta, and gamma. The following figure shows the typical frequency ranges for each of the 5 brainwave bands.



After EEG data is collected, time domain features are extracted from the raw data and used as input to the machine learning model which performs classification of the data. In order to train the machine learning model, we collected EEG data with positive and negative emotions by watching Youtube videos with strong emotional impact. The trained model is then used in our web application to do real-time prediction of the user's emotion. The prediction results are stored in remote database and provided to the user through visualizations.

1.2 Prior Work

"Integration of a Low Cost EEG Headset with The Internet of Thing Framework" [4] uses Mindwave Mobile as EEG sensor and collects data from 6 different participants to train emotion classification models. To obtain training data, the author uses short video stimulants to induce emotion and a self-assessment manikin to evaluate participants emotional state. The author uses the dimensional mode which classifies emotion into high arousal/positive valence, high arousal/negative valence, low arousal/positive valence, and low arousal/negative valence. 24 features are extracted from the EEG data, and Linear Discriminant Analysis is selected as the classifier model. The resulting accuracies range from 52.78% to 86.1%, with an average accuracy of 70.52%.

2. Description

In this section, we provide detailed description of our system design. Section 2.1 list the objectives of our system and technical challenges faced during the design process. Section 2.2 includes the description of the problem and the system block diagram. Section 2.3 discusses the software design, which includes the machine learning and web application design.

2.1. Objectives and Technical Challenges

The objectives of our end-to-end system are the following:

- Connect our low-cost EEG headset (Mindwave Mobile 2 from NeuroSky) to the user's laptop through Bluetooth.
- Collect EEG data from the headset with the click of a button on the web application.
- Train a machine learning model to classify positive and negative emotions from EEG data as accurate as possible.
- Use the trained model to predict user's emotions based on collected data.
- Provide the user visualizations of the results, as well as history of all prediction results collected from the user.
- Provide entertainment recommendation to the user based on emotion.

The technical challenges we faced include the following:

- Training data collection: since we need correctly labeled EEG data to train our machine learning model, the data has to be collected from multiple participants in a controlled environment with no distraction. The participants also need to provide self-assessment on how positive/negative their emotions are during the data collection process.
- Machine learning model: classifying EEG data is not an easy task due to the noise present in the brainwave and the randomness of the participants emotions during data collection, so we need to use various techniques to make our machine learning models as accurate as possible.
- Web application: the web application we develop needs to have multiple features including starting data collection from the headset, display graphs of the user's brainwave data measured by the sensor, and retrieve emotion classification results from a remote database to display graphs of the user's historical data.

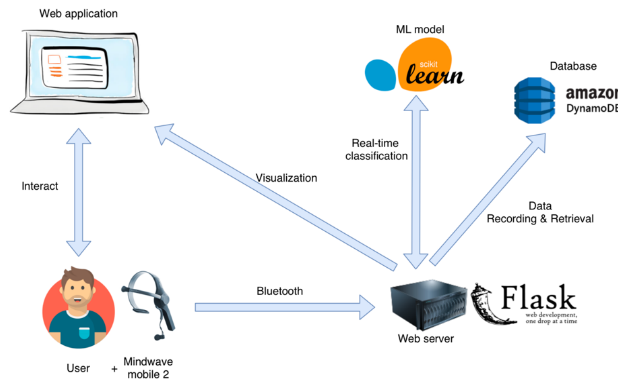
2.2. Problem Formulation and Design

2.2.1 Problem Formulation

To fully achieve the goal, it is required for the system to implement the functionalities of device connection, data collection, data classification, data recording and retrieval as well as data visualization.

Mindwave mobile 2 is connected to the server using Bluetooth through its own module. The rest of the system contains three main parts, a machine learning model which deals with the data classification, a web server backend that controls the data flowing in the structure and a web app frontend that provides interactions and data visualization to the users. In addition, we also choose to use Amazon DynamoDB as our database here to record the brainwave data from the users with the classification results and timestamp. The three main parts will be discussed in detail in the following sections.

2.2.2 System Design Block Diagram



2.3 Software Design

2.3.1 Machine Learning

2.3.1.1 Data Collection

The sensor we use is the Mindwave Mobile 2 EEG headset from NeuroSky, which costs $99. When requested, the headset sends 1 reading of data every 1 second, which includes the following values: attention (user's attention level measured by the headset), meditation (user's relaxation level measured by the headset), delta, theta, low alpha, high alpha, low beta, high beta, low gamma, and high gamma. The last 8 values in the data are brainwave bands values, which we use in our emotion classification algorithm. The first 2 values are obtained from the headset's own algorithm (not raw brainwave values) and therefore are not used in our algorithm. A sample of the sensor reading is shown below.

	attention	meditation	delta	theta	lowAlpha	highAlpha	lowBeta	highBeta	lowGamma	highGamma
10	74	70	820731	146902	17318	22871	3570	6476	3650	5248
11	63	57	488976	61010	20354	14840	19181	12088	11619	10010
12	60	41	605721	11887	11184	2975	4448	2469	1338	1233
13	66	77	417771	99357	23036	38880	17311	15195	22382	9703
14	56	70	216534	39424	61009	17977	12372	25014	18632	10707

Our training data is collected from 3 participants, 2 males and 1 female. The positive/negative emotion data are collected by asking the participants to attentively watch funny or sad Youtube videos in a quiet room without any distraction. The emotional self-assessment of the participants after watching the video is also taken into account, and only the data assessed as truly positive or negative is used in training. A total of 4800 seconds of data is collected for training, in which about 45% is positive and 55% is negative.

2.3.1.2 Data Preprocessing

After the training data is collected, cleaning of the data is done to remove missing values and zero values from the data. The cleaned data are then divided into 8-second sequences in order to extract time series features. Therefore, the sequences each has 8 rows corresponding to 8 seconds of readings, and 8 columns corresponding to the 8 brainwave band values. To extract features from the sequences, the "tsfresh" package in Python is used, and 64 features are extracted from each sequence; thus, each sequence is transformed into 1 row of data with 64 columns. A sample of the data after feature extraction is shown below.

delta__maximum	delta__mean	delta__median	delta__minimum	delta__standard_deviation	delta__sum_values	delta__variance
820731.0	358632.250	311352.5	43084.0	2.538519e+05	2869058.0	6.444077e+10
696413.0	413403.250	458594.0	76912.0	2.132983e+05	3307226.0	4.549615e+10
2062550.0	968694.750	684693.0	194843.0	5.754738e+05	7749558.0	3.311701e+11

The data after feature extraction is then used as training data for the machine learning classifier models.

2.3.1.3 Model Selection

A wide variety of machine learning classifier models have been used in previous research papers, so experiments had to be done for us to pick the best model. Using Python and Scikit-learn, we initially tried 10 classifier, including K nearest neighbors (KNN), SVM, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, Gaussian Naive Bayes, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis, and XGBoost (XGB).

After calculating the 5-fold cross validation score for each of the models, we observed that 3 of the models had poor performance on all datasets, while the other 7 models had good performance on some dataset but the performances were not stable. Therefore, we decided to build a voting ensemble with these 7 models to balance out the weaknesses of each model. The experiment result of the XGB classifier is shown below as an example.

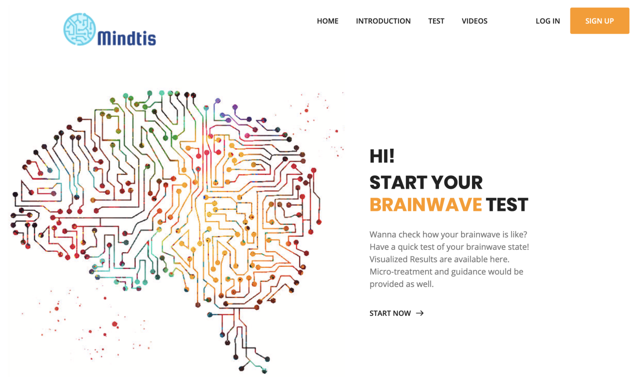
```
Model saved to  ./Models/XGBClassifier_model.pkl
****Results****
5-Fold Cross Validation Score: [0.75       0.703125    0.74603175 0.57142857 0.66666667]
Accuracy: 75.0000%
Log Loss: 0.5897759100355741
```

The 7 models we selected to build the voting ensemble are the following: KNN, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, LDA, and XGB. To further increase the accuracy, we used Grid Search to find the best hyperparameter set for each of the 7 models. The Grid Search result of the XGB classifier is shown below as an example.

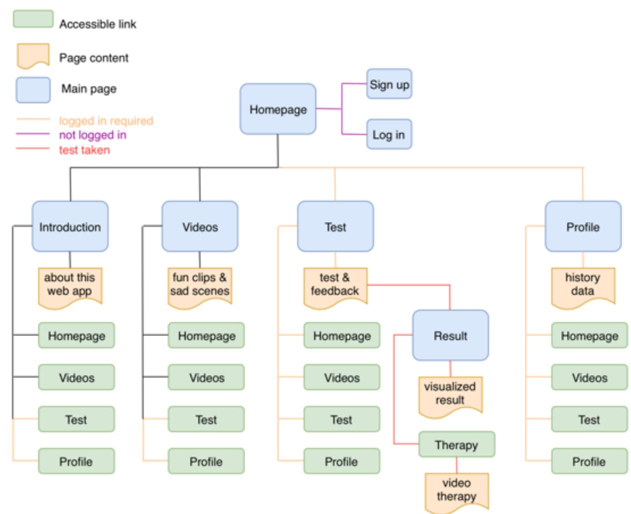
```
****Results****
Best parameters set:  {'learning_rate': 0.05, 'max_depth': 1, 'n_estimators': 100}
Best estimator:  XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, gamma=0, learning_rate=0.05, max_delta_step=0,
       max_depth=1, min_child_weight=1, missing=None, n_estimators=100,
       n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
       silent=True, subsample=1)
```

After training the 7 models with best hyperparameter sets, we saved the trained models and used them to build the voting ensemble, which is applied in our final system.

2.3.2 Web Application

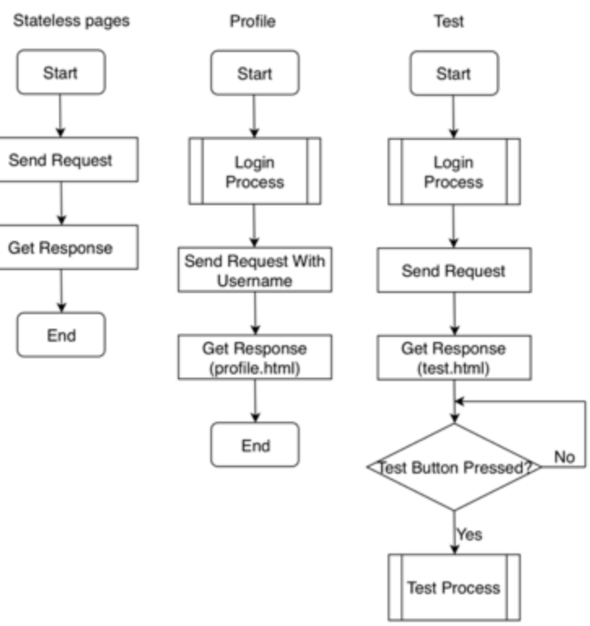


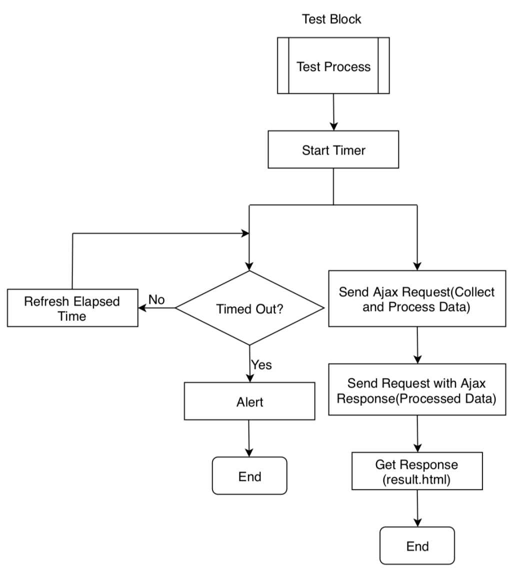
2.3.2.1 Web App Site Map



2.3.2.2 Requests and Responses

Half of the main page links here are stateless. Towards these pages, requests are sent without carrying extra information and then such static page contents are rendered and returned. For the others, a user is required to login before approaching. As for the data retrieval block on Profile, user's corresponding username is required to match the history data. For the testing block on Test, a data collecting and processing process is started for the feedback shown on Result page. This process is started by sending an asynchronous request using Ajax in order to maintain the original webpage for video playing and elapsed time counting. Details of the implementations are shown in the following flowcharts.





2.3.3 Back End Design

2.3.3.1 Signup & Login Module

Signup and Login functionality for this web application is implemented based on Flask-Login module and Flask SQLAlchemy as the storage of user information. This extension allows us to work with python objects instead of managing data tables. Thus, user as an object is stored in or read from the database. Moreover, a login session would be maintained with in local proxy of the browser to keep the login state for the user.

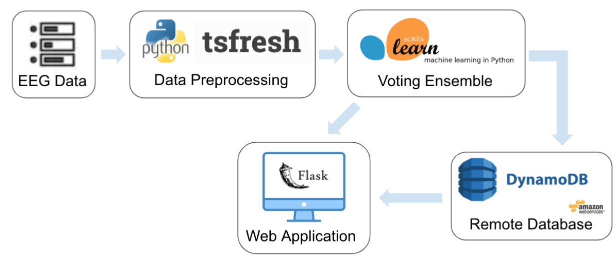
2.3.3.2 Request Routing

Web framework Flask is used for routing the requests sent from the frontend. Requests are decomposed into fractions to route correctly and pass necessary parameters. In addition, to prevent anonymous users from accessing key pages, the decorator *@login_required*is applied.

3. Results

3.1 Machine Learning Results

The machine learning model used in our system is described in detail in section 2.3.1. The average accuracy of the voting ensemble classifier is about 80%. The system diagram of the machine learning process is shown below.



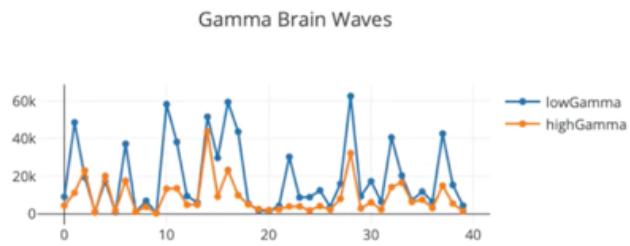
In the real-time prediction process, we first collect 40 seconds (5 sequences * 8 seconds/sequence) of valid data from the user (data with missing values or zeros is discarded during the collection process). The data is then processed and passed to the voting ensemble. Based on the predicted probabilities of each class, we divide the result into 5 subcategories: very positive, positive, neutral, negative, and very negative. The prediction results, which include both the class labels (positive/negative) and the probabilities for each class, are stored in DynamoDB and used to draw visualization graphs on the web application.

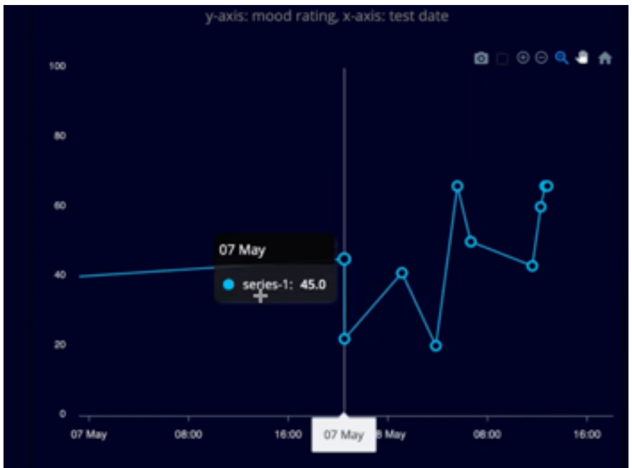
3.2 Visualization Results

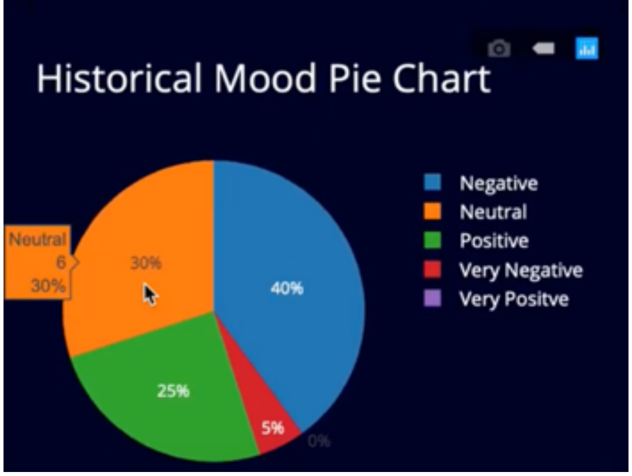
3.2.1 Visualization Tools

Plot.ly and ApexCharts are introduced for results visualization. Plot.ly provides online data analytics and visualization tools. ApexCharts is a charting library that helps developers create interactive visualization graphs. Here we us Plot.ly to produce brainwave line chart and historical classification result pie chart for users. ApexCharts is used for producing interactive line chart for historical result tracking with detailed datetime.

3.2.2 Figures







4. Demonstration

A video demonstrating the features of our web application is linked below:

https://www.youtube.com/watch?v=LPz2ly0CdqY

5. Discussion and Further Work

Our model is able to classify the emotion of users into 2 categories (positive and negative) and achieves over 80% of accuracy, which outperforms some of the classification performances presented in other research papers. Our web-application is able to handle a few users' daily request. Although we have achieved our basic goal, there are still many work left for the future. Future work concerns improving the accuracy of classification of our model and increasing the scalability for our web-application. Due to the lack of time, the training set only contains data collected from self-assessment of our team members, in order to increase the classification accuracy, we can increase the size of training set by finding more participants to collect data from. In order to make our web-application more scalable to handle more requests, we must deploy our web server on cloud, which requires us to develop an end device that can interact with out web-application to send data to our server.

6. Conclusion

We developed a machine learning model that can classify the emotion of user into 2 categories: positive and negative. On top of our model, we implemented a web-application that is able to monitor, analyze and visualize the real-time emotion condition of user then provide video and advice based on the result. Besides, our web-application allows user to track the history record of emotion in order to provide more detailed analysis. During the project we have a thorough understanding of how to build a IOT web-application, especially how to divide the application into different modules to implement independently and work together to make them functional. Although we have basically achieved the goal we set in our proposal, improvement can be made by collecting more data to improve the emotion recognition ability of our model and deploying our web server on cloud to increase the scalability.

7. Acknowledgements

We would like to thank Professor Kostic for the knowledge he taught in class and the advice he gave which greatly helped us in designing our project. We would also like to thank the TAs of this class for their patience and support during office hours.

8. References

[1] H. Li, "Author Guidelines for CMPE 146/242 Project Report", *Lecture Notes of CMPE 146/242*, Computer Engineering Department, College of Engineering, San Jose State University, March 6, 2006, pp. 1.

[2] "Mental Health By The Numbers." *National Alliance on Mental Illness*, www.nami.org/learn-more/mental-health-by-the-numbers.

[3] "Making Sense of EEG Bands." *NeuroSky*, neurosky.com/2015/05/greek-alphabet-soup-making-sense-of-eeg-bands/.

[4] Mohammed, Alnemari. "Integration of a Low Cost EEG Headset with The Internet of Thing Framework." *UC Irvine*, 2017, escholarship.org/uc/item/0d90x267.

[5] Weebly website link: https://iotcolumbia2019lxyz.weebly.com/