This is the documentation/ Flowchart for classification of state of mind.

Libraries Used -

1. Pandas (pd) - for data manipulation
2. Numpy (np)- for computation
3. Sklearn - for Machine Learning model selection
4. Matplotlib - Plotting
5. OS- for getting files

Procedure and Files-

Complete_data_visuals- This file contains graphs plotted for all the 4 data files separately. To visualize the amplitude values matplotlib's scatter plot is used different colors for different categories.

Value, tag and colour

a. 0 unlabeled Black
b. 32 peak red
c. 37 som med blue
d. 38 good med yellow
e. 39 mind wonder green
f. 40 sleepy white

Mind Wonder(39) and sleepy(40) are 5-10 sec after it actually occurred

Data Insights -

1. Amplitude ranges of different files are different
2. Starting data (2-3 seconds) is random and need to be removed
3. Amplitude ranges of good meditation is higher than light meditation
4. Amplitude of peak meditation i either near highest of good meditation or near lowest of good meditation.
5. There are many scattered dots(artifacts) that need to be removed
6. Most of labeled data is for Light Meditation and mind wonder

Data extraction and other models using amplitude- This file deals with the O3 file, as it contained maximum number of categories of data.

a. Reading the file with pd.read_csv

b. Labeling mind wonder correctly (0:05-1:05) before every label 39 starts as 1
c. Remove the data labeled 0(unlabeled) and 39(mind wonder labeled) from the read file
d. Train test split using Sklearn.train_test_split 67% train, 33% test - random distribution
e. Training by 67%(train data), Predicting the Test Data results and comparing to actual labels
f. Accuracy was skewed 48-52% from K-nearest neighbours, Decision tree, naive bayes, svm

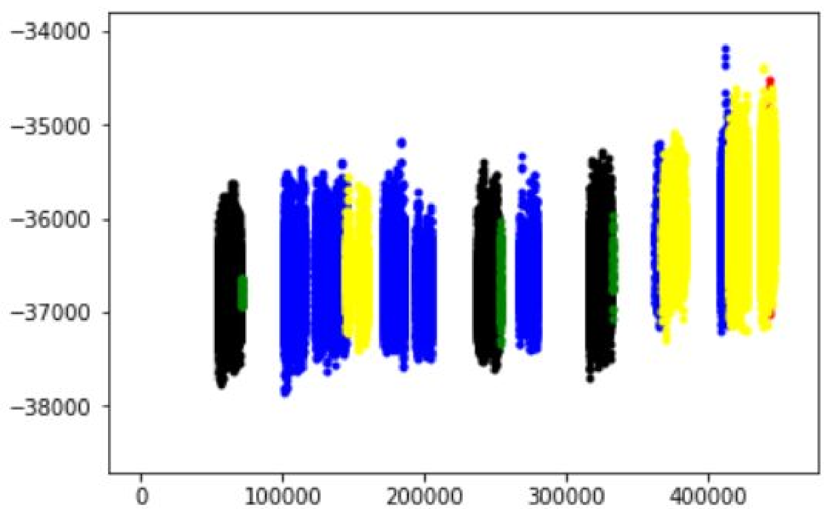

Fig 1

Results Insights-

1. It Learned (see fig1)-
    a. -34k to -36k good meditation Yellow
    b. -36k to -37k some meditation Blue
    c. -37k above Mind Wonder Black

    This is clearly wrong

Deleting Unlabeled data- this file reads all the data files, label the mind wonder and sleepy states properly and then remove Unlabeled, Previous Mind wonder and previous Sleepy labels \

a. Reading all the files and trimming the 1st second
b. Clubbing all files together in a pd.DataFrame

c. Labeling mind wonder and sleepy correctly (0:05-1:05) before every label 39(Mind Wonder) starts as 1 and 40(sleepy) as 2
d. Delete Unlabeled, Previously labeled (Sleepy and mind wonder)
e. Saving the remaining data to trimmedData.csv

Total Number of rows in trimmed data.csv is 657431

Creating Chunks of data and basic algos- This files divides the data into windows after clubbing same labeled together and uses Machine learning models after basic feature extraction.

a. Reading from TrimmedData.csv
b. Grouping all amplitudes by Label
c. Creating Widows of window_size - 250(1 second)
d. Extracting features - Max, min, avg, std, fft
e. Train test Split
f. Finding accuracy via Decision tree, Svm , KNN, Random Forest

Insights-

1. Total number of chunks in each class-
    a. 1 -new mind wonder - 1260
    b. 2 -new sleepy - 261
    c. 32 -peak experience - 8
    d. 37 -some meditation - 785
    e. 38 -good meditation - 314
2. Training by 67%(train data), Predicting the Test Data results and comparing to actual labels
3. Analysing Results - Accuracy, precision, recall, F1-score for each class in Random Forest Classifier with number of estimators =100

```
Confusion Matrix :
[[398   9   0  19   2]
 [ 18  53   0   2   0]
 [  0   0   2   0   0]
 [ 20  11   0 206  11]
 [  6   0   0  15  96]]
Accuracy Score : 0.869815668202765
              precision    recall  f1-score   support

         1.0       0.90      0.93      0.91       428
         2.0       0.73      0.73      0.73        73
        32.0       1.00      1.00      1.00         2
        37.0       0.85      0.83      0.84       248
        38.0       0.88      0.82      0.85       117
```

here the class with the worst accuracy is 2 - sleepy (Probably due to very small training set and high resemblance with class 1 - mind wonder(18 points shifted) this may be due to labeling the same way.

4. Best accuracy for class 1 (so if we increase the data we can expect the accuracy to go high)

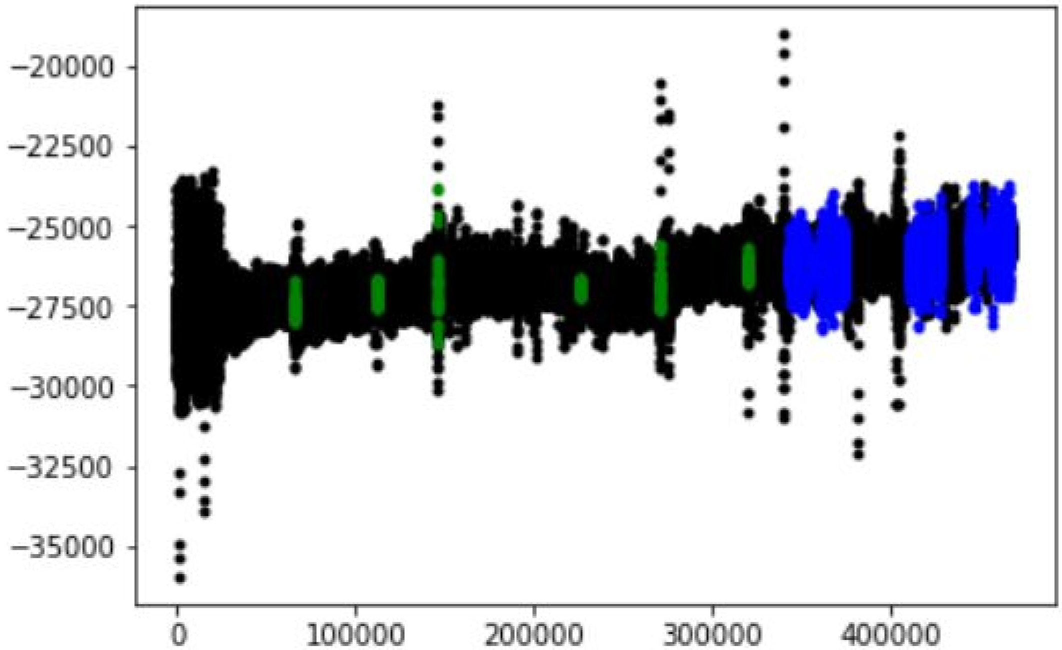

Fig 2

Loading Artifact files - As there were artifacts in the data which can be seen by the image fig 2 we need to remove them.

a. There are 10 classes
    i. both eye blink open-close-open q 81
    ii. both eye left open w 87
    iii. both eye right open e 69
    iv. both eye left closed r 82
    v. both eye right closed t 84
    vi. both eye shift down y 89
    vii. both eye blink close-open-close u 85
    viii. forehead squeeze i 73
    ix. Any artifact a 65
    x. NO Artifact o 79

b. Load all artifact files, remove the 1st second, club them to create a single Dataframe
c. Create chuks for each labeled artifact - 338 (last 6 are non artifact)

d. Non artifacts are large chunks so we need to divide them in ranges of other files (100- 400) data points this makes (440 files approx)
    i. Extract 79 labeled chunks
    ii. Divide them into small random chunks
    iii. Append them back
e. Mean subtraction from each chunk independently
f. Feature extraction - Minimum, Maximum, Max-Min, std
g. Train Test split and algorithms
    i. We can’t classify between the artifacts with just these features, Accuracy is 48%
    ii. Label all Artifacts as 1, non artifacts as 79
    iii. Now accuracy is 99% so it can be used to remove artifacts

Insights-

1. Data Should be labeled as
    a. Start just before an artifact, continue to press until the artifact is complete
    b. Don’t start to label the next until the graph seems stable
2. Just the amplitude can be used as a factor to classify artifact vs non artifact

Predicting artifacts vs non artifacts on classification data - This files predict artifact vs non artifact on mind state classification 1 second shunk files and save all the indices in a file.

a. Train an svm classifier with the help of Load Artifact files.ipynb file
b. Load mind state classification file -trimmed data.csv
c. Use mean subtraction in each chunk of second
d. Divide the amplitude valued by mean(Max-min) of training(artifact files)/ mean(max-min) of predicting(mind state classification files)==1.8
e. Predict artifact using the trained model
f. Examine the predictions and save indices to a file (artifacts predicted scm xtreme.csv)

Insights-

1. Sometimes the model learn in such a way that artifact predictions are too much in number and many non artifacts are also classified. So we need to choose the divide number properly and examine some files by plotting

1. **Removing artifacts and predicting with other models -** This file removes the artifacts saved in the artifacts_predicted_svm.csv file, scale the features and predict
    a. Remove all the chunks whose indices are saved in artifacts_predicted_scm.csv file
    b. Extract features
    c. Preprocess the features using min max scaler
    d. Predict

    Insights -
    1. We cannot use mean subtraction from each chunk separately to classify between the mind stages, it reduces accuracy - Amplitude has a key role in predictions
    2. Preprocessing using standard scaler also had a bad impact on the prediction

References -

1. https://pandas.pydata.org/
2. https://www.numpy.org/
3. https://scikit-learn.org/
4. https://matplotlib.org/
5. https://docs.python.org/3/library/os.html
6. https://matplotlib.org/api/_as_gen/matplotlib.pyplot.scatter.html
7. https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html
8. http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
9. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
10. https://en.wikipedia.org/wiki/Decision_tree
11. https://en.wikipedia.org/wiki/Naive_Bayes_classifier
12. https://en.wikipedia.org/wiki/Support-vector_machine
13. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html
14. https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
15. https://scikit-learn.org/stable/modules/preprocessing.html