**Binary classification of two workload level(low->class 1,high->class 2)**

**of users during multitasking**

**TASK-1,2:**

1. First I have **plotted** the **given signals** ( in total 14) to visualize how the points are distributed.
2. Then i have **checked** is there any **missing value** or **value=0** volt i.e. not acceptable very much. But From the result i got i saw there is no missing value and very few Zero values are there
3. Next I remove the **noise** (**Artifact Removal**) using a method named **ICA**(Independent Component analysis). In **python** I have used **Scipy library** to do this. Here I used a method named **FastICA()** and **FastICA().fit\_transform()** to do this noise removal process.
4. After that again I **plotted the noise free signal** to visualize the points.
5. From every signal I have **extracted 5 EEG bands** (alpha,beta,gamma,delta,and theta)

Using Analog **BandPass Filter** (**Butterworth filter** of order 10 )

**NOTE : Analog filter of order 10 is practically very hard to implement.but i have used this just for simplicity as more the order of the filter more it closer to be an ideal filter.**

1. Then I saved 5 CSV files containing the 5 EEG bands of the 14 signals for further analysis

**Task-3:**

1. I have applied a **temporal filter(** window size 500**)** over **14 EEG signals.** I have made a plot **(14\*3)**

Each row has **1.** Plot a signal using **temporal filter over class 1**

**2**. Plot a signal using **temporal filter over class 2**

**3**. Plot a signal using **temporal filter over both class 1 and class 2**

1. Then save that result in a CSV file named “**avg\_.csv**”

Task- 4:

1. The CSV files that is generated from Task-1,2 is being used to analyze the features of each EEG bands of each signal
2. I did analysis over each bands ie. Alpha,Beta,Gamma,Delta and Theta . And extract some features from that.
3. First i did the **box plot**
4. Then i **plot the signal in time domain**(Divided that in 2 classes)
5. Then I have also given the **description** of that signal of **individual classes** like **5-points summary (min,25%,50%,75%,max),mean, median,variance** etc**.**
6. Then I have plotted the each band of each signal in Freq Domain **(Amplitude Spectrum**) .
7. Then i have plotted the **PSD(**|H(w)|^2 vs W graph**)** of that band of each signal in **db(20\*log10) scale**
8. Then I have plotted the **phase spectrum** of each band of each signal

**C**. Then I did the **spectrogram** **analysis** separately in a jupyter notebook.there i have checked the **Spectrogram of each class(1 and 2)** and under-stand how they differ in each time and each freq.

Task- 5:

1. Here i made a **RNN(Recurrent Neural Network)** using **LSTM**
2. The model summary is ----

Model: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

lstm\_1 (LSTM) (None, 1, 50) 13000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 1, 50) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_2 (LSTM) (None, 1, 50) 20200

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 1, 50) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_3 (LSTM) (None, 50) 20200

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 50) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 1) 51

=================================================================

Total params: 53,451

Trainable params: 53,451

Non-trainable params: 0

Note :I did **PCA(principal component analysis)** after applying **ICA().But I can not reduce the input dimensions as the signals all of them play a significant role for classifying into the two classes.That is why I did not do dimensional reduction over the independent 14 signals.**

1. Then I split the whole dataset into the train test using **train\_tes\_split()** into an 80-20ratio.
2. But i got accuracy near about 45.77% which is not so much appreciable.The paper i have read there they have used SVM classifier to solve this problem and they got an accuracy 69.5%
3. Hence I conclude that here **ML is a better choice over DL.**