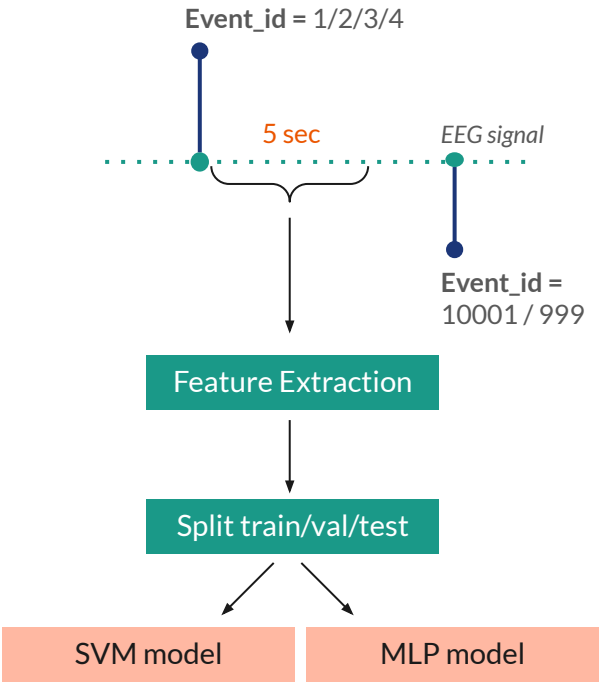
Table of Contents

- Task 1 (SSVEP data)
- Task 2 (P300 data)
- Task 3 (VEP data)
- Task 4 (EMG data)
- My learnings

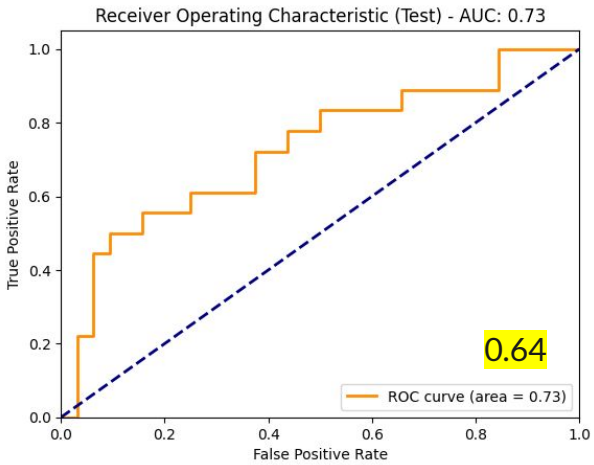
*These results were obtained earlier before the presentation

Task 1: Binary classification of SSVEP response based on focused flickering stimuli

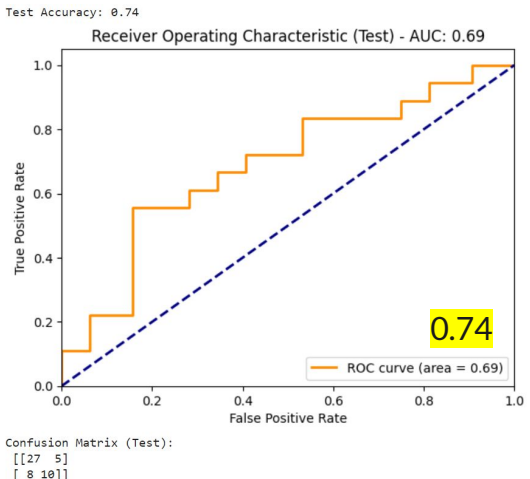


- Signal processing: Notch filter, bandpass filter, wavelet transform
- Feature extraction: Linear envelope + RMS, CWT, FFT stats, DWT stats, FFT stats + entropy, all combined, all combined w PCA (principal_df)
- Classes: **10001** (7.5 Hz confirmed stimuli), **999** (confirmed no stimuli)

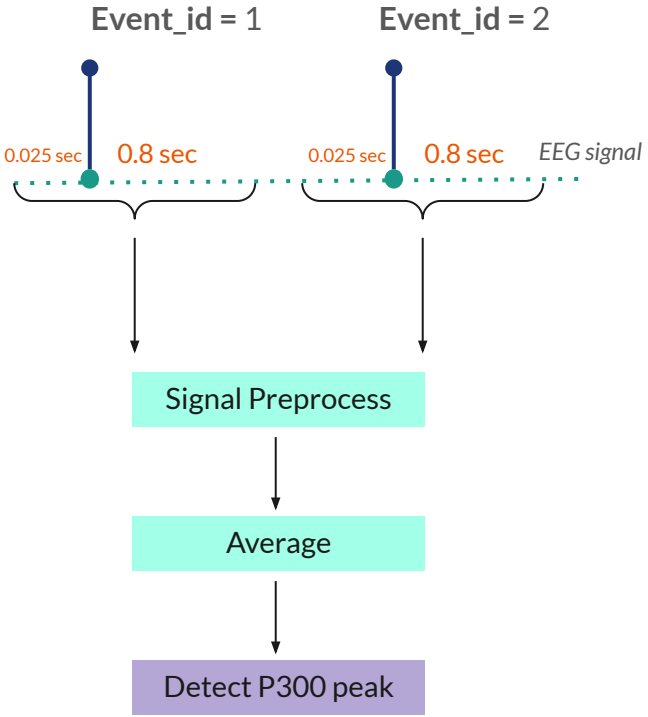
SVM accuracy* for all features principal_df components

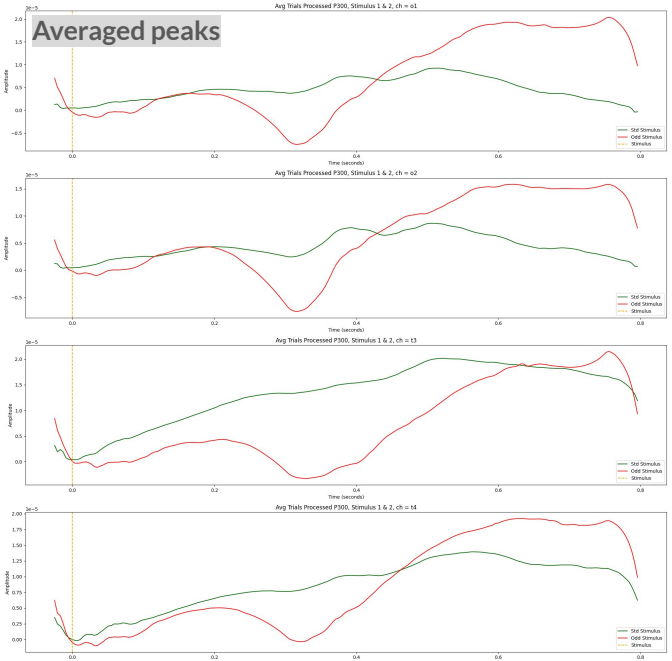


Hyper parameter-tuned MLP accuracy* for FFTstat_entropy features



Task 2: Detecting P300 peak from stimuli 1 (std) & stimuli 2 (odd)





Channel	o1	o2	t3	t4
Amplitude	-0.000016	-0.000014	-0.000010	-0.000008
Latency	0.315413	0.315413	0.331432	0.315413
Index	60.000000	60.000000	64.000000	60.000000

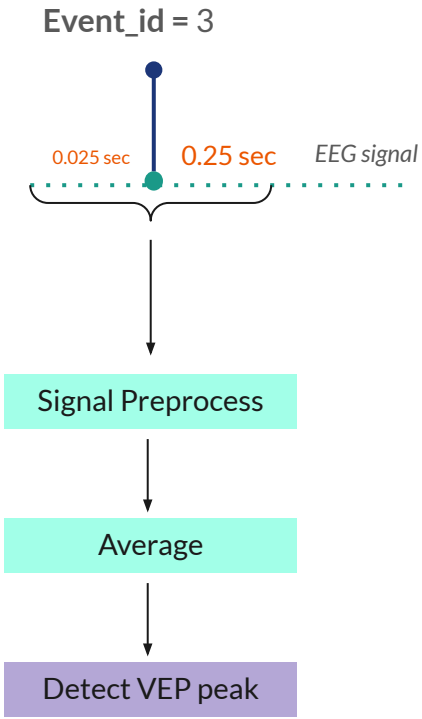
Preprocessing signals: Data imputation, baseline correction, notch filter, bandpass filter, wavelet transform

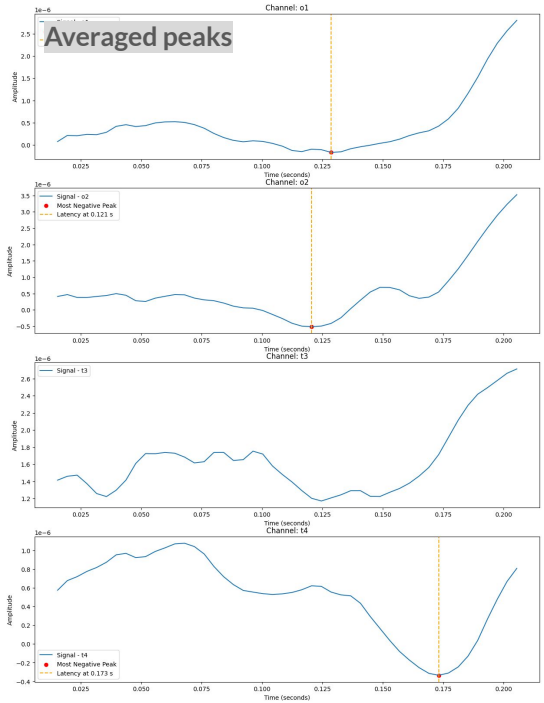
Odd stimuli

- Stronger response
- > 7 uV
- Amplitude and latency detection method: 'find_peaks' from from scipy.signal

Improvement: Could remove outlier files to improve signal to noise ratio while averaging

Task 3: Detecting VEP peak from stimuli (event_id = 3)





Channel	o1	o2	t4
Amplitude	-1.682060e-07	-5.134233e-07	-3.344624e-07
Latency	1.286765e-01	1.205882e-01	1.731618e-01
Index	2.800000e+01	2.600000e+01	3.900000e+01

Preprocessing signals: Data imputation, high pass filter, notch filter, bandpass filter, wavelet transform

VEP response

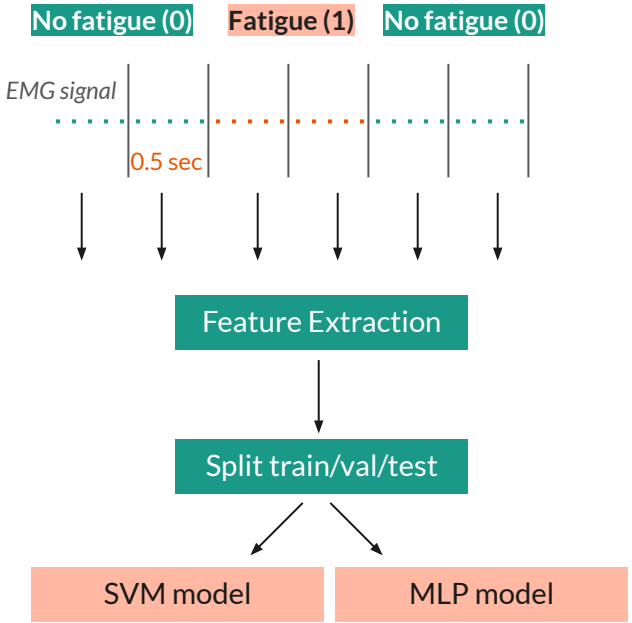
- Negative peak
- Shape partially visible after averaging
- Peak too small after averaging
- Amplitude and latency detection method: 'find_peaks' from from scipy.signal

Improvements

- Could remove outlier files to improve signal to noise ratio while averaging
- More refined preprocessing techniques to remove noise, quantify using signal to noise ratio

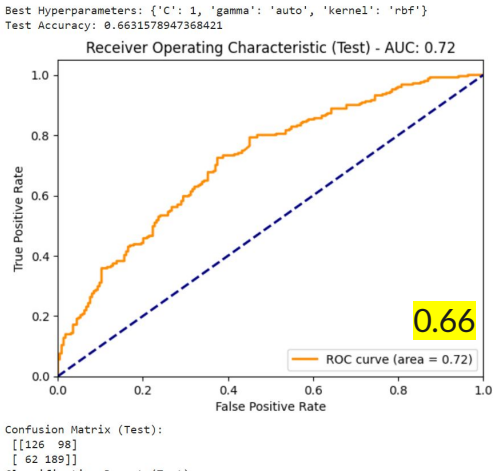
*These results were obtained earlier before the presentation

Task 4: Binary classification of fatigue and no-fatigue response from EMG segments

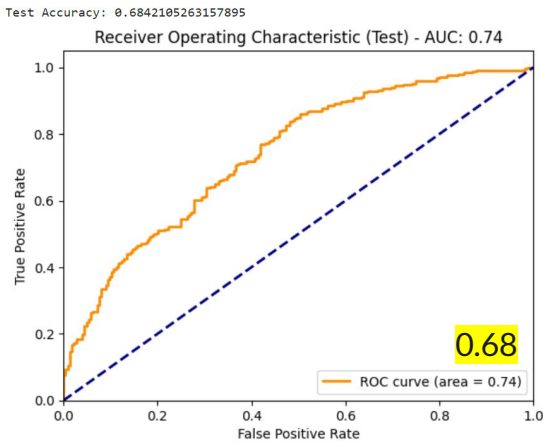


- Signal processing: Notch filter, wavelet transform, bandpass filter
- Feature extraction: Linear envelope + RMS, CWT, FFT stats, DWT stats, FFT stats + entropy, all combined, all combined w PCA
- Classes: **1** (fatigue), **0** (no fatigue)

Hyper parameter-tuned SVM accuracy* for all features data



MLP model accuracy* for all features data



My learnings

- EEG data is complex, requires higher level of analysis (outlier detection through clustering) and more robust processing techniques.
- EEG signals are sensitive and unique, would need to ensure each data point is useful for analysis (through thresholding or a quantitative assessment). Could lead to class imbalance for machine learning.
- Machine learning is an iterative process, would only achieve a good accuracy when you train with more data and experiment with more features and hyper parameters.
- More complex machine learning models would be required to perform accurate classification (CNN, ResNET, Transformers, deeper MLP).

Thank you!