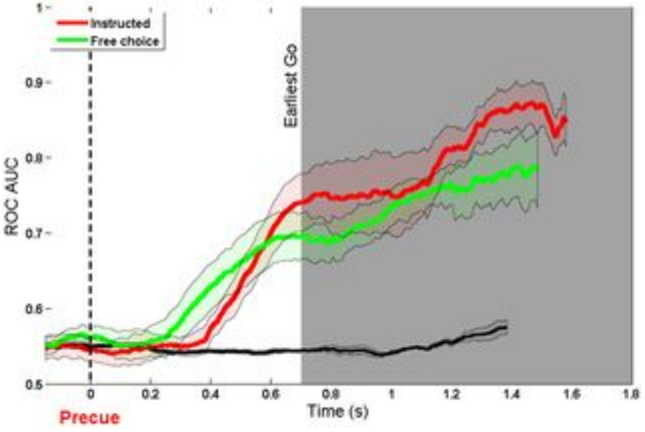
Decoding motor intentions

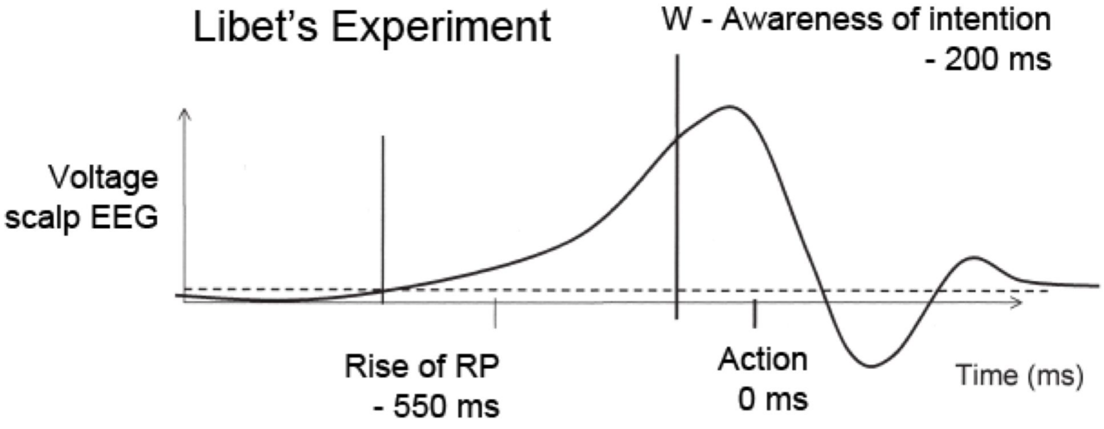
An EEG and machine learning approach

How free are free decisions?

If the brain is governed by deterministic physical processes - when asked to make a ‘free decision’, are we really free or is the decision already encoded in our brains?

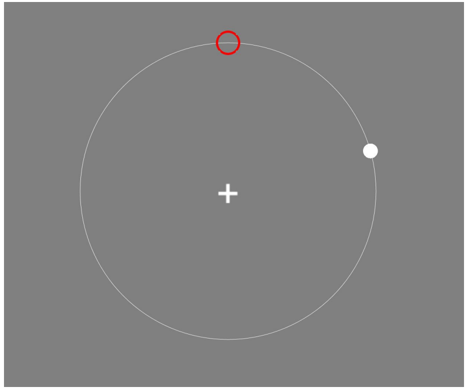


Salvaris & Haggard, 2014



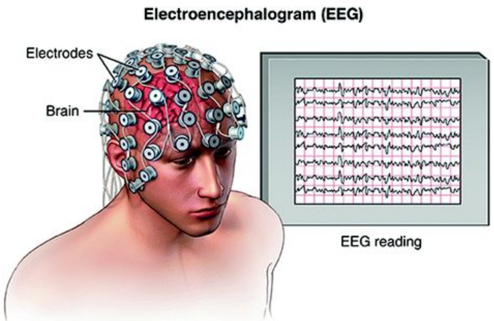
Libet, 1983 (figure from https://wmpeople.wm.edu/asset/index/cvance/libet)

The task and the data



- Left/Right button press with two conditions:
    - Condition A: Instructed
    - Condition B: Spontaneous - free

- We took EEG recordings (64 channels on the scalp) of 2 participants
    - Subject 1: 100 trials condition A, 100 trials condition B
    - Subject 2: 226 trials condition A, 225 trials condition B



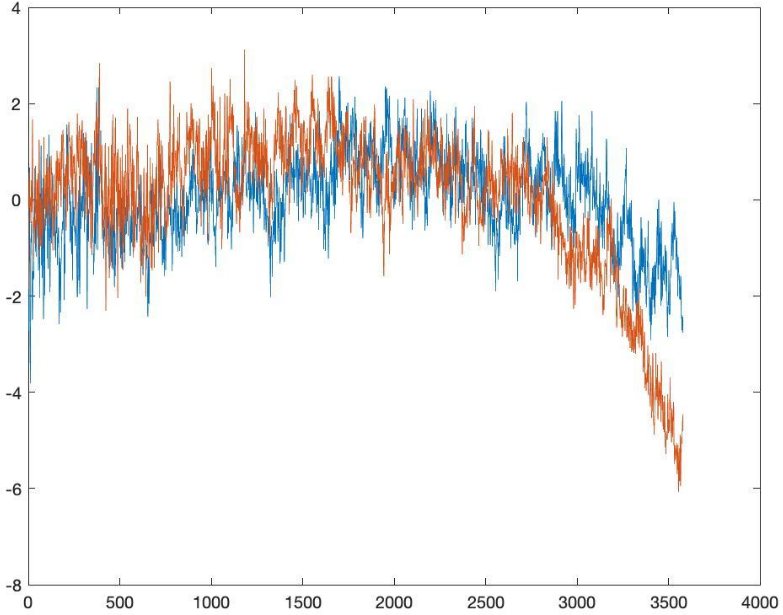
(Figure from Sylius, 2017)

Preprocessing steps

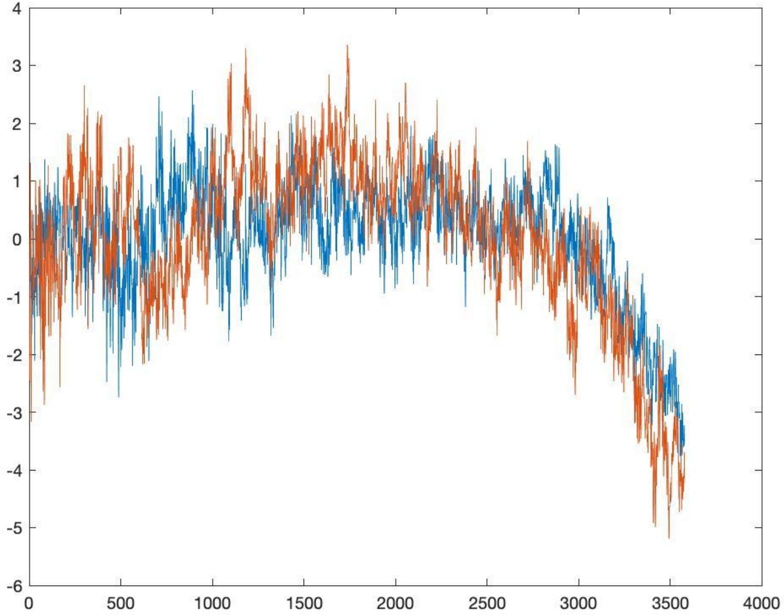
- Eyeblinks were removed with independent component analysis (ICA)
  - Using the fieldtrip toolbox in MATLAB
- Muscle and channel artifacts
  - removed visually (i.e. manually) in MATLAB
- The data was transformed into epochs of 2 s length preceding the movements
  - Each epoch was associated with a condition A/B label
  - Each epoch was associated with a Left/Right press label
- Filtering [not yet implemented]
  - The data was filtered using a lowpass filter of 35 Hz to remove high-frequency noise

Lateralized Readiness Potential

Condition A: Instructed



Condition B: Spontaneous



Orange: Left-hand movement
Blue: Right-hand movement

Hyperparameter, Model and Dimensionality Reduction Method Selection:

Models:

1. Linear Discriminant Analysis Classifier
2. Support Vector Classifier
3. Random Forest Classifier
4. AdaBoost Classifier

Dimensionality Reductions:

1. Take Lateralized Readiness Potential from Data Cube
2. Use MNE CSP to reduced dimensionality of the Data

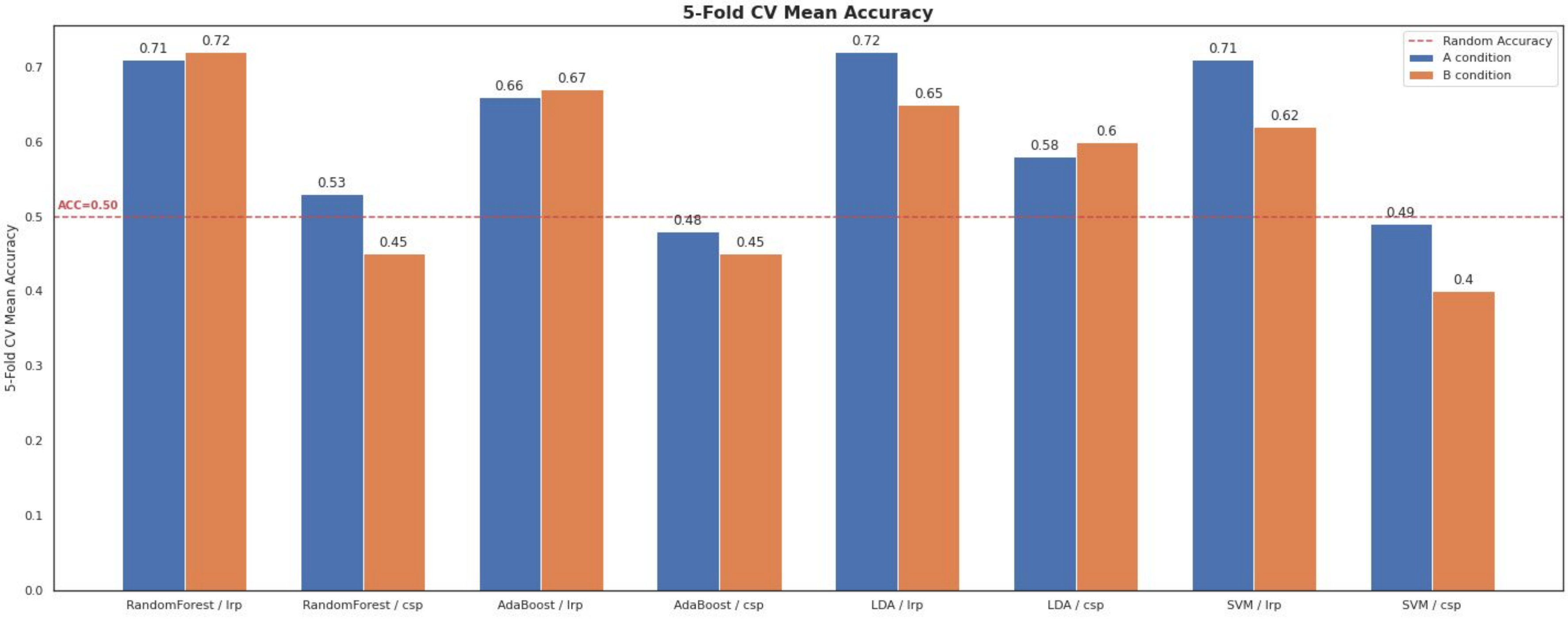
Testing Pipelines

For each clf in [LDA, SVC, RandomForest, AdaBoost]:

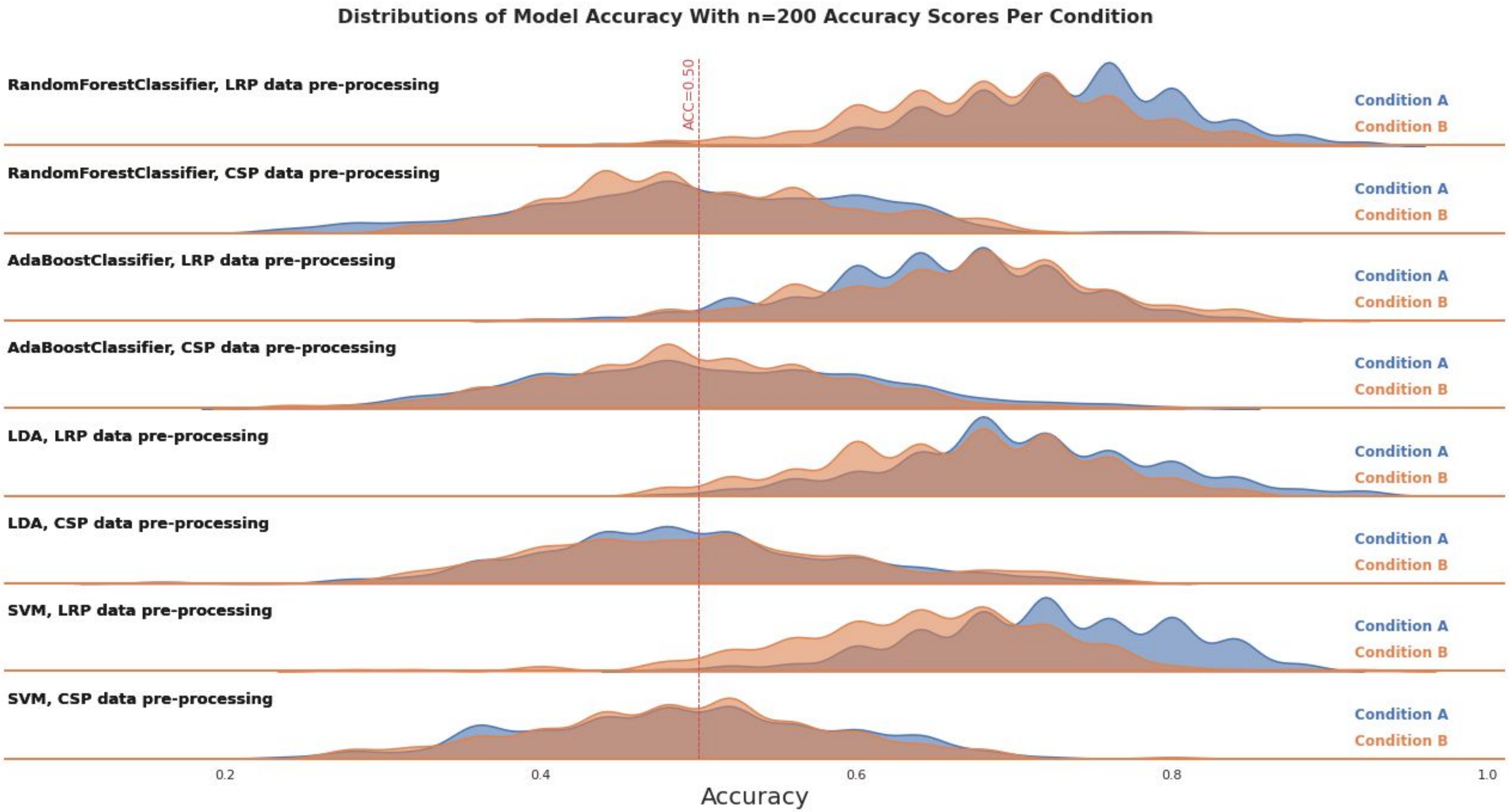
- Lateralized Readiness Potential:
  - Hyperparameter Tuning:
    - Cleaned data ⇨ Transformed Data ⇨ Hyperparameter Tuning
  - Model Performance Evaluation:
    - Cleaned data ⇨ Transformed Data ⇨ Train Test Split ⇨Accuracy Score
- CSP:
  - Hyperparameter Tuning with Default CSP n_components:
    - Cleaned data ⇨ Fit CSP to subset of Cleaned Data
    - ⇨ Transformed Data ⇨Hyperparameter Tuning
  - N_components Tuning Via 5-fold CV:
    - Cleaned data ⇨Test Train Split ⇨ Fit CSP to Training data ⇨ Transformed Data
    - ⇨ Accuracy Score ⇨Select best n_component
  - Model Performance Evaluation:
    - Cleaned data ⇨Test Train Split ⇨Fit CSP to Training Data ⇨Transformed Data
    - ⇨ Accuracy Score

Major Disadvantage to CSP: 30min cross-validated tuning time

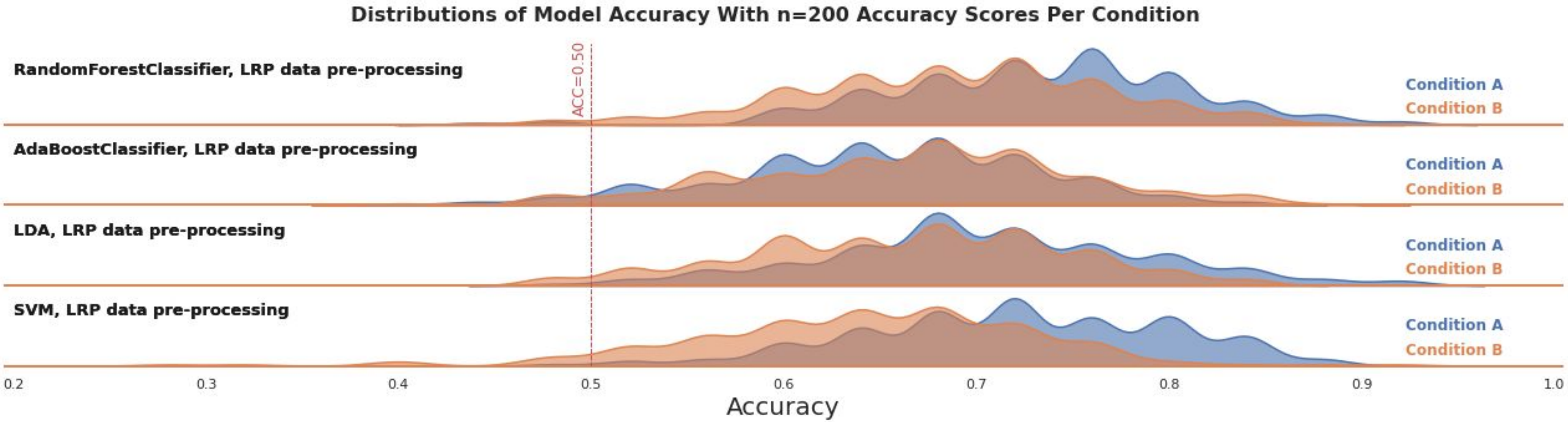
Model Evaluation Results:



Model Evaluation Results:



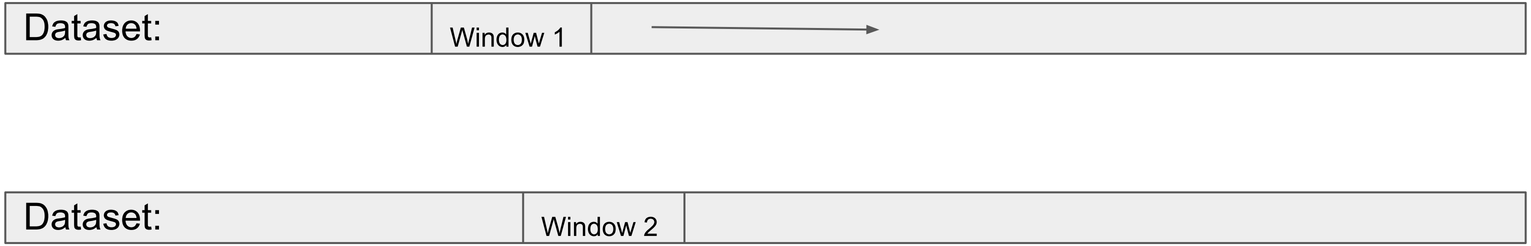
Model Evaluation Results



	RandomForest / lrp	AdaBoost / lrp	LDA / lrp	SVM / lrp
Hyperparameter Tuning Time	1hr 19min 19s	0hr 31min 54s	0hr 1min 47s	0hr 1min 49s

Total Model Tuning, Evaluation and Selection Time: 6hrs

Predictions over a Sliding Window



Local Modeling

On each window:

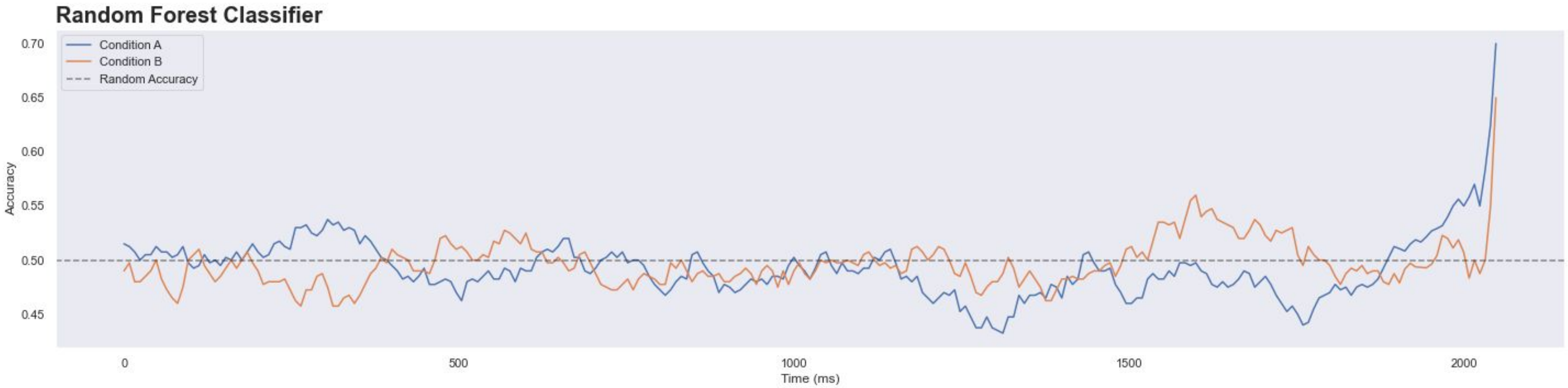
- Split into training and testing set
- Dimension Reduction
- Fit using globally tuned model
- Generate an accuracy score

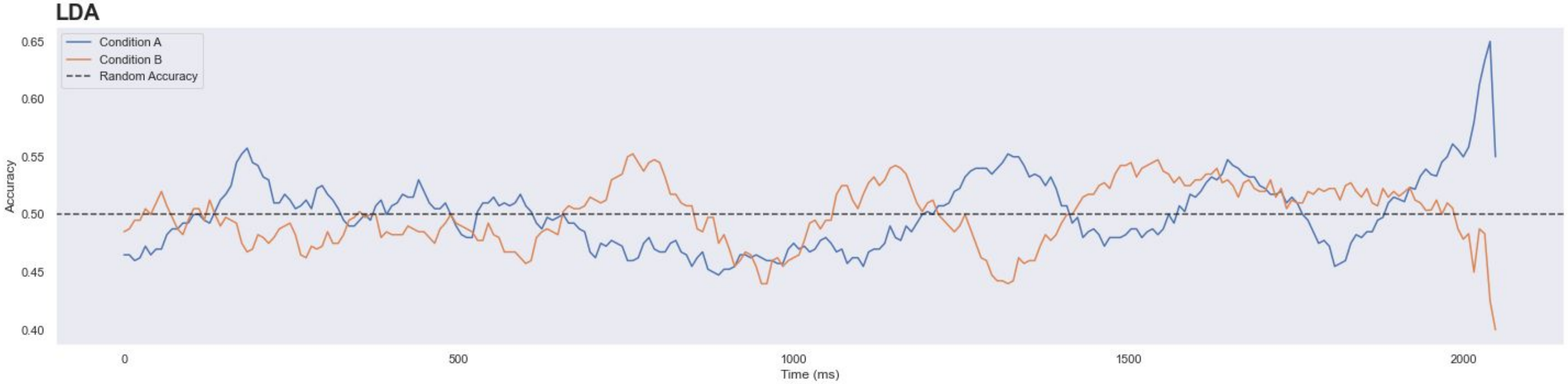
Hopefully we can see at what point the pertinent information for predicting LH or RH becomes active in the brain!

Local CSP

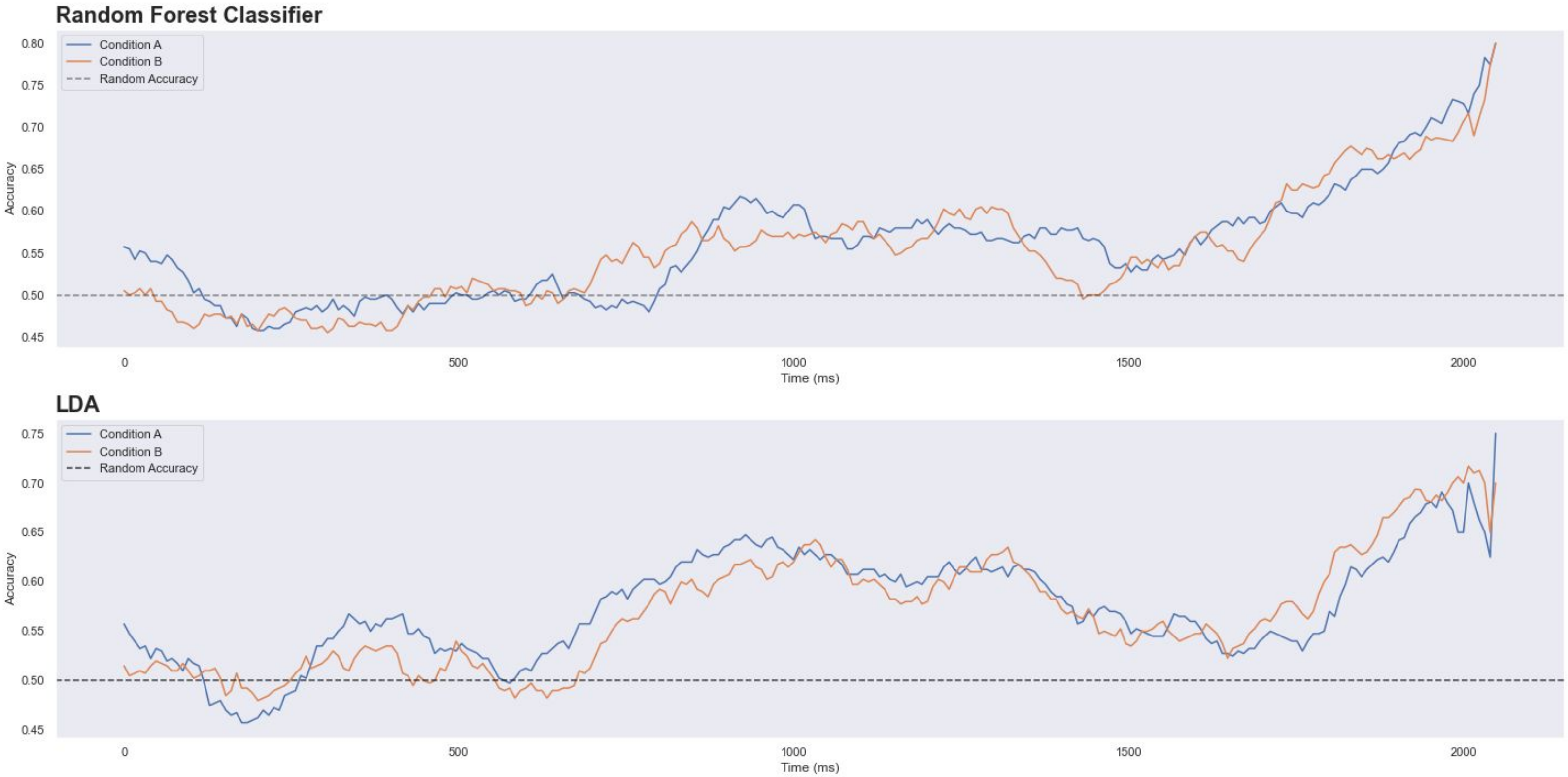
Although CSP was shown to be an unsuccessful dimension reduction method when testing on the entire dataset, how will it perform when fitted locally?

Results using CSP





Results using LRP



Next Steps

- Reject CSP, continue examining LRP
- Tune LDA hyperparameters locally
- Tweak sliding window parameters to explore time windows of interest
  - Window length
  - Step size
  - Rolling average

Conclusion

- Although RF works best, LDA requires very little processing power and time for hyper-parameter tuning
- Accuracies of up to 0.72 can be attained
- Our sliding windows results suggest that information is there well before the participants make their decisions