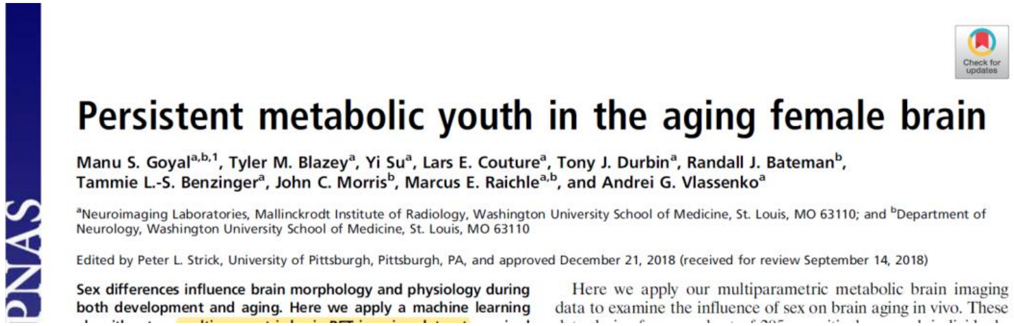
EEG/ERP & Machine Learning Regression Models

-a project framework testing on ChongQing EEG studies

Yaozhong Li

PNAS paper on RandomForest & PET


PNAS

Persistent metabolic youth in the aging female brain

Manu S. Goyal[a,b,1], Tyler M. Blazey[a], Yi Su[a], Lars E. Couture[a], Tony J. Durbin[a], Randall J. Bateman[b], Tammie L.-S. Benzinger[a], John C. Morris[b], Marcus E. Raichle[a,b], and Andrei G. Vlassenko[a]

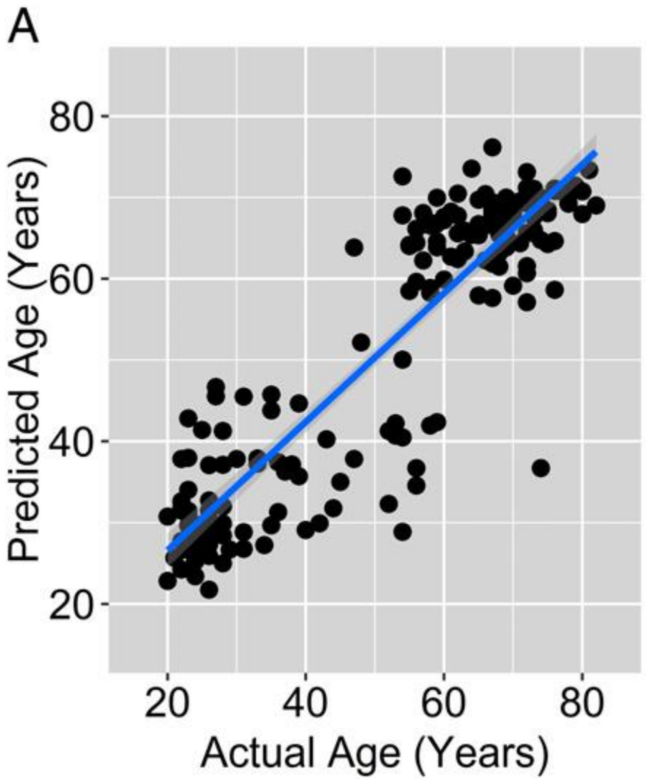
[a]Neuroimaging Laboratories, Mallinckrodt Institute of Radiology, Washington University School of Medicine, St. Louis, MO 63110; and [b]Department of Neurology, Washington University School of Medicine, St. Louis, MO 63110

Edited by Peter L. Strick, University of Pittsburgh, Pittsburgh, PA, and approved December 21, 2018 (received for review September 14, 2018)

Sex differences influence brain morphology and physiology during both development and aging. Here we apply a machine learning

Here we apply our multiparametric metabolic brain imaging data to examine the influence of sex on brain aging in vivo. These

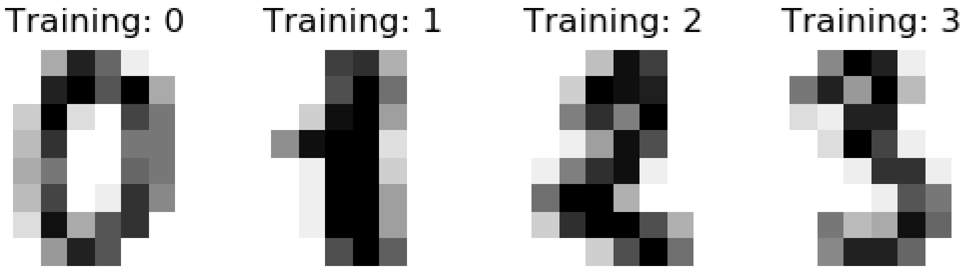
1. **Collect PET data from 200+**
2. **Train on a RandomForest Regression Model**
3. **Predict age from PET (Brain Age)**
4. **Conclusion:**
   - ✓ **Find way to represent Brain Age**
   - ✓ **Women's brain aging slower**

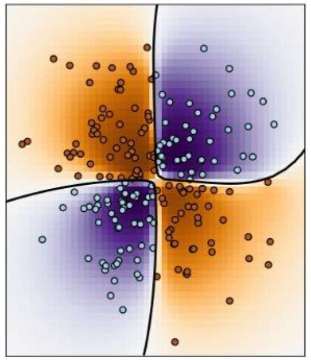


"Ten-fold cross-validation demonstrates that the predicted age based on this algorithm—defined as metabolic brain age—closely matches the actual chronological age of the participants (Pearson's r = 0.88–0.90 over 10 runs)"

PNAS paper on RandomForest & PET

- **Not a traditional experiment**
  - Data driven, no hypothesis or manipulation
- **Nor a traditional modeling**
  - Unknown ‘HOW’ and ‘WHAT’
- **Yet if strong enough … ?**

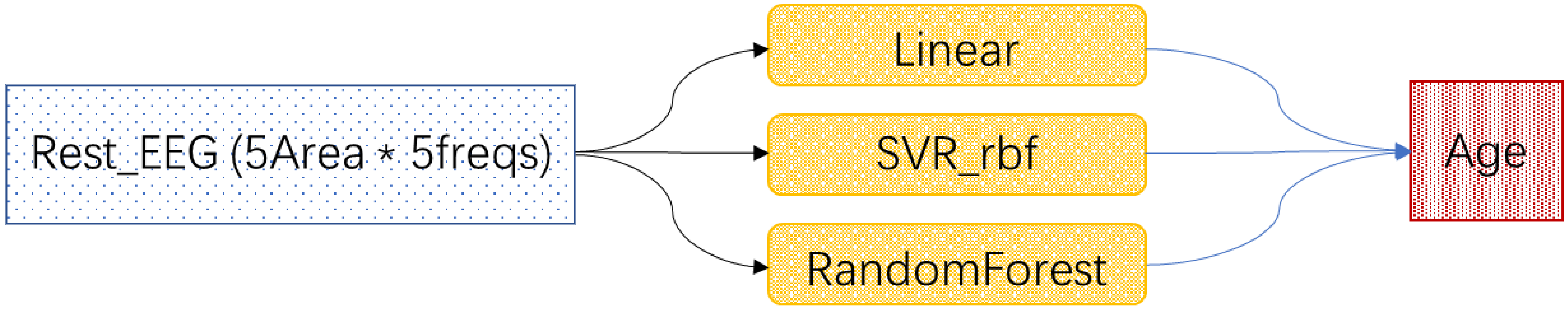




Blueprint

Setting principles in during constructing and running

Constructing : Features–Models–Target



- **Features – Models – Target**
  - Features : sets of EEG/ERP data
  - Models : regression models
  - Target : questionnaire result

-Features

- 16 sets of features

Group	Feature Sets	Length	Contains
Pain_EEG	pain_eeg_norm	400	Pain(2) * Pic_gender(2) * TimeWindow(4) * Freq(5) * Channels(5)
	pain_eeg_norm_Pain_	200	
	pain_eeg_norm_Neutral_	200	
	pain_eeg_orig	400	
	pain_eeg_orig_Pain_	200	
	pain_eeg_orig_Neutral_	200	
Pain_ERP_amplitude	pain_erp_amplitude_norm	80	Pain(2) * Pic_gender(2) * TimeWindow(4) * Channels(5)
	pain_erp_amplitude_norm_Pain_	40	
	pain_erp_amplitude_norm_Neutral_	40	
	pain_erp_amplitude_orig	80	
	pain_erp_amplitude_orig_Pain_	40	
	pain_erp_amplitude_orig_Neutral_	40	
Pain_ERP_peak	pain_erp_peak	16	Pain(2) * Pic_gender(2) * TimeWindow(4)
Rest_EEG	rest_norm	25	Freq(5) * Channels(5)
	rest_orig	25	
*Test	test_X	50	25 informative + 25 noise

-Models

- 12 regression models

```
def model_set_2():
    model_dict = dict()
    model_dict['linear'] = get_linear()
    model_dict['riged'] = get_riged(alpha=0.1)
    model_dict['lars'] = get_lars(alpha=0.5)
    model_dict['bayesian'] = get_bayesian()
    model_dict['svr_rbf'] = get_svm(C=10.0, gamma='scale')
    model_dict['svr_linear'] = get_svm(kernel='linear',C=10.0, gamma='scale')
    model_dict['svr_poly'] = get_svm(kernel='poly',C=10.0, gamma='scale')
    model_dict['gaussian'] = get_gaussian(alpha=0.001)
    model_dict['kneighbors'] = get_kneighbors(n_neighbors=5)
    model_dict['random_forest'] = get_random_forest(max_depth=50, n_estimators=5000)
    model_dict['gradient_trees'] = get_gradient_trees(max_depth=5, n_estimators=1000)
    model_dict['neural_network'] = get_neural_network(hidden_layer_sizes=(100,50,))
    return model_dict
```

http://github.com/MoonKuma/pandas_regression/blob/master/README.md

-Target

- 14 targets

Group	Target
Age	age
Race	race_identity
Face recognition	face_recognition
Day Dream	day_dream
IRI	IRI_FS
	IRI_EC
	IRI_PT
	IRI_PD
Self	Interdependent
	Independent
Altruism	altruism
Anxiety	anxiety
Depression	depression
*Test	Test_Y

Constructing : Features–Models–Target

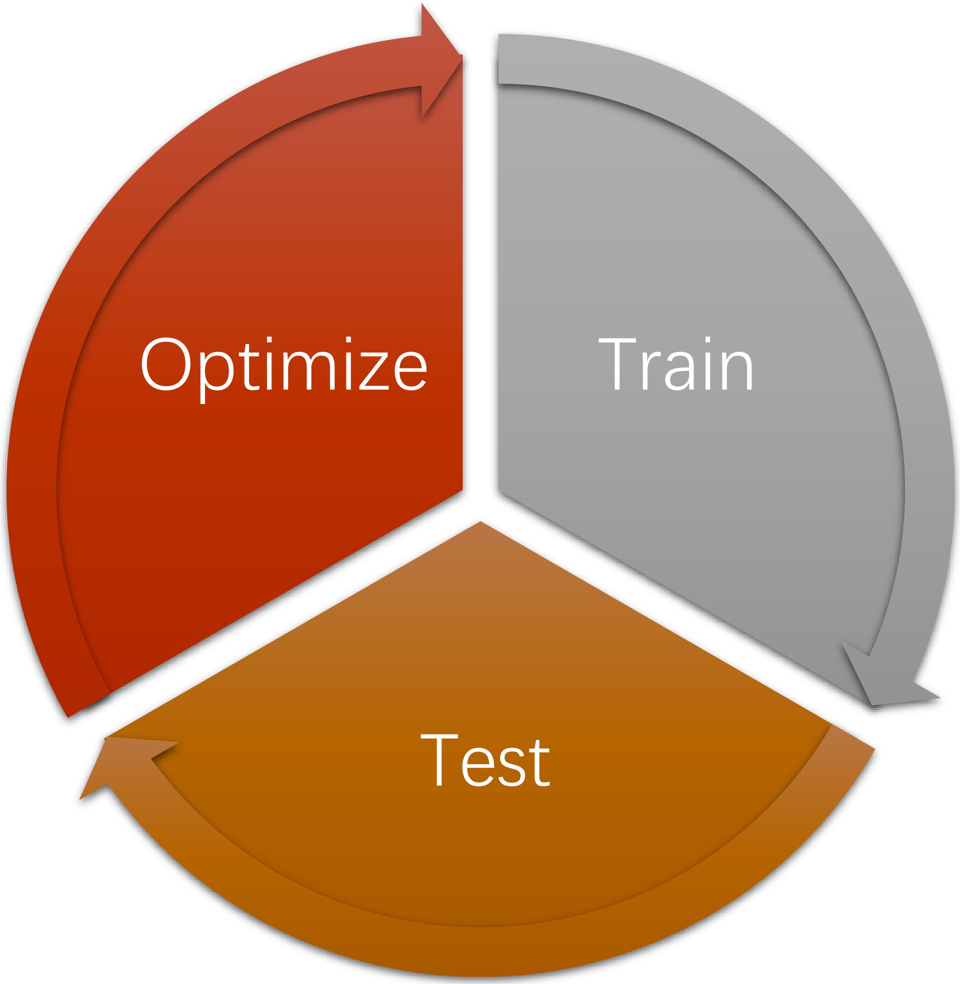
- 16 features * 12 models * 14 targets = 2688 train & test

Feature Sets
pain_eeg_norm
pain_eeg_norm_Pain_
pain_eeg_norm_Neutral_
pain_eeg_orig
pain_eeg_orig_Pain_
pain_eeg_orig_Neutral_
pain_erp_amplitude_norm
pain_erp_amplitude_norm_Pain_
pain_erp_amplitude_norm_Neutral_
pain_erp_amplitude_orig
pain_erp_amplitude_orig_Pain_
pain_erp_amplitude_orig_Neutral_
pain_erp_peak
rest_norm
rest_orig
test_X

Models
linear
riged
lars
bayesian
svr_rbf
svr_linear
svr_poly
gaussian
kneighbors
random_forest
gradient_trees
neural_network

Target
age
race_identity
face_recognition
day_dream
IRI_FS
IRI_EC
IRI_PT
IRI_PD
Interdependent
Independent
altruism
anxiety
depression
Test_Y

Running: Train-Test-Optimizing



- **Test: 10%-fold, 10 run times**
- **How to optimize**
  - Enhance data
    - Better preprocessing
  - Enhance Models
    - Tuning hyper-parameters
  - Speed up

Implementation

Preprocessing data, building models, testing, tuning parameters as well as self-checking

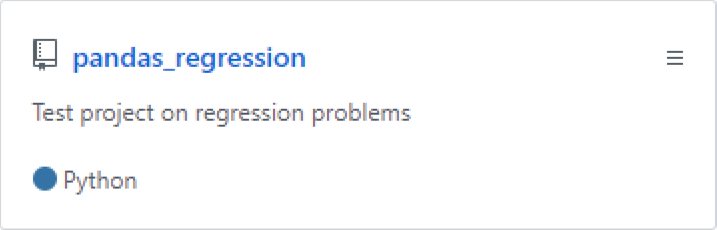
Two projects on Github

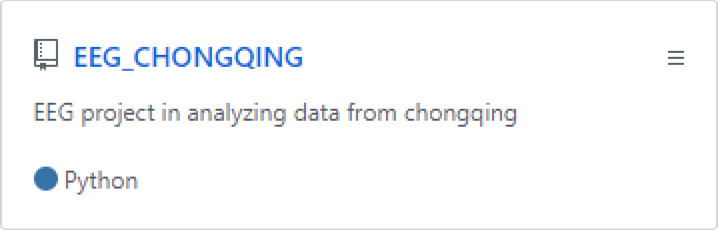
- **EEG_CHONGQING**

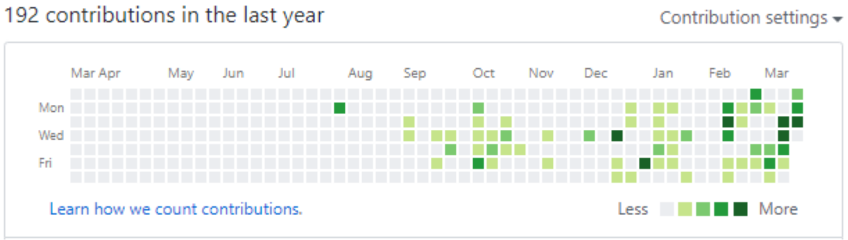
- Goal : EEG/ERP data pre-processing
- Packages
  - mne : Read cnt, ICA, Morlet, Epoch, Peak …
  - numpy : custom (avg across channel area)
- https://github.com/MoonKuma/EEG_CHONGQING

- **pandas_regression**

- Goal : structuring and training regression models
- Packages
  - pandas： store and preprocess data(rating questionnaire, merging)
  - sklearn : machine learning models
- https://github.com/MoonKuma/pandas_regression







EEG/ERP preprocessing

- Raw -> Power/Amplitude Epochs -> Time Window averaged
  - https://github.com/MoonKuma/EEG_CHONGQING/blob/master/eeg_pre_processing/pre-processing_pipeline.md
- Common procedures

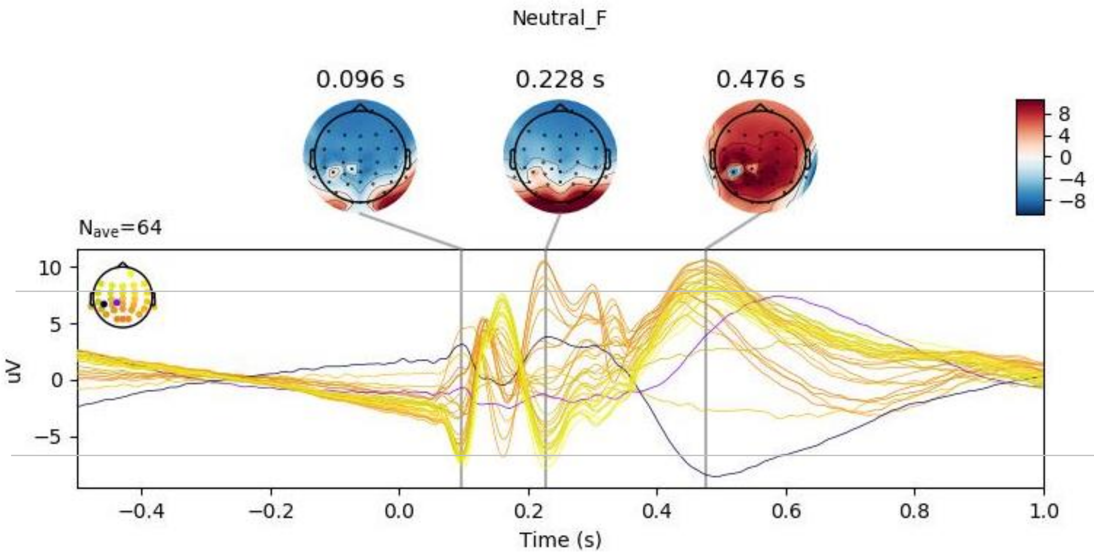
  Load, down-sample, Filter, ICA, Baseline, Epoch, Morlet, Average across time-window

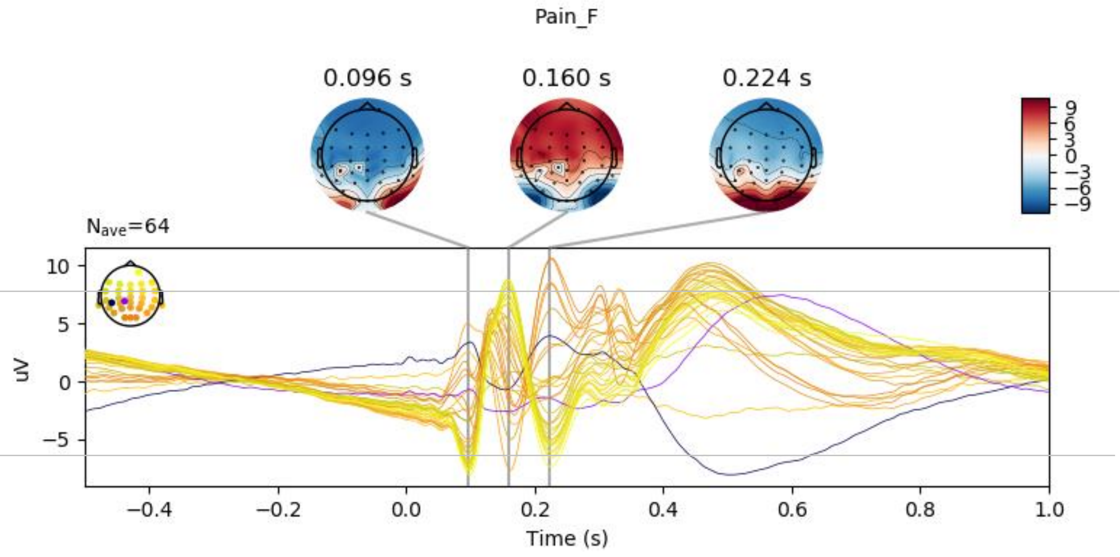
- Custom procedures
  - Events of resting states
    - 2s time window randomly separated in each 4s

  - Normalize (L2)
  - Average across channel areas
    - [F/C/T/P/O]

EEG/ERP – Self-checking

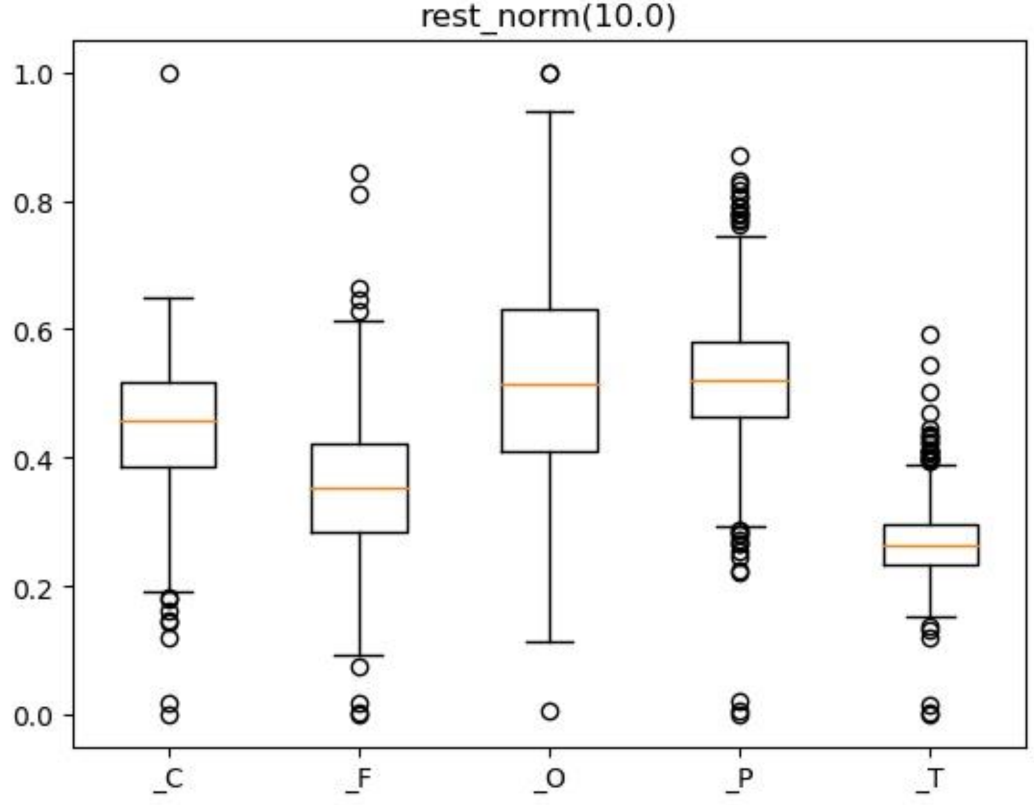
- Preprocessed

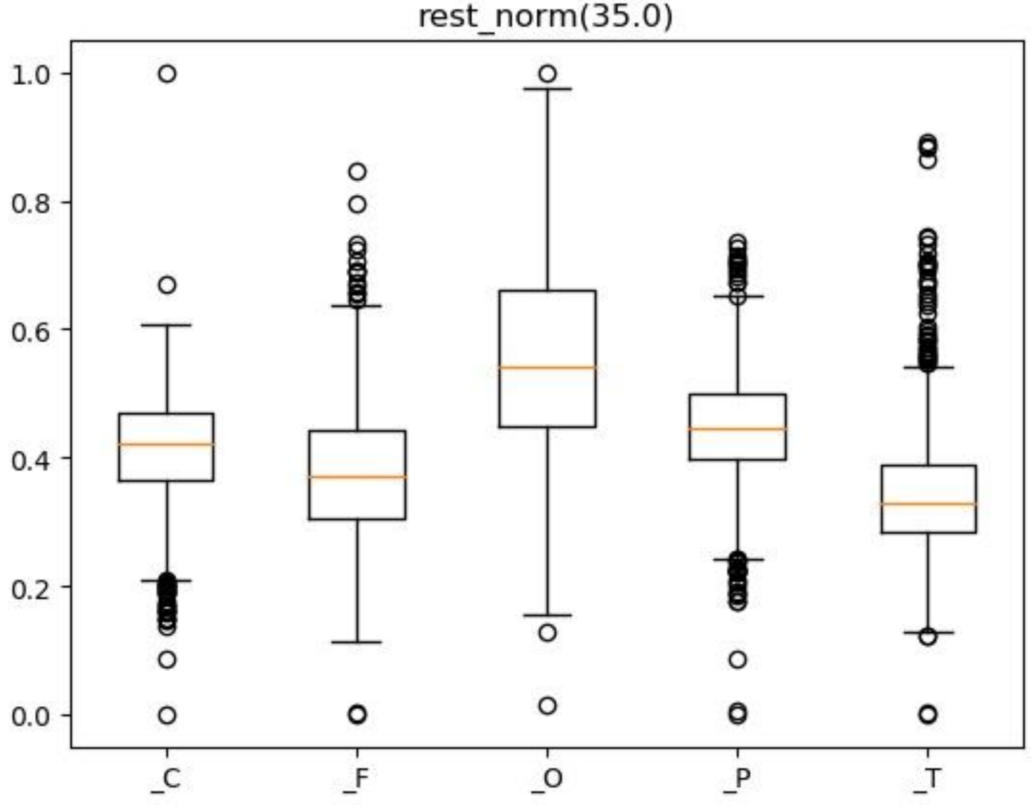




EEG/ERP – Self-checking

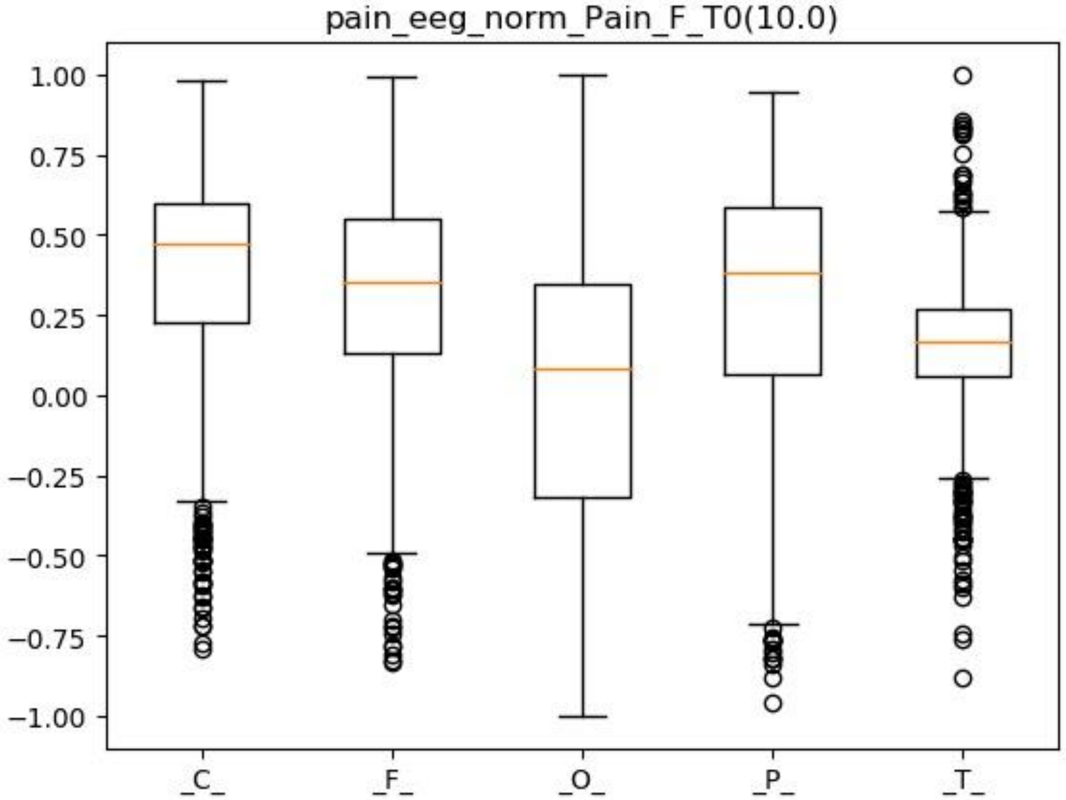
- Rest EEG across channels/freqs

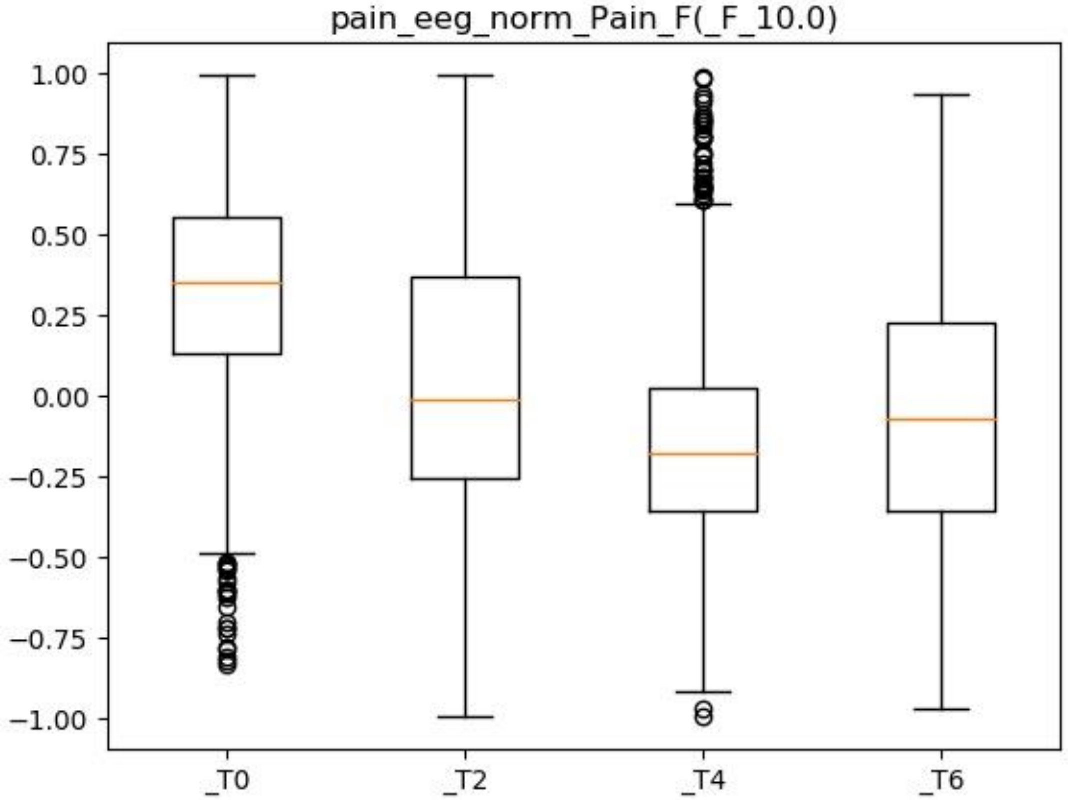




EEG/ERP – Self-checking

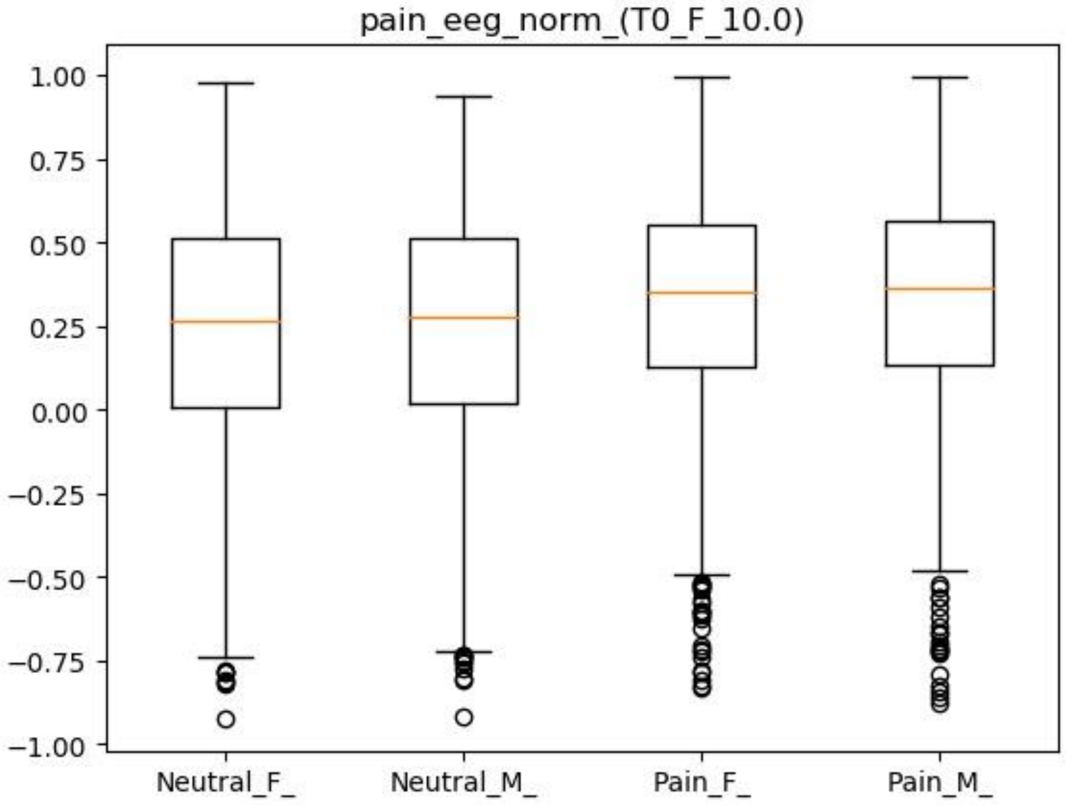
- Pain EEG across channels/time window

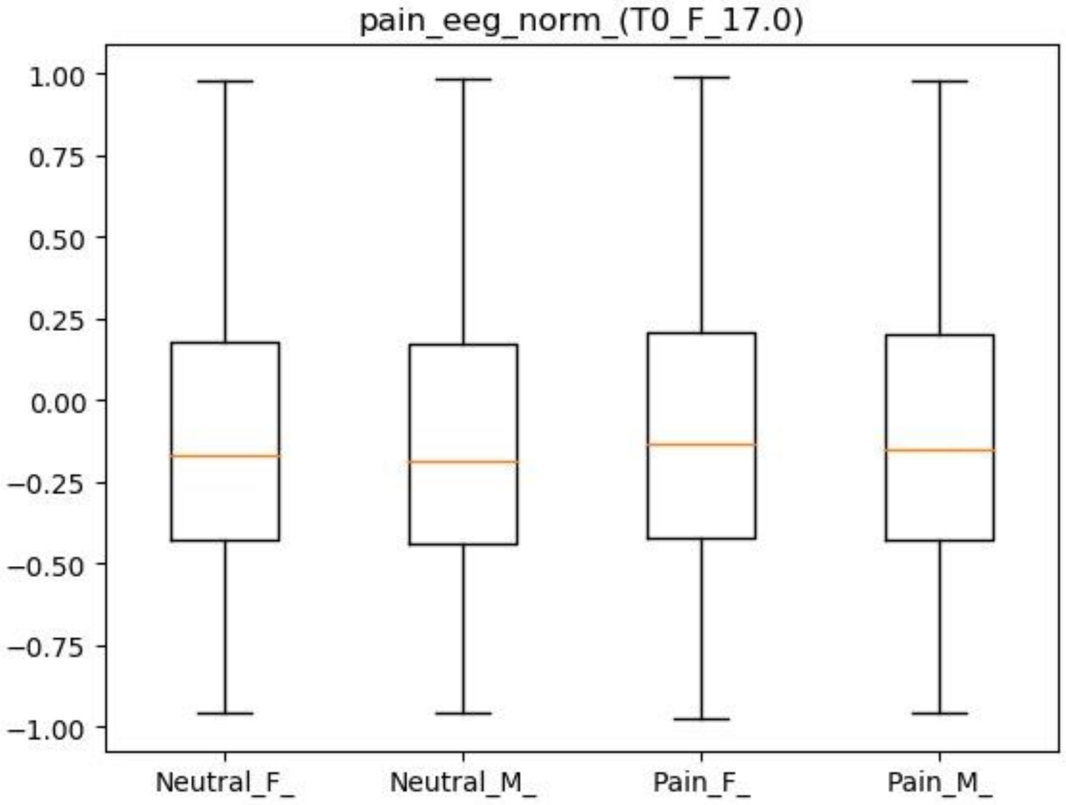




EEG/ERP – Self-checking

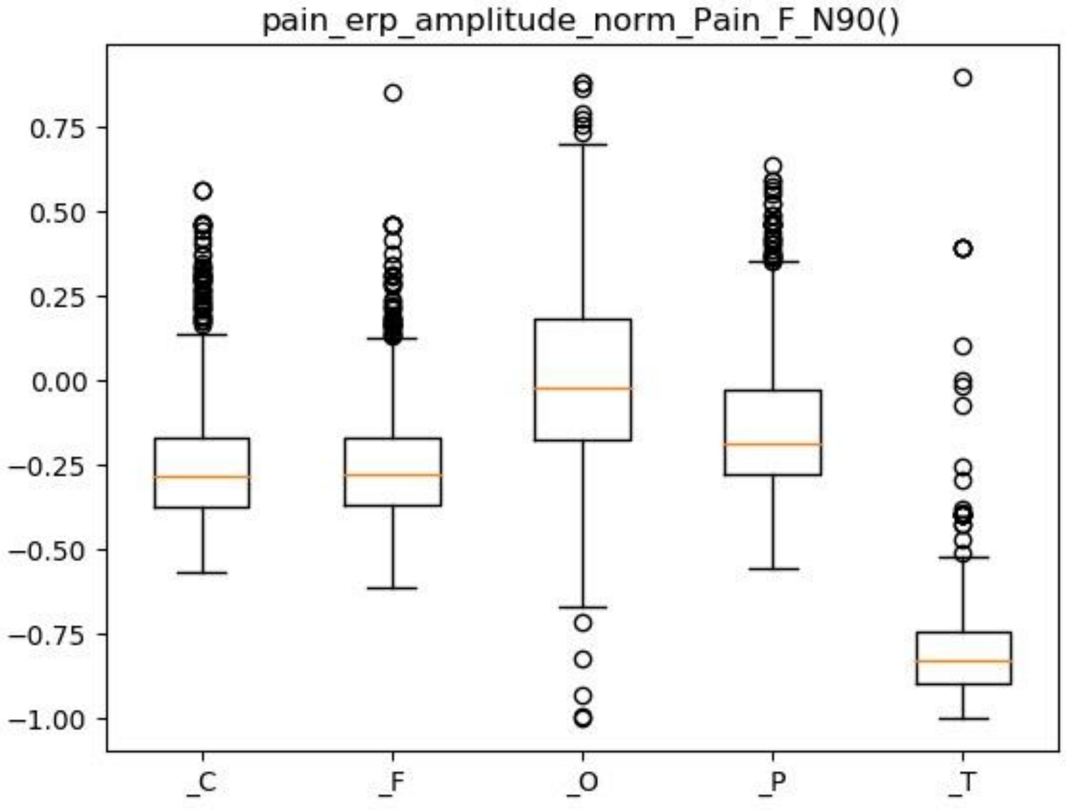
- Pain EEG across conditions (Pain vs. Neutral, Male vs. Female)

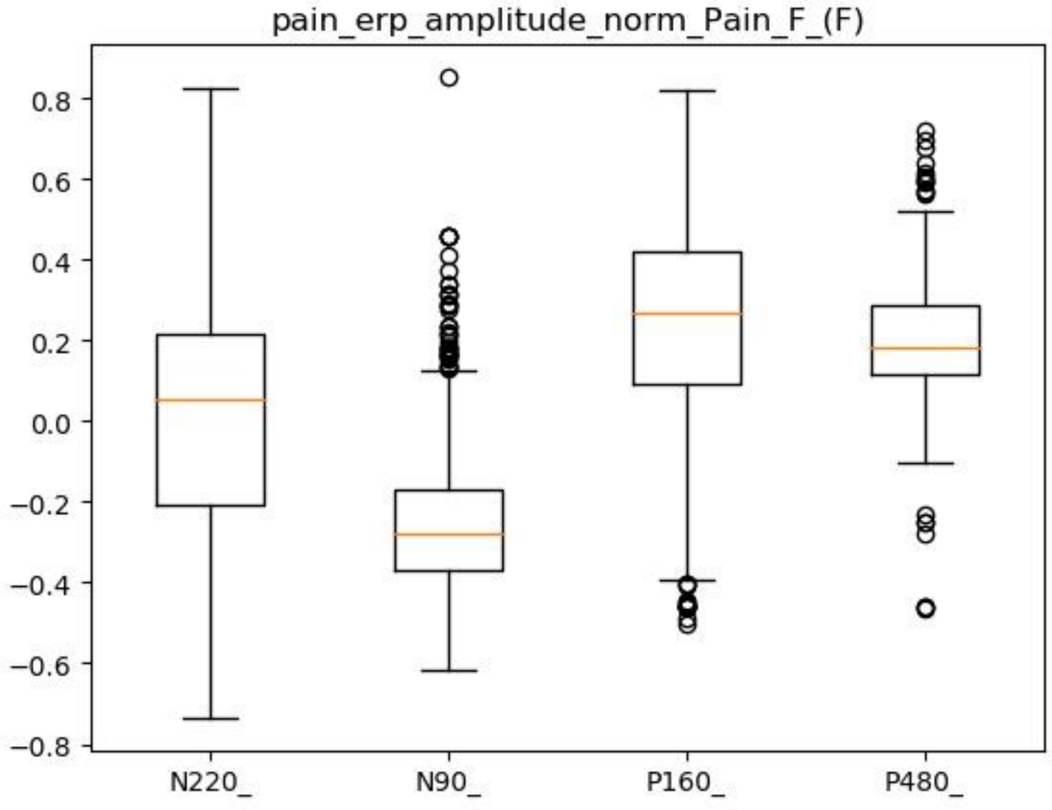




EEG/ERP – Self-checking

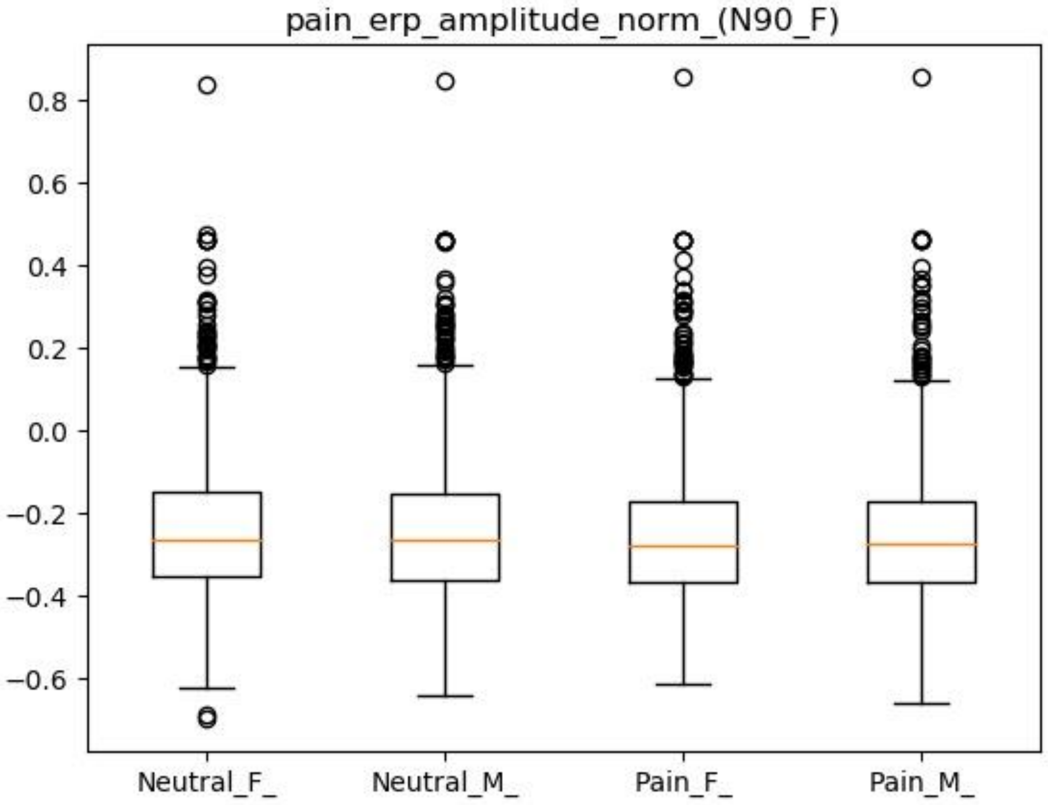
- Pain ERP across channels/time window

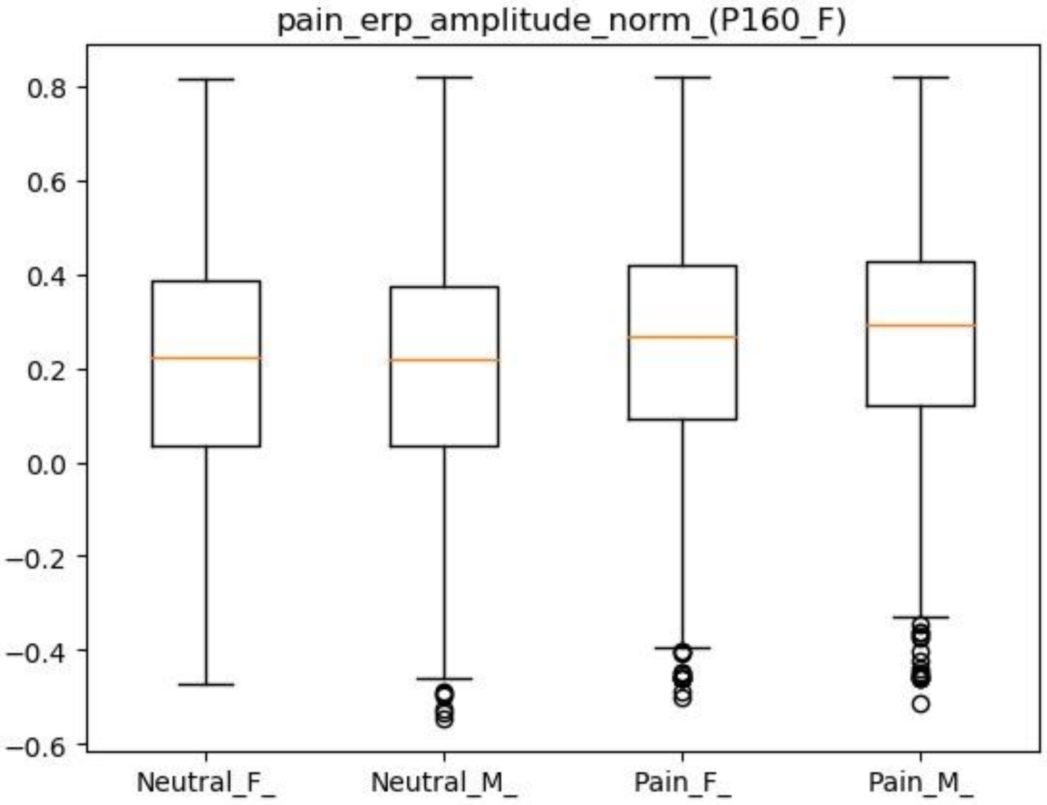




EEG/ERP – Self-checking

- Pain ERP across conditions (Pain vs. Neutral, Male vs. Female)





Result of model sets

- Model set 1

```
def model_set_1():
    model_dict = dict()
    model_dict['linear'] = get_linear()
    model_dict['riged'] = get_riged()
    model_dict['lars'] = get_lars()
    model_dict['bayesian'] = get_bayesian()
    model_dict['svr_rbf'] = get_svm()
    model_dict['svr_linear'] = get_svm(kernel='linear')
    model_dict['svr_poly'] = get_svm(kernel='poly')
    model_dict['gaussian'] = get_gaussian()
    model_dict['kneighbors'] = get_kneighbors(n_neighbors=10)
    model_dict['random_forest'] = get_random_forest(max_depth=10, n_estimators=1000)
    model_dict['gradient_trees'] = get_gradient_trees(max_depth=10, n_estimators=1000)
    model_dict['neural_network'] = get_neural_network(hidden_layer_sizes=(50, 25, 10))
    return model_dict
```

```
# test>0.5: o
# test>0.1: +
# train>0.5 and test<0.1: *
# train<0.5 : -
```

	age	race_identity	face_recognition	day_dream	IRI_FS	IRI_EC	IRI_PT	IRI_PD	Interdependent	Independent	altruism	anxiety	depression	test_Y
pain_eeg_norm	-**--*-**---	-**----*----	-**--*-**---	-**----*----	-**----*----	-**----*----	-**--*-**---	-**--*-**---	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**--*-*----
pain_eeg_norm_Pain_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
pain_eeg_norm_Neutral_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
pain_eeg_orig	------------	------------	-----*------	------------	------------	------------	------------	------------	------------	------------	------------	------------	-----*------	------------
pain_eeg_orig_Pain_	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
pain_eeg_orig_Neutral_	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
pain_erp_amplitude_norm	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
pain_erp_amplitude_norm_Pain_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**---------	-**----*----	-**----*----	-**----*----	-**----*----
pain_erp_amplitude_norm_Neutral_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**---------	-**----*----	-**----*----	-**----*----	-**---------	-**----*----	-**----*----
pain_erp_amplitude_orig	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------
pain_erp_amplitude_orig_Pain_	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------
pain_erp_amplitude_orig_Neutral_	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------	--*---------
pain_erp_peak	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
rest_norm	-**----*----	-**----*----	-**---------	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
rest_orig	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
test_X	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	o*++-o-+o---

['bayesian', 'gaussian', 'gradient_trees', 'kneighbors', 'lars', 'linear', 'neural_network', 'random_forest', 'riged', 'svr_linear', 'svr_poly', 'svr_rbf']

Result of model sets

- Model set 2 (More penalty)

```
def model_set_2():
    model_dict = dict()
    model_dict['linear'] = get_linear()
    model_dict['riged'] = get_riged(alpha=0.1)
    model_dict['lars'] = get_lars(alpha=0.5)
    model_dict['bayesian'] = get_bayesian()
    model_dict['svr_rbf'] = get_svm(C=10.0, gamma='scale')
    model_dict['svr_linear'] = get_svm(kernel='linear',C=10.0, gamma='scale')
    model_dict['svr_poly'] = get_svm(kernel='poly',C=10.0, gamma='scale')
    model_dict['gaussian'] = get_gaussian(alpha=0.001)
    model_dict['kneighbors'] = get_kneighbors(n_neighbors=5)
    model_dict['random_forest'] = get_random_forest(max_depth=50, n_estimators=5000)
    model_dict['gradient_trees'] = get_gradient_trees(max_depth=5, n_estimators=1000)
    model_dict['neural_network'] = get_neural_network(hidden_layer_sizes=(100,50,))
    return model_dict
```

	age	race_identity	face_recognition	day_dream	IRI_FS	IRI_EC	IRI_PT	IRI_PD	Interdependent	Independent	altruism	anxiety	depression	test_Y
pain_eeg_norm	-**--*-**---	-**----*----	-**--*-**---	-**----*----	-**----*----	-**----*----	-**--*-**---	-**--*-**---	-**----*----	-**----*----	-**----*----	-**---**----	-**----*----	-**--****---
pain_eeg_norm_Pain_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**---**----
pain_eeg_norm_Neutral_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**---**----
pain_eeg_orig	------------	------------	-----*------	------------	------------	------------	------------	------------	------------	-----*------	------------	------------	-----*------	-----*------
pain_eeg_orig_Pain_	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
pain_eeg_orig_Neutral_	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
pain_erp_amplitude_norm	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
pain_erp_amplitude_norm_Pain_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
pain_erp_amplitude_norm_Neutral_	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----
pain_erp_amplitude_orig	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----
pain_erp_amplitude_orig_Pain_	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----
pain_erp_amplitude_orig_Neutral_	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----
pain_erp_peak	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----
rest_norm	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----	--*----*----
rest_orig	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------
test_X	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	-**----*----	o*o+-o-+o---

['bayesian', 'gaussian', 'gradient_trees', 'kneighbors', 'lars', 'linear', 'neural_network', 'random_forest', 'riged', 'svr_linear', 'svr_poly', 'svr_rbf']

Result of model sets

- Model set 3 (More complicate, still running after 20 hours)

```
def model_set_3():
    model_dict = dict()
    model_dict['linear'] = get_linear()
    model_dict['riged'] = get_riged(alpha=0.1)
    model_dict['gaussian'] = get_gaussian(alpha=0.01)
    model_dict['kneighbors'] = get_kneighbors(n_neighbors=20)
    model_dict['random_forest'] = get_random_forest(max_depth=None, n_estimators=10000)
    model_dict['gradient_trees'] = get_gradient_trees(max_depth=200, n_estimators=10000)
    return model_dict
```

For future optimization

- Data : maybe remove outliers
- Models :
  - Squeeze down feature set size
  - Regression -> classification
  - Hyper-parameters searching