

**Description:** This week you all will be working in pairs to build a model for the kaggle house price competition. Each pair will consist of an experienced programmer along with a less experienced programmer who will work through the entire ml development pipeline. At next Sunday's meeting, you all will present your models and your results.

Dataset link: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

Download train.csv only

Requirements:

- build at least one ANN with pytorch
  - you may use sklearn or other libraries to build generic ml models as well to see how it fares, but the ANN is the bare minimum
- practice using software style guidelines
- use mse loss to train and evaluate your models
- Make slideshow presenting model pipeline and results
  - talk about preprocessing steps
  - data visualization
  - feature extraction
  - model architecture

Results Evaluation:

- To ensure people don't luck out, folder with train and test csv files will be provided in the github in a folder called houseprice\_data
- Two factors will be considered in deciding the winner
  - the test set rmse
  - the number of features used for the model
- The score formula  
 $\log(10 * \text{features used} / 79) * (\text{test set rmse loss})$
- lower score is better

**BCI Competition:** For those who are more advanced, you will be working with the BCI competition 4 2a dataset that I used for the NN\_sample\_pipeline tutorial notebook.

<https://www.kaggle.com/datasets/aymanmostafa11/eeg-motor-imagery-bciciv-2a>

Requirements:

- Build a CNN or RNN with pytorch
- practice using software style guidelines
- use cross entropy loss or binary cross entropy to train and evaluate models
- Do two class classification (left hand and right hand)
- Make slideshow presenting model pipeline and results
  - talk about preprocessing steps
  - data visualization
  - feature extraction
  - model architecture

Results Evaluation:

- To ensure people don't luck out, two folders with train and test npy files called MI\_train and MI\_test will be provided in the github
- Two factors will be considered in deciding the winner
  - the test set accuracy
  - the number of channels used for the model
- The score formula  
 $\log(220 / \text{channels used}) * (\text{test set accuracy})$
- higher score is better