- **Apply pre-processing to the data**

Standardization is often a good starting point
https://scikit-learn.org/stable/modules/preprocessing.html

- **Machine learning algorithms to evaluate**

1) Ridge regression: https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification
2) Kernel ridge regression: https://scikit-learn.org/stable/modules/kernel_ridge.html
3) Adaboost: https://scikit-learn.org/stable/modules/ensemble.html#adaboost
4) Random Forest: https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees
5) Neural network: https://scikit-learn.org/stable/modules/neural_networks_supervised.html#regression
6) Support vector machine: https://scikit-learn.org/stable/modules/svm.html#regression

Reference: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

- Use crossvalidation (either leave-one-out or 10-fold)
  It is described here:
  https://scikit-learn.org/stable/modules/cross_validation.html

- **For each machine learning algorithms, perform an optimization of the parameters**

Example for Kernel ridge regression using gridsearch to optimize the parameters alpha and gamma:

```
kr = GridSearchCV(KernelRidge(kernel='rbf', gamma=0.1),
                  param_grid={"alpha": [1e0, 0.1, 1e-2, 1e-3],
                              "gamma": np.logspace(-2, 2, 5)})
```

reference:
https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_kernel_ridge_regression.html#sphx-glr-auto-examples-miscellaneous-plot-kernel-ridge-regression-py

- **Perform the evaluation using a metric for regression**

MSE is a very common; you could report others
https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics

Questions:

1. **Do we need to transfer the X and Y in the project?** That is, if we set X as Total Area (cm²) and Y as MVPA_minutes.week, should we set Y as Total Area (cm²) and X as MVPA_minutes.week?

Yes, the best way is to create a csv file with the X,Y data that you need and load it in python

Below is one way:

```
import numpy as np
import pandas as pd

data_train = pd.read_csv('../input/train.csv')
data_test = pd.read_csv('../input/test.csv')
```

2. We discovered that **some data are not complete**, aka some subjects have less than 12 trials (especially subject 122 has only four data). We double-checked and found that the original dataset is incomplete. So we want to know **whether this will affect our project**? Personally thinking, it does not matter that much, but we still want to make sure.

That's okay, you can use it. Or you can exclude subjects that don't have enough data