Throughout the Final Project assignments, you will be working in-depth with a modified version of a real-life dataset. The modified dataset describes a control problem, namely the task of controlling a robot. There are 40 features (continuous) that describe the status of the robot, and the target value (continuous) describes the appropriate action corresponding to the feature. The training dataset which we will provide you has 8250 samples.

The final project will be comprised of three milestones and a final project report. In Milestone 1 (due 11/18), you will do basic data exploration and compare a few baseline models. In Milestone 2 (due 12/2), you will work on data preprocessing to improve model performance. In Milestone 3 (due 12/9), you will choose a specific model and tune its hyperparameters to successfully train your own machine learning model and compete with other students in a classroom Kaggle competition. Each milestone will have a corresponding report you must submit, and these reports comprise your final project report.

In Milestone 1 of the final project, you will first try to get some insight into the dataset and run a few baseline models and compare their performances.

Data Exploration (30 pts)

Let us first look at the **target** column, which is the target value that we wish to predict.

- Compute a few statistics for the target (mean, standard deviation, range, etc.).
- Make a kernel density estimate of the distribution of the target values and interpret the distribution.

Now, we will need to look at the features and how they relate to the target.

- First, similarly to above, compute a few statistics and make a kernel density estimate of the distribution for each feature.
- Compute the correlation between each feature and the target and find the 3 most correlated features.
- Make a scatter plot for each of the 3 features found above between the feature and the target. Interpret the relationship between the two.

Finally, we take a brief look at the relationships between features.

- Make a correlation matrix for the features. Interpret the matrix to see which features are highly correlated to each other. Since there are a lot of features, you don't have to identify all of them.

- Select 1 feature and take the 3 fe[illegible] are most correlated to your chosen feature. Plot a 4x4 matrix of scatter plots where each off-diagonal scatter plot is between the corresponding two distinct features and the diagonals are kernel density estimates for each corresponding feature. Interpret this matrix of scatter plots.

Baseline Models (10 pts)

In this part, you will choose 3 of the following models to fit on the data without much preprocessing. Note that the goal of this part is not to find the final model. Instead, it is to try a few models and to see how differently they perform so that we may gain some insight into the data. So do **not** invest too much time in trying to fine-tune your models. Try to choose which model to fit based on your findings from the data exploration part above!

- Linear Regression (or Kernelized Regression)
- k-Nearest Neighbors
- Gaussian Process
- Neural Network
- Random Forest Regressor
- Adaptive or Gradient Boosting Regressor

For each model, do the following:

- Use k-fold cross-validation to compute the in-sample and out-of-sample errors. Choose a suitable loss function for computing the errors and justify your loss function (same across models).
- Make a kernel density estimate of the z-scores of the error of each data point, for each model. Interpret this distribution.
- Perform a paired t-test comparing the in-sample errors of your 3 models. What are the p-values? Interpret the results.
- Perform a paired t-test comparing the out-of-sample errors of your 3 models. What are the p-values? Interpret the results.
- Interpret the performances of your models focusing on what they say about our data.

Submission

You will submit your code via your team GitHub repository and your report via Gradescope. Be sure to write your team's name on the written report you submit on Gradescope, and be sure to submit as one group!

Your Github team repository creation link: https://classroom.github.com/a/TVVLt7bf

All you have to do for the Github repo to be graded is to push your code before the deadline so that we can grade it according to completion. Just make sure to create 1 repo for your team. We'll grade based on your last version before the deadline (including the 3-day extension). Ideally, we recommend using a Jupyter notebook for the code, but using just Python is okay. Using other programming languages is also permitted, but not recommended.

For the report, we are looking for a typeset document to be submitted to Gradescope. The corresponding assignments will be available on Gradescope soon, where you'll submit the reports as if you were submitting one of the written homework. However, unlike the written homework, we will **not** accept handwritten documents. Ideally, we recommend using a LaTeX typeset pdf document, but using any other word processor is okay.