CSE 416: Introduction to Machine Learning
Assignment #2
Instructor: Pemi Nguyen
due: Friday, April 15, 2022, 11:59 pm.

Instructions:

- **Answers:** Please provide detailed yet concise explanations for your work.
- **Turn-in:** Do not write your name on your pages (your Gradescope account will identify you to us) and do not include a copy of the exercise's question in what you turn in. You must use Gradescope to upload your written homework solutions. You will submit a single PDF file containing your solutions to all the exercises in the homework. You must follow the Gradescope prompts that have you link exercise numbers to your pages. **We do not accept handwritten solutions**, so please typeset them, such as Microsoft Word or LaTeX editor.

1. Please determine whether the following practices are good or not. Please explain why in either case.

   (a) Wuwei's linear regression model on a housing dataset has 1000 features. She decides to use Ridge Regression (L2-regularization) in order to better obtain more prominent features and interpret how they determine a house's price.

   (b) Tanmay is aware of how time-consuming it is to find the globally optimal subset of features for a prediction task, so he resorts to the greedy algorithm, which he considers a way to significantly reduce the overall runtime, even though the solution might not be optimal.

   (c) After training various models that apply L1-regularization, Sahil notices that the training error is almost 0, but the validation error is greater than 10000. He thinks the best way to fix this is to decrease the regularization parameter.

   (d) Neuroscientist Rahul wants to determine the impact of water consumption on brain activity. He collects the average amount of water intake over a period of time from participants in a research study. However, the recorded values for that feature are in different units, such as liter / day, gallon / year, etc. (thus that feature has a wide range of numerical values). Rahul decides to normalize these raw recorded values as they originally are (without any further transformations) so that they can be on the same scale.

   (e) Sasha wants to find a good regularization parameter λ for a regression task using the regularized loss function $L'(w) = MSE(w) + \lambda \|w\|_1$. For each model, he tries a different λ and implements gradient descent on the training set to find a set of optimized weights $\widehat{w}$. Then he also uses the same loss function L' to calculate the validation error. He determines that the best model to perform on unseen data is the one that yields the smallest validation error.

2. The code below contains two errors regarding normalization of data. Identify one of them, explain why it's not correct, and propose a solution to fix it. For extra credit, identify both.

   Note: None of the errors are related to coding. Assume each line of code is correctly written, and the programmer has a conceptual misunderstanding of normalization.

```python
import numpy as np
from sklearn import linear_model
from some_library import load_dataset

# Load a full dataset from a file with n datapoints and separate into inputs X and outputs y
X, y = load_dataset('house_datasets.csv')

# Get the number of datapoints in the full dataset
n = len(X)

# Split the full dataset into train (80%) and test (20%)
indices = np.arange(n)
np.random.shuffle(indices)
train_length = int(n * 0.8)
test_length = n - train_length
X_train, y_train = X[indices[:train_length]], y[indices[:train_length]]
X_test, y_test = X[indices[train_length:]], y[indices[train_length:]]

# ------ THE ERRORS ARE IN THIS PART ------ #
# Get the mean and standard deviation of the full dataset
mean_X = np.mean(X, axis=0)
std_X = np.std(X, axis=0) # Assume there is no zero standard deviation
mean_y = np.mean(y)
std_y = np.std(y)

X_train = (X_train - mean_X) / std_X
X_test = (X_test - mean_X) / std_X
y_train = (y_train - mean_y) / std_y
y_test = (y_test - mean_y) / std_y

model = linear_model.LinearRegression()
model.fit(X_train, y_train)

pred_train = model.predict(X_train)
pred_test = model.predict(X_test)

print("The predicted prices of the houses from the train set are:", pred_train)
print("The predicted prices of the houses from the test set are:", pred_test)
```

3. Based on our experiments in the programming assignment, which of these models (LinearRegression, Ridge, Lasso) would we expect to have the lowest error on future error? Explain.