# Linear Regression Analysis

A. Nagar

Due Date Mentioned in eLearning

## Instructions

- This assignment requires you to build linear regression model in Python using standard machine learning libraries.
- You should store your dataset in a public location, such as Github or AWS. Do not submit the dataset (which could be quite large) on eLearning,
- You are allowed to work in teams of maximum two students. Please write the names and NetIDs of each group member on the cover page.
  Only 1 final submission per team.
- **You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After four days have been used up, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions on Piazza, not via email.

# 1 Project and Dataset Selection

For this assignment, you will need to choose a dataset from the UCI ML repository, which is located at: https://archive.ics.uci.edu/ml/datasets.php

You will need to set the **Default Task** to **Regression**, and after that you are free to choose any dataset of your choice. You will need to understand the dataset by reading its description and making sure you know which is the predicted variable and which are the predictors i.e. independent variables.

# 2 Regression Model Building

In this section, you will perform data pre-processing, loading, model creation and results analysis. You will need to create two different models for regression:

1. Stochastic Gradient Descent using SGDRegressor library of Scikit-learn.
2. Ordinary Linear Regression using the statsmodels library.

## 2.1 Pre-Processing

The following are the required pre-processing steps. Of course, you can add more as per your requirements.

- Loading the data into Pandas DataFrame object. Remember to use public URLs to read the file.
- Examining data for consistency: Check for null values, missing data, and any data inconsistency and handle them before proceeding forward.
- Examining attributes and target variable(s): Be sure you clearly understand each of the attributes and the target variable. Examine the various attributes and convert any categorical ones to numerical ones, if needed. Obtain and output summary of the attributes. Are the attributes normally distributed? If not, what could be the reason?
- Standardize and normalize the attributes.
- Find how the attributes are correlated to each other and the target variable. Perform numerical and visual analysis and output plots and results.
- Identify a few important attributes and proceed forward. Do not use all attributes blindly.
- Split the data into training and testing parts. The ratio is up to you.

### 2.2 Model Construction

As stated earlier, you need to create two types of models using **SGDRegressor** library of Scikit-Learn and **OLS** library of statsmodels.

For the SGDRegressor library, you are required to tune the various hyper-parameters such as learning rate, maximum iterations, loss, penalty, etc. Make sure to keep track of the hyper-parameters used and results obtained. Do not just use all default values. You would need to figure out which combination of hyper-parameters works best for your dataset. Be sure to output as many result metrics as possible such as training and test error and accuracy, R-squared statistic, etc. It is highly recommended that you standardize the data before model creation using SGDRegressor. You can use the StandardScaler method available in scikit-learn.

For the OLS library of statsmodels, you will need to output the model summary and **interpret and explain** all of the output diagnostics, such as coef, standard error, t-value, p-value, R-squared, R-squared adjusted, F-statistic, etc.

### 2.3 Result Analysis

You should create a report that includes results and your interpretation for each of the above models. You are free to add any additional detail. Tabular results and visual plots are preferred in all cases followed by your interpretation. Remember not to copy definitions or long explanations from external sources, but to write *your* analysis and interpretation of the results. **Please do not include code or code snippets in your report. Instead, submit them as a separate file.**

## 3 Requirements

The following are requirements that **cannot be changed**

1. You are allowed to work in teams of maximum size 2
2. Treat this as a data science project. You have to interpret the output diagnostics. Also, try to include as many plots as you can. As stated previously, your interpretation and analysis of results is what we want to see.
3. You cannot copy any publicly available solutions. There will be penalty for plagiarism.
4. Submit your Python code file and report file. Please do not hard code any local paths in your code. You can put the data in a public location, such as Github, and read from that link.
5. Python code can be on Google Colab or Jupyter Notebook.

You need to tune as many of the hyper-parameters as possible. The list will not be mentioned here, but you can see them in the documentation and sample code. You have to keep a log of your experiments with the hyper-parameters used and results obtained.

If you have made any assumptions, please state them completely. Also include instructions on how to compile and run your code in a README file.