

# C S 519 Applied Machine Learning

## Compare regression methods

### 1. Objective

In this **individual** homework, you are required to understand and compare several regression algorithms.

### 2. Requirements

#### 2.1 Tasks

1. [45 points] Write code to conduct regression by
  - a. [35 points] utilizing several linear regression functions provided by Python scikit-learn library: (i) LinearRegression, (ii) RANSACRegressor, (iii) Ridge, (iv) Lasso, and (v) ElasticNet.
  - b. [10 points] using one approach to conduct non-linear regression.
2. [25 points] Each regressor needs to be tested using the California housing dataset, which can be loaded using `fetch_california_housing` from `sklearn.datasets`. You need to use all the columns and all the instances in this dataset. The lecture notes may just use one column and a subset of the dataset for demonstration purpose. If you decide to use fewer columns/instances of the dataset, you need to show an analysis of why you chose to use only a subset of features/instances. For example, you can provide a correlation analysis to justify why you are not using all columns or show a random sample of the instances. If you do not justify why you are using fewer columns/instances, points will be deducted.
3. [25 points] Compare the performance of the different regressors' behavior by applying the knowledge that discussed in the lecture notes. Such analysis should include at least Mean squared error (MSE), R2 score, and the fitting (or training) time. You may want to change different parameters (e.g., regularization strength) to test the effect of different parameters to the model performance. Put the analysis to **report.pdf** file.
4. [5 points] Write a readme file **readme.txt** with detailed instructions on how to run your program.

#### 2.2 Other requirements

- Your Python code should be written for Python version 3.10 or higher.
- Please write proper comments in your code to help the instructor and teaching assistants to understand it.
- Please properly organize your Python code (e.g., create proper classes, modules). You can put your code to Jupyter Notebook or a .py file.

### 3. Submission instructions

Put all your files (Python code, readme file, report, datasets, etc.) to a zip file named **hw5\_<YourName>.zip** and upload it to Canvas.

#### **4. Grading criteria**

- **ZERO point** will be given if your code does not work. Please do not submit code that you did not test and make sure it works.
- The score allocation has been put beside the questions.
- **FIVE** points will be deducted if files are not submitted in the required format.
- If the total points are more than 100. Your grades will be scaled to the range of [0,100].
- Please make sure that you test your code thoroughly by considering all possible test cases.