Lab 2 - Model Selection

Lab 2 on Model Selection for DS3010 - Machine Learning

OVERVIEW & PURPOSE

In this lab, you will conduct model selection and hyper-parameter tuning using multi-fold cross validation on Ridge regression and Lasso.

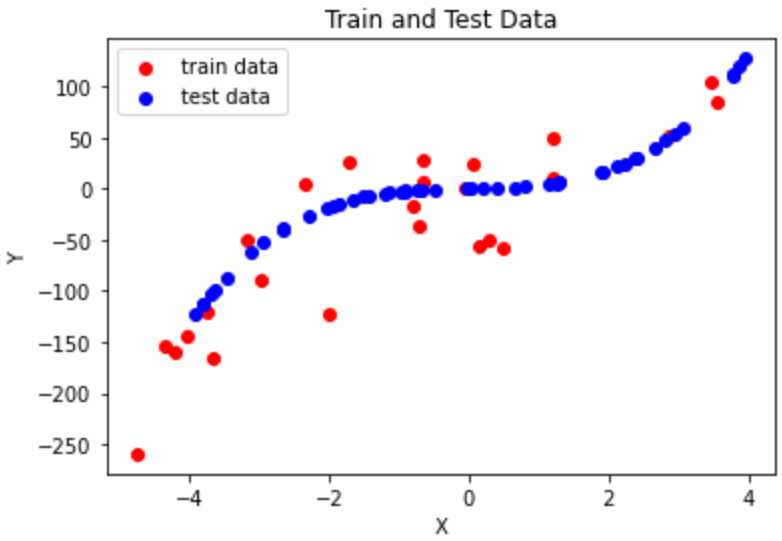
Instructions

1. Please submit the assignment through Moodle in .ipynb format (python notebook)
2. The submission should contain a single notebook containing all the solutions, including the requested documentation, observations, and findings.
3. The naming convention for the notebook is <firstname>_<lastname>_<rollnumber>.ipynb
4. You must adequately comment on the code to improve its readability.
5. The lab is worth 5 points
6. This graded lab is due on September 15th 5.00pm

Lab

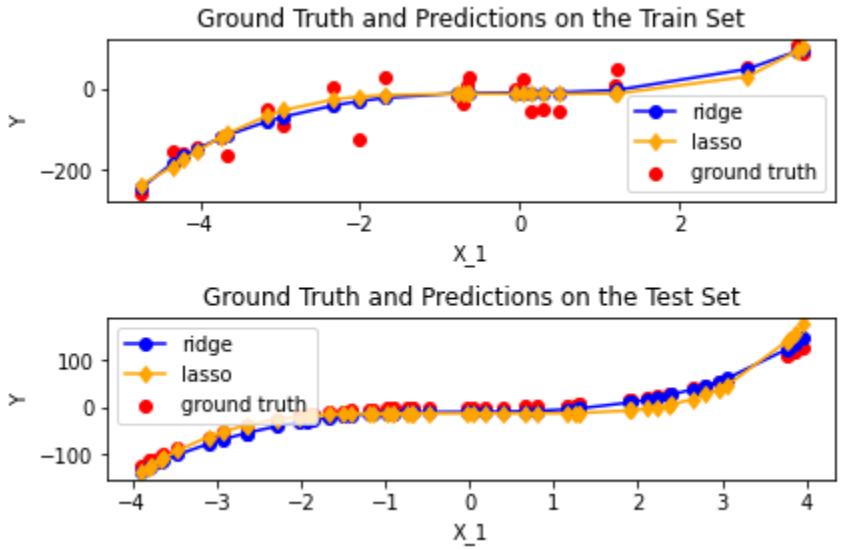
1. **Synthetic Data Generation**

   Execute the first code block to generate and load a synthetic polynomial dataset. The figure below illustrates the generated data. The output is governed by a polynomial y= w_0+w_1x+w_2x^2+...+w_7x^7. We generated x randomly from a uniform distribution. We have also defined the w vector to obtain the outputs. The dependent variable of the training dataset is slightly corrupted by adding some noise sampled from a normal distribution. You can experiment with this code to understand the dataset, but this is not the lab's focus.



2. **Ridge Regression with Cross Validation (1 point)**
    a. Read through the documentation of the RidgeCV class in the SKLearn library and describe the arguments - *alphas, cv, store_cv_values, cv_values_, coef_,* and *alpha_*.
    b. Define a range of values for alpha - 1e-2 to 100 in multiples of 10.
    c. Define a variable containing the number of cross validation folds.
    d. Create an instance of the RidgeCV class with the above defined parameters.
    e. Fit the model to the training set.
    f. Print the best estimate for alpha
    g. Print the coefficients of the regression model.
    h. Compute the predictions for the train and test data and save it in variables.
3. **LASSO Regression with Cross Validation (1 point)**
    a. Read through the documentation of the LassoCV class in the SKLearn library and describe the arguments - *alphas, cv, coef_,* and *alpha_*.
    b. Define a range of values for alpha - 1e-2 to 100 in multiples of 10.
    c. Define a variable containing the number of cross validation folds.
    d. Create an instance of the LassoCV class with the above defined parameters.
    e. Fit the model to the training set.
    f. Print the best estimate for alpha
    g. Print the coefficients of the regression model.
    h. Compute the predictions for the train and test data and save it in variables.
4. **Visualizing the Output (0.5 point)**
    a. Use the predictions on the train and test to visualize the models' performance.
    b. Use the subplot function to add two figures to the same plot as illustrated

below



5. **Comment on the two models' performance (0.5 point)**
    a. Add a text block discussing your observations on the models' performance.
6. **Load the diabetes dataset that is part of the SKLearn library. (0.5 point)**
    a. Add a text block to describe the dataset (number of data points, list of attributes. Identify the categorical and continuous attributes.
7. **Grid Search Cross Validation for Hyper-Parameter Tuning (1.5 points)**
    a. Read the documentation for SKlearn model selection class GridSearchCV and describe the following parameters and attributes- *estimator, param_grid, refit, cv, scoring, cv_results_, best_Estimator_,* and *best_params_*
    b. Define a param_grid dictionary with the list of permissible values for the hyper-parameter alpha.
    c. Create an instance of the GridSearchCV class using the negative mean squared error as the scoring function, with default number cross validation folds, and the ability to check the intermediate outputs (verbose parameter).
    d. Call the fit function using the training data.
    e. Print the best parameters for each of the models.

Ungraded Part: Advanced Lab

- Process the output of the grid search to compute the average score for every parameter. Plot the average score as a function of hyper-parameter value and discuss any observable trend.