Lab 1 - Regression

Lab 1 on Regression for DS3010 - Machine Learning

OVERVIEW & PURPOSE

In this lab, you will experiment with linear regression and its variants (Ridge 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 3rd 5.00pm

Lab

1. **Loading the data and computing the basic statistics (0.5 point)**
    a. Load the training and test data
    b. Compute the basic statistics of the training data. The output should look something like this.

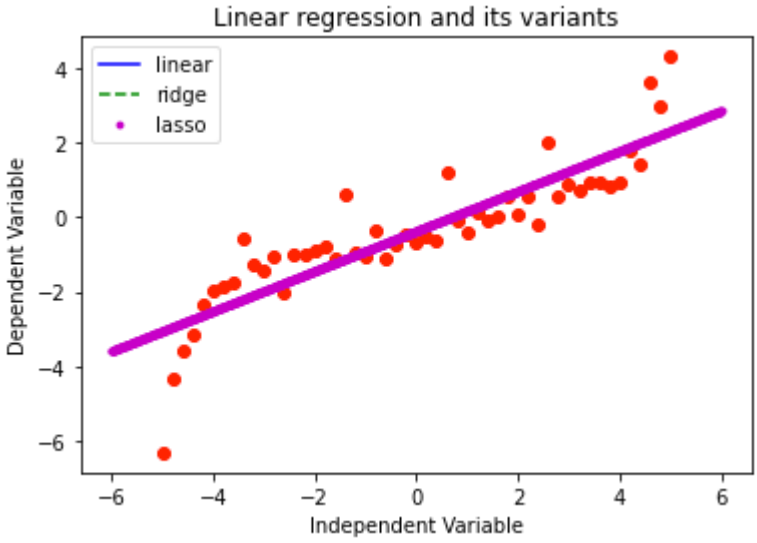
```
The number of training data points is:  50
The number of independent variables is:  1
Independent variable summary statistics:
mean:  -1.7415263131375004e-17
minimum:  -5.0
maximum:  5.0
Dependent variable summary statistics:
mean:  -0.3703166274509804
minimum:  -6.3204
maximum:  4.3024
```

c. Load the submission.csv file. You will be appending the predictions generated by the different models to this file.

2. **Linear Regression (1 point)**
    a. Read through the documentation of the linear regression models in the SKLearn library and describe the arguments - *fit_intercept* and *normalize*.
    b. Create an instance of the linear regression model with *normalize* parameter set to true.
    c. Fit the model to the training data.
    d. Store the model predictions for the training set.
    e. Store the model predictions for the test set and append it to the submission file. Name the column as “Y_test_linear”
    f. Compute the mean squarred error of the model on the training set.
3. **Ridge Regression (1 point)**
    a. Read through the documentation of ridge regression in the SKLearn library and describe the argument *alpha*.
    b. Define a variable alpha that will store its value. Pass it as an input when creating a ridge regressor instance.
    c. Fit the ridge model to the training set.
    d. Compute the mean squarred error of the model on the training set.
    e. Vary alpha between 1e-5 to 1 in multiples of 10. For each alpha, note the MSE of the model on the training set.
    f. Select the best alpha value to train the final model.
    g. Store the model predictions for the training set.
    h. Store the model predictions for the test set and append it to the submission file. Name the column as “Y_test_ridge”
4. **Lasso Regression (1 point)**
    a. Read through the documentation of LASSO regression in the SKLearn library and describe the argument *alpha*.
    b. Define a variable alpha that will store its value. Pass it as an input when creating a ridge regressor instance.
    c. Fit the ridge model to the training set.
    d. Compute the mean squarred error of the model on the training set.
    e. Vary alpha between 1e-5 to 1 in multiples of 10. For each alpha, note the MSE of the model on the training set.
    f. Select the best alpha value to train the final model.
    g. Store the model predictions for the training set.
    h. Store the model predictions for the test set and append it to the submission file. Name the column as “Y_test_lasso”

5. **Plot the data and the regression (1 point)**
    a. Use a scatter function to plot the training data.
    b. Plot the outputs for linear regression and its variants. Use the following convention.
        i. Red colored points for training data
        ii. Blue colored line for linear regression
        iii. Green colored dashed line for Ridge regression
        iv. Magenta colored dotted line for LASSO regression.
    c. All the plots must be on a single figure.
    d. Add labels to x and y axis
    e. Add a title to the figure
    f. Add a legend to the figure.

    The final output should look something like this.



6. Saving the output (0.5 point)

    a. Save the final submission structure to the submission.csv file