Assignment 3 Report

Names: Kevin Spike & Andrew Gabler

Work Done for Regression

Load The Data

For the regression we chose to work on the machine dataset. This can be found in the *regression/csvreader.py* file. This assumes that the last column in the CSV is the target, or rather, ERP. This file mostly functions as a library file, per the instructions, however, you can choose to run it independently and it will ask to be supplied the name of a file to read. Please note, it is designed to read a file that is formatted like the file *regression/machine.data*; in order for the file to be read properly.

Meet The Data

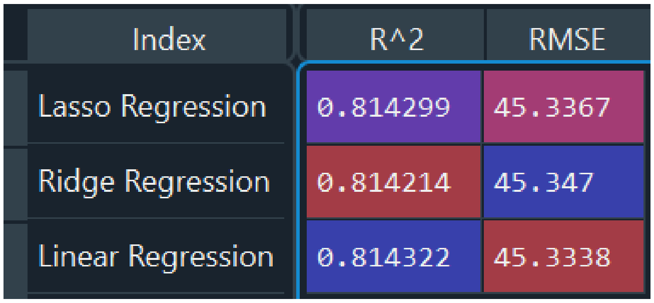
This task is accomplished by the *regression/meet_the_data.py* file. This does a series of print statements on the required data The correlation does not include the manufacturer name nor model name since these are not integers and we treat these as data indices as opposed to actual data since they are all unique.

Parameter Tuning

This task is accomplished by *regression/parameter_tuning.py* file. This file does a simple grid search on both lasso and ridge models to find the best parameters. Just a note, the instructions say to include 0 as an option but this does not since the models do not handle a 0 for alpha well. This file serves as both a library function and a callable. This is so that in other files, we can use this file to find the best parameters.
These were the best alpha values that we found.
Ridge Regression Alpha:100.0
Lasso Regression: 0.1

Model Metrics

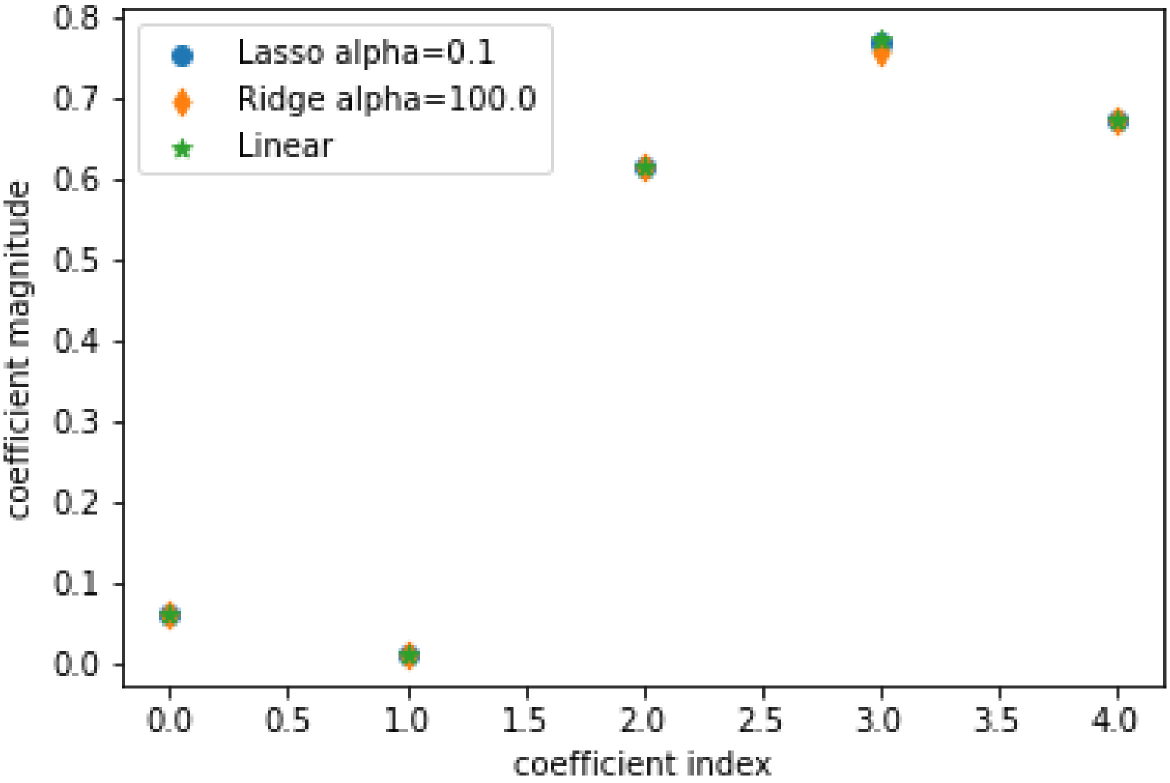
This task is accomplished by *regression/scoring.py* file. This uses the “score” function for R^2 and uses the square root of the “mean_squared_error” function as the RMSE. This is then thrown into a dataframe to get the results.



Index	R^2	RMSE
Lasso Regression	0.814299	45.3367
Ridge Regression	0.814214	45.347
Linear Regression	0.814322	45.3338

Parameter Values Graph

This task was accomplished by the *regression/parameter_values.py* file.This file creates the plot of the coefficients and their magnitude. Note, a lot of the points on the graph are hidden due to the parameters being so close together in all three models.



Work Done for Classification

Load The Data

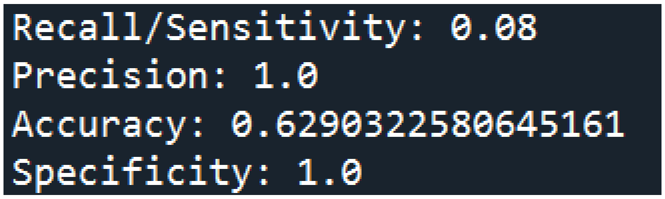
For the classification we chose to work on the Haberman dataset. This can be found in the *classification/csvreader.py* file. This assumes that the last column in the CSV is the target, or rather, survival status. This file mostly functions as a library file, per the instructions, however, you can choose to run it independently and it will ask to be supplied the name of a file to read. Please note, it is designed to read a file that is formatted like the file *classification/haberman.data*; in order for the file to be read properly.

Meet The Data

This task is accomplished by the *classification/meet_the_data.py* file. This does a series of print statements on the required data

Logistic Regression and Result Analysis

Both the logistic regression model building and the gathering of metrics is performed by the file *classification/logistic_regression.py*. The file uses a confusion matrix to calculate the required metrics.

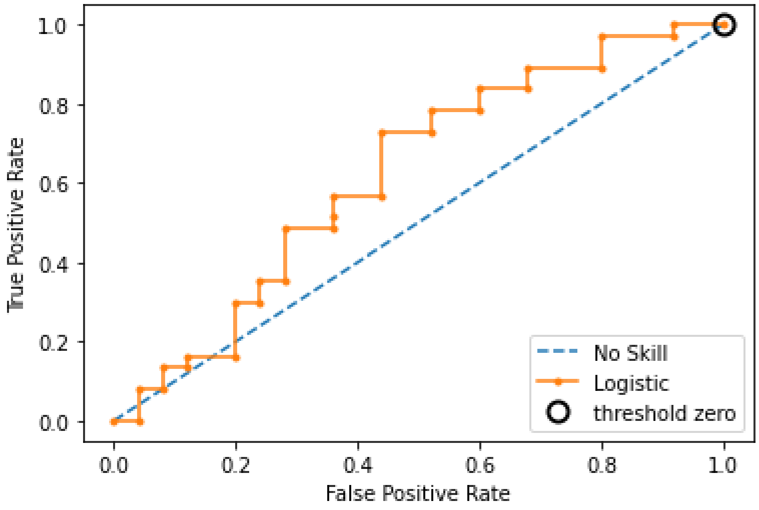


```
Recall/Sensitivity: 0.08
Precision: 1.0
Accuracy: 0.6290322580645161
Specificity: 1.0
```

We experimented with using L1 instead of L2 but we found that this made Recall and Accuracy go down so we kept L2.

ROC Curve

This task was accomplished by the *classification/roc_curve.py* file.



File Manifest

Regression

Please note, every file for regression is under the “regression” folder.

cvsreader.py

Responsible for loading the CSV as data we can use to develop models with.

machine.data

CSV data we are working with.

machine.names

Description of the data we are working with for our own benefit.

meet_the_data.py

Data at a glance.

paramter_tuning.py

Responsible for finding the best alpha values for lasso and Ridge Regression

parameter_values.py

Plots out the coefficient index vs coefficient magnitude graph from the model.

preprocess.py

Library file we built for our own purposes to preprocess the data so we get more accurate results.

scoring.py

Calculates and displays scores for Ridge, Lasso and Linear models.

Classification:

Please note, every file for regression is under the “classification” folder.

cvsreader.py

Responsible for loading the CSV as data we can use to develop models with.

haberman.data

CSV data we are working with.

haberman.names

Description of the data we are working with for our own benefit.

logistic_regression.py

Development of the logistic regression model and scoring of it.

meet_the_data.py
Data at a glance.

roc_curve.py
Builds a ROC curve for the model.