COMP 4983: Lab Exercise #6 Mark: /70

[Due: Oct 10, 2024 @0930
Assignment Submission
Folders]

Instructions:

In this lab, you will preprocess and analyze data from a fund-raising campaign for a charitable organization to

- determine the most-suitable model (among linear regression, ridge regression and the lasso) to predict the amount that a donor would donate
- determine the top three (3) features that exhibit the strongest effect for the lasso

Part 1: Data Preprocessing and Linear Regression

[20 marks] In this part of the lab, you will preprocess the fund-raising campaign data for a charitable organization. In addition, you will perform linear regression on the fund-raising campaign data to predict the amount that a potential donor would donate.

Steps:

1) Download the fund-raising campaign dataset and attribute file, *data_lab6.csv* and *DataAttributes_lab6.txt*, from BCIT Learning Hub (Content | Laboratory Material | Lab 6) and save it in your working directory. The dataset, *data_lab6.csv*, contains 8,928 rows (including a header row) and 377 columns. Each row contains 376 attributes of a potential donor, followed by the donation amount the donor donated under **TARGET_D**. The attributes included in this dataset are described in *DataAttributes_lab6.txt*.

2) Create a new Python script using the filename *loaddata_lab6.py*, to implement a function called **load()** as defined below which (i) loads data from a specified file path using **pandas.read_csv()** with the option **'low memory'** set to **'False'**, (ii) encodes any categorical attribute (column) using one-hot encoding scheme and (iii) returns the expanded data as a **DataFrame** object. Save *loaddata_lab6.py* in your working directory.

```
# load and encode categorical data
# Inputs:
#   filepath: path to the csv file
# Outputs:
#   data: a DataFrame object containing encoded data
def load (filepath):
  ???

  return data
```

Recall that one-hot encoding scheme transforms a categorical attribute with m possible values into m binary columns. Each sample in the categorical attribute is assigned 1 in one of m binary columns to indicate it as **True** and 0 in all other. You may assume that an attribute is considered categorical if (i) its data type (dtype) is 'object' or (ii) the number of unique values is less than 20. Ensure that the order of the attributes is preserved with the one-hot encoding scheme.

3) Create another new Python script using the filename *regression_lab6.py*, and include the following lines at the top of your script. Save *regression_lab6.py* in your working directory.

```
import loaddata_lab6 as loader
import pandas as pd
import numpy as np
```

4) Download the one-hot encoded fund-raising campaign dataset, *data_lab6_expanded.csv*, from BCIT Learning Hub (Content | Laboratory Material | Lab 6) and save it in your working directory. The dataset in *data_lab6_expanded.csv* contains 8,928 rows (including a header row) and 3159 columns (including **TARGET_D**). In *regression_lab6.py*, verify the correctness of your *loaddata_lab6.py* script implementation by comparing the output returned by your **loader.load()** with the expected output, *data_lab6_expanded.csv*, as shown below, which should return **True**.

```
data = loader.load(path to <data_lab6.csv>)
data_expected = pd.read_csv(path to <data_lab6_expanded.csv>)
print(np.allclose(data, data_expected))
```

5) If you are unable to correctly verify your data preprocessing implementation in the *loaddata_lab6.py* script in Step (4), you may proceed with the lab using *data_lab6_expanded.csv* as your preprocessed data, however, you will lose 20 marks (no partial marks) for Part 1. Proceed with implementing the following in *regression_lab6.py*:
   a) Load the preprocessed data using **loader.load()**.
   b) Split the data into training and test sets, with the first 30% of the data for training and the remaining for testing.
   c) Fit a linear regression model to the training set and evaluate the Mean Absolute Error (MAE) of the model on both the training set and test set. Output the MAE values. You may use **LinearRegression.fit()** and **LinearRegression.predict()** functions from **sklearn.linear_model**.

<u>Deliverable</u>:

All work submitted is subject to the standards of conduct as specified in BCIT Policy 5104. Late submissions will not be accepted.

Ensure that your source code is adequately commented and submit using the filename *loaddata_lab6.py* to BCIT Learning Hub (Laboratory Submission | Lab 6).

Part 2: Ridge Regression

[20 marks] In this part of the lab, you will (i) apply ridge regression for different values of the tuning parameter λ, (ii) select the best value of λ using 5-fold cross-validation on the training set and (iii) evaluate the MAE of the ridge regression model on the test set. Continue the implementation in *regression_lab6.py*.

Steps:

1) Load the preprocessed data using **loader.load()**.
2) Standardize all columns in the data, except for the last column, **TARGET_D**, to zero mean and unit variance (i.e., $\mu = 0$ and $\sigma = 1$). You may use the **StandardScaler.fit_transform()** function from **sklearn.preprocessing** to standardize the data.
3) After standardizing the data, split the data into training and test sets, with the first 30% of the data for training and the remaining for testing.
4) Apply ridge regression for each $\lambda = [10^{-3}, 10^{-2}, 10^{-1}, \ldots, 10^{10}]$ and evaluate the MAE on the training set using 5-fold cross-validation. Plot the average MAE of the training sets and the average MAE of the cross-validation sets from the five (5) trials as a function of λ. Include in your plot, a terse descriptive title, x-axis label, y-axis label and a legend. You may use **Ridge.fit()** and **Ridge.predict()** from **sklearn.linear_model**.
5) Determine the best value of λ from Step (4).
6) Using the best value of λ, evaluate the MAE on the test set. Output the MAE value.
7) Describe how the MAE value obtained using ridge regression compares to that from the linear regression from Part 1.

Part 3: The Lasso

[30 marks] In this part of the lab, you will (i) apply the lasso for different values of the tuning parameter λ, (ii) select the best value of λ using 5-fold cross-validation on the training set and (iii) evaluate the MAE of the lasso model on the test set. In addition, you will determine (iv) the most suitable model to predict the amount that a donor would donate and (v) the top three (3) features that exhibit the strongest effect for the lasso. Continue the implementation in *regression_lab6.py*.

Steps:

1) Apply the lasso for each $\lambda = [10^{-2}, 10^{-1.75}, 10^{-1.5}, \ldots, 10^{1.75}, 10^{2}]$ and evaluate the MAE on the training set using 5-fold cross-validation. Plot the average MAE of the training sets and the average MAE of the cross-validation sets from the five (5) trials as a function of λ. Include in your plot, a terse descriptive title, x-axis label, y-axis label and a legend. You may use **Lasso.fit()** and **Lasso.predict()** from **sklearn.linear_model**.
2) Determine the best value of λ from Step (1).
3) Using the best value of λ, evaluate the MAE on the test set. Output the MAE value.
4) Describe how the MAE value obtained using the lasso compares to that from the linear regression from Part 1 and that of the ridge regression from Part 2.
5) Determine the most suitable model (among linear regression, ridge regression and the lasso) to predict the amount that a donor would donate.
6) [10 marks] Determine the top (3) features that exhibit the strongest effect for the lasso.

Deliverable:

All work submitted is subject to the standards of conduct as specified in BCIT Policy 5104. Late submissions will not be accepted.

Ensure that your source code is adequately commented and submit using the filename *regression_lab6.py* to BCIT Learning Hub (Laboratory Submission | Lab 6).