Assignment 1: Regression Analysis

Submission Deadline: February 6, 2020 (9 PM)

Deliverables: Project Report in BMVC format (template can be downloaded from https://www.overleaf.com/project/5e301ad33c2f38000171a776, Ensure you insert your name and Roll Number at the top of your solution) and **code in Python/Matlab/C/C++**. **You are nevertheless strongly encouraged to code in Python!**

Make a zip file containing the code and report, and upload the zip file with your name as title on Google classroom.

Objective: .In this assignment, you will get familiar with the different variants of linear regression. You will probably need to install Anaconda with Python3, Pandas, Matplotlib and Scikitlearn to complete this assignment using Python code. The best practice would be to create a virtual conda environment and install these packages within the environment.

Dataset: You will need to download the **Boston house prices dataset** for this assignment. The dataset can be downloaded via the following commands.

```
>>> from sklearn.datasets import load_boston

>>> X, y = load_boston(return_X_y=True)

>>> print(X.shape)

(506, 13) #Dataset contains 506 data points and thirteen dimensions.
```

Alternatively, in Pandas do:

```
# load dataset

house_price = load_boston()

df = pd.DataFrame(house_price.data, columns=house_price.feature_names)

df['PRICE'] = house_price.target
```

Description:

The simplest/basic version of linear regression that we have done in class is Ordinary Least Squares (OLS) regression. Here, the cost/loss function is defined as:

- $J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(h_i(\theta)-y_i)^2$

which seeks to reduce the errors between the model predictions $h_i(\theta)$ for the training data and the actual values $y(i)$.

Task 1: Split the Boston dataset into two parts: *training* and *test* sets. While it is advisable to adopt a cross-validation procedure to synthesize the training and test sets, another *generally accepted* methodology is to adopt a 70:30 random split of the training and test sets.

On obtaining the training a test splits, derive an OLS regressor from the training data. **Plot the values of the regression coefficients for the different predictor variables using a bar graph.**

Task 2: However, the OLS model is prone to overfitting, implying that it could result in ***low bias*** and ***high variance***. To ensure that the regression model does not overfit, **ridge regression** involves determining the vector of regression coefficients$B=\{\beta_i\}$, whose components βi are constrained such that:

$$\beta_0^2 + \beta_1^2 + \cdots + \beta_p^2 \leq C^2$$

which is the equation of a hyper-sphere, The best fit regression coefficients are found as:

$$\hat{\beta}^{ridge} = \underset{\beta \in \mathbb{R}}{argmin} \|y - XB\|_2^2 + \lambda \|B\|_2^2$$

where λ is known as the ***regularization coefficient***, that is designed to synthesize a ‘smoother’ (see class notes relating to overfitting) line fit.

Higher the **λ**, higher the value of the penalty term, **λ**$\|B\|_2^2$. ***The ridge regression model generally makes better predictions than the OLS model.*** Indirectly, this ridge regression gives higher importance to more informative features, while not dropping unimportant features.

Plot the regression coefficient (β¿ estimates with ridge regression for predictors: *room, residential zone, highway access, crime rate* and *tax* as λ varies from 0-200.

Task 3: Another variant of the ridge regression formulation is the ***lasso regressor***, where ***lasso*** stands for ***Least absolute shrinkage and selection operator***. This method is similar to ridge regression except for the way in which the regularization term is modelled. Here, the penalty term involves the sum of absolute values of the features, i.e., mathematically,

$$\hat{\beta}^{\text{lasso}} = \underset{\beta\in\mathbb{R}}{argmin}\|y - XB\|_2^2 + \lambda\|B\|_1$$

This type of regularization (L1) can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output. So Lasso regression not only helps in reducing over-fitting but it can help us in feature selection.

Plot the regression coefficient (β¿ estimates with lasso regression for predictors: *room, residential zone, highway access, crime rate* and *tax* as λ varies from 0-200.

Task 4: For all the above models, plot the residuals obtained for the training data. You may choose any *three* values of λ to plot the residuals for ridge and lasso regression.

Task 5: Tabulate the mean training and test errors for each of the above models. Based on the results observed, summarize your conclusions and learnings from this exercise.