LAHORE UNIVERSITY OF MANAGEMENT SCIENCES
Syed Babar Ali School of Science and Engineering

EE514/CS535 Machine Learning
Spring Semester 2021

Programming Assignment 3 – Linear Regression

Issued: Sunday 28th February, 2021

Total Marks: 100

Submission: 11:55 pm, Monday 8th March, 2021.

Instructions

- Submit your code both as notebook file (.ipynb) and python script (.py) on LMS. The name of both files should be your roll number. Failing to submit any one of them will result in the reduction of marks.
- The code MUST be implemented independently. Any plagiarism or cheating of work from others or the internet will be immediately referred to the DC.
- 10% penalty per day for 3 days after due date. No submissions will be accepted after that.
- Use procedural programming style and comment your code properly.

Goal

The goal of this assignment is to get you familiar with Linear Regression and to give hands on experience of basic python tools and libraries which will be used in implementing the algorithm.

NOTE:

You are not allowed to use scikit-learn or any other machine learning toolkit for part 1 and 2. You have to implement your own Linear Regression model from scratch. You may use Pandas, NumPy, Matplotlib and other standard python libraries.

Part 1: Linear Regression for single variable from scratch (20 Marks)

Dataset:

Dataset for this part has already been split (80%, 20%) into training and test data which can be found here as well under the assignment tab on LMS.

Tasks:

1. Implement Linear Regression from scratch to predict the chance of admission of a student based on his/her GRE Score. You will need to implement the following functions:

   - *Predict* function which calculates the hypothesis for input sample given the values of weights.

     $$h(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x,$$

     where $\boldsymbol{\theta} \in \mathbb{R}^2$ is the weight vector given by $\boldsymbol{\theta} = [\theta_0, \theta_1]^T$.

   - *Mean Square Error* function which calculates the cost of using weights as parameters for linear regression. The formula to calculate Mean Square Error is given below:

     $$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^{n} (\hat{y}^i - y^i)^2,$$

     where y^i and $\hat{y}^i$ are the actual and predicted labels of the i-th training instance respectively and n is the total number of training samples.

   - *Batch Gradient Descent* function which learns the values of weights when given as parameter the learning rate α and the number of iterations called epoch. Experiment with different values to determine the best parameters.

     For $j = 0$ and $j = 1$ repeat until convergence {

     $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$

     }

   - Use 5-fold cross validation on the training set to determine the best value of α and number of epochs.

   - Plot these graphs for different α values:

     - No. of epochs (y-axis) vs training loss (x-axis).
     - No. of epochs (y-axis) vs validation loss (x-axis).

   - After analyzing the graphs, justify the best value of α and number of epochs in comments.

   - Plot the linear regression model with GRE Score on the x-axis and the chance of admission on the y-axis, and explain the correlation.

Part 2: Multivariate Linear Regression from scratch (50 Marks)

Dataset:

For this part, you will use the COVID-19 Dataset of South America which can be found here. There are 1500 examples in the training set, 173 examples in the validation set and 200 examples in the test set. Each instance has 15 input variables (features) and 1 output variable (new deaths). Description of each feature is given below:

- population = Population in 2020
- median_age = Median age of the population, UN projection for 2020
- gdp_per_capita = Gross domestic product at purchasing power parity
- human_development_index = Summary measure of average achievement in key dimensions of human development
- extreme_poverty = Share of the population living in extreme poverty
- cardiovasc_death_rate = Death rate from cardiovascular disease in 2017
- diabetes_prevalence = Diabetes prevalence (% of population aged 20 to 79) in 2017
- life_expectancy = Life expectancy at birth in 2019
- reproduction_rate = Real-time estimate of the effective reproduction rate (R) of COVID-19
- new_cases = New confirmed cases of COVID-19
- total_cases = Total confirmed cases of COVID-19
- total_tests = Total tests for COVID-19
- positive_rate = The share of COVID-19 tests that are positive, given as a rolling 7-day average
- stringency_index = composite measure based on 9 response indicators including school closures, workplace closures, and travel bans
- hospital_beds_per_thousand = Hospital beds per 1,000 people

Tasks:

Your tasks in this part are:

- You are required to select 10 best features by drawing scatter plots and using Pearson's correlation coefficient.
- *Data Normalization*: Normalize the dataset by subtracting the mean of each feature from feature value and then divide by the standard deviation of that feature.

$$x_{\text{norm}} = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

(For normalization of validation set and test set, use mean and standard deviation of training set.)

- Now implement *Predict* function, *Mean Square Error* function and *Batch Gradient Descent* function as explained in Part 1.
- Use the validation set provided to you to find the best value of α and number of epochs.
- Plot these graphs for different α values:
  - No. of epochs (y-axis) vs training loss (x-axis).
  - No. of epochs (y-axis) vs validation loss (x-axis).
- After analyzing the graphs, justify the best value of α and number of epochs in comments.

Part 3: Regularized Linear Regression (30 marks)

Regularization is a technique that assumes smaller weights generate simple models and helps avoid overfitting. In this part, you will be using various regularization techniques on the COVID-19 Dataset of South America (provided in Part 2).

Tasks:

Implement the least squares Linear Regression, Lasso Regression, Ridge Regression, and Elastic Net Regression using scikit-learn. You are required to:

- Try out different values of the regularization parameter (alpha in scikit-learn document) and use the validation set to determine the best value of the regularization parameter by computing validation loss using Mean Squared Error.
- For Ridge Regression and Elastic Net Regression, plot regularization coefficients on the x-axis and learned parameters θ on the y-axis. Please read this blog as reference.
- After evaluating the best value of the regularization parameter, use the Mean Squared Error to compute the loss on the test set for each regression.