

National University of Computer and Emerging Sciences



Lab Manual 07 CL461-Artificial Intelligence Lab

Course Instructor	Eesha Tur Razia Babar
Lab Instructor (s)	Mateen Fatima Hamza Mustafa
Section	BDS-6B
Semester	Spring 2024

Lab Task:

You have been given a Google Colab starter code, you must perform the following tasks:

Coding Exercise: Run Linear Regression Model

1. Upload dataset in the colab

Run the loader cell and load the “student grades” dataset into google colab.

2. Run the build in linear regression model

Run the build-in model to predict the target variable.

Coding Exercise 2: Implement a Manual Regression Model

Implement algorithm:

1. Initialization:

- Initialize the weights (theta) and the bias (theta_0) to some random values or zeros.
- Define hyperparameters like learning rate (alpha) and the number of iterations (num_iterations).

2. Training:

Iterate num_iterations times:

- Compute predictions:

$$\hat{y} = X \cdot \theta + \theta_0$$

- Compute the cost function (mean squared error):

$$J(\theta, \theta_0) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$$

- Compute gradients:

$$\begin{aligned} \frac{\partial J}{\partial \theta} &= \frac{1}{m} X^T \cdot (\hat{y} - y) \\ \frac{\partial J}{\partial \theta_0} &= \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i) \end{aligned}$$

- Update weights and bias:

$$\begin{aligned} \theta &:= \theta - \alpha \cdot \frac{\partial J}{\partial \theta} \\ \theta_0 &:= \theta_0 - \alpha \cdot \frac{\partial J}{\partial \theta_0} \end{aligned}$$

3. Prediction:

Given new input features X_{test} , predict the output y_{pred} using the learned parameters:

$$X_{\text{test}} \cdot \theta + \theta_0$$

```
LinearRegression:
  initialize:
    Initialize theta (weights) and theta_0 (bias)
    Initialize learning rate (alpha) and number of iterations (num_iterations)

  train:
    for iter in range(num_iterations):
      Compute predictions: y_hat = X.dot(theta) + theta_0
      Compute cost: J = (1/2m) * sum((y_hat - y)^2)
      Compute gradients:
        dtheta = (1/m) * X.T.dot(y_hat - y)
        dtheta_0 = (1/m) * sum(y_hat - y)
      Update weights and bias:
        theta = theta - alpha * dtheta
        theta_0 = theta_0 - alpha * dtheta_0

  predict:
    Compute predictions: y_pred = X_test.dot(theta) + theta_0
    Return y_pred
```

Remember:

Linear regression assumes that there is a linear relationship between the input features and the target variable. It models this relationship using a linear equation of the form:

$$y = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \dots + \theta_n \cdot x_n$$

where:

- y is the predicted output (target variable).
- $\theta_0, \theta_1, \dots, \theta_n$ are the parameters (weights or coefficients) to be learned.
- x_1, x_2, \dots, x_n are the input features.
- θ_0 is the bias term (intercept).

Instructions:

1. Implement the above-mentioned algorithm
2. Compare it with the build-in model
3. Write a detailed analysis