Practical 3
19BCE248
2CS501

AIM:- Simple and Multiple Linear Regression using Gradient Descent & Normal Equation Method (without using sklearn or equivalent library for both)

About Dataset (Boston):

Boston is a pre-built dataset for practising Regression problems.This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass.
Rows: 506
Columns: 13
The target column describes the target variable i.e. cost of house according to parameters.

Preprocessing:

- Firstly dividing the entire dataset into Training and Testing dataset.
- Training rows: 404 Testing Row:102
- 20% testing 80%training
- Next preprocessing is done on data.
- The MinMaxScaler method is used to normalize the data.

Equation:

Linear Regression:

$$Y_i = \beta_0 + \beta_1 X_i$$

Here as there is only one variable so the standard line equation y=mx+c is more than enough to build our model. Here the two parameters B0 and B1 will act as theta parameter.

Gradient descendent:

Gradient descent algorithm

repeat until convergence {

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

$$(\text{for } j = 1 \text{ and } j = 0)$$

}

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Above shown is the formula for linear regression using gradient descent.
Θ: Known as theta specifies the multiplying parameter for linear regression. In other terms also specified as weight.

J(Θ): Cost function that signifies how different our model is predicting from actual value.

Convergence formula updates value of theta at regular intervals to achieve minimum error.

∝: Learning rate specifies how much the function should converge. It plays a very significant role in building the model and increasing accuracy of the model.

Normal Equation:

$$\boxed{\theta = \left(X^T X\right)^{-1} . \left(X^T y\right)}$$

A very simple way to approximate theta values by just using this equation.
X: Input parameters
Y:Target Variable.

Conclusion:

MAE	MSE
0.184252	0.059532
0.184235	0.059527
0.243400	0.093998
0.184039	0.059462
0.184039	0.059462
0.185880	0.060063
0.184039	0.059462
0.460756	0.294235
0.184039	0.059462
0.234718	0.088561
0.074675	0.009881

The above partition shows Using Gradient descent and the last row using Normal Equation.

Here we can see drastic change in MAE and MSE values. Thus we can conclude that for these dataset Normal Equation gives better results.

Does the Normal Equation work well in every case?
No, consider a situation with more than 1lac row and compute the time it takes to carry out Matrix operation.

Thus it drastically depends on the dataset size for choosing the Algorithm to be used.