

Lab Assignment 4:

Objective: To implment Linear Regression in Python from scratch.

Name: AAKASH VERMA

Reg. No. : 24-08026

Course: M.Tech.(Cyber Security)

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
```

```
In [10]: # Generate a dataset
np.random.seed(0)
num_samples = 1000 # Number of samples
features = 2 * np.random.rand(num_samples, 1) # Random features between 0 and 1
targets = 4 + 3 * features + np.random.randn(num_samples, 1) # Linear relationship
```

```
In [11]: # Prepare the feature matrix by adding a column of ones
def prepareMatrix(X):
    num_samples = len(X)
    X_b = np.c_[np.ones((num_samples, 1)), X] # Add x0 = 1 to each instance
    return X_b
```

```
In [12]: X_b_prepared = prepareMatrix(features)
```

```
In [13]: # Compute parameters using Normal Equation
def computeParam(X_b, y):
    theta = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
    return theta
```

```
In [14]: params = computeParam(X_b_prepared, targets)
intercept = params[0][0]
slope = params[1][0]
print(f"Learned parameters: Intercept = {intercept:.2f}, Slope = {slope:.2f}")

Learned parameters: Intercept = 4.08, Slope = 2.95
```

```
In [15]: # Make predictions using the Learned parameters
def predictValues(X, intercept, slope):
    return intercept + slope * X
```

```
In [16]: predictions = predictValues(features, intercept, slope)
```

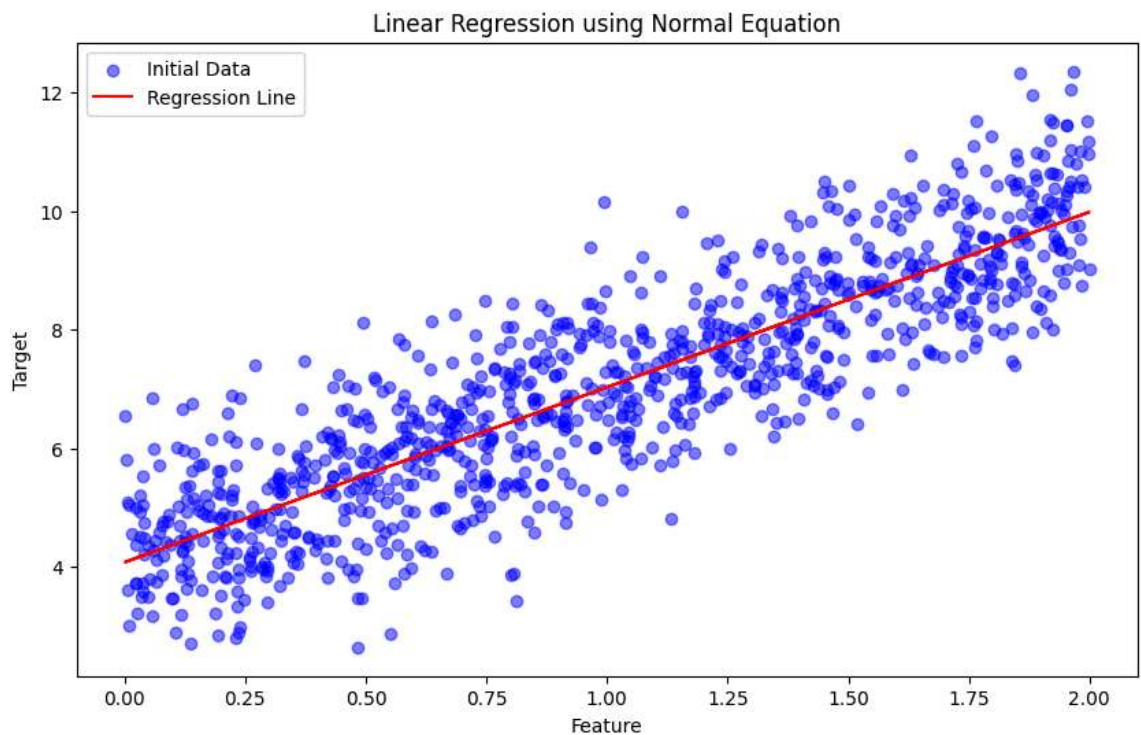
```
In [17]: # Compute residuals and Residual Sum of Squares (RSS)
def computeResiduals(y_true, y_pred):
    return y_true - y_pred

def computeRSS(residuals):
    return np.sum(np.square(residuals))
```

```
In [18]: residuals = computeResiduals(targets, predictions)
rss = computeRSS(residuals)
print(f"Residual Sum of Squares (RSS): {rss:.2f}")
```

Residual Sum of Squares (RSS): 932.69

```
In [19]: # Plot the initial data and regression line
plt.figure(figsize=(10, 6))
plt.scatter(features, targets, color='blue', label='Initial Data', alpha=0.5)
plt.plot(features, predictions, color='red', label='Regression Line')
plt.xlabel('Feature')
plt.ylabel('Target')
plt.title('Linear Regression using Normal Equation')
plt.legend()
plt.show()
```



In []: