19/3/25

LAB - 3

LINEAR REGRESSION :-

First type :-

Using $y = mx + c$ or $y = ax + b$ formula to find the slope and intercept for the dataset given below :-

X (week)	Y (sales in thousands)
1	2
2	4
3	5
4	9

```
import pandas as pd
# Load the data from the csv file
data = pd.read_csv('/content/linear_data.csv')
x = data['x']
y = data['y']
# Calculate the mean of x and y
x_mean = x.mean()
y_mean = y.mean()
# Calculate the slope (b1)
numerator = ((x - x_mean) * (y - y_mean)).sum()
denominator = ((x - x_mean)**2).sum()
b1 = numerator / denominator
# calculate the intercept (b0)
b0 = y_mean - (b1 * x_mean)
print("slope (b1):", b1)
print("Intercept (b0):", b0)
x_new = 5
y_predicted = b0 + (b1 * x_new)
print("Predicted value for x=5:", y_predicted)
```

Bafna Gold
Date: Page:

Output :-
slope (b1) : 2.2
Intercept (b0) : -0.5
Predicted value for x = 5 : 10.5

Second type :-
Using the matrix form, where we want to fit the model using $Y = \beta_0 + \beta_1 X$ where slope and Intercept are calculated using formula $a = ((X^T X)^{-1} X^T) Y$

```
import numpy as np
import pandas as pd
data = pd.read_csv('/content/linear_data.csv')
x = data['x']
y = data['y']
X = np.column_stack((np.ones(len(x)), x))
beta = np.linalg.inv(X.T @ X) @ X.T @ y
b0 = beta[0]
b1 = beta[1]
print("slope (b1):", b1)
print("Intercept (b0):", b0)
x_new = 5
y_predicted = b0 + (b1 * x_new)
print("Predicted value for x = 5:", y_predicted)
```

Output :-
slope (b1) : 2.2
Intercept (b0) = -0.5
Predicted value for x = 5 : 10.5

Graphs :-

For Type-1 & Type-2 :-