LAB-3 LINFAR REGRESSION: First type of Using y-mx+c or y = ax+b fromula to find the glope and intercept for the dotaset given below o- X (week) Y (soles in thousands) A 4 B 5 4 9 Impart pondas as pd # Load the dota from the csv file data = pd. read-csv ('/content)/linear_data.csv') x = data ['x'] y = data ['x'] # (alculate the mean of x and y x-meon = x-meon () y-meon = y-mean () # (alculate the slope (b1) namerator = ((x-x-mean)*(y-y-mean)). even () denominators = ((x-x-mean)*x1). sum () b1 = numerators / denominators # calculate the intercept (b0) b0 = y-mean - (b1x-mean) print ("glope (1)" - y-mean) print ("glope (1)" - y-mean)	ا ما ما ما	
First type 3- Using y-mxt c 08 y = ax+b formula to find the slope and intercept for the dataset given below 3- X(week) Y(sales in thousands) 2 4 3 5 4 9 import pandas as pd # Load the data from the esv file data = pd. read-csv ('/content)/linear_data.csv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x-mean = x.mean () # (alculate the slope (b1) numerator = ((x-x-mean)* (y-y-mean)). sum () denominates = ((x-x-mean)*x1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1 x-mean) print ("slope (1)" x-mean)	19/3/25	LAB-3
First type 3- Using y-mxt c 08 y = ax+b formula to find the slope and intercept for the dataset given below 3- X(week) Y(sales in thousands) 2 4 3 5 4 9 import pandas as pd # Load the data from the esv file data = pd. read-csv ('/content)/linear_data.csv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x-mean = x.mean () # (alculate the slope (b1) numerator = ((x-x-mean)* (y-y-mean)). sum () denominates = ((x-x-mean)*x1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1 x-mean) print ("slope (1)" x-mean)		LINFAR REGRESSION:
to find the slope and intercept formula to find the slope and intercept for the dataset given below 6- X (week) Y (sales in thousands) 2 4 3 5 4 9 import pondas as pd # Load the dota from the esv file data = pd. **ead-csv ('/conten)/linear_data.csv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x-mean = x.mean () # (alculate the slope (b1) pamerator = ((x-x-mean)* (y-y-mean)). sum () denominator = ((x-x-mean)**x1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (1):" x-mean)		First type or
to find the slope and intercept to the dataset given below 6- X (week) Y (sales in thousands) A 4 3 5 4 9 import pandas as pd # Load the data from the esv file data = pd. read - (sv ('/content) / linear - data. (sv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x-mean = x-mean () # (alculate the slope (b1) namerator = ((x - x - mean)* (y - y - mean)). sum () denominates = ((x - x - mean)* x + 1). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y - mean - (b1* x - mean) print ("slope (1)" '' v - y - mean)		
X (week) X (week) Y (sales in thousands) A 4 3 5 4 9 Import pandas as gd # Load the data from the esv file data = pd. read_csv ('/content)/linear_data.csv') X = data ['x'] y = data ['y'] # (alculate the mean of x and y X-mean = x-mean () # (alculate the slope (b1) namesator = ((x-x-mean)*(y-y-mean)). sum () denominates = ((x-x-mean)**x2). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (1):"		to find the slope and intercept to
impost pandas as gd # Load the data from the esv file data = pd. read - (sv ('/content) / linear - data . (sv') x = data ['x'] y = data['y'] # (alculate the mean of x and y x - mean = x - mean () # (alculate the slope (b1) numerator = ((x - x - mean)* (y - y - mean)). sum () denominator = ((x - x - mean)* x 2). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y - mean - (b1* x - mean) print ("glope (1):" x - mean)		the dataset given below in
impost pandas as gd # Load the data from the esv file data = pd. read - (sv ('/content) / linear - data . (sv') x = data ['x'] y = data['y'] # (alculate the mean of x and y x - mean = x - mean () # (alculate the slope (b1) numerator = ((x - x - mean)* (y - y - mean)). sum () denominator = ((x - x - mean)* x 2). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y - mean - (b1* x - mean) print ("glope (1):" x - mean)		X (week) Y (sales in thousands)
impost pandas as pd # Load the data from the esv file data = pd. read_csv ('/content)/linear_data.csv') x = data ['x'] # (alculate the mean of x and y x-mean = x-mean () # (alculate the slope (b1) numerator = ((x-x-mean)* (y-y-mean)). sum () denominator = ((x-x-mean)*x2). sum () b1 = numerators / denominator # calculate the intercept (b0) b0 = y-mean - (b1*x-mean) print ("slope (11):" - mean)		- A
import pandas as pd # Load the data from the esv file data = pd. read_csv ('/content)/linear_data.csv') x = data ['x'] # (akulate the mean of x and y x-mean = x-mean () # cakulate the slope (b1) numerator = ((x-x-mean)* (y - y-mean)). sum () denominator = ((x-x-mean)**x1). sum () b1 = numerator / denominator # cakulate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (11):"		
impost pandas as pd # Load the data from the esv file data = pd. read = csv ('/content)/linear = data.csv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x = mean = x - mean () # (alculate the slope (b1) numerator = ((x - x - mean)* (y - y - mean)). sum () denominator = ((x - x - mean)* * x). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y - mean = (b1* x - mean) print ("slope (1):" + x - mean)		
# Load the dota from the esv file data = pd. read_csv ('/content/linear_data.csv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x-meon = x-mean () y-meon = y. mean () # (alculate the slope (b1) numerator - ((x-x-mean)* (y - y-mean)). sum () denominator = ((x-x-mean)**x1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1*x-mean) print ("slope (11):" + intercept)		
# Load the dota from the esv file data = pd. read_csv ('/content/linear_data.csv') x = data ['x'] y = data ['y'] # (alculate the mean of x and y x-meon = x-mean () y-meon = y. mean () # (alculate the slope (b1) numerator - ((x-x-mean)* (y - y-mean)). sum () denominator = ((x-x-mean)**x1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1*x-mean) print ("slope (11):" + intercept)	4	
# Calculate the slope (b1) The server of the slope (b1) The	4	thought panaas as pol
# (alculate the mean of x and y x-mean = x-mean () # calculate the slope (b1) numerator = ((x-x-mean)* (y-y-mean)). sum () denominator = ((x-x-mean)**1). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (11):" 1:"		data-od in data from the esv file
# (alculate the mean of x and y x-mean = x-mean () # calculate the slope (b1) numerator = ((x-x-mean)* (y-y-mean)). sum () denominator = ((x-x-mean)**1). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (11):" 1:"		x = data ['x']
# (alculate the mean of x and y x-mean = x-mean () # (alculate the slope (b1) numerator = ((x-x-mean)* (y-y-mean)). sum () denominator = ((x-x-mean)**1). sum () b1 = numerator denominator # calculate the intercept (b0) b0 = y-mean = (b1* x-mean) print ("slope (11):" 1)	4.1	
y-mean = x-mean () # calculate the slope (b1) numerator - ((x-x-mean)* (y - y-mean)). sum () denominator = ((x-x-mean)***1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (11):" 1:)	- 11	
# calculate the slope (b1) numerator = ((x-x-mean)* (y - y-mean)). sum () denominator = ((x-x-mean)**1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (11):" 1:)		= x-mean ()
# calculate the slope (b1) numerator - ((x-x-mean)* (y - y-mean)). sum () denominator = ((x-x-mean)**1). sum () b1 = numerator / denominator # calculate the intercept (b0) b0 = y-mean - (b1* x-mean) print ("slope (1):" 1:)		y- mean = y. mean ()
denominates = ((x-x-mean)* (y - y-mean)). sum () bl = numerators / denominators # calculate the intercept (b0) bo = y-mean - (b1* x-mean) print ("slope (1):" 1)	7	calculate the slope (b)
bl = numerators / denominators # calculate the intercept (b0) bo = y-mean - (b1* x-mean) print ("slope (1):" 1:)	r	umeso tos - ((x-x moss) * (u
# calculate the intercept (b0) bo= y-mean - (b1* x-mean) print ("slope (1):" 1:)	- 11	1 X - X - mean 1 x x)
print ("slope (1):" 1:)		denomina to
print ("slope (1):" 1:)		to calculate the intercept (60)
51000		yemenn e (L 1x
1 10	The second secon	210 De (11)
print ("Intercept (60):", 60) X-new= 5		X-new = 5
y-predicted - 60-1 (11)		y-predicted - bod (1)
print ("Predicted" x new)		print ("Predicted"
print ("predicted value for x=5:", y-predicted)		value for x:5:", y = predicted
1 g species		of the parties of the

Bafna Gold output: slope (b1): 2.2 : 2000 3 1 Intercept (bo): -0.5 Predicted value for x - 5:10.5 Second type :-Using the matrix form, where we want to git the model using $Y = Bo + B_1 \times$ where slope and Intercept are calculated using formula a = $((x + x) - 1 \times +) \times$ import numpy as np impost pandas as pd data = pd. read-csv ('/content/linear_data.csv')
x = data ['x'] y = data[iyi] X = np. column_stack ((np. ones (len(x)),x)) beta = np. linalg. inv (X. IQX) @ X.T@ y bo = betaco] bl=betati] print (" slope (b1):", b1) print (Intercept (60):", 60) X_new = 5 y-predicted = bo + (b) * x-new) prind ("Predicted value for x = 5:", y - predicted) Output : slope (b1): 2.2 Intercept (60) = -0.5 Predicted value for x = 5: 10.5

