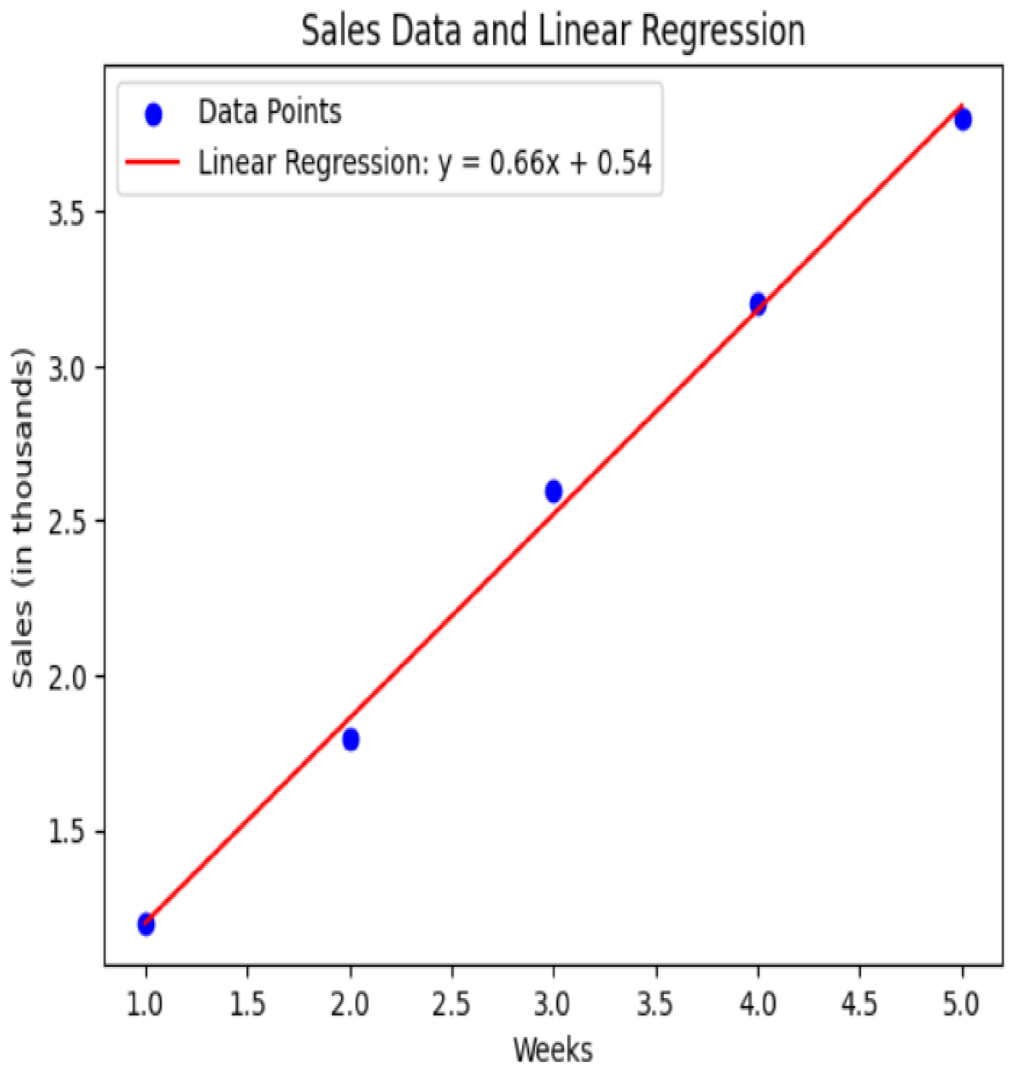
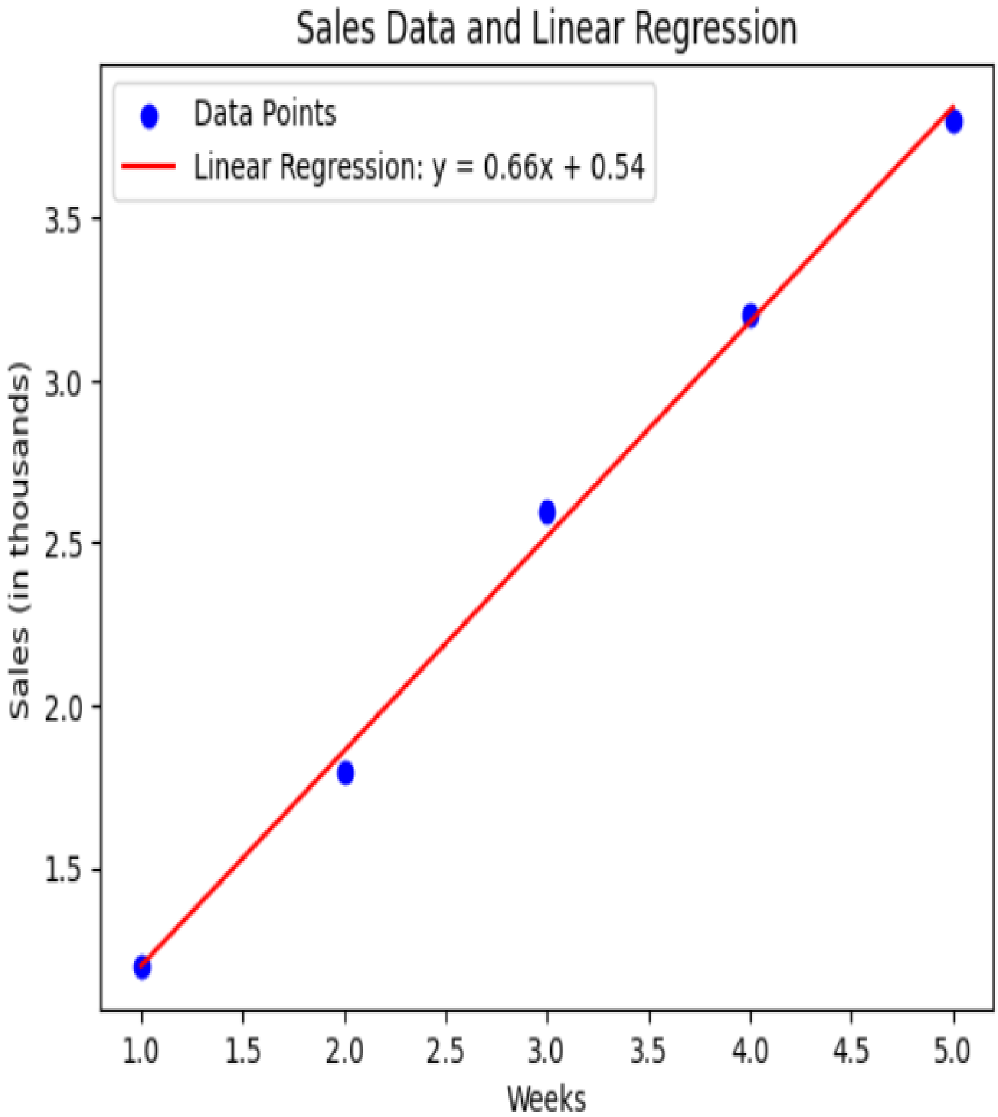
1BM22CS241

SANJEET PRAJWAL PANDIT

```
   xi(week)  yi(Sales in thousands)
0         1                     1.2
1         2                     1.8
2         3                     2.6
3         4                     3.2
4         5                     3.8
The regression equation is: y = 0.66x + 0.54
Predicted sales for the 7th week: 5.16 thousand
Predicted sales for the 9th week: 6.48 thousand
```



```
   xi(week)  yi(Sales in thousands)
0         1                     1.2
1         2                     1.8
2         3                     2.6
3         4                     3.2
4         5                     3.8
The regression equation is: y = 0.66x + 0.54
Predicted sales for the 7th week: 5.16 thousand
Predicted sales for the 9th week: 6.48 thousand
```



DATE: PAGE

19/3/25

Q (1) Consider binary classification problem where we want to predict whether a student will pass or fail based on their study hours. Logistic regression model has been trained, the learned parameters are $a_0 = -5$ (intercept) & $a_1 = 0.8$ (coeff for study hrs)

(a) Write logistic regression equation for this problem
(b) Calculate probability that student who studies for 7 hours
(c) Determine predicted class (pass or fail) for this student based on threshold of 0.5.

(2) Consider $z = [2,1,0]$ for three classes. Apply softmax function to find probability values of three classes.

Linear regression

sales.csv

Week	Sales(in th)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

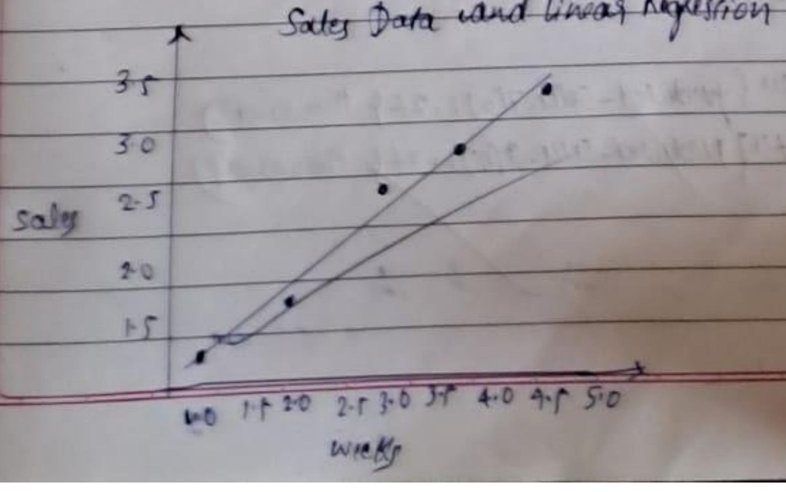
(1)
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset = pd.read_csv('/content/sales.csv')
print(dataset.head())

weeks = dataset['xi(week)'].values
sales = dataset['yi(sales in th)'].values
n = len(weeks)

sum_x = np.sum(weeks)
sum_y = np.sum(sales)
sum_x2 = np.sum(weeks**2)
sum_xy = np.sum(weeks*sales)
```

DATE: PAGE:

```
# slope(m) , intercept (b)

m = (n * sum_xy - sum_x * sum_y) / (n* sum_x2 - sum_x**2)
b = (sum_y-m * sum_x) /n

print(f" Reg eqn is : y = {m: .2f}x + {b: .2f}")
week_7 = 7
week_9 = 9

predicted_sales_7 = m * week_7 + b
predicted_sales_9 = m* week_9 +b

print(f" {predicted_sales_7 : .2f} thousand")
print(f" {predicted_sales_9: .2f} thousand")

plt.scatter(weeks, sales, color='blue', label='Data Points')
plt.plot( weeks, m * weeks +b, color='red', label= f'Lin Reg : y= {m: .2f}x
                                                              + {b: .2f} ')

plt.legend()
plt.show()
```

Output:

Regression equation is: y = 0.66x + 0.54
Predicted sales for 7th week : 5.16 thousand
Predicted sales for 9th week : 6.48 thousand



DATE:

(2) Matrix method

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset = pd.read_csv('/content/sales.csv')
print(dataset.head())
weeks = dataset['xi(weeks)'].values
sales = dataset['yi (sales in th)'].values

x = weeks.reshape(-1,1)
y = sales.reshape(-1,1)

x_b = np.c_[np.ones((len(x),1)),x]

theta = np.linalg.inv(x_b.T.dot(x_b)).dot(x_b.T).dot(y)

b = theta[0]
m = theta[1]

print(f"Reg eqn is: y= {m[0]:.2f} x + {b[0]:.2f}")

predicted_sales_7 = m*7 +b
predicted_sales_9 = m*9+b

print(f"{predicted_sales_7[0]:.2f} thousand")
print(f"{predicted_sales_9[0]:.2f} thousand")
```

DATE: PAGE:

```
plt.scatter(weeks, sales, color='blue', label='Data points')
plt.plot(weeks, m*weeks+b, color='red', label=f'Lin Reg:
        y={m[0]:.2f}x+{b[0]:.2f}')
plt.xlabel('Weeks')
plt.ylabel('Sales (in thousands)')
plt.title('Sales Data and Linear Regression')
plt.legend()
plt.show()
```

Output:

Regression equation is: $y = 0.66x + 0.54$

predicted sales for 7th week: 5.16 thousand

predicted sales for 9th week: 6.48 thousands