Linear Regression

Assignment

Weight (X)	Price (Y)
2	35
4	60
5	20
3	50
6	50
5	55
7	60

Task 01:
Your objective is to manually compute the slope (M) and y-intercept (C) using Ordinary Least Squares Linear Regression. Once determined, apply these values to predict the price when the vegetable weight is 6.

Task 02:
Compute the residuals for each data point.

Task 03:
Calculate both the Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Final Task:
Generate an Excel file for the given dataset. Utilize Python for all the calculations.

Note: To validate your manual calculations, use the entire dataset. It's unnecessary to split the dataset.

Task 01:

Your objective is to manually compute the slope (M) and y-intercept (C) using Ordinary Least Squares Linear Regression. Once determined, apply these values to predict the price when the vegetable weight is 6.

Solutions:

To calculate OLS regression parameters m and c, we can use the formula:

$Y = mX + c$

$c = \bar{Y} - m\bar{X}$

Where Y is dependent variable

X is the independent variable

m is the slope(co-efficient) and c is intercept

and

$m = (\Sigma(X - \bar{X})(Y - \bar{Y})) / \Sigma(X - \bar{X})^2$

$\Sigma X = 2+4+5+3+6+5+7 = 32$

$\bar{X} = 32/7= 4.57$

$\Sigma Y= 35+60+20+50+50+55+60= 330$

$\bar{Y}= 47.14$

$\Sigma X = 2+4+5+3+6+5+7 = 32$

$\Sigma Y= 35+60+20+50+50+55+60= 330$

$n = 7$

$m = (\Sigma(X - \bar{X})(Y - \bar{Y})) / \Sigma(X - \bar{X})^2$

$m = (2- 4.57)(35-47.14)+(4- 4.57)(60-47.14)+(5-4.57)(20-47.14)+(3-4.57)(50-47.14)+(6-4.57)(50-47.14)+(5-4.57)(55-47.14)+(7-4.57)(60-47.14)/$
$(2 - 4.57)^2 + (4 - 4.57)^2+ (5-4.57)^2 + (3-4.57)^2 + (6-4.57)^2 + (5-4.57)^2 + (7- 4.57)^2$

$= [(-2.57 * -12.14) + (-0.57 * 12.86) + (0.43 * -27.14) + (-1.57 * 2.86) + (1.43 * 2.86) + (0.43 * 7.86) + (2.43 * 12.86)] /$
$[(6.57)^2 + (1.43)^2 + (0.43)^2 + (1.57)^2 + (1.43)^2 + (0.43)^2 + (2.43)^2]$

$= (31.1998-7.3302-11.6702-4.4902+4.0898+3.3798+31.2498)/$
$(43.1649+2.0449+0.1849+2.4649+2.0449+0.1849+2.43)$
$=46.4286/52.5194= 0.88402$

$c = \bar{Y} - m* \bar{X}$
$= 47.14 - 0.88402 * 4.57$
$= 47.14- 4.0399714$
$= 43.1000286$

$m= 0.88402$
$c= 43.1000286$
So, the regression equation is:

Price= 0.88402 * weight + 43.1000286

So predict the price when the vegetable weight is 6

Price= 0.88402 * 6 + 43.1000286

The predicted Price = 48.4041468

Task 02:

Compute the residuals for each data point.

Solution:

Now, we have the regression equation:

Price= 0.88402 * weight + 43.1000286

Now we have to calculate the predicted Price for each data point based on this equation.

Weight (X)	**Price (Y)**	**Predicted Price**	**Residual**
2	35	0.88402 * 2 + 43.1000286=44.8680686	-9.8680686
4	60	0.88402 * 4 + 43.1000286= 46.6361068	13.3638932
5	20	0.88402 * 5 + 43.1000286= 47.5201286	-27.5201286
3	50	0.88402 * 3 + 43.1000286= 45.7520886	4.2479114
6	50	0.88402 * 6 + 43.1000286= 48.4041486	1.5958514
5	55	0.88402 * 5 + 43.1000286= 47.5201286	7.4798714
7	60	0.88402 * 7 + 43.1000286= 49.2881686	10.7118314

Task 03:

Calculate both the Mean Squared Error (MSE) and the Mean Absolute Error (MAE)

Solutions:

Residuals = [-9.8680686, 13.3638932, -27.5201286, 4.2479114, 1.5958514, 7.4798714, 10.7118314]

Squared Residuals = $[(-9.8680686)^2, (13.3638932)^2, (-27.5201286)^2, (4.2479114)^2, (1.5958514)^2, (7.4798714)^2, (10.7118314)^2]$

Mean Squared Error (MSE):

MSE = (1/n) * Σ(squared residuals)

$= (1/7) * [(-9.8680686)^2 + (13.3638932)^2 + (-27.5201286)^2 + (4.2479114)^2 + (1.5958514)^2 + (7.4798714)^2 + (10.7118314)^2]$

=1/7 * (97.3788 + 178.5936 + 757.3575 + 18.0448 + 2.5467 + 55.9485 + 114.7433)

= (1/7) * 1224.6132= 174.9447

MSE = 174.9447

Absolute Residuals = [|-9.8680686|, |13.3638932|, |-27.5201286|, |4.2479114|, |1.5958514|, |7.4798714|, |10.7118314|]

Mean Absolute Error (MAE):

MAE = (1/n) * Σ(absolute residuals) = (1/7) * [|-9.8680686| + |13.3638932| + |-27.5201286| +|4.2479114| + |1.5958514| + |7.4798714| + |10.7118314|]

=(1/7) * 74.787556

= 10.6839

MAE = 10.6839

Final Task:
Generate an Excel file for the given dataset. Utilize Python for all the calculations.

```
#Python Code

import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
import openpyxl


datatable = {
   "Weight": [2, 4, 5, 3, 6, 5, 7],
   "Price": [35, 60, 20, 50, 50, 55, 60]
}
df = pd.DataFrame(datatable)

model = LinearRegression()
model.fit(df[['Weight']], df['Price'])

slope = model.coef_[0]
intercept = model.intercept_

df['Predicted_Price'] = model.predict(df[['Weight']])
df['Residual'] = df['Price'] - df['Predicted_Price']
```

```python
mse = mean_squared_error(df['Price'], df['Predicted_Price'])
mae = mean_absolute_error(df['Price'], df['Predicted_Price'])

summary_df = pd.DataFrame({
    'Slope (m)': [slope],
    'Intercept (c)': [intercept],
    'Mean Squared Error (MSE)': [mse],
    'Mean Absolute Error (MAE)': [mae]
})


wb = openpyxl.Workbook()
ws1 = wb.active
ws1.append(["Weight", "Price", "Predicted Price", "Residual"])

for row in df.itertuples():
    ws1.append([row.Weight, row.Price, row.Predicted_Price, row.Residual])

ws2 = wb.create_sheet("Regression Summary")
ws2.append(["Slope (m)", "Intercept (c)", "Mean Squared Error (MSE)", "Mean Absolute Error (MAE)"])
ws2.append([slope, intercept, mse, mae])

wb.save("food_data.xlsx")

print('Excel file \'food_data.xlsx\' created with the data sheet.')
```