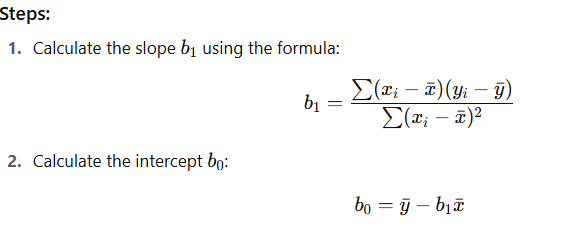
**Practical Assignment – 5**

**1. Consider following observations/data. And apply simple linear regression and find out estimated coefficients b0 and b1. (Use numpy package) x= [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13] y = ([1, 3, 2, 5, 7, 8, 8, 9, 10, 12, 16, 18]**

****

import numpy as np

import matplotlib.pyplot as plt

x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13])

y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12, 16, 18])

x\_mean = np.mean(x)

y\_mean = np.mean(y)

b1 = np.sum((x - x\_mean) \* (y - y\_mean)) / np.sum((x - x\_mean)\*\*2)

b0 = y\_mean - b1 \* x\_mean

print(b0, b1)

# Predicted values

y\_pred = b0 + b1 \* x

# Plot

plt.scatter(x, y, color='blue', label='Original data')

plt.plot(x, y\_pred, color='red', label='Regression line')

plt.xlabel('x')

plt.ylabel('y')

plt.title('Simple Linear Regression')

plt.legend()

plt.grid(True)

plt.show()

**2. Consider following observations/data. And apply simple linear regression and find out estimated coefficients b1 and b1 Also analyse the performance of the model (Use sklearn package) x = np.array([1,2,3,4,5,6,7,8]) y = np.array([7,14,15,18,19,21,26,23])**

import numpy as np

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# Data

x = np.array([1,2,3,4,5,6,7,8]).reshape(-1, 1) #sklearn expects 2D input for X

y = np.array([7,14,15,18,19,21,26,23])

# Create and train the model

model = LinearRegression()

model.fit(x, y)

# Estimated coefficients

b0 = model.intercept\_

b1 = model.coef\_[0]

# Predictions

y\_pred = model.predict(x)

# Performance metrics

mse = mean\_squared\_error(y, y\_pred)

r2 = r2\_score(y, y\_pred)

print(f"Estimated intercept (b0): {b0}")

print(f"Estimated slope (b1): {b1}")

print(f"Mean Squared Error (MSE): {mse}")

print(f"R^2 score: {r2}")

**3. Consider the student data set It can be downloaded from: https://drive.google.com/open?id=1oakZCv7g3mlmCSdv9J8kdSaqO5\_6dIOw Write a programme in python to apply simple linear regression and find out mean absolute error, mean squared error and root mean squared error.**

import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

# 1. Load the dataset

df = pd.read\_csv('student\_scores.csv') # adjust filename if needed

# 2. Inspect data—print the first few rows to see column names

print(df.head())

# e.g., perhaps columns are: 'Hours' and 'Scores'

# 3. Define feature(s) and target

X = df[['Hours']].values # 2D array for sklearn

y = df['Scores'].values # 1D array

# 4. Initialize and train the linear regression model

model = LinearRegression()

model.fit(X, y)

# 5. Make predictions on the same data (or a test split)

y\_pred = model.predict(X)

# 6. Compute error metrics:

mae = mean\_absolute\_error(y, y\_pred)

mse = mean\_squared\_error(y, y\_pred)

rmse = np.sqrt(mse)

print("Mean Absolute Error (MAE):", mae)

print("Mean Squared Error (MSE):", mse)

print("Root Mean Squared Error (RMSE):", rmse)

**Set B**

1. **Write a python program to implement multiple Linear Regression model for a car dataset. Dataset can be downloaded from:** [**https://www.w3schools.com/python/python\_ml\_multiple\_regression.asp**](https://www.w3schools.com/python/python_ml_multiple_regression.asp)

import pandas as pd

from sklearn import linear\_model

#import matplotlib.pyplot as plt

# Load the dataset

data = {

'Weight': [2300, 2500, 2700, 3000, 3200, 3400, 3600, 3800, 4000, 4200],

'Volume': [1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200],

'CO2': [99, 100, 103, 105, 107, 109, 110, 113, 115, 117]

}

#data=pd.read\_csv('data.csv')

df = pd.DataFrame(data)

# Features (X) and Target (y)

X = df[['Weight', 'Volume']] # Independent variables

y = df['CO2'] # Dependent variable

# Create linear regression model

model = linear\_model.LinearRegression()

model.fit(X, y)

# Print coefficients

print("Intercept:", model.intercept\_)

print("Coefficients:", model.coef\_)

# Predicting CO2 for a new car

# Example: Weight=3300, Volume=1800

predicted\_co2 = model.predict([[3300, 1800]])

print(f"Predicted CO2 for Weight=3300 & Volume=1800: {predicted\_co2[0]:.2f}")

1. **Write a python programme to implement multiple linear regression model for stock market data frame as follows:**

**Stock\_Market =**

**{'Year':[2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2016,2016,20 16,2016,2016,2016,2016,2016,2016,2016,2016,2016],**

**'Month':[12,11,10,9,8,7,6,5,4,3,2,1,12,11,10,9,8,7,6,5,4,3,2,1],**

**'Interest\_Rate':[2.75,2.5,2.5,2.5,2.5,2.5,2.5,2.25,2.25,2.25,2,2,2,1.75,1.75,1.75,1.75,1.75,1.75,**

**1.75,1.75,1.75,1.75,1.75],**

**'Unemployment\_Rate':[5.3,5.3,5.3,5.3,5.4,5.6,5.5,5.5,5.5,5.6,5.7,5.9,6,5.9,5.8,6.1,6.2,6.1,6.1,6.1,5.9,6.2,6.2,6.1],**

**'Stock\_Index\_Price':[1464,1394,1357,1293,1256,1254,1234,1195,1159,1167,1130,1075,1047,965,943, 958,971,949,884,866,876,822,704,719] }**

**And draw a graph of stock market price verses interest rat**e.

import pandas as pd

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

import numpy as np

# Data

Stock\_Market = {

'Year': [2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,

2016,2016,2016,2016,2016,2016,2016,2016,2016,2016,2016,2016],

'Month': [12,11,10,9,8,7,6,5,4,3,2,1,

12,11,10,9,8,7,6,5,4,3,2,1],

'Interest\_Rate': [2.75,2.5,2.5,2.5,2.5,2.5,2.5,2.25,2.25,2.25,2,2,

2,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75],

'Unemployment\_Rate': [5.3,5.3,5.3,5.3,5.4,5.6,5.5,5.5,5.5,5.6,5.7,5.9,

6,5.9,5.8,6.1,6.2,6.1,6.1,6.1,5.9,6.2,6.2,6.1],

'Stock\_Index\_Price': [1464,1394,1357,1293,1256,1254,1234,1195,1159,1167,

1130,1075,1047,965,943,958,971,949,884,866,876,822,704,719]

}

# Create DataFrame

df = pd.DataFrame(Stock\_Market)

# Features and target variable

X = df[['Year', 'Month', 'Interest\_Rate', 'Unemployment\_Rate']]

y = df['Stock\_Index\_Price']

# Create and train the model

model = LinearRegression()

model.fit(X, y)

# Print coefficients

print("Intercept:", model.intercept\_)

print("Coefficients:", list(zip(X.columns, model.coef\_)))

# Plot Stock Index Price vs Interest Rate

plt.figure(figsize=(8,5))

plt.scatter(df['Interest\_Rate'], y, color='blue', label='Actual Data')

# For visualization: predict Stock Price just varying Interest Rate, keeping others at mean

interest\_rate\_range = np.linspace(df['Interest\_Rate'].min(), df['Interest\_Rate'].max(), 100)

mean\_year = df['Year'].mean()

mean\_month = df['Month'].mean()

mean\_unemployment = df['Unemployment\_Rate'].mean()

X\_plot = pd.DataFrame({

'Year': [mean\_year]\*100,

'Month': [mean\_month]\*100,

'Interest\_Rate': interest\_rate\_range,

'Unemployment\_Rate': [mean\_unemployment]\*100

})

predicted\_prices = model.predict(X\_plot)

plt.plot(interest\_rate\_range, predicted\_prices, color='red', label='Regression Line')

plt.xlabel('Interest Rate')

plt.ylabel('Stock Index Price')

plt.title('Stock Index Price vs Interest Rate')

plt.legend()

plt.grid(True)

plt.show()