**WEEK 4 COM624 LAB ACTIVITIES: MULTIPLE LINEAR REGRESSION**

**Part 2: Multiple Linear Regression**

**Scenario**

You are working for a property analytics firm. Your task is to predict **house prices** based on multiple features such as **square footage**, **number of bedrooms**, and **location quality**.

**Dataset Download**

We will use the **USA Housing dataset** from GitHub. It contains columns like Avg. Area Income, Avg. Area House Age, Avg. Area Number of Rooms, and Price.

Download USA\_Housing.csv

Save the file as usa\_housing.csv in your project folder.

**Step-by-Step Instructions**

**Step 1: Set Up Your Environment**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

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

**Step 2: Load the Dataset**

df = pd.read\_csv('usa\_housing.csv')

print(df.head())

print(df.info())

**Step 3: Visualise Relationships**

sns.pairplot(df)

plt.suptitle("Pairwise Relationships", y=1.02)

plt.show()

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap")

plt.show()

**Step 4: Prepare the Data**

# Define independent variables (X) and dependent variable (y)

X = df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',

'Avg. Area Number of Bedrooms', 'Area Population']]

y = df['Price']

# Split into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Step 5: Build the Model**

model = LinearRegression()

model.fit(X\_train, y\_train)

# Print coefficients

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

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

**Step 6: Make Predictions**

y\_pred = model.predict(X\_test)

comparison = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

print(comparison.head())

**Step 7: Evaluate the Model**

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

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

print("Mean Squared Error:", mse)

print("R-squared:", r2)

**Step 8: Visualise Predictions**

plt.scatter(y\_test, y\_pred, color='purple')

plt.xlabel("Actual Prices")

plt.ylabel("Predicted Prices")

plt.title("Actual vs Predicted House Prices")

plt.show()

**THE END**

**Final Complete Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

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

df = pd.read\_csv('usa\_housing.csv')

print(df.head())

print(df.info())

sns.pairplot(df)

plt.suptitle("Pairwise Relationships", y=1.02)

plt.show()

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap")

plt.show()

X = df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',

'Avg. Area Number of Bedrooms', 'Area Population']]

y = df['Price']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

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

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

y\_pred = model.predict(X\_test)

comparison = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

print(comparison.head())

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

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

print("Mean Squared Error:", mse)

print("R-squared:", r2)

plt.scatter(y\_test, y\_pred, color='purple')

plt.xlabel("Actual Prices")

plt.ylabel("Predicted Prices")

plt.title("Actual vs Predicted House Prices")

plt.show()