**Question 1**

USING THE IRIS DATA SET IMPLEMENT MULTIPLE LINEAR REGRESSION USING LEAST SQUARE METHOD AND FIND R-SQUARE, MSE, MAE, RMSE.

**Source Code**

**Batch Gradient Descent**  
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

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

dataset\_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

columns = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'class']

iris = pd.read\_csv(dataset\_url, header=None, names=columns)

X = iris[['petal\_width', 'petal\_length', 'sepal\_length']]

y = iris['sepal\_width']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=40)

model = LinearRegression()

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

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

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

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

rmse = np.sqrt(mse)

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

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

print(f"Mean Absolute Error (MAE): {mae}")

print(f"Root Mean Squared Error (RMSE): {rmse}")

print(f"R-Squared (R²): {r2}")

plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)

plt.scatter(iris['sepal\_length'], iris['petal\_length'], color='blue', alpha=0.6)

plt.title("Sepal Length vs Petal Length")

plt.xlabel("Sepal Length")

plt.ylabel("Petal Length")

plt.subplot(2, 2, 2)

plt.scatter(iris['sepal\_width'], iris['petal\_length'], color='green', alpha=0.6)

plt.title("Sepal Width vs Petal Length")

plt.xlabel("Sepal Width")

plt.ylabel("Petal Length")

plt.subplot(2, 2, 3)

plt.scatter(iris['petal\_width'], iris['petal\_length'], color='red', alpha=0.6)

plt.title("Petal Width vs Petal Length")

plt.xlabel("Petal Width")

plt.ylabel("Petal Length")

plt.tight\_layout()

plt.show()

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred, color='blue', alpha=0.6, label="Predicted vs Actual")

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'r--', lw=2, label="Perfect Fit")

plt.xlabel("Actual Petal Length")

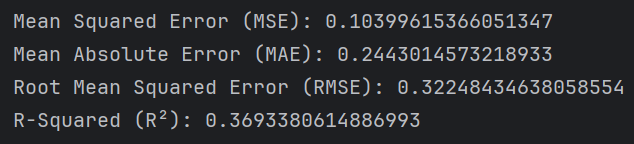
plt.ylabel("Predicted Petal Length")

plt.title("Actual vs Predicted Petal Length")

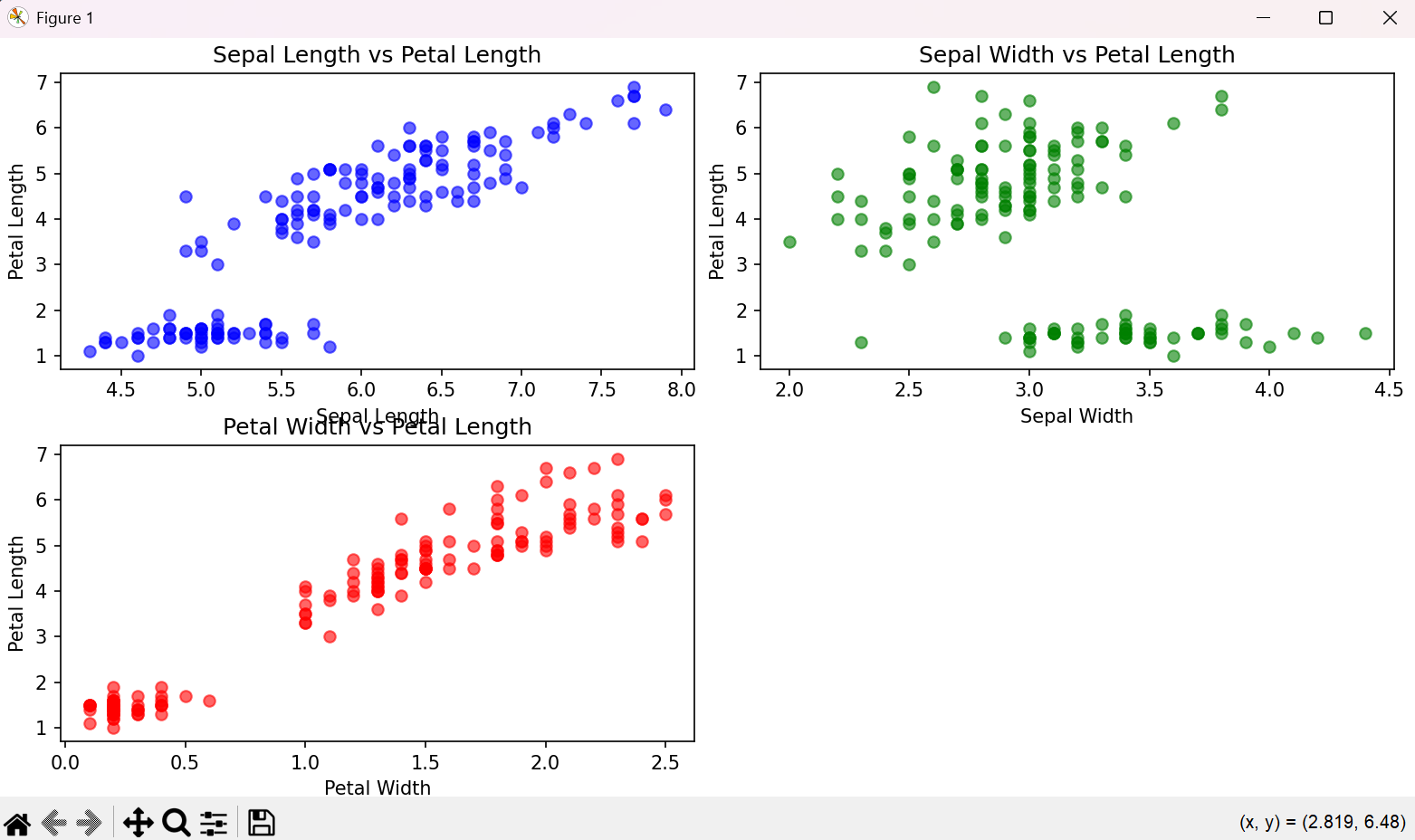
plt.legend()

plt.grid(True)

plt.show()

**Output**

*Terminal*

*Visualization*

**Question 2**

IMPLEMENT MULTIPLE LINEAR REGRESSION IN THE DIABETIC DATA SET IN THE SKLEARN LIBRARY.

**Source Code**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

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

from sklearn.datasets import load\_diabetes

diabetes\_data = load\_diabetes()

X = pd.DataFrame(diabetes\_data.data, columns=diabetes\_data.feature\_names)

y = diabetes\_data.target

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)

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

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

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

rmse = np.sqrt(mse)

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

print("Metrics:")

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

print(f"Mean Absolute Error (MAE): {mae}")

print(f"Root Mean Squared Error (RMSE): {rmse}")

print(f"R-Squared (R²): {r2}")

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred, color='blue', alpha=0.6)

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'r--', lw=2)

plt.xlabel("Actual")

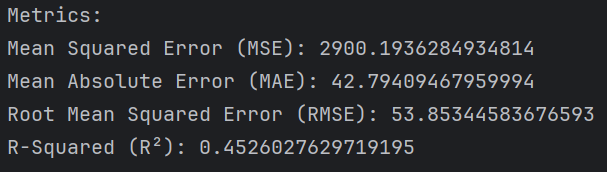
plt.ylabel("Predicted")

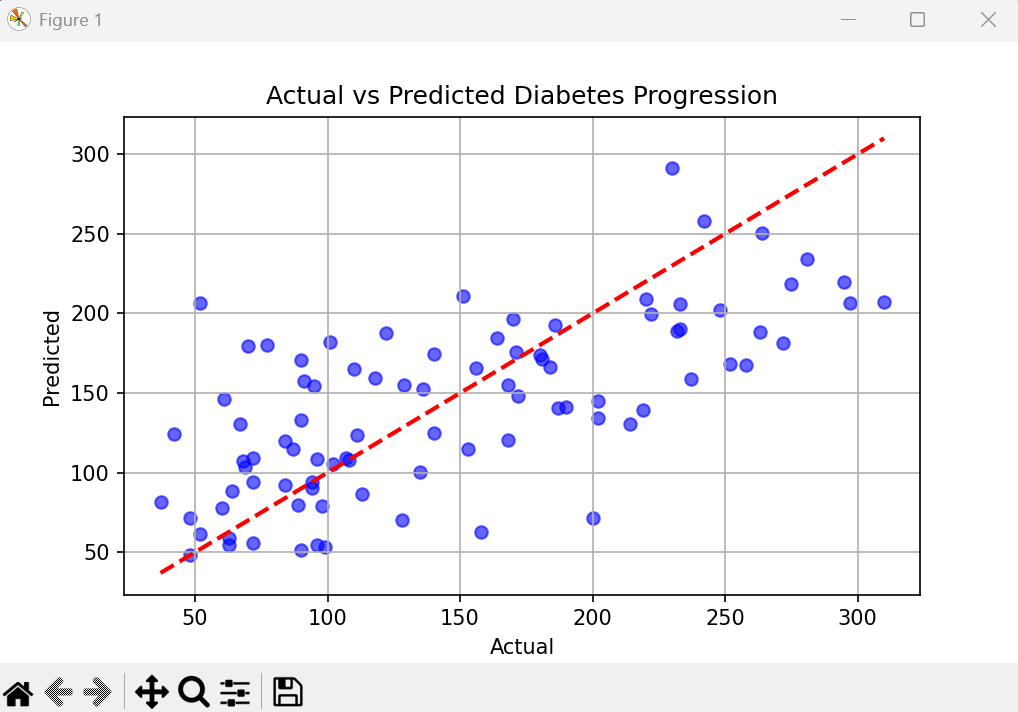
plt.title("Actual vs Predicted Diabetes Progression")

plt.grid(True)

plt.show()

**Output**

*Terminal*

*Visualization*