Q6. Project: Machine Learning

Program

```
import java.io.*;
import java.util.*;

public class LinearRegression {

  public static Double N = 0.0; // Number of values or elements.

  private static List<Double> X_Value; // First Score.
  private static List<Double> Y_Value; // Second Score.

  private static Double Sum_X = 0.0; // 'ΣX' - Sum of First Scores.
  private static Double Sum_Y = 0.0; // 'ΣY' - Sum of Second Scores.
  private static Double Sum_XY = 0.0; // 'ΣXY' - Sum of the product of first and Second Scores.
  private static Double Sum_X_Square = 0.0; // 'ΣX^2' - Sum of square First Scores.

  private static Double Slope_b = 0.0; // The slope of the regression line.
  private static Double Intercept_a = 0.0; // The intercept point of the regression line and the y axis.

  public LinearRegression(List<Double> X, List<Double> Y) {
    X_Value = X;
    Y_Value = Y;
    LinearRegressionEquationModel();
  }

  /**
   * Linear Regression Model:
   */
  public static void LinearRegressionEquationModel() {
    // 'N' - Number of values or elements.
    N = Math.round((double) X_Value.size() * 100.0) / 100.0;

    // 'ΣX' - Sum of First Scores.
    Sum_X = Math.round(X_Value.stream().mapToDouble(f -> f.doubleValue()).sum() * 100.0) / 100.0;

    // 'ΣY' - Sum of Second Scores.
    Sum_Y = Math.round(Y_Value.stream().mapToDouble(f -> f.doubleValue()).sum() * 100.0) / 100.0;

    // ΣXY and ΣX^2
    for (int i = 0; i < X_Value.size(); i++) {
      Double X = X_Value.get(i);
      Double Y = Y_Value.get(i);
      Sum_XY += X * Y; // ΣXY
      Sum_X_Square += X * X; // ΣX^2
    }

    // Sum of the product of first and Second Scores.
    Sum_XY = Math.round(Sum_XY * 100.0) / 100.0;

    // Sum of square First Scores.
    Sum_X_Square = Math.round(Sum_X_Square * 100.0) / 100.0;

    /**
     * Slope(b) = (NΣXY - (ΣX)(ΣY)) / (NΣX^2 - (ΣX)^2)
     *
     * The slope of the regression line.
     */
    Slope_b = Math.round((N * Sum_XY - (Sum_X * Sum_Y)) / (N * Sum_X_Square - (Sum_X *
Sum_X)) * 100.0) / 100.0;

    /**
     * Intercept(a) = (ΣY - b(ΣX)) / N
     *
     * The intercept point of the regression line and the y axis.
     */
    Intercept_a = (Sum_Y - (Slope_b * Sum_X)) / N;

  }

  /**
   * Prediction Regression Equation(y) = a + bx
   */
  public Double Predict(Double x) {
    return Math.round((Intercept_a + (Slope_b * x)) * 100.0) / 100.0;
  }

  public static void main(String args[]) {
    try {
      List<Double> X = new ArrayList<Double>(); // First Score.
      List<Double> Y = new ArrayList<Double>(); // Second Score.

      // Collection Data:
      BufferedReader bReader = new BufferedReader(new FileReader("input.txt"));
      String line = bReader.readLine();
      Double x, y;
      while (line != null) {
        String[] fields = line.split("\\s+");
        x = Double.valueOf(fields[0]);
        X.add(x);
        y = Double.valueOf(fields[1]);
        Y.add(y);
        // System.out.println(x + " " + y);
        line = bReader.readLine();
      }
      bReader.close();

      // Linear Regression Model Creation:
      LinearRegression lr = new LinearRegression(X, Y);

      // Prediction:
      System.out.println("Prediction: Y = " + lr.Predict(64.0));

    } catch (Exception ex) {
      System.out.println("Exception: " + ex);
```

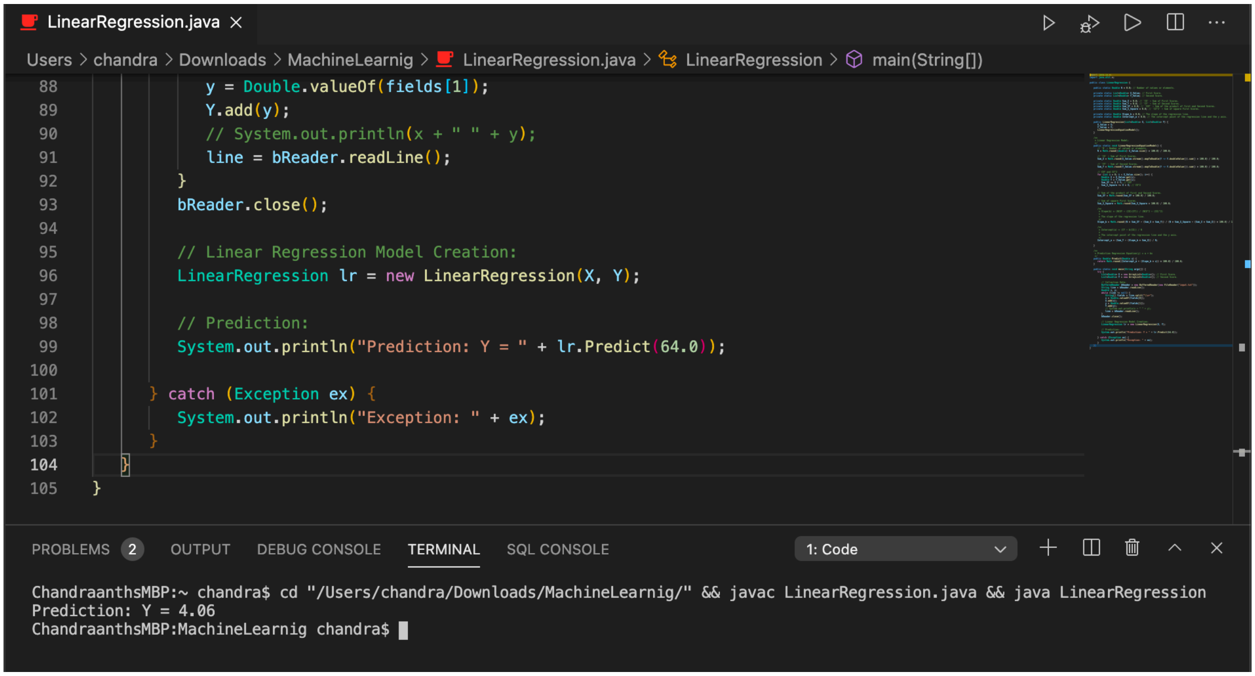
```
        }
    }
}
```

Output

Prediction: Y = 4.06

Github Link

https://github.com/HaripriyaReddy880/Machine-Learning/blob/main/Supervised%20Learning/Linear%20Regression/Project:%20Machine%20Learning%20Java


LinearRegression.java
Users > chandra > Downloads > MachineLearnig > LinearRegression.java > LinearRegression > main(String[])
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
SQL CONSOLE
1: Code

```java
                y = Double.valueOf(fields[1]);
                Y.add(y);
                // System.out.println(x + " " + y);
                line = bReader.readLine();
            }
            bReader.close();

            // Linear Regression Model Creation:
            LinearRegression lr = new LinearRegression(X, Y);

            // Prediction:
            System.out.println("Prediction: Y = " + lr.Predict(64.0));

        } catch (Exception ex) {
            System.out.println("Exception: " + ex);
        }
    }
}
```

```
ChandraanthsMBP:~ chandra$ cd "/Users/chandra/Downloads/MachineLearnig/" && javac LinearRegression.java && java LinearRegression
Prediction: Y = 4.06
ChandraanthsMBP:MachineLearnig chandra$ 
```