

Mini Project: Student Performance Prediction using Java

Date: 26-06-2025

1. Abstract

This mini project presents a Java-based application that predicts student performance using linear regression. By taking key academic factors such as study hours, attendance, and previous scores, the application estimates the expected final exam score. It showcases the application of statistical learning techniques and numerical operations in Java for education analytics.

2. Objective

To develop a simple Java console application that predicts a student's final exam score based on multiple input features using multivariate linear regression. The goal is to understand how data-driven insights can assist in educational decision-making.

3. Software Requirements

- **Programming Language:** Java (JDK 11 or above)
 - **Development Environment:** IntelliJ IDEA / Eclipse / VS Code
 - **Libraries:** None (uses only core Java)
 - **Platform:** Windows / macOS / Linux
-

4. Key Features

Multivariate Linear Regression

- Computes regression coefficients using the least squares method
- Handles multiple input variables for prediction

Student Data Input

- Uses fixed arrays to simulate training data (study hours, attendance, previous scores)
- Accepts test data for prediction

Final Score Prediction

- Computes predicted exam score for a new student input
 - Prints predicted value to the console
-

5. Java Code (Simplified, ≤ 40 Lines)

```
import java.util.Arrays;

public class StudentPerformance {

    public static void main(String[] args) {

        double[][] X = {

            {1, 60, 30}, {2, 65, 40}, {3, 70, 50},

            {4, 75, 60}, {5, 80, 65}, {6, 85, 70},

            {7, 87, 72}, {8, 90, 80}, {9, 95, 85},

            {10, 98, 90}

        };

        double[] y = {35, 45, 55, 60, 67, 73, 75, 85, 90, 95};

        int n = X.length, m = X[0].length;

        double[] meanX = new double[m];

        double meanY = Arrays.stream(y).sum() / n;

        for (int j = 0; j < m; j++)

            for (int i = 0; i < n; i++)

                meanX[j] += X[i][j];

        for (int j = 0; j < m; j++)

            meanX[j] /= n;

        double[] b = new double[m];

        for (int j = 0; j < m; j++) {

            double num = 0, den = 0;

            for (int i = 0; i < n; i++) {

                num += (X[i][j] - meanX[j]) * (y[i] - meanY);

                den += (X[i][j] - meanX[j]) * (X[i][j] - meanX[j]);

            }

        }

    }

}
```

```
    }  
    b[j] = num / den;  
}  
double intercept = meanY;  
for (int j = 0; j < m; j++)  
    intercept -= b[j] * meanX[j];  
double[] newStudent = {6, 90, 80};  
double prediction = intercept;  
for (int j = 0; j < m; j++)  
    prediction += b[j] * newStudent[j];  
System.out.println("Predicted Final Score: " + Math.round(prediction));  
}  
}
```

6. Conclusion

This project introduces the fundamentals of regression-based prediction using plain Java. It demonstrates how basic statistical modeling can be applied to student datasets to estimate performance, promoting the value of data in educational analysis. This hands-on project builds core Java logic, loop handling, and array computation skills.

7. Future Enhancements

- Accept user input from the console or a file for flexibility
- Use matrix-based regression with Apache Commons Math for scalability
- Integrate a GUI for easier input and output
- Add data visualization for trend analysis
- Train with larger datasets from real academic sources