Student Performance Prediction

1. Project Overview: Predict student math scores based on various features including demographics, educational background, and previous test scores.
2. Data Source

Data is taken from kaggle and present in csv file

1. Features:
   * 1. Gender: Categorical feature representing the student's gender.
     2. Race/Ethnicity: Categorical feature indicating the student's race/ethnicity.
     3. Parental Level of Education: Categorical feature showing the highest education level of the student's parents.
     4. Lunch: Categorical feature indicating the type of lunch (standard or free/reduced).
     5. Test Preparation Course: Categorical feature showing whether the student completed a test preparation course.
     6. Reading Score: Numerical feature representing the student’s score in reading.
     7. Writing Score: Numerical feature representing the student’s score in writing.
   1. Target Variable: Math Score
      1. Description: Numerical feature representing the student's math score, which the model aims to predict.
2. Machine Learning Model
   1. Model Used: [Linear Regression]
   2. R^2 Score: 87.6%
3. Data Specifications
   1. Data Source: CSV file.
   2. Total Records: 1000 records.
   3. Training Data: 800 records.
   4. Testing Data: 200 records.
4. Error Handling and Logging
   1. Description: Manages errors and logs important events and errors during processing.
   2. Components: Exception handling and logging modules.
5. Data Flow
6. User Input: Users enter data through the web interface.
7. Data Submission: The frontend sends data to the backend Flask server.
8. Backend Processing:
   * Data Ingestion: Reads the CSV file and splits data into training and testing datasets.
   * Data Transformation: Encodes categorical features and scales numerical features.
   * Model Training: The model is trained using the training dataset.
   * Prediction: The model makes predictions based on user input data.
9. Results: Prediction results are sent back to the frontend and displayed to the user.

8. Deployment

* Local Deployment: Local server.
* Future Deployment: Strategy for deploying the application to a cloud environment for scalability and accessibility.

9. Technology Stack

* Frontend: HTML, CSS
* Backend: Flask (Python).
* Machine Learning: Python (scikit-learn, numpy, pandas).

10. Security Considerations

* Data Protection: Ensuring user data is securely handled during transmission and processing.