Github Link: https://github.com/Thulasimathi26/Data-Science.git

Project Title: Predicting Student Performance Using Academic and Demographic Data

PHASE-2

1. Problem Statement

Predicting student academic performance is a critical challenge in the educational sector. The goal is to estimate students' final grades based on a combination of academic history, demographic background, and behavioral indicators. Early prediction of low performance can enable timely intervention by teachers, parents, and institutions to improve student outcomes.

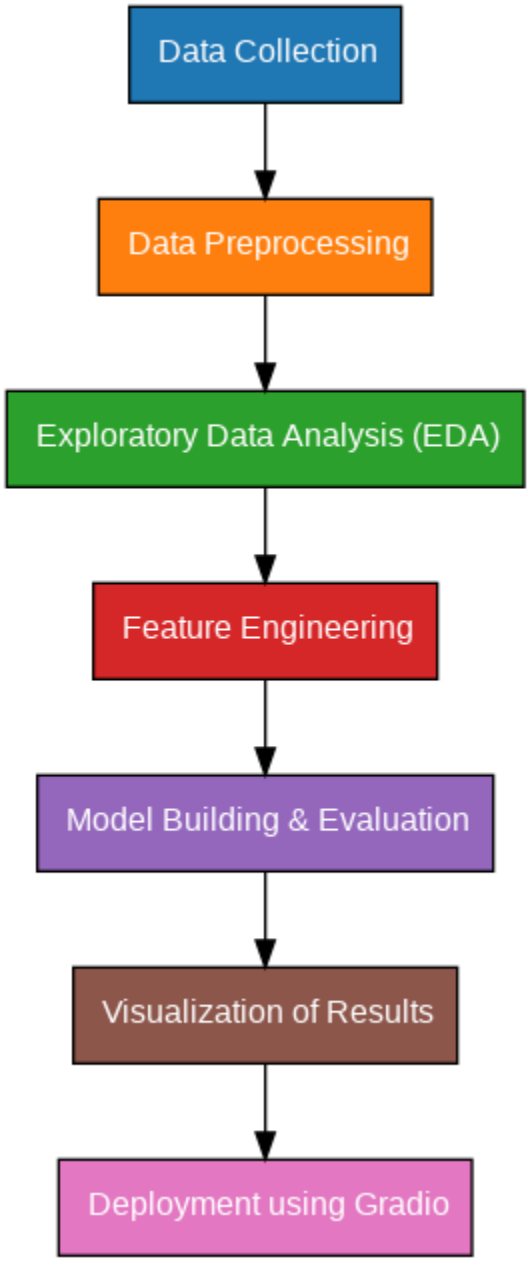
This project focuses on building a predictive model that uses real-world student data to estimate their final grade (G3). The problem type is **regression**, as the target variable (G3) is a continuous numeric score ranging from 0 to 20.

The significance of solving this problem lies in its potential application for academic advising, performance monitoring, scholarship consideration, and dropout prevention.

2. Project Objectives

- Develop a machine learning model that accurately predicts the final grade (G3) of students.
- Identify and rank the most influential features that impact academic performance.
- Provide insights into how socio-economic and behavioral variables affect learning outcomes.
- Ensure model interpretability and usability in real-world educational settings.
- Incorporate a user-friendly interface using Gradio for testing predictions.
- Evolved goal: After initial data exploration, the focus was refined to include stronger predictors like G1, G2, failures, and study time.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name**: Student Performance Data Set
- **Source**: UCI Machine Learning Repository
- **Type of Data**: Structured tabular data
- **Records and Features**: 395 student records and 33 features (numeric + categorical)
- **Target Variable**: G3 (final grade, numeric)
- **Static or Dynamic**: Static dataset

- **Attributes Covered**: Demographics (age, address, parents' education), academics (G1, G2, study time), and behavior (alcohol consumption, absences)
- Dataset Link: [Student Performance - UCI Machine Learning Repository]

5. Data Preprocessing

- Verified dataset integrity: no missing or null values.
- Removed irrelevant features with very low variance (e.g., school if only one value).
- Checked and confirmed absence of duplicate rows.
- Categorical features were one-hot encoded for machine learning.
- Applied **StandardScaler** to numerical columns to normalize them.
- Detected outliers using boxplots and z-scores; extreme outliers were investigated.

6. Exploratory Data Analysis (EDA)

- **Univariate Analysis**:
  - Histogram of G3 to understand performance distribution
  - Boxplots for variables like alcohol consumption, study time, failures
  - Count plots for categorical features (e.g., internet access, parental job)
- **Bivariate & Multivariate Analysis**:
  - Correlation matrix shows strong linear correlation between G1, G2, and G3
  - Scatter plots of G1 vs G3 and G2 vs G3 confirm positive trends
  - Grouped bar charts reveal differences in performance based on study time, failures, and support
- **Key Insights**:
  - G1 and G2 are the strongest indicators of G3
  - More study time correlates with higher G3
  - Students with more failures or absences tend to score lower

7. Feature Engineering

- Created interaction features like total_alcohol = Dalc + Walc
- Derived binary feature: higher_edu = (yes/no) from parents' education levels
- Removed highly correlated or redundant features to reduce multicollinearity
- Performed label encoding for binary features like internet, nursery
- Scaled numeric features using StandardScaler for uniformity

8. Model Building

- **Algorithms Used**:
  - Linear Regression: for baseline comparison

    - Random Forest Regressor: for capturing non-linear patterns and feature importance
- **Model Selection Rationale**:
    - Linear Regression: interpretable and fast
    - Random Forest: robust to overfitting, handles mixed data types well
- **Train-Test Split**:
    - 80% training, 20% testing
    - Used train_test_split with random_state for reproducibility
- **Evaluation Metrics**:
    - **MAE (Mean Absolute Error)**: Measures average error magnitude
    - **RMSE (Root Mean Squared Error)**: Penalizes larger errors
    - **R^2 Score**: Explains proportion of variance captured by the model

9. Visualization of Results & Model Insights

- **Feature Importance**:
    - Visualized using bar plots from Random Forest
    - G1 and G2 ranked highest in importance, followed by study time and failures
- **Model Comparison**:
    - Plotted MAE, RMSE, and R^2 for both models
    - Random Forest significantly outperformed Linear Regression in terms of RMSE
- **Residual Plots**:
    - Checked prediction errors against actual grades to ensure no major bias
- **User Testing**:
    - Integrated model into a Gradio interface to test predictions by inputting feature values

10. Tools and Technologies Used

- **Programming Language**: Python 3
- **Notebook Environment**: Google Colab
- **Key Libraries**:
    - pandas, numpy for data handling
    - matplotlib, seaborn, plotly for visualizations
    - scikit-learn for preprocessing and modeling
    - Gradio for interface deployment

11. Team Members and Contributions

[List names and responsibilities.

- *Clearly mention who worked on:*
    - *Data cleaning*
    - *EDA*
    - *Feature engineering*
    - *Model development*
    - *Documentation and reporting]*