

AI-201 Lab Final Project Report
Student Score Analyzer & Grade Predictor
Project Members
Batool Binte Fazal (2024140)
Rida Syed (2024540)

1. Introduction

This report presents the final implementation of the **Student Score Analyzer & Grade Predictor**, developed as part of the AI-201 Lab. The project is an enhanced, intelligent, and modular web-based system built using Python, Object-Oriented Programming (OOP), and Machine Learning techniques.

The system allows users to upload student performance datasets, perform automated data cleaning and feature engineering, visualize academic patterns, train multiple machine learning models, and generate predictions such as scores, grades, pass/fail status, and risk levels. A Streamlit-based interface makes the application interactive and user-friendly.

2. Project Objectives

The final system successfully fulfills the objectives outlined in the proposal:

- Upload and validate student performance datasets
- Clean and preprocess data automatically with exception handling
- Engineer meaningful academic and behavioral features using NumPy
- Perform exploratory data analysis using Pandas
- Visualize patterns using Matplotlib
- Train and evaluate multiple ML models
- Predict student performance and risk levels
- Save and reload trained models using Pickle
- Provide an interactive Streamlit dashboard

3. Dataset Description

- **Source:** Kaggle – Students Performance in Exams
- **Size:** ~1000 records
- **Features:**
 - Demographic: gender, race/ethnicity, parental level of education
 - Behavioral: study time, test preparation course
 - Academic: math score, reading score, writing score

This dataset is well-suited for educational analytics and predictive modeling.

4. Tools & Technologies Used

Programming Language

- Python 3

Libraries

- **Pandas:** data loading, cleaning, grouping, and analysis
- **NumPy:** numerical computations, feature engineering, normalization, outlier detection
- **Matplotlib:** all visualizations (histograms, bar charts, pie charts, 3D plots)
- **Scikit-learn:** machine learning models and evaluation
- **Streamlit:** interactive web application
- **Pickle:** model persistence

5. System Architecture & OOP Design

The project follows a clean Object-Oriented Programming structure. Each major task is handled by a dedicated class:

5.1 DataCleaner

- Removes null values and duplicates
- Validates required columns (math, reading, writing scores)
- Encodes categorical variables using LabelEncoder
- Includes exception handling for invalid datasets

5.2 FeatureEngineer

Uses NumPy extensively to create advanced features:

- Average Score
- Study Efficiency and Study Engagement
- Weighted Performance Index:

$$PI = 0.5(\text{math}) + 0.3(\text{reading}) + 0.2(\text{writing})$$

- Polynomial interaction features
- Z-score based outlier detection
- Percentile ranks

- Risk score and risk category assignment

5.3 Visualizer

Generates all plots using Matplotlib:

- Histogram with gradient coloring
- Subject-wise bar chart with error bars
- Donut pie chart with demographic distribution
- Advanced 3D scatter plot (study time \times parental education \times score)

5.4 PandasAnalyzer

- Answers research questions using groupby, agg, and correlation analysis
- Performs demographic and performance comparisons

5.5 ModelTrainer

Trains and evaluates three ML models:

- **Linear Regression:** score prediction (RMSE, R^2)
- **Logistic Regression:** pass/fail classification (Accuracy)
- **KNN Classifier:** risk level classification (Accuracy)

Features are scaled using StandardScaler before training.

5.6 Predictor

- Generates real-time predictions
- Predicts score, grade, pass/fail status, and risk level

5.7 ModelPersistence

- Saves trained models, scaler, and feature names using Pickle
- Loads saved models to avoid retraining

6. Data Preprocessing

The preprocessing pipeline includes:

- Validation of required columns
- Removal of missing and duplicate records
- Encoding of categorical variables

- Feature scaling before model training

Robust exception handling ensures system stability.

7. Feature Engineering

Feature engineering significantly enhances analytical depth:

- Average academic performance
- Behavioral efficiency metrics
- Weighted performance index
- Outlier flags using Z-scores
- Percentile-based ranking
- Custom risk scoring system

All major computations are performed using NumPy.

8. Data Visualization

The following visualizations are generated:

- Score distribution histogram
- Subject-wise performance comparison bar chart
- Demographic distribution (donut + bar view)
- 3D performance analysis plot

These plots help in understanding academic trends and relationships.

9. Machine Learning Models & Evaluation

Model	Task	Metric
Linear Regression	Score Prediction	RMSE, R ²
Logistic Regression	Pass/Fail	Accuracy
KNN Classifier	Risk Classification	Accuracy

The models are trained on engineered features and evaluated on test data.

10. Streamlit Application

The Streamlit app provides:

- CSV upload interface
- Step-by-step analysis workflow
- Interactive plots
- Model training buttons
- Model saving/loading
- Real-time prediction form

11. Exception Handling

The system safely handles:

- Missing or invalid files
- Incorrect column names
- Model training failures
- Prediction errors

12. Expected vs Achieved Outcomes

All expected outcomes mentioned in the proposal were successfully achieved, including:

- Complete ML pipeline
- Interactive dashboard
- Model persistence
- Advanced analytics

13. Learning Outcomes

Through this project, we gained hands-on experience in:

- Object-Oriented Programming
- Data preprocessing and feature engineering

- NumPy-based numerical analysis
- Pandas-based data analysis
- Matplotlib visualization
- Machine learning model training and evaluation
- Streamlit app development
- Exception handling and debugging

14. Conclusion

The **Student Score Analyzer & Grade Predictor** is a complete, modular, and production-style AI system that meets all AI-201 Lab requirements. It demonstrates strong integration of data science, machine learning, visualization, and software engineering concepts, making it a successful and comprehensive semester project.