Intelligent Student Performance Dashboard

(Full Project Documentation, Implementation & Presentation Content)

---

1. Documentation

Project Overview

The **Intelligent Student Performance Dashboard** is an **AI-powered interactive analytics platform** built using **Streamlit**.
It is designed to **monitor, analyze, and predict** students' academic performance by integrating:

- Machine Learning models
- Data visualization
- What-if analysis tools

It assists teachers and educational decision-makers in understanding learning behavior, identifying trends, and predicting outcomes accurately.

Technologies Used

Category	Tools / Libraries
Programming Language	Python 3.11
Framework	Streamlit
Visualization	Plotly, Plotly Express
Machine Learning	Scikit-learn, NumPy, Pandas
Model Storage	Pickle (student_multi_model.pkl)
Label Mapping	JSON (label_mappings.json)

Dashboard Architecture

The dashboard is divided into **five main tabs**, each providing a unique analytical perspective:

Tab	Function
1. Overview	Displays general statistics on performance & attendance
2. Subjects	Shows subject-level comparisons (Math, Reading, Writing)
3. Attendance	Analyzes attendance distribution & effect on scores
4. Insights	Presents summarized analytics & narrative insights
5. AI Prediction (What-If Analysis)	Allows users to modify variables and get instant ML predictions

---

Data Overview

Dataset: DATA.csv
Includes both academic and behavioral attributes.

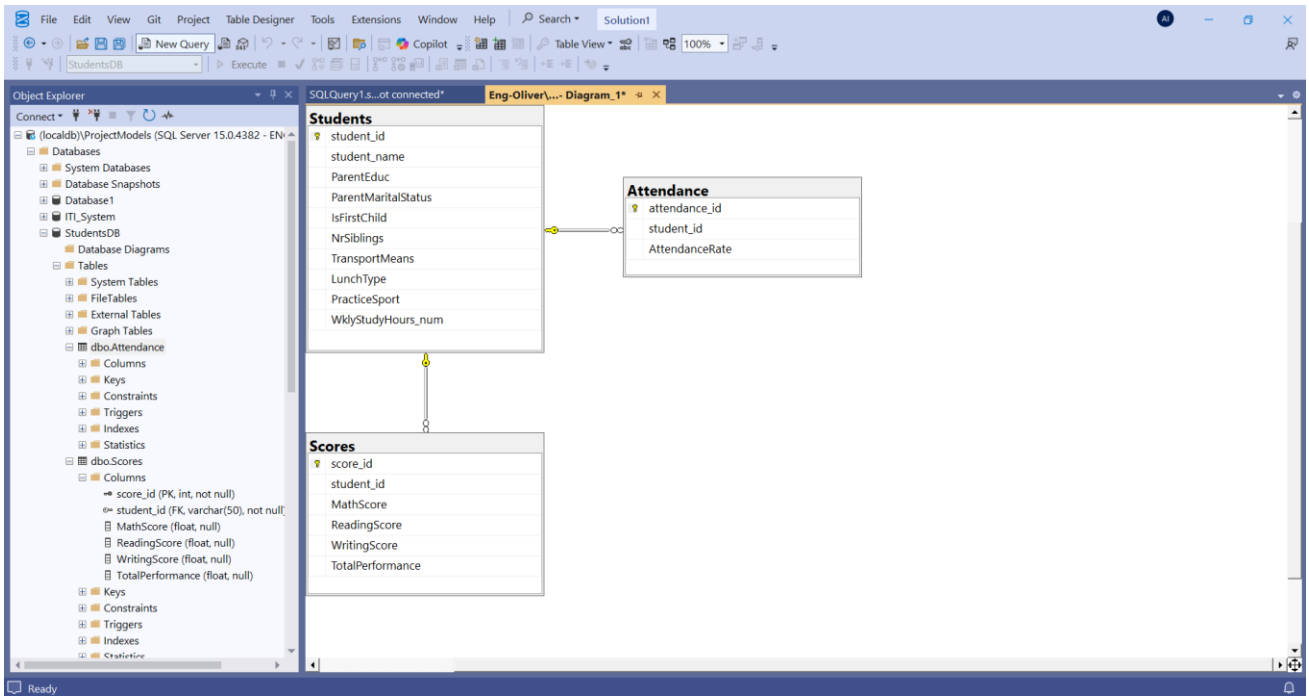
Type	Example Features
Academic	Math, Reading, Writing, AcademicIndex, OverallPerformance
Behavioral	AttendanceRate, StudyHours, SleepHours, BehaviorIndex, SocialIndex

Socio-demographic Parent Education, Marital Status, Transport Means

Preprocessing Steps:

- Removed non-relevant identifiers (student_id, student_name)
- Encoded categorical variables using LabelEncoder
- Split data for model training/testing

ER Diagram:



Machine Learning Model

Property	**Description**
Type	MultiOutputRegressor(RandomForestRegressor)
Purpose	Predict both *Academic Average* & *Overall Performance*

n_estimators	150
max_depth	12
random_state	42

Performance Metrics:

Metric	Academic Average	Overall Performance
R^2 Score	1.000	0.971
MAE	0.166	1.551

Excellent accuracy achieved due to Random Forest's robustness and high-quality data.

Model files:

- student_multi_model.pkl
- label_mappings.json

Dashboard Functionalities

Tab 1 – Overview:

- Shows key KPIs (total students, avg. scores, attendance rates).
- Displays best/worst performers using metrics and visuals.

Tab 2 – Subjects:

- Compares each student's performance across subjects.
- Visualizes performance gaps using bar charts.

Tab 3 – Attendance:

- Displays histograms showing attendance impact on performance.

Tab 4 – Insights:

- Summarizes averages (study hours, attendance, grades).
- Highlights top and low performers.
- Smart narrative summary explaining findings.

Tab 5 – AI Prediction (What-If):

- Accepts user inputs (Math, Reading, Sleep Hours, etc.).
- Encodes automatically using mappings.
- Predicts *Academic Average* and *Overall Performance*.
- Displays results dynamically with visuals (progress bars, metrics, balloons).

Prediction Workflow

1. User inputs or selects attributes.
2. System encodes categorical variables.
3. Model predicts both targets simultaneously.
4. Dashboard displays metrics + visual output instantly.

File & Folder Structure

```
DEPI-Project/
|
├── DATA.csv                  # Dataset
├── student_multi_model.pkl       # Trained ML model
├── label_mappings.json          # Encoded labels
├── app_pro.py                # Streamlit dashboard app
├── model_training.py           # Model training script
└── Project Report Dashboard.pdf  # Final report
```

Deployment and Access

Run locally:

streamlit run app_pro.py

For online sharing:

pip install pyngrok

ngrok config add-authtoken <your_token>

streamlit run app_pro.py

ngrok http 8501

Generated **public link** allows others to access the dashboard online.

Key Insights

- Strong correlation between **attendance** and **academic success**.

- Writing & reading proficiency directly influence total performance.
- AI predictions show minimal error (high accuracy).
- Enables **early identification** of underperforming students.

Future Improvements

- Add model explainability (SHAP, LIME).
- Integrate live student databases.
- Implement clustering by performance level.
- Deploy to Streamlit Cloud / Render for public access.

Implementation Summary

GitHub Repository:
https://github.com/Eng-Oliver/DEPI-Project

Execution Steps:

pip install -r requirements.txt

python model_training.py

streamlit run app_pro.py

Tools: Python | Streamlit | Plotly | PostgreSQL | Scikit-learn | NumPy | Pandas

Output: Interactive web dashboard with AI-powered analytics and prediction.

3.Content

1 — Title

Intelligent Student Performance Dashboard
A Smart AI-Based Academic Analytics System

---

2 — Problem

- Schools lack automated systems to analyze student performance efficiently.
- Manual analysis delays interventions for struggling students.

---

3 — Solution

- Build an intelligent, AI-powered dashboard.
- Visualize performance patterns and predict academic outcomes.

---

4 — System Architecture

- Python + Streamlit frontend
- ML backend (RandomForestRegressor)
- CSV dataset as data source
- Local/online deployment using ngrok

---

5 — Dashboard Tabs

1. Overview
2. Subjects
3. Attendance
4. Insights
5. AI Prediction

---

6 — Machine Learning Model

Metric	Academic	Performance
R^2	1.000	0.971
MAE	0.166	1.551

---

7 — Key Findings

- Attendance & study habits are major predictors.
- Writing and reading correlate strongly with success.
- AI provides accurate, actionable insights.

---

8 — Future Scope

- Add explainable AI features.
- Deploy on Streamlit Cloud.
- Expand to multi-school datasets.

Conclusion

The project demonstrates how **AI and visualization** can enhance educational data analytics. It transforms static data into actionable insights for educators, making performance tracking **smarter, faster, and data-driven**.

---

9 — References

- GitHub Repository: https://github.com/Eng-Oliver/DEPI-Project
- Tools: Python, Streamlit, Plotly, scikit-learn
- Author: **Abdullah Ahmed Hussien**