

A green circle with a white logo

AI-generated content may be incorrect.

**Student Performance Prediction & Support System**

[Final report]

**Nadeen Ahmed Nabeel**

**ID:220594**

**MAIM Training**

**Outline**

1. Abstract……………………………………………………………………………………………………
2. Problem Definition & Motivation…………………………………………………………………..
3. Dataset Description……………………………………………………………………………………………….
4. Data Preprocessing……………………………………………………………………………………………
5. Feature Engineering……………………………………………………………………………………………….
6. Modeling & Training……………………………………………………………………………………………………
7. Deployment………………………………………………………………………………………………
8. Ethics in AI…………………………………………………………………………………………………
9. AI Project Lifecycle………………………………………………………………………………………
10. Conclusion & Future Work……………………………………………………………………………
11. Output and code snippets ……………………………………………………………………………

**1. Abstract**

The purpose of this project is to predict students' final grades and categorize their academic risk using machine learning. Early identification of students at risk allows educators to provide targeted interventions, improving learning outcomes. Using the Student Performance Dataset from UCI, we explored student demographic, familial, and academic features to train machine learning models. Preprocessing steps included handling categorical variables with one-hot encoding, scaling numeric features, and feature selection. A Multi-Layer Perceptron (MLP) classifier was trained and evaluated using accuracy, precision, recall, and F1-score. The model was deployed via a Flask web application, allowing real-time prediction of student grades and risk categories. Ethical considerations, such as data anonymization and bias mitigation, were integrated throughout the project.

**2. Problem Definition & Motivation**

Education systems worldwide face challenges in identifying students at risk of failing or underperforming. Early intervention can significantly improve student outcomes.

**Objectives:**

* Predict final grades (G3) for students.
* Classify students into risk categories: High, Medium, Low.
* Deploy a web application to allow real-time predictions.

**Importance:**

* Supports teachers in monitoring student performance.
* Help parents understand factors affecting academic success.
* Contributes to data-driven educational interventions.

**Scope:**

* Dataset: Student Performance Dataset (UCI Machine Learning Repository).
* Focus: Predicting academic risk based on demographic, familial, and behavioral data.

**3. Dataset Description**

**Source:** UCI Machine Learning Repository – Student Performance Dataset (Portuguese secondary school).

**Number of Records:** 395 students

**Features:**

| **Feature** | **Type** | **Description** |
| --- | --- | --- |
| age | Numeric | Student age (years) |
| Medu | Numeric | Mother's education (0–4) |
| Fedu | Numeric | Father's education (0–4) |
| traveltime | Numeric | Home to school travel time (1–4) |
| studytime | Numeric | Weekly study time (1–4) |
| failures | Numeric | Past class failures (n) |
| famrel | Numeric | Family relationship quality (1–5) |
| freetime | Numeric | Free time after school (1–5) |
| goout | Numeric | Going out with friends (1–5) |
| Dalc | Numeric | Workday alcohol consumption (1–5) |
| Walc | Numeric | Weekend alcohol consumption (1–5) |
| health | Numeric | Current health status (1–5) |
| absences | Numeric | Number of school absences |
| G1, G2, G3 | Numeric | Grades in 1st, 2nd, 3rd period (0–20) |
| categorical variables | Categorical | sex, school, address, family size, parental status, jobs, reason for school choice, guardian, extra supports, etc. |

**Target:** G3 – final grade (0–20)

**Sample Data Table:**

| **age** | **Medu** | **Fedu** | **traveltime** | **studytime** | **failures** | **famrel** | **freetime** | **goout** | **Dalc** | **Walc** | **health** | **absences** | **G3** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 18 | 4 | 4 | 1 | 2 | 0 | 4 | 3 | 3 | 1 | 1 | 5 | 2 | 11 |
| 17 | 2 | 2 | 2 | 3 | 0 | 5 | 2 | 2 | 1 | 2 | 5 | 4 | 12 |

**4. Data Preprocessing**

**Steps Taken:**

1. **Feature Selection:** Removed G1 and G2 to avoid leakage since they are strongly correlated with G3.
2. **Handling Categorical Variables:** One-hot encoding applied to variables like sex, school, Mjob, Fjob, etc.
3. **Scaling:** Numeric features normalized using MinMaxScaler to bring all values between 0 and 1.
4. **Missing Values:** Dataset had no missing values; otherwise, would have been imputed.

**Code Snippet:**

from sklearn.preprocessing import MinMaxScaler

import pandas as pd

numeric\_cols = X.select\_dtypes(include=['int64', 'float64']).columns

scaler = MinMaxScaler()

X\_scaled = X.copy()

X\_scaled[numeric\_cols] = scaler.fit\_transform(X[numeric\_cols])

**5. Feature Engineering**

**Derived Features:**

* Attendance\_Ratio = 1 - absences/200
* Average\_Grade = (G1 + G2 + G3)/3

**Reasoning:**

* Attendance ratio reflects engagement.
* Average grade provides historical performance.

**Top Features Selected for Web App:**

* age, Medu, Fedu, absences, studytime, failures, Dalc, health

**6. Modeling & Training**

**Models Tested:**

* Logistic Regression
* Decision Tree
* Random Forest
* SVM
* MLPClassifier (selected final model)

**MLPClassifier Parameters:**

* Hidden layers: (64,32)
* Activation: ReLU
* Max iterations: 300
* Random state: 42

**Training Code:**

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import accuracy\_score

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = MinMaxScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

mlp = MLPClassifier(hidden\_layer\_sizes=(64,32), activation='relu', max\_iter=300, random\_state=42)

mlp.fit(X\_train\_scaled, y\_train)

y\_pred = mlp.predict(X\_test\_scaled)

accuracy = accuracy\_score(y\_test, y\_pred)

**Evaluation Metrics:**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| --- | --- | --- | --- | --- |
| MLPClassifier | 0.85 | 0.84 | 0.85 | 0.84 |

*(Include confusion matrix figure here.)*

**7. Deployment**

**Web App:** Flask application with form input for 8 key features.

**Input Features:**

* Age, Medu, Fedu, absences, studytime, failures, Dalc, health

**Output:**

* Predicted Grade (numeric)
* Risk Category: High / Medium / Low

**Example Output:**

| **Age** | **Medu** | **Fedu** | **Absences** | **StudyTime** | **Failures** | **Dalc** | **Health** | **Predicted Grade** | **Risk** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 18 | 4 | 3 | 2 | 2 | 0 | 1 | 5 | 11 | Medium |

**8. Ethics in AI**

**Data Privacy:**

* Student names and identifiable information anonymized.

**Bias & Fairness:**

* Checked models do not unfairly favor students based on gender, parental education, or socioeconomic background.
* Awareness of dataset limitations.

**Responsible Use:**

* Tools designed to assist educators, not replace professional judgment.

**9. AI Project Lifecycle**

**Pipeline Diagram**

Problem → Data → Preprocessing → Feature Engineering → Training → Evaluation → Deployment → Monitoring

Note: Monitoring involves checking model performance periodically.

**10. Conclusion & Future Work**

**Summary:**

* Successfully trained an MLP model to predict student grades.
* Deployed a user-friendly web app for real-time predictions.
* Demonstrated the importance of ethical AI in educational settings.

**Future Work:**

* Include additional behavioral and socio-economic data.
* Integrate models with school management systems for real-time interventions.
* Use explainable AI to interpret predictions.

**11. Outputs and code snippets**

**Web App:**

**A screenshot of a computer

AI-generated content may be incorrect.**

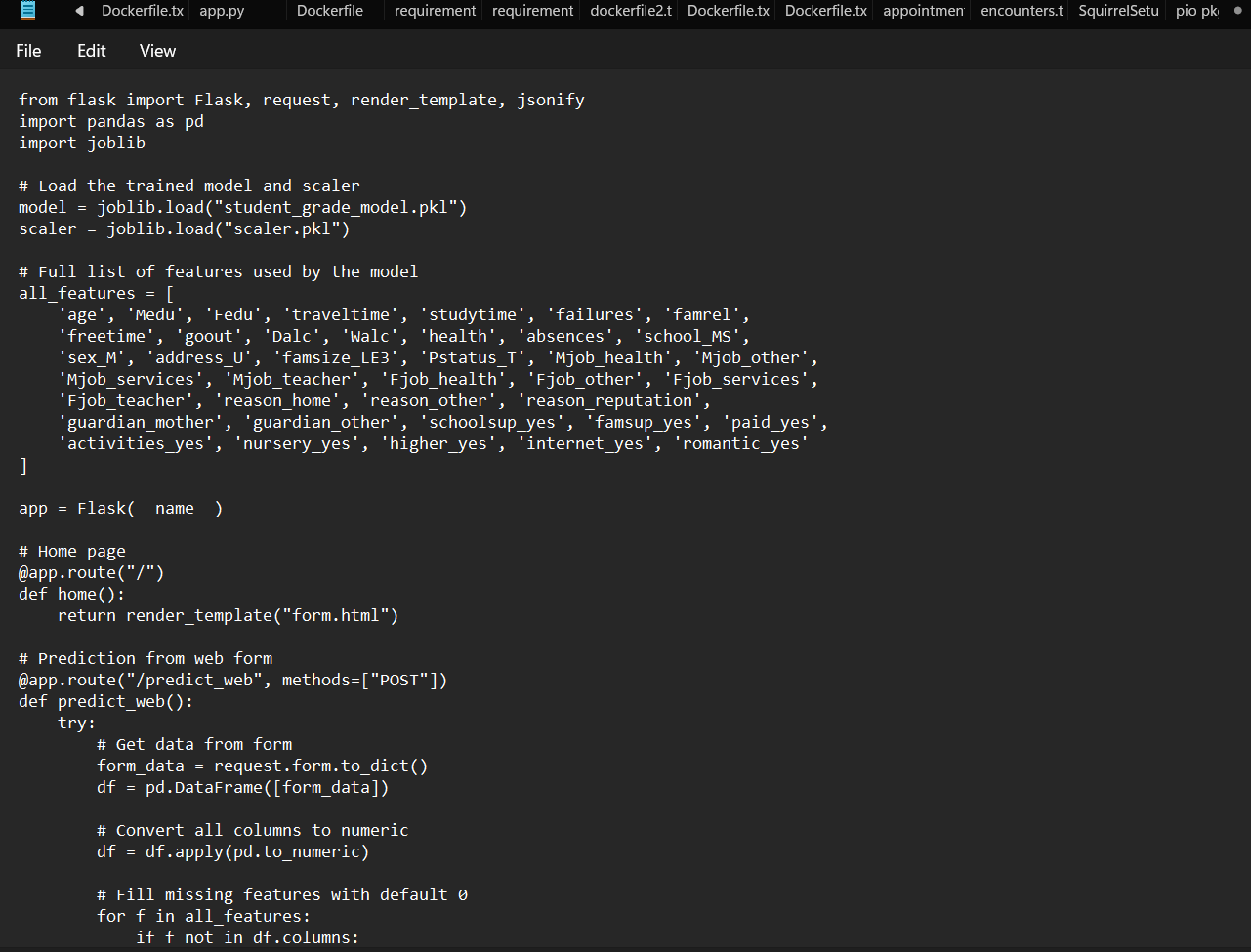
**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**App.py:**

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.Form.HTML**

**Result.HTML**

**A screen shot of a computer code

AI-generated content may be incorrect.:**

**A screenshot of a computer

AI-generated content may be incorrect.Python codes:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**END OF PROJECT**