Student Performance Analysis Project

📋 Project Overview

Real-World Problem

Universities need **early, data-driven signals** about students who may underperform. Using real student records, you will:

- Clean and transform data
- Explore drivers of performance
- Segment students (unsupervised learning)
- Predict risk (supervised learning)
- Turn insights into concrete recommendations

📊 Dataset Information

Source & License

- **Repository**: UCI ML Repository - "Student Performance"
- **Size**: 649 rows, 30+ features
- **Subjects**: Mathematics and Portuguese
- **Features**: Demographics, study time, absences, grades (G1/G2/G3)
- **License**: CC BY 4.0
- **URL**: https://archive.ics.uci.edu/dataset/320/student+performance

Download Options

Option A: Python (Recommended)

```
pip install ucimlrepo
from ucimlrepo import fetch_ucirepo
ds = fetch_ucirepo(id=320)
X = ds.data.features
y = ds.data.targets
```

Option B: Manual Download

- Download zip from UCI website
- Contains: `student-mat.csv` and `student-por.csv`
- Both files have identical schema

🎯 Required Tasks & Deliverables

A) Data Preparation

Document every decision

- ✅ Load dataset (student-mat.csv, student-por.csv, or merged view)
- ✅ Validate schema & data types
- ✅ Check for duplicates
- ✅ Assess and handle missing values and outliers (justify methods)
- ✅ Write **Data Quality Report**

B) Data Transformation

- **Encoding**: One-hot encode categoricals (school, sex, address, Mjob)
- **Scaling**: Standardize numeric features for ML and K-Means
- **Feature Engineering**:
  - Attendance proxy from absences
  - Average of G1–G3
  - Binary target: pass = G3≥10 or 3-tier risk
- **Data Leakage**: Create two variants:
  - (i) **With G1/G2** when predicting G3
  - (ii) **Without G1/G2** when predicting G3
  - Compare results and discuss trade-offs

C) Exploratory Data Analysis (EDA)

- ✅ Descriptive statistics table for key features
- ✅ Correlation analysis (identify strongest relations with G3)
- ✅ Group comparisons (studytime, failures, schoolsup vs outcomes)
- ✅ **3–5 testable hypotheses** stated and addressed

D) Visualization Requirements

Minimum required figures (labeled and readable):

- ✅ Histograms of 3+ numeric variables
- ✅ Boxplot/violin of G3 across studytime or schoolsup
- ✅ Scatter plot (e.g., absences vs G3) with interpretation
- ✅ Correlation heatmap of numeric features

E) Unsupervised Learning (K-Means)

- ✅ Feature set for behavior segmentation:
  - studytime, absences, goout, freetime, famsup, schoolsup
- ✅ Select optimal k using elbow method and silhouette analysis
- ✅ Profile clusters (size, centroids, typical behaviors)
- ✅ Compare average G3 (or pass rate) across clusters
- ✅ Interpret implications

F) Supervised Learning

- ✅ Define target: binary pass/fail or 3-class risk
- ✅ Train **at least 3 algorithms**:

    - Logistic Regression
    - Decision Tree/Random Forest
    - Support Vector Machine (SVM)
- ✅ Use hold-out and 5-fold cross-validation
- ✅ Perform basic hyperparameter tuning
- ✅ Report full metrics:
    - Accuracy, Precision, Recall, F1
    - ROC-AUC (for binary classification)
- ✅ Interpret models (feature importances/coefficients)

G) Model Evaluation & Comparison

- ✅ Summarize performance across models
- ✅ Compare data-leakage variants (with/without G1/G2)
- ✅ Discuss overfitting/underfitting and generalization

H) Storytelling & Recommendations

- ✅ **5–8 actionable insights** tied to specific actions
    - Example: "*High absences + ≥2 failures = X× failure odds → propose attendance intervention + early tutoring*"
- ✅ **Ethical considerations**:
    - Privacy protection
    - Fairness and bias mitigation
    - Sensitive attributes handling

📦 Submission Requirements

1. Jupyter Notebooks

Sequential analysis with clear headings, comments, and results:

- `01_data_preparation.ipynb` - Cleaning & Transformation
- `02_eda_visualization.ipynb` - EDA & Visualization
- `03_unsupervised_learning.ipynb` - K-Means Clustering
- `04_supervised_learning.ipynb` - Classification Models
- `05_model_evaluation.ipynb` - Evaluation & Recommendations

2. Technical Report (10–15 pages)

Required sections:

- Abstract
- Problem Statement & Value Proposition
- Dataset (source, schema, limitations)
- Methodology
- Results & Analysis
- Ethics & Considerations

- Recommendations
- Limitations & Future Work

3. Slide Deck (10–12 slides)

- Key charts and visualizations
- Main findings and decisions
- Actionable recommendations

4. Reproducibility

- ✅ `requirements.txt` (dependencies)
- ✅ `README.md` (setup and run instructions)
- ✅ Clean, documented code

🚀 Success Criteria

Technical Excellence

- ✅ Proper data handling and preprocessing
- ✅ Appropriate ML techniques and evaluation
- ✅ Valid statistical analysis and hypothesis testing
- ✅ Clear visualizations with interpretations

Business Impact

- ✅ Actionable insights for university stakeholders
- ✅ Evidence-based recommendations
- ✅ Ethical considerations addressed
- ✅ Clear communication of findings

Academic Rigor

- ✅ Methodological transparency
- ✅ Proper documentation and reproducibility
- ✅ Critical analysis of limitations
- ✅ Professional presentation quality