STUDENT PERFORMANCE PREDICTION USING

Machine Learning & Generative AI

Team members:
Joudy Alnounou - 444200139
Showq Alhadlaq - 444204111
Najd Albabtain - 444201076
Duna Alnujaidi - 444200589
Hessa Alhozaimy - 444200799

SWE485 - Selected Topics in Software Engineering
Artificial Intelligence
December 2025

Problem Statement

Many students fail without early warning.
Teachers and advisors don’t always know who needs help or when to intervene.

Objective

- *Predicts if a student will pass or fail*
- *Groups students into profiles based on their behavior*
- *Provides AI-generated advice based on each student's data*

Business Value

- *Helps identify struggling students early*
- *Gives advisors clear, data-based insights*
- *Saves time by generating automatic academic advice*
- *Can improve student success and reduce failure rates*

Data Overview

Dataset Source: Kaggle – Student Performance Prediction
Initial Size: 40,000 student records
Final Size: 30,986 records after cleaning
Target Variable: Pass / Fail (Balanced 50% – 50%)

Key Features:

- Study Hours per Week
- Attendance Rate
- Previous Grades
- Outside School Activities (Yes/No)

Preprocessing Steps:

- Removed missing values (22.5%)
- Dropped low-impact features
- Min-Max Normalization (0–1)

Process/Approach

Machine Learning Pipeline

Phase 1: Data Preparation

- Cleaning, Feature Selection, Normalization
- Exploratory Data Analysis (EDA)

Phase 2: Supervised Learning

- Models: Logistic Regression, Random Forest, Extra Trees
- Train/Test Split: 80% / 20%
- Evaluation: Accuracy, F1-score, ROC-AUC

Phase 3: Unsupervised Learning

- K-Means Clustering (k = 2)
- PCA for visualization

Phase 4: Generative AI

- Personalized academic advice
- Automated performance reports

Tools

Programming Environment:

- Python, Jupyter Notebook

Data Processing:

- Pandas (data manipulation)
- NumPy (numerical operations)

Visualization:

- Matplotlib (plotting)
- Seaborn (statistical graphics)

Machine Learning:

- Scikit-learn:
  - Models: Logistic Regression, Random Forest, Extra Trees, K-Means
  - Evaluation: accuracy, precision, recall, F1-score, ROC-AUC, Silhouette Score
  - Dimensionality Reduction: PCA

Generative AI:

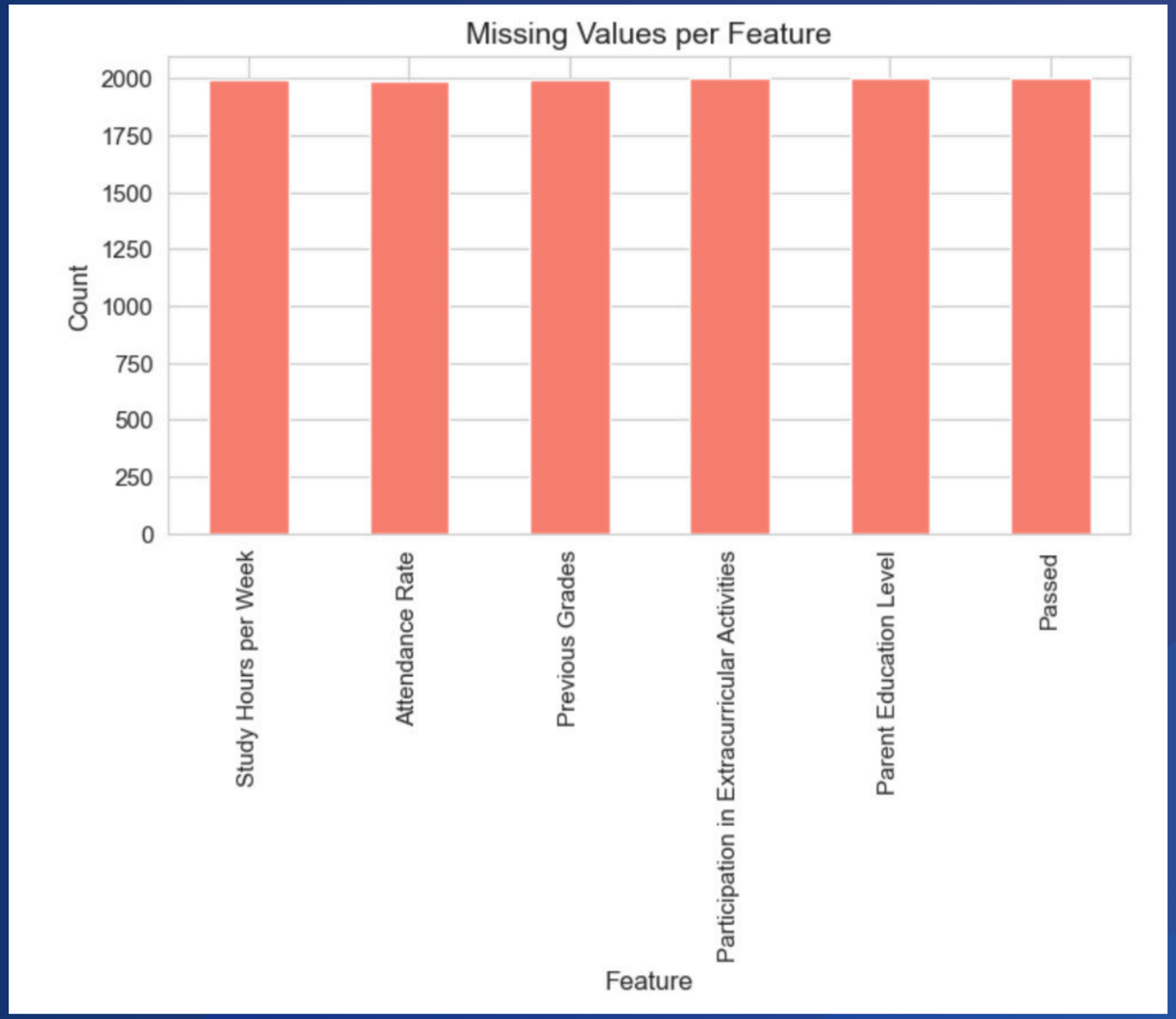
- OpenAI API (GPT-4o-mini)

Data Storage:

- CSV files for dataset management

PREPROCESSING

Missing Value Analysis



PREPROCESSING

Missing Value Analysis

Before:

	Missing Values	Percentage (%)
Student ID	0	0.00
Study Hours per Week	1995	4.99
Attendance Rate	1992	4.98
Previous Grades	1994	4.98
Participation in Extracurricular Activities	2000	5.00
Parent Education Level	2000	5.00
Passed	2000	5.00

After:

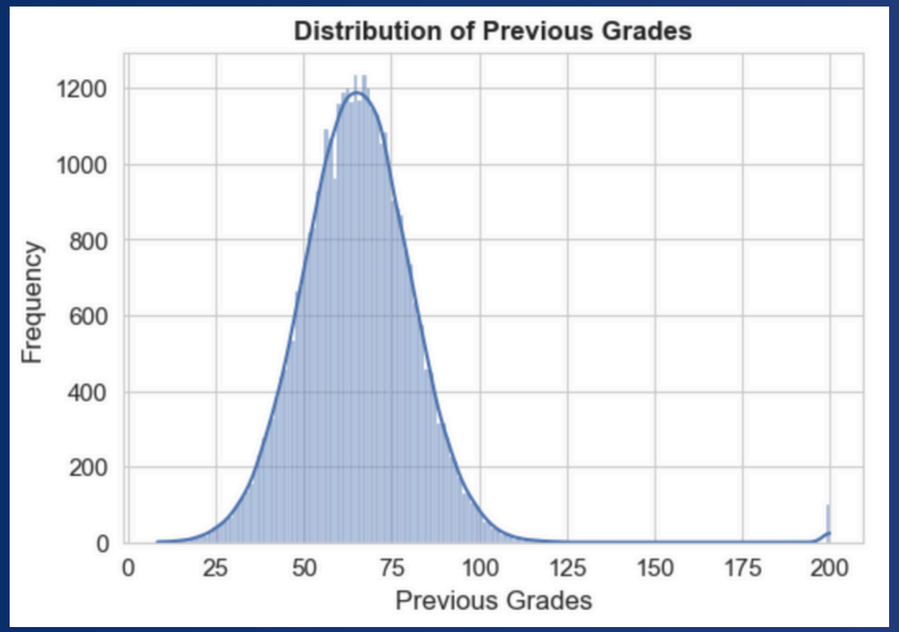
	Missing Values	Percentage (%)
Student ID	0	0.0
Study Hours per Week	0	0.0
Attendance Rate	0	0.0
Previous Grades	0	0.0
Participation in Extracurricular Activities	0	0.0
Passed	0	0.0

- Data integrity: Every row has complete information for all features
- Target variable “Passed” missing made imputation inappropriate, and so chose deletion

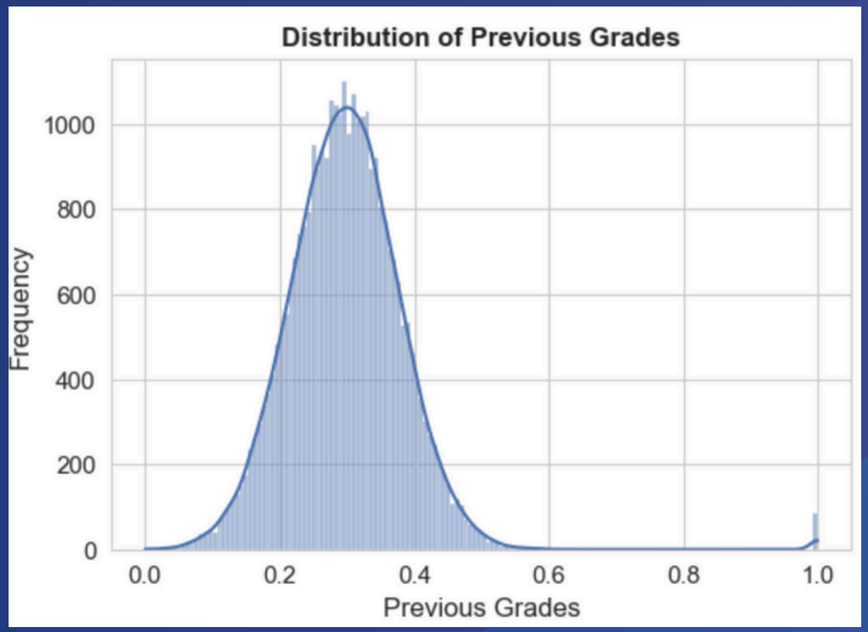
PREPROCESSING

Min-Max Normalization

Before:



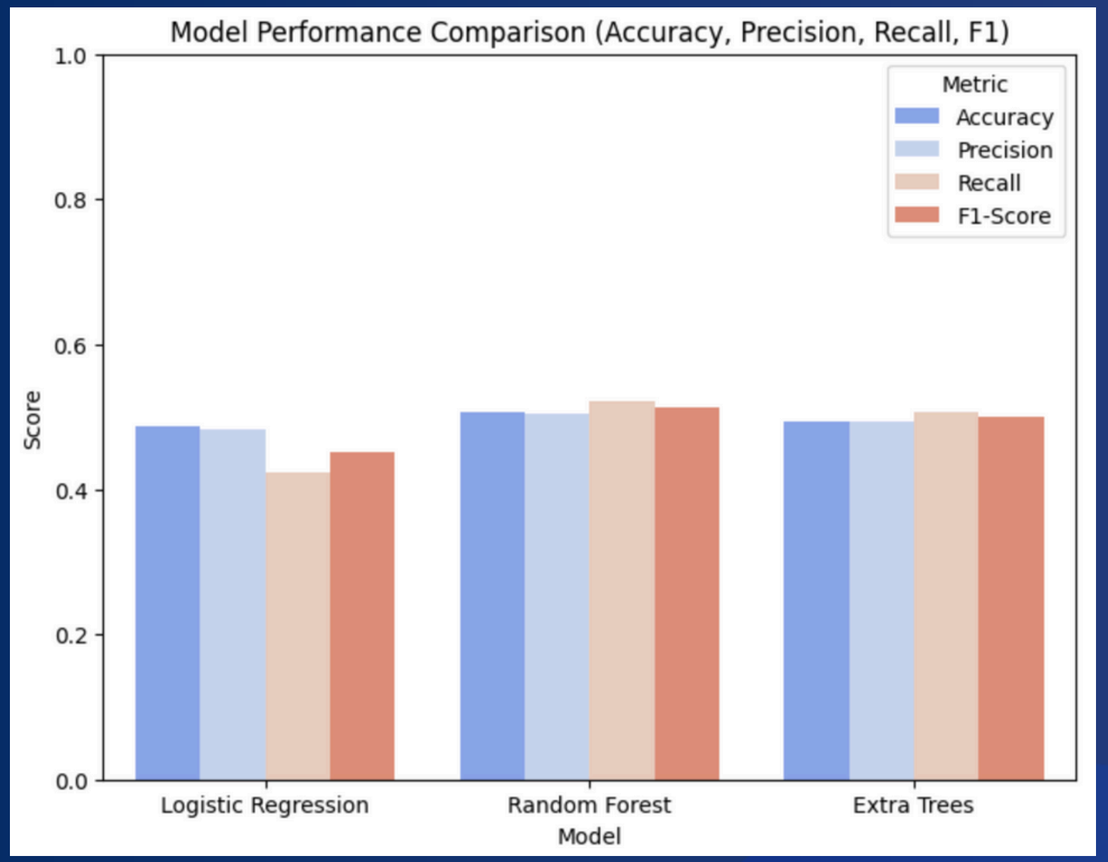
After:



- Equal Feature Importance: all features weighted equally (0-1)
- Outlier Mitigation: outlier becomes 1.0, limiting its disproportionate influence
- Performance: essential for K-Means clustering and speeds up Logistic Regression convergence

SUPERVISED LEARNING

	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	0.485641	0.483076	0.424096	0.451668
1	Random Forest	0.504840	0.504215	0.521641	0.512780
2	Extra Trees	0.493869	0.493539	0.505814	0.499601



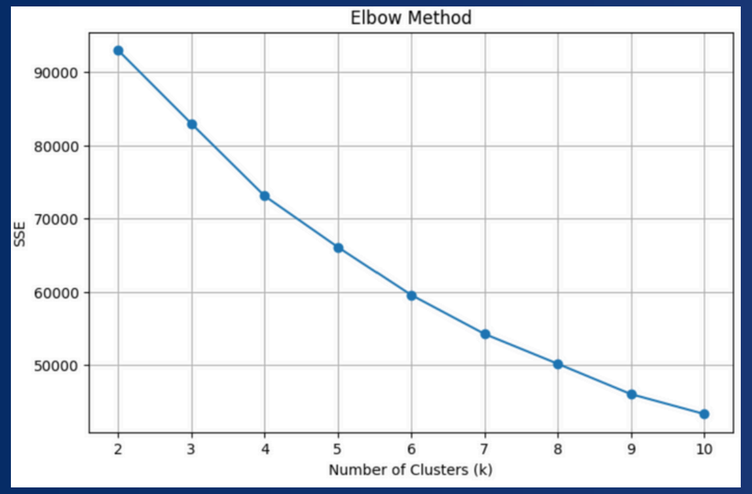
Root Cause of Low Performance: **Weak Features**

- Dataset only captures: study hours, attendance, grades, participation
- Missing critical factors: motivation, learning style, personal circumstances, consistency, mental health

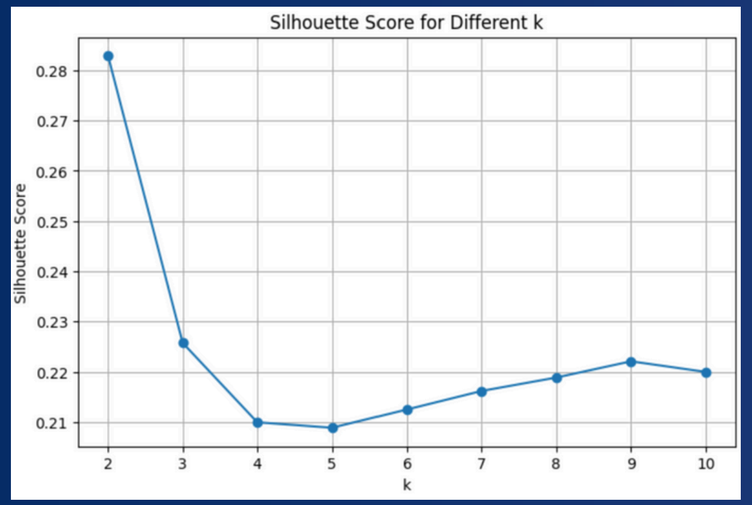
Improvement Attempts:

- Feature Engineering
- Hyperparameter Tuning

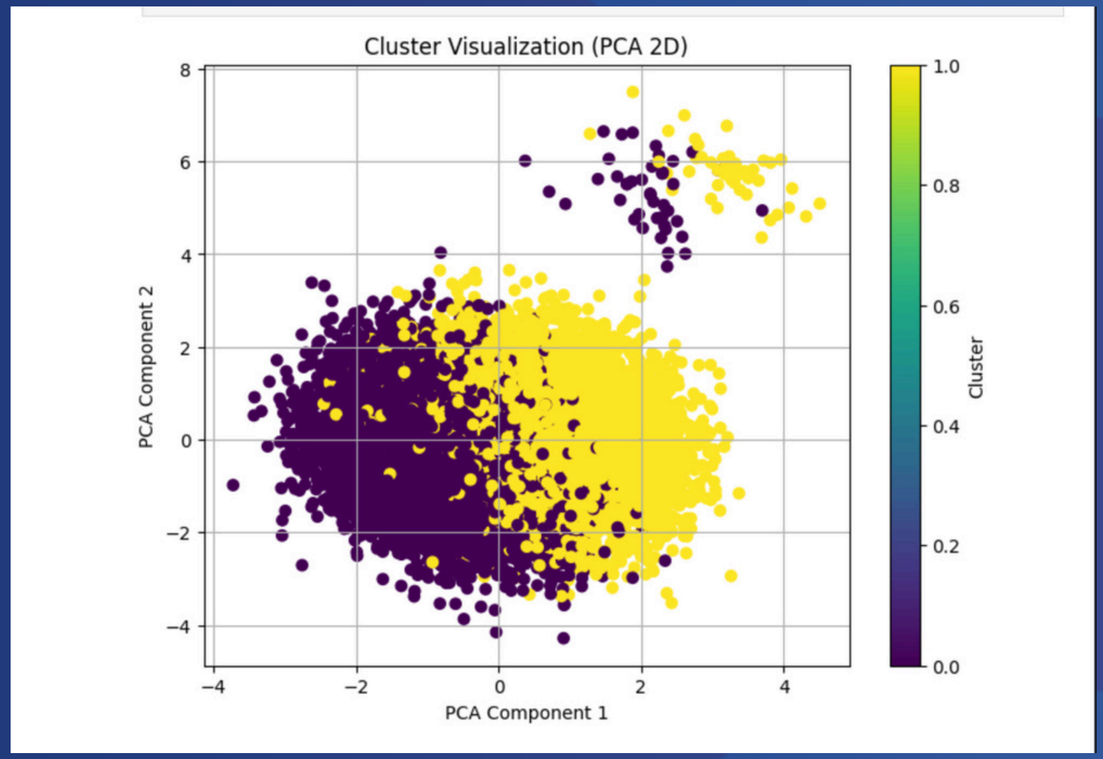
UNSUPERVISED LEARNING



“how compact clusters are”



“how separated clusters are ”



- **Cluster 0:** Low engagement (below-average study, attendance, grades, participation)
- **Cluster 1:** High engagement (above-average study, attendance, grades, participation)

GENERATIVE AI

Template 1:

You are an academic advisor. Provide a short, simple performance advice
for a student based on the following details:

Study Hours: {row['Study Hours per Week']}
Attendance Rate: {row['Attendance Rate']}
Previous Grades: {row['Previous Grades']}
Participation in Activities: {row['Participation in Extracurricular Activities']}
Cluster Group: {row['Cluster']}

Keep the advice in 2 sentences maximum.

- ~50 words total
- Minimal structure and vague instruction
- Length constraint: "2 sentences maximum"

GENERATIVE AI

Template 2:

You are an experienced academic advisor. Provide a detailed academic advising report in markdown.

Student Data:
- Study Hours per Week: {row['Study Hours per Week']:.2f}
- Attendance Rate: {row['Attendance Rate']:.2f}
- Previous Grades: {row['Previous Grades']:.2f}
- Extracurricular Activities: {row['Participation in Extracurricular Activities']}
- Cluster Group: {row['Cluster']}

Write your response using these sections:

Performance Summary
Explain the student's overall performance.

Biggest Weakness
Identify and explain the weakest metric.

Key Strength
State at least one strength.

Recommended Action Plan
Provide 4–6 clear, actionable steps to improve performance.

Cluster Insight
Explain what Cluster {row['Cluster']} means and how students in this cluster typically behave.

- ~150 words total
- Highly structured: 5 specific sections
- Role definition: "experienced academic advisor"
- Format specification: "markdown"
- Clear expectations: "4-6 actionable steps"

Conclusion

Key Insights

- Data quality limited model accuracy (~50%), indicating student success depends on deeper, non-captured factors.
- Generative AI improved usability by providing clear, personalized feedback.

Recommendations

- Use clustering as an early warning tool.
- Improve data collection (motivation, support, well-being).
- Pilot an AI-based student advisor tool.

Future Work

- Short term: add stronger features + advanced models to reach 65–70% accuracy.
- Long term: build a scalable student-success platform validated across universities.

Lessons Learned

Data Preprocessing is Crucial

Handling missing values and cleaning the dataset ensures reliable data, which is essential for accurate model performance

Feature Engineering Improves Model Performance

Creating new features like “Engagement_Score” and “Study_Efficiency” can improve the model’s predictive ability

Choosing the Right Model

Experimenting with different models, like Random Forest, helps identify the best fit for the dataset’s complexity

Generative AI Provides More Tailored Insights

Integrating Generative AI enhances personalization and provides more actionable, context-specific insights

THANK YOU!

SWE485 - Selected Topics in Software Engineering - Artificial Intelligence
December 2025