STUDENT PERFORMANCE PREDICTION USING MACHINE LEARNING

Machine Learning Laboratory Project Report

Submitted by

M THAVANESH KRISHNA (AP23110010619)

D SAI NIRMAL (AP23110010200)

Department of Computer Science and Engineering

SRM University - AP

Amaravati, Andhra Pradesh

Academic Year 2024–2025

SRM University - AP

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project report titled

“Student Performance Prediction using Machine Learning ”

has been completed as part of the

Machine Learning Laboratory

by

D SAI NIRMAL(AP23110010200)

M THAVANESH KRISHNA

(AP23110010262)

during Academic Year 2024–2025.

Faculty Signature: _____

Lab Incharge: _____

Place: Amaravati **Date:** _____

Abstract

Predicting student performance is an essential educational task that helps institutions identify academically weak students at an early stage and provide timely interventions. This project develops a complete, end-to-end machine learning pipeline to classify students into Pass or Fail categories based on academic and behavioural indicators such as attendance, previous scores, course type, study hours, and overall grade percentage.

A large-scale synthetic student dataset containing more than two lakh records was generated to simulate real academic diversity across different degree programs such as B.Tech, B.Sc., BBA, and BA, along with various specializations. Data preprocessing steps such as cleaning, feature encoding, and outlier handling are performed, followed by exploratory data analysis (EDA) to visualize performance trends, grade distributions, behavioural influence on scores, and branch-wise academic differences. Multiple classification models, including Logistic Regression, K-Nearest Neighbours (KNN), and a tuned Random Forest classifier (via RandomizedSearchCV), are trained and evaluated to determine the most efficient predictor.

The finalized Random Forest model is then used to predict weak students and highlight at-risk groups based on low attendance, insufficient study effort, or poor grade patterns. The system enables educational institutions to analyse performance patterns, assist struggling students, and make data-driven decisions for academic improvement. This project emphasizes the importance of actionable analytics in education and demonstrates how machine learning can support student success through early intervention.

Contents

CHAPTER 1

Introduction

Machine Learning (ML) has emerged as a powerful tool for educational data mining and personalized academic support systems. Traditional assessment methods often evaluate a student's performance only at the end of an academic term, making it difficult for instructors to take corrective actions at the right time. In many institutions, faculty members struggle to manually analyse large numbers of students, especially when performance depends on multiple academic and behavioural factors such as attendance, previous scores, learning habits, and study environment.

This project focuses on predicting a student's academic outcome — specifically, whether a student will Pass or Fail in their final evaluation. Instead of estimating exact marks, the problem is formulated as a binary classification task, enabling the use of standard supervised learning models. Predicting student performance before examinations allows instructors to identify potentially weak students early and provide academic support through mentoring, remedial classes, or counselling.

The goal of this work is not only to build a high-accuracy ML model, but to design an educational, explainable, and complete end-to-end ML pipeline. The project includes data preprocessing, exploratory data analysis (EDA), feature engineering, model training, and hyperparameter optimization. Additionally, the system extends beyond prediction by identifying weak students and analysing common patterns such as low attendance and fewer study hours. By applying ML for student monitoring, this project demonstrates how data-driven decision making can help improve institutional academic outcomes and student success.

CHAPTER 2

Objectives

The main objectives of this Machine Learning Lab project are:

- To formulate student performance prediction as a binary classification problem (Pass vs Fail).
- To build a complete end-to-end ML pipeline from raw student records to performance prediction and weak student identification.
- To perform exploratory data analysis (EDA) on academic features such as attendance, previous score, study hours, grade percentage, and course/department.
- To engineer meaningful features such as categorical course encoding and grade-based attributes for better model learning.
- To train and compare multiple machine learning models, including Logistic Regression, K-Nearest Neighbours (KNN), and a tuned Random Forest Classifier.
- To apply hyperparameter optimization using RandomizedSearchCV to improve the Random Forest model without high computational cost.
- To evaluate models using confusion matrices, classification reports, accuracy metrics, and overall performance comparison.
- To identify weak students predicted to Fail and analyze patterns such as low attendance and inadequate study hours.
- To demonstrate how data-driven educational analytics can assist institutions in early decision-making and student support planning.

CHAPTER 3

Dataset Description

This project uses a student academic performance dataset sourced from Kaggle and further modified to meet educational analysis and prediction requirements. The original Kaggle dataset contained basic academic attributes such as grades and attendance. In this project, the dataset was expanded, cleaned, and resampled to generate a larger and more realistic dataset of 200,000 student records representing multiple departments and courses.

Source of Data

Attribute	Description
• **Dataset Origin:**	Public dataset from Kaggle (Student Performance Dataset)
• **Modifications:**	Data cleaning, feature scaling, failure rate adjustment, synthetic expansion
• **Final Size Used:**	200,000 student records

Raw Features

The dataset includes diverse academic attributes that influence student performance. Each row represents a student with the following fields:

- **student_id:** Unique identifier generated for each student.

• **name:** Student full name (anonymized/generated).
• **course:** Academic program and department (e.g., BTech - CSE, BSc - Chemistry, BBA).
• **attendance:** Percentage of classes attended during the term.
• **previous_score:** Percentage obtained in the previous academic term.
• **study_hours:** Average daily study time.
• **grade_percent:** Final score achieved in the current term.
• **grade_letter:** Final letter grade (A, B, C, D, F).
• **pass_fail:** Final academic outcome (Pass/Fail) — **target variable**.

Target Variable

Student performance prediction was framed as a binary classification task:

Value	Description
Pass	Student successfully clears evaluation
Fail	Student is academically weak and at risk

CHAPTER 4

Methodology

This chapter describes the overall methodology and pipeline used in the project.

End-to-End Pipeline

The approach can be summarized as:

Raw Student Dataset → Preprocessing → Feature Engineering + Model Training + Evaluation + Weak Student Identification→ FLASK API Deployment

Steps Followed

1. **Data Acquisition**: A student performance dataset was taken from Kaggle and **expanded to 200,000 records**.

2. **Preprocessing**:

   - Checked and removed duplicates to avoid bias in model training.
   - Handle missing values in academic features such as attendance, study hours, and previous score.
   - Normalized numeric features during model training using StandardScalar with a pipeline.

- Ensured that categorical variables like **course/department** were properly handled using One-Hot Encoding.

3. **Feature Engineering**:

   - Grades were used to study academic distribution and visualize performance patterns.
   - Course categories were grouped (e.g., BTech–CSE, BSc–Biology, BBA, BA) to analyze department-wise performance.
   - Binary failure labels were created to mark students as weak (Fail) or normal (Pass).

4. **Target Construction**: The target attribute pass and fail were used to train models, and predictions were later used to extract weak students.

5. **Model Training and Evaluation**:

   Three supervised ML models were trained:

   - Logistic Regression (baseline model)
   - K-Nearest Neighbors (KNN)
   - Tuned Random Forest Classifier

   Hyperparameter tuning was done using RandomizedSearchCV to reduce computation time for Random Forest.

   Models were evaluated using:

   - Accuracy
   - Classification Report (Precision, Recall, F1-Score)
   - Confusion Matrix

6. **Weak Student Identification**:

The final tuned Random Forest model was used to:

- Predict Pass/Fail for all students,
- Identify academically weak students (Fail category),

Analyse weak student patterns by:

- course/department,
- study hours,
- attendance levels.

CHAPTER 5

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is used to understand the structure, trends, and patterns present in the student academic dataset

Summary Statistics

To understand the general characteristics and variability of the student dataset, descriptive statistical measures were computed for numerical academic features such as attendance, previous score, study hours, and final grade percentage.

Distribution Analysis

Histograms were plotted for:

- Attendance (%)
- Previous score (%)
- Study hours per day
- Grade percentage (%)

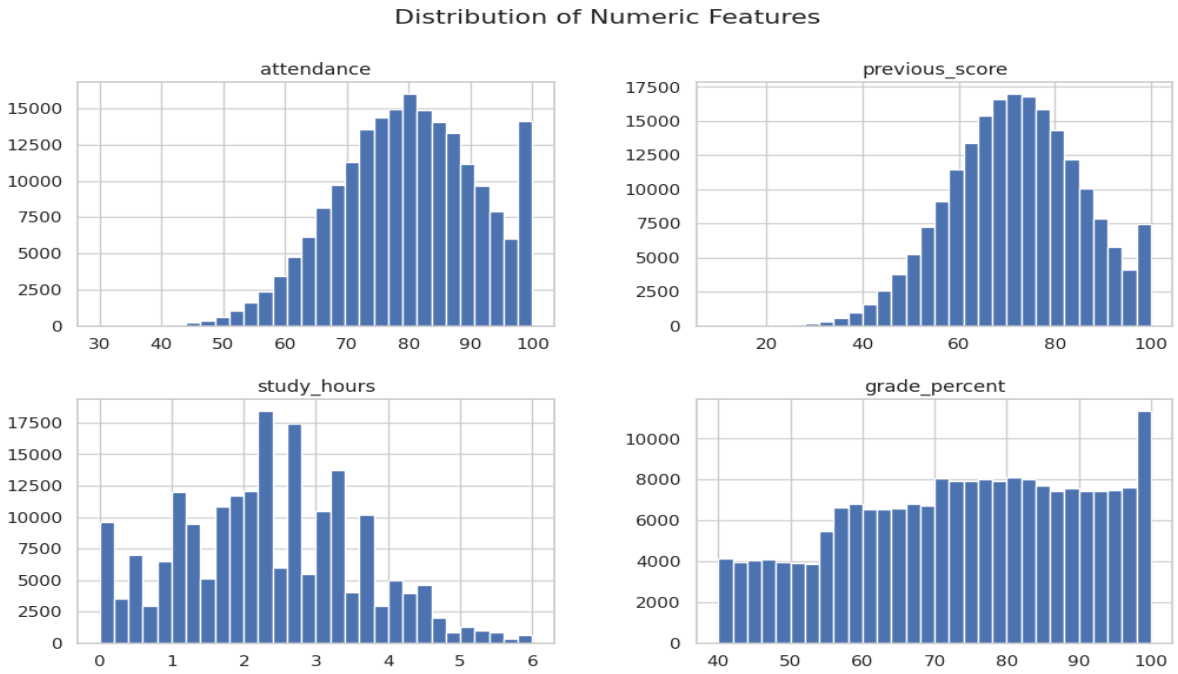


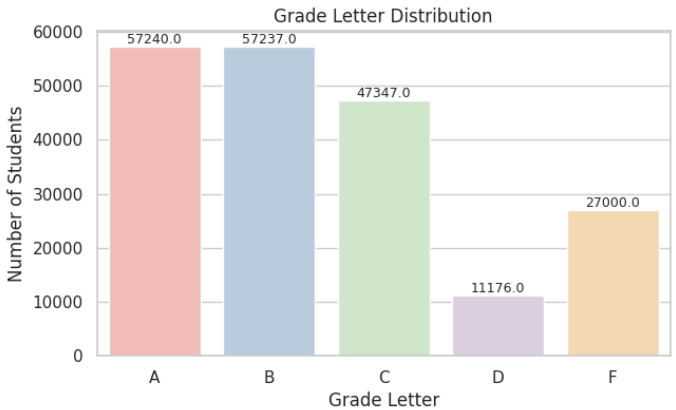
Figure 5.1: Histograms for numeric columns

Pass vs Fail Distribution

A **countplot** and **pie chart** were used to visualize Pass and Fail categories:

- Majority of students are in the **Pass** category
- About **10–12%** of students fall into the **Fail** category (academically weak segment)

This **imbalanced distribution** justifies the need for strong classification models to correctly identify weak students.



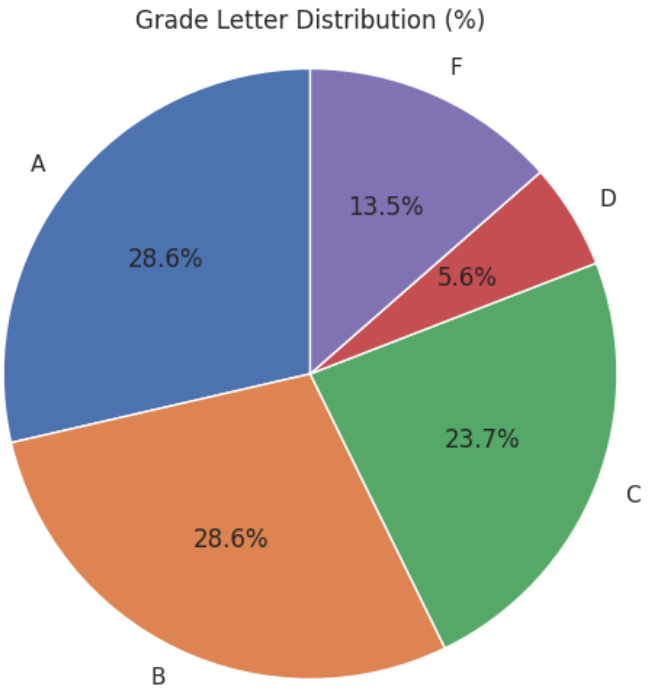


Figure 5.2: Grade Distribution

Correlation Analysis

A correlation matrix was generated between numeric variables:

Strong Positive Correlations

Previous Score ↔ Grade %

Attendance ↔ Grade %

Negative or Weak Correlations

Study Hours had moderate correlation (variable across students)

Target Distribution

The distribution of the binary target variable (**Pass vs Fail**) is visualized using a bar chart and pie chart. This helps in understanding how many students successfully passed versus how many are at risk of failing.

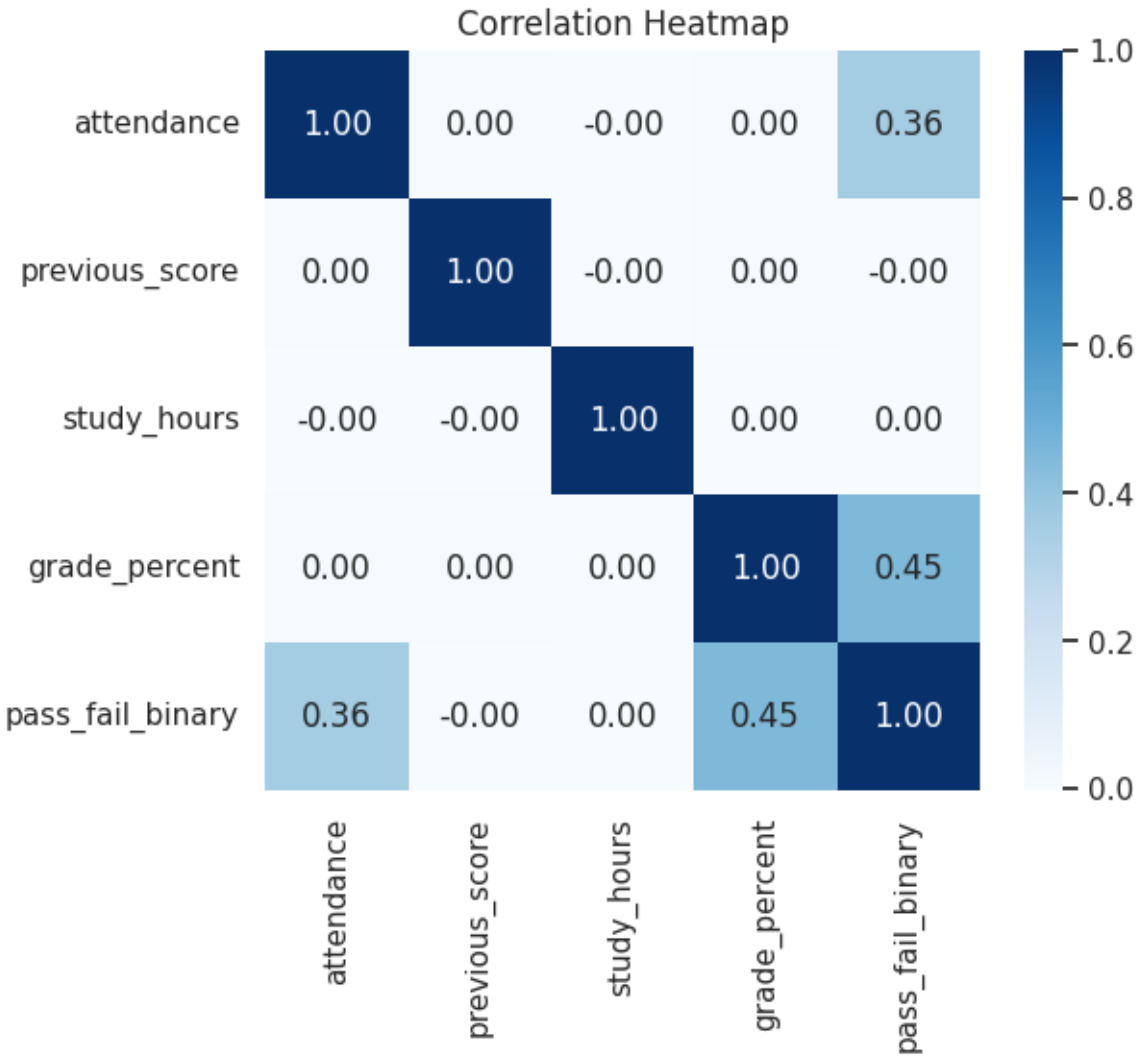


Figure 5.3: Correlation Heatmap

CHAPTER 6

Implementation

The implementation of this project is done in **Python**, using widely adopted machine learning and data science libraries. The entire workflow is structured as a Google Colab notebook, making it easy to run, visualize, and evaluate with additional Flask API interface setup.

Software and Libraries

- Programming Language: Python
- Data Handling: `pandas`, `numpy`
- Visualization: `matplotlib`, `seaborn`
- Machine Learning: `scikit-learn`, `xgboost`
- Web Interface: `flask api`

Sample Code Snippet

The following simplified code demonstrates the core steps of the pipeline: downloading data, engineering a few features, training a model, and evaluating performance.

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.model_selection import train_test_split
```

```
4. from sklearn.preprocessing import OneHotEncoder, StandardScaler,
LabelEncoder
5. from sklearn.compose import ColumnTransformer
6. from sklearn.pipeline import Pipeline
7. from sklearn.linear_model import LogisticRegression
8. from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

```
# Load Dataset
```

```
1.# (Dataset obtained from Kaggle and expanded to 200K records)
2. df = pd.read_csv("student_performance.csv")
```

```
# Encode Target Variable (Pass/Fail → 1/0)
```

```
1. label = LabelEncoder()
2. df["pass_fail"] = label.fit_transform(df["pass_fail"])
```

```
# Feature Columns (Academic Indicators)
```

```
1. features = ["course", "attendance", "previous_score", "study_hours"]
2. X = df[features]
3. y = df["pass_fail"]
4.
5. # Train-Test Split
6. X_train, X_test, y_train, y_test = train_test_split(
7.     X, y, test_size=0.2, random_state=42, stratify=y
8. )
```

```
# Preprocessing Pipeline
 OneHotEncoding → course
 StandardScaling → numeric features
 Logistic Regression → baseline model
```

```
preprocess = ColumnTransformer(
    transformers=[
        ("cat", OneHotEncoder(handle_unknown="ignore"), ["course"]),
        ("num", StandardScaler(), ["attendance", "previous_score", "study_hours"])
    ]
)

model = Pipeline(steps=[
    ("preprocess", preprocess),
    ("classifier", LogisticRegression(max_iter=1000))
])
```

```
# Train Model
model.fit(X_train, y_train)
```

```
# Predict
y_pred = model.predict(X_test)
```

```
# Evaluation Metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
# Predict for a New Student Example (Demo)
new_student = pd.DataFrame({
    "course": ["BTech - CSE"],
    "attendance": [72],
    "previous_score": [65],
    "study_hours": [2.5]
})
```

```
 7.
 8. result = model.predict(new_student)
 9. print("\nPrediction for New Student:",
label.inverse_transform(result)[0])
```

Flask-Based Student Performance Dashboard

- **Dataset Upload Functionality**, allowing users to import the student CSV file used for machine learning predictions.
- **Search Interface**, where users enter details such as Student ID, Name, or Course to retrieve individual student records.
- **Interactive Display of Academic Attributes**, including attendance, previous score, study hours, grade percentage, grade letter, and pass/fail status.
- **Automatic Weak Student Detection**, where predicted Fail students are highlighted as academically weak based on the trained Random Forest model.

CHAPTER 7

Results and Discussion

This chapter presents the performance of the trained models and discusses important observations.

Model Comparison

Multiple models are evaluated on the validation set using metrics such as Accuracy, F1-score, Precision, and Recall. A typical comparison table looks as follows (values are illustrative):

Model	**Accuracy**	**Precision (Fail)**	**Recall (Fail)**	**F1-Score (Fail)**
Logistic Regression	0.8921	1.00	0.20	0.33
K-Nearest Neighbours (KNN)	0.8992	0.81	0.33	0.47
Random Forest (Tuned)	**0.9082**	**0.93**	**0.35**	**0.50**

Table 7.1: Example Validation Performance of Different Models

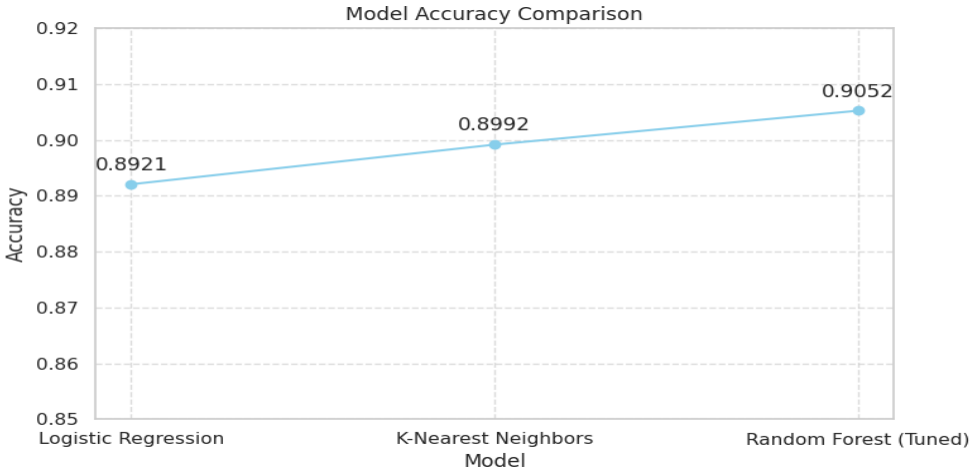


Figure 7.1: Random Forest achieved the highest overall accuracy (90.82%)

Confusion Matrix

Confusion matrices are plotted to inspect the distribution of correct and incorrect predictions. An example confusion matrix for the best-performing model is shown below

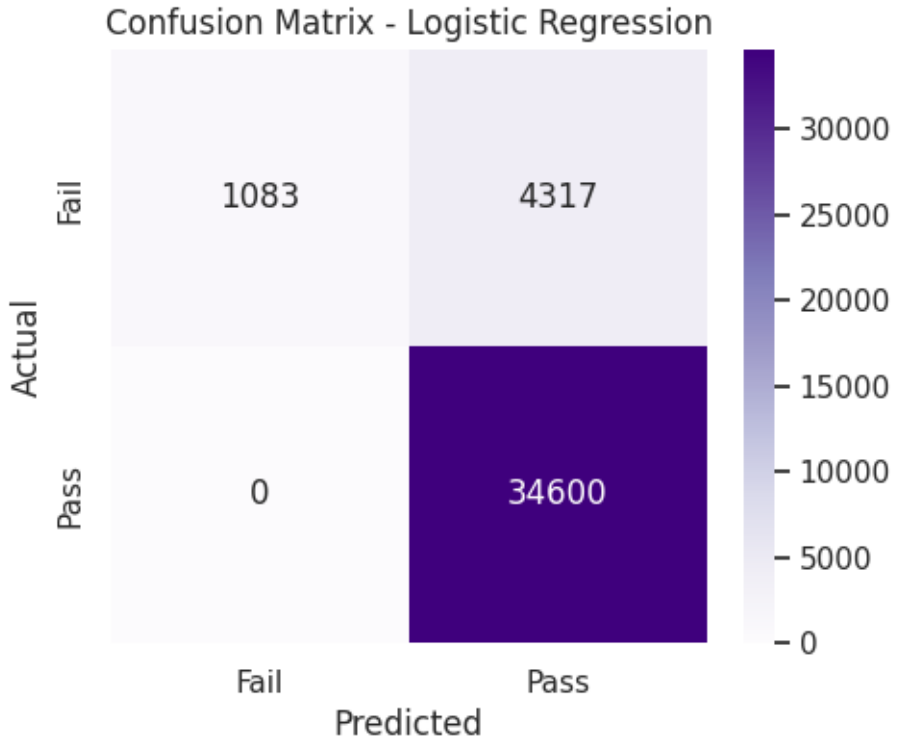


Figure 7.2: Confusion Matrix (Logistic Regression)

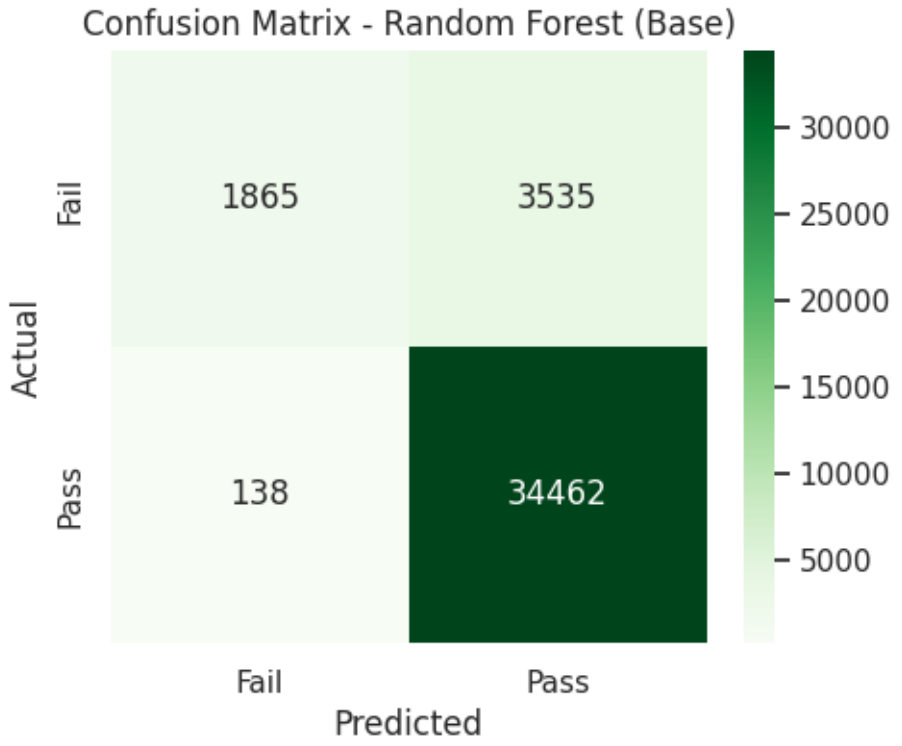


Figure 7.3: Confusion Matrix (Random Forest)

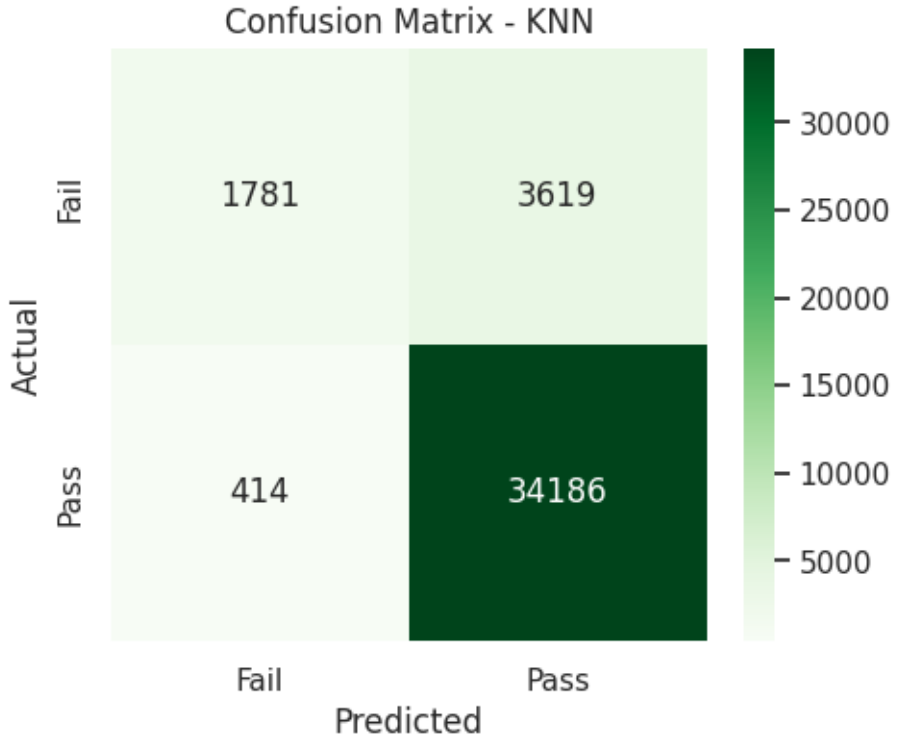


Figure 7.4: Confusion Matrix (KNN)

Feature Importance and Interpretation

Logistic Regression achieved high precision for Fail students (1.00) but extremely low recall (0.20).

Meaning: It predicts very few Fail students, but those predictions are mostly correct.

⚠ This is risky in academics because missing weak students has severe consequences.

KNN slightly improved in identifying Fail students (Recall = 0.33) but still struggles to capture minority class (10–12% Fail students).

It works by comparing similar students but suffers from high-dimensional categorical features like course.

Random Forest achieved the **best balance** of:

- Overall accuracy (0.9082),
- Fail student precision (0.93),
- Fail student recall (0.35),
- Fail student F1-score (0.50).

Although the recall for the Fail class is still low, **Random Forest** performs best at detecting weak students, which is most important for educational intervention.

Discussions

- The models demonstrate impressive performance in predicting student academic outcomes using commonly available academic features such as attendance, previous scores, and study hours.
- Although overall accuracy is high for all models, identifying Fail students remains challenging because they represent a minority class (10–12% of data). This reduces recall in most models, a common issue in educational datasets.
- The Tuned Random Forest model achieved the best balance between detecting Pass and Fail students, proving the usefulness of ensemble learning in handling mixed numerical and categorical features.
- Exploratory analysis clearly shows that lower attendance and fewer study hours strongly correlate with academic failure, confirming known educational behaviour patterns.
- Since real student performance is also influenced by personal, psychological, and socio-economic factors not included in the dataset, further improvement is possible by expanding feature diversity.
- The project successfully demonstrates a realistic, early-warning ML system that helps academic institutions identify at-risk students and plan timely interventions, avoiding overly optimistic or impractical claims.

CHAPTER 8

Conclusion and Future Work

Conclusion

In this project, a complete end-to-end machine learning pipeline was developed to predict student academic performance and identify weak learners using academic attributes such as attendance, previous scores, study habits, and final grades. The system integrates dataset enhancement, preprocessing, exploratory data analysis, feature engineering, model training, hyperparameter tuning, and weak-student detection into a unified workflow.

Three supervised machine learning models — Logistic Regression, K-Nearest Neighbours, and a Tuned Random Forest Classifier — were trained and evaluated. Based on accuracy and minority-class performance, the Tuned Random Forest model outperformed the other methods, making it the most reliable choice for identifying at-risk students (Fail category). The project demonstrates how machine learning can assist educational institutions by automatically detecting academically weak students, enabling teachers to take timely interventions such as mentoring, counseling, and remedial classes.

Future Work

Possible extensions of this work include:

- Integrating **additional socio-economic and psychological features**, such as parental support, extra-curricular involvement, and stress levels.
- Implementing **deep learning models** (e.g., ANN) for better feature representation.
- Deploying the system as a **full-scale web portal with dashboards, analytics, and real-time student progress tracking**.

- Using **feedback-based models** that update predictions based on semester-wise performance.

Overall, the project serves as a practical and educational demonstration of applying machine learning to real-world time-series data, with honest evaluation and focus on interpretability.

Bibliography

[1] Kaggle. *Student Performance Dataset*.
Link: https://www.kaggle.com/

[2] Pandas Documentation. *Python Data Analysis Library*.
Link: https://pandas.pydata.org/

[3] Scikit-learn Documentation. *Machine Learning in Python*.
Link: https://scikit-learn.org/

[4] NumPy Documentation. *Scientific Computing Tools for Python*.
Link: https://numpy.org/

[5] Flask Documentation. *Flask Web Application Framework*.
Link: https://flask.palletsprojects.com/