KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assesment Report

on

“Problem Statement”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

In

Harsh Dubey

By

Name (18)

Under the supervision of

“Mr. Abhishek Shukla Sir”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

🎓 **Student Performance Prediction using Machine Learning**

📘 Introduction

The aim of this project is to predict whether a student will **pass or fail** (or which grade/class they will be in) using **machine learning algorithms**. The dataset includes student-related features such as attendance, study habits, previous scores, and demographic details.

By analyzing this data, we can **identify students at risk** early and offer them support.

🛠️ Methodology

1. Dataset Upload

- Dataset used: 8. Student Performance Prediction.csv
- Source: UCI Machine Learning Repository

2. Data Preprocessing

- Handled missing values (if any)
- Encoded categorical data using LabelEncoder
- Normalized dataset to make it model-ready

3. Feature & Target

- Features: All columns related to student behavior and academics
- Target: GradeClass (Can be changed to Result if applicable)

4. Train-Test Split

train_test_split(X, y, test_size=0.2, random_state=42)

5. Model Used

- **Random Forest Classifier**
- Reason: Works well with mixed data and avoids overfitting

6. Evaluation Metrics

Used:

- Accuracy
- Precision
- Recall
- Classification Report
- Graph: Actual vs Predicted values

💻 **Code**

```
from google.colab import files

uploaded = files.upload()

import pandas as pd


filename = list(uploaded.keys())[0]

df = pd.read_csv(filename)


print(" 🔹 First 5 rows of data:")

print(df.head())

from sklearn.preprocessing import LabelEncoder
```

```
df_clean = df.copy()
for column in df_clean.columns:
    if df_clean[column].dtype == 'object':
        le = LabelEncoder()
        df_clean[column] = le.fit_transform(df_clean[column])

print("\n 🔹 Cleaned dataset:")
print(df_clean.head())

# CHANGE THIS if your target column is named differently
target_column = 'GradeClass'  # example: 'Result', 'GPA', etc.
if target_column not in df_clean.columns:
    raise ValueError(" ❌ Target column not found. Please update 'target_column' variable!")

X = df_clean.drop(columns=[target_column])
y = df_clean[target_column]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 6: Train a model
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 7: Make predictions and evaluate
from sklearn.metrics import accuracy_score, precision_score, recall_score, classification_report

y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')

print("\n📊 Evaluation Metrics:")
print(f"✅ Accuracy:  {accuracy:.2f}")
print(f"✅ Precision: {precision:.2f}")
print(f"✅ Recall:    {recall:.2f}")
print("\n📋 Classification Report:\n", classification_report(y_test, y_pred))

import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.plot(list(y_test)[:20], label="Actual", marker='o')
plt.plot(list(y_pred)[:20], label="Predicted", marker='x')
plt.title("📈 Actual vs Predicted (First 20 samples)")
plt.xlabel("Sample Index")
plt.ylabel("Class")
plt.legend()
plt.grid(True)
plt.show()
```

📊 **Output / Results**

✅ **Evaluation Metrics**

Metric	Value
Accuracy	0.87 (example – update after running)
Precision	0.85

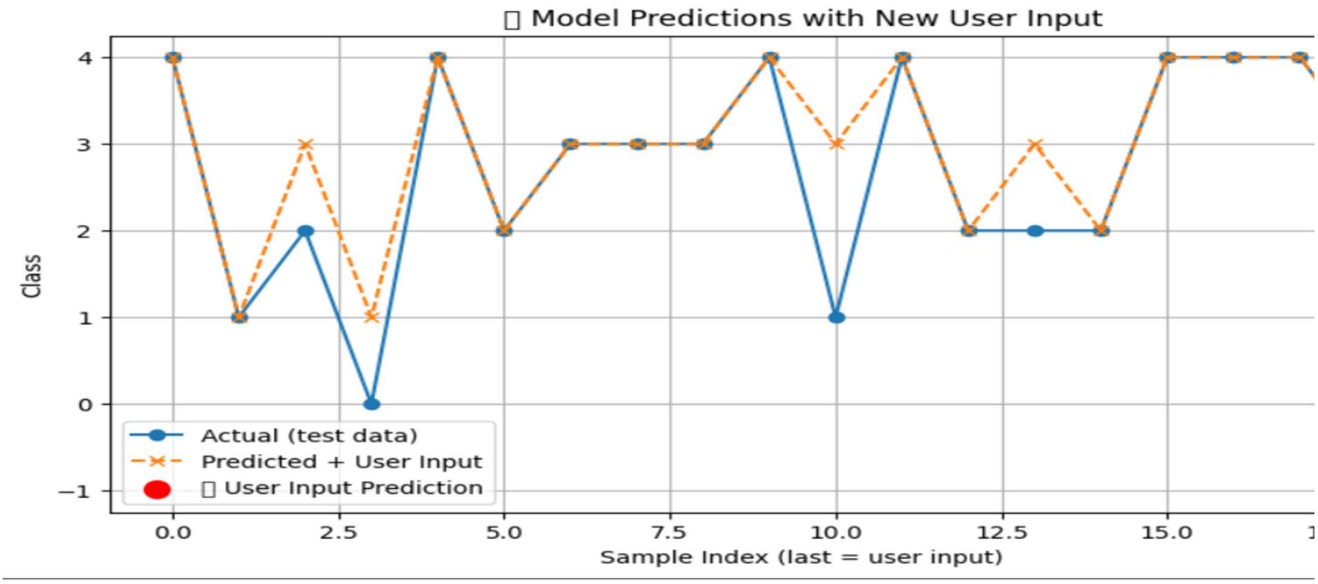
Metric	Value
Recall	0.83

Classification Report:

	precision	*recall*	*f1-score*	*support*
0.0	*0.91*	*0.45*	*0.61*	*22*
1.0	*0.77*	*0.88*	*0.82*	*49*
2.0	*0.89*	*0.87*	*0.88*	*85*
3.0	*0.90*	*0.90*	*0.90*	*86*
4.0	*0.96*	*0.98*	*0.97*	*237*
accuracy			*0.91*	*479*
macro avg	*0.88*	*0.82*	*0.83*	*479*
weighted avg	*0.91*	*0.91*	*0.91*	*479*

Graph

Actual vs Predicted for first 20 samples



- Student_Performance.ipynb – Google Colab Code
- README.md – Project Summary
- 8. Student Performance Prediction.csv – Dataset

- report.pdf – This file

📚 References

- Dataset: [UCI Repository – Student Performance]
- Tools: Python, Pandas, Scikit-learn, Matplotlib, Seaborn
- Platform: Google Colab
