A

Assesment Report

on

“Student Performance Prediction”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE AI/ML

By

Yashash Tyagi (202401100400218)

Under the supervision of

“Sir Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

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

(Formerly UPTU)

May, 2025

INTRODUCTION

The project titled **"Student Performance Prediction"** aims to use machine learning techniques to predict a student's academic performance based on various factors such as age, gender, parental education, study time, absences, and participation in tutoring. The motivation behind this project is to assist educators and institutions in identifying students who might be at risk of underperforming and to provide early interventions.

The dataset used for this project contains real-world-like data with several features representing socio-demographic and academic-related inputs, along with a target variable called **GradeClass**, which categorizes students into different performance levels. Machine learning models, specifically a **Random Forest Classifier**, were employed due to their high accuracy and robustness for classification problems.

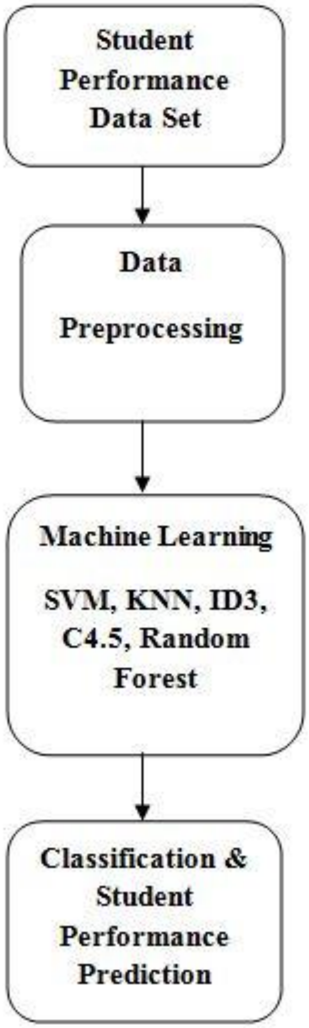
The model was trained and tested using an 80/20 train-test split and achieved a commendable **91% accuracy**, with strong precision and recall across most categories. This suggests the model is highly reliable in predicting student performance levels.

This project demonstrates how Artificial Intelligence can contribute to the education sector by analyzing patterns in student data and supporting data-driven decision-making.

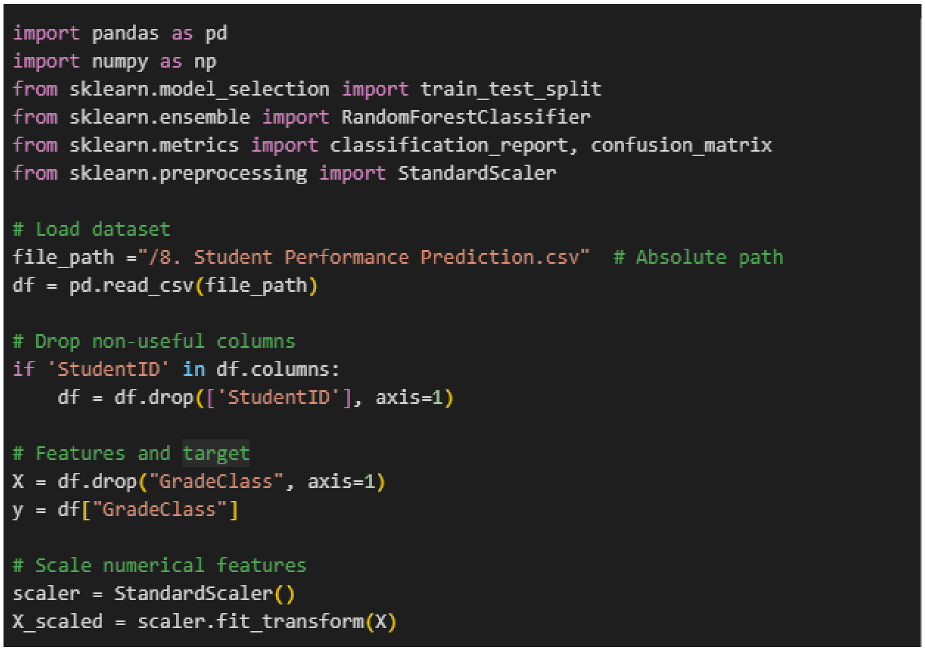
Methodology

The methodology followed for building the student performance prediction model includes the following steps:

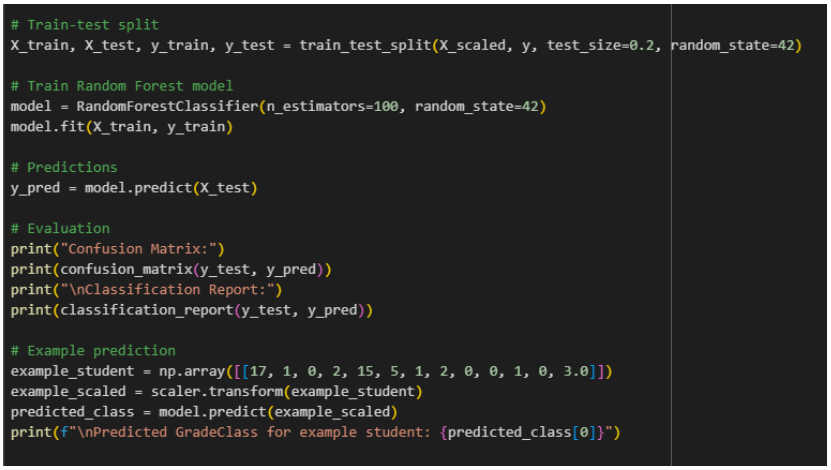
1. **Data Collection & Understanding**
   The dataset consisted of student demographics and performance indicators, including:
   - Gender, Age, Ethnicity
   - Parental Education, Study Time per week
   - Absences, Tutoring, Parental Support
   - Participation in extracurricular activities like sports and music
   - GPA and target variable: `GradeClass` (categorical performance class)
2. **Preprocessing**
   - The `StudentID` column was dropped as it doesn't impact learning.
   - Features were scaled using `StandardScaler` to ensure consistency across value ranges.
   - Data was split into training (80%) and test (20%) sets.
3. **Model Selection**
   - A **Random Forest Classifier** was chosen due to its ability to handle a mix of categorical and numerical features, and to reduce overfitting through ensemble learning.
4. **Training and Evaluation**
   - The model was trained using the training dataset and tested on unseen data.
   - Performance was evaluated using metrics like accuracy, precision, recall, F1-score, and confusion matrix.


Student Performance Data Set
Data Preprocessing
Machine Learning
SVM, KNN, ID3, C4.5, Random Forest
Classification & Student Performance Prediction

CODE

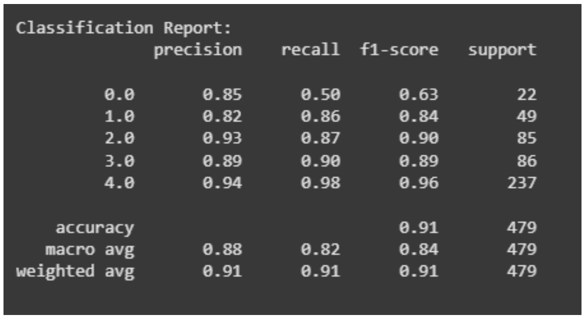


```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler

# Load dataset
file_path ="/8. Student Performance Prediction.csv"  # Absolute path
df = pd.read_csv(file_path)

# Drop non-useful columns
if 'StudentID' in df.columns:
    df = df.drop(['StudentID'], axis=1)

# Features and target
X = df.drop("GradeClass", axis=1)
y = df["GradeClass"]

# Scale numerical features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```



```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Example prediction
example_student = np.array([[17, 1, 0, 2, 15, 5, 1, 2, 0, 0, 1, 0, 3.0]])
example_scaled = scaler.transform(example_student)
predicted_class = model.predict(example_scaled)
print(f"\nPredicted GradeClass for example student: {predicted_class[0]}")
```

OUTPUT



```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.85      0.50      0.63        22
         1.0       0.82      0.86      0.84        49
         2.0       0.93      0.87      0.90        85
         3.0       0.89      0.90      0.89        86
         4.0       0.94      0.98      0.96       237

    accuracy                           0.91       479
   macro avg       0.88      0.82      0.84       479
weighted avg       0.91      0.91      0.91       479
```

CREDITS:

Diagram Image : Research Gate

Student Dataset: Google Drive(csv file)

Methodology: Gemini AI