Assessment Report

on

“Student Performance Prediction”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Priya Chauhan

Roll Number : 202401100400148,

Section: C

Under the supervision of

“ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

In today’s education system, identifying students at risk of failing is crucial. Various factors such as attendance, study habits, parental support, and previous academic performance play a vital role in predicting student success. This project focuses on building a machine learning model that uses these factors to classify students into "Pass" or "Fail" categories. Such prediction systems can help in timely intervention to support students in need.

Objective:

To develop a classification model that predicts whether a student will pass or fail based on attendance, previous scores, and study habits.

Methodology

Data Preprocessing

- Loaded and cleaned the dataset provided in CSV format.
- Converted the GPA into a binary target variable:
  - 1 → Pass (GPA $\geq$ 2.0)
  - 0 → Fail (GPA $<$ 2.0)

Feature Selection

- Dropped irrelevant features such as StudentID, GPA, and GradeClass.
- Retained features related to demographics, study behavior, attendance, and parental support.

Model Used

- Used a Random Forest Classifier, which is an ensemble model known for its robustness and high accuracy.
- Model was trained using an 80-20 train-test split.

Evaluation

- Accuracy, Precision, and Recall metrics were calculated.
- A confusion matrix was plotted to visualize prediction performance.

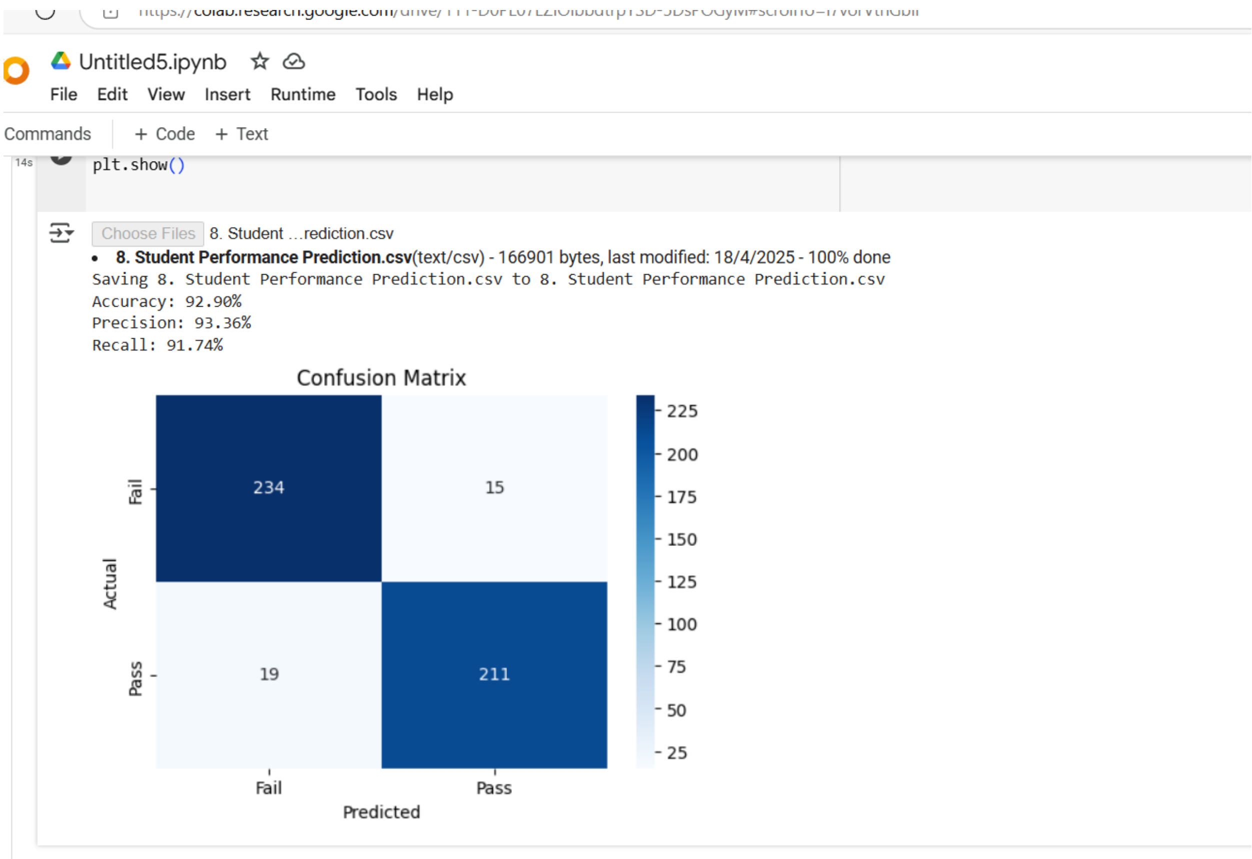
CODE

```
# Upload CSV file
# from google.colab import files
# uploaded = files.upload()

# # Import libraries
# import pandas as pd
# import seaborn as sns
# import matplotlib.pyplot as plt
# from sklearn.model_selection import train_test_split
# from sklearn.ensemble import RandomForestClassifier
# from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score

# # Load dataset
# df = pd.read_csv("8. Student Performance Prediction.csv")  # Replace
with your filename if different

# # Create binary target variable: Pass = 1 if GPA >= 2.0, else 0
# df['Pass'] = (df['GPA'] >= 2.0).astype(int)

# # Select features and target
# X = df.drop(columns=['StudentID', 'GPA', 'GradeClass', 'Pass'])
# y = df['Pass']

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

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

# # Make predictions
# y_pred = model.predict(X_test)
# # Evaluation metrics
# accuracy = accuracy_score(y_test, y_pred)
# precision = precision_score(y_test, y_pred)
# recall = recall_score(y_test, y_pred)
# cm = confusion_matrix(y_test, y_pred)

# # Print metrics
# print(f"Accuracy: {accuracy * 100:.2f}%")
# print(f"Precision: {precision * 100:.2f}%")
# print(f"Recall: {recall * 100:.2f}%")

# # Plot confusion matrix heatmap
# plt.figure(figsize=(6, 4))
# sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=["Fail", "Pass"], yticklabels=["Fail", "Pass"])
# plt.xlabel("Predicted")
# plt.ylabel("Actual")
# plt.title("Confusion Matrix")
# plt.show()
```

Output/Result

- **Model Accuracy:** 92.90%
- **Precision:** 93.36%
- **Recall:** 91.74%

□ *Confusion Matrix Heatmap:*



```
plt.show()
```

- **8. Student Performance Prediction.csv**(text/csv) - 166901 bytes, last modified: 18/4/2025 - 100% done

```
Saving 8. Student Performance Prediction.csv to 8. Student Performance Prediction.csv
Accuracy: 92.90%
Precision: 93.36%
Recall: 91.74%
```