Assessment 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

Vanshika Arora(202401100400208)

Under the supervision of

“Mr.Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

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

(Formerly UPTU)

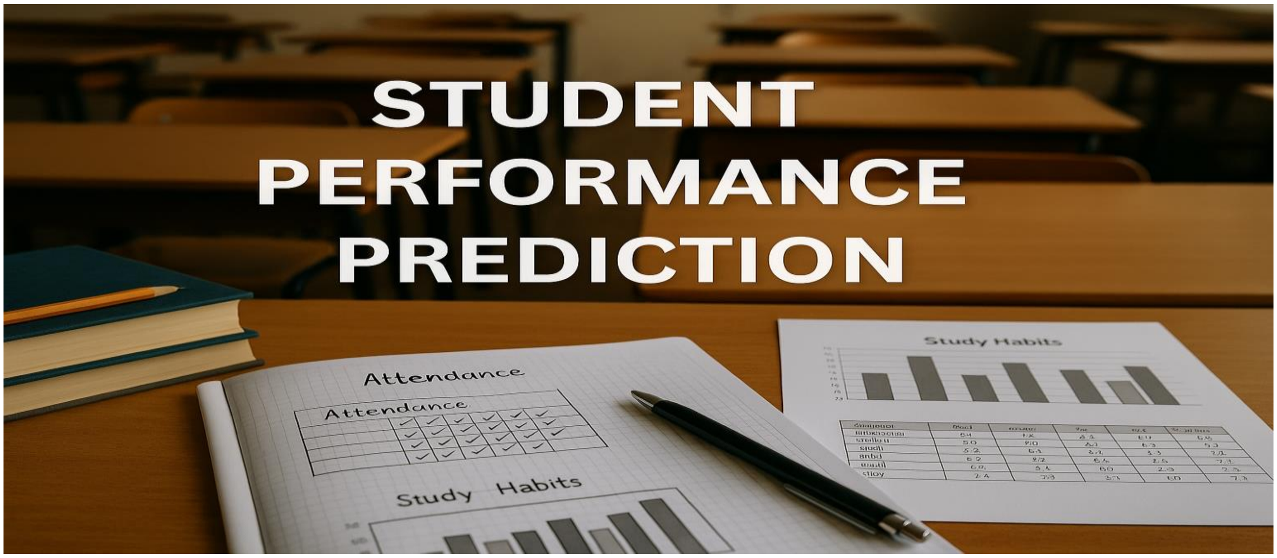
INTRODUCTION

Predicting student performance is a valuable tool for educational institutions to identify at-risk students and implement timely interventions. In this project, we used a classification model to determine whether a student will **pass or fail** based on their academic behaviour, attendance, and support systems.

We used a dataset containing features such as:

- Absences
- Weekly study time
- Tutoring support
- Parental support
- GPA (used to define pass/fail)

The goal was to build a machine learning model that could classify students accurately based on these features. We selected a **Random Forest Classifier** for this task, given its ability to handle both categorical and numerical data effectively and its robustness against overfitting. The model is trained and evaluated using standard performance metrics like **accuracy**, **precision**, and **recall**, with visualization through a **confusion matrix heatmap** for better interpretability.



METHODOLOGY

1. **Dataset Preparation**:
   - Uploaded the dataset Student Performance Prediction.csv to Google Colab.
   - A new column Pass was created: students with GPA ≥ 2.0 were labeled as "Pass" (1), otherwise "Fail" (0).
2. **Feature Selection**:
   - **Selected features:**

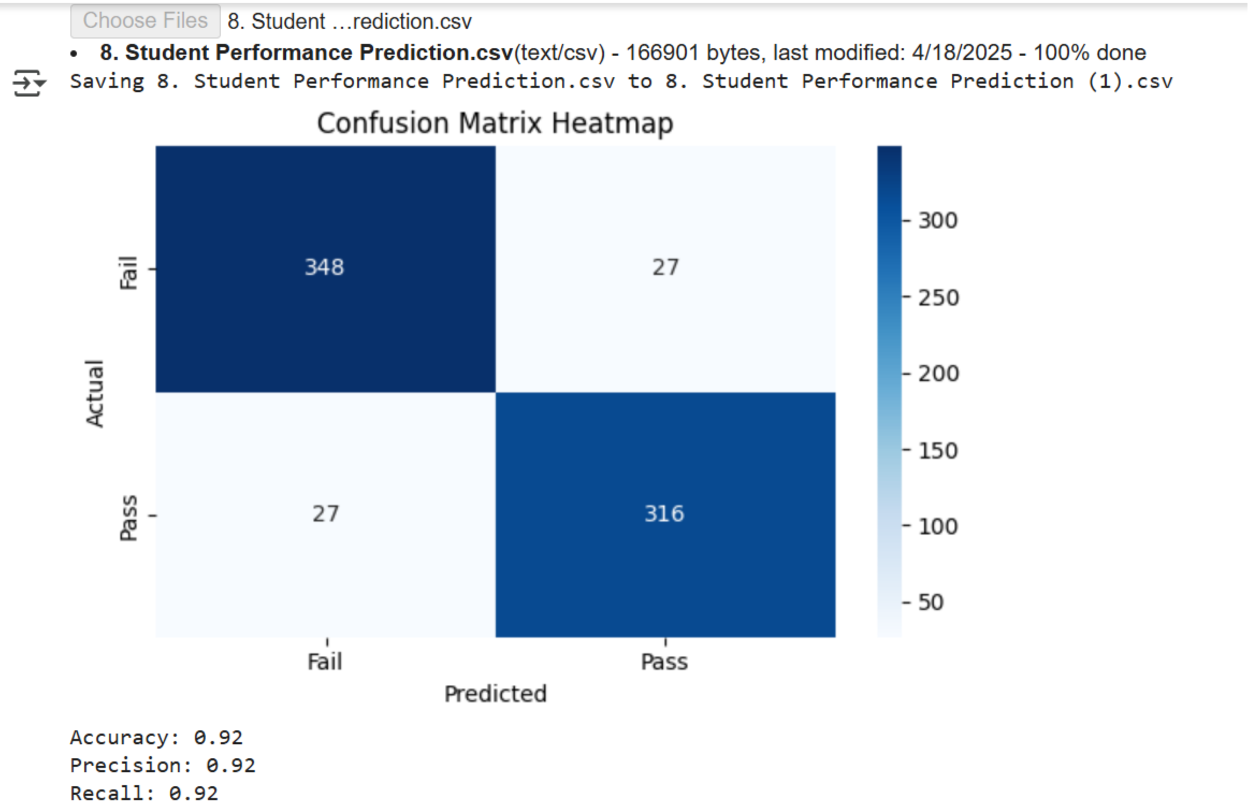
     Absences, StudyTimeWeekly, Tutoring, and ParentalSupport.
3. **Model Building**:
   - Used a **Random Forest Classifier**, known for handling mixed data types and providing high accuracy.
   - The dataset was split into training and testing sets (70/30 split).
4. **Evaluation**:
   - **Confusion Matrix:** Visualized as a heatmap to assess the number of true positives, true negatives, false positives, and false negatives.
   - **Accuracy:** Measures the overall correctness of the model.
   - **Precision:** Indicates the proportion of correctly predicted positive observations among all predicted positives.
   - **Recall:** Measures the proportion of actual positive cases that were correctly identified.

CODE :

```
# Step 1: Upload the dataset
from google.colab import files
uploaded = files.upload()

# Step 2: Import libraries and load the data
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 the dataset
df = pd.read_csv("8. Student Performance Prediction.csv")

# Step 3: Define Pass/Fail label based on GPA
df['Pass'] = (df['GPA'] >= 2.0).astype(int)

# Step 4: Select features and labels
features = ['Absences', 'StudyTimeWeekly', 'Tutoring', 'ParentalSupport']
X = df[features]
y = df['Pass']

# Step 5: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
# Step 6: Train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 7: Make predictions
y_pred = model.predict(X_test)

# Step 8: Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
labels = ['Fail', 'Pass']

# Step 9: Plot Confusion Matrix Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.tight_layout()
plt.show()

# Step 10: Evaluation Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
```

OUTPUT



8. Student …rediction.csv

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

```
Saving 8. Student Performance Prediction.csv to 8. Student Performance Prediction (1).csv
```

```
Accuracy: 0.92
Precision: 0.92
Recall: 0.92
```

REFERENCES/CREDITS

- Dataset: *Student Performance Prediction.csv* (provided)
- Libraries: pandas, scikit-learn, matplotlib, seaborn
- Platform: Google Colab (https://colab.research.google.com/)
- Random Forest Classifier: Scikit-learn Documentation