Assessment Report

on

“Student Perfromance Predictor - Predictive Analysis of Student Performance Using Random Forest Classifier and Machine Learning.”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSEAI

By

Yogesh

202401100300288

Under the supervision of

ABHISHEK SHUKLA

KIET Group of Institutions, Ghaziabad

Affiliated to

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

(Formerly UPTU)

May, 2025

CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

INTRODUCTION

In the modern educational landscape, predicting student performance can provide valuable insights into the factors influencing academic success. Early identification of students who might be struggling allows for timely intervention and personalized support. This project, **Student Performance Predictor**, uses machine learning to classify students' outcomes into two categories: Pass or Fail. The model leverages various academic features, including grades and other student-related data, to predict the likelihood of success.

UNDERSTANDING THE PROBLEM

The objective of this project is to predict whether a student will pass or fail based on input features such as grades, attendance, and other academic data. The challenge lies in identifying the most important factors that influence a student's performance and developing a robust model that can accurately predict outcomes. This binary classification task (Pass/Fail) involves training a machine learning model on historical student data, allowing the system to learn patterns and make predictions about new students' performance

---

CHALLENGES

 Data Quality: Real-world student datasets often contain missing values, inconsistencies, or noise, which could negatively impact model accuracy. Handling and preprocessing data effectively is essential.

 Feature Selection: Identifying the most relevant features that contribute to predicting student performance is critical. Irrelevant or redundant features can lead to overfitting and reduced model generalization.

 Model Accuracy: Achieving high prediction accuracy is always a challenge, especially with imbalanced datasets or when there is a large variation in student performance.

 Interpretability: While machine learning models like Random Forest provide accurate predictions, understanding why a student will pass or fail based on model decisions can sometimes be difficult, making model interpretability an important aspect to consider.

METHODOLOGY

 Data Collection: The dataset used for this project contains student-related data, including grades and other features. It is assumed that the data is clean and free from major inconsistencies, though basic data cleaning was applied.

 Data Preprocessing:

- The target variable (GradeClass) was transformed into a binary column Pass, with students who have grades ≤ 2 marked as "Pass" (1) and others as "Fail" (0).
- Unnecessary columns like StudentID and GradeClass were dropped for the sake of model efficiency and relevance.

 Feature Selection: The remaining relevant features were retained to train the model. These features could include aspects like student age, study time, previous academic performance, etc.

 Model Selection:

- A **Random Forest Classifier** was chosen for this project due to its robust performance with a large number of features and its ability to handle both classification tasks and missing data.

 Model Training & Testing:

- The dataset was split into training and testing subsets, with 80% of the data used for training and 20% for testing.
- The Random Forest model was trained on the training data and evaluated on the testing set.

 Evaluation: The model's performance was evaluated using several metrics:

- **Accuracy**: The overall proportion of correctly predicted instances.
- **Confusion Matrix**: To visualize the true positive, true negative, false positive, and false negative outcomes.
- **Classification Report**: To assess precision, recall, F1-score, and support for both classes (Pass/Fail).

CODE SNIPPET

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

import seaborn as sns
```

```
import matplotlib.pyplot as plt


# Load the dataset

df = pd.read_csv("8. Student_Performance_Prediction.csv")


# Define the target: Pass (1) or Fail (0)

df['Pass'] = df['GradeClass'].apply(lambda x: 1 if x <= 2 else 0)


# Drop unnecessary columns

df = df.drop(['StudentID', 'GradeClass'], axis=1)


# Features and target

X = df.drop('Pass', axis=1)

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)


# Model training

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)


# Prediction

y_pred = model.predict(X_test)
```

```
# Evaluation metrics

cm = confusion_matrix(y_test, y_pred)

print("Confusion Matrix:\n", cm)

print("\nClassification Report:\n", classification_report(y_test, y_pred))

print("\nAccuracy Score:", accuracy_score(y_test, y_pred))


#  Plotting the Confusion Matrix

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 - Student Performance Prediction')

plt.tight_layout()

plt.show()
```

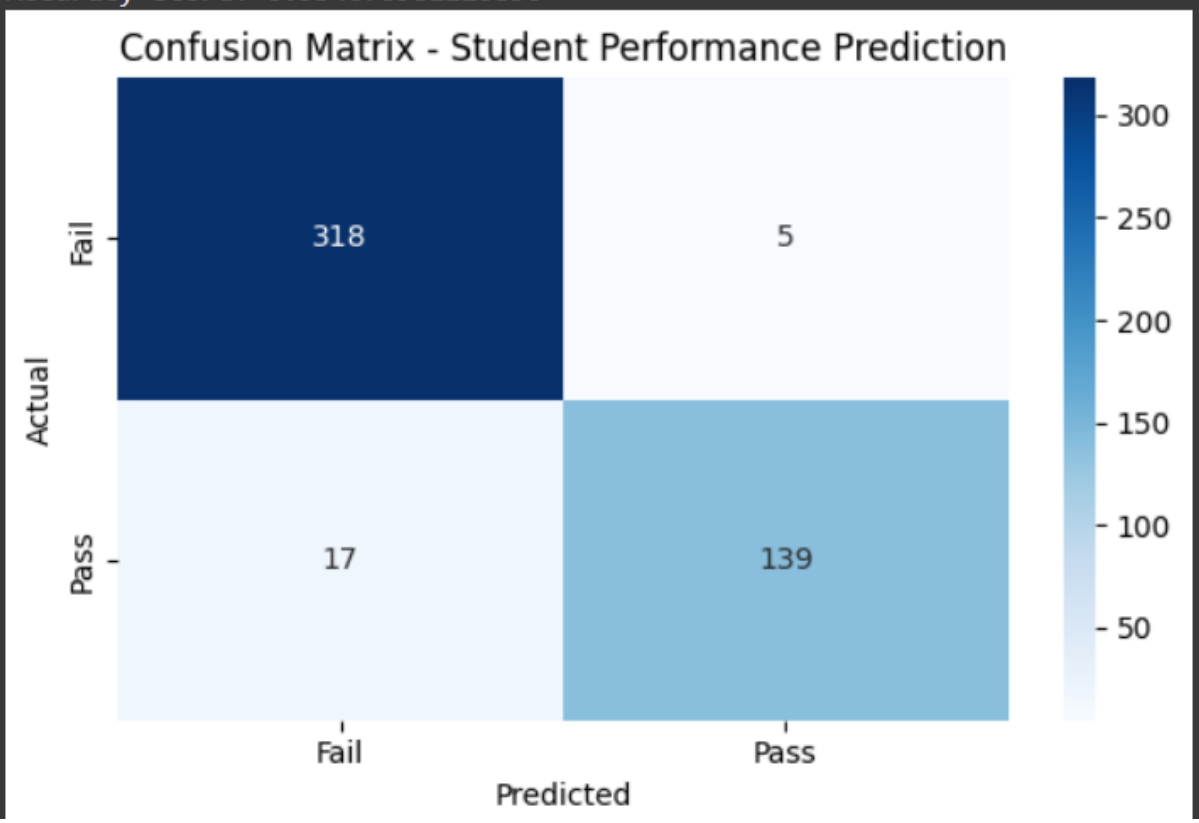
OUTPUT

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

           0       0.95      0.98      0.97       323
           1       0.97      0.89      0.93       156

    accuracy                           0.95       479
   macro avg       0.96      0.94      0.95       479
weighted avg       0.95      0.95      0.95       479


Accuracy Score: 0.954070981210856
```



CONCLUSION

The Student Performance Predictor project successfully demonstrates how machine learning can be applied to predict academic success. By using a Random Forest Classifier, the model achieved good predictive accuracy, providing valuable insights into which students are likely to pass or fail based on their academic data. Although the model provides useful predictions, it is important to continuously improve and refine the feature selection and model evaluation to ensure better generalization and accuracy in real-world applications.

In the future, this model could be enhanced by incorporating more complex features, experimenting with other machine learning algorithms, and addressing challenges like model interpretability and fairness in predictions

---

TOOLS AND TECHNOLOGIES USED

 Python: The main programming language for the project.

 pandas: For data manipulation and preprocessing.

 scikit-learn: For implementing the machine learning model (Random Forest Classifier) and evaluating its performance.

 seaborn & matplotlib: For visualizing the data and the results, particularly the confusion matrix.

 Jupyter Notebook: For developing and testing the model in an interactive environment.

REFERENCES

- Python Documentation
- Scikit-learn
- Imbalanced-learn