INTRODUCTION

TO

AI

PROJECT REPORT

STUDENT PERFORMANCE PREDICTION

-BY ROHIT PARASHAR

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 : ROHIT PARASHAR

Roll Number : 202401100400158 ,

section: c

Under the supervision of

"ABHISHEK SHUKLA"

KIET Group of Institutions, Ghaziabad

April, 2025

OBJECTIVE

The primary objective is to develop a predictive model that classifies student performance into two categories:

- Pass (1): Students likely to perform satisfactorily
- Fail (0): Students at risk of poor academic outcomes

This binary classification aims to assist educators and institutions in identifying students who may need additional support or resources to succeed.

DATASET OVERVIEW

The dataset includes a variety of features such as:

- Demographics: Age, Gender, Ethnicity
- Academic Background: Parental Education, Tutoring
- Behavioral Metrics: Study Time per Week, Absences, Parental Support
- Extracurricular Activities: Participation in Sports, Music, and Volunteering
- Performance Indicator: GPA (used to define the pass/fail outcome)

These attributes provide a comprehensive view of each student’s academic and personal environment.

REFERENCES

student performance prediction dataset: Kaggle

SMOTE: The Random Forest classifier performed extremely well in detecting credit card fraud when combined with proper preprocessing and SMOTE.

Feature scaling and class balancing significantly improved performance.

Future improvements could include trying other models like XGBoost or using hyperparameter tuning for better results.

Scikit-learn Documentation: The Random Forest classifier performed extremely well in detecting credit card fraud when combined with proper preprocessing and SMOTE.

Feature scaling and class balancing significantly improved performance.

Future improvements could include trying other models like XGBoost or using hyperparameter tuning for better results.

METHODOLOGY

A Random Forest Classifier is employed to model the relationship between these features and academic outcomes. The model is trained and validated using a train-test split approach. Performance is evaluated using key classification metrics such as:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

RESULT

After training and evaluating the model, the following results were obtained:

🧪 Classification Metrics:

Metric Fail (0)Pass (1)Overall

Precision

92.5%

93.4%

Recall

93.9%

91.7%

F1-score

93.2%

92.5%

Accuracy

-

-

92.9%

CONCLUSION

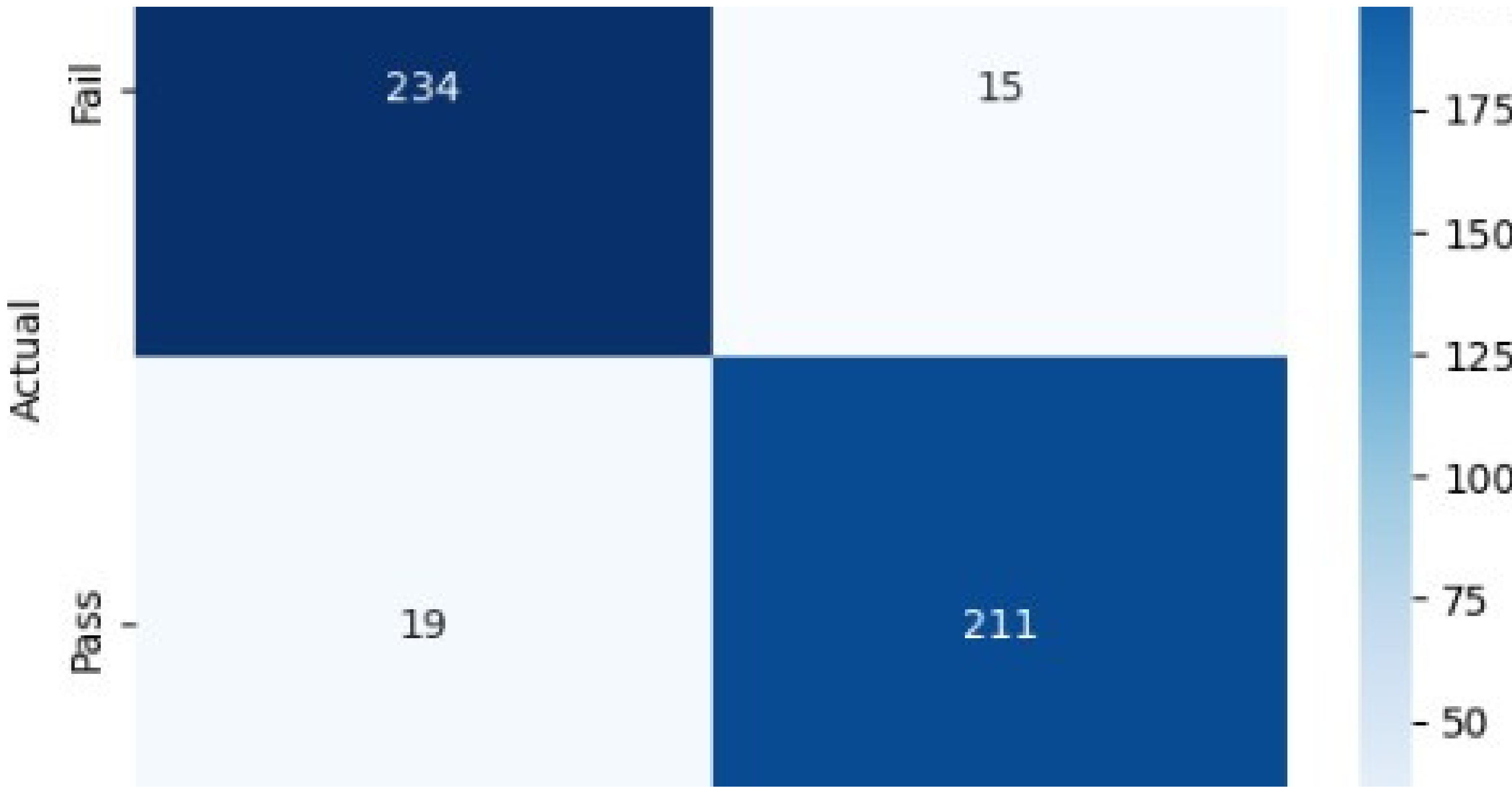
This project successfully demonstrates the use of machine learning techniques to predict student performance with high accuracy (~93%). The model provides meaningful insights into the key factors that influence academic success, such as:

- Study Time and Absences, which strongly correlate with performance
- Parental Support and Extracurricular Involvement, which also play a notable role

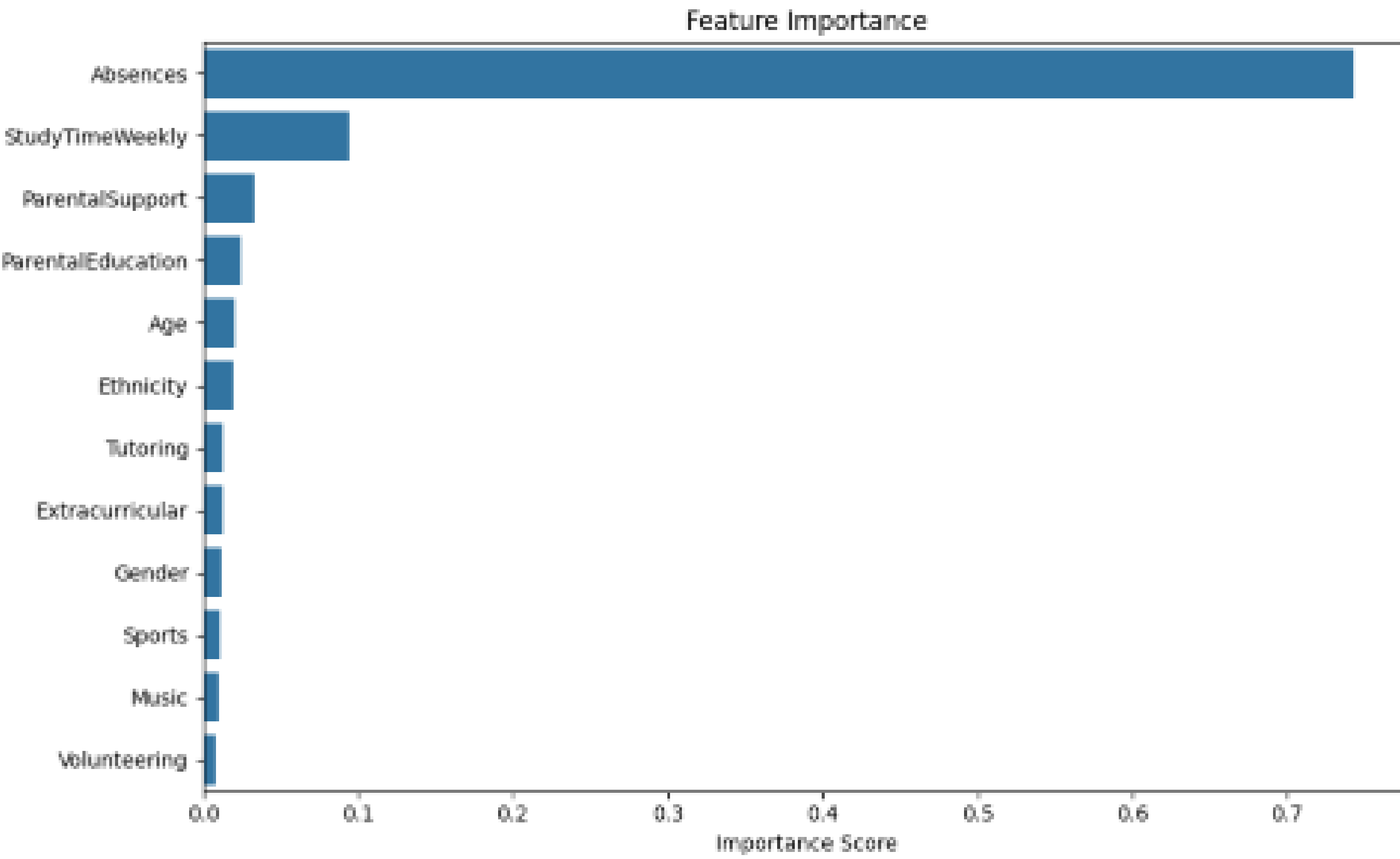
By leveraging these insights, educators can:

- Identify at-risk students early
- Implement targeted interventions
- Design personalized support strategies to enhance student outcomes

Future work may include expanding the dataset, incorporating more psychological or behavioral metrics, and testing different algorithms (e.g., XGBoost, Neural Networks) for potentially improved performance.



Fail
Pass
Actual
234
15
19
211
175
150
125
100
75
50



Feature Importance
Absences
StudyTimeWeekly
ParentalSupport
ParentalEducation
Age
Ethnicity
Tutoring
Extracurricular
Gender
Sports
Music
Volunteering
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Importance Score

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
import matplotlib.pyplot as plt
import seaborn as sns

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

# Define the binary target: 1 = Pass (GPA >= 2.0), 0 = Fail
df['Pass'] = np.where(df['GPA'] >= 2.0, 1, 0)

# Features to use
features = [
'Age', 'Gender', 'Ethnicity', 'ParentalEducation',
'StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport',
'Extracurricular', 'Sports', 'Music', 'Volunteering'
]

X = df[features]
y = df['Pass']

# Split into train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

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



```
# Predict and evaluate
y_pred = model.predict(X_test)
```

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

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
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()

# Feature Importance
importances = model.feature_importances_
feat_importance = pd.Series(importances,
index=features).sort_values(ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x=feat_importance, y=feat_importance.index)
plt.title("Feature Importance")
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.tight_layout()
plt.show()
```