A

As
se
sment Report

on

“STUDENT PERFORMACE PREDICTION”

submitted as partial fulfillment for the award

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE AIML

By

Name (202401100400174)

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

STUDENT PERFORMANCE PREDICTION

Student performance prediction involves using historical academic and personal data to forecast how a student is likely to perform in the future. Machine learning models analyze patterns in various factors such as:

- **Demographics** (age, gender, family background)
- **Academic records** (grades, attendance, study hours)
- **Behavioral data** (participation, discipline records)

The goal is to:

- Identify students at risk of underperforming
- Offer personalized support or interventions
- Improve teaching strategies and academic outcomes

Common algorithms used include **Linear Regression**, **Decision**

Decision Random Forest, and **Support Vector Machines**. These models can predict final grades, exam scores, or even pass/fail outcomes with considerable accuracy.

Overall, it's a powerful tool for data-driven decision-making in education.

METHODOLOGY

1. Problem Definition

Define the objective:

Predict student performance (e.g., final grade or score) based on various input features like study time, attendance, socio-economic factors, etc.

2. Data Collection

Use a dataset containing:

- Academic records (grades, attendance)
- Demographic info (age, gender, parental education)
- Behavioral aspects (study time, failures, support)

3. Data Preprocessing

- **Handle Missing Values**: Fill or drop missing data
- **Data Cleaning**: Remove duplicates, fix inconsistencies
- **Encoding**: Convert categorical variables using one-hot encoding or label encoding
- **Feature Scaling** (optional): Normalize or standardize if needed for some models

4. Exploratory Data Analysis (EDA)

- Analyze relationships between features and the target
- Visualize using histograms, boxplots , heatmaps
- Check for multicollinearity and distribution

5. Model Building

Train models:

- Linear Regression

- Random Forest Regressor
- Decision Tree
- XGBoost, etc.

6. Model Evaluation

Use metrics like:

- **MAE** (Mean Absolute Error)
- **MSE / RMSE** (Mean Squared Error)
- **R² Score** (explains variance)

Visualize:

- Actual vs Predicted
- Residual plots

7. Model Optimization

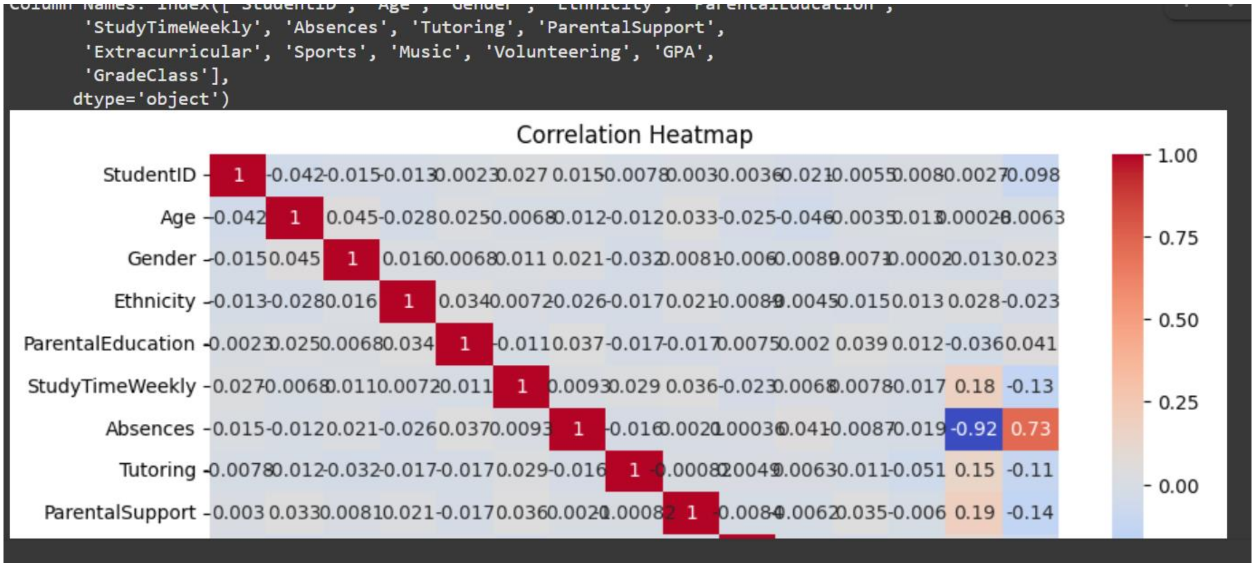
- Hyperparameter tuning (Grid Search, Randomized Search)
- Cross-validation for stable results

CODE

```python
 Step 1: Upload the CSV file
from google.colab import files
uploaded = files.upload()

# Step 2: Load the dataset
import pandas as pd

# Use the exact file name from the upload
df = pd.read_csv('8. Student Performance Prediction.csv')

# Step 3: Preview the dataset
print("First 5 rows:")
print(df.head())

print("\nDataset Info:")
df.info()

print("\nMissing Values:")
print(df.isnull().sum())

# Step 4: Basic EDA
import seaborn as sns
import matplotlib.pyplot as plt

# Distribution of target (assumes last column is the target, adjust if needed)
print("\nColumn Names:", df.columns)

# Visualize correlation
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()

# Step 5: Preprocess data
# Automatically detect categorical variables
categorical_cols = df.select_dtypes(include='object').columns
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

# Assuming the target is the last column
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# Step 6: Train/test split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 7: Train a Random Forest model
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

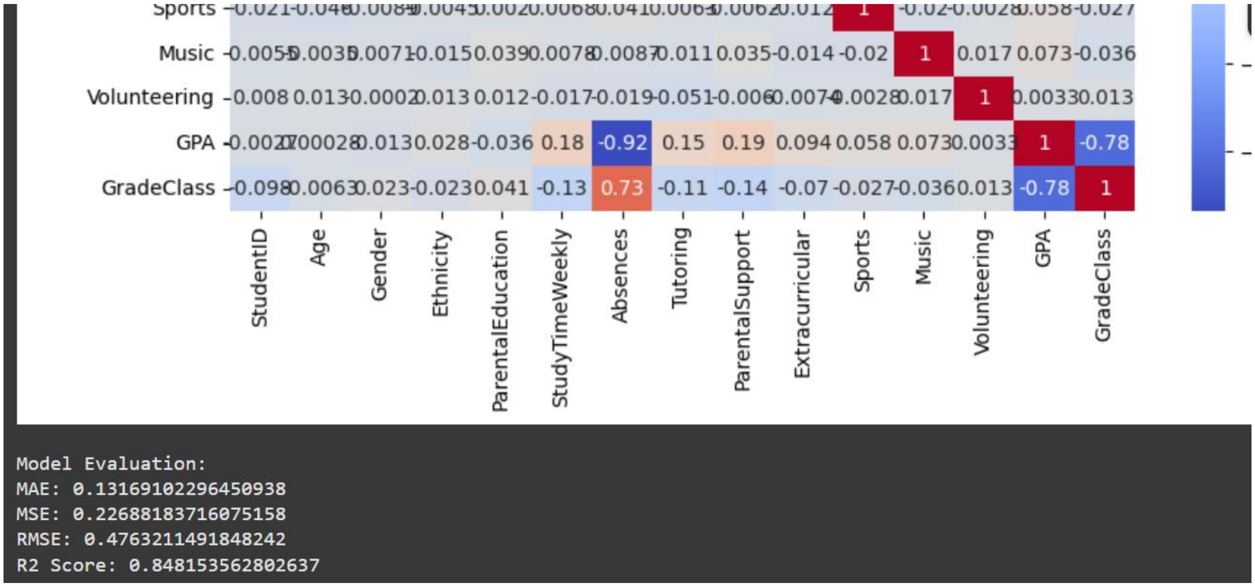
```python
# Step 8: Evaluate the model
y_pred = model.predict(X_test)

print("\nModel Evaluation:")
print("MAE:", mean_absolute_error(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))

print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred)))

print("R2 Score:", r2_score(y_test, y_pred))
```

OUTPUT



```
Column Names: Index(['StudentID', 'Age', 'Gender', 'Ethnicity', 'ParentalEducation',
       'StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport',
       'Extracurricular', 'Sports', 'Music', 'Volunteering', 'GPA',
       'GradeClass'],
      dtype='object')
```

Correlation Heatmap
StudentID 1 -0.042 -0.015 -0.013 0.0023 0.027 0.015 -0.0078 0.003 -0.0036 -0.021 -0.0055 0.008 -0.0027 -0.098
Age -0.042 1 0.045 -0.028 0.025 -0.0068 -0.012 -0.012 0.033 -0.025 -0.046 -0.0035 0.013 0.00028 -0.0063
Gender -0.015 0.045 1 0.016 0.0068 0.011 0.021 -0.032 0.0081 -0.006 -0.0089 0.0071 -0.0002 0.013 0.023
Ethnicity -0.013 -0.028 0.016 1 0.034 0.0072 -0.026 -0.017 0.021 -0.0089 -0.0045 -0.015 0.013 0.028 -0.023
ParentalEducation 0.0023 0.025 0.0068 0.034 1 -0.011 0.037 -0.017 -0.017 0.0075 0.002 0.039 0.012 -0.036 0.041
StudyTimeWeekly 0.027 -0.0068 0.011 0.0072 -0.011 1 0.0093 0.029 0.036 -0.023 0.0068 0.0078 -0.017 0.18 -0.13
Absences 0.015 -0.012 0.021 -0.026 0.037 0.0093 1 -0.016 0.0020 0.00036 0.041 -0.0087 -0.019 -0.92 0.73
Tutoring -0.0078 0.012 -0.032 -0.017 -0.017 0.029 -0.016 1 -0.00082 0.0049 0.0063 -0.011 -0.051 0.15 -0.11
ParentalSupport 0.003 0.033 0.0081 0.021 -0.017 0.036 0.0020 -0.00082 1 -0.0084 -0.0062 0.035 -0.006 0.19 -0.14
1.00
0.75
0.50
0.25
0.00



Sports -0.021 -0.046 -0.0089 -0.0045 0.002 0.0068 0.041 0.0063 -0.0062 -0.012 1 -0.02 -0.0028 0.058 -0.027
Music -0.0055 -0.0035 0.0071 -0.015 0.039 0.0078 -0.0087 -0.011 0.035 -0.014 -0.02 1 0.017 0.073 -0.036
Volunteering -0.008 0.013 -0.0002 0.013 0.012 -0.017 -0.019 -0.051 -0.006 -0.0074 -0.0028 0.017 1 0.0033 0.013
GPA -0.0027 0.00028 0.013 0.028 -0.036 0.18 -0.92 0.15 0.19 0.094 0.058 0.073 0.0033 1 -0.78
GradeClass -0.098 -0.0063 0.023 -0.023 0.041 -0.13 0.73 -0.11 -0.14 -0.07 -0.027 -0.036 0.013 -0.78 1
StudentID
Age
Gender
Ethnicity
ParentalEducation
StudyTimeWeekly
Absences
Tutoring
ParentalSupport
Extracurricular
Sports
Music
Volunteering
GPA
GradeClass

```
Model Evaluation:
MAE: 0.13169102296450938
MSE: 0.22688183716075158
RMSE: 0.4763211491848242
R2 Score: 0.848153562802637
```

REFRENCES

Kaggle :-https://www.kaggle.com/datasets

chatgpt