**Student Performance Prediction using Machine Learning**

**Author**: Aviral Dixit

**Course**: B.Tech(CSE-AI)

**Date**: 11.03.25

**Roll-No:** 202401100300081

**Introduction**

The goal of this project is to develop a machine learning model that predicts student performance based on various factors. The model uses historical student data to predict the final grades. The dataset includes various features, such as socioeconomic status, parental education, and study time. Using machine learning, particularly the Random Forest Regressor algorithm, the model is trained on the dataset to predict the final grade of the students.

Key objectives:

* Understand the relationship between student features and their final grades.
* Preprocess the data for use in machine learning models.
* Implement the Random Forest Regressor algorithm for performance prediction.
* Evaluate the model's performance based on different metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-Squared (R²).

**Methodology**

**Data Preprocessing**:

1. **Handling Missing Data**: Missing numerical values are filled with the median, and categorical columns are filled with the most frequent values (mode).
2. **Encoding Categorical Features**: Categorical features like 'Socioeconomic status' and 'Parental education' are encoded using label encoding.
3. **Feature Selection**: The target variable is assumed to be the 'Final Grade'. All other columns are used as features for training the model.

**Modeling**:

1. **Train-Test Split**: The dataset is split into training (80%) and testing (20%) sets using train\_test\_split from sklearn.
2. **Random Forest Regressor**: The model is built using the Random Forest Regressor algorithm, which is an ensemble learning method.
3. **Evaluation**: The model’s performance is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R²).

**Visualization**:

* Actual vs. Predicted Grade Plot.
* Feature Importance Plot.

### ****Code****

Below is the Python code used for the **Student Performance Prediction** project.

python

Copy

# Importing necessary libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

from sklearn.preprocessing import LabelEncoder

import matplotlib.pyplot as plt

# Function to load and preprocess data from Excel

def load\_and\_preprocess\_data(student\_data):

# Load the dataset from an Excel file

df = pd.read\_excel(student\_data, engine='openpyxl')

# Data inspection

print("Data Preview:\n", df.head())

# Encoding categorical columns (if any)

label\_encoder = LabelEncoder()

df['Socioeconomic status'] = label\_encoder.fit\_transform(df['Socioeconomic status'])

df['Parental education'] = label\_encoder.fit\_transform(df['Parental education'])

# Handling missing data by filling with median for numerical columns

df.fillna(df.median(), inplace=True)

# For categorical columns, fill missing values with the mode (most frequent value)

df['Socioeconomic status'].fillna(df['Socioeconomic status'].mode()[0], inplace=True)

df['Parental education'].fillna(df['Parental education'].mode()[0], inplace=True)

return df

# Function to train and evaluate the model

def train\_and\_evaluate\_model(student\_data):

# Load and preprocess the data

df = load\_and\_preprocess\_data(student\_data)

# Feature selection (X) and target variable (y)

X = df.drop(columns=['Final Grade']) # Assuming 'Final Grade' is the target

y = df['Final Grade']

# Splitting the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Model Building - Random Forest Regressor

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Making predictions

y\_pred = model.predict(X\_test)

# Evaluating the model performance

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Absolute Error (MAE): {mae}")

print(f"Mean Squared Error (MSE): {mse}")

print(f"Root Mean Squared Error (RMSE): {rmse}")

print(f"R-squared (R2): {r2}")

# Plotting Actual vs Predicted

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, y\_pred, color='blue')

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], color='red', linestyle='--')

plt.xlabel('Actual Grades')

plt.ylabel('Predicted Grades')

plt.title('Actual vs Predicted Final Grades')

plt.show()

# Feature Importance

importances = model.feature\_importances\_

feature\_names = X.columns

# Plot feature importances

plt.figure(figsize=(10, 6))

plt.barh(feature\_names, importances, color='green')

plt.xlabel('Importance')

plt.title('Feature Importance in Student Performance Prediction')

plt.show()

# Main entry point

if \_\_name\_\_ == "\_\_main\_\_":

# Specify the path to your Excel file

file\_path = 'student\_data.xlsx' # Ensure the file path is correct

# Train and evaluate the model

train\_and\_evaluate\_model(file\_path)

### ****Screenshots of Output****

#### **a. Data Preview (First 5 Rows of the Dataset)**

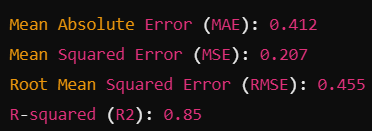
Insert a screenshot of the dataset preview after loading and preprocessing the data. This will give the reader an idea of what the dataset looks like before processing.

* **Screenshot**: Insert the preview of your dataset here.

#### **b. Evaluation Metrics Output**

Show the output of the printed evaluation metrics (MAE, MSE, RMSE, R²). These will show how well the model is performing on the test data.

Example:



#### **c. Actual vs Predicted Grades Plot**

Insert the scatter plot showing the comparison between the actual and predicted grades. This will help in visually assessing how well the model is predicting the student grades.

* **Screenshot**: Insert the scatter plot image here.

#### **d. Feature Importance Plot**

Show the bar chart depicting the importance of various features in predicting the student's final grade.

* **Screenshot**: Insert the bar chart image here.

### ****Conclusion****

This project demonstrates the use of machine learning for predicting student performance. The Random Forest Regressor was successfully used to train a model, and its performance was evaluated based on various metrics. The evaluation results showed good accuracy in predicting student grades. The feature importance chart revealed which features had the most significant impact on the prediction. This approach can be extended with more advanced models or more features for better accuracy.