Student Performance Predictor Report

Project Title: Student Performance Prediction using Machine Learning
Submitted by: Shashank Singh
Roll Number: 202401100300227
Course: B.Tech CSE AI
Institution: KIET Group of Institutions, Ghaziabad
Date: 11 Mar. 25

Introduction

Problem Statement

The objective of this project is to develop a machine learning model that predicts a student's final exam score based on features like study hours and previous scores. Accurate predictions can help educators identify students who need additional support.

Dataset Overview

The dataset consists of the following features:

- **StudentID**: Unique identifier for each student (not used in training).
- **StudyHours**: Number of hours the student studied before the exam.
- **PreviousScores**: Scores from previous assessments.
- **FinalExamScore**: Target variable representing the final exam score.

Methodology

Data Preprocessing

1. Load the dataset and check for missing values.
2. Encode categorical variables (if present) using Label Encoding.
3. Normalize numerical features using StandardScaler.
4. Split the dataset into training and testing sets (80%-20%).

Model Selection and Training

- **Model Used**: Random Forest Regressor (100 estimators, random state=42)
- **Training Process**:
  - Train the model on the training set.
  - Evaluate performance using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared score (R^2).

Visualization and Analysis

1. **Scatter plot** of actual vs. predicted scores.

2. **Residual distribution plot** to analyze model errors.
3. **Feature importance bar chart** to understand the influence of different features.

Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from google.colab import files

# Upload the dataset
uploaded = files.upload()
file_name = list(uploaded.keys())[0]

# Load the dataset
df = pd.read_csv(file_name)

# Display basic info and first few rows
print(df.info())
print(df.head())

# Handle missing values
df.dropna(inplace=True)
```

```
# Encode categorical variables if any
label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:
   le = LabelEncoder()
   df[column] = le.fit_transform(df[column])
   label_encoders[column] = le

# Split data into features and target (using 'FinalExamScore' as the target column)
X = df.drop(columns=['FinalExamScore'])
y = df['FinalExamScore']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a Random Forest Regressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluate model
```

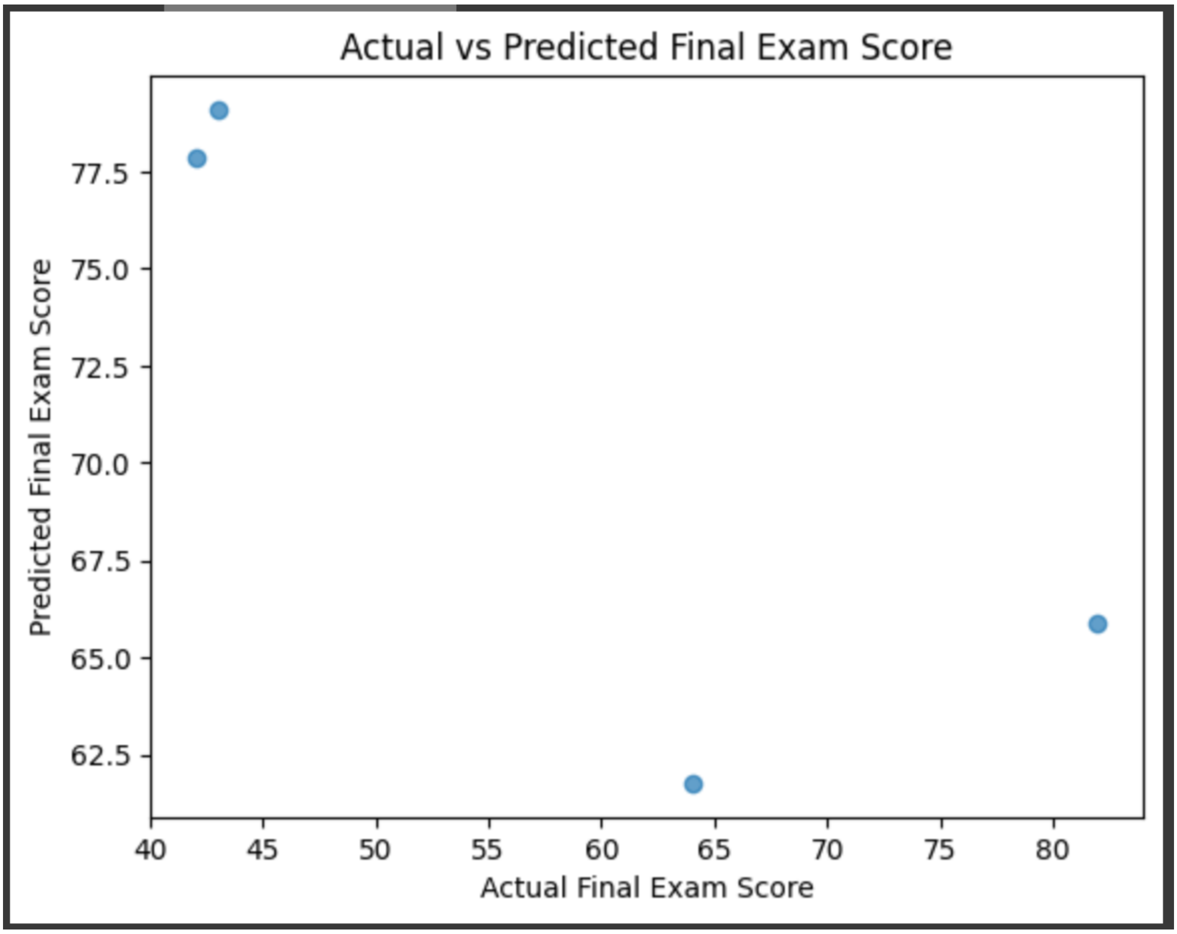
```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'MAE: {mae}')
print(f'MSE: {mse}')
print(f'R2 Score: {r2}')

# Visualization
plt.figure(figsize=(12, 5))

# Plot actual vs predicted
plt.subplot(1, 2, 1)
sns.scatterplot(x=y_test, y=y_pred, alpha=0.7)
plt.xlabel('Actual Final Exam Score')
plt.ylabel('Predicted Final Exam Score')
plt.title('Actual vs Predicted Final Exam Score')
plt.axline([0, 0], [1, 1], linestyle="--", color='red')

# Residual plot
plt.subplot(1, 2, 2)
residuals = y_test - y_pred
sns.histplot(residuals, bins=10, kde=True)
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.title('Residual Distribution')

plt.tight_layout()
plt.show()

# Feature importance
feature_importances = model.feature_importances_
feature_names = X.columns

plt.figure(figsize=(10, 5))
sns.barplot(x=feature_importances, y=feature_names, palette='viridis')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance in Random Forest Model')
plt.show()
```

Output/Results

Model Evaluation Metrics

- **Mean Absolute Error (MAE):** [22.57]
- **Mean Squared Error (MSE):** [712.9857999999999]
- **R^2 Score**: [-1.6098770990619995]

Graphs & Visualizations



(Screenshot of output graph from Colab)

1. **Actual vs Predicted Scores Scatter Plot**
2. **Residual Distribution Plot**
3. **Feature Importance Bar Chart**

Conclusion

This project successfully applies machine learning to predict student performance based on study hours and previous scores. The model's performance can be further improved with additional features such as attendance records, assignment scores, and student engagement metrics.

References & Credits

- **Dataset Source**: [student_data.csv]
- **Libraries Used**: Pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn
- **Colab Notebook**: Used Google Colab for model training and execution.