REPORT

Name: Preeti Singh
Roll No: 73
Course: AI
Institution: Kiet Group of Institutions

Introduction

In the current education system, it is necessary to predict students' performance

to determine students who need additional help and to maximize educational achievement. This project aims to predict students final exam scores based on various factors like study

time, past exam scores, attendance rate, parental education, access to the internet, and extracurricular activities. Through the analysis of these factors using machine learning, we can gain better insights into how they influence students' performance and take remedial action to enhance learning outcomes.

Methodology

The approach used to solve this problem consists of the following steps:

1. **Data Collection:** The dataset consists of various attributes such as Study Hours per Week, Attendance Rate, Past Exam Scores, Parental Education Level, Internet Access, Extracurricular Activities, and Final Exam Scores.
2. **Data Preprocessing:**
   - Handling missing values if any.
   - Encoding categorical variables such as Parental Education Level and Internet Access using one-hot encoding.
   - Standardizing numerical values if required.
3. **Feature Selection:** Selecting relevant independent variables like Study Hours per Week, Attendance Rate, and Past Exam Scores to predict the dependent variable (Final Exam Score).
4. **Model Selection:** A Linear Regression model is chosen to analyze and predict student performance.

5. **Model Training:** The dataset is split into training and testing sets (80%-20%). The model is trained using the training set.
6. **Prediction & Evaluation:** The trained model predicts final exam scores for test data. The performance of the model is evaluated using Mean Squared Error (MSE) and R-Squared (R^2) values.
7. **Visualization:** A scatter plot is created to visualize the relationship between study hours and exam scores with a regression line.

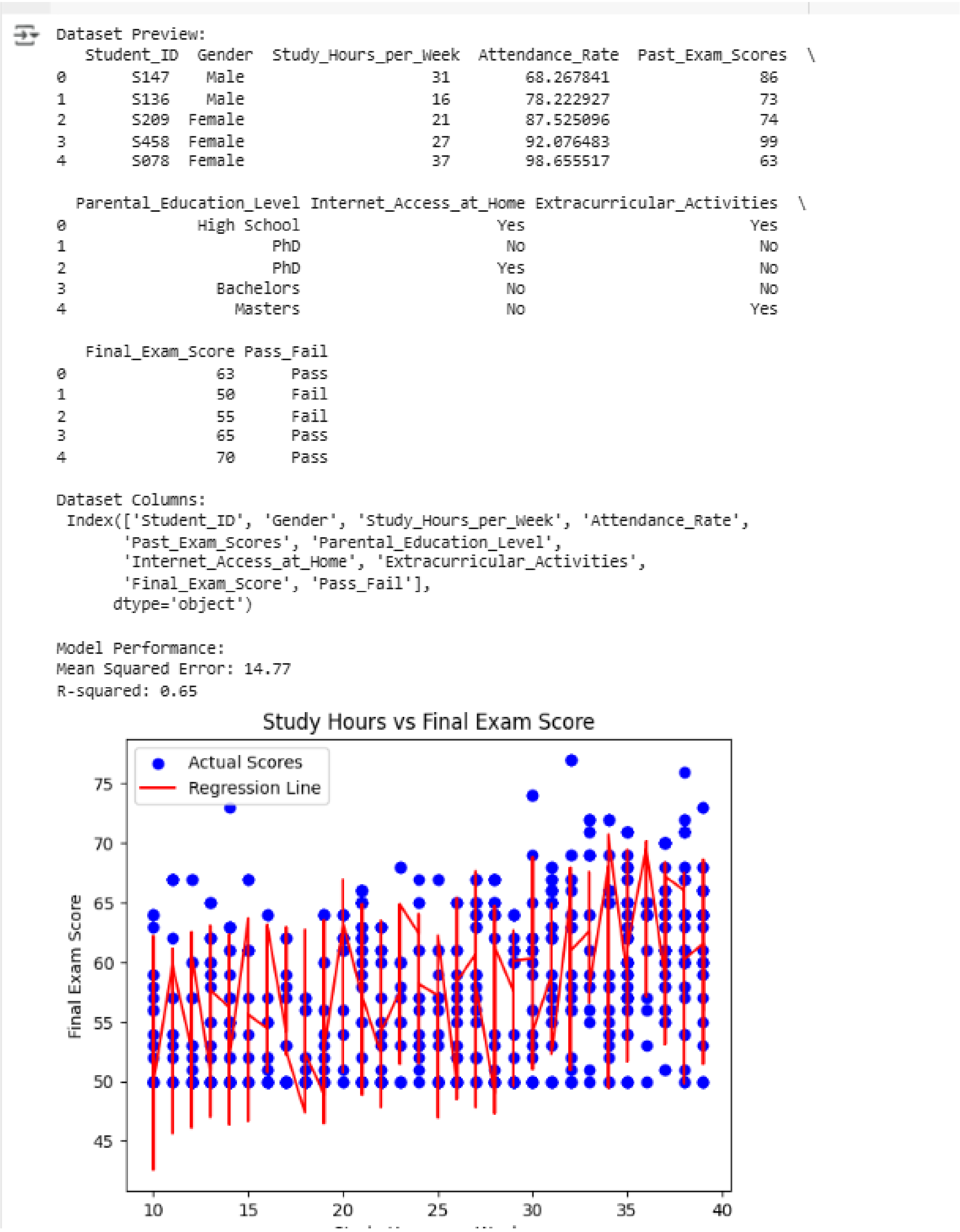
Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import OneHotEncoder

# Load the dataset
file_path = "/mnt/data/student_performance_dataset.csv"
df = pd.read_csv(file_path)

# Selecting relevant features and target variable
features = ['Study_Hours_per_Week', 'Attendance_Rate', 'Past_Exam_Scores']
categorical_features = ['Parental_Education_Level', 'Internet_Access_at_Home', 'Extracurricular_Activities']
target = 'Final_Exam_Score'

# One-hot encoding categorical variables
df_encoded = pd.get_dummies(df[categorical_features], drop_first=True)
```

```
# Combining numerical and encoded categorical features
X = pd.concat([df[features], df_encoded], axis=1)
y = df[target]

# Splitting data 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)

# Creating and training the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predicting exam scores
y_pred = model.predict(X_test)

# Model evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'\nModel Performance:')
print(f'Mean Squared Error: {mse:.2f}')
print(f'R-squared: {r2:.2f}')

# Visualization
plt.scatter(df['Study_Hours_per_Week'], df['Final_Exam_Score'], color='blue', label='Actual Scores')

# Sorting for a smooth regression line
sorted_indices = np.argsort(df['Study_Hours_per_Week'])
```

```
sorted_hours = df['Study_Hours_per_Week'].iloc[sorted_indices]
sorted_predictions = model.predict(X.iloc[sorted_indices])

plt.plot(sorted_hours, sorted_predictions, color='red', label='Regression Line')
plt.xlabel('Study Hours per Week')
plt.ylabel('Final Exam Score')
plt.legend()
plt.title('Study Hours vs Final Exam Score')
plt.show()
```

RESULT



```
Dataset Preview:
   Student_ID  Gender  Study_Hours_per_Week  Attendance_Rate  Past_Exam_Scores  \
0        S147    Male                    31        68.267841                86
1        S136    Male                    16        78.222927                73
2        S209  Female                    21        87.525096                74
3        S458  Female                    27        92.076483                99
4        S078  Female                    37        98.655517                63

  Parental_Education_Level Internet_Access_at_Home Extracurricular_Activities  \
0              High School                     Yes                        Yes
1                      PhD                      No                         No
2                      PhD                     Yes                         No
3                Bachelors                      No                         No
4                  Masters                      No                        Yes

   Final_Exam_Score Pass_Fail
0                63      Pass
1                50      Fail
2                55      Fail
3                65      Pass
4                70      Pass

Dataset Columns:
 Index(['Student_ID', 'Gender', 'Study_Hours_per_Week', 'Attendance_Rate',
       'Past_Exam_Scores', 'Parental_Education_Level',
       'Internet_Access_at_Home', 'Extracurricular_Activities',
       'Final_Exam_Score', 'Pass_Fail'],
      dtype='object')

Model Performance:
Mean Squared Error: 14.77
R-squared: 0.65
```

References/Credits

- Dataset: TAKEN FROM GOOGLE
- Libraries Used: Pandas, NumPy, Matplotlib, Scikit-learn