Student Score Prediction Project Report

Objective: This project demonstrates a simple **Linear Regression** model to predict student exam scores based on study hours. It highlights the complete **data analysis workflow** including data exploration, visualization, model building, evaluation, and prediction. Tools used include **Python, Pandas, NumPy, Matplotlib, and Scikit-Learn**.

Sample Dataset:

Study_Hours	Scores
4.37	39.77
9.56	84.54
7.59	68.77
6.39	47.57
2.4	20.5

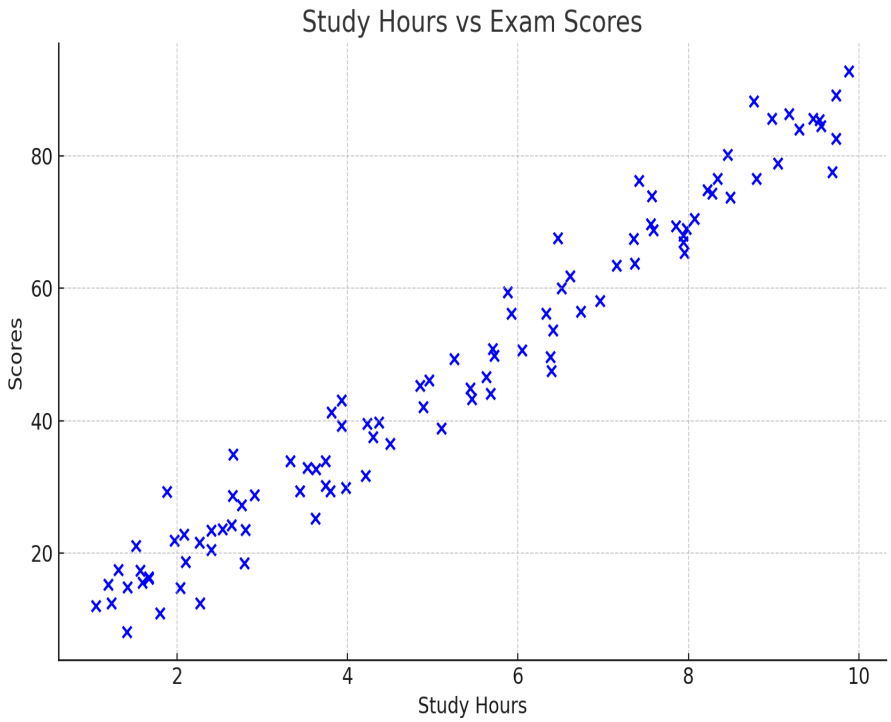
Project Code:

```
# Import libraries
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_absolute_error, mean_squared_error, r2_score

# Load dataset
df = pd.read_csv("student_scores.csv")

# Scatter plot
plt.scatter(df['Study_Hours'], df['Scores'], color='blue')
plt.title("Study Hours vs Exam Scores")
plt.xlabel("Study Hours")
plt.ylabel("Scores")
plt.show()

# Train-Test Split
X = df[['Study_Hours']]
y = df['Scores']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
print("MAE:", mean_absolute_error(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))

# Regression Line
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, y_pred, color='red', linewidth=2)
```

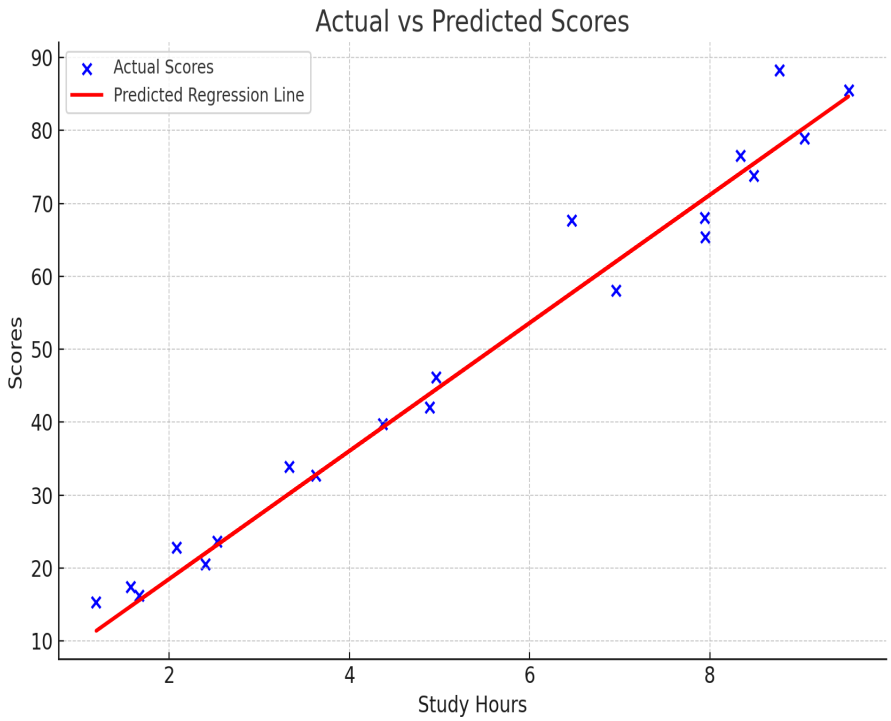
```
plt.show()

# New Prediction
new_hours = [[8.5]]
predicted_score = model.predict(new_hours)
print("Predicted Score for 8.5 hours:", predicted_score)
```

Outputs:

1. Scatter Plot: Study Hours vs Exam Scores



2. Regression Line: Actual vs Predicted Scores



3. Predicted Score: For 8.5 study hours = **75.54**