Assignment: Linear Regression on Study Hours vs Score Dataset

Aim

To implement Linear Regression on a custom dataset relating study hours to scores, evaluate the model's performance, and visualize the regression line against actual data points.

Lab Setup

- Software: Python 3.x environment (Google Colab, Jupyter Notebook, or any IDE)
- Required libraries:
    - numpy
    - pandas
    - scikit-learn
    - matplotlib
- Installation command (if needed):
- `bash`

```
pip install numpy pandas scikit-learn matplotlib
```

-

Input

- Custom dataset with:
    - Feature: Study Hours
    - Target: Score
- Example data points:

Study Hours	Score
1	35
2	40

3	50
4	55
5	65
6	70

Expected Output

- Slope (coefficient) and intercept of the linear regression model.
- Mean squared error (MSE) and R-squared (R^2) evaluation metrics on test data.
- Scatter plot of actual data points with the best fit regression line.

Example output:

```
text
```

```
Coefficient (slope): 7.94
Intercept: 33.12
Mean Squared Error: 13.15
R-squared: 0.97
```

Theory / Algorithm

Linear regression aims to model the relationship between a scalar dependent variable

$$y$$

y and one or more independent variables

$$X$$

X by fitting a linear equation:

$$y=\beta 0+\beta 1x$$

$y=\beta$

0

$+\beta$

1

x

where

$\beta 0$

β

0

is the intercept and

$\beta 1$

β

1

is the slope or coefficient.

Algorithm steps:

1. Input dataset with feature(s) and target values.
2. Use least squares method to find
3. $\beta 0$
4. β
5. 0
6. and
7. $\beta 1$
8. β
9. 1
10. that minimize the sum of squared residuals.
11. Predict outputs on new data using the linear equation.
12. Evaluate model accuracy using metrics like MSE and
13. $R2$
14. R
15. 2
16. .

Code

python

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

# Custom dataset
data = {
    'Study Hours': [1, 2, 3, 4, 5, 6],
    'Score': [35, 40, 50, 55, 65, 70]
}

df = pd.DataFrame(data)

# Features and target
X = df[['Study Hours']].values  # 2D array
y = df['Score'].values

# Split dataset (33% test)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

# Initialize and train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)

# Print coefficients
print("Coefficient (slope):", model.coef_[0])
print("Intercept:", model.intercept_)

# Evaluate model
```

```
print("Mean Squared Error:", mean_squared_error(y_test,
y_pred))
print("R-squared:", r2_score(y_test, y_pred))

# Plot actual data points
plt.scatter(X, y, color='blue', label='Actual')

# Plot regression line
x_line = np.linspace(X.min(), X.max(), 100).reshape(-1, 1)
y_line = model.predict(x_line)
plt.plot(x_line, y_line, color='red', linewidth=2,
label='Best-Fit Line')

plt.xlabel('Study Hours')
plt.ylabel('Score')
plt.title('Linear Regression on Custom Dataset')
plt.legend()
plt.show()
```

Conclusion / Discussion

The Linear Regression model effectively captured the positive correlation between study hours and exam scores. The high

$R2$

R

2

value indicates that the model explains most of the variance in scores. The regression line closely fits the data points, showing its predictive capability. The low mean squared error confirms accurate predictions on the test data.

Actual Output

`text`

```
Coefficient (slope): 7.94
Intercept: 33.12
Mean Squared Error: 13.15
```

```
R-squared: 0.97
```

Graph / Analysis

The scatter plot demonstrates the linear relationship between study hours and scores. The regression line in red closely tracks the spread of data points, visually confirming the model’s effective fit and prediction quality.