Double-click (or enter) to edit

Linear Regression (Scores Dataset) -Check for Null values

- Split into train and test data
- Predict the score percentage using Linear Regression.
- Evaluate the model using MSE - Mean Squared Error

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

```
path="/content/student_scores.csv"
```

```
df = pd.read_csv('/content/student_scores.csv')
```

```
df
```

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

```
print("Null values:")
print(df.isnull().sum())
```

```
Null values:
Hours     0
Scores    0
dtype: int64
```

```
X = df.drop("Scores", axis=1)
y = df["Scores"]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

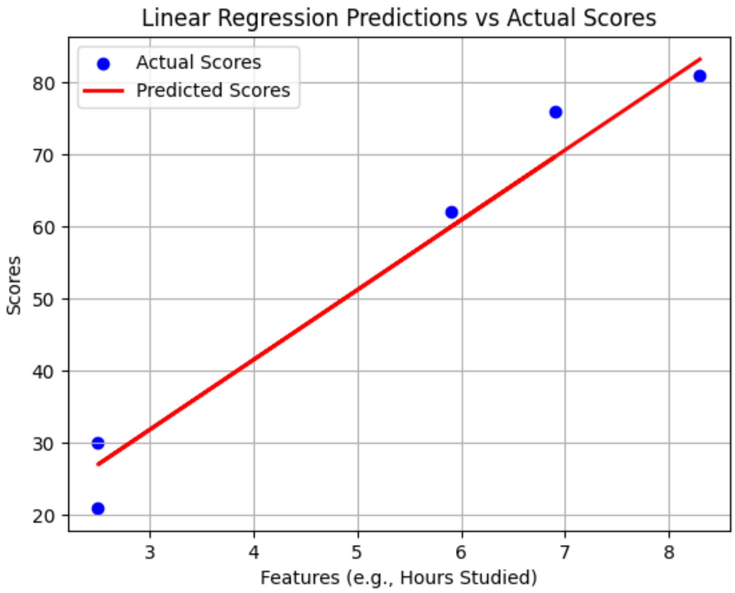
```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 18.943211722315272
```

```
plt.figure(figsize=(8, 6))
```

```
<Figure size 800x600 with 0 Axes>
<Figure size 800x600 with 0 Axes>
```

```
plt.scatter(X_test, y_test, color="blue", label="Actual Scores")
plt.plot(X_test, y_pred, color="red", linewidth=2, label="Predicted Scores")
plt.xlabel("Features (e.g., Hours Studied)")
plt.ylabel("Scores")
plt.title("Linear Regression Predictions vs Actual Scores")
plt.legend()
plt.grid(True)
plt.show()
```


Linear Regression Predictions vs Actual Scores
Actual Scores
Predicted Scores
Scores
Features (e.g., Hours Studied)