NAME : VAISHNAVI BIRAJDAR

PRN : RBT23CS028

BATCH : A2

PRACTICAL : 1
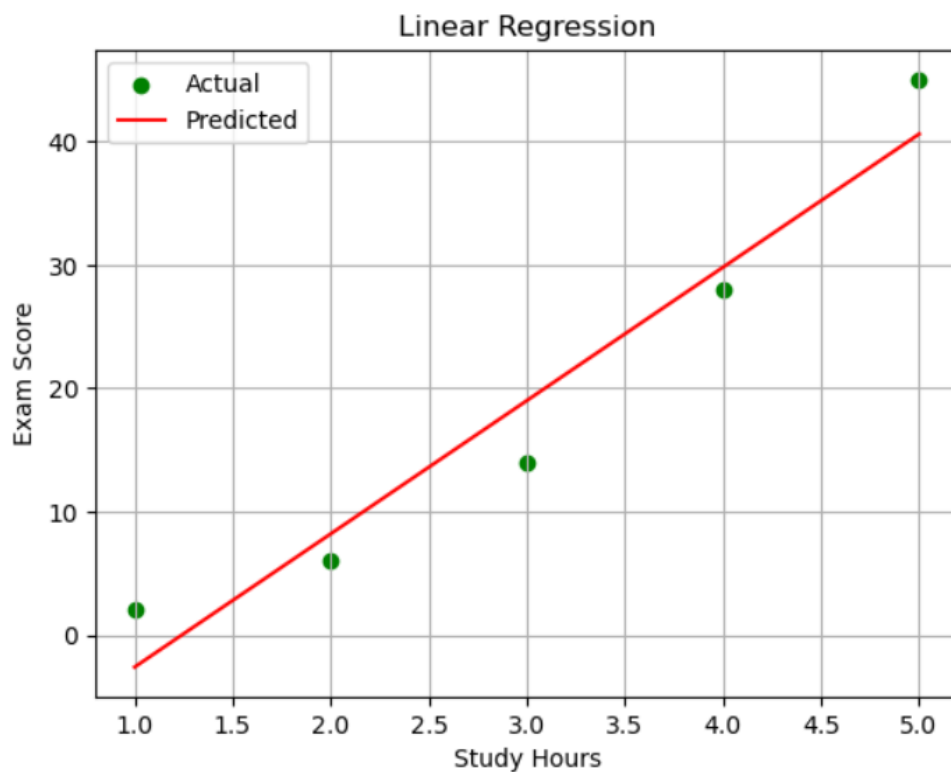
```python
import numpy as np
import matplotlib.pyplot as plt
```

```python
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

```python
X = np.array([10,20,30,40,50,60,70,80,90]).reshape(-1, 1)
y = np.array([25,32,55,69,75,82,89,90,95])
```

```python
model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)
```

```python
plt.scatter(X, y, color='green', label='Actual')
plt.plot(X, y_pred, color='red', label='Predicted')
plt.title('Linear Regression')
plt.xlabel('Study Hours')
plt.ylabel('Exam Score')
plt.legend()
plt.grid(True)
plt.show()
```

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
```

```python
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([2, 6, 14, 28, 45])
```

```python
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)
```
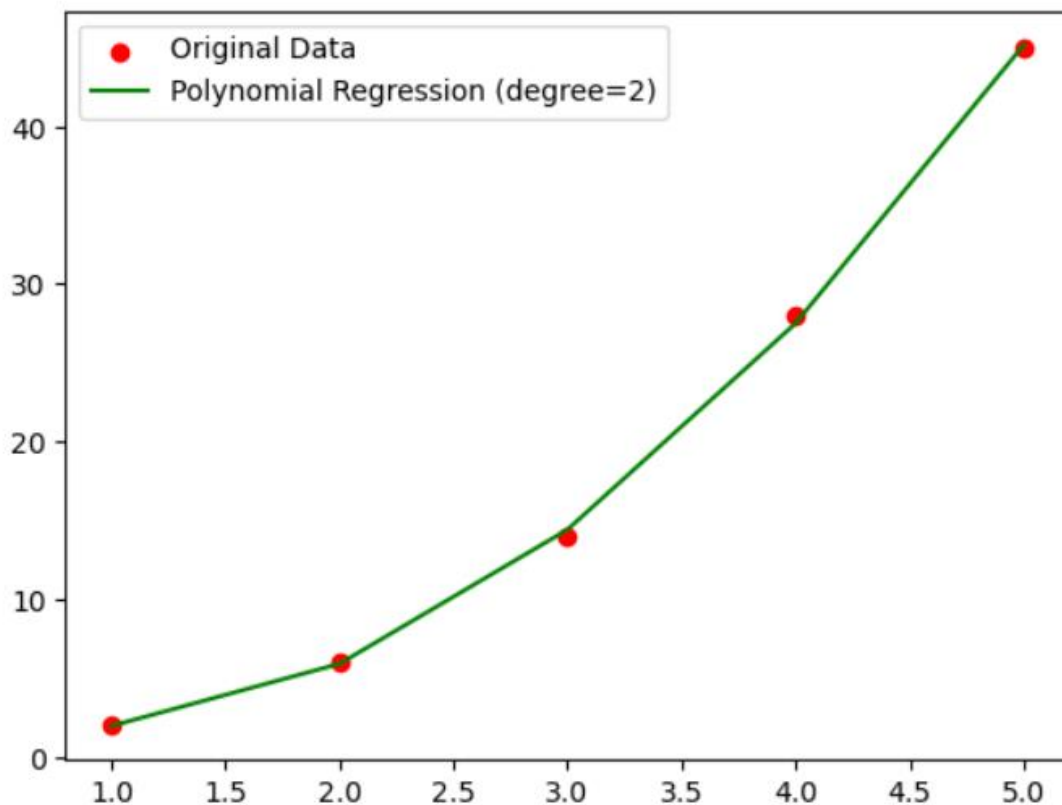
```python
poly_model = LinearRegression()
poly_model.fit(X_poly, y)
```

```
▾  LinearRegression   ⓘ ⓘ
LinearRegression()
```

```python
y_pred = poly_model.predict(X_poly)
```

```python
plt.scatter(X, y, color="red", label="Original Data")
plt.plot(X, y_pred, color="green", label="Polynomial Regression (degree=2)")
plt.legend()
plt.show()
```

```python
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```python
iris = load_iris()
X = iris.data[:, :2]
y = (iris.target == 0).astype(int)
```

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
log_model = LogisticRegression()
log_model.fit(X_train, y_train)
y_pred = log_model.predict(X_test)
```

```python
print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Logistic Regression Accuracy: 1.0
Confusion Matrix:
 [[20  0]
 [ 0 10]]
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        20
           1       1.00      1.00      1.00        10

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```