Name: Harsh Patel

Roll NO: A42

Experiment No:5

Aim: Write a python program to evaluate a Over fitting and Under fitting on custom dataset

Code:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Generate some synthetic data for demonstration
np.random.seed(0)
X = np.linspace(0, 10, 100)
y = 2 * X + np.random.normal(0, 1, 100)

print("Original Data:")
print("X:", X)
print("y:", y)

# Split the 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=0)

print("\nTraining Data:")
print("X_train:", X_train)
print("y_train:", y_train)

print("\nTesting Data:")
```

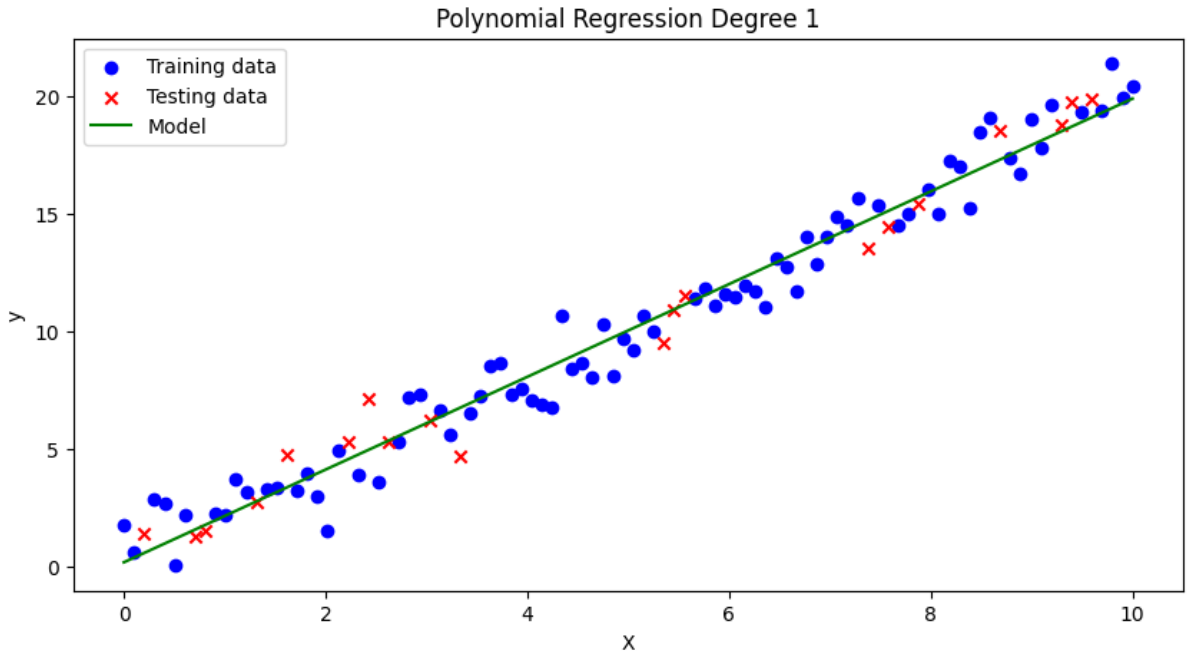
```
print("X_test:", X_test)
print("y_test:", y_test)

# Function to plot the data and model predictions
def plot_model(X_train, y_train, X_test, y_test, model, degree):
    plt.figure(figsize=(10, 5))
    plt.scatter(X_train, y_train, color='blue', label='Training data', marker='o')
    plt.scatter(X_test, y_test, color='red', label='Testing data', marker='x')

    X_range = np.linspace(0, 10, 100)
    X_poly = PolynomialFeatures(degree=degree).fit_transform(X_range.reshape(-1, 1))
    y_pred = model.predict(X_poly)

    plt.plot(X_range, y_pred, color='green', label='Model')
    plt.title(f'Polynomial Regression Degree {degree}')
    plt.xlabel('X')
    plt.ylabel('y')
    plt.legend()
    plt.show()

# Function to train a polynomial regression model of given degree
def train_polynomial_regression(X_train, y_train, X_test, y_test, degree):
    poly = PolynomialFeatures(degree=degree)
    X_poly_train = poly.fit_transform(X_train.reshape(-1, 1))
    X_poly_test = poly.transform(X_test.reshape(-1, 1))

    model = LinearRegression()
    model.fit(X_poly_train, y_train)
```

```
    train_rmse = mean_squared_error(y_train, model.predict(X_poly_train), squared=False)
    test_rmse = mean_squared_error(y_test, model.predict(X_poly_test), squared=False)

    plot_model(X_train, y_train, X_test, y_test, model, degree)
    print(f"Degree {degree} - Train RMSE: {train_rmse}, Test RMSE: {test_rmse}")

# Train polynomial regression models of different degrees
train_polynomial_regression(X_train, y_train, X_test, y_test, degree=1)  # Underfitting
train_polynomial_regression(X_train, y_train, X_test, y_test, degree=4)  # Balanced model
train_polynomial_regression(X_train, y_train, X_test, y_test, degree=15)  # Overfitting
```

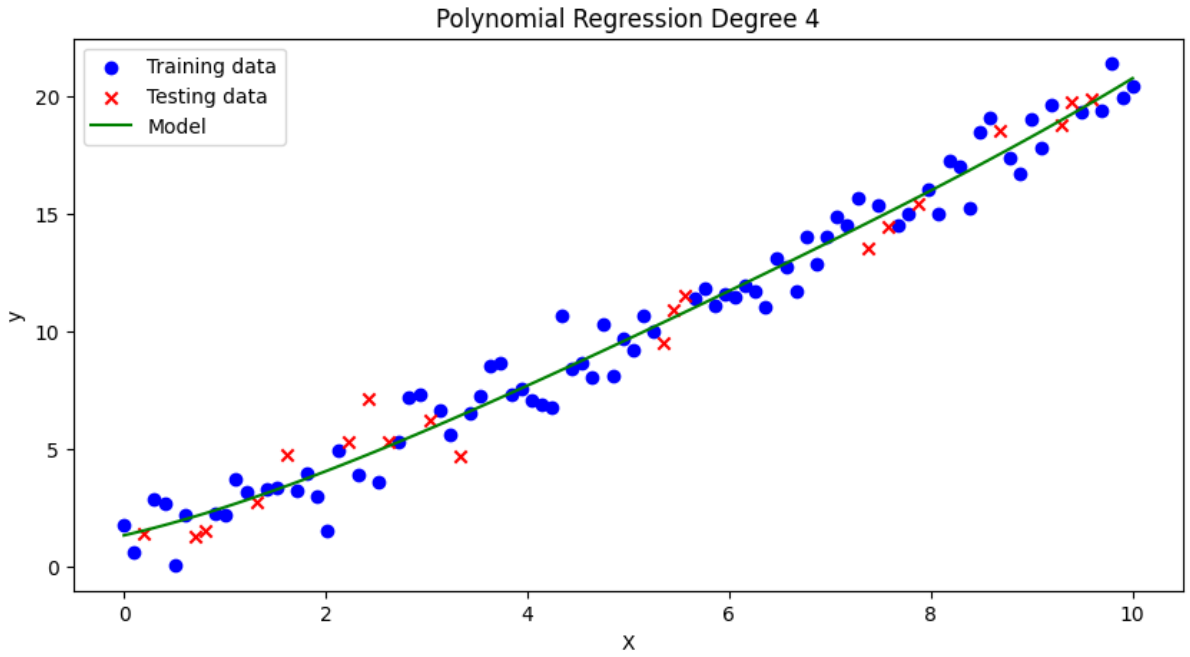
OUTPUT:

```
Original Data:
X: [ 0.          0.1010101   0.2020202   0.3030303   0.4040404   0.50505051
  0.60606061  0.70707071  0.80808081  0.90909091  1.01010101  1.11111111
  1.21212121  1.31313131  1.41414141  1.51515152  1.61616162  1.71717172
  1.81818182  1.91919192  2.02020202  2.12121212  2.22222222  2.32323232
  2.42424242  2.52525253  2.62626263  2.72727273  2.82828283  2.92929293
  3.03030303  3.13131313  3.23232323  3.33333333  3.43434343  3.53535354
  3.63636364  3.73737374  3.83838384  3.93939394  4.04040404  4.14141414
  4.24242424  4.34343434  4.44444444  4.54545455  4.64646465  4.74747475
  4.84848485  4.94949495  5.05050505  5.15151515  5.25252525  5.35353535
  5.45454545  5.55555556  5.65656566  5.75757576  5.85858586  5.95959596
  6.06060606  6.16161616  6.26262626  6.36363636  6.46464646  6.56565657
  6.66666667  6.76767677  6.86868687  6.96969697  7.07070707  7.17171717
  7.27272727  7.37373737  7.47474747  7.57575758  7.67676768  7.77777778
  7.87878788  7.97979798  8.08080808  8.18181818  8.28282828  8.38383838
  8.48484848  8.58585859  8.68686869  8.78787879  8.88888889  8.98989899
  9.09090909  9.19191919  9.29292929  9.39393939  9.49494949  9.5959596
  9.6969697   9.7979798   9.8989899  10.        ]
y: [ 1.76405235  0.60217741  1.38277839  2.84695381  2.6756388   0.03282313
  2.16220963  1.26278421  1.51294276  2.22878032  2.16424559  3.67649573
  3.18528015  2.74793764  3.27214606  3.36397736  4.72640231  3.22918517
  3.94943134  2.9842881   1.48741422  4.89604284  5.30888064  3.90429963
  7.11823947  3.59613938  5.29828377  5.2673616   7.18934487  7.32794463
  6.21555349  6.64078878  5.57686072  4.6858702   6.52077472  7.22705604
  8.50301795  8.67712732  7.28944086  7.57648513  7.03225512  6.86281035
  6.77857829 10.63764408  8.37923671  8.65283479  8.04013393 10.27243985
  8.08307185  9.68624962  9.20554354 10.6899328   9.99424537  9.52643852
 10.88090868 11.53944298 11.37964854 11.81762341 11.08284962 11.55645075
 11.44875167 11.96367916 11.71210624 11.00099012 13.10671907 12.7295322
 11.70313499 13.99813579 12.83007537 13.99133934 14.8705047  14.47241725
 15.68485523 13.51264893 15.35183659 14.46670506 14.4827382  14.97670589
 15.44602323 16.0157613  14.99646632 17.26446285 17.03131901 15.23143308
 18.45794916 19.06760635 18.55251694 17.39583274 16.70702516 19.03424971
 17.77864123 19.60628345 18.79413356 19.76451782 19.34626539 19.89849236
 19.40443941 21.38183009 19.92489189 20.40198936]
```

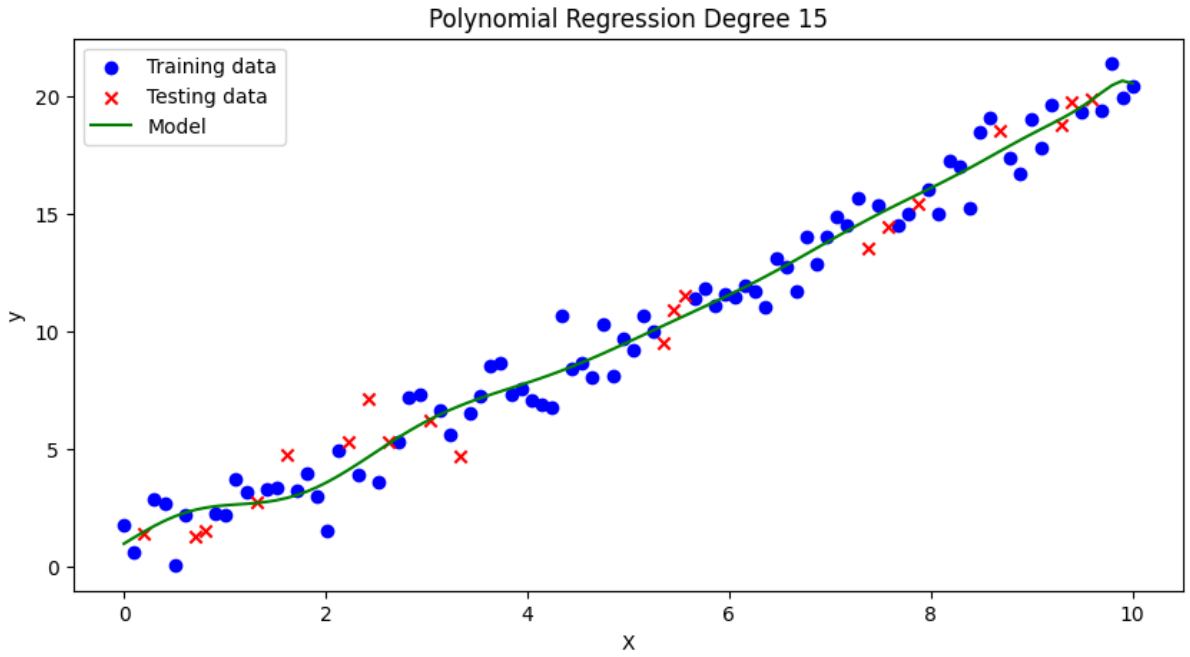
```
Training Data:
X_train: [ 4.34343434  6.26262626  0.3030303   7.17171717  4.54545455
4.84848485
  0.60606061 10.          8.28282828  7.67676768  6.06060606  8.08080808
  9.09090909  6.86868687  5.15151515  2.72727273  1.81818182  5.65656566
  6.36363636  7.47474747  0.1010101   6.16161616  4.24242424  4.14141414
  0.4040404   1.51515152  1.71717172  4.04040404  3.83838384  0.50505051
  9.19191919  5.95959596  0.          3.43434343  2.82828283  5.05050505
  1.11111111  3.53535354  2.32323232  5.25252525  1.01010101  3.13131313
  6.66666667  5.75757576  7.97979798  8.58585859  3.23232323  8.48484848
  1.41414141  8.98989899  1.91919192  2.92929293  4.94949495  9.7979798
  9.8989899   6.96969697  2.02020202  9.49494949  7.27272727  7.77777778
  2.52525253  3.73737374  8.18181818  4.64646465  3.93939394  6.56565657
  5.85858586  1.21212121  8.88888889  7.07070707  8.78787879  3.63636364
  2.12121212  8.38383838  0.90909091  9.6969697   6.76767677  6.46464646
  4.74747475  4.44444444]
y_train: [10.63764408 11.71210624  2.84695381 14.47241725  8.65283479
8.08307185
  2.16220963 20.40198936 17.03131901 14.4827382  11.44875167 14.99646632
 17.77864123 12.83007537 10.6899328   5.2673616   3.94943134 11.37964854
 11.00099012 15.35183659  0.60217741 11.96367916  6.77857829  6.86281035
  2.6756388   3.36397736  3.22918517  7.03225512  7.28944086  0.03282313
 19.60628345 11.55645075  1.76405235  6.52077472  7.18934487  9.20554354
  3.67649573  7.22705604  3.90429963  9.99424537  2.16424559  6.64078878
 11.70313499 11.81762341 16.0157613  19.06760635  5.57686072 18.45794916
  3.27214606 19.03424971  2.9842881   7.32794463  9.68624962 21.38183009
 19.92489189 13.99133934  1.48741422 19.34626539 15.68485523 14.97670589
  3.59613938  8.67712732 17.26446285  8.04013393  7.57648513 12.7295322
 11.08284962  3.18528015 16.70702516 14.8705047  17.39583274  8.50301795
  4.89604284 15.23143308  2.22878032 19.40443941 13.99813579 13.10671907
 10.27243985  8.37923671]

Testing Data:
X_test: [2.62626263 8.68686869 0.2020202  5.55555556 7.57575758 9.39393939
 1.61616162 7.37373737 5.45454545 9.5959596  5.35353535 9.29292929
 7.87878788 1.31313131 0.70707071 3.03030303 2.22222222 2.42424242
 3.33333333 0.80808081]
y_test: [ 5.29828377 18.55251694  1.38277839 11.53944298 14.46670506
19.76451782
  4.72640231 13.51264893 10.88090868 19.89849236  9.52643852 18.79413356
 15.44602323  2.74793764  1.26278421  6.21555349  5.30888064  7.11823947
  4.6858702   1.51294276]
```



Degree 1 - Train RMSE: 1.010797483665746, Test RMSE: 0.9794756420120921



Degree 4 - Train RMSE: 0.9287095161440835, Test RMSE: 0.9198303817054946




```
Degree 15 - Train RMSE: 0.9025545461403657, Test RMSE: 1.0617821798164102
```