10.Problem: Implement the non-parametric Locally Weighted Regression (LOWESS) algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

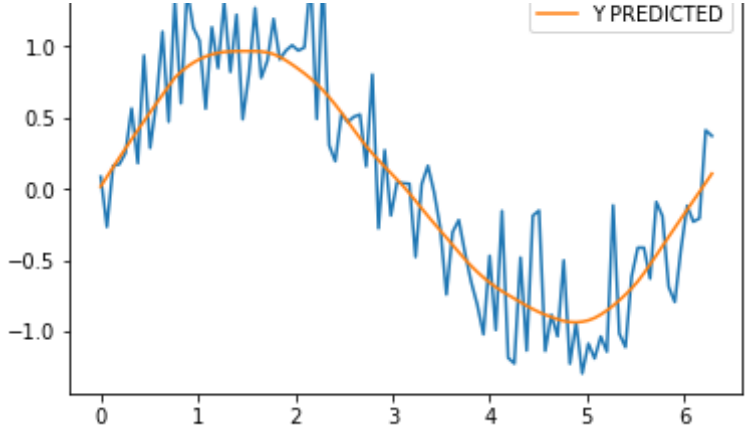
Example 1: Fitting to Give n Ploynomial Of degree 1 (Linear Regression)

In [23]:
```
# Author: Dr Thyagaraju GS , Context Innovations Lab

from math import ceil
import numpy as np
from scipy import linalg


def lowess(x, y, f= 2. / 3., iter=3):

    n = len(x) # Number of x  points
    r = int(ceil(f * n))  # Computing the residual of smoothing functio
ns
    h = [np.sort(np.abs(x - x[i]))[r] for i in range(n)] #
    w = np.clip(np.abs((x[:, None] - x[None, :]) / h), 0.0, 1.0)  # Wei
ght Function
    w = (1 - w ** 3) ** 3  # Tricube Weight Function
    ypred = np.zeros(n) # Initialisation of predictor
    delta = np.ones(n)  # Initialisation of delta

    for iteration in range(iter):
        for i in range(n):
```

```
            weights = delta * w[:, i] # Cumulative Weights
            b = np.array([np.sum(weights * y), np.sum(weights * y * x
)]) # Matrix B
            A = np.array([[np.sum(weights), np.sum(weights * x)],
                          [np.sum(weights * x), np.sum(weights * x * x
)]]) # Matrix A

            beta = linalg.solve(A, b) # Beta,Solution of AX= B equation

            ypred[i] = beta[0] + beta[1] * x[i]

        residuals = y - ypred   # Finding Residuals
        s = np.median(np.abs(residuals))  # Median of Residuals
        delta = np.clip(residuals / (6.0 * s), -1, 1)  # Delta
        delta = (1 - delta ** 2) ** 2   # Delta

    return ypred

if __name__ == '__main__':  # Main Function

    import math

    n = 100  # Number of data points

    #Case1: Sinusoidal Fitting
    x = np.linspace(0, 2 * math.pi, n)
    print(x)
    y = np.sin(x) + 0.3 * np.random.randn(n)

    #Case2 : Straight Line Fitting
    #x=np.linspace(0,2.5,n) # For Linear
    #y= 1 + 0.25*np.random.randn(n) # For Linear

    f = 0.25
    ypred = lowess(x, y, f=f, iter=3)

    import pylab as pl
    pl.clf()
    pl.plot(x, y, label='Y NOISY')
```

```
    pl.plot(x, ypred, label='Y PREDICTED')
    pl.legend()
    pl.show()
```

```
[ 0.         0.06346652  0.12693304  0.19039955  0.25386607  0.3173325
9
  0.38079911  0.44426563  0.50773215  0.57119866  0.63466518  0.6981317
  0.76159822  0.82506474  0.88853126  0.95199777  1.01546429  1.0789308
1
  1.14239733  1.20586385  1.26933037  1.33279688  1.3962634   1.4597299
2
  1.52319644  1.58666296  1.65012947  1.71359599  1.77706251  1.8405290
3
  1.90399555  1.96746207  2.03092858  2.0943951   2.15786162  2.2213281
4
  2.28479466  2.34826118  2.41172769  2.47519421  2.53866073  2.6021272
5
  2.66559377  2.72906028  2.7925268   2.85599332  2.91945984  2.9829263
6
  3.04639288  3.10985939  3.17332591  3.23679243  3.30025895  3.3637254
7
  3.42719199  3.4906585   3.55412502  3.61759154  3.68105806  3.7445245
8
  3.8079911   3.87145761  3.93492413  3.99839065  4.06185717  4.1253236
9
  4.1887902   4.25225672  4.31572324  4.37918976  4.44265628  4.5061228
  4.56958931  4.63305583  4.69652235  4.75998887  4.82345539  4.8869219
1
  4.95038842  5.01385494  5.07732146  5.14078798  5.2042545   5.2677210
2
  5.33118753  5.39465405  5.45812057  5.52158709  5.58505361  5.6485201
2
  5.71198664  5.77545316  5.83891968  5.9023862   5.96585272  6.0293192
3
  6.09278575  6.15625227  6.21971879  6.28318531]
```



Example 2: House Price Prediction

In [7]:
```
import pandas as pd
#df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learningdatabases/housing/housing.data',header=None, sep='\s+')
df = pd.read_csv('C:\\Users\\Dr.Thyagaraju\\Desktop\\Data\\housing.data.txt',header=None, sep='\s+')
df.columns = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS',
 'RAD','TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']
df.head(5)
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90

In [31]:
```
X = df[['RM']].values # Number of Rooms
Y = df[['MEDV']].values #
```

```
[ 6.575  6.421  7.185  6.998  7.147  6.43   6.012  6.172  5.631  6.004]
```

In [35]:
```
# Author: Dr Thyagaraju GS , Context Innovations Lab

from math import ceil
import numpy as np
from scipy import linalg


def lowess(x, y, f= 2. / 3., iter=3):

    n = len(x) # Number of x  points
    r = int(ceil(f * n))  # Computing the residual of smoothing functio
ns
    h = [np.sort(np.abs(x - x[i]))[r] for i in range(n)] #
    w = np.clip(np.abs((x[:, None] - x[None, :]) / h), 0.0, 1.0)  # Wei
ght Function
    w = (1 - w ** 3) ** 3  # Tricube Weight Function
    ypred = np.zeros(n) # Initialisation of predictor
    delta = np.ones(n)  # Initialisation of delta

    for iteration in range(iter):
        for i in range(n):
            weights = delta * w[:, i] # Cumulative Weights
            b = np.array([np.sum(weights * y), np.sum(weights * y * x
)]) # Matrix B
            A = np.array([[np.sum(weights), np.sum(weights * x)],
                          [np.sum(weights * x), np.sum(weights * x * x
)]]) # Matrix A

            beta = linalg.solve(A, b) # Beta,Solution of AX= B equation

            ypred[i] = beta[0] + beta[1] * x[i]

        residuals = y - ypred   # Finding Residuals
        s = np.median(np.abs(residuals))  # Median of Residuals
```

```
        delta = np.clip(residuals / (6.0 * s), -1, 1)  # Delta
        delta = (1 - delta ** 2) ** 2   # Delta

    return ypred

if __name__ == '__main__':  # Main Function

    import math

    n = 100  # Number of data points

    #Case1: Sinusoidal Fitting
    x = X[:100].ravel()   # House Room Data for modeling
    y = Y[:100].ravel()   # Noisy House Price data



    f = 0.66
    ypred = lowess(x, y, f=f, iter=3)  # Predicted House Price Data

    import pylab as pl
    pl.clf()
    pl.plot(x, y, label='Y NOISY')
    pl.plot(x, ypred, label='Y PREDICTED')
    pl.legend()
    pl.show()
```