```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


tou= 0.5 #tou is the bandwidth parameter and controls the rate at which w^{(i)} fall


#TRAINSET DATA
X_train = np.array(list(range(3, 33)) )
print(X_train)
X_train=X_train[:,np.newaxis]
print(X_train)
y_train = np.array([1,2,1,2,1,1,3,4,5,4,5,6,5,6,7,8,9,10,11,11,12,11,11,10,12,11,11,10,9,8])
print(y_train)
#the newaxis is used to increase the dimension of the existing array by one more dimension,
#1D array will become 2D array
#2D array will become 3D array
```

```
[ 3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 27 28 29 30 31 32]
[[ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
 [10]
 [11]
 [12]
 [13]
 [14]
 [15]
 [16]
 [17]
 [18]
 [19]
 [20]
 [21]
 [22]
 [23]
 [24]
 [25]
 [26]
 [27]
 [28]
 [29]
 [30]
 [31]
 [32]]
[ 1  2  1  2  1  1  3  4  5  4  5  6  5  6  7  8  9 10 11 11 12 11 11 10
 12 11 11 10  9  8]
```

```
X_train.shape
```

```
(30, 1)
```

```
y_train.shape
```

```
(30,)
```

```
#TESTSET DATA
X_test = np.array([i/10. for i in range(400)])
print(X_test)
X_test=X_test[:,np.newaxis]
print(X_test)
y_test·=·[]
print(y_test)
```

```
[13.6]
[13.7]
[13.8]
[13.9]
[14. ]
[14.1]
[14.2]
[14.3]
[14.4]
[14.5]
[14.6]
[14.7]
[14.8]
[14.9]
[15. ]
[15.1]
[15.2]
[15.3]
[15.4]
[15.5]
[15.6]
[15.7]
[15.8]
[15.9]
[16. ]
[16.1]
[16.2]
[16.3]
[16.4]
[16.5]
[16.6]
[16.7]
[16.8]
[16.9]
[17. ]
[17.1]
[17.2]
```

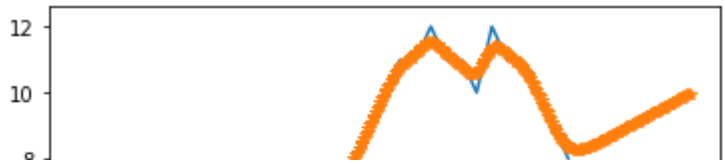
```
        [17.3]
        [17.4]
        [17.5]
        [17.6]
        [17.7]
        [17.8]
        [17.9]
        [18. ]
        [18.1]
        [18.2]
        [18.3]
        [18.4]
        [18.5]
        [18.6]
        [18.7]
        [18.8]
        [18.9]
        [19. ]
        [19.1]
        [19.2]
        [19.3]
        [19.4]
```

```
X_test.shape
```

```
    (400, 1)
```

```
for r in range(len(X_test)):
  wts=np.exp(-np.sum((X_train-X_test[r])**2,axis=1)/(2*tou)**2)
  W=np.diag(wts)
  factor1 = np.linalg.inv(X_train.T.dot(W).dot(X_train)) # find inverse of (X.T*W*X)
  parameters=factor1.dot(X_train.T).dot(W).dot(y_train) # final values of theta
  prediction=X_test[r].dot(parameters)
  y_test.append(prediction)
```

```
y_test = np.array(y_test)
plt.plot(X_train, y_train, '-')
plt.plot(X_test, y_test, '*')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```




```
y_test
```

```
[0.0,
 0.03336626264371145,
 0.06674706801578055,
 0.1001472174499804,
 0.1335729111313204,
 0.1670321239306116,
 0.200535074503284,
 0.23409480803998448,
 0.267727916205648,
 0.30145542069032344,
 0.33530384883957354,
 0.36930653008472697,
 0.4035051388610867,
 0.4379515011250285,
 0.4727096641561312,
 0.5078581984171009,
 0.543492649621489,
 0.5797279809990463,
 0.6167007310995465,
 0.6545704527462434,
 0.693519788900799,
 0.7337522855094878,
 0.7754867624870513,
 0.8189468148981478,
 0.8643438931998357,
 0.9118525598687867,
 0.9615771240574923,
 1.0135100942013495,
 1.067484843844596,
 1.12312742303328,
 1.1798151036605942,
 1.2366512275045582,
 1.292466294151718,
 1.345853292339315,
 1.395241020606524,
 1.4390033759235352,
 1.4755966657601793,
 1.503712160688679,
 1.5224279192847672,
 1.5313422688132512,
 1.5306709859738028,
 1.5212915100783366,
 1.504721383612303,
 1.4830253842585248,
 1.458656098663361,
 1.434243832296798,
 1.412360493984037,
 1.3952857287647398,
```

```
        1.3848014553464987,
        1.3820347781658313,
        1.3873618026780885,
        1.400378375643271,
        1.4199387073711072,
        1.444258314232065,
        1.4710726931623068,
        1.4978375839870588,
        1.521951834424793,
        1.540981405727344,
```