## Program 9:

```
import numpy as np
import matplotlib.pyplot as plt
def local_regression(x0, X, Y, tau):
 x0 = [1, x0]
 X = [[1, i] for i in X]
 X = np.asarray(X)
 xw = (X.T) * np.exp(np.sum((X - x0) ** 2, axis=1) / (-2 * tau))
 beta = np.linalg.pinv(xw @ X) @ xw @ Y @ x0
 return beta
def draw(tau):
 prediction = [local_regression(x0, X, Y, tau) for x0 in domain]
 plt.plot(X, Y, 'o', color='black')
 plt.plot(domain, prediction, color='red')
 plt.show()
X = np.linspace(-3, 3, num=1000)
domain = X
Y = np.log(np.abs(X ** 2 - 1) + .5)
draw(10)
draw(0.1)
draw(0.01)
draw(0.001)
```

# Program 8:

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import datasets
iris=datasets.load_iris()
print("Iris Data set loaded...")
x_train, x_test, y_train, y_test = train_test_split(iris.data,iris.target,test_size=0.1)
#random_state=0
for i in range(len(iris.target_names)):
    print("Label", i , "-",str(iris.target_names[i]))
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train, y_train)
y_pred=classifier.predict(x_test)
print("Results of Classification using K-nn with K=5 ")
for r in range(0,len(x_test)):
    print(" Sample:", str(x_test[r]), " Actual-label:", str(y_test[r])," Predicted-label:", str(y_pred[r]))

    print("Classification Accuracy :" , classifier.score(x_test,y_test));
```

# program 3:

```
import pandas as pd
df=pd.read_csv("trainingexamplesT.csv")
import numpy as np
features=np.array(df.iloc[:,0:-1])
target=np.array(df.iloc[:,-1])
specific_h=features[0].copy()
generic_h=[['?' for i in range (len(specific_h))] for i in range(len(specific_h))]
for i,h in enumerate(features):
 if target[i]=="Yes":
  for x in range(len(specific_h)):
   if h[x]!=specific_h[x]:
     specific_h[x]='?'
     generic_h[x][x]='?'
else:
    for x in range(len(specific_h)):
      if h[x]!=specific_h[x]:
        generic_h[x][x]=specific_h[x]
      else:
        generic_h[x][x]='?'
indices=[i for i,val in enumerate(generic_h) if val==['?','?','?','?','?','?'] ]
for i in indices:
 generic_h.remove(['?','?','?','?','?','?'])
print(specific_h)
print(generic_h)
```