

EEE 591 Machine Learning with deployment to FPGA

HOMEWORK 5
ASU ID: 1225713099
Praveen Paidi

1.

1) Select all data for the first class 0

2) For each feature determine the mean m and standard deviation s for data from each feature that falls into that class

3) Repeat for each class

Now you have the mean and standard deviation for each class within each feature this gives you $P(x_i | C_j)$ that is feature i with the value x_i could have been measured given class j .

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1(x-\mu)^2}{2\sigma^2}}$$

$$P(C/X) = \frac{P(X/C) \cdot P(C)}{P(X)}$$

The highest probability for each observation is picked after

$$P(C/X) = P(C) \prod_{i=1}^N P\left(\frac{x_i}{C}\right)$$

Basically:

- i. Selecting the iris data.
- ii. Splitting data
- iii. Finding mean and standard deviation
- iv. Probability of each class
- v. Naive Bayes classifier
- vi. Max probability Classifier
- vii. Training and Testing results.

2.

```
Question 2
Training samples 105
Train Misclassified samples 2
Training Accuracy 0.9809523809523809
Training errors at indices (array([83, 98], dtype=int64),) actual classificiton [2 1] pred myNB [1 2]

Testing samples 45
Test Misclassified samples 4
Testing Accuracy 0.9111111111111111
Testing errors at indices (array([ 3, 18, 30, 34], dtype=int64),) actual classificiton [2 1 1 2] pred myNB [1 2 2 1]
```

3.

```
Question 3
samples 150
Misclassified samples 6
Canned Algo Accuracy 0.96
Errors at indices (array([ 52,  70,  77, 106, 119, 133], dtype=int64),) actual classificiton [1 1 1 2 2 2] pred myNB [2 2 2
1 1 1]
```

There is difference between Canned algo and the normal algo made.

These results can be changed by changing the split size of the test and train as well as data observations number.

```
In [6]: from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
iris = datasets.load_iris()
X = iris.data[:,0:4] ## from this only take features 0,1,2,3
y = iris.target

#Splitting data
X_train,X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.3)

#intializing empty arrays
u= np.zeros((4,3))
s= np.zeros((4,3))

#indices finding
for c in range(3):
    for f in range(4):
        u[f,c] = X_train[np.where(Y_train==c),f].mean()
        s[f,c] = X_train[np.where(Y_train==c),f].std()

# prediction lists
ypredtrain=[]
ypredtest=[]

def PGauss(mu, sig, x):
    return np.exp(-np.power(x -mu, 2.) / (2 * np.power(sig, 2.) + 1e-300) )

for i in range(105):
    P=[1,1,1]
    for c in range(3):
        Pc = y.tolist().count(c)/105 ## P(c)
        for f in range(4):
            P[c] *= PGauss(u[f,c], s[f,c], X_train[i,f])
        P[c] *= Pc
    ypredtrain.append(P.index(max(P)))

for i in range(45):
    P=[1,1,1]
    for c in range(3):
        Pc = y.tolist().count(c)/45 ## P(c)
        for f in range(4):
            P[c] *= PGauss(u[f,c], s[f,c], X_test[i,f])
        P[c] *= Pc
    ypredtest.append(P.index(max(P)))

print('Question 2')
print('Training samples', len(Y_train))
Train_Misclassified=(Y_train != ypredtrain).sum()
print('Train Misclassified samples', Train_Misclassified )
Acc=accuracy_score(Y_train, ypredtrain)
print('Training Accuracy', Acc)

err=np.where(Y_train!=ypredtrain)
ypredtrain=np.array(ypredtrain)
print('Training errors at indices ', err, 'actual classificiton ', Y_train[err],' pred myNB ', ypredtrain[err])

print('\nTesting samples', len(Y_test))
Test_Misclassified=(Y_test != ypredtest).sum()
print('Test Misclassified samples', Test_Misclassified )
Acc=accuracy_score(Y_test, ypredtest)
print('Testing Accuracy', Acc)

err=np.where(Y_test!=ypredtest)
ypredtest=np.array(ypredtest)
print('Testing errors at indices ', err, 'actual classificiton ', Y_test[err],' pred myNB ', ypredtest[err])

from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X,y)
Y_Canpred = model.predict(X)
print('\nQuestion 3')
print('samples', len(y))
Misclassified=(y != Y_Canpred).sum()
print('Misclassified samples', Misclassified )
Acc=accuracy_score(y, Y_Canpred)
print('Canned Algo Accuracy', Acc)

err=np.where(y != Y_Canpred)
Y_Canpred=np.array(Y_Canpred)
print('Errors at indices ', err, 'actual classificiton ', y[err],' pred myNB ', Y_Canpred[err])
```

```
Question 2
Training samples 105
Train Misclassified samples 2
Training Accuracy 0.9809523809523809
Training errors at indices (array([83, 98], dtype=int64),) actual classificiton [2 1] pred myNB [1 2]

Testing samples 45
Test Misclassified samples 4
Testing Accuracy 0.9111111111111111
Testing errors at indices (array([ 3, 18, 30, 34], dtype=int64),) actual classificiton [2 1 1 2] pred myNB [1 2 2 1]

Question 3
samples 150
Misclassified samples 6
Canned Algo Accuracy 0.96
Errors at indices (array([ 52,  70,  77, 106, 119, 133], dtype=int64),) actual classificiton [1 1 1 2 2 2] pred myNB
[2 2 2 1 1 1]
```

In []: