IT 542: Pattern Recognition and Machine Learning

Assignment 2

ID:201811024

Name- Rajat Kumar

(1) Draw 200 samples each from N(10,20) and N(20,25). From these samples calculate sample means and sample variances. Consider these as the parameters for two new Gaussian distributions. Consider the apriori probabilities as (0.5,0.5), (0.3,0.7) and (0.7,0.3).Draw 40 random numbers from [5,20]. Classify each of these random numbers as sample from one of the two Gaussian distributions obtained from samples using three sets of apriori probabilities.

Code:

```
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
mu1,mu2,sigma1,sigma2=10,20,20,25
s1 = np.random.normal(mu1, sigma1, 200)
s2 = np.random.normal(mu2, sigma2, 200)
# Calclulating sample means
smu1=s1.mean()
smu2=s2.mean()
ssigma1=s1.var()
ssigma2=s2.var()
random_nos= np.random.randint(5, 20, 40)
pdf1=1/(ssigma1*np.sqrt(2*np.pi))*np.exp(-(random_nos-smu1)**2/(2*ssigma1**2))
pdf2=1/(ssigma2*np.sqrt(2*np.pi))*np.exp(-(random_nos-smu2)**2/(2*ssigma2**2))
p1,p2=0.5,0.5
p3,p4=0.3,0.7
p5,p6=0.7,0.3
set1=[]
set2=[]
set3=[]
for i,j in zip(pdf1,pdf2):


  A=i*p1
  B=j*p2
```

```
    C=i*p3
    D=j*p4
    E=i*p5
    F=j*p6
    if(A>=B):
        set1.append('class:pdf1')
    else:
        set1.append('class:pdf2')
    if(C>=D):
        set2.append('class:pdf1')
    else:
        set2.append('class:pdf2')
    if(E>=F):
        set3.append('class:pdf1')
    else:
        set3.append('class:pdf2')
print('Classes of 40 Samples using '+str(p1)+','+str(p2)+' as apriori probabilities:\n',*set1,sep='\n')
print('\n')
print('Classes of 40 Samples using '+str(p3)+','+str(p4)+' as apriori probabilities:\n',*set2,sep='\n')
print('\n')
print('Classes of 40 Samples using '+str(p5)+','+str(p6)+' as apriori probabilities:\n',*set3,sep='\n')
```

Output:

```
Classes of 40 Samples using 0.5,0.5 as apriori probabilities:

class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
```

class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1

Classes of 40 Samples using 0.3,0.7 as apriori probabilities:

class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2
class:pdf2

Classes of 40 Samples using 0.7,0.3 as apriori probabilities:

class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1

```
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
class:pdf1
```

(2) Consider 3 Gaussian distribution N(5,20), N(10,15) and N(20,25). Repeat the experiment-1 with 40 random numbers drawn from [0,15].

Code:

```
import numpy as np
import matplotlib.pyplot as plt
from random import randint
p1=0.3
p2=0.5
p3=0.2
samples1=np.random.normal(5,20,size=200)
samples2=np.random.normal(10,15,size=200)
samples3=np.random.normal(20,25,size=200)
samples1_mean=np.mean(samples1)
samples2_mean=np.mean(samples2)
samples3_mean=np.mean(samples3)
samples1_var=np.var(samples1)
samples2_var=np.var(samples2)
samples3_var=np.var(samples3)
samples_40=np.random.randint(0,15,size=40)
pdf1=1/(samples1_var*np.sqrt(2*np.pi))*np.exp(-(samples_40-samples1_mean)**2/(2*samples1_var**2))
pdf2=1/(samples2_var*np.sqrt(2*np.pi))*np.exp(-(samples_40-samples2_mean)**2/(2*samples2_var**2))
pdf3=1/(samples3_var*np.sqrt(2*np.pi))*np.exp(-(samples_40-samples3_mean)**2/(2*samples3_var**2))
set=[]
for i,j,k in zip(pdf1,pdf2,pdf3):
  A = i * p1
  B = j * p2
  C = k * p3
  if(A>=B and A>=C):
    set.append('class:N(5,20)')
```

```
    elif(B>=A and B>=C):
        set.append('class:N(10,15)')
    else:
        set.append('class:N(20,25)')
print('Classes of 40 Samples using '+str(p1)+','+str(p2)+','+str(p3)+' as apriori probabilities:\n',*set,sep='\n')
```

Output:

```
Classes of 40 Samples using 0.3,0.5,0.2 as apriori probabilities:

class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
class:N(10,15)
```