

a)

$$1.a) P(x, y=1) = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}$$

$$P(x, y=0) = \frac{1}{\sigma_0 \sqrt{2\pi}} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}}$$

Kemal Yağız

Daşkıran

040210034

$\mu_1 = 2, \sigma_1 = 1$ ve $\mu_0 = 0, \sigma_0 = 1$ olduğunu biliyoruz.

Bayes sınıflandırıcısının ideal olması için karar eşiğinde $P(x, y=0) = P(x, y=1)$ eşitliği sağlanmalıdır.

$$\frac{1}{1 \cdot \sqrt{2\pi}} \cdot e^{-\frac{(x-2)^2}{2 \cdot 1^2}} = \frac{1}{1 \cdot \sqrt{2\pi}} \cdot e^{-\frac{(x-0)^2}{2 \cdot 1}}$$

$$e^{-\frac{(x-2)^2}{2}} = e^{-\frac{x^2}{2}} \xrightarrow{\ln} \frac{-(x-2)^2}{2} = -\frac{x^2}{2}$$

$$(x-2)^2 = x^2, x^2 - 4x + 4 = x^2, 4x = 4 \quad \boxed{x=1}$$

Bayes sınıflandırıcımızda karar eşiği $x=1$ 'de olmalıdır.

$$y = \begin{cases} 1, & x > 1 \\ 0, & x < 1 \end{cases}$$

b) (10 puan) BİR sınıfa ait dağılımdan 200 ölçüm değeri üretiniz.

c) (10 puan) SIFIR sınıfa ait dağılımdan 200 ölçüm değeri üretiniz.

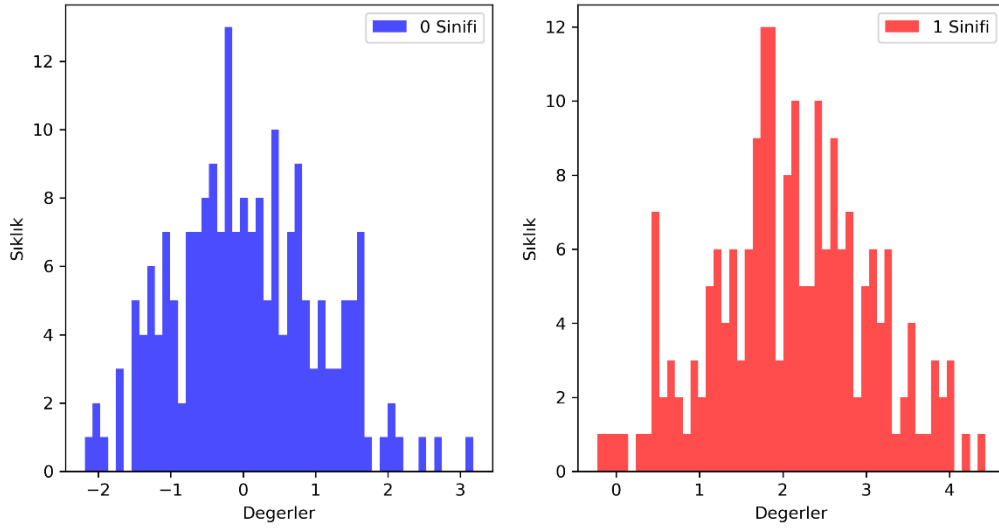
Ödevin bu kısmını oluşturduğum bu kod ile gerçekleştirdim.

```
1  from random import gauss
2  import matplotlib.pyplot as plt
3  import csv
4
5  """
6  Sınıf başına 200 adet 0 ve 1 sınıflı gauss dağılımlı bir veri havuzu oluşturur.
7
8  Sonuçları csv dosyasına yazar.
9  """
10
11  n_of_samples = 200
12
13  class_0, class_1 = [], []
14  avg_0_class, sd_0_class = 0, 1
15  avg_1_class, sd_1_class = 2, 1
16
17  for i in range (n_of_samples):
18      class_0.append(gauss(avg_0_class, sd_0_class))
19      class_1.append(gauss(avg_1_class, sd_1_class))
20
21  plt.figure(figsize=(10,5))
22  plt.suptitle(f"Her bir sınıf için {n_of_samples} örnek değer oluşturulmuştur.")
23  plt.subplot(1,2,1)
24  plt.hist(class_0, bins = 50, alpha=0.7, color="Blue", label="0 Sinifi")
25  plt.xlabel("Değerler")
26  plt.ylabel("Sıklık")
27  plt.legend()
28
29  plt.subplot(1,2,2)
30  plt.hist(class_1, bins = 50, alpha=0.7, color="Red", label="1 Sinifi")
31  plt.xlabel("Değerler")
32  plt.ylabel("Sıklık")
33  plt.legend()
34  plt.savefig('./q_histograms.png', dpi=300)
35
36
37  with open("./q_0_class.csv", 'w', newline='') as csvfile:
38      writer = csv.writer(csvfile)
39      writer.writerow(class_0)
40
41  with open("./q_1_class.csv", 'w', newline='') as csvfile:
42      writer = csv.writer(csvfile)
43      writer.writerow(class_1)
44
45
```

Ve Her bir sınıf için 200 adet örnek veri oluşturdum.

Her çalıştırmamda farklı değerler aldığım için sonuç değişecek olsa da, örnek çıktıları bu şekilde olmaktadır:

Her bir sınıf için 200 örnek değer oluşturulmuştur.



d) (10 puan) b ve c de üretilen öznitelik değerlerini kullanarak BİR ve SIFIR sınıflarının herbirisi için ortalama değer ve standart sapmayı hesaplayınız.

```
1 import pandas as pd
2 import numpy as np
3 from math import sqrt
4 import matplotlib.pyplot as plt
5
6
7
8 Gauss.py'dan elde edilen csv dosyalarını okuyarak bu değerlerden Confusion Matrix, Precision, Recall ve F1 değerlerini üretir.
9
10
11
12 df1 = pd.read_csv("./odev_1/q_0_class.csv", header=None)
13 df2 = pd.read_csv("./odev_1/q_1_class.csv", header=None)
14 data1 = df1.values.flatten()
15 data2 = df2.values.flatten()
16
17 def ortalama(data):
18     return sum(data)/len(data)
19
20 def standart_sapma(data):
21     ort = ortalama(data)
22     x = 0
23     for item in data:
24         x += (item - ort)**2
25     x /= (len(data)-1)
26     return sqrt(x)
27
28 #Numpy ve kendi fonksiyonlarının kullanılarak ortalamanın elde edilmesi ve printlenmesi
29 print(f"\n(Numpy Versiyon) 1 sınıflı verilerin ortalama değeri: {np.average(data2)}\n0 sınıflı verilerin ortalama değeri: {np.average(data1)}")
30 print(f"\n(Ben Versiyon) Oluşturulan 1 sınıflı verilerin ortalama değeri: {ortalama(data2)}\n0 sınıflı verilerin ortalama değeri: {ortalama(data1)}")
31
32 #Numpy ve kendi fonksiyonlarının kullanılarak standart sapmanın elde edilmesi ve printlenmesi
33 print(f"\n(Numpy Versiyon) 1 sınıflı verilerin standart sapma değeri: {np.std(data2)}\n0 sınıflı verilerin standart sapma değeri: {np.std(data1)}")
34 print(f"\n(Ben Versiyon) Oluşturulan 1 sınıflı verilerin standart sapma değeri: {standart_sapma(data2)}\n0 sınıflı verilerin standart sapma değeri: {standart_sapma(data1)}")
35
```

Standart sapma, ortalama, Precision, Recall, Confusion Matrix ve F1 Skorlarını da hesapladığım kodun gösterilen kısmı ortalama ve standart sapmayı oluşturmaktadır. Hem Numpy kütüphanesi ile hem de özel yazdığım fonksiyon ile bu iki değeri hesapladım.

```
(Numpy Versiyon) 1 sınıflı verilerin ortalama değeri: 2.1131376308125476  
0 sınıflı verilerin ortalama değeri: 0.059913329141126924
```

```
(Ben Versiyon) Oluşturulan 1 sınıflı verilerin ortalama değeri: 2.113137630812548  
0 sınıflı verilerin ortalama değeri: 0.05991332914112687
```

```
(Numpy Versiyon) 1 sınıflı verilerin standart sapma değeri: 0.9354017484924189  
0 sınıflı verilerin standart sapma değeri: 1.0054786744810833
```

```
(Ben Versiyon) Oluşturulan 1 sınıflı verilerin standart sapma değeri: 0.9377490589547546  
0 sınıflı verilerin standart sapma değeri: 1.0080018369789816
```

1 Sınıflı değerler iki hesaplama türünde de 2 ortalamasına yakın olurken, 0 sınıflı değerler ise 0 değerine yakın. İki sınıflandırma türünde de beklediğimiz gibi standart sapma değerleri 1'e çok yakın.

- e) (10 puan) Ölçüm değerlerini üretmede verilen py rutinini ya da kaynak belirterek başka bir rutini kullanabilirsiniz. Gauss dağılımına uygun sayı üretmenin algoritmik açıklamasını yapınız.

Gauss dağılımına uygun veri setleri üretmek için işlemleri hızlandıracığını düşünerek random kütüphanesindeki gauss aracını kullandım. Dokümantasyonda belirtildiği gibi

random.gauss()

`gauss()` is an inbuilt method of the `random` module. It is used to return a random floating point number with [gaussian distribution](#).

Syntax : `random.gauss(mu, sigma)`

Parameters :

mu : mean

sigma : standard deviation

Returns : *a random gaussian distribution floating number*

Ortalama değeri ve standart sapma eklendiğinde bu girdilere uygun gauss dağılımlı bir veri oluşturuyor. Bunu `n_of_samples` değişkeni kadar (bunu soruda olduğu gibi 200 olarak belirledim, artırılıp azaltılabilir.) döngüye aldım ve her döngüde bir kez 2 ortalamalı 1 standart sapmalı Sınıf 1 değişkeni, bir kez de 0 ortalamalı 1 standart sapmalı Sınıf 0 değişkeni oluşturdum. Bahsettiğim kısım aşağıda belirtilmiştir.

```

1  n_of_samples = 200
2
3  class_0, class_1 = [], []
4  avg_0_class, sd_0_class = 0, 1
5  avg_1_class, sd_1_class = 2, 1
6
7  for i in range(n_of_samples):
8      class_0.append(gauss(avg_0_class, sd_0_class))
9      class_1.append(gauss(avg_1_class, sd_1_class))
10

```

Sonrasında ise Matplotlib'den pyplot aracını kullanarak iki sınıftan oluşan 200 verinin histogramını çizdirdim:

```

plt.figure(figsize=(10,5))
plt.suptitle(f"Her bir sınıf için {n_of_samples} örnek değer oluşturulmuştur.")
plt.subplot(1,2,1)
plt.hist(class_0, bins = 50, alpha=0.7, color="Blue", label="0 Sinifi")
plt.xlabel("Degerler")
plt.ylabel("Sıklık")
plt.legend()

plt.subplot(1,2,2)
plt.hist(class_1, bins = 50, alpha=0.7, color="Red", label="1 Sinifi")
plt.xlabel("Degerler")
plt.ylabel("Sıklık")
plt.legend()
plt.savefig('./q_histograms.png', dpi=300)

```

Ardından da bunları precision gibi hesaplamalarda kullanacağımız için csv dosyasına kaydettim:

```

with open("./q_0_class.csv", 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(class_0)

with open("./q_1_class.csv", 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(class_1)

```

- f) (15 puan) (a)'da hesapladığınız genlik karar eşiği değerini kullanarak (b) ve (c) de ürettiğiniz ölçümleri sınıflandırınız. TP, FP, FN, TN değerlerini belirleyerek karıştırma (confusion) metrisini hesaplayınız.
- g) (15 puan) Karıştırma matrisini kullanarak her bir sınıf için Precision, Recall ve F1 metriklerini hesaplayınız.

TP,FP,FN,TN değerlerini; sonrasında da iki sınıf için de Precision, Recall ve F1 skorlarını hesaplamak için standart sapma ve ortalama değerlerini hesapladığım kodun devamında bu kodu oluşturdum.

```
#Confusion Matrix'i oluşturmak için gerekli değişkenlerin tanımı
tp,fp,fn,tn = 0,0,0,0
esik_degeri = 1

# 0 sınıflı veillerde Esik değerinden düşük ve eşit değerler True Negative olarak toplanıyor. Aksiler ise False Negative olarak.
tn = sum(1 for item in data1 if item <= esik_degeri)
fn = sum(1 for item in data1 if item > esik_degeri)

# 1 sınıflı verilerde 1 esik değerinin üstünde ve eşit değerler True Positive olarak değerlendirilirken aksi olanlar False Positive olarak değerlendiriliyor.
tp = sum(1 for item in data2 if item >= esik_degeri)
fp = sum(1 for item in data2 if item < esik_degeri)

print(f"\nTrue Positives: {tp}, True Negatives: {tn}, False Positives: {fp}, False Negatives: {fn}\n")

confusion_list = ["True Positive", "False Positive", "False Negative", "True Negative"],[tp,fp,fn,tn]

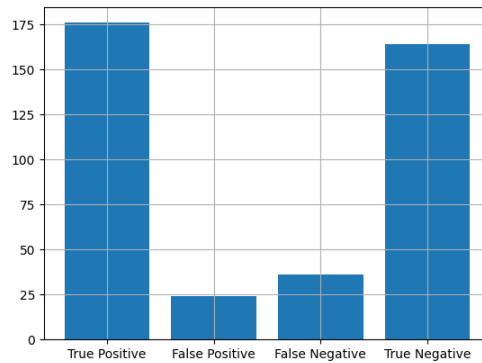
plt.figure()
plt.bar(confusion_list)
plt.grid(True)
plt.savefig("./bar_confusion_plot.png")

precision_1, recall_1 = tp/(tp+fp), tp/(tp+fn)
f1_score_1 = 2*(precision_1*recall_1)/(precision_1+recall_1)

precision_0, recall_0 = tn/(tn+fp), tn/(tn+fn)
f1_score_0 = 2*(precision_0*recall_0)/(precision_0+recall_0)

print(f"For Class 1:\nPrecision: {precision_1}\nRecall:{recall_1}\nF1 Score: {f1_score_1}\n\n")
print(f"For Class 0:\nPrecision: {precision_0}\nRecall:{recall_0}\nF1 Score: {f1_score_0}")
```

Kodun bu kısmının çalışmasından elde ettiğim TP, TN, FP, FN değerleri ve her sınıf için Precision Recall ve F1 skorunu konsol çıktısı olarak bu şekilde aldım, Aynı zamanda bu verileri görselleştirdim



```
True Positives: 176, True Negatives: 164, False Positives: 24, False Negatives: 36

For Class 1:
Precision: 0.88
Recall:0.8301886792452831
F1 Score: 0.8543689320388349

For Class 0:
Precision: 0.82
Recall:0.8723404255319149
F1 Score: 0.8453608247422681
PS C:\Users\PC\Desktop\myz\odev_1>
```

h) (10 puan) Tasarladığınız sınıflandırıcının performansı yeterli midir? Neden?

Tasarlanan sınıflandırıcı iki sınıf için de hesaplama zahmetine göre yüksek ölçüde precision, recall ve f1 skoru elde etmiştir. Fakat bu sınıflandırıcının yeterli olup olmadığı kullanılacağı ortama, yere, ve amaca göre çokça değişebileceğinden kesin bir şey söylemek doğru olmaz.