فاطمه ناشیان ۸۱۰۱۹۸۴۷۹ تمرین ۱ ماشین لرنینگ

۱

$$\Sigma = \begin{bmatrix} var(x) & Cov(x,y) \\ Cov(x,y) & Var(y) \end{bmatrix} \qquad Var(x) = \frac{1}{n}\sum (x_i - \mu_x)^۲$$

$$Cov(x,y) = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)$$

Class A :

$$\mu_{xA} = \frac{-۲-۱+۳+۳+۴+۵+۱}{۷} = \frac{۱۳}{۷}$$

$$\mu_{yA} = \frac{۰+۱+۱+۲+۲+۳+۵}{۷} = \frac{۱۴}{۷} \quad \rightarrow \quad \mu_A = \begin{bmatrix} \frac{۱۳}{۷} \\ ۲ \end{bmatrix}$$

$$Var(x) = \frac{1}{۷}\sum (x_i - \frac{۱۳}{۷})^۲ = ۵,۸۳۹۷$$

$$var(y) = \frac{1}{۷}\sum (y_i - ۲)^۲ = ۲,۲۸۵۷$$

$$Cov(x,y) = \frac{1}{۷}\sum (x_i - \frac{۱۳}{۷})(y_i - ۲) = ۳,۱۴۲۸$$

$$\rightarrow \Sigma_A = \begin{bmatrix} ۵,۸۳ & ۲,۲۸ \\ ۲,۲۸ & ۳,۱۴ \end{bmatrix}$$

Class B :

$$\mu_{xB} = \frac{(۰-۴-۳-۴)}{۵} = -\frac{۱۱}{۵}$$

$$\mu_{yB} = \frac{-۴-۲-۱+۱}{۵} = \frac{-۸}{۵} \quad \rightarrow \quad \mu_B = \begin{bmatrix} -\frac{۱۱}{۵} \\ -\frac{۸}{۵} \end{bmatrix}$$

$$Var(x) = \frac{1}{۵}\sum (x_i + \frac{۱۱}{۵})^۲ = ۶,۷۶$$

$$Var(y) = \frac{1}{۵}\sum (y_i + \frac{۸}{۵})^۲ = ۳,۱۲$$

$$Cov(x,y) = \frac{1}{۵}\sum (x_i + \frac{۱۱}{۵})(y_i + \frac{۸}{۵}) = -۲,۰۴$$

$$\rightarrow \Sigma_B = \begin{bmatrix} ۶۷۶ & -۲,۰۴ \\ -۲,۰۴ & ۳,۱۲ \end{bmatrix}$$

$$g_A(x) = -\frac{d}{۲}\ln ۲\pi - \frac{d}{۲}\ln|\Sigma_A| - \frac{1}{۲}(x-\mu_A)^T \Sigma_1^{-1}(x-\mu_A) + \ln(P(A))$$

$$g_A(x) = -\ln ۱۳,۱۲ + \ln\frac{۷}{۱۲} - \frac{1}{۲}\begin{bmatrix} x-\frac{۱۳}{۷} & y-۲ \end{bmatrix} \begin{bmatrix} ۰,۲۴ & -۰,۱۷۵ \\ -۰,۱۷۵ & ۰,۴۴۶ \end{bmatrix} \begin{bmatrix} x-\frac{۱۳}{۷} \\ y-۲ \end{bmatrix}$$

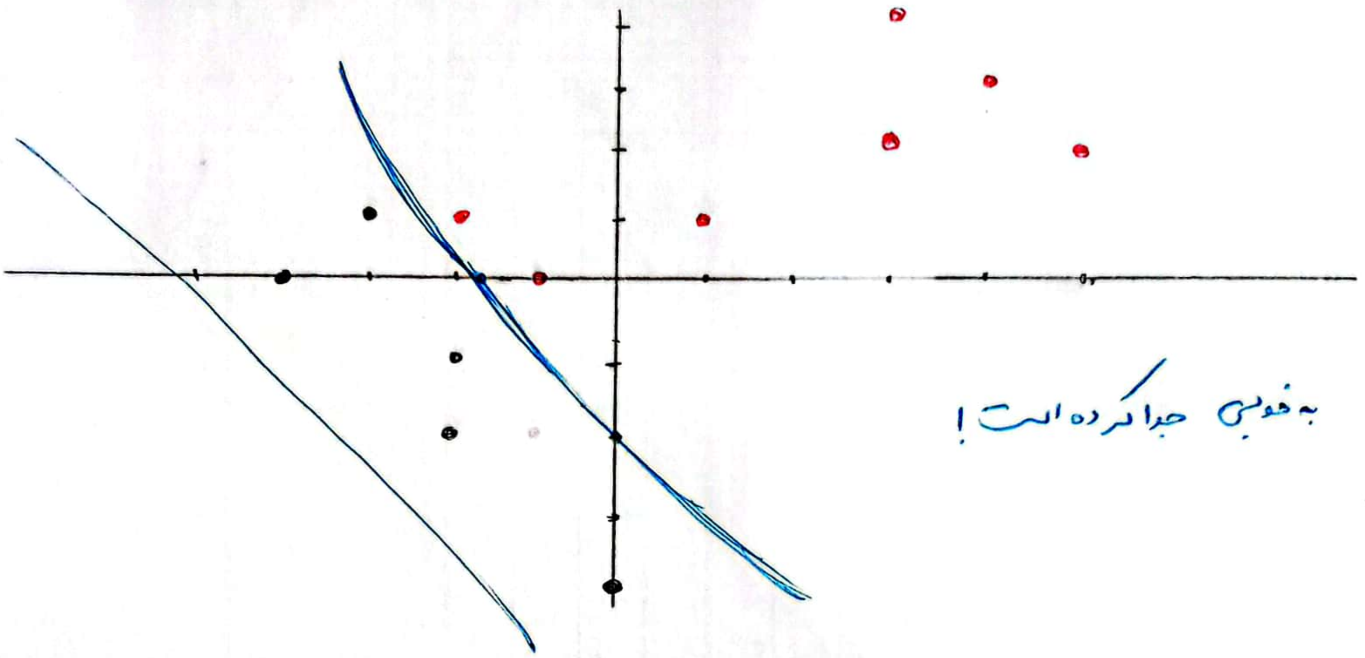
$$g_A(x) = -۳,۱۱ - \frac{1}{۲}(۰,۴۴۶y^۲ - ۰,۳۵yx - ۱,۱۳۴y + ۰,۲۴x^۲ - ۱,۳۲ - ۰,۲x)$$

$$g_B(x) = -\ln ۱,۳۳ + \ln\frac{۵}{۱۲} - \frac{1}{۲}\begin{bmatrix} x+\frac{۱۱}{۵} & y+\frac{۸}{۵} \end{bmatrix} \begin{bmatrix} ۲,۳۵ & ۶۵۳۵ \\ ۶۵۳۵ & ۱,۳۳۴ \end{bmatrix} \begin{bmatrix} x+\frac{۱۱}{۵} \\ y+\frac{۸}{۵} \end{bmatrix}$$

$$g_B(x) = -۱,۱۶ - \frac{1}{۲}\left[۲,۳۵x^۲ + ۳,۰۷xy + ۱۵,۲۵x + ۱,۳۲۴y^۲ + ۱۱y + ۲۵,۵۷\right]$$

$g_A(x) = g_B(x)$

$1.0555x^2 + 1.71yx + 7.7235x + 0.439y^2 + 4.0624y + 10.17 = 0$



(۱

(۲ $P(\omega_1) = P(\omega_2) = \frac{1}{2}$ $\quad P(x|\omega_i) = \frac{1}{\pi b} \frac{1}{1+(\frac{x-a_i}{b})^2}$ $\quad i = 1,2$

الف) $P(error) = P(\omega_1) - \int [P(\omega_1)P(x|\omega_1) - P(\omega_2)P(x|\omega_2)]\, dx$

برای پیدا کردن حداقل احتمال خطا از $P(error)$ نسبت به x مشتق می‌گیریم

ب)

$$\frac{dP(error)}{dx} = -[P(\omega_1)P(x|\omega_1) - P(\omega_2)P(x|\omega_2)] = 0$$

$$P(\omega_1)P(x|\omega_1) = P(\omega_2)P(x|\omega_2)$$

$$\frac{1}{2}\frac{1}{\pi b}\frac{1}{1+(\frac{x-a_1}{b})^2} = \frac{1}{2}\frac{1}{\pi b}\frac{1}{1+(\frac{x+a_2}{b})^2}$$

$$\Rightarrow \left(\frac{x-a_1}{b}\right)^2 = \left(\frac{x+a_2}{b}\right)^2$$

(1) $\frac{x-a_1}{b} = \frac{x-a_2}{b} \Rightarrow \boxed{a_1 = a_2}$ وقتی $a_1=a_2$ باشد یعنی احتمال هر دو کلاس در هر نقطه با هم برابر است، یعنی خطای ما به حالت Max خودش می‌رسد

(2) $\frac{x-a_1}{b} = \frac{a_2-x}{b} \Rightarrow \boxed{x = \frac{a_1+a_2}{2}}$

در این حالت خطای ما Min

حال جایگذاری می‌کنیم

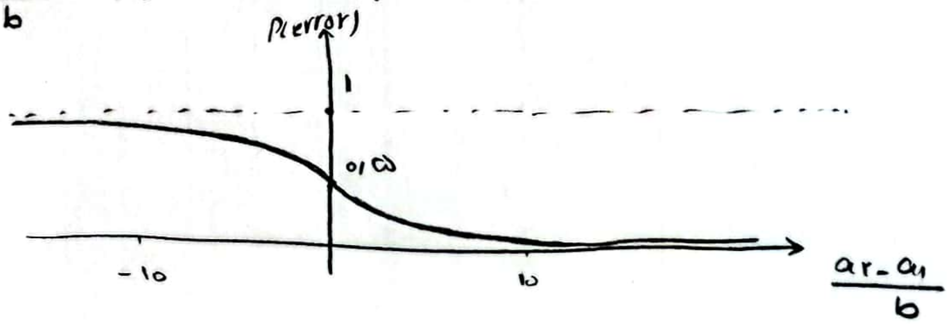
$$P(error) = P(w_1) - \int \left[P(w_1) \times \frac{1}{\pi b} \frac{1}{1+\left(\frac{x-a_1}{b}\right)^2} - P(w_2) \frac{1}{\pi b} \frac{1}{1+\left(\frac{x-a_2}{b}\right)^2} \right] dx$$

$$P(error) = P(w_1) - \left[P(w_1) \frac{1}{\pi} \tan^{-1}\left(\frac{x-a_1}{b}\right) \Big|_{-\infty}^{x=\frac{a_1+a_2}{2}} - P(w_2) \frac{1}{\pi} \tan^{-1}\left(\frac{x-a_2}{b}\right) \Big|_{-\infty}^{x=\frac{a_1+a_2}{b}} \right]$$

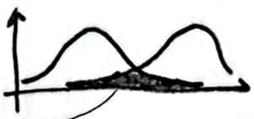
$$P(error) = \frac{1}{2} - \frac{1}{2\pi} \tan^{-1}\left(\frac{a_2-a_1}{2b}\right) + \frac{1}{2\pi} \tan^{-1}\left(\frac{a_1-a_2}{2b}\right)$$

$$P(error) = \frac{1}{2} - \frac{1}{\pi} \tan^{-1}\left(\frac{a_2-a_1}{2b}\right) \quad ✓✓$$

ب) فرض $u = \frac{a_2-a_1}{b} \rightarrow P(error) = \frac{1}{2} - \frac{1}{\pi} \tan^{-1}\left(\frac{u}{2}\right)$



ج) به صورت شهودی بیشترین مقدار خطا وقتی رخ می‌دهد که مثلاً همه w_1 ها را w_2 تشخیص دهیم (و یعنی دو عوامل همپوشانی داشته اند) یعنی مثلاً در حالتی که دو توزیع نرمال داشته باشیم و دو توزیع کاملاً به هم منطبق نباشند ناحیه



بیشینه خطا در اینجا رخ می‌دهد

حال اگر دو عوامل همپوشانی داشته باشند بیشینه خطا $Max(P(w_1), P(w_2))$ می‌شود که یعنی بیشینه خطا برابر 0.5 خواهد بود

از راه دیگری هم داریم

$$\frac{dP(error)}{dx} = 0 \quad \xrightarrow{\text{حالت}} ① \quad a_1 = a_2 = a$$

$$\rightarrow P(x|w_1) = P(x|w_2) = \frac{1}{\pi b} \frac{1}{1+\left(\frac{x-a}{b}\right)^2}$$

$$P(error) = P(w_1) - \int \underbrace{\left[P(w_1) \cdot P(x|w_1) - P(w_2) \cdot P(x|w_2)\right]}_{=0} dx$$

$$\boxed{P(error) = P(w_1) = 0.5}$$

۳) الف)

$$f(x_1, \dots, x_m \mid p_1, \dots, p_m) = \frac{n!}{\prod x_i!} \prod p_i^{x_i}$$

Likelihood $\hat{p} = \arg\max_{p} P(x \mid P)$

$$L(p) = \frac{n!}{\prod x_i!} \prod p_i^{x_i} \xrightarrow{\log} \log L(p) = \log n! - \sum_{i=1}^{n} \log x_i! + \sum_{i=1}^{n} x_i \log p_i$$

برای پیدا کردن Max باید مشتق بگیریم و مشتق را برابر صفر قرار دهیم

$$\nabla_p(L(p)) = 0 + 0 + \sum \frac{x_i}{p_i} = 0$$

می بینیم این روش پاسخی ندارد

از روش لاگرانژ استفاده می‌کنیم. می دانیم $\sum p_i = 1$ پس $1 - \sum_{i=1}^{m} p_i = 0$

$$L(p, \lambda) = \log n! - \sum_{i=1}^{n} \log x_i! + \sum_{i=1}^{n} x_i \log p_i + \lambda(1 - \sum p_i)$$

$$\nabla_p(L(p, \lambda)) = 0 + 0 + \sum \frac{x_i}{p_i} - \sum \lambda$$

$$\nabla_p(L(p, \lambda)) = 0 \rightsquigarrow \boxed{p_i = \frac{x_i}{\lambda}}$$

$$\sum p_i = 1 \rightsquigarrow \sum \frac{x_i}{\lambda} = 1 \rightsquigarrow \sum x_i = \lambda \rightsquigarrow \lambda = n$$

می دانیم که x_i ها تعداد ظاهر شدن هر وجه است. ما به صورت کلی n بار پرتاب کرده بودیم

پس $\sum x_i = n$ خواهد بود پس داریم

$$\boxed{p_i = \frac{x_i}{n}}$$

ب)

$$X \sim Unif([0, \theta]) \rightsquigarrow P(x \mid \theta) = \frac{1}{\theta} \qquad 0 \leq x_i \leq \theta$$

به ازای n نمونه: $P(x_1, x_2, \dots, x_n \mid \theta) = \frac{1}{\theta^n} = L(\theta)$

$$\log L\theta = -n \log \theta \rightsquigarrow \nabla_\theta L(\theta) = \frac{-n}{\theta} = 0 \rightsquigarrow$$ جواب مشخصی ندارد

برای Minimize کردن θ باید نهایتاً θ بزرگتر مساوی بزرگترین x_i باشد $\text{Max}\, x_i$

در بازه $0 \leq x_i \leq \theta$ قرار بگیرند یعنی $x_i \leq \theta$ و همه x_i ها باید چک شوند

و رابطه $\text{Max}(x_1, \dots, x_n) \leq \theta$ برقرار باشد. در نهایت داریم

$$\hat{\theta} = \text{Max}(x_1, \dots, x_n)$$

ج) فرض می کنیم که یک $M \leq \theta$ با احتمال 1 داریم که رابطه $M = Max\ x_i$ در آن برقرار باشد و M یک bias estimator باشد توزیع تخمین گر آن به صورت زیر است

$$P(M \leq m) = P(x_1 \leq m, x_2 \leq m, \dots, x_m \leq m) = \left(\frac{x}{\theta}\right)^n$$

اگر از رابطه بالا امید ریاضی بگیریم داریم

$$E_\theta(M_n) = \int_0^\theta P(M \geq m)\, dx = \int_0^\theta \left(1 - \left(\frac{x}{\theta}\right)^n\right) dx = \theta - \frac{\theta}{n+1} = \frac{n}{n+1}\theta$$

همانطور که مشاهده می شود $M \neq \theta$ است و بدین این تخمینگر biased است و کاملا به نمونه ها بستگی دارد

د)

توزیع پواسون $f(x) = \dfrac{\lambda^x e^{-\lambda}}{x!}$

نمونه داریم n $\quad P(x_1, \dots, x_n \mid \lambda) = \prod_{i=1}^{n} \dfrac{\lambda^{x_i} e^{-\lambda}}{x_i!} = L(\lambda)$

$$L(\lambda) = e^{-\lambda n} \frac{\lambda^{\sum x_i}}{\prod x_i!} \xrightarrow{\log} \log(L(\lambda)) = -\lambda n + \log(\lambda) \sum_{i=1}^{n} x_i - \log\left(\prod x_i!\right)$$

$$\nabla_\lambda \log(L(\lambda)) = \frac{d}{d\lambda} \log(L(\lambda)) = -n + \frac{\sum x_i}{\lambda} = 0$$

$$\rightarrow \hat{\lambda} = \frac{\sum x_i}{n}$$

مشاهده می شود تخمینی که به دست آمده میانگین نمونه هاست و می دانیم λ اصلی هم میانگین نمونه هاست

$$\lambda_{\text{اصلی}} = \frac{\sum x_i}{n}$$

پس $\hat{\lambda}$ به دست آمده unbiased است.

---

۴) الف)

$$P(\mu \mid x) = \frac{P(x \mid \mu)\, P(\mu)}{P(x)} \qquad x \sim N(\mu, \Sigma) \qquad \mu \sim N(m_0, \Sigma_0)$$

$$\hat{\mu}_{map} = \arg\max_{\mu} P(\mu \mid x)$$

$$P(D \mid \mu) = \prod_{k=1}^{n} P(x_k \mid \mu)$$

می خواهیم از log likelihood استفاده کنیم پس از آن log می گیریم

$$\log P(D \mid \mu) = \log\left(\prod_{k=1}^{n} P(x_k \mid \mu)\right) = \sum_{k=1}^{n} \log P(x_k \mid \mu)$$

می دانیم $\quad p(x_i \mid \mu) = \dfrac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$

جایگذاری می‌کنیم

$$\log P(D|\mu) = -\frac{n}{2}\log((2\pi)^d|\Sigma|) - \frac{1}{2}\sum_{k=1}^{n}(x_k-\mu)^T\Sigma^{-1}(x_k-\mu)$$

همچنین داریم $\log P(\mu) = -\frac{n}{2}\log((2\pi)^d|\Sigma_0|) - \frac{1}{2}(\mu-m_0)^T\Sigma_0^{-1}(\mu-m_0)$

حال که $\log P(\mu)$ و $\log P(D|\mu)$ را داریم به سراغ محاسبه $\hat{\mu}_{map}$ میرویم

$$\hat{\mu}_{map} = \arg\max_{\mu} P(\mu|x) = \arg\max_{\mu} P(D|\mu)P(\mu)$$

لگاریتم میگیریم، جایگذاری میکنیم

$$\hat{\mu}_{map} = \arg\max_{\mu}\left\{-\frac{1}{2}\sum_{k=1}^{n}(x_k-\mu)^T\Sigma^{-1}(x_k-\mu) - \frac{1}{2}(\mu-m_0)^T\Sigma_0^{-1}(\mu-m_0)\right\}$$

ب) ابتدا تبدیل خطی $x' = Ax$ را بر روی میانگین و کوواریانس قبل اعمال میکنیم

$$\mu' = E[x'] = E[Ax] = A\mu$$

$$\Sigma' = E[(x'-\mu')(x'-\mu')^T] = [(Ax-A\mu)(Ax-A\mu)^T] = A\Sigma A^T$$

$$P(\mu') = \frac{1}{(2\pi)^{d/2}|\Sigma_0'|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\mu'-m_0)^T\Sigma_0^{-1}(\mu'-m_0)\right]$$

$$P(\mu') = \frac{1}{(2\pi)^{d/2}|\Sigma_0|^{1/2}}\exp\left[-\frac{1}{2}(A\mu-Am_0)^T(A\Sigma_0A^T)^{-1}(A\mu-Am_0)\right]$$

رابطه بالا را ساده میکنیم و A و A^{-1} باهم ساده میشوند

$$P(\mu') = \underbrace{\frac{1}{(2\pi)^{d/2}|\Sigma_0|^{1/2}}}_{\text{ثابت}}\exp\left[-\frac{1}{2}(\mu-m_0)^T\Sigma_0^{-1}(\mu-m_0)\right]$$

$$\log P(\mu') = -\frac{n}{2}\log((2\pi)^d|\Sigma_0|) - \frac{1}{2}(\mu-m_0)^T\Sigma_0^{-1}(\mu-m_0)$$

حال $P(D'|\mu')$ را بررسی میکنیم. برای سادگی ابتدا $\log$ میگیریم تا محاسبات کم شود

$$\log(P(D'|\mu')) = -\frac{n}{2}\log((2\pi)^d|A\Sigma A^T|) - \sum_{k=1}^{n}\frac{1}{2}(Ax_k-A\mu)^T(A\Sigma A^T)^{-1}(Ax_k-A\mu)$$

$$\log(P(D'|\mu')) = -\frac{n}{2}\log((2\pi)^d|A\Sigma A^T|) - \frac{1}{2}\sum_{k=1}^{n}(x_k-\mu)^T\Sigma^{-1}(x_k-\mu)$$

دوباره A و A^{-1} ها ساده میشوند و همه را مانند بخش الف شد، پس تخمین بخش الف تخمین مناسبی برای μ است.

$$\hat{\mu}' = \arg\max \log\{P(D'|\mu')P(\mu')\} = \arg\max_{\mu}\left\{-\frac{1}{2}\sum_{k=1}^{n}(x_k-\mu)^T\Sigma^{-1}(x_k-\mu) - \frac{1}{2}(\mu-m_0)^T\Sigma_0^{-1}(\mu-m_0)\right\}$$

۴) الف) Naive Bayes : به گروهی از classifier ها گفته میشود که فرض می کنند بین متغیرهای تصادفی استقلال وجود دارد. این classifier دقت خوبی دارد

$$P(C_k \mid x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n \mid C_k)\, P(C_k)}{P(x)}$$

$$P(C_k \mid x_1, \dots, x_n) = P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)$$

Optimal Bayes : متغیرهای تصادفی از هم مستقل نیستند و برای پیدا کردن $\hat{y}$ باید بیشترین احتمال را پیدا کنیم

$$\hat{y} = \arg\max_{y \in Y} P(y \mid x)$$

5) ابتدا کد و نتایج ان را مشاهده میکنیم

In [1]:
```python
import cv2
import glob
from PIL import Image
import numpy as np
```

In [2]:
```python
Tp = 0
Fn = 0
Fp = 0
Tn = 0
for name in glob.glob('Images\*[0-9].*'):
    img = cv2.imread(name, cv2.IMREAD_UNCHANGED)
    height , width ,x =img.shape
    rgb=[0,0,0]
    for i in range(height):
        for j in range(width):
            rgb += img[i,j]
    if (name[7] == 'c' and rgb[2]<rgb[0]):
        Tp += 1
    if (name[7] == 'c' and rgb[2]>rgb[0]) :
        Fn += 1
    if (name[7] == 'm' and rgb[2]<rgb[0]):
        Fp += 1
    if (name[7] == 'm' and rgb[2]>rgb[0]):
        Tn += 1
```

In [3]:
```python
print('      c       |       m')
print('  --------------------------------------------------')
print('c | TP= ',Tp,'    |   FN=  ',Fn,'    |   recall= ',Tp/(Tp+Fn))
print('  ---------------------------------------------------')
print('m | FP=  ',Fp,'    |   TN=   ',Tn,'    |   specifity= ',Tn/(Tn+Fp))
print('--------------------------------------------------')
print('   percision= ',Tp/(Tp+Fp),'   | negetive predictive value =',Tn/(Tn+Fn))
print('   Accuracy=',(Tp+Tn)/(Tp+Tn+Fn+Fp))
```

```
      c       |       m
  --------------------------------------------------
c | TP=  46     |   FN=   18     |   recall=  0.71875
  ---------------------------------------------------
m | FP=   1     |   TN=    57    |   specifity=  0.9827586206896551
--------------------------------------------------
   percision=  0.9787234042553191    | negetive predictive value = 0.76
   Accuracy= 0.8442622950819673
```

نتایج به خوبی نشان میدهد طبقه بندی که طراحی کردیم توانسته منچستر را جدا کند ولی دقت ان برای چلسی کمتر بوده است. در کل دقتی بیشتر از 97 درصد داریم که خوب است. ولی recall به نسبت ضعیف بوده است.

6)

ج)

part 3 : naive bayes with library

```
[ ]
from sklearn import datasets
import pandas as pd
df_iris = pd.read_csv('Iris.csv')
```

```
df_iris.head()
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
from sklearn.model_selection import train_test_split

X = df_iris[['Sepal_Length', 'Sepal_Width',
             'Petal_Length', 'Petal_Width']]

y = df_iris['Class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=8)
```

```
from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()

# fitting the classifier
clf.fit(X_train, y_train);
```

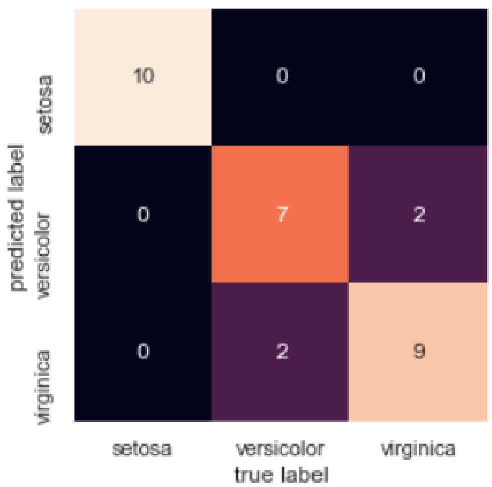
```
y_pred = clf.predict(X_test)

from sklearn.metrics import accuracy_score

print("The accuracy of the model is: %.1f%%" % (accuracy_score(y_test, y_pred)*100))
```

```
The accuracy of the model is: 86.7%
```

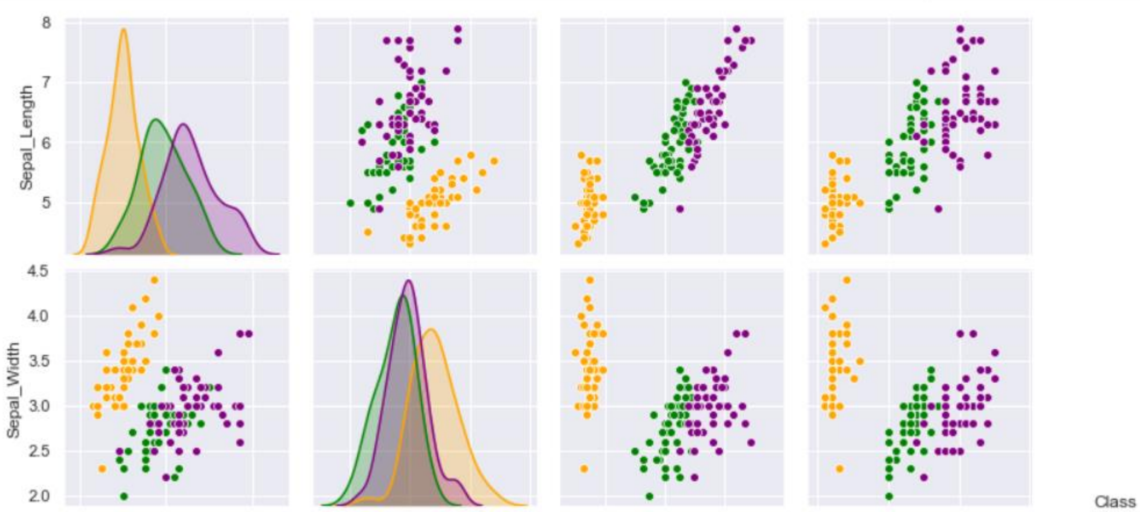
```
from sklearn.model_selection import cross_val_score
import numpy as np
acc = cross_val_score(clf, X, y, cv=5, scoring='accuracy')

print(acc)

print("\nwe are 95 percent confident that our accuracy is within: %.2f%% +- %.2f%%"
      %(np.mean(acc)*100, 2 * np.std(acc) / np.sqrt(5) *100))
```

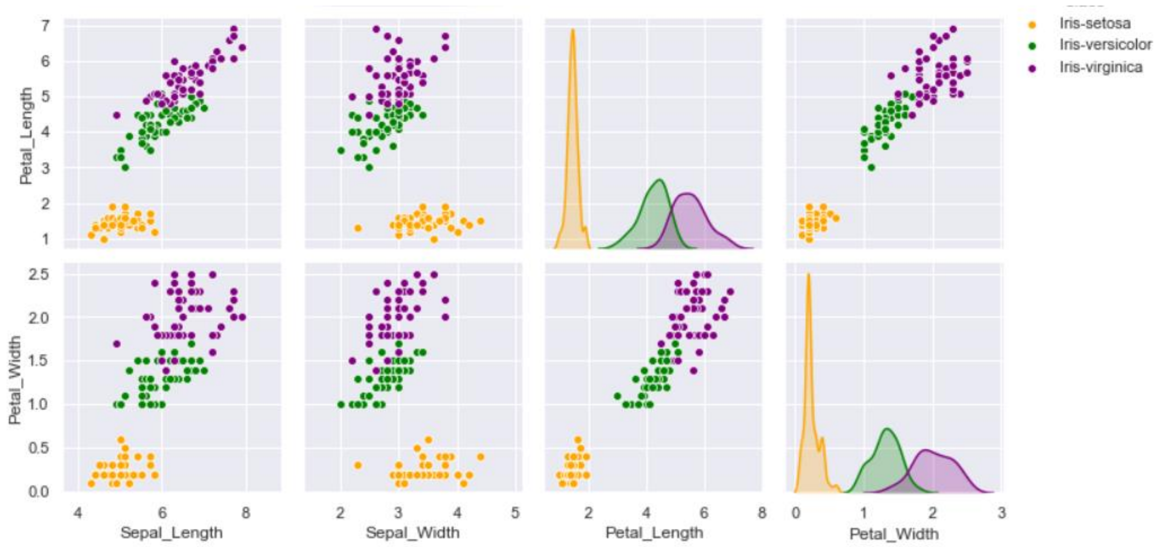
```
[0.93333333 0.96666667 0.93333333 0.93333333 1.        ]

we are 95 percent confident that our accuracy is within: 95.33% +- 2.39%
```

```python
from sklearn.metrics import confusion_matrix
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
confusion_mtx = confusion_matrix(y_test, y_pred)
target=['setosa' , 'versicolor' , 'virginica']
sns.heatmap(confusion_mtx.T, square=True, annot=True, fmt='d',cbar=False ,
            xticklabels=target, yticklabels=target)
plt.xlabel('true label')
plt.ylabel('predicted label');

```




```python
palette = {'Iris-setosa': 'orange', 'Iris-versicolor': 'green', 'Iris-virginica': 'purple'}

g = sns.pairplot(df_iris, vars = df_iris.columns[0:4], hue="Class", palette=palette)
```






```python
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred, target_names=target))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       0.78      0.78      0.78         9
   virginica       0.82      0.82      0.82        11

    accuracy                           0.87        30
   macro avg       0.87      0.87      0.87        30
weighted avg       0.87      0.87      0.87        30
```