SPR ASSIGNMENT 1

Jayanth S (201081003)

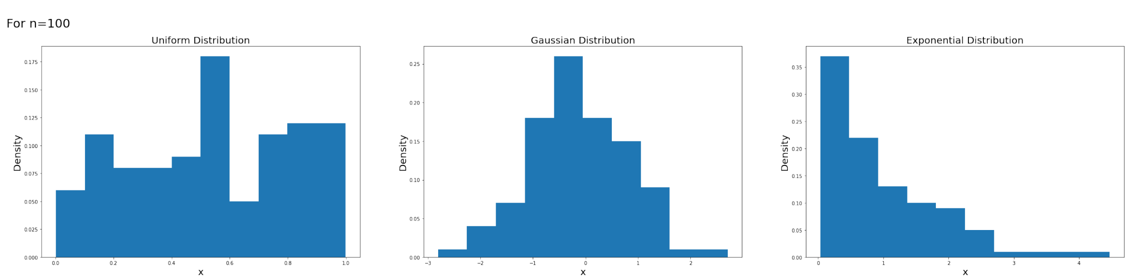
January 19, 2021

[16]:

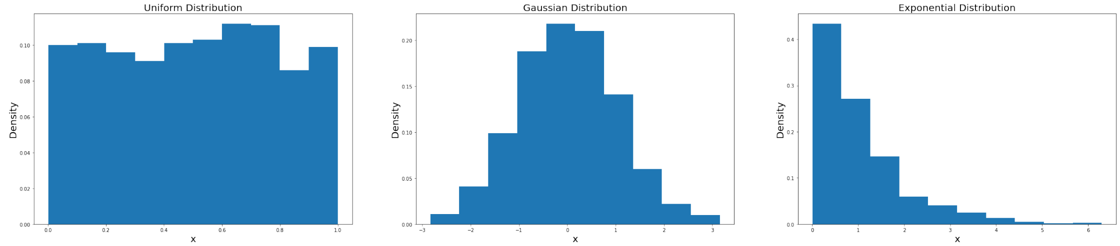
```
#CODE FOR QUESTION 1
import numpy as np
import matplotlib.pyplot as plt



n=100    #specifies the no. of sample points
w=np.ones(n)/n                        #helps in plotting the histogram with␣
 ↪fraction of points when w is given as weights to histogram plot

#Uniform Distribution
unf=np.random.uniform(0,1,n)         #generates n sample points from Uniform␣
 ↪distribution
plt.figure(figsize=(40,8))           #sets the figure width to 30 and height to 5
plt.subplot(1,3,1)                   #specifies the no. rows and columns of the␣
 ↪plot and also the index of the present sub-plot
plt.hist(unf,weights=w)              #plots the histogram of sampled points
plt.title("Uniform Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

#normal distribution
nrml=np.random.normal(0,1,n)         #generates n sample points from Normal␣
 ↪distribution
plt.subplot(1,3,2)
plt.hist(nrml,weights=w)
plt.title("Gaussian Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

#Exponential distribution
expl=np.random.exponential(1,n)    #generates n sample points from Exponential␣
 ↪distribution
plt.subplot(1,3,3)
plt.hist(expl,weights=w)
plt.title("Exponential Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)
```

```python
print('\n')
plt.suptitle("For n=100",fontsize=25,x=0.1,ha='left')   #Title for all 3 subplots
plt.show()


n=1000
w=np.ones(n)/n

#Uniform Distribution
unf=np.random.uniform(0,1,n)          #generates n sample points from Uniform␣
 ↪distribution
plt.figure(figsize=(40,8))
plt.title("For n=100")
plt.subplot(1,3,1)
plt.hist(unf,weights=w)
plt.title("Uniform Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

#normal distribution
nrml=np.random.normal(0,1,n)        #generates n sample points from Normal␣
 ↪distributionplt.subplot(1,3,2)
plt.subplot(1,3,2)
plt.hist(nrml,weights=w)
plt.title("Gaussian Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

#Exponential distribution
expl=np.random.exponential(1,n)    #generates n sample points from Exponential␣
 ↪distribution
plt.subplot(1,3,3)
plt.hist(expl,weights=w)
plt.title("Exponential Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

print('\n')
plt.suptitle("For n=1000",fontsize=25,x=0.1,ha='left')
plt.show()


n=10000
w=np.ones(n)/n


#Uniform Distribution
```

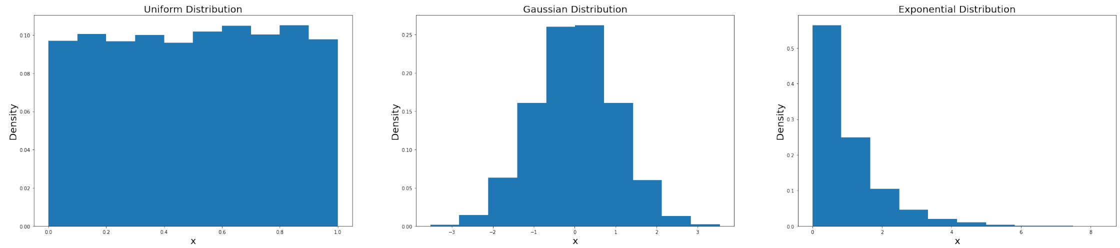
```
unf=np.random.uniform(0,1,n)        #generates n sample points from Uniform␣
 ↪distribution
x=plt.figure(figsize=(40,8))
x.suptitle("For n=10000")
plt.subplot(1,3,1)
plt.hist(unf,weights=w)
plt.title("Uniform Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

#normal distribution
nrml=np.random.normal(0,1,n)        #generates n sample points from Normal␣
 ↪distributionplt.subplot(1,3,2)
plt.subplot(1,3,2)
plt.hist(nrml,weights=w)
plt.title("Gaussian Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)


#Exponential distribution
expl=np.random.exponential(1,n)    #generates n sample points from Exponential␣
 ↪distribution
plt.subplot(1,3,3)
plt.hist(expl,weights=w)
plt.title("Exponential Distribution",fontsize=20)
plt.xlabel("x",fontsize=20)
plt.ylabel("Density",fontsize=20)

print('\n')
plt.suptitle("For n=10000",fontsize=25,x=0.1,ha='left')
plt.show()
```



For n=1000



For n=10000



[11]:

```python
#CODE FOR QUESTION 2
import numpy as np

n=np.array([2,3,5])                                      #points that are required to be sampled
p=np.zeros((1,len(n)))                                   #To store the probability for different no. of sample points
for i in range(len(n)):
      for s in range(100000):                                #no. of times we will sample 'n' points. (n can be 2 or 3 or 5)
        u_smp=np.random.uniform(0,360,n[i])                  #take n[i] no. of samples  from 0 to 360 degrees following 'uniform distribution'
        u_neg=np.min(u_smp,initial=360,where=u_smp>180)   #obtain the angle that is greater than 180 and is also close to 180

        u_smp=((360-u_neg)+u_smp)%360                       #rotate the points in anti-clockwise direction such that the the point at 'u_neg' angle will now be at the '0' angle
        u_smp=np.where(u_smp>180,u_smp-360,u_smp)          #Now place all the angles between (180,-180)
```

```
        u_smp.sort()                                        #sort the array in␣
 ↪ascending order
        u_smpdif=np.ediff1d(u_smp)                          #find the diffence␣
 ↪between consecutive angles in the array

        if np.sum(u_smpdif)<=180:                           #the sum of all these␣
 ↪consecutive differences in angles should be less than 180 for all the points␣
 ↪to be inside a semi-circle.
            p[0,i]+=1
p=p/100000                                                  #gives the estimated␣
 ↪probability values


print(f"Theoritical Probability that both the sampled points lie with some␣
 ↪semi-circle is 1.")
print(f"Estimated Probability that both the sampled points lie with some␣
 ↪semi-circle is {p[0,0]}.")
print()
print(f"Theoritical Probability that all 3 sampled points lie with some␣
 ↪semi-circle is 0.75.")
print(f"Estimated Probability that all 3 sampled points lie with some␣
 ↪semi-circle is {p[0,1]}.")
print()
print(f"Theoritical Probability that all 5 sampled points lie with some␣
 ↪semi-circle is 0.3125.")
print(f"Estimated Probability that all 5 sampled points lie with some␣
 ↪semi-circle is {p[0,2]}.")
```

```
Theoritical Probability that both the sampled points lie with some semi-circle
is 1.
Estimated Probability that both the sampled points lie with some semi-circle is
1.0.

Theoritical Probability that all 3 sampled points lie with some semi-circle is
0.75.
Estimated Probability that all 3 sampled points lie with some semi-circle is
0.75273.

Theoritical Probability that all 5 sampled points lie with some semi-circle is
0.3125.
Estimates Probability that all 5 sampled points lie with some semi-circle is
0.31062.
```

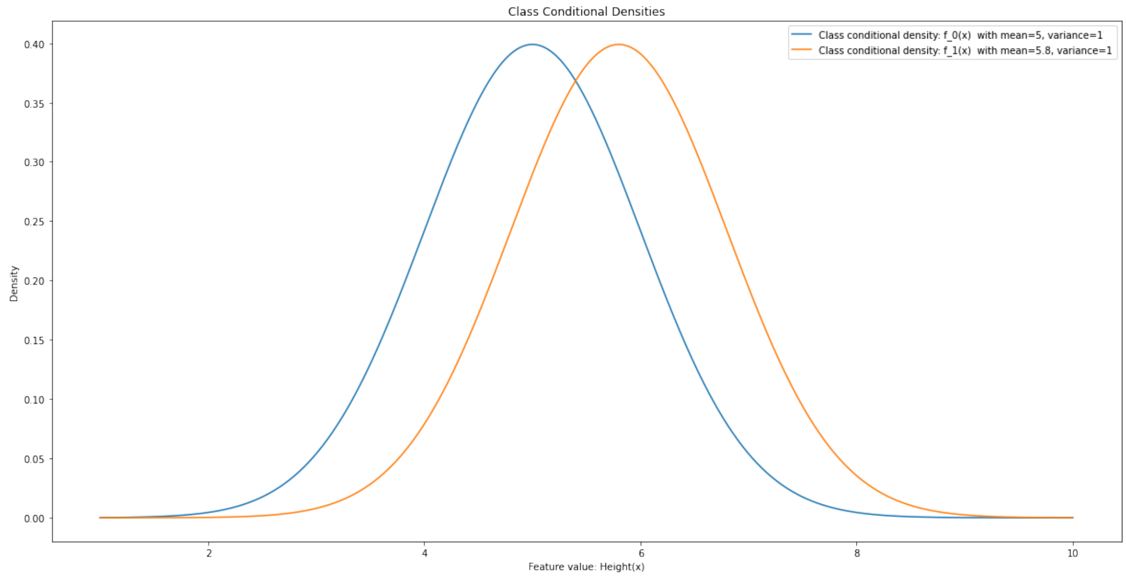
[17]:

```
#QUESTION 3
import numpy as np
import matplotlib.pyplot as plt
```

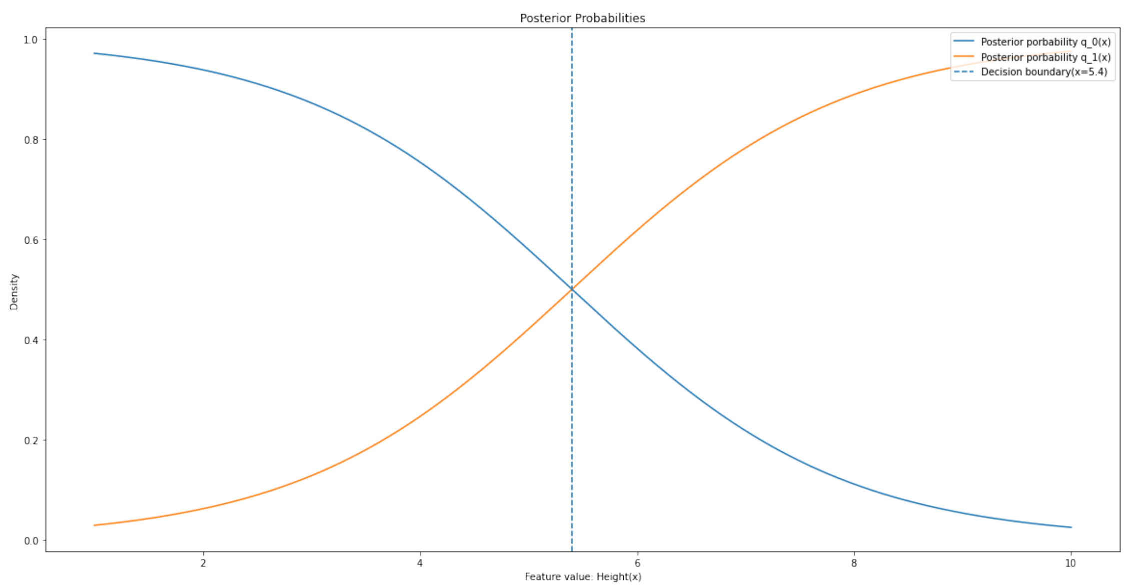
```python
from scipy import stats


#class conditional density f_0(x) (Male class)   : Gaussian distribution with␣
↪mean = 5 and variance = 1 and x represents the feature value height
#class conditional density f_1(x) (Female class) : Gaussian distribution with␣
↪mean = 5.8 and variance = 1 and x represents the feature value height
mu_ml=5.8                 #mean value for f_1(x)
mu_fml=5                  #mean value for f_1(x)
variance=1                #same variance for both class conditional densities
h=np.linspace(1,10,1000)             #gives 1000 equally spaced points between 1␣
↪and 10
f1_pdf=stats.norm.pdf(h,mu_ml,1)     #generates the probability density function␣
↪of f_1(x) with mean=5.8 and variance=1
f0_pdf=stats.norm.pdf(h,mu_fml,1)    #generates the probability density function␣
↪of f_0(x) with mean=5.8 and variance=1

plt.figure(figsize=(20,10))
plt.plot(h,f0_pdf,label=f'Class conditional density: f_0(x)  with mean={mu_fml},␣
↪variance={variance}') #plot the class conditonal density f_0(x)
plt.plot(h,f1_pdf,label=f'Class conditional density: f_1(x)  with mean={mu_ml},␣
↪variance={variance}')  #plot the class conditonal density f_0(x)
plt.legend(loc='upper right')
plt.title("Class Conditional Densities")
plt.xlabel("Feature value: Height(x)")
plt.ylabel("Density")
plt.show()


#CASE a

p0=0.5                          #specifies the prior probability of FEMALE class
p1=0.5                          #specifies the prior probability of MALE class

fx_pdf=p0*f0_pdf+p1*f1_pdf      #fx_pdf denotes the PDF of the feature value␣
↪height

q0_pdf=(p0*f0_pdf)/fx_pdf       #posterior probability of FEMALE class
q1_pdf=(p1*f1_pdf)/fx_pdf       #posterior probability of MALE class

dec_bound=(mu_ml+mu_fml)/2      #Decision boundary for Baye's classifier
print(f"The decision boundary for the Baye's classifier with p0={p0} and p1={p1}␣
↪is {dec_bound}.")
```

```
misclf_error=p0*stats.norm.sf(dec_bound,loc=mu_fml,scale=variance)+p1*stats.norm.
↪cdf(dec_bound,loc=mu_ml,scale=variance) #calcuates the misclassficion error␣
↪for the Baye's classifier
print(f"Misclassification error is {misclf_error:0.4}")

plt.figure(figsize=(20,10))
plt.plot(h,q0_pdf,label='Posterior porbability q_0(x)')
plt.plot(h,q1_pdf,label='Posterior porbability q_1(x)')                        ␣
↪                        #To plot the posterior probabilities of MALE and FEMALE␣
↪classes
plt.axvline(dec_bound,ls='--',label=f'Decision boundary(x={dec_bound:0.4})')    ␣
↪                         #To plot the decision boundary of Baye's classifier
plt.legend(loc='upper right')
plt.title("Posterior Probabilities")
plt.xlabel("Feature value: Height(x)")
plt.ylabel("Density")
plt.show()

#CASE b
p1=0.1               #specifies the prior probability of FEMALE class
p0=0.9               #specifies the prior probability of MALE class
# print(dec_bound)

fx_pdf=p0*f0_pdf+p1*f1_pdf          #fx_pdf denotes the PDF of the feature value␣
↪height

q0_pdf=(p0*f0_pdf)/fx_pdf           #posterior probability of FEMALE class
q1_pdf=(p1*f1_pdf)/fx_pdf           #posterior probability of MALE class

dec_bound=(2*np.log(p0/p1)+mu_ml**2-mu_fml**2)/(2*(mu_ml-mu_fml))          ␣
↪#Decision boundary for Baye's classifier
print(f"The decision boundary for the Baye's classifier with p0={p0} and p1={p1}␣
↪is {dec_bound:0.4}.")

misclf_error=p0*stats.norm.sf(dec_bound,loc=mu_fml,scale=variance)+p1*stats.norm.
↪cdf(dec_bound,loc=mu_ml,scale=variance)  #calcuates the misclassficion error␣
↪for the Baye's classifier
print(f"Misclassification error is {misclf_error:0.4}")

plt.figure(figsize=(20,10))
plt.plot(h,q0_pdf,label='Posterior porbability q_0(x)')
plt.plot(h,q1_pdf,label='Posterior porbability q_1(x)')                        ␣
↪                        #To plot the posterior probabilities of MALE and FEMALE␣
↪classes
plt.axvline(dec_bound,ls='--',label=f'Decision boundary(x={dec_bound:0.4})')    ␣
↪                         #To plot the decision boundary of Baye's classifier
```

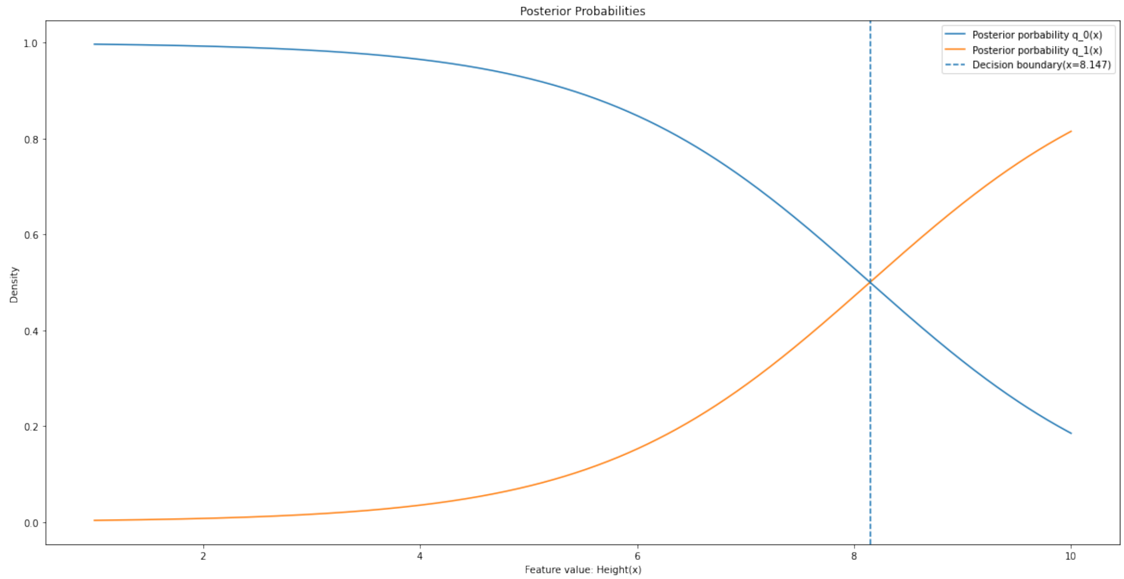
```
plt.legend(loc='upper right')
plt.title("Posterior Probabilities")
plt.xlabel("Feature value: Height(x)")
plt.ylabel("Density")
plt.show()
```



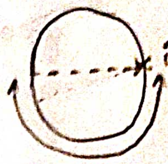

```
The decision boundary for the Baye's classifier with p0=0.5 and p1=0.5 is 5.4.
Misclassification error is 0.3446
```



```
The decision boundary for the Baye's classifier with p0=0.9 and p1=0.1 is 8.147.
Misclassification error is 0.0998
```



2) Let 'n' be the no. of points we sample uniformly on the circumference of the circle.



- Picking a point from 'n' points can be done in ${}^nC_1$ ways.
- Let s_i be the sample point that is picked.
- Then, we get two semi-circles with starting point as s_i.
- The probability that remaining points will lie within one of these semi-circle is,

$$\underbrace{(1/2) \times (1/2) \times \cdots \times 1/2}_{(n-1)\ \text{times}} = (1/2)^{n-1}$$

(∵ there are $(n-1)$ sample points left)

$(1/2)$ because, each sample point is taken from uniform distribution and it should either belong to one of the two semi-circles formed on the circle, and also picking a point is a independant event.

⇒ The probability that all 'n' points lie with in a semi-circle is $p_n = {}^nC_1 \cdot (1/2)^{n-1} = \frac{n}{2^{n-1}}$

$$\boxed{p_n = \frac{n}{2^{n-1}}} \quad ; \quad n \geq 2$$

- For $n=2$: $p_2 = 1$
- For $n=3$: $p_3 = 0.75$
- For $n=5$: $p_5 = 0.3125$

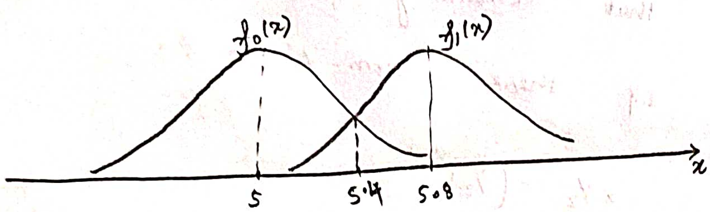
3) Two classes Male and Female.

Let $y=0$ represent Female class,

Let $y=1$ represent Male class.

Let 'x' represent feature value height,

$\rightarrow$ $\left.\begin{aligned} f_0(x) &= p_{x/y=0}(x) \sim N(5.8, 1) \\ \text{and} \quad f_1(x) &= p_{x/y=1}(x) \sim N(5, 1) \end{aligned}\right\}$ Class - conditional densities.



a) Given prior probabilities are :

$p_0 = P(y=0) = 0.5$, $p_1 = P(y=1) = 0.5$

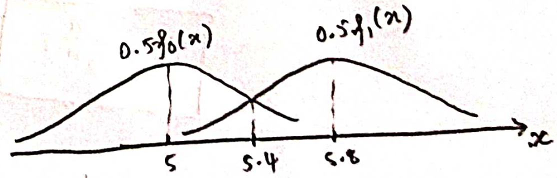
$\Rightarrow$ $q_0(x) = \cancel{P_{Y/X=x}(y)}\ P(y=0/x) = p_0 f_0(x)/f(x)$

$q_1(x) = P(y=1/x) = p_1 f_1(x)/f(x)$

where $f(x) = p_0 f_0(x) + p_1 f_1(x)$

$\Rightarrow$ $q_0(x) = \dfrac{f_0(x)}{f_0(x) + f_1(x)}$ $(\because p_0 = 0.5 = p_1)$

$q_1(x) = \dfrac{f_1(x)}{f_0(x) + f_1(x)}$



Bayes Classifier (h_B) :

$$h_B(x) = \begin{cases} 1 & ; \text{ if } q_1(x) \geq q_0(x) \\ 0 & ; \text{ if } q_0(x) > q_1(x) \end{cases}$$

- For the given problem,

$$h_B(x) = \begin{cases} 1 & ; \text{ if } x \geq \frac{5+5.8}{2} = 5.4 \\ 0 & ; \text{ if } x < 5.4 \end{cases}$$

- Missclassification error : $err(h_B) = P[\,h_B(X) \neq Y\,]$

$$err(h_B) = \int_{-\infty}^{5.4} P_1 f_1(x)\,dx + \int_{5.4}^{\infty} P_0 f_0(x)\,dx$$

$$= 0.5 \int_{-\infty}^{5.4} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x-5.8)^2}{2}\right) dx + 0.5 \int_{5.4}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x-5.0)^2}{2}\right) dx$$

put $x - 5.8 = t$

$\Rightarrow dx = dt$

when $x = -\infty$, $t = -\infty$

$x = 5.4$, $t = -0.4$

put $x - 5 = t$

$dx = dt$

when $x = 5.4$ $t = 0.4$

$x = \infty$, $t = \infty$

$$= 0.5 \underbrace{\int_{-\infty}^{-0.4} \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right) dt}_{= Q(0.4)} + 0.5 \underbrace{\int_{0.4}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right) dt}_{= Q(0.4)}$$

(OR)

$$= 0.5 \left(1 - \underbrace{\int_{-0.4}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right) dt}_{Q(-0.4)}\right) + 0.5\ Q(0.4)$$

$$= 0.5\,(1 - 0.655) + 0.5\,(0.345)$$

$$\boxed{err(h_B) = 0.345}$$

b) for $p_0 = 0.9$, $p_1 = 0.1$

$$\Rightarrow \quad h_B(x) = 1 \quad \text{if} \quad p_1 f_1(x) \geq p_0 f_0(x)$$

$$\Rightarrow \quad 0.1 \, \cancel{\frac{1}{\sqrt{2\pi}}} \exp\left(-\frac{(x-5.8)^2}{2}\right) \geq 0.9 \, \cancel{\frac{1}{\sqrt{2\pi}}} \exp\left(-\frac{(x-5)^2}{2}\right)$$

$$\ln(0.1) - \frac{(x-5.8)^2}{2} \geq \ln(0.9) - \frac{(x-5)^2}{2}$$

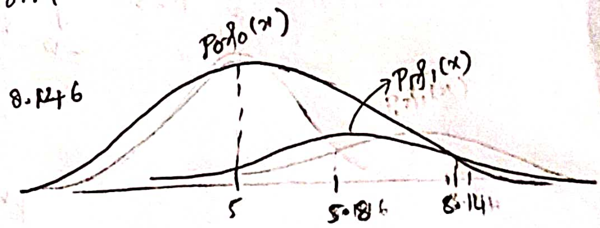
$$\frac{1}{2}\left[(x-5)^2 - (x-5.8)^2\right] \geq \ln(9)$$

$$\frac{1}{2}\left[\cancel{x^2} + 25 - 10x - \cancel{x^2} - 5.8^2 + 2\times 5.8x\right] \geq \ln(9)$$

$$0.8x - 4.32 \geq \ln(9)$$

$$\boxed{x \geq 8.146}$$

$$\Rightarrow \quad h_B(x) = \begin{cases} 1 \; ; & \text{if } x \geq 8.146 \\ 0 \; ; & \text{if } x < 8.146 \end{cases}$$



Missclassification error $=$ err(h_B) $= P[h_B \neq y]$

$$\text{err}(h_B) = 0.9 \int_{8.14}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-5)^2}{2}\right) dx + 0.1 \int_{-\infty}^{8.14} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-5.8)^2}{2}\right) dx$$

$\downarrow$

put $x - 5 = t$

$\Rightarrow dx = dt$

for $x = 8.14$, $t = 3.14$

$x = \infty$, $t = \infty$

$\downarrow$

put $x - 5.8 = t$

$dx = dt$

for $x = 8.14$, $t = 2.34$

$x = -\infty$, $t = -\infty$

$$\text{err}(h_B) = 0.9 \int_{3.14}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right) dt + 0.1 \int_{-\infty}^{2.34} \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right) dt$$

$$= 0.9 \, Q(3.14) + 0.1\left(1 - Q(2.34)\right)$$

$$err(h_B) = 0.9 \times 0.000845 + 0.1(1 - 0.009642)$$

$$* \cancel{0.9995855}$$

$$err(h_B) = 0.00076 + 0.0990$$

$$err(h_B) = \underline{0.09976}$$