Ajuste de distribuciones de probabilidad a datos de train y la distribución predictiva (test) para distribución gaussiana unidimensional.

1. Defina n>10 puntos aleatorios extraídos de una distribución gaussiana (μ0, σ0).
   μ0=5, σ0=1.5
2. Grafique 3 figuras donde muestre los puntos, los valores de likelihood para 3 gaussianas.
   μ1=3, σ1=1
   μ2=6, σ2=1.6
   μ3=5.1, σ3=1.4
3. Genera el mapa térmico de las probabilidades $\mu \in [2.5, 6.5]$ y $\sigma \in [0, 2]$ y colocar una cruz en el máximo ML.
   Dividir el intervalo μ en 10 intervalos
   Dividir el intervalo σ en 10 intervalos
4. Calcule con ML, μ, σ y vea que tanto coincide con su mapa termico.

Para obtener los puntos aleatorios se utiliza la función random.gauss de la librería random de python, a la cual se le ingresa como valores de entrada μ0=5, σ0=1.5 y nos da como dato de salida puntos correspondientes a la gaussiana de manera aleatoria.

La ecuación utilizadas para calcular el likelihood es:

$$Pr(x_{1\ldots I}|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{I/2}} \exp\left[-0.5\sum_{i=1}^{I}\frac{(x_i-\mu)^2}{\sigma^2}\right]$$

Mientras que la ecuación para calcular el μ, σ a partir de los datos obtenidos es:

$$\hat{\mu} = \frac{\sum_{i=1}^{I} x_i}{I}$$

$$\hat{\sigma}^2 = \sum_{i=1}^{I}\frac{(x_i-\hat{\mu})^2}{I}$$

Codigo en python:

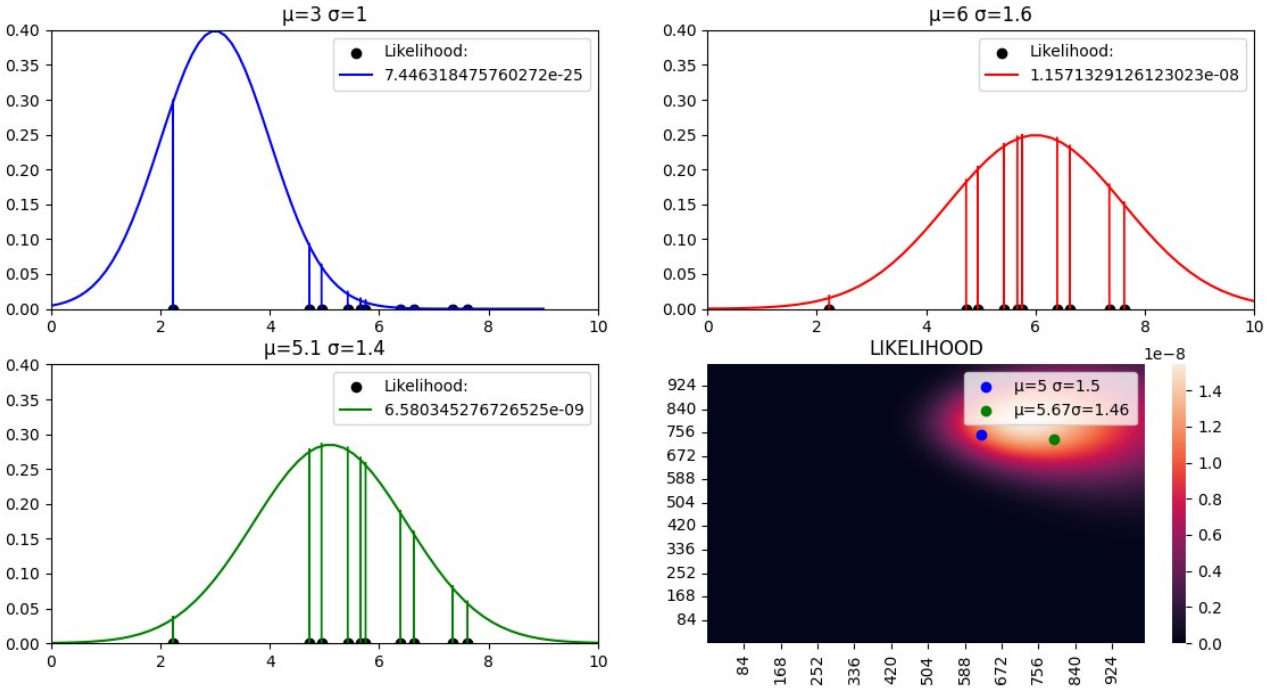
```
import pandas as pd
import numpy as np
import seaborn as sns
import random
import matplotlib.pyplot as plt
import math

#calculo de likelihood
def ml (mu,sigma,xi):
        suma=0
        for i in range(100):
                suma=((xi[i]-mu)**2)/(sigma**2)+suma

        expo=math.exp(-0.5*suma)
        aux=1/((2*math.pi*(sigma**2))**50)
        return aux*expo

def normal (x,mu,sigma):
        return np.exp(-1*((x-mu)**2)/(2*(sigma**2)))/(math.sqrt(2*np.pi) * sigma)

# VALORES DADOS
mux=5
sigmax=1.5
mu1=3
sigma1=1
mu2=6
sigma2=1.6
mu3=5.1
sigma3=1.4

nums=np.array([])
ynums=np.array([])
auxi=0
auxi2=0
sigmap=0

#CREACION DE PUNTOS ALEATORIOS BASADO EN DATOS DE GAUSSIANA
for i in range(100):
        x=random.gauss(mux, sigmax)
        nums=np.append(nums,x)
        ynums=np.append(ynums,0)
        auxi=auxi+x
mup=auxi/100

for i in range(100):
```

```
        auxi2=((nums[i]-mup)**2)/100
        sigmap=sigmap+auxi2

sigmap=sigmap**0.5
mlf=ml(mup,sigmap,nums)

#CREACION DE 3 GAUSSIANAS
plt.subplot(2,2,1)
li1=ml(mu1,sigma1,nums)
plt.scatter(nums,ynums,color='k')
plt.title('μ=3 σ=1')
plt.xlim(0,10)
plt.ylim(0,0.40)
x1=np.linspace(mu1-6*sigma1,mu1+6*sigma1,100)
y1=normal(x1,mu1,sigma1)
plt.plot(x1,y1,'b')
plt.legend(['Likelihood:',li1])
for i in range (100):
        plt.arrow(nums[i],ynums[i], 0,normal(nums[i],mu1,sigma1),color='b')

plt.subplot(2,2,2)
li2=ml(mu2,sigma2,nums)
plt.scatter(nums,ynums,color='k')
plt.title('μ=6 σ=1.6')
plt.xlim(0,10)
plt.ylim(0,0.40)
x2=np.linspace(mu2-6*sigma2,mu2+6*sigma2,100)
y2=normal(x2,mu2,sigma2)
plt.plot(x2,y2,'r')
plt.legend(['Likelihood:',li2])
for i in range (100):
        plt.arrow(nums[i],ynums[i], 0,normal(nums[i],mu2,sigma2),color='r')

plt.subplot(2,2,3)
li3=ml(mu3,sigma3,nums)
plt.scatter(nums,ynums,color='k')
plt.title('μ=5.1 σ=1.4')
plt.xlim(0,10)
plt.ylim(0,0.40)
x3=np.linspace(mu3-6*sigma3,mu3+6*sigma3,100)
y3=normal(x3,mu3,sigma3)
plt.plot(x3,y3,'g')
plt.legend(['Likelihood:',li3])
for i in range (100):
        plt.arrow(nums[i],ynums[i], 0,normal(nums[i],mu3,sigma3),color='g')

#CREACION DE MAPA DE CALOR DE LIKELIHOOD
```

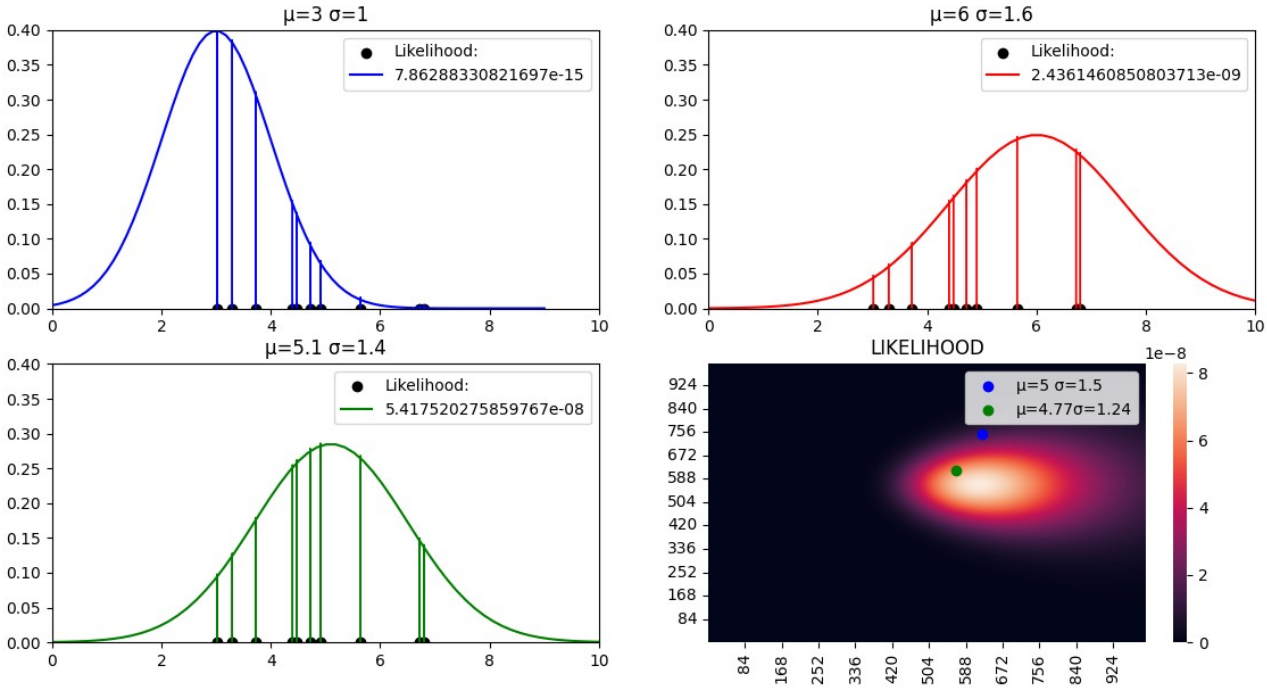
```
mus=np.array([])
for i in range(0,1001,1):
        mus=np.append(mus,(i*0.004)+2.5)
        if (i*0.004)+2.5==mux:
                mux1=i
        if (i*0.004)+2.5==mup:
                mup1=i

sigmas=np.array([])
for i in range(0,1001,1):
        sigmas=np.append(sigmas,i*0.002)

mls=np.zeros((1000,1000))
for i in range (1,1000,1) :
        for j in range(1,1000,1):
                mls[i][j]=ml(mus[i],sigmas[j],nums)


plt.subplot(2,2,4)
sns.heatmap(mls)
plt.scatter(mux1,sigmax*500,color='blue')
plt.scatter((mup-2.5)*250,sigmap*500,color='green')
mu=str('μ=')
muc=repr(round(mup,2))
sigc=repr(round(sigmap,2))
sig=str('σ=')
pt=mu+muc+sig+sigc
plt.legend(['μ=5 σ=1.5',pt])
plt.xlim(1,1000)
plt.ylim(1,1000)
plt.title('LIKELIHOOD')
plt.show()
```

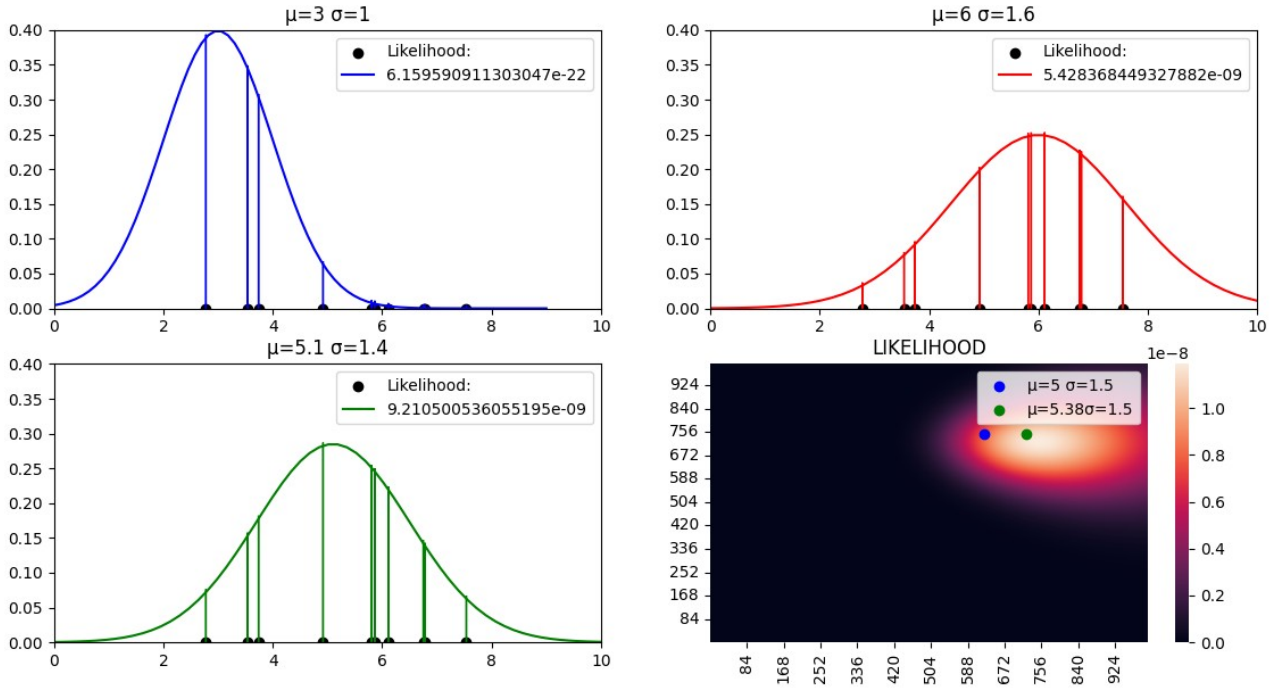
A la hora de correr el código se pudo notar que que al generar los datos de la gaussiana de manera aleatoria en las diferentes corridas en que se ejecuta el código los valores de likelihood van cambiando en cada una de las tres gaussianas establecidas, haciendo que incluso en algunas corridas la gaussiana de μ2=6, σ2=1.6 sea la que tenga un mayor likelihood, cuando lo común es que la gaussiana de μ3=5.1, σ3=1.4 sea la que tiene el mayor likelihood, esto debido a que es la tiene valor de μ y σ mas cercano a los valores con los que se generaron los datos aleatorios.

Como se puede ver en el siguiente grafico, este fue uno de los pocos casos en los que era posible que los valores aleatorios fueran mas inclinados hacia la derecha, haciendo que el likelihood de la gaussiana roja sea la de mayo valor:

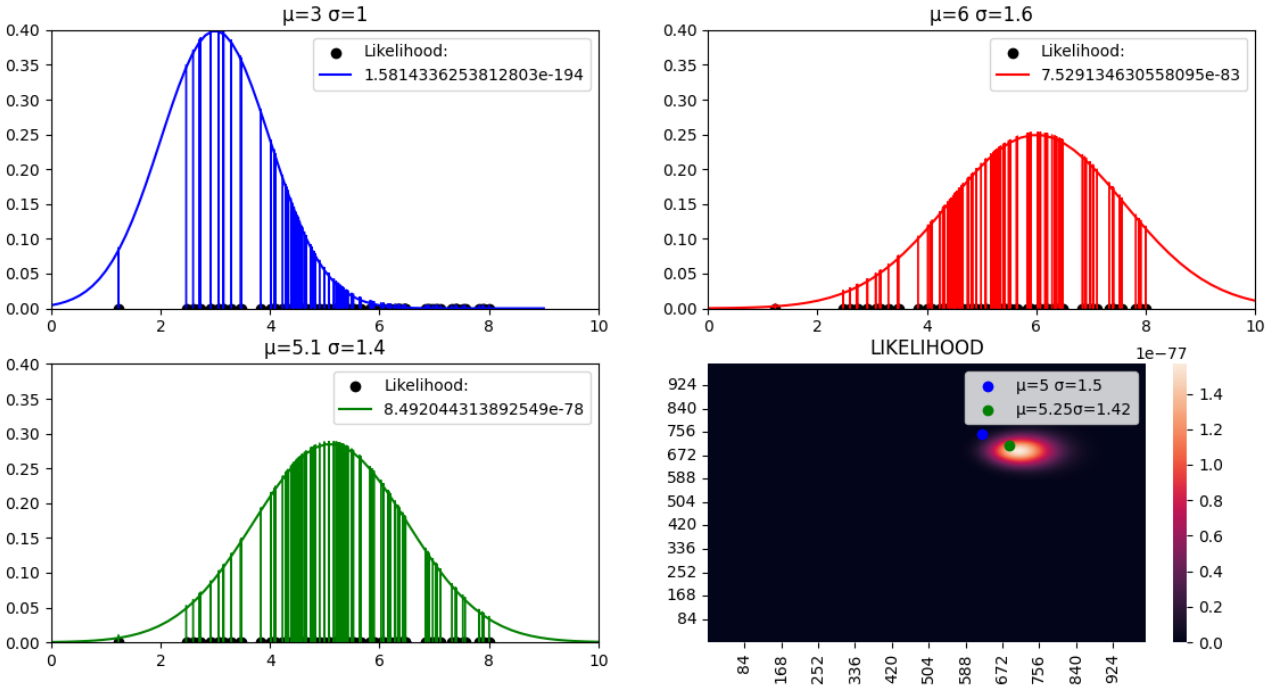


En las graficas siguientes, se puede observar que el likelihood mas alto es el de la gaussiana verde, que es lo que se espera que suceda de manera mas común de acuerdo a los datos de μ y σ establecidos al inicio.





En el siguiente grafico se puede observar como al aumentar el numero de valores aleatorios generados a cien, el mapa térmico se disminuye en su área haciendo que solo una parte especifica sea la que tiene mayor probabilidad comparada con los gráficos anteriores donde el área era de mayor tamaño.



Finalmente se puede concluir que a pesar de generar datos de manera aleatoria, al estar generados de acuerdo a una distribución gaussiana especifica estos tenderán a estar en un rango especifico, aunque en una que otra corrida pueda ser que los datos lleguen a estar tendidos mas hacia un lado u otro, ademas de que mientras mas datos se vayan generando el mapa térmico ira teniendo un área menor donde los likelihoods tendrán mayor valor.