Nombre y Apellidos: Washington Acero M.

Distribución gaussiana: practica de lab N 9

Implementar el algoritmo de detección de anomalías en un lenguaje funcional.

```
# recibira una columna
def f_sigma(x_i ,u_i):
    m = len(x_i)
    return np.sum((x_i - u_i)**2)/m
```

$$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m}(x_i^{(j)} - \mu_i)^2.$$

Implementación de función para obtener el valor de sigma al cuadrado:

```
# x_j, u_j, sigma_j
def __p(x, u, sigma):
    ex = (-1) * ((x - u)**2)/(2 * sigma)
    return np.exp(ex) / np.sqrt((2 * np.pi * sigma))
```

$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

Implementación de la función p(x,u,sigma)

```
def p1(x, u, sigma, m, n):
    res = np.ones(m)
    for j in range(n):
        res *= __p(x[:,j],u[j],sigma[j])
    return res
```

$$p(x) = \prod_{j=1}^{n} p(x_j;\mu_j,\sigma_j^2)$$

Función p(x)

Probar el algoritmo con un conjunto de datos de prueba(Ejm. libreria Python).

PRUEBA:

```
    # caractertica_1 = c1
    # caractertica_2 = c2
    # caractertica_2 = c3
    #             +--+--+--+
    #             |c1|c2|c3|
    #             +--+--+--+
    # sujeto 1    |  |  |  |
    # sujeto 2    |  |  |  |
    # sujeto 3    |  |  |  |
    # sujeto 4    |  |  |  |
    #             +--+--+--+

    datos = [
        # anómalo
        [1,3,11],
        # normal
        [4,8,6],
        [5,7,5],
        [4,9,4],
        [3,7,7]
    ]

    # épsilon = 0.0005
    # para [1,3,11] el elemento p[0,:]
    # p(x) = 0.000067
    # 0.000067 < 0.0005
    # True
    # Podemos decir que existe anomalía para esta muestra
    # a simple vista se puede ver que esta fila(tupla) tiene valores
muy diferentes a los de su columna

    # para [4,8,6] el elemento p[1,:]
    # p(x) = 0.007023
    # 0.007023 < 0.0005
    # false
    # esta muestra no presenta anomalías, los valores de esta
fila(tupla) son similares a las demás muestras no hay una gran
diferencia
```