**----------------------------performance metrics----------------EXERCISE#1-----------------------------------------**

import numpy as np

x=input('Enter Shape (round/ellipse)')

if x=='round':

shape=1

else:

shape=-1

x=input('Enter texture (smooth/rough)')

if x=='smooth':

texture=1

else:

texture=-1

x=int(input('Enter weight in pound'))

if x>=1:

wt=1

else:

wt=-1

in\_vect=np.array([shape,texture,wt])

print(in\_vect.T)

w=np.array([0,1,0])

b=0

#

y=(w\*x).sum()+b

#

def perceptron(x):

if x>=0:

return 1

else:

return -1

out=perceptron(y)

print(out)

if out==1:

print('APPLE')

else:

print('ORANGE')

**----------------------------Data Division----------------EXERCISE#2-----------------------------------------**

**Take different wts.**

----------------------------Activation function----------------EXERCISE#3-----------------------------------------

import numpy as np

x=np.arange(1,11,dtype=np.int16)

print(x)

print(x.size)

#

w=np.random.randint(-10,10,size=10,dtype=np.int16)

print(w)

print(w.size)

#

b=np.random.random()

print(b)

#

net\_input=((w\*x).sum()+b)

print(net\_input)

#threshold

def threshold(x):

if x>=0:

return 1

else:

return 0

out=threshold(net\_input)

print(out)