**----------------------------KNN ----------------EXERCISE#1-----------------------------------------**

from sklearn import datasets

import numpy as np

iris=datasets.load\_iris()

X=iris.data[:,[2,3]] # only two features

y=iris.target

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(

X,y,test\_size=0.3,random\_state=0)

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

sc.fit(X\_train)

X\_train\_std=sc.transform(X\_train)

X\_test\_std=sc.transform(X\_test)

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=5, p=2,

metric='minkowski')

knn.fit(X\_train\_std, y\_train)

y\_pred=knn.predict(X\_test\_std)

print('misclassified samples: %d'%(y\_test!=y\_pred).sum())#compute

from sklearn.metrics import accuracy\_score

print('Accuracy:%.2f'%accuracy\_score(y\_test,y\_pred))

**---------------------------- GaussNB ----------------EXERCISE#2-----------------------------------------**

import numpy as np

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

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

from sklearn.naive\_bayes import GaussianNB

clf = GaussianNB()

GaussianNB(priors=None)

clf.fit(X, Y)

print(clf.predict([[1, 1,0]]))

print(clf.predict([[2, 0,0]]))

y=clf.predict(X)

print('misclassified samples: %d'%(y!=Y).sum())#compute

from sklearn.metrics import accuracy\_score

print('Accuracy:%.2f'%accuracy\_score(y,Y))