**---------------------------- ----------------EXERCISE#1-----------SVM------------------------------**

from sklearn import datasets

import numpy as np

iris=datasets.load\_iris()

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

y=iris.target

print(y)

print(X)

print(np.unique(y)) # unique value

###

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)

print(np.shape(X))

print(np.shape(X\_train))

print(np.shape(X\_test))

####

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.svm import SVC

svm = SVC(kernel='linear', C=1, random\_state=0)

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

y\_pred=svm.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))

**---------------------------- Home Assignment-----------------------------------------**

---------------------------- ----------------EXERCISE#3-----------------------------------------

from sklearn import datasets

import numpy as np

iris=datasets.load\_wine()

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

y=iris.target

print(y)

print(X)

print(np.unique(y)) # unique value

###

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)

print(np.shape(X))

print(np.shape(X\_train))

print(np.shape(X\_test))

####

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.svm import SVC

svm = SVC(kernel='linear', C=1, random\_state=0)

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

y\_pred=svm.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))