**---------------------------STD/NORM- ----------------EXERCISE#1-----------------------------------------**

import numpy as np

import pandas as pd

df\_wine = pd.read\_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data', header=None)

df\_wine.columns = ['Class label', 'Alcohol',

'Malic acid', 'Ash',

'Alcalinity of ash', 'Magnesium',

'Total phenols', 'Flavanoids',

'Nonflavanoid phenols',

'Proanthocyanins',

'Color intensity', 'Hue',

'OD280/OD315 of diluted wines',

'Proline']

print('Class labels', np.unique(df\_wine['Class label']))

print(df\_wine.head())

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

X, y = df\_wine.iloc[:, 1:].values, df\_wine.iloc[:, 0].values

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

stdsc = StandardScaler()

X\_train\_std = stdsc.fit\_transform(X\_train)

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

##

from sklearn.preprocessing import MinMaxScaler

mms=MinMaxScaler()

X\_train\_norm = mms.fit\_transform(X\_train)

X\_test\_norm = mms.transform(X\_test)

##

print(X\_train[1,:])

print(X\_train\_std[1,:])

print(X\_train\_norm[1,:])

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

import numpy as np

import pandas as pd

df\_wine = pd.read\_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data', header=None)

df\_wine.columns = ['Class label', 'Alcohol',

'Malic acid', 'Ash',

'Alcalinity of ash', 'Magnesium',

'Total phenols', 'Flavanoids',

'Nonflavanoid phenols',

'Proanthocyanins',

'Color intensity', 'Hue',

'OD280/OD315 of diluted wines',

'Proline']

print('Class labels', np.unique(df\_wine['Class label']))

print(df\_wine.head())

#

from sklearn.cross\_validation import train\_test\_split

X, y = df\_wine.iloc[:, 1:].values, df\_wine.iloc[:, 0].values

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

stdsc = StandardScaler()

X\_train\_std = stdsc.fit\_transform(X\_train)

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

#

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression(penalty='l1', C=0.1)

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

print('Training accuracy:', lr.score(X\_train\_std, y\_train))

print('Test accuracy:', lr.score(X\_test\_std, y\_test))

print(lr.intercept\_)

print(lr.coef\_)

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