**----------------------------Gini\_entropy\_misclassification--------EXERCISE#1-----------------------------------------**

import matplotlib.pyplot as plt

import numpy as np

def gini(p):

return (p)\*(1 - (p)) + (1 - p)\*(1 - (1-p))

def entropy(p):

return - p\*np.log2(p) - (1 - p)\*np.log2((1 - p))

def error(p):

return 1 - np.max([p, 1 - p])

x = np.arange(0.0, 1.0, 0.01)

ent = [entropy(p) if p != 0 else None for p in x]

err = [error(i) for i in x]

fig = plt.figure()

ax = plt.subplot(111)

for i, lab, ls, c, in zip([ent,gini(x), err],

['Entropy','Gini Impurity','Misclass Error'],

['-','--', '-.'],

['black','red', 'green']):

line = ax.plot(x, i, label=lab,

linestyle=ls, lw=2, color=c)

ax.legend()

ax.axhline(y=0.5, linewidth=1, color='k', linestyle='--')

ax.axhline(y=1.0, linewidth=1, color='k', linestyle='--')

plt.ylim([0, 1.1])

plt.xlabel('p(i=1)')

plt.ylabel('Impurity Index')

plt.show()

**----------------------------Decision Tree Classifier ----------------EXERCISE#2-----------------------------------------**

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.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(criterion='entropy',

max\_depth=3, random\_state=0)

tree.fit(X\_train, y\_train)

y\_pred=tree.predict(X\_test)

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))

#to perform graph visulization

from sklearn.tree import export\_graphviz

export\_graphviz(tree,

out\_file='tree.dot',

feature\_names=['petal length', 'petal width'])

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

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.ensemble import RandomForestClassifier

forest = RandomForestClassifier(criterion='entropy',

n\_estimators=10,

random\_state=1,

n\_jobs=2)

forest.fit(X\_train, y\_train)

y\_pred=forest.predict(X\_test)

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))