EX:3

a).  
from sklearn.datasets import load\_breast\_cancer

from sklearn.model\_selection import train\_test\_split,cross\_val\_score,learning\_curve

from sklearn.metrics import accuracy\_score

from sklearn.tree import DecisionTreeClassifier

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

data=load\_breast\_cancer()

X=data.data

y=data.target

x\_train,x\_test,y\_train,y\_test=train\_test\_split(X,y,random\_state=42)

dt=DecisionTreeClassifier(random\_state=42,max\_de)

dt.fit(x\_train,y\_train)

train\_acc=accuracy\_score(y\_train,dt.predict(x\_train))

test\_acc=accuracy\_score(y\_test,dt.predict(x\_test))

print("accuracy:\n1.train accuracy:",train\_acc,"\n2.test accuracy",test\_acc)

cv\_val\_score=cross\_val\_score(dt,X,y,cv=5)

print("cv score:",cv\_val\_score)

print("average cv score:",np.mean(cv\_val\_score))

print("std:",np.std(cv\_val\_score))

train\_size,train\_score\_dt,test\_score\_dt=learning\_curve(dt,X,y,cv=5,scoring='accuracy',train\_sizes=np.linspace(0.1,1.0,10),random\_state=42)

plt.figure(figsize=(10,6))

plt.plot(train\_size,np.mean(train\_score\_dt,axis=1),'o-', label='training Score')

plt.plot(train\_size,np.mean(test\_score\_dt,axis=1),'o-', label='validation Score')

plt.title("decision tree - learning curve")

plt.xlabel("training Size")

plt.ylabel("accuracy")

plt.legend()

plt.grid(True)

plt.show()  
  
output:  
accuracy:

1.train accuracy: 1.0

2.test accuracy 0.951048951048951

cv score: [0.9122807 0.90350877 0.92982456 0.95614035 0.88495575]

average cv score: 0.9173420276354604

std: 0.02419491828674519  
  
  
b).

from sklearn.ensemble import BaggingClassifier

bagging\_clf = BaggingClassifier(estimator=DecisionTreeClassifier(), n\_estimators=100, random\_state=42)

bagging\_clf.fit(x\_train, y\_train)

train\_acc=accuracy\_score(y\_train,bagging\_clf.predict(x\_train))

test\_acc=accuracy\_score(y\_test,bagging\_clf.predict(x\_test))

print("accuracy:\n1.train accuracy:",train\_acc,"\n2.test accuracy",test\_acc)

cv\_val\_score=cross\_val\_score(bagging\_clf,X,y,cv=5)

print("cv score:",cv\_val\_score)

print("average cv score:",np.mean(cv\_val\_score))

train\_size,train\_score\_bc,test\_score\_bc=learning\_curve(bagging\_clf,X,y,cv=5,scoring='accuracy',train\_sizes=np.linspace(0.1,1.0,10),random\_state=42)

plt.figure(figsize=(10,6))

plt.plot(train\_size,np.mean(train\_score\_bc,axis=1),'o-', label='training Score')

plt.plot(train\_size,np.mean(test\_score\_bc,axis=1),'o-', label='validation Score')

plt.title("bagging clf - learning curve")

plt.xlabel("training Size")

plt.ylabel("accuracy")

plt.legend()

plt.grid(True)

plt.show()

output:  
accuracy:

1.train accuracy: 1.0

2.test accuracy 0.958041958041958

cv score: [0.89473684 0.93859649 0.99122807 0.96491228 1. ]

average cv score: 0.9578947368421054  
