Compile Result:

```
================================================ RESTART: D:\大學課業\大四\ML\0811002Hw5.py =
======
(1)Logistic Regression:
C=0.1
5-fold cross validation scores[0.96842105 0.94736842 0.90526316 0.95789474 0.96842105]
5-fold cross validation average score: 0.949
train score=0.952, test score= 0.936
C=1
5-fold cross validation scores[0.97894737 0.95789474 0.90526316 0.96842105 0.96842105]
5-fold cross validation average score: 0.956
train score=0.962, test score= 0.936
C=10
5-fold cross validation scores[0.96842105 0.96842105 0.92631579 0.96842105 0.96842105]
5-fold cross validation average score: 0.960
train score=0.971, test score= 0.947
C=100
5-fold cross validation scores[0.96842105 0.97894737 0.93684211 0.97894737 0.97894737]
5-fold cross validation average score: 0.968
train score=0.979, test score= 0.947
C=1000
5-fold cross validation scores[0.94736842 0.96842105 0.92631579 0.98947368 0.97894737]
5-fold cross validation average score: 0.962
train score=0.983, test score= 0.979
(1)Maximum average test score is 0.970, when C value equal to 1000.00.
(2)Random Forest:
number of estimators =100
5-fold cross validation scores:[0.95789474 0.96842105 0.90526316 0.95789474 0.97894737]
5-fold cross validation average score: 0.954
score Train/Test: 1.000/0.957
number of estimators =110
5-fold cross validation scores:[0.92631579 0.96842105 0.92631579 0.96842105 0.97894737]
5-fold cross validation average score: 0.954
score Train/Test: 1.000/0.968
number of estimators =120
5-fold cross validation scores:[0.93684211 0.96842105 0.92631579 0.95789474 0.96842105]
5-fold cross validation average score: 0.952
```

```
number of estimators =120
5-fold cross validation scores:[0.93684211 0.96842105 0.92631579 0.95789474 0.96842105]
5-fold cross validation average score: 0.952
score Train/Test: 1.000/0.957
number of estimators =130
5-fold cross validation scores:[0.95789474 0.96842105 0.92631579 0.95789474 0.98947368]
5-fold cross validation average score: 0.960
score Train/Test: 1.000/0.968
number of estimators =140
5-fold cross validation scores:[0.94736842 0.97894737 0.92631579 0.95789474 0.97894737]
5-fold cross validation average score: 0.958
score Train/Test: 1.000/0.968
number of estimators =150
5-fold cross validation scores:[0.92631579 0.96842105 0.92631579 0.95789474 0.96842105]
5-fold cross validation average score: 0.949
score Train/Test: 1.000/0.957
number of estimators =160
5-fold cross validation scores:[0.94736842 0.96842105 0.92631579 0.95789474 0.97894737]
5-fold cross validation average score: 0.956
score Train/Test: 1.000/0.947
number of estimators =170
5-fold cross validation scores:[0.91578947 0.96842105 0.92631579 0.95789474 0.98947368]
5-fold cross validation average score: 0.952
score Train/Test: 1.000/0.957
number of estimators =180
5-fold cross validation scores:[0.95789474 0.96842105 0.92631579 0.95789474 0.97894737]
5-fold cross validation average score: 0.958
score Train/Test: 1.000/0.968
number of estimators =190
5-fold cross validation scores:[0.93684211 0.96842105 0.92631579 0.95789474 0.97894737]
5-fold cross validation average score: 0.954
score Train/Test: 1.000/0.957
number of estimators =200
5-fold cross validation scores:[0.93684211 0.96842105 0.92631579 0.95789474 0.98947368]
5-fold cross validation average score: 0.956
score Train/Test: 1.000/0.957
```

```
number of estimators =190
5-fold cross validation scores:[0.93684211 0.96842105 0.92631579 0.95789474 0.97894737]
5-fold cross validation average score: 0.954
score Train/Test: 1.000/0.957
number of estimators =200
5-fold cross validation scores:[0.93684211 0.96842105 0.92631579 0.95789474 0.98947368]
5-fold cross validation average score: 0.956
score Train/Test: 1.000/0.957
(2)Maximum test score is 0.964, while number of estimators is 130.
(3)GradientBoosting
Learning rate is 0.010
5-fold cross validation scores: [0.90526316 0.93684211 0.88421053 0.95789474 0.95789474]
5-fold cross validation average score: 0.928
train/test scores:0.983/0.968
Learning rate is 0.100
5-fold cross validation scores: [0.95789474 0.98947368 0.92631579 0.95789474 0.97894737]
5-fold cross validation average score: 0.962
train/test scores:1.000/0.957
Learning rate is 1.000
5-fold cross validation scores: [0.96842105 0.96842105 0.92631579 0.94736842 0.98947368]
5-fold cross validation average score: 0.960
train/test scores:1.000/0.957
Learning rate is 10.000
5-fold cross validation scores: [0.86315789 0.36842105 0.34736842 0.76842105 0.89473684]
5-fold cross validation average score: 0.648
train/test scores:0.709/0.691
Learning rate is 100.000
5-fold cross validation scores: [0.34736842 0.48421053 0.58947368 0.10526316 0.67368421]
5-fold cross validation average score: 0.440
train/test scores:0.088/0.053
Learning rate is 1000.000
5-fold cross validation scores: [0.90526316 0.91578947 0.88421053 0.10526316 0.91578947]
5-fold cross validation average score: 0.745
train/test scores:0.657/0.585
(3)Maximum test score is 0.960, while learning rate is 0.100.
```

Code Screenshot:

0811002 Shwan HW5.py - D:\大學課業\大四\ML\0811002 Shwan HW5.py (3.9.12)

File Edit Format Run Options Window Help

```python
#Hw5 Handout
#target: avg.validation score-->the higher, the better. Test_score>=0.940
import sklearn
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import cross_val_score
##############################################################
def readHw5Cancer(inFileName):
    # init
    recArr = []
    clsArr = []

    # open input text data file, format is given
    inFile = open(inFileName, 'r')
    s = inFile.readline() # skip

    row = 0
    while True:
        s = inFile.readline()
        datal = s.strip() # remove leading and ending blanks
        if (len(datal) <= 0):
            break

        # since we use append, value must be created in the loop
        value = []

        strs31 = datal.split(',') # array of 31 str

        # convert to real
        for ix in range(30):
            value.append( eval(strs31[ix]) )
        # end for
```

```
        # end for

        target = eval(strs31[30])

        recArr.append(value) ;  # add 1 record at end of array
        clsArr.append(target) ; # add 1 record at end of array

        row = row+1 # total read counter
    # end while

    # close input file
    inFile.close()

    # convert list to Numpy array
    npXY = np.array(recArr)
    npC  = np.array(clsArr)

    # pass out as Numpy array
    return npXY, npC

# end function

#Main starts
X,y=readHw5Cancer("d:\\temp\\breast_cancer_scikit_Xy.csv")

#print(X_5fold.shape)
#print(X_test.shape)
totalnum=569
nfold = 5
mxi = 100000

#LogisticRegression
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 94, random_state = 0)
cvalue=[0.1,1,10,100,1000]
lavgscores = []
print("(1)Logistic Regression:")
```

```
#Main starts
X,y=readHw5Cancer("d:\\temp\\breast_cancer_scikit_Xy.csv")

#print(X_5fold.shape)
#print(X_test.shape)
totalnum=569
nfold = 5
mxi = 100000

#LogisticRegression
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 94, random_state = 0)
cvalue=[0.1,1,10,100,1000]
lavgscores = []
print("(1)Logistic Regression:")
for C in cvalue:
    logreg = LogisticRegression(C=C,max_iter=mxi).fit(X_train,y_train)
    train_score = logreg.score(X_train,y_train)
    scores = cross_val_score(logreg,X_train,y_train,cv=nfold,scoring='accuracy')
    test_score = logreg.score(X_test,y_test)
    lavg = 0.5*(test_score+scores.mean())
    print("C="+str(C))
    print('5-fold cross validation scores'+str(scores))
    print("5-fold cross validation average score: {:.3f}".format(scores.mean()))
    print("train score={:.3f}, test score= {:.3f}".format(train_score,test_score))
    lavgscores.append(lavg)
max_lscore = 0.
max_i = 0
for i, num in enumerate(lavgscores):
    if (max_lscore == 0 or num > max_lscore):
        max_lscore = num
        max_i = i

print("(1)Maximum average test score is {:.3f}, when C value equal to {:.2f}. ".format(max_lscore,cvalue[max_i]))

#RandomForest
print("(2)Random Forest: ")
```

```
print("(1)Maximum average test score is {:.3f}, when C value equal to {:.2f}. ".format(max_lscore,cvalue[max_i]))

#RandomForest
print("(2)Random Forest: ")
favgscores=[]
estinum = list(range(100,210,10))
for ix in estinum:
    forest = RandomForestClassifier(n_estimators=ix ,criterion="gini", max_depth=None, bootstrap=True, random_state=None)
    forest.fit(X_train, y_train)
    ftrains = forest.score(X_train,y_train)
    fscores = cross_val_score(forest,X_train,y_train,cv=nfold,scoring='accuracy')
    ftests = forest.score(X_test,y_test)
    favg = 0.5*(ftests+fscores.mean())
    favgscores.append(favg)
    print("number of estimators ={:d} ".format(ix))
    print("5-fold cross validation scores:"+str(fscores))
    print("5-fold cross validation average score: {:.3f}".format(fscores.mean()))
    print("score Train/Test: {:.3f}/{:.3f}".format(ftrains,ftests))
    #print(ftrains)
max_fscore = 0.
max_j = 0
for j, num in enumerate(favgscores):
    if (max_fscore == 0 or num > max_fscore):
        max_fscore = num
        max_j = j
print("(2)Maximum test score is {:.3f}, while number of estimators is {:d}. ".format(max_fscore,estinum[max_j]))

#GradientBoosting
learnrate = [0.01,0.1,1,10,100,1000]
gavgscores = []
print("(3)GradientBoosting")
for x in learnrate:
    gbrt = GradientBoostingClassifier(learning_rate=x,n_estimators=100,random_state=0)
    gbrt.fit(X_train,y_train)
    g_test = gbrt.score(X_test,y_test)
    g_train = gbrt.score(X_train,y_train)
```

0811002 Shwan HW5.py - D:\大學課業\大四\ML\0811002 Shwan HW5.py (3.9.12)

File Edit Format Run Options Window Help

```
    print("5-fold cross validation scores:"+str(fscores))
    print("5-fold cross validation average score: {:.3f}".format(fscores.mean()))
    print("score Train/Test: {:.3f}/{:.3f}".format(ftrains,ftests))
    #print(ftrains)
max_fscore = 0.
max_j = 0
for j, num in enumerate(favgscores):
    if (max_fscore == 0 or num > max_fscore):
        max_fscore = num
        max_j = j
print("(2)Maximum test score is {:.3f}, while number of estimators is {:d}. ".format(max_fscore,estinum[max_j]))

#GradientBoosting
learnrate = [0.01,0.1,1,10,100,1000]
gavgscores = []
print("(3)GradientBoosting")
for x in learnrate:
    gbrt = GradientBoostingClassifier(learning_rate=x,n_estimators=100,random_state=0)
    gbrt.fit(X_train,y_train)
    g_test = gbrt.score(X_test,y_test)
    g_train = gbrt.score(X_train,y_train)
    gscores = cross_val_score(gbrt,X_train,y_train,cv=nfold,scoring='accuracy')
    gavg = 0.5*(g_test+gscores.mean())
    gavgscores.append(gavg)
    print("Learning rate is {:.3f}".format(x))
    print("5-fold cross validation scores: "+str(gscores))
    print("5-fold cross validation average score: {:.3f}".format(gscores.mean()))
    print("train/test scores:{:.3f}/{:.3f} ".format(g_train,g_test))
max_gscore = 0.
max_j = 0
for k, num in enumerate(gavgscores):
    if (max_gscore == 0 or num > max_gscore):
        max_gscore = num
        max_k = k
print("(3)Maximum test score is {:.3f}, while learning rate is {:.3f}. ".format(max_gscore, learnrate[max_k]))
```