0316219 林語新

Programming Language: Python 3
Environment: OS: MACOS 10.12.3
Import/Using Library: numpy spicy scikit-learn

Code:

importing libraries

```
import numpy as np
from sklearn import tree
from sklearn.datasets import load_iris
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier
```

load and print iris

```
iris= load_iris()
print(iris)
```

Resubstitution Validation

```
print("Resubstitution Validation (pre-pruning depth<3 with boosting&bagging)\n")
dtc= BaggingClassifier(AdaBoostClassifier( tree.DecisionTreeClassifier(max_depth=3)))
```

decision tree with bagging and boost

```
r= dtc.fit(iris.data, iris.target)
print("Score: ", r.score(iris.data, iris.target))
```

The Classification Accuracy

```
pred= dtc.predict(iris.data)
print(pred)
```

The prediction

```
cnf= confusion_matrix(iris.target, pred)
```

Confusion Matrix

```
print(cnf)
```

```
print("Resubstitution Validation (pre-pruning depth<3)\n")
```

```
digraph Tree {
node [shape=box] ;
0 [label="X[2] <= 2.45\ngini = 0.6667\nsamples = 150\nvalue = [50, 50, 50]"] ;
1 [label="gini = 0.0\nsamples = 50\nvalue = [50, 0, 0]"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label="X[3] <= 1.75\ngini = 0.5\nsamples = 100\nvalue = [0, 50, 50]"] ;
0 -> 2 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
3 [label="X[2] <= 4.95\ngini = 0.168\nsamples = 54\nvalue = [0, 49, 5]"] ;
2 -> 3 ;
4 [label="gini = 0.0408\nsamples = 48\nvalue = [0, 47, 1]"] ;
3 -> 4 ;
5 [label="gini = 0.4444\nsamples = 6\nvalue = [0, 2, 4]"] ;
3 -> 5 ;
6 [label="X[2] <= 4.85\ngini = 0.0425\nsamples = 46\nvalue = [0, 1, 45]"] ;
2 -> 6 ;
7 [label="gini = 0.4444\nsamples = 3\nvalue = [0, 1, 2]"] ;
6 -> 7 ;
8 [label="gini = 0.0\nsamples = 43\nvalue = [0, 0, 43]"] ;
6 -> 8 ;
}
```

```
dtc= tree.DecisionTreeClassifier(max_depth=3)
r= dtc.fit(iris.data, iris.target)
print("Score: ", r.score(iris.data, iris.target))
pred= dtc.predict(iris.data)
print(pred)
cnf= confusion_matrix(iris.target, pred)
print(cnf)
```

K-fold Validation

repeat with k=2,5,10

```
print("\nK-fold cross validation (pre-pruning depth<3)")
print("\n[K-Fold K=2]")
kf2= KFold(n_splits=2, shuffle=True)

normal= []

for train, test in kf2.split(iris.data, iris.target):

    dtc=
tree.DecisionTreeClassifier(max_depth=3)
    r= dtc.fit(iris.data[train], iris.target[train])
    print("\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\")
    score = r.score(iris.data[test],
iris.target[test])
    print("Classification Accuracy = ", score)
    normal.append(score)
    pred= dtc.predict(iris.data[test])
    print(pred)
    cnf= confusion_matrix(iris.target[test], pred)
    print(cnf)

print("lllllllllllllllllllllllllllll")
print("Overall Classification accuracy= ", np.asarray(normal).mean())
```

```
digraph Tree {
node [shape=box] ;
0 [label="X[3] <= 0.8\ngini = 0.6657\nsamples = 135\nvalue = [48, 42, 45]"] ;
1 [label="gini = 0.0\nsamples = 48\nvalue = [48, 0, 0]"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label="X[3] <= 1.75\ngini = 0.4994\nsamples = 87\nvalue = [0, 42, 45]"] ;
0 -> 2 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
3 [label="X[2] <= 4.95\ngini = 0.1901\nsamples = 47\nvalue = [0, 42, 5]"] ;
2 -> 3 ;
4 [label="gini = 0.0476\nsamples = 41\nvalue = [0, 40, 1]"] ;
3 -> 4 ;
5 [label="gini = 0.4444\nsamples = 6\nvalue = [0, 2, 4]"] ;
3 -> 5 ;
6 [label="gini = 0.0\nsamples = 40\nvalue = [0, 0, 40]"] ;
2 -> 6 ;
}
```

Result:

We can see that in resubstitution validation with preprinting depth<3, boosting&bagging will raise the classification accuracy. Sometimes may reach to 1.

```
Resubstitution Validation (pre-pruning depth<3 with boosting&bagging)

Score:  1.0
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
[[50  0  0]
 [ 0 50  0]
 [ 0  0 50]]
Resubstitution Validation (pre-pruning depth<3)

Score:  0.973333333333
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
[[50  0  0]
 [ 0 47  3]
 [ 0  1 49]]
```

```
Resubstitution Validation (pre-pruning depth<3 with boosting&bagging)

Score:  0.993333333333
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
[[50  0  0]
 [ 0 49  1]
 [ 0  0 50]]
Resubstitution Validation (pre-pruning depth<3)

Score:  0.973333333333
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
[[50  0  0]
 [ 0 47  3]
 [ 0  1 49]]
```

K-fold K=2,5

K-fold K=10

```
[K-Fold K=10]
Classification Accuracy =  1.0
Classification Accuracy =  1.0
Classification Accuracy =  0.8
Classification Accuracy =  0.866666666667
Classification Accuracy =  1.0
Classification Accuracy =  0.866666666667
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
||||||||||||||||||||||||||
Overall Classification accuracy=  0.94

[K-Fold K=10] with boosting&bagging
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  0.866666666667
Classification Accuracy =  1.0
Classification Accuracy =  0.933333333333
Classification Accuracy =  0.933333333333
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  1.0
Classification Accuracy =  1.0
||||||||||||||||||||||||||
Overall Classification accuracy=  0.96
```

```
[K-Fold K=10]
Classification Accuracy =  1.0
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  0.866666666667
Classification Accuracy =  1.0
Classification Accuracy =  0.866666666667
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  1.0
Classification Accuracy =  0.933333333333
||||||||||||||||||||||||||
Overall Classification accuracy=  0.953333333333

[K-Fold K=10] with boosting&bagging
Classification Accuracy =  0.866666666667
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  1.0
Classification Accuracy =  0.933333333333
Classification Accuracy =  0.933333333333
Classification Accuracy =  1.0
Classification Accuracy =  1.0
Classification Accuracy =  0.866666666667
Classification Accuracy =  1.0
||||||||||||||||||||||||||
Overall Classification accuracy=  0.953333333333
```

```
[K-Fold K=2]
\\\\\\\\\\\\\\\\
Classification Accuracy =  0.946666666667
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 1 1 1 1 1 1 2 1 2
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2]
[[26  0  0]
 [ 0 22  3]
 [ 0  1 23]]
\\\\\\\\\\\\\\\\
Classification Accuracy =  0.946666666667
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2
 1 1 1 2 1 1 2 1 1 1 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2]
[[24  0  0]
 [ 0 22  3]
 [ 0  1 25]]
||||||||||||||||||||||||||||
Overall Classification accuracy=  0.946666666667

[K-Fold K=5]
\\\\\\\\\\\\\\\\
Classification Accuracy =  0.9
[0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 2 1 1 2 1 2 1 2 2 2 2 2 2 2]
[[11  0  0]
 [ 0  7  1]
 [ 0  2  9]]
\\\\\\\\\\\\\\\\
Classification Accuracy =  0.933333333333
[0 0 0 0 0 0 0 0 0 1 1 1 2 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2]
[[ 9  0  0]
 [ 0  9  2]
 [ 0  0 10]]
\\\\\\\\\\\\\\\\
Classification Accuracy =  0.9
[0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 1 1 1 1 2 2 1 1 1 2 2 2 2 2 2]
[[12  0  0]
 [ 0  9  3]
 [ 0  0  6]]
\\\\\\\\\\\\\\\\
Classification Accuracy =  0.966666666667
[0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1]
[[11  0  0]
 [ 0  7  0]
 [ 0  1 11]]
\\\\\\\\\\\\\\\\
Classification Accuracy =  1.0
[0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2]
[[ 7  0  0]
 [ 0 12  0]
 [ 0  0 11]]
||||||||||||||||||||||||||||
Overall Classification accuracy=  0.94
```