LAB1

November 9, 2021

1 LAB 1 Mahshid Ghaffari i6255207

[90]:
```
import pandas as pd
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

2 1.1) one-level decition tree (diabetes data)

with max_depth 1

[91]:
```
data = pd.read_csv('diabetes.csv')
Y = data['class']
X = data.drop(['class'],axis=1)
clf = tree.DecisionTreeClassifier(criterion = 'entropy',
max_depth = 1)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34, random_state=10)

clf = clf.fit(X_train, Y_train)
Yp = clf.predict(X_train)
accTrain = accuracy_score(Y_train, Yp)
print("Accuracy of the train data:",accTrain)

Yp = clf.predict(X_test)
accTest = accuracy_score(Y_test, Yp)
print("Accuracy of the test data:",accTest)
```

```
Accuracy of the train data: 0.7648221343873518
Accuracy of the test data: 0.7213740458015268
```

As you can see the accuracy of train data is more than test. this is due to that tree is made by train data

3 1.2) infinit-level decition tree (diabetes data)

with max_depth = None

[92]:
```
clf = tree.DecisionTreeClassifier(criterion = 'entropy',
max_depth = None)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34, random_state=10)

clf = clf.fit(X_train, Y_train)
Yp = clf.predict(X_train)
accTrain = accuracy_score(Y_train, Yp)
print("Accuracy of the train data:",accTrain)

Yp = clf.predict(X_test)
accTest = accuracy_score(Y_test, Yp)
print("Accuracy of the test data:",accTest)
```

```
Accuracy of the train data: 1.0
Accuracy of the test data: 0.7290076335877863
```

-we got above result for multi level decision tree, as you can see the accuracy of the train data is 1 beacuse there is no limitation for tree so we are going to have new branch per each data , but as test it's not one because still there are some data wich not include in tree

With min_samples_leaf as the size of the datasets

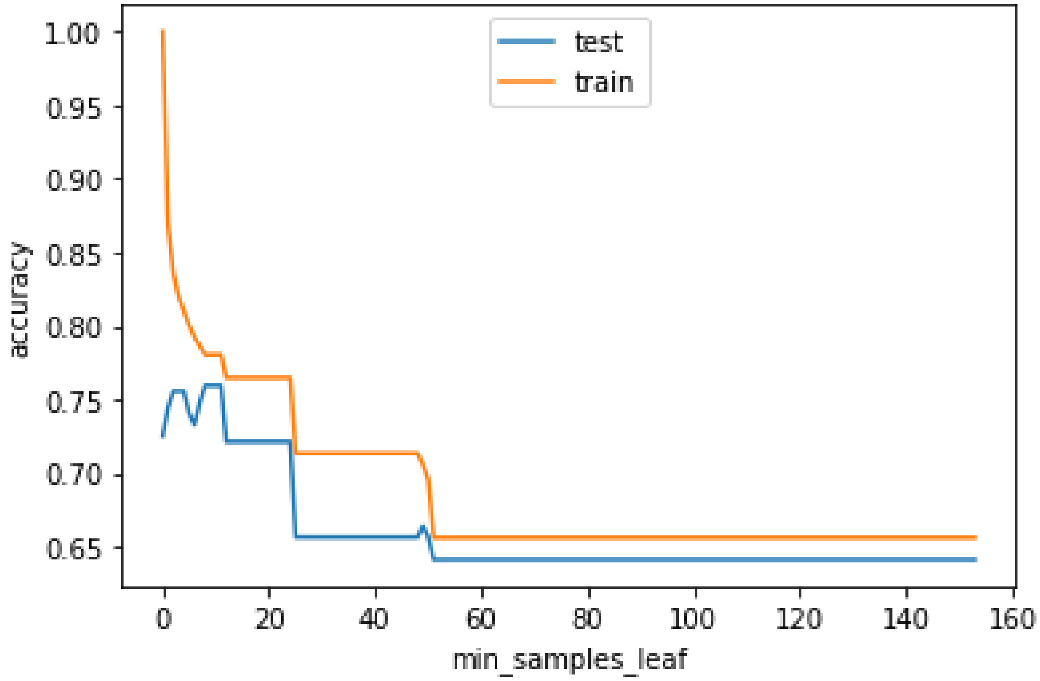
[93]:
```
clf = tree.DecisionTreeClassifier(criterion = 'entropy',
max_depth = None, min_samples_leaf = 768)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34, random_state=10)
clf = clf.fit(X_train, Y_train)
Yp = clf.predict(X_train)
accTrain = accuracy_score(Y_train, Yp)

print("Accuracy of the train data:",accTrain)
Yp = clf.predict(X_test)
accTest = accuracy_score(Y_test, Yp)
print("Accuracy of the test data:",accTest)
```

```
Accuracy of the train data: 0.6561264822134387
Accuracy of the test data: 0.6412213740458015
```

result :

[94]:
```
test =[]
train = []
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34,random_state=10)
```

```
for x in range(1,768,5):
    clf = tree.DecisionTreeClassifier(criterion = 'entropy',
    max_depth = None, min_samples_leaf = x)
    clf = clf.fit(X_train, Y_train)
    Yp_test = clf.predict(X_test)
    Yp_train = clf.predict(X_train)
    test.append(accuracy_score(Y_test, Yp_test))
    train.append(accuracy_score(Y_train, Yp_train))
plt.plot(test,label = 'test')
plt.plot(train,label = 'train')
plt.ylabel('accuracy')
plt.xlabel('min_samples_leaf')
plt.legend(loc ='upper center')
plt.show()
```



from the graph we can observe that: 1- at the begining the min sample leaf is low. 2- multi level tree predict train better than test. 3- with growing min sample leaf train and test geting similar Finally we can see overfitting of data

4 2.1) one-level decition tree (Glass data)

I'm doing all of the above steps for this glass data too and as result I'm getting the same result

With Max_depth 1

[95]:

```
data = pd.read_csv('glass.csv')
Y = data['class']
X = data.drop(['class'],axis=1)
clf = tree.DecisionTreeClassifier(criterion = 'entropy',
max_depth = 1)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34, random_state=10)

clf = clf.fit(X_train, Y_train)
Yp = clf.predict(X_train)
accTrain = accuracy_score(Y_train, Yp)
print("Accuracy of the train data:",accTrain)

Yp = clf.predict(X_test)
accTest = accuracy_score(Y_test, Yp)
print("Accuracy of the test data:",accTest)
```

```
Accuracy of the train data: 0.46099290780141844
Accuracy of the test data: 0.4246575342465753
```

5 2.2) infinit-level decition tree (diabetes data)

with max_depth = None

[96]:

```
clf = tree.DecisionTreeClassifier(criterion = 'entropy',
max_depth = None)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34, random_state=10)

clf = clf.fit(X_train, Y_train)
Yp = clf.predict(X_train)
accTrain = accuracy_score(Y_train, Yp)
print("Accuracy of the train data:",accTrain)

Yp = clf.predict(X_test)
accTest = accuracy_score(Y_test, Yp)
print("Accuracy of the test data:",accTest)
```

```
Accuracy of the train data: 1.0
Accuracy of the test data: 0.5753424657534246
```

With min_samples_leaf as the size of the datasets

[97]:

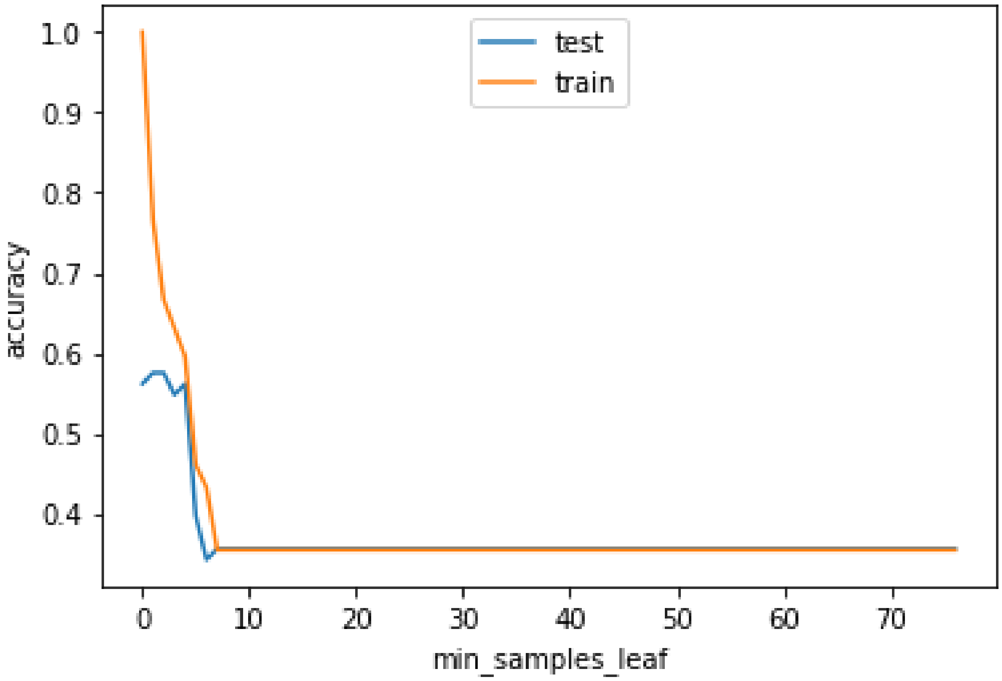
```
clf = tree.DecisionTreeClassifier(criterion = 'entropy',
max_depth = None, min_samples_leaf = 768)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34, random_state=10)
clf = clf.fit(X_train, Y_train)
Yp = clf.predict(X_train)
accTrain = accuracy_score(Y_train, Yp)

print("Accuracy of the train data:",accTrain)
Yp = clf.predict(X_test)
accTest = accuracy_score(Y_test, Yp)
print("Accuracy of the test data:",accTest)
```

```
Accuracy of the train data: 0.3546099290780142
Accuracy of the test data: 0.3561643835616438
```

Result : with 10 step

[98]:

```
test =[]
train = []
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.34,random_state=10)
for x in range(1,768,10):
    clf = tree.DecisionTreeClassifier(criterion = 'entropy',
    max_depth = None, min_samples_leaf = x)
    clf = clf.fit(X_train, Y_train)
    Yp_test = clf.predict(X_test)
    Yp_train = clf.predict(X_train)
    test.append(accuracy_score(Y_test, Yp_test))
    train.append(accuracy_score(Y_train, Yp_train))
plt.plot(test,label = 'test')
plt.plot(train,label = 'train')
plt.ylabel('accuracy')
plt.xlabel('min_samples_leaf')
plt.legend(loc ='upper center')
plt.show()
```


test
train
accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
10
20
30
40
50
60
70
min_samples_leaf