Step 1

```
7 from sklearn.datasets import load_boston
  from sklearn.model_selection import train_test_split
  from sklearn.tree import DecisionTreeClassifier,plot_tree
  import matplotlib.pyplot as plt

  boston = load_boston()
  x_data = boston.data
  y_data = boston.target
  name_data = boston.feature_names
  x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.3)
  y_train=y_train.astype('int')
  y_test=y_test.astype('int')
```

Step 2

```
8 dtc=DecisionTreeClassifier()
  dtc=dtc.fit(x_train,y_train)
  dtc_y_predict=dtc.predict(x_test)
```

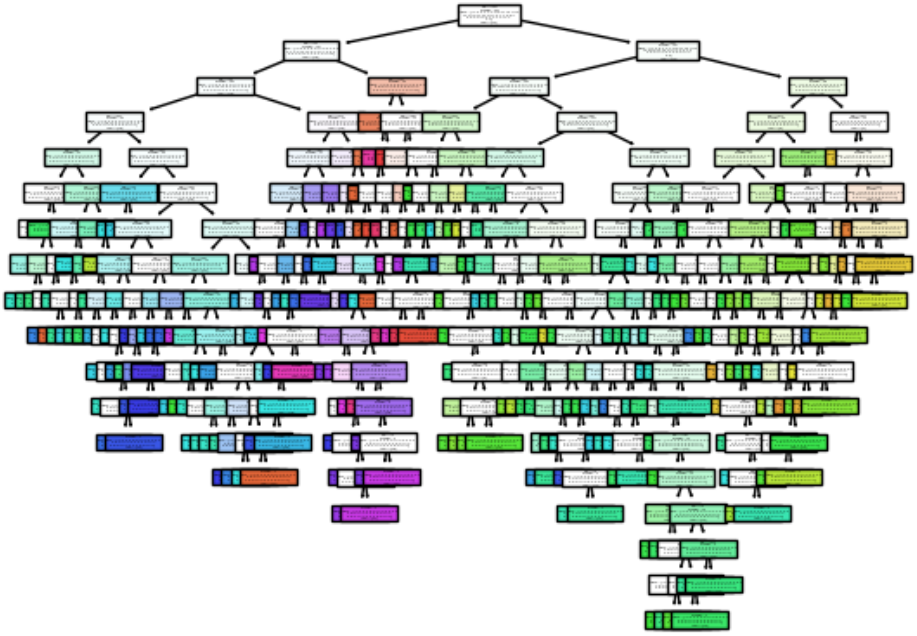
Step 3

```
9 Td=dtc.get_depth()
  print("Tree depth:",Td)
  print("Num of leaves:",dtc.get_n_leaves())
  print("Feature importance:",dtc.feature_importances_)
  print("Train score:",dtc.score(x_train,y_train))
  print("Test score:",dtc.score(x_test,y_test))
```

```
Tree depth: 19
Num of leaves: 214
Feature importance: [0.1538838  0.01980076 0.02475305 0.00887307 0.04623855 0.16602341
 0.08940545 0.10394129 0.02771148 0.03381626 0.06669322 0.09445318
 0.16440649]
Train score: 1.0
Test score: 0.14473684210526316
```

Step 4

```
10 plt.figure()
   plot_tree(dtc,feature_names=y_data,class_names=True)
   plt.show()
```



Step 5&6

17
```
ts_high=0
dtc_high=dtc.fit(x_train,y_train)
for depth in range(1,Td):
    dtc = DecisionTreeClassifier(max_depth=depth)
    dtc = dtc.fit(x_train, y_train)
    dtc_y_predict = dtc.predict(x_test)
    print("Tree depth:", depth)
    print("Num of leaves:", dtc.get_n_leaves())
    print("Feature importance:", dtc.feature_importances_)
    print("Train score:", dtc.score(x_train, y_train))
    print("Test score:", dtc.score(x_test, y_test))
    if ts_high < dtc.score(x_test, y_test):
        ts_high=dtc.score(x_test, y_test)
        dtc_high=dtc
```

Tree depth: 1

Num of leaves: 2

Feature importance: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]

Train score: 0.11864406779661017

Test score: 0.09868421052631579

Tree depth: 2

Num of leaves: 4

Feature importance: [0.16769637 0. 0. 0. 0. 0.38854089
0. 0. 0. 0. 0. 0.

0.44376273]
Train score: 0.1553672316384181
Test score: 0.09868421052631579
Tree depth: 3
Num of leaves: 8
Feature importance: [0.24328957 0. 0. 0. 0. 0.38001023
0. 0. 0. 0. 0. 0.
0.3767002]
Train score: 0.2062146892655367
Test score: 0.14473684210526316
Tree depth: 4
Num of leaves: 16
Feature importance: [0.14452064 0. 0. 0. 0. 0.3202346
0. 0.10070509 0. 0.06000593 0.11528403 0.
0.25924971]
Train score: 0.2824858757062147
Test score: 0.11842105263157894
Tree depth: 5
Num of leaves: 31
Feature importance: [0.12159521 0.03205957 0.02391494 0.03308323 0. 0.24379498
0. 0.10648295 0.02066878 0. 0.09126891 0.0954126
0.23171884]
Train score: 0.3898305084745763
Test score: 0.1118421052631579
Tree depth: 6
Num of leaves: 58
Feature importance: [0.10860821 0.02058967 0.01535893 0.0212471 0. 0.22684156
0.06474341 0.06525252 0. 0.02096332 0.12928016 0.06894651
0.25816861]
Train score: 0.5112994350282486
Test score: 0.09868421052631579
Tree depth: 7
Num of leaves: 91
Feature importance: [0.11616543 0.01504035 0.03428698 0.02112301 0.03516188 0.20278985
0.09563693 0.08698452 0. 0.03188046 0.08289099 0.08528584
0.19275376]
Train score: 0.6271186440677966
Test score: 0.09868421052631579
Tree depth: 8
Num of leaves: 126
Feature importance: [0.13349056 0.01632072 0.05161766 0.01670041 0.02858032 0.20189482
0.10242865 0.08907376 0.00442942 0.02225258 0.08217074 0.07270041
0.17833994]
Train score: 0.7344632768361582

Test score: 0.10526315789473684
Tree depth: 9
Num of leaves: 153
Feature importance: [0.11500299 0.01794159 0.03974347 0.01061009 0.03253891 0.18944888
 0.09298944 0.1012738 0.01174501 0.01732572 0.08151721 0.09503144
 0.19483146]
Train score: 0.8107344632768362
Test score: 0.13157894736842105
Tree depth: 10
Num of leaves: 172
Feature importance: [0.1129328 0.0175051 0.03799804 0.02143501 0.05089916 0.17664736
 0.11637146 0.09046687 0.02153919 0.02876352 0.07544962 0.07660366
 0.17338822]
Train score: 0.8700564971751412
Test score: 0.13157894736842105
Tree depth: 11
Num of leaves: 183
Feature importance: [0.08569351 0.01911764 0.0444596 0.00931677 0.05041637 0.19323632
 0.10866162 0.08765313 0.01762907 0.02137938 0.08234398 0.09622714
 0.18386546]
Train score: 0.9011299435028248
Test score: 0.11842105263157894
Tree depth: 12
Num of leaves: 195
Feature importance: [0.08231419 0.01180507 0.04641366 0.01207971 0.04616896 0.17666851
 0.08389129 0.115424   0.02138696 0.03595977 0.08250505 0.10299329
 0.18238955]
Train score: 0.9350282485875706
Test score: 0.11842105263157894
Tree depth: 13
Num of leaves: 202
Feature importance: [0.10249275 0.01728385 0.04834249 0.00857392 0.0401527 0.1524569
 0.08812657 0.12870902 0.01876782 0.03099325 0.08435746 0.08989582
 0.18984746]
Train score: 0.963276836158192
Test score: 0.1513157894736842
Tree depth: 14
Num of leaves: 206
Feature importance: [0.11759584 0.01113281 0.04913719 0.01441324 0.06465864 0.15445527
 0.07445231 0.110365   0.01913573 0.0359982 0.073449   0.10344438
 0.17176238]
Train score: 0.9830508474576272
Test score: 0.09868421052631579
Tree depth: 15

Num of leaves: 213

Feature importance: [0.09025385 0.01388162 0.05307304 0.01117239 0.03158903 0.1647459
 0.08779213 0.12074364 0.0162978 0.03236982 0.09086101 0.11350243
 0.17371733]

Train score: 0.9971751412429378

Test score: 0.13157894736842105

Step 7

```
18 print("The output of highest test score with depth",dtc_high.get_depth())
   plt.figure()
   plot_tree(dtc_high,feature_names=y_data,class_names=True,filled=True)
   plt.show()
```

```
The output of highest test score with depth 13
```