Decision Tree on Boston Housing-Prices

April 15, 2021

[1]:
```
import numpy as np
import pylab as pl

from sklearn.datasets import load_boston
from sklearn.tree import DecisionTreeClassifier
```

[2]:
```
# Load the data
data = load_boston()
X = data.data
y = data.target
```

[3]:
```
# Split the range of target values into low, mid, and high and reassign the␣
 ↪target values into
# three categorical values 0, 1, and 2, representing low, mid and high range of␣
 ↪values, respectively.
maximum = np.max(y)
minimum = np.min(y)
r = maximum - minimum + 1
low = r / 3
mid = r / 3 * 2

for idx in range(0, int(len(y))):
    if y[idx] < low:
        y[idx] = 0
    elif y[idx] < mid:
        y[idx] = 1
    else:
        y[idx] = 2
```

[4]:
```
# 1. Split the dataset into 70% training set and 30% test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
```

[5]:
```
# 2. Use scikit-learn's DecisionTreeClassifier to train a supervised learning␣
 ↪model
from sklearn import tree
clf0 = tree.DecisionTreeClassifier()
```

```
clf0 = clf0.fit(X_train, y_train)
```

[6]:

```
# 3. Report the tree depth, number of leaves, feature importance, train score,␣
↪and test score of the tree
tree_depth = clf0.get_depth()
print("tree depth: " + str(tree_depth))
print("number of leaves: " + str(clf0.get_n_leaves()))
print("feature importance:\n" + str(clf0.feature_importances_))
print("train score: " + str(clf0.score(X_train, y_train)))
print("test score: " + str(clf0.score(X_test, y_test)))
```

```
tree depth: 10
number of leaves: 42
feature importance:
[0.30266115 0.         0.00742582 0.         0.05929137 0.30703239
 0.04588491 0.02734225 0.00792088 0.00179132 0.05690196 0.03172667
 0.15202128]
train score: 1.0
test score: 0.7828947368421053
```

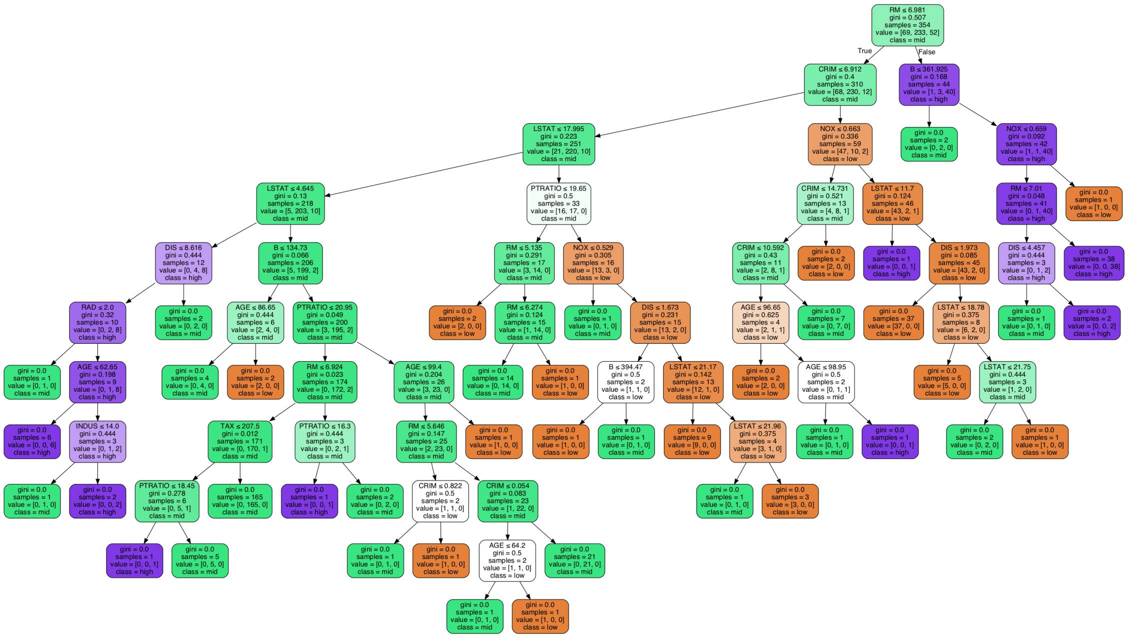
[7]:

```
# 4. Show the visual output of the decision tree
feature_names = data.feature_names
class_names = ['low', 'mid', 'high']
```

[8]:

```
import pydotplus
from IPython.display import Image
dot_data = tree.export_graphviz(clf0, out_file=None,␣
↪feature_names=feature_names, class_names=class_names, filled=True,␣
↪rounded=True,
                                special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())
```

[8]:


RM ≤ 6.981 gini = 0.507 samples = 354 value = [69, 233, 52] class = mid
True
False
CRIM ≤ 6.912 gini = 0.4 samples = 310 value = [68, 230, 12] class = mid
B ≤ 361.925 gini = 0.168 samples = 44 value = [1, 3, 40] class = high
LSTAT ≤ 17.995 gini = 0.223 samples = 251 value = [21, 220, 10] class = mid
NOX ≤ 0.663 gini = 0.336 samples = 59 value = [47, 10, 2] class = low
gini = 0.0 samples = 2 value = [0, 2, 0] class = mid
NOX ≤ 0.659 gini = 0.092 samples = 42 value = [1, 1, 40] class = high
LSTAT ≤ 4.645 gini = 0.13 samples = 218 value = [5, 203, 10] class = mid
PTRATIO ≤ 19.65 gini = 0.5 samples = 33 value = [16, 17, 0] class = mid
CRIM ≤ 14.731 gini = 0.521 samples = 13 value = [4, 8, 1] class = mid
LSTAT ≤ 11.7 gini = 0.124 samples = 46 value = [43, 2, 1] class = low
RM ≤ 7.01 gini = 0.048 samples = 41 value = [0, 1, 40] class = high
gini = 0.0 samples = 1 value = [1, 0, 0] class = low
DIS ≤ 8.616 gini = 0.444 samples = 12 value = [0, 4, 8] class = high
B ≤ 134.73 gini = 0.066 samples = 206 value = [5, 199, 2] class = mid
RM ≤ 5.135 gini = 0.291 samples = 17 value = [3, 14, 0] class = mid
NOX ≤ 0.529 gini = 0.305 samples = 16 value = [13, 3, 0] class = low
CRIM ≤ 10.592 gini = 0.43 samples = 11 value = [2, 8, 1] class = mid
gini = 0.0 samples = 2 value = [2, 0, 0] class = low
gini = 0.0 samples = 1 value = [0, 0, 1] class = high
DIS ≤ 1.973 gini = 0.085 samples = 45 value = [43, 2, 0] class = low
DIS ≤ 4.457 gini = 0.444 samples = 3 value = [0, 1, 2] class = high
gini = 0.0 samples = 38 value = [0, 0, 38] class = high
RAD ≤ 2.0 gini = 0.32 samples = 10 value = [0, 2, 8] class = high
gini = 0.0 samples = 2 value = [0, 2, 0] class = mid
AGE ≤ 86.65 gini = 0.444 samples = 6 value = [2, 4, 0] class = mid
PTRATIO ≤ 20.95 gini = 0.049 samples = 200 value = [3, 195, 2] class = mid
gini = 0.0 samples = 2 value = [2, 0, 0] class = low
RM ≤ 6.274 gini = 0.124 samples = 15 value = [1, 14, 0] class = mid
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
DIS ≤ 1.673 gini = 0.231 samples = 15 value = [13, 2, 0] class = low
AGE ≤ 96.65 gini = 0.625 samples = 4 value = [2, 1, 1] class = low
gini = 0.0 samples = 7 value = [0, 7, 0] class = mid
gini = 0.0 samples = 37 value = [37, 0, 0] class = low
LSTAT ≤ 18.78 gini = 0.375 samples = 8 value = [6, 2, 0] class = low
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 2 value = [0, 0, 2] class = high
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
AGE ≤ 62.65 gini = 0.198 samples = 9 value = [0, 1, 8] class = high
gini = 0.0 samples = 4 value = [0, 4, 0] class = mid
gini = 0.0 samples = 2 value = [2, 0, 0] class = low
RM ≤ 6.924 gini = 0.023 samples = 174 value = [0, 172, 2] class = mid
AGE ≤ 99.4 gini = 0.204 samples = 26 value = [3, 23, 0] class = mid
gini = 0.0 samples = 14 value = [0, 14, 0] class = mid
gini = 0.0 samples = 1 value = [1, 0, 0] class = low
B ≤ 394.47 gini = 0.5 samples = 2 value = [1, 1, 0] class = low
LSTAT ≤ 21.17 gini = 0.142 samples = 13 value = [12, 1, 0] class = low
gini = 0.0 samples = 2 value = [2, 0, 0] class = low
AGE ≤ 98.95 gini = 0.5 samples = 2 value = [0, 1, 1] class = mid
gini = 0.0 samples = 5 value = [5, 0, 0] class = low
LSTAT ≤ 21.75 gini = 0.444 samples = 3 value = [1, 2, 0] class = mid
gini = 0.0 samples = 6 value = [0, 0, 6] class = high
INDUS ≤ 14.0 gini = 0.444 samples = 3 value = [0, 1, 2] class = high
TAX ≤ 207.5 gini = 0.012 samples = 171 value = [0, 170, 1] class = mid
PTRATIO ≤ 16.3 gini = 0.444 samples = 3 value = [0, 2, 1] class = mid
RM ≤ 5.646 gini = 0.147 samples = 25 value = [2, 23, 0] class = mid
gini = 0.0 samples = 1 value = [1, 0, 0] class = low
gini = 0.0 samples = 1 value = [1, 0, 0] class = low
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 9 value = [9, 0, 0] class = low
LSTAT ≤ 21.96 gini = 0.375 samples = 4 value = [3, 1, 0] class = low
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 1 value = [0, 0, 1] class = high
gini = 0.0 samples = 2 value = [0, 2, 0] class = mid
gini = 0.0 samples = 1 value = [1, 0, 0] class = low
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 2 value = [0, 0, 2] class = high
PTRATIO ≤ 18.45 gini = 0.278 samples = 6 value = [0, 5, 1] class = mid
gini = 0.0 samples = 165 value = [0, 165, 0] class = mid
gini = 0.0 samples = 1 value = [0, 0, 1] class = high
gini = 0.0 samples = 2 value = [0, 2, 0] class = mid
CRIM ≤ 0.822 gini = 0.5 samples = 2 value = [1, 1, 0] class = low
CRIM ≤ 0.054 gini = 0.083 samples = 23 value = [1, 22, 0] class = mid
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 3 value = [3, 0, 0] class = low
gini = 0.0 samples = 1 value = [0, 0, 1] class = high
gini = 0.0 samples = 5 value = [0, 5, 0] class = mid
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 1 value = [1, 0, 0] class = low
AGE ≤ 64.2 gini = 0.5 samples = 2 value = [1, 1, 0] class = low
gini = 0.0 samples = 21 value = [0, 21, 0] class = mid
gini = 0.0 samples = 1 value = [0, 1, 0] class = mid
gini = 0.0 samples = 1 value = [1, 0, 0] class = low

[9]:

```
# 5. Generate (Td-1) decision trees on the same training set
# 6. For each of the (Td-1) trees, report tree depth, number of leaves, feature␣
 ↪importance,
# train score, and test score of the tree.
max_test_score = 0
max_clf = None
max_depth = 0

for d in range(1, tree_depth):
    clf1 = tree.DecisionTreeClassifier(max_depth=d)
    clf1 = clf1.fit(X_train, y_train)
    depth = clf1.get_depth()
    print("tree depth: " + str(depth))
    print("number of leaves: " + str(clf1.get_n_leaves()))
    print("feature importance:\n" + str(clf1.feature_importances_))
    print("train score: " + str(clf1.score(X_train, y_train)))
    test_score = clf1.score(X_test, y_test)
    if test_score > max_test_score:
        max_test_score = test_score
        max_clf = clf1
        max_depth = depth
    print("test score: " + str(test_score) + "\n")
```

```
tree depth: 1
number of leaves: 2
feature importance:
[0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
```

```
train score: 0.7627118644067796
test score: 0.6973684210526315

tree depth: 2
number of leaves: 4
feature importance:
[0.48217588 0.         0.         0.         0.         0.48226339
 0.         0.         0.         0.         0.         0.03556074
 0.        ]
train score: 0.8728813559322034
test score: 0.7828947368421053

tree depth: 3
number of leaves: 7
feature importance:
[0.40048526 0.         0.         0.         0.07681512 0.40055794
 0.         0.         0.         0.         0.         0.02953601
 0.09260566]
train score: 0.8870056497175142
test score: 0.7697368421052632

tree depth: 4
number of leaves: 12
feature importance:
[0.36983348 0.         0.         0.         0.05203498 0.34626177
 0.         0.         0.         0.         0.04732912 0.025209
 0.15933165]
train score: 0.9350282485875706
test score: 0.7894736842105263

tree depth: 5
number of leaves: 19
feature importance:
[0.34258769 0.         0.         0.         0.07827385 0.33887571
 0.         0.03155299 0.         0.         0.04357365 0.03068824
 0.13444787]
train score: 0.9548022598870056
test score: 0.8026315789473685

tree depth: 6
number of leaves: 26
feature importance:
[0.3210027  0.         0.         0.         0.08306996 0.32895308
 0.02448959 0.033364   0.         0.         0.04418508 0.0287547
 0.13618088]
train score: 0.9717514124293786
test score: 0.7894736842105263
```

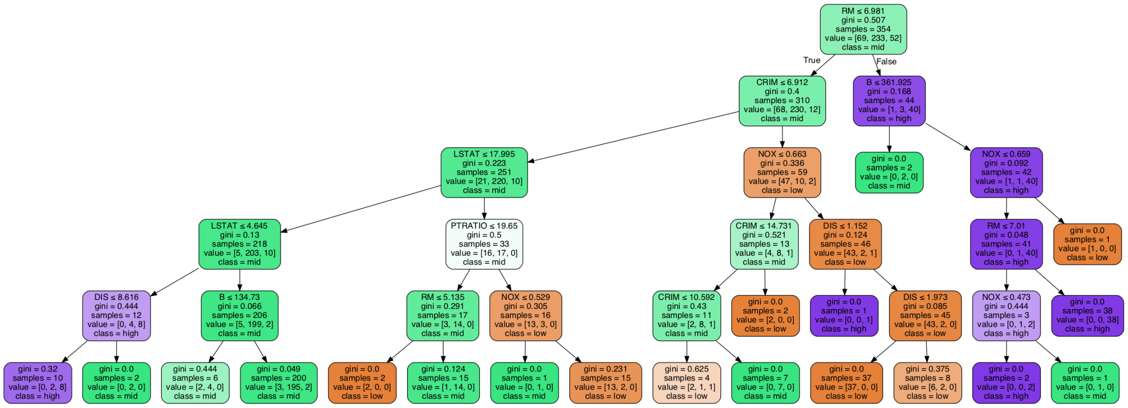
```
tree depth: 7
number of leaves: 33
feature importance:
[0.32113735 0.         0.         0.01103844 0.07110711 0.32816165
 0.02455043 0.02107056 0.00203957 0.         0.04252305 0.04731343
 0.1310584 ]
train score: 0.9830508474576272
test score: 0.7960526315789473

tree depth: 8
number of leaves: 38
feature importance:
[0.31617142 0.         0.00762018 0.         0.06084325 0.31068529
 0.03311555 0.05305265 0.00812819 0.0018382  0.04124589 0.03255708
 0.13474231]
train score: 0.9915254237288136
test score: 0.7697368421052632

tree depth: 9
number of leaves: 41
feature importance:
[0.29875567 0.00746741 0.         0.         0.07051341 0.31136553
 0.04085247 0.06150997 0.00796524 0.00180135 0.04041898 0.0263038
 0.13304617]
train score: 0.9971751412429378
test score: 0.75
```

[10]:

```python
# 7. Show the visual output of the decision tree with highest test score from␣
 ↪the (Td-1)trees.
print("tree depth of the tree with highest test score: " + str(max_depth))
dot_data1 = tree.export_graphviz(max_clf, out_file=None,␣
 ↪feature_names=feature_names, class_names=class_names, filled=True,␣
 ↪rounded=True,
                                special_characters=True)
graph1 = pydotplus.graph_from_dot_data(dot_data1)
Image(graph1.create_png())
```

```
tree depth of the tree with highest test score: 5
```

[10]:


RM ≤ 6.981
gini = 0.507
samples = 354
value = [69, 233, 52]
class = mid
True
False
CRIM ≤ 6.912
gini = 0.4
samples = 310
value = [68, 230, 12]
class = mid
B ≤ 361.925
gini = 0.168
samples = 44
value = [1, 3, 40]
class = high
LSTAT ≤ 17.995
gini = 0.223
samples = 251
value = [21, 220, 10]
class = mid
NOX ≤ 0.663
gini = 0.336
samples = 59
value = [47, 10, 2]
class = low
gini = 0.0
samples = 2
value = [0, 2, 0]
class = mid
NOX ≤ 0.659
gini = 0.092
samples = 42
value = [1, 1, 40]
class = high
LSTAT ≤ 4.645
gini = 0.13
samples = 218
value = [5, 203, 10]
class = mid
PTRATIO ≤ 19.65
gini = 0.5
samples = 33
value = [16, 17, 0]
class = mid
CRIM ≤ 14.731
gini = 0.521
samples = 13
value = [4, 8, 1]
class = mid
DIS ≤ 1.152
gini = 0.124
samples = 46
value = [43, 2, 1]
class = low
RM ≤ 7.01
gini = 0.048
samples = 41
value = [0, 1, 40]
class = high
gini = 0.0
samples = 1
value = [1, 0, 0]
class = low
DIS ≤ 8.616
gini = 0.444
samples = 12
value = [0, 4, 8]
class = high
B ≤ 134.73
gini = 0.066
samples = 206
value = [5, 199, 2]
class = mid
RM ≤ 5.135
gini = 0.291
samples = 17
value = [3, 14, 0]
class = mid
NOX ≤ 0.529
gini = 0.305
samples = 16
value = [13, 3, 0]
class = low
CRIM ≤ 10.592
gini = 0.43
samples = 11
value = [2, 8, 1]
class = mid
gini = 0.0
samples = 2
value = [2, 0, 0]
class = low
gini = 0.0
samples = 1
value = [0, 0, 1]
class = high
DIS ≤ 1.973
gini = 0.085
samples = 45
value = [43, 2, 0]
class = low
NOX ≤ 0.473
gini = 0.444
samples = 3
value = [0, 1, 2]
class = high
gini = 0.0
samples = 38
value = [0, 0, 38]
class = high
gini = 0.32
samples = 10
value = [0, 2, 8]
class = high
gini = 0.0
samples = 2
value = [0, 2, 0]
class = mid
gini = 0.444
samples = 6
value = [2, 4, 0]
class = mid
gini = 0.049
samples = 200
value = [3, 195, 2]
class = mid
gini = 0.0
samples = 2
value = [2, 0, 0]
class = low
gini = 0.124
samples = 15
value = [1, 14, 0]
class = mid
gini = 0.0
samples = 1
value = [0, 1, 0]
class = mid
gini = 0.231
samples = 15
value = [13, 2, 0]
class = low
gini = 0.625
samples = 4
value = [2, 1, 1]
class = low
gini = 0.0
samples = 7
value = [0, 7, 0]
class = mid
gini = 0.0
samples = 37
value = [37, 0, 0]
class = low
gini = 0.375
samples = 8
value = [6, 2, 0]
class = low
gini = 0.0
samples = 2
value = [0, 0, 2]
class = high
gini = 0.0
samples = 1
value = [0, 1, 0]
class = mid