week_5

October 1, 2023

[ ]:
```
# step 0: install scikit learn
#     py -m pip install scikit-learn
#     py -m pip install graphviz

# load libraries
import pandas as pd
from sklearn import tree # package to make decision tree
from sklearn.tree import export_graphviz
import numpy as np
import graphviz

# read in data file
df = pd.read_excel("/Users/avery/OneDrive/Documents/GitHub/
 ↪Clinical_TLB_2023-2024/lung_cancer_tlb.xlsx")
```

[ ]:
```
# SET UP
# keep only control and adenocarcinoma
df['CancerType'] = np.where(df['CancerType'].isna(), 'Control',␣
 ↪df['CancerType'])

df_tree = df[(df['CancerType'] == 'Control') | (df['CancerType'] ==␣
 ↪'Adenocarcinoma')]
```

[ ]:
```
# summary of samples being predicted
df_tree.groupby('CancerType')['pub_id'].nunique()
```

[ ]:
```
CancerType
Adenocarcinoma    72
Control           51
Name: pub_id, dtype: int64
```

[ ]:
```
# create separate dfs for features and labels, both stores as pandas df
# allegedly skl ignores index columns (chatgpt)

features = df_tree.drop(['CancerType', 'pub_id', 'sample_id'], axis=1)
labels = df_tree['CancerType']

# initiate classifier
```

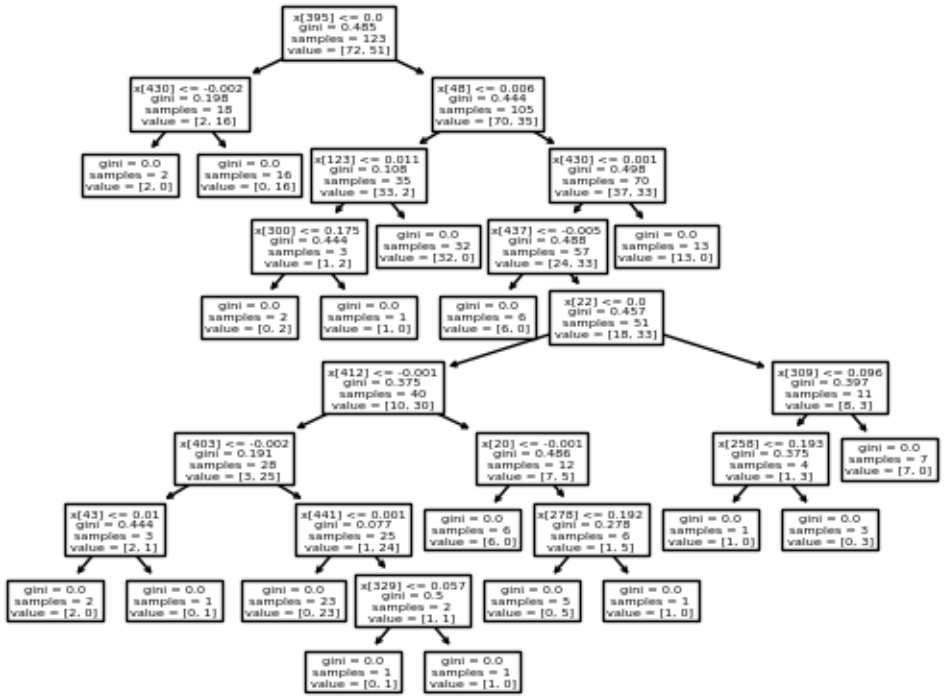
```
clf = tree.DecisionTreeClassifier()

# fit the tree
clf = clf.fit( features, labels )

# view the tree - ugly graphic. exploring other options
tree.plot_tree(clf)
```

[ ]:
```
[Text(0.359375, 0.95, 'x[395] <= 0.0\ngini = 0.485\nsamples = 123\nvalue = [72,
51]'),
 Text(0.203125, 0.85, 'x[430] <= -0.002\ngini = 0.198\nsamples = 18\nvalue = [2,
16]'),
 Text(0.140625, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.265625, 0.75, 'gini = 0.0\nsamples = 16\nvalue = [0, 16]'),
 Text(0.515625, 0.85, 'x[48] <= 0.006\ngini = 0.444\nsamples = 105\nvalue = [70,
35]'),
 Text(0.390625, 0.75, 'x[123] <= 0.011\ngini = 0.108\nsamples = 35\nvalue = [33,
2]'),
 Text(0.328125, 0.65, 'x[300] <= 0.175\ngini = 0.444\nsamples = 3\nvalue = [1,
2]'),
 Text(0.265625, 0.55, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.390625, 0.55, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.453125, 0.65, 'gini = 0.0\nsamples = 32\nvalue = [32, 0]'),
 Text(0.640625, 0.75, 'x[430] <= 0.001\ngini = 0.498\nsamples = 70\nvalue = [37,
33]'),
 Text(0.578125, 0.65, 'x[437] <= -0.005\ngini = 0.488\nsamples = 57\nvalue =
[24, 33]'),
 Text(0.515625, 0.55, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.640625, 0.55, 'x[22] <= 0.0\ngini = 0.457\nsamples = 51\nvalue = [18,
33]'),
 Text(0.40625, 0.45, 'x[412] <= -0.001\ngini = 0.375\nsamples = 40\nvalue = [10,
30]'),
 Text(0.25, 0.35, 'x[403] <= -0.002\ngini = 0.191\nsamples = 28\nvalue = [3,
25]'),
 Text(0.125, 0.25, 'x[43] <= 0.01\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.0625, 0.15, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.1875, 0.15, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.375, 0.25, 'x[441] <= 0.001\ngini = 0.077\nsamples = 25\nvalue = [1,
24]'),
 Text(0.3125, 0.15, 'gini = 0.0\nsamples = 23\nvalue = [0, 23]'),
 Text(0.4375, 0.15, 'x[329] <= 0.057\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.375, 0.05, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5, 0.05, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5625, 0.35, 'x[20] <= -0.001\ngini = 0.486\nsamples = 12\nvalue = [7,
5]'),
 Text(0.5, 0.25, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.625, 0.25, 'x[278] <= 0.192\ngini = 0.278\nsamples = 6\nvalue = [1,
```

```
5]'),
 Text(0.5625, 0.15, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.6875, 0.15, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.875, 0.45, 'x[309] <= 0.096\ngini = 0.397\nsamples = 11\nvalue = [8,
3]'),
 Text(0.8125, 0.35, 'x[258] <= 0.193\ngini = 0.375\nsamples = 4\nvalue = [1,
3]'),
 Text(0.75, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.875, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9375, 0.35, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.703125, 0.65, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]')]
```



[ ]:
```
dot_data = export_graphviz(clf, out_file=None,
                           feature_names=features.columns,  # Replace with your
 ↪feature names
                           class_names=labels.unique(),   # Replace with your
 ↪class names
                           filled=True, rounded=True, special_characters=True)

graph = graphviz.Source(dot_data)

# Render and display the decision tree
```

```
graph.render("decision_tree.pdf")  # Saves the tree as "decision_tree.pdf" or another format
graph.view("decision_tree.pdf")
```

[ ]: 'decision_tree.pdf.pdf'

To make a better image: go this this website https://graphviz.gitlab.io/download/

Download the windows installer. Then run the code above.