Decision Trees

In [58]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import sklearn.tree as tree
```

About the dataset

Imagine that you are a medical researcher compiling data for a study. You have collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of 5 medications, Drug A, Drug B, Drug c, Drug x and y.

Part of your job is to build a model to find out which drug might be appropriate for a future patient with the same illness. The features of this dataset are Age, Sex, Blood Pressure, and the Cholesterol of the patients, and the target is the drug that each patient responded to.

It is a sample of multiclass classifier, and you can use the training part of the dataset to build a decision tree, and then use it to predict the class of an unknown patient, or to prescribe a drug to a new patient.

Inserting Dataset

In [8]:
```
import os
path=os.path.abspath("/Users/Ben Ashael/.ipynb_checkpoints/drug200.csv")
my_data=pd.read_csv(path)
my_data[0:5]
```

Out[8]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

In [9]:
```
my_data.shape
```

Out[9]: (200, 6)

Pre-processing

In [32]:
```
X = my_data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values
X[0:5]
```

Out[32]:
```
array([[23, 'F', 'HIGH', 'HIGH', 25.355],
       [47, 'M', 'LOW', 'HIGH', 13.093],
       [47, 'M', 'LOW', 'HIGH', 10.113999999999999],
       [28, 'F', 'NORMAL', 'HIGH', 7.797999999999999],
       [61, 'F', 'LOW', 'HIGH', 18.043]], dtype=object)
```

In [33]:
```
from sklearn import preprocessing
le_sex = preprocessing.LabelEncoder()
le_sex.fit(['F','M'])
X[:,1] = le_sex.transform(X[:,1])


le_BP = preprocessing.LabelEncoder()
le_BP.fit([ 'LOW', 'NORMAL', 'HIGH'])
X[:,2] = le_BP.transform(X[:,2])


le_Chol = preprocessing.LabelEncoder()
le_Chol.fit([ 'NORMAL', 'HIGH'])
X[:,3] = le_Chol.transform(X[:,3])

X[0:5]
```

Out[33]:
```
array([[23, 0, 0, 0, 25.355],
       [47, 1, 1, 0, 13.093],
       [47, 1, 1, 0, 10.113999999999999],
       [28, 0, 2, 0, 7.797999999999999],
       [61, 0, 1, 0, 18.043]], dtype=object)
```

In [34]:
```
y = my_data["Drug"]
y[0:5]
```

Out[34]:
```
0    drugY
1    drugC
2    drugC
3    drugX
4    drugY
Name: Drug, dtype: object
```

Setting up the Decision Tree

In [35]:
```
from sklearn.model_selection import train_test_split
```

Now train_test_split will return 4 different parameters. We will name them: X_trainset, X_testset, y_trainset, y_testset

In [36]:
```
X_trainset, X_testset, y_trainset, y_testset = train_test_split(X, y, test_
size=0.3, random_state=3)
```

In [37]:
```
print('Shape of X training set {}'.format(X_trainset.shape),'&',' Size of Y
training set {}'.format(y_trainset.shape))
```

```
Shape of X training set (140, 5) &  Size of Y training set (140,)
```

In [38]:
```
print('Shape of X training set {}'.format(X_testset.shape),'&',' Size of Y
training set {}'.format(y_testset.shape))
```

```
Shape of X training set (60, 5) &  Size of Y training set (60,)
```

Modelling

We will first create an instance of the DecisionTreeClassifier called drugTree. Inside of the classifier, specify criterion="entropy" so we can see the information gain of each node.

In [39]:
```
drugTree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
drugTree # it shows the default parameters
```

Out[39]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=4,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [41]:
```
drugTree.fit(X_trainset,y_trainset)
```

```
C:\Users\Ben Ashael\Anaconda3\lib\site-packages\sklearn\tree\tree.py:149: D
eprecationWarning: `np.int` is a deprecated alias for the builtin `int`. To
silence this warning, use `int` by itself. Doing this will not modify any b
ehavior and is safe. When replacing `np.int`, you may wish to use e.g. `np.
int64` or `np.int32` to specify the precision. If you wish to review your c
urrent use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/
devdocs/release/1.20.0-notes.html#deprecations
  y_encoded = np.zeros(y.shape, dtype=np.int)
```

Out[41]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=4,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

Prediction

Let's make some predictions on the testing dataset and store it into a variable called predTree.

In [42]:
```
predTree = drugTree.predict(X_testset)
```

In [43]:
```
print (predTree [0:5])
print (y_testset [0:5])
```

```
['drugY' 'drugX' 'drugX' 'drugX' 'drugX']
40     drugY
51     drugX
139    drugX
197    drugX
170    drugX
Name: Drug, dtype: object
```

Evaluation

Next, let's import metrics from sklearn and check the accuracy of our model.

In [44]:
```
from sklearn import metrics
import matplotlib.pyplot as plt
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_testset, predT
ree))
```

```
DecisionTrees's Accuracy:  0.9833333333333333
```

Visualization

In [46]:
```
!pip install pydotplus
!pip install python-graphviz
```

```
Collecting pydotplus
  Downloading https://files.pythonhosted.org/packages/60/bf/62567830b700d9f
6930e9ab6831d6ba256f7b0b730acb37278b0ccdffacf/pydotplus-2.0.2.tar.gz (278k
B)
Requirement already satisfied: pyparsing>=2.0.1 in c:\users\ben ashael\anac
onda3\lib\site-packages (from pydotplus) (2.3.1)
Building wheels for collected packages: pydotplus
  Building wheel for pydotplus (setup.py): started
  Building wheel for pydotplus (setup.py): finished with status 'done'
  Stored in directory: C:\Users\Ben Ashael\AppData\Local\pip\Cache\wheels\3
5\7b\ab\66fb7b2ac1f6df87475b09dc48e707b6e0de80a6d8444e3628
Successfully built pydotplus
Installing collected packages: pydotplus
Successfully installed pydotplus-2.0.2
Collecting python-graphviz

  Could not find a version that satisfies the requirement python-graphviz
(from versions: )
No matching distribution found for python-graphviz
```

In [61]:
```
plt.show()
```