Decision Tree

Will use the classification algorithm to build a model from historical data of patients, and their response to different medications. Then use the trained decision tree to predict the class of a unknown patient, or to find a proper drug for a new patient.

Import the Libraries

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
```

About the dataset

Research data about a set of patients, all of whom suffered from the same illness. Each patient respond to one of 5 medications, Drug A, Drug B, Drug c, Drug x and Drug y.

Build a model to find out which drug might be appropriate for a future patient with the same illness. The feature sets of this dataset are Age, Sex, Blood Pressure, and Cholesterol of patients, and the target is the drug that each patient responded to.
It is a sample of binary classifier, and using the training part of the dataset to build a decision tree, and then use it to predict the class of a unknown patient, or to prescribe it to a new patient.

In [2]:

```
df = pd.read_csv("drug200.csv", delimiter=",")
df[0:5]
```

Out[2]:

	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 [3]:

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

Out[3]:

```
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 [4]:

```
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[4]:

```
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 [5]:

```
y = df["Drug"]
y[0:5]
```

Out[5]:

```
0    drugY
1    drugC
2    drugC
3    drugX
4    drugY
Name: Drug, dtype: object
```

Setting up the Decision Tree

In [6]:

```
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, random_state=4
)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (140, 5) (140,)
Test set: (60, 5) (60,)
```

Modeling

In [7]:

```
drugTree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
drugTree
```

Out[7]:

```
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=Non
e,
            splitter='best')
```

In [8]:

```
drugTree.fit(X_train,y_train)
```

Out[8]:

```
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=Non
e,
            splitter='best')
```

Prediction

In [11]:

```
predTree = drugTree.predict(X_test)
print (predTree [0:5])
print (y_test [0:5])
```

```
['drugY' 'drugY' 'drugY' 'drugY' 'drugC']
11     drugY
99     drugY
128    drugY
175    drugY
1      drugC
Name: Drug, dtype: object
```

Evaluation

In [13]:

```
from sklearn import metrics
import matplotlib.pyplot as plt
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_test, predTree))
```

```
DecisionTrees's Accuracy:  0.9666666666666667
```

In [ ]: