In [1]:
```
# importing all necessary packages
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
```

In [3]:
```
# reading the data from csv file
drug_data = pd.read_csv("c:\\Users\\Lenovo\\Desktop\\drug_Dataset.csv")
drug_data[0:10]
```

Out[3]:

	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
5	22	F	NORMAL	HIGH	8.607	drugX
6	49	F	NORMAL	HIGH	16.275	drugY
7	41	M	LOW	HIGH	11.037	drugC
8	60	M	NORMAL	HIGH	15.171	drugY
9	43	M	LOW	NORMAL	19.368	drugY

In [4]:
```
#let's check the shape of our data.
drug_data.shape
```

Out[4]:
```
(200, 6)
```

In [5]:
```
# Now we'll separate the coloumn of data in Feature data and target data.
X = drug_data[['Age','Sex','BP','Cholesterol','Na_to_K']].values
X[0:6]
```

Out[5]:
```
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],
       [22, 'F', 'NORMAL', 'HIGH', 8.607000000000001]], dtype=object)
```

Since Sklearn Decision Tree doesn't support Categorical values, we need to convert these values into numerical values.

In [6]:
```
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]
# X [:,1]  is slicing the array by keeping all the rows and just taking colomn
1
```

Out[6]:
```
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 [7]:
```
Y = drug_data['Drug']
Y[0:5]
```

Out[7]:
```
0    drugY
1    drugC
2    drugC
3    drugX
4    drugY
Name: Drug, dtype: object
```

Before setting up the Decision Tree, we need to split the data into Training and Testing dataset. For that we'll use **Train/test Split** from sklearn.model_selection library

In [8]:
```
from sklearn.model_selection import train_test_split
# train_test_split function needs parameter as X,Y,random_state and test_size

X_trainset,X_testset,Y_trainset,Y_testset = train_test_split(X,Y,test_size=0.3
0,random_state = 31)

# we'll print and check the new variables
print(X_trainset.shape,X_testset.shape,Y_trainset.shape,Y_testset.shape)
```

```
(140, 5) (60, 5) (140,) (60,)
```

Now we'll create an instance of **DecisionTreeClassifier** as **DrugTree**.

Inside of the classifier, specify criterion="entropy" so we can see the information gain of each node.

In [9]:
```
drugTree = DecisionTreeClassifier(criterion = "entropy", max_depth = 4)
drugTree
```

Out[9]:
```
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')
```

Next, we will fit the data with the training feature matrix X_trainset and training response vector y_trainset

In [10]:
```
drugTree.fit(X_trainset,Y_trainset)
```

Out[10]:
```
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

In [11]:
```
predTree = drugTree.predict(X_testset)
# Now we can verify some of the model's prediction with the Actual test data.
print(predTree[0:5] )
print(X_testset[0:5])
print(Y_testset[0:5])
```

```
['drugX' 'drugY' 'drugX' 'drugA' 'drugA']
[[61 0 1 1 7.34]
 [60 1 2 0 15.171]
 [32 0 1 1 10.84]
 [35 0 0 0 12.894]
 [26 0 0 1 12.307]]
148    drugX
8      drugY
79     drugX
137    drugA
147    drugA
Name: Drug, dtype: object
```

Evaluation

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

In [12]:
```
from sklearn import metrics
from matplotlib import pyplot as plt
print("Decision tree's Accuracy:",(metrics.accuracy_score(Y_testset,predTree))
*100,"%")
```

```
Decision tree's Accuracy: 100.0 %
```

Visualization

In [24]:
```
# Let's visualize the tree
from sklearn.externals.six import StringIO
import pydotplus
import matplotlib.image as mpimg
from sklearn import tree
import graphviz
%matplotlib inline
```

In [ ]:
```
dot_data = StringIO()
filename = "drugtree.png"
featureNames = drug_data.columns[0:5]
targetNames = drug_data["Drug"].unique().tolist()
out=tree.export_graphviz(drugTree,feature_names=featureNames, out_file=dot_dat
a, class_names= np.unique(Y_trainset), filled=True,  special_characters=True,r
otate=False)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png(filename)
img = mpimg.imread(filename)
plt.figure(figsize=(100, 200))
plt.imshow(img,interpolation='nearest')
```

In [ ]:
```
# there is problem with the vsualization code and I'm unable to fix it. I woul
d be thankful if anyone can fix it and share it with us.
# Graphviz module is intalled in my system still problem arises.
# I have attached simillar decision tree with my code in the mail for visualiz
ation.
```