```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
import numpy as np
```

```
#Splitting the dataset into train and test
from sklearn.model_selection import train_test_split
```

```
#import the decisiontree classifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import classification_report
```

```
# for pre-processing
from sklearn import preprocessing
```

```
#import the datasets

drug200 = pd.read_csv("drug200.csv", index_col=0)
```

```
drug200.head()
```

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

```
drug200['Drug'].value_counts()
```

```
    drugY    91
    drugX    54
    drugA    23
    drugC    16
    drugB    16
    Name: Drug, dtype: int64
```

```
# use label encoder to convert Drug into numbers

label_encoder = preprocessing.LabelEncoder()
drug200['Drug'] = label_encoder.fit_transform(drug200['Drug'])
```

```
drug200['Drug'].value_counts()
```

```
    4    91
    3    54
    0    23
    2    16
    1    16
    Name: Drug, dtype: int64
```

```
drug200['Sex'].value_counts()
```

```
    M    104
    F     96
    Name: Sex, dtype: int64
```

```
# use label encoder to convert Sex into numbers

label_encoder = preprocessing.LabelEncoder()
drug200['Sex'] = label_encoder.fit_transform(drug200['Sex'])
```

```
drug200['BP'].value_counts()
```

```
HIGH      77
LOW       64
NORMAL    59
Name: BP, dtype: int64
```

```
# use label encoder to convert BP into numbers

label_encoder = preprocessing.LabelEncoder()
drug200['BP'] = label_encoder.fit_transform(drug200['BP'])
```

```
drug200['Cholesterol'].value_counts()
```

```
HIGH      103
NORMAL     97
Name: Cholesterol, dtype: int64
```

```
# use label encoder to convert Cholesterol into numbers

label_encoder = preprocessing.LabelEncoder()
drug200['Cholesterol'] = label_encoder.fit_transform(drug200['Cholesterol'])
```

```
drug200.head()
```

	Sex	BP	Cholesterol	Na_to_K	Drug
Age					
23	0	0	0	25.355	4
47	1	1	0	13.093	2
47	1	1	0	10.114	2
28	0	2	0	7.798	3
61	0	1	0	18.043	4

```
drug200.iloc[:,0:4]
```

	Sex	BP	Cholesterol	Na_to_K
Age				
23	0	0	0	25.355
47	1	1	0	13.093
47	1	1	0	10.114
28	0	2	0	7.798
61	0	1	0	18.043
...	...	...	...	...
56	0	1	0	11.567
16	1	1	0	12.006
52	1	2	0	9.894
23	1	2	1	14.020
40	0	1	1	11.349

200 rows × 4 columns

```
x = drug200.iloc[:,0:4]
y = drug200.iloc[:,4]
```

```
x
```

	Sex	BP	Cholesterol	Na_to_K
Age				
23	0	0	0	25.355
47	1	1	0	13.093
47	1	1	0	10.114
28	0	2	0	7.798
61	0	1	0	18.043
...	...	...	...	...
56	0	1	0	11.567
16	1	1	0	12.006
52	1	2	0	9.894
23	1	2	1	14.020

```
y
```

```
Age
23    4
47    2
47    2
28    3
61    4
     ..
56    2
16    2
52    3
23    3
40    3
Name: Drug, Length: 200, dtype: int64
```

```
print(drug200.shape)
```

```
(200, 5)
```

```
y = drug200.iloc[:, -1]
```

```
y
```

```
Age
23    4
47    2
47    2
28    3
61    4
     ..
56    2
16    2
52    3
23    3
40    3
Name: Drug, Length: 200, dtype: int64
```

```
# Splitting the dataset into train and test

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2, random_state=44)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((160, 4), (40, 4), (160,), (40,))
```

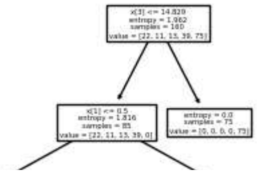
```
model =  DecisionTreeClassifier(criterion='entropy',max_depth = 4)

model.fit(x_train,y_train)
```

```
▾                   DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=4)
```

Plot the decsion Tree

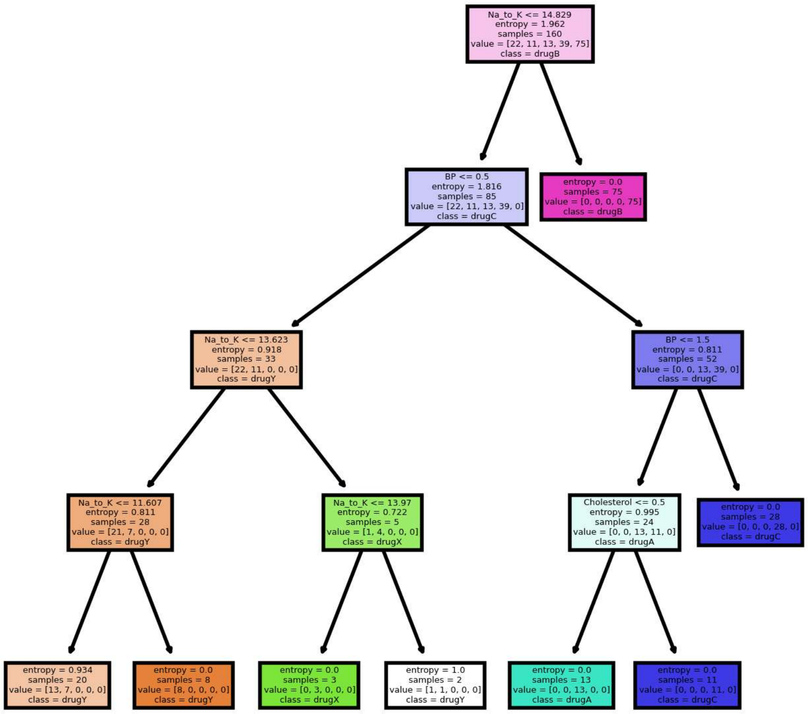
```
tree.plot_tree(model)
```

```
[Text(0.6538461538461539, 0.9, 'x[3] <= 14.829\nentropy = 1.962\nsamples = 160\nvalue
= [22, 11, 13, 39, 75]'),
 Text(0.5769230769230769, 0.7, 'x[1] <= 0.5\nentropy = 1.816\nsamples = 85\nvalue =
[22, 11, 13, 39, 0]'),
 Text(0.3076923076923077, 0.5, 'x[3] <= 13.623\nentropy = 0.918\nsamples = 33\nvalue =
[22, 11, 0, 0, 0]'),
 Text(0.15384615384615385, 0.3, 'x[3] <= 11.607\nentropy = 0.811\nsamples = 28\nvalue
= [21, 7, 0, 0, 0]'),
 Text(0.07692307692307693, 0.1, 'entropy = 0.934\nsamples = 20\nvalue = [13, 7, 0, 0,
0]'),
 Text(0.23076923076923078, 0.1, 'entropy = 0.0\nsamples = 8\nvalue = [8, 0, 0, 0,
0]'),
 Text(0.46153846153846156, 0.3, 'x[3] <= 13.97\nentropy = 0.722\nsamples = 5\nvalue =
[1, 4, 0, 0, 0]'),
 Text(0.38461538461538464, 0.1, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3, 0, 0,
0]'),
 Text(0.5384615384615384, 0.1, 'entropy = 1.0\nsamples = 2\nvalue = [1, 1, 0, 0, 0]'),
 Text(0.8461538461538461, 0.5, 'x[1] <= 1.5\nentropy = 0.811\nsamples = 52\nvalue =
[0, 0, 13, 39, 0]'),
 Text(0.7692307692307693, 0.3, 'x[2] <= 0.5\nentropy = 0.995\nsamples = 24\nvalue =
[0, 0, 13, 11, 0]'),
 Text(0.6923076923076923, 0.1, 'entropy = 0.0\nsamples = 13\nvalue = [0, 0, 13, 0,
0]'),
 Text(0.8461538461538461, 0.1, 'entropy = 0.0\nsamples = 11\nvalue = [0, 0, 0, 11,
0]'),
 Text(0.9230769230769231, 0.3, 'entropy = 0.0\nsamples = 28\nvalue = [0, 0, 0, 28,
0]'),
 Text(0.7307692307692307, 0.7, 'entropy = 0.0\nsamples = 75\nvalue = [0, 0, 0, 0,
75]')]
```



```
fn = ['Sex','BP','Cholesterol','Na_to_K']
cn = ['drugY','drugX','drugA','drugC','drugB']
fig,axes = plt.subplots(nrows=1, ncols=1, figsize=(4,4), dpi=400)
tree.plot_tree(model,feature_names=fn,class_names = cn, filled=True)
```

```
 Text(0.23076923076923078, 0.1, 'entropy = 0.0\nsamples = 8\nvalue = [8, 0, 0, 0,
0]\nclass = drugY'),
 Text(0.46153846153846156, 0.3, 'Na_to_K <= 13.97\nentropy = 0.722\nsamples =
5\nvalue = [1, 4, 0, 0, 0]\nclass = drugX'),
 Text(0.38461538461538464, 0.1, 'entropy = 0.0\nsamples = 3\nvalue = [0, 3, 0, 0,
0]\nclass = drugX'),
 Text(0.5384615384615384, 0.1, 'entropy = 1.0\nsamples = 2\nvalue = [1, 1, 0, 0,
0]\nclass = drugY'),
 Text(0.8461538461538461, 0.5, 'BP <= 1.5\nentropy = 0.811\nsamples = 52\nvalue =
[0, 0, 13, 39, 0]\nclass = drugC'),
 Text(0.7692307692307693, 0.3, 'Cholesterol <= 0.5\nentropy = 0.995\nsamples =
24\nvalue = [0, 0, 13, 11, 0]\nclass = drugA'),
 Text(0.6923076923076923, 0.1, 'entropy = 0.0\nsamples = 13\nvalue = [0, 0, 13, 0,
0]\nclass = drugA'),
 Text(0.8461538461538461, 0.1, 'entropy = 0.0\nsamples = 11\nvalue = [0, 0, 0, 11,
0]\nclass = drugC'),
 Text(0.9230769230769231, 0.3, 'entropy = 0.0\nsamples = 28\nvalue = [0, 0, 0, 28,
0]\nclass = drugC'),
 Text(0.7307692307692307, 0.7, 'entropy = 0.0\nsamples = 75\nvalue = [0, 0, 0, 0,
75]\nclass = drugB')]
```



```
preds = model.predict(x_test)
```

```
pd.Series(preds).value_counts()
```

```
4    16
3    15
0     6
2     3
dtype: int64
```

```
y_test
```

```
Age
74    3
41    4
53    4
39    4
48    4
32    3
68    1
28    3
20    3
```

```
53    1
22    4
47    3
20    3
60    1
43    4
54    4
46    3
66    3
50    4
68    2
25    4
66    4
60    1
41    2
43    4
49    3
59    3
55    1
37    3
39    4
23    4
58    4
35    3
64    3
36    3
23    3
59    2
47    0
15    4
68    4
Name: Drug, dtype: int64
```

```
pd.crosstab(y_test,preds)
```

col_0	0	2	3	4
Drug				
0	1	0	0	0
1	5	0	0	0
2	0	3	0	0
3	0	0	15	0
4	0	0	0	16

Accuracy of the above model

```
np.mean(preds==y_test)
```

```
0.875
```