Decision trees

An approach used in supervised machine learning, a technique which uses labelled input and output datasets to train models. The approach is used mainly to solve classification problems, which is the use of a model to categorise or classify an object. Decision trees in machine learning are also used in regression problems, an approach used in predictive analytics to forecast outputs from unseen data.

About the Data Set - Drug200.csv

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.

Objective

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.

In [1]:
```
import numpy as np
import os
import pandas as pd
```

In [2]:
```
os.chdir(r'C:\Users\HP\Downloads\Machine Learning with Python')
my_data = pd.read_csv("drug200.csv", delimiter=",")
my_data.head()
```

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]:
```
my_data.duplicated().sum()
```

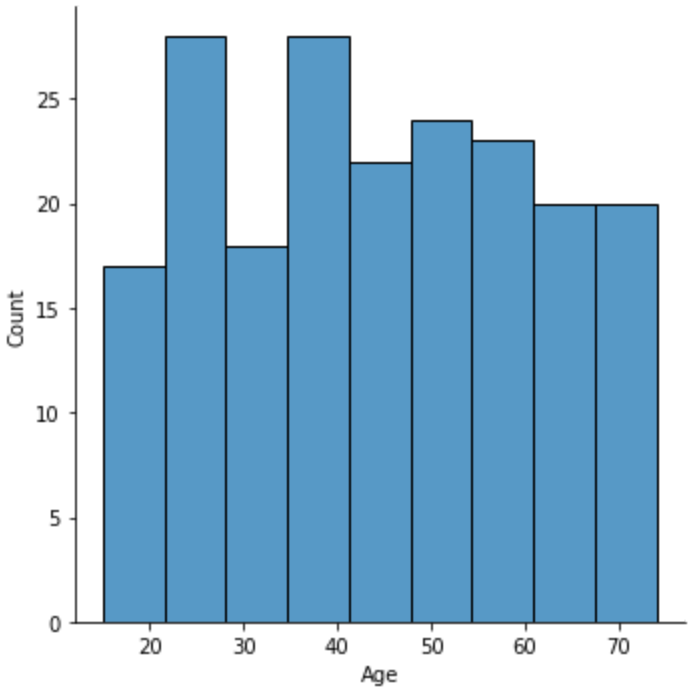
Out[3]:
```
0
```

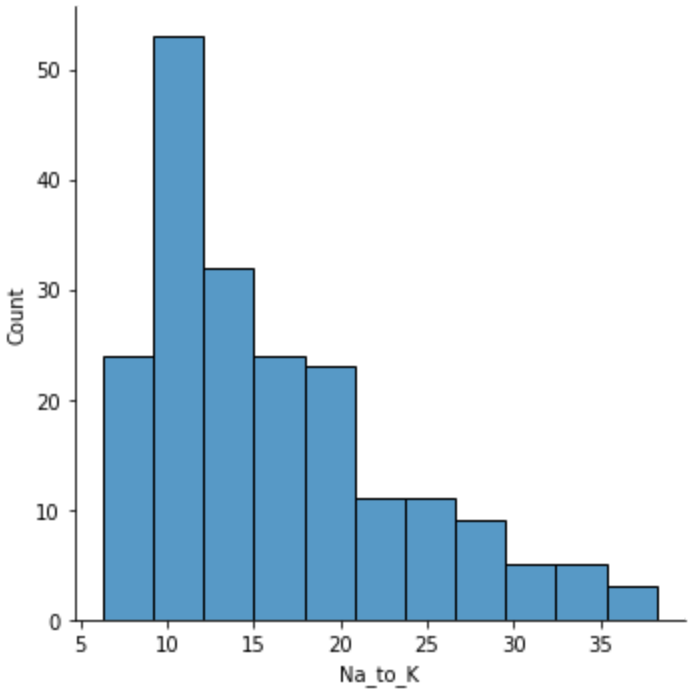
In [4]:
```
my_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          200 non-null    int64
 1   Sex          200 non-null    object
 2   BP           200 non-null    object
```

```
 3   Cholesterol  200 non-null    object
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [5]:
```
import seaborn as sns
for features in my_data.select_dtypes('number'):
    sns.displot(data= my_data, x= features)
```





In [6]:
```
print('Data Columns',my_data.columns,'\n',"Data Shape",my_data.shape)
```

```
Data Columns Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='objec
t')
 Data Shape (200, 6)
```

Pre-processing

X as the Feature Matrix (data of my_data)

y as the response vector (target)

In [7]:
```
X = my_data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].to_numpy()
X[0:5]
```

Out[7]:
```
array([[23, 'F', 'HIGH', 'HIGH', 25.355],
       [47, 'M', 'LOW', 'HIGH', 13.093],
       [47, 'M', 'LOW', 'HIGH', 10.114],
       [28, 'F', 'NORMAL', 'HIGH', 7.798],
       [61, 'F', 'LOW', 'HIGH', 18.043]], dtype=object)
```

Some features in this dataset are categorical, such as Sex or BP. Sklearn Decision Trees does not handle categorical variables. We can still convert these features to numerical values.

In [8]:
```
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[8]:
```
array([[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]], dtype=object)
```

In [9]:
```
y = my_data["Drug"]
y[0:5]
```

Out[9]:
```
0    drugY
1    drugC
2    drugC
3    drugX
4    drugY
Name: Drug, dtype: object
```

Setting up the Decision Tree

In [10]:
```
from sklearn.model_selection import train_test_split
X_trainset, X_testset, y_trainset, y_testset = train_test_split(X, y, test_size=0.3, ran
```

In [11]:
```
print('Shape of X training set {}'.format(X_trainset.shape),'&',' Size of Y training set
```

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

In [12]:
```
print('Shape of X test set {}'.format(X_testset.shape),'&',' Size of Y test set {}'.form
```

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

Modeling

In [13]:
```
from sklearn.tree import DecisionTreeClassifier
drugTree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
drugTree.fit(X_trainset,y_trainset)
predTree = drugTree.predict(X_testset)
```

In [14]:
```
print (predTree [0:5])
print("**********************")
print (y_testset [0:5])
```

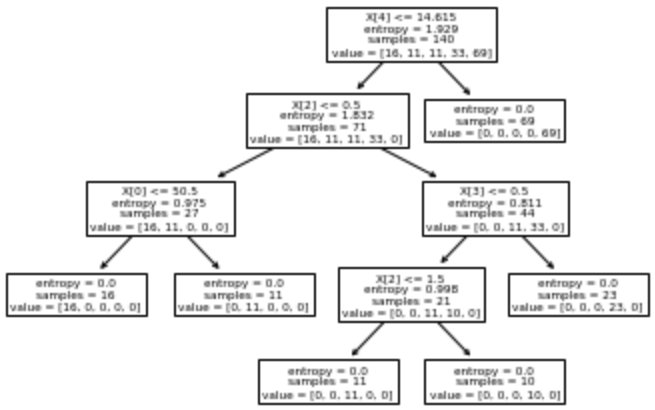
```
['drugY' 'drugX' 'drugX' 'drugX' 'drugX']
**********************
40     drugY
51     drugX
139    drugX
197    drugX
170    drugX
Name: Drug, dtype: object
```

Evaluation

In [15]:
```
from sklearn import metrics
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_testset, predTree))
```

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

In [16]:
```
import matplotlib.pyplot as plt
import sklearn.tree as tree
tree.plot_tree(drugTree)
plt.show()
```



In [ ]:
```
```