

In [13]:

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [14]:

```
df=pd.read_csv(r"C:\Users\prajapath Arjun\Downloads\drug200.csv")
df
```

Out[14]:

	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
...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [15]:

```
df.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 [16]:

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

Out[16]:

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

In [17]:

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

Out[17]:

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

In [18]:

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

Out[18]:

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

In [19]:

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

Out[19]:

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

In [20]:

```
convert={"Sex":{"M":1, "F":0}}
df=df.replace(convert)
df
```

Out[20]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	HIGH	HIGH	25.355	drugY
1	47	1	LOW	HIGH	13.093	drugC
2	47	1	LOW	HIGH	10.114	drugC
3	28	0	NORMAL	HIGH	7.798	drugX
4	61	0	LOW	HIGH	18.043	drugY
...
195	56	0	LOW	HIGH	11.567	drugC
196	16	1	LOW	HIGH	12.006	drugC
197	52	1	NORMAL	HIGH	9.894	drugX
198	23	1	NORMAL	NORMAL	14.020	drugX
199	40	0	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [21]:

```
convert={"BP":{"HIGH":1, "NORMAL":2, "LOW":3}}
df=df.replace(convert)
df
```

Out[21]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	1	HIGH	25.355	drugY
1	47	1	3	HIGH	13.093	drugC
2	47	1	3	HIGH	10.114	drugC
3	28	0	2	HIGH	7.798	drugX
4	61	0	3	HIGH	18.043	drugY
...
195	56	0	3	HIGH	11.567	drugC
196	16	1	3	HIGH	12.006	drugC
197	52	1	2	HIGH	9.894	drugX
198	23	1	2	NORMAL	14.020	drugX
199	40	0	3	NORMAL	11.349	drugX

200 rows × 6 columns

In [22]:

```
convert={"Cholesterol":{"HIGH":1,"NORMAL":2}}
df=df.replace(convert)
df
```

Out[22]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	1	1	25.355	drugY
1	47	1	3	1	13.093	drugC
2	47	1	3	1	10.114	drugC
3	28	0	2	1	7.798	drugX
4	61	0	3	1	18.043	drugY
...
195	56	0	3	1	11.567	drugC
196	16	1	3	1	12.006	drugC
197	52	1	2	1	9.894	drugX
198	23	1	2	2	14.020	drugX
199	40	0	3	2	11.349	drugX

200 rows × 6 columns

In [23]:

```
x=["Sex","BP","Cholesterol"]
y=["DrugY","DrugX","DrugA","DrugC","DrugB"]
all_inputs=df[x]
all_classes=df["Drug"]
```

In [24]:

```
x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=0.3)
clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

Out[24]:

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [25]:

```
score=clf.score(x_test,y_test)
print(score)
```

0.4666666666666667

In []: