

In [12]:

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

In [13]:

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

Out[13]:

	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 [14]:

```
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 [15]:

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

Out[15]:

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

In [16]:

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

Out[16]:

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

In [17]:

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

Out[17]:

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

In [18]:

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

Out[18]:

Age

47	8
23	7
28	7
49	7
39	6
32	6
50	5
37	5
58	5
60	5
22	5
34	4
72	4
51	4
42	4
26	4
24	4
74	4
67	4
68	4
61	4
56	4
20	4
36	4
45	4
41	4
31	4
43	4
65	4
57	4
53	3
40	3
70	3
59	3
16	3
38	3
15	3
69	3
35	3
18	3
64	3
52	2
55	2
62	2
19	2
29	2
66	2
73	2
46	2
48	2
54	1
17	1
33	1
63	1
30	1
21	1
25	1

Name: count, dtype: int64

In [19]:

```
convert={"Sex":{"M":1, "F":2}}
df=df.replace(convert)
df
```

Out[19]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	2	HIGH	HIGH	25.355	drugY
1	47	1	LOW	HIGH	13.093	drugC
2	47	1	LOW	HIGH	10.114	drugC
3	28	2	NORMAL	HIGH	7.798	drugX
4	61	2	LOW	HIGH	18.043	drugY
...
195	56	2	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	2	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [30]:

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

Out[30]:

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

200 rows × 6 columns

In [31]:

```
x=["Age","Sex","BP","Cholesterol","Na_to_K"]  
y=["drugX","drugY","drugC"]  
all_inputs=df[x]  
all_classes=df["Drug"]
```

In [32]:

```
(x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.3)
```

In [33]:

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

In [34]:

```
clf.fit(x_train,y_train)
```

Out[34]:

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

In [35]:

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

1.0

In []: