

```
In [1]: ▶ 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 [2]: ▶ df=pd.read_csv(r"C:\Users\munigreeshma\Downloads\drug200.csv")
df
```

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
...	...	...	...	...	...	...
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 [3]: `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 [4]: `df['Drug'].value_counts()`

```
Out[4]: Drug
drugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: count, dtype: int64
```

In [5]: ▶ df['Age'].value\_counts()

```
Out[5]: 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 [6]: ▶ convert={"Sex":{"F":1,"M":0}}
df=df.replace(convert)
df
```

Out[6]:

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

200 rows × 6 columns

```
In [7]: ▶ convert={'BP':{'HIGH':1,"LOW":2, "NORMAL":3}}
df=df.replace(convert)
df
```

Out[7]:

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

200 rows × 6 columns

```
In [8]: ▶ x=["Age", "Sex"]
y=["Yes", "No"]
all_inputs=df[x]
all_classes=df["Cholesterol"]
```

```
In [11]: ▶ (x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.5)
```

```
In [12]: ▶ clf=DecisionTreeClassifier(random_state=0)
```

In [13]: ► `clf.fit(x_train,y_train)`

Out[13]:

▼ DecisionTreeClassifier

DecisionTreeClassifier(random\_state=0)

In [14]: ► `score=clf.score(x_test,y_test)`  
`print(score)`

0.47