```
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\SATHI\OneDrive\Documents\drug200.csv")
df

## Out[2]:

Age	Sex	ВР	Cholesterol	Na_to_K	Drug
23	F	HIGH	HIGH	25.355	drugY
47	М	LOW	HIGH	13.093	drugC
47	М	LOW	HIGH	10.114	drugC
28	F	NORMAL	HIGH	7.798	drugX
61	F	LOW	HIGH	18.043	drugY
56	F	LOW	HIGH	11.567	drugC
16	М	LOW	HIGH	12.006	drugC
52	М	NORMAL	HIGH	9.894	drugX
23	М	NORMAL	NORMAL	14.020	drugX
40	F	LOW	NORMAL	11.349	drugX
	23 47 47 28 61  56 16 52 23	23 F 47 M 47 M 28 F 61 F 56 F 16 M 52 M 23 M	23 F HIGH 47 M LOW 47 M LOW 28 F NORMAL 61 F LOW 56 F LOW 16 M LOW 52 M NORMAL 23 M NORMAL	23 F HIGH HIGH 47 M LOW HIGH 47 M LOW HIGH 28 F NORMAL HIGH 61 F LOW HIGH 56 F LOW HIGH 16 M LOW HIGH 52 M NORMAL HIGH 23 M NORMAL NORMAL	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 56 F LOW HIGH 11.567 16 M LOW HIGH 12.006 52 M NORMAL HIGH 9.894 23 M NORMAL NORMAL 14.020

200 rows × 6 columns

memory usage: 9.5+ KB

## 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			
<pre>dtypes: float64(1), int64(1), object(4)</pre>						

localhost:8888/notebooks/drugs.ipynb

```
In [4]: df['BP'].value_counts()
        df['Na_to_K'].value_counts()
Out[4]: Na_to_K
        12.006
                  2
        18.295
                  2
        25.355
                  1
        11.939
                  1
        16.347
                  1
                  . .
        24.658
                  1
        24.276
                  1
        13.967
                  1
        19.675
                  1
        11.349
                  1
        Name: count, Length: 198, dtype: int64
In [5]: convert={"BP":{"HIGH":129,"NORMAL":80,"LOW":50}}
        df=df.replace(convert)
        df
```

## Out[5]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	129	HIGH	25.355	drugY
1	47	М	50	HIGH	13.093	drugC
2	47	М	50	HIGH	10.114	drugC
3	28	F	80	HIGH	7.798	drugX
4	61	F	50	HIGH	18.043	drugY
					•••	
195	56	F	50	HIGH	11.567	drugC
196	16	М	50	HIGH	12.006	drugC
197	52	М	80	HIGH	9.894	drugX
198	23	М	80	NORMAL	14.020	drugX
199	40	F	50	NORMAL	11.349	drugX

200 rows × 6 columns

```
In [6]: convert={"Cholesterol":{"HIGH":103,"NORMAL":97}}
df=df.replace(convert)
df
```

Out[6]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	129	103	25.355	drugY
1	47	М	50	103	13.093	drugC
2	47	М	50	103	10.114	drugC
3	28	F	80	103	7.798	drugX
4	61	F	50	103	18.043	drugY
195	56	F	50	103	11.567	drugC
196	16	М	50	103	12.006	drugC
197	52	М	80	103	9.894	drugX
198	23	М	80	97	14.020	drugX
199	40	F	50	97	11.349	drugX

200 rows × 6 columns

```
In [7]: x=["Age","BP","Cholesterol","Na_to_K"]
    y=["drugY","drugC"]
    all_inputs=df[x]
    all_classes=df["Drug"]
```

In [8]: (x\_train,x\_test,y\_train,y\_test)=train\_test\_split(all\_inputs,all\_classes,test\_size=0.3

```
In [9]: clf=DecisionTreeClassifier(random_state=0)
```

```
In [10]: clf.fit(x_train,y_train)
```

Out[10]: 

DecisionTreeClassifier

DecisionTreeClassifier(random\_state=0)

```
In [11]: score=clf.score(x_test,y_test)
print(score)
```

1.0

```
In [ ]:
```