

In [1]:

```

1 import numpy as np
2 import pandas as pd
3 import seaborn as sb
4 from sklearn.model_selection import train_test_split
5 from sklearn.tree import DecisionTreeClassifier
6 dv=pd.read_csv(r"C:\Users\kunam\Downloads\drug200.csv")
7 dv

```

Out[1]:

	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 [2]:

```
1 dv.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 [3]:

```
1 dv.columns
```

Out[3]:

```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

In [4]:

```
1 dv['Sex'].value_counts()
```

Out[4]:

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

In [5]:

```
1 dv['BP'].value_counts()
```

Out[5]:

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

In [6]:

```

1 convert={'Sex':{'M':1,'F':0}, 'BP':{'LOW':1,'NORMAL':2,'HIGH':3}, 'Cholesterol':{'L
2 dv=dv.replace(convert)
3 dv

```

Out[6]:

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

200 rows × 6 columns

In [7]:

```

1 x=['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']
2 y=['drugX','drugY','drugC']
3 all_inputs=dv[x]
4 all_classes=dv['Drug']

```

In [8]:

```

1 x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,train_size=0
2 clf1=DecisionTreeClassifier(random_state=0)
3 clf1.fit(x_train,y_train)

```

Out[8]:

DecisionTreeClassifier(random_state=0)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [9]:

```
1 score=clf1.score(x_test,y_test)
2 print(score)
```

1.0

In []:

```
1
```

In []:

```
1
```