

```
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\Sushma sree\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 [5]: df['Sex'].value_counts()
```

```
Out[5]: Sex
M      104
F       96
Name: count, dtype: int64
```

```
In [6]: df['BP'].value_counts()
```

```
Out[6]: BP
HIGH      77
LOW       64
NORMAL    59
Name: count, dtype: int64
```

```
In [7]: df['Cholesterol'].value_counts()
```

```
Out[7]: Cholesterol
HIGH      103
NORMAL     97
Name: count, dtype: int64
```

```
In [13]: df['Drug'].value_counts()
```

```
Out[13]: Drug
drugY     91
drugX     54
drugA     23
drugC     16
drugB     16
Name: count, dtype: int64
```

```
In [10]: convert={'Sex':{'F':1,'M':0}}
df=df.replace(convert)
df
```

Out[10]:

| | 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 [11]: convert={'BP':{'HIGH':1,'LOW':3,'NORMAL':2}}
df=df.replace(convert)
df
```

Out[11]:

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

200 rows × 6 columns

```
In [12]: convert={'Cholesterol':{'HIGH':1,'NORMAL':2}}
df=df.replace(convert)
df
```

Out[12]:

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

200 rows × 6 columns

```
In [14]: x=['Age', 'BP', 'Cholesterol']  
y=['drugX', 'drugY', 'drugA', 'drugB', 'drugC']  
all_inputs=df[x]  
all_classes=df['Drug']
```

```
In [15]: x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=0.2)
```

```
In [16]: clf=DecisionTreeClassifier(random_state=0)
```

```
In [17]: clf.fit(x_train,y_train)
```

```
Out[17]: ▾      DecisionTreeClassifier  
DecisionTreeClassifier(random_state=0)
```

```
In [18]: score=clf.score(x_test,y_test)  
print(score)
```

0.5

```
In [ ]:
```