

# Decision Tree

In [ ]:

```
1 # Loan Dataset
```

In [1]:

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as ana
4 from sklearn.model_selection import train_test_split
5 from sklearn.tree import DecisionTreeClassifier
```

In [2]:

```
1 df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\loan1.csv")
2 df
```

Out[2]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

In [4]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Home Owner            10 non-null    object  
1   Marital Status        10 non-null    object  
2   Annual Income         10 non-null    int64   
3   Defaulted Borrower    10 non-null    object  
dtypes: int64(1), object(3)
memory usage: 448.0+ bytes
```

In [5]:

```
1 df['Marital Status'].value_counts()
```

Out[5]:

```
Marital Status
Single      4
Married     4
Divorced    2
Name: count, dtype: int64
```

In [6]:

```
1 df['Annual Income'].value_counts()
```

Out[6]:

```
Annual Income
125      1
100      1
70       1
120      1
95       1
60       1
220      1
85       1
75       1
90       1
Name: count, dtype: int64
```

In [7]:

```
1 convert={"Home Owner":{"Yes":1,"No":0}}
2 df=df.replace(convert)
3 df
```

Out[7]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	1	Single	125	No
1	0	Married	100	No
2	0	Single	70	No
3	1	Married	120	No
4	0	Divorced	95	Yes
5	0	Married	60	No
6	1	Divorced	220	No
7	0	Single	85	Yes
8	0	Married	75	No
9	0	Single	90	Yes

In [8]:

```
1 convert={"Marital Status":{"Single":1,"Married":2,"Divorced":3}}
2 df=df.replace(convert)
3 df
```

Out[8]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	1	1	125	No
1	0	2	100	No
2	0	1	70	No
3	1	2	120	No
4	0	3	95	Yes
5	0	2	60	No
6	1	3	220	No
7	0	1	85	Yes
8	0	2	75	No
9	0	1	90	Yes

In [9]:

```
1 x=["Home Owner","Marital Status","Annual Income"]
2 y=["Yes","No"]
3 all_inputs=df[x]
4 all_classes=df["Defaulted Borrower"]
```

In [10]:

```
1 (x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.2)
```

In [11]:

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

In [12]:

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

Out[12]:

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

In [13]:

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

0.75

In [2]:

```
1 # Drug Dataset
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 from sklearn.model_selection import train_test_split
6 from sklearn.tree import DecisionTreeClassifier
```

In [3]:

```
1 df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\drug200.csv")
2 df
```

Out[3]:

	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 [4]:

```
1 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]:

```
1 df['Drug'].value_counts()
```

Out[5]:

Drug  
drugY 91  
drugX 54  
drugA 23  
drugC 16  
drugB 16  
Name: count, dtype: int64

In [7]:

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

Out[7]:

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

In [9]:

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

Out[9]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	1	HIGH	HIGH	25.355	drugY
1	47	2	LOW	HIGH	13.093	drugC
2	47	2	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	2	LOW	HIGH	12.006	drugC
197	52	2	NORMAL	HIGH	9.894	drugX
198	23	2	NORMAL	NORMAL	14.020	drugX
199	40	1	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [11]:

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

Out[11]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	1	1	1	25.355	drugY
1	47	2	3	1	13.093	drugC
2	47	2	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	2	3	1	12.006	drugC
197	52	2	2	1	9.894	drugX
198	23	2	2	2	14.020	drugX
199	40	1	3	2	11.349	drugX

200 rows × 6 columns

In [13]:

```
1 x=['Sex','BP','Cholesterol']
2 y=['drugY','drugC','drugX','drugA','drugB']
3 all_inputs=df[x]
4 all_classes=df['Drug']
```

In [14]:

```
1 x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=0.4)
```

In [18]:

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

In [19]:

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

Out[19]:

▼

DecisionTreeClassifier

DecisionTreeClassifier(random\_state=0)

In [22]:

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

0.475

In [ ]:

```
1
```