

# LOAN DATASET USING DECISION TREE

```
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\manis\OneDrive\Pictures\Documents\loan1.csv")
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 [3]: 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: 452.0+ bytes
```

```
In [4]: df['Marital Status'].value_counts()
```

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

```
In [5]: df['Annual Income'].value_counts()
```

```
Out[5]: 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 [6]: covert={"Home Owner":{"Yes":1,"No":0}}
df=df.replace(covert)
df
```

Out[6]:

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

Out[7]:

	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 [13]: x=["Home Owner","Marital Status","Annual Income"]
y=["Yes","No"]
all_inputs=df[x]
all_classes=df["Defaulted Borrower"]
x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=
```

```
In [14]: clf=DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
```

Out[14]:

▼ DecisionTreeClassifier  
DecisionTreeClassifier(random\_state=0)

```
In [15]: Score=clf.score(x_test,y_test)
print(Score)
```

0.5

## DRUG DATASET USING DECISION TREE

```
In [2]: 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 [3]: df=pd.read_csv(r"C:\Users\manis\OneDrive\Pictures\Documents\drug200.csv")
df
```

Out[3]:

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

200 rows × 6 columns

In [4]: `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['Drug'].value_counts()`

```
Out[5]: Drug
drugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: count, dtype: int64
```

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

```
Out[6]: 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 [7]: covert={"Cholesterol":{"HIGH":1,"NORMAL":0}}
df=df.replace(covert)
df
```

Out[7]:

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

200 rows × 6 columns

```
In [8]: covert={"BP":{"HIGH":1,"LOW":2,"NORMAL":3}}
df=df.replace(covert)
df
```

Out[8]:

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

200 rows × 6 columns

In [9]:

```
covert={"Drug":{"drugX":1,"drugY":2,"drugA":3,"drugB":4,"drugC":5}}
df=df.replace(covert)
df
```

Out[9]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	1	1	25.355	2
1	47	M	2	1	13.093	5
2	47	M	2	1	10.114	5
3	28	F	3	1	7.798	1
4	61	F	2	1	18.043	2
...	...	...	...	...	...	...
195	56	F	2	1	11.567	5
196	16	M	2	1	12.006	5
197	52	M	3	1	9.894	1
198	23	M	3	0	14.020	1
199	40	F	2	0	11.349	1

200 rows × 6 columns

In [10]:

```
x=["BP","Cholesterol","Drug","Na_to_K","Age"]
y=["HIGH","NORMAL","LOW"]
all_inputs=df[x]
all_classes=df["Sex"]
x_train,x_test,y_train,y_test=train_test_split(all_inputs,all_classes,test_size=
```

```
In [11]: clf=DecisionTreeClassifier(random_state=0)
         clf.fit(x_train,y_train)
```

```
Out[11]: ▾      DecisionTreeClassifier
         DecisionTreeClassifier(random_state=0)
```

```
In [12]: Score=clf.score(x_test,y_test)
         print(Score)
```

0.575

```
In [ ]:
```