

In [1]:

```
import pandas as pd

df=pd.read_csv("./datasets/original.csv")
df
```

Out[1]:

	clientid	income	age	loan	default
0	1	66155.925095	59.017015	8106.532131	0
1	2	34415.153966	48.117153	6564.745018	0
2	3	57317.170063	63.108049	8020.953296	0
3	4	42709.534201	45.751972	6103.642260	0
4	5	66952.688845	18.584336	8770.099235	1
...
1995	1996	59221.044874	48.518179	1926.729397	0
1996	1997	69516.127573	23.162104	3503.176156	0
1997	1998	44311.449262	28.017167	5522.786693	1
1998	1999	43756.056605	63.971796	1622.722598	0
1999	2000	69436.579552	56.152617	7378.833599	0

2000 rows × 5 columns

In [2]:

```
df.isna().sum()
```

Out[2]:

```
clientid    0
income      0
age         3
loan        0
default     0
dtype: int64
```

In [3]:

```
df.dropna(inplace=True)
df
```

Out[3]:

	clientid	income	age	loan	default
0	1	66155.925095	59.017015	8106.532131	0
1	2	34415.153966	48.117153	6564.745018	0
2	3	57317.170063	63.108049	8020.953296	0
3	4	42709.534201	45.751972	6103.642260	0
4	5	66952.688845	18.584336	8770.099235	1
...
1995	1996	59221.044874	48.518179	1926.729397	0
1996	1997	69516.127573	23.162104	3503.176156	0
1997	1998	44311.449262	28.017167	5522.786693	1
1998	1999	43756.056605	63.971796	1622.722598	0
1999	2000	69436.579552	56.152617	7378.833599	0

1997 rows × 5 columns

In [4]:

```
#2 points

features=df[["income","age","loan"]] #best contributors!
label=df[["default"]]
```

In [5]:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(features,label,test_size=0.2)
```

In [6]:

```
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier() #Decision Tree
```

In [7]:

```
model.fit(x_train,y_train)
```

Out[7]:

```
DecisionTreeClassifier()
```

In [8]:

```
model.score(x_test,y_test)
```

Out[8]:

```
0.9875
```

In [9]:

```
ans=model.predict(x_test)
ans
```

Out[9]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0])
```

In [10]:

```
predicted=pd.DataFrame(ans)
predicted.head(5)
```

Out[10]:

```
0
0 0
1 0
2 0
3 0
4 0
```

In [11]:

```
actual_test=pd.DataFrame(y_test).reset_index(drop=True)
actual_test.head(5)
```

Out[11]:

```
default
0      0
1      0
2      0
3      0
4      0
```

In [12]:

```
pd.concat([predicted,actual_test],axis=1).sample(5)
```

Out[12]:

	0	default
126	0	0
103	0	0
29	0	0
51	1	1
324	0	0

In [13]:

```
from sklearn.metrics import accuracy_score  
  
accuracy_score(actual_test,predicted)
```

Out[13]:

0.9875

In [14]:

```
from sklearn.metrics import confusion_matrix  
confusion_matrix(actual_test,predicted)
```

Out[14]:

```
array([[341,  4],  
       [ 1, 54]])
```

In []: