In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
heart_disease = pd.read_csv("D:\_teachings\heart-disease.csv")
heart_disease
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

In [3]:
```
heart_disease.dtypes
```

Out[3]:
```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

Make data labelled

x is input [features]

y is output [target]

split data into x and y

In [4]:
```
x = heart_disease.drop("target",axis=1)
x
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2

303 rows × 13 columns

In [5]:
```
y = heart_disease["target"]
y
```

Out[5]:
```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64
```

In [6]:
```
#split data into training and testing data
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

In [7]:
```
x_train
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
270	46	1	0	120	249	0	0	144	0	0.8	2	0	3
86	68	1	2	118	277	0	1	151	0	1.0	2	1	3
171	48	1	1	110	229	0	1	168	0	1.0	0	0	3
209	59	1	0	140	177	0	1	162	1	0.0	2	1	3

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
41	48	1	1	130	245	0	0	180	0	0.2	1	0	2
234	70	1	0	130	322	0	0	109	0	2.4	1	3	2
111	57	1	2	150	126	1	1	173	0	0.2	2	1	3
52	62	1	2	130	231	0	1	146	0	1.8	1	3	3
55	52	1	1	134	201	0	1	158	0	0.8	2	1	2

242 rows × 13 columns

In [8]:
```
x_test
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
267	49	1	2	118	149	0	0	126	0	0.8	2	3	2
10	54	1	0	140	239	0	1	160	0	1.2	2	0	2
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
288	57	1	0	110	335	0	1	143	1	3.0	1	1	3
68	44	1	1	120	220	0	1	170	0	0.0	2	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
61	54	1	1	108	309	0	1	156	0	0.0	2	0	3
33	54	1	2	125	273	0	0	152	0	0.5	0	1	2
223	56	0	0	200	288	1	0	133	1	4.0	0	2	3
280	42	1	0	136	315	0	1	125	1	1.8	1	0	1
135	49	0	0	130	269	0	1	163	0	0.0	2	0	2

61 rows × 13 columns

In [9]:
```
y_train
```

Out[9]:
```
270    0
86     1
171    0
209    0
301    0
      ..
41     1
234    0
111    1
52     1
55     1
Name: target, Length: 242, dtype: int64
```

In [10]:
```
y_test
```

Out[10]:
```
267    0
```

```
10     1
1      1
288    0
68     1
      ..
61     1
33     1
223    0
280    0
135    1
Name: target, Length: 61, dtype: int64
```

In [11]:

```
#bulid machine learning model

from sklearn.ensemble import RandomForestClassifier

model=RandomForestClassifier()

model.get_params()
```

Out[11]:

```
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_impurity_split': None,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

In [12]:

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

Out[12]:

```
RandomForestClassifier()
```

In [19]:

```
#make pediction
y_prediction = model.predict(x_test)
y_prediction
```

Out[19]:

```
array([1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1], dtype=int64)
```

In [14]:

```
y_test
```

Out[14]:

```
267    0
10     1
1      1
288    0
68     1
      ..
61     1
33     1
```

```
223    0
280    0
135    1
Name: target, Length: 61, dtype: int64
```

In [15]:
```
#evaluate the model
model.score(x_train,y_train)
```

Out[15]:
```
1.0
```

In [16]:
```
model.score(x_test,y_test)
```

Out[16]:
```
0.819672131147541
```

In [20]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_prediction)
```

Out[20]:
```
0.819672131147541
```

In [ ]: