

In [1]: `# Experiment NO.:12`

In [ ]: `#Aim :To perform and Analysis of Decision Tree Algorithm`

In [2]: `#Name: Sakshi Rambhau Wankhade  
#Roll No: 72  
#Sec: A  
#Subject: ET-1  
#Date:30-09-2025`

In [3]: `#importing the basic library  
import pandas as pd  
import numpy as np`

In [4]: `import os`

In [5]: `os.getcwd()`

Out[5]: `'C:\\Users\\ADMIN\\DSS_practical'`

In [6]: `os.chdir('C:\\Users\\ADMIN\\DSS_practical')`

In [7]: `data=pd.read_csv("heart.csv")`

In [8]: `data.head()`


Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	ti
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	

In [9]: `data.tail()`

Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
<b>1020</b>	59	1	1	140	221	0	1	164	1	0.0	2	0	2
<b>1021</b>	60	1	0	125	258	0	0	141	1	2.8	1	1	3
<b>1022</b>	47	1	0	110	275	0	0	118	1	1.0	1	1	2
<b>1023</b>	50	0	0	110	254	0	0	159	0	0.0	2	0	2
<b>1024</b>	54	1	0	120	188	0	1	113	0	1.4	1	1	3



```
In [10]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         1025 non-null   int64
1   sex         1025 non-null   int64
2   cp          1025 non-null   int64
3   trestbps    1025 non-null   int64
4   chol        1025 non-null   int64
5   fbs         1025 non-null   int64
6   restecg     1025 non-null   int64
7   thalach     1025 non-null   int64
8   exang       1025 non-null   int64
9   oldpeak     1025 non-null   float64
10  slope       1025 non-null   int64
11  ca          1025 non-null   int64
12  thal        1025 non-null   int64
13  target      1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
In [11]: data.describe()
```

Out[11]:

	age	sex	cp	trestbps	chol	fbs	restecg
<b>count</b>	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
<b>mean</b>	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.500000
<b>std</b>	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.500000
<b>min</b>	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
<b>25%</b>	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
<b>50%</b>	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
<b>75%</b>	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000
<b>max</b>	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

In [12]: `data.shape`

Out[12]: (1025, 14)

In [13]: `data.size`

Out[13]: 14350

In [14]: `data.ndim`

Out[14]: 2

## Data preprocessing \_ data cleaning \_ missing value treatment

In [15]: `# check Missing Value by record`  
`data.isna()`

Out[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
<b>0</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>1</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>2</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>3</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>4</b>	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
<b>1020</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>1021</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>1022</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>1023</b>	False	False	False	False	False	False	False	False	False	False	False	False
<b>1024</b>	False	False	False	False	False	False	False	False	False	False	False	False

1025 rows × 14 columns

In [16]: `data.isna().any()`

```
Out[16]: age      False
sex        False
cp          False
trestbps   False
chol        False
fbs         False
restecg     False
thalach     False
exang       False
oldpeak     False
slope       False
ca          False
thal        False
target      False
dtype: bool
```

In [17]: `data.isna().sum()`

```
Out[17]: age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

## Independent and Dependent Variables

```
In [18]: x=data.drop("target", axis=1)
y=data["target"]
```

```
In [19]: from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd
```

```
In [20]: # Load dataset
data = load_iris()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)
```

```
In [21]: # Use the same random state and test size that likely gave you 0.985...
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.35, random_st
```

```
In [22]: # Use RandomForestClassifier with default params (or tweak n_estimators if needed)
rf = RandomForestClassifier(random_state=0)
rf.fit(x_train, y_train)
```

```
Out[22]: ▼ RandomForestClassifier ⓘ ?
RandomForestClassifier(random_state=0)
```

```
In [23]: # Predict
y_pred5 = rf.predict(x_test)
```

```
In [24]: y_pred5 = rf.predict(x_test)
```

```
In [25]: accuracy = accuracy_score(y_test, y_pred5)
print("Accuracy:", accuracy)
```

Accuracy: 0.9622641509433962

In [ ]: