

## ▼ Data Cleaning

```
import pandas as pd

data = pd.read_csv(r'/content/heartdisease.csv');
```

data

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpe
0	1	63	1	typical	145	233	1	2	150	0	2
1	2	67	1	asymptomatic	160	286	0	2	108	1	1
2	3	67	1	asymptomatic	120	229	0	2	129	1	2
3	4	37	1	nonanginal	130	250	0	0	187	0	3
4	5	41	0	nontypical	130	204	0	2	172	0	1
...	...	...	...	...	...	...	...	...	...	...	...
298	299	45	1	typical	110	264	0	0	132	0	1
299	300	68	1	asymptomatic	144	193	1	0	141	0	3
300	301	57	1	asymptomatic	130	131	0	0	115	1	1
301	302	57	0	nontypical	130	236	0	2	174	0	0
302	303	38	1	nonanginal	138	175	0	0	173	0	0

303 rows × 15 columns

```
data.shape

(303, 15)
```

```
data.dtypes

Unnamed: 0      int64
Age             int64
Sex             int64
ChestPain      object
RestBP         int64
Chol           int64
Fbs            int64
RestECG        int64
MaxHR          int64
ExAng          int64
Oldpeak        float64
Slope          int64
Ca             float64
Thal           object
```

```
AHD          object
dtype: object
```

```
data.columns
```

```
Index(['Unnamed: 0', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs',
       'RestECG', 'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal', 'AHD'],
      dtype='object')
```

```
data.describe()
```

	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	152.000000	54.438944	0.679868	131.689769	246.693069	0.148515	0.990099
std	87.612784	9.038662	0.467299	17.599748	51.776918	0.356198	0.994977
min	1.000000	29.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	76.500000	48.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	152.000000	56.000000	1.000000	130.000000	241.000000	0.000000	1.000000
75%	227.500000	61.000000	1.000000	140.000000	275.000000	0.000000	2.000000

```
data.head(10)
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak
0	1	63	1	typical	145	233	1	2	150	0	2.3
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5
4	5	41	0	nontypical	130	204	0	2	172	0	1.4
5	6	56	1	nontypical	120	236	0	0	178	0	0.8
6	7	62	0	asymptomatic	140	268	0	2	160	0	3.6
7	8	57	0	asymptomatic	120	354	0	0	163	1	0.6
8	9	63	1	asymptomatic	130	254	0	2	147	0	1.4

## ▼ Handling Missing Values

```
data.isnull()
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak
0		False	False	False	False	False	False	False	False	False	
1		False	False	False	False	False	False	False	False	False	
2		False	False	False	False	False	False	False	False	False	
3		False	False	False	False	False	False	False	False	False	
4		False	False	False	False	False	False	False	False	False	
...		...	...	...	...	...	...	...	...	...	...
298		False	False	False	False	False	False	False	False	False	
299		False	False	False	False	False	False	False	False	False	
300		False	False	False	False	False	False	False	False	False	
301		False	False	False	False	False	False	False	False	False	
302		False	False	False	False	False	False	False	False	False	

```
data.isnull().sum()
```

```
Unnamed: 0      0
Age             0
Sex             0
ChestPain       0
RestBP          0
Chol            0
Fbs             0
RestECG         0
MaxHR           0
ExAng           0
Oldpeak         0
Slope           0
Ca              4
Thal            2
AHD             0
dtype: int64
```

```
# drop column with missing values

temp = data
temp.dropna()
```

Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak
0	1	63	1	typical	145	233	1	2	150	0
1	2	67	1	asymptomatic	160	286	0	2	108	1
2	3	67	1	asymptomatic	120	229	0	2	129	1
3	4	37	1	nonanginal	130	250	0	0	187	0
4	5	41	0	nontypical	130	204	0	2	172	0
...	...	...	...	...	...	...	...	...	...	...
297	298	57	0	asymptomatic	140	241	0	0	123	1
298	299	45	1	typical	110	264	0	0	132	0

```
# filling missing values
```

```
# 1. filling with 0 `
filled_dataset = data.fillna(0);
filled_dataset.isna().sum()
```

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            0
Thal          0
AHD           0
dtype: int64
```

```
# 2. filling with mean
# mean only works with numeric datatype
```

```
filled_dataset = data.fillna(data.mean())
filled_dataset.isna().sum()
```

```
<ipython-input-12-093ec2471699>:4: FutureWarning: The default value of numeric_only is
filled_dataset = data.fillna(data.mean())
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
```

```

ExAng      0
Oldpeak    0
Slope      0
Ca         0
Thal       2
AHD        0
dtype: int64

```

```
# 3. filling with mode
```

```

temp = data
temp['Ca'] = temp['Ca'].fillna(temp['Ca'].mode()[0])
temp['Thal'] = temp['Thal'].fillna(temp['Thal'].mode()[0])

temp.isna().sum()

```

```

Unnamed: 0    0
Age          0
Sex          0
ChestPain     0
RestBP       0
Chol         0
Fbs          0
RestECG      0
MaxHR        0
ExAng        0
Oldpeak      0
Slope        0
Ca           0
Thal         0
AHD          0
dtype: int64

```

```
# 4. forward and backward filling
```

```

temp = data
temp['Ca'].fillna(method='ffill', inplace=True)
temp['Thal'].fillna(method='bfill', inplace=True)

```

```
temp.isna().sum()
```

```

Unnamed: 0    0
Age          0
Sex          0
ChestPain     0
RestBP       0
Chol         0
Fbs          0
RestECG      0
MaxHR        0
ExAng        0
Oldpeak      0
Slope        0
Ca           0
Thal         0

```

```
AHD      0
dtype: int64
```

## ▼ Handling Duplicates

```
# adding a duplicate row
data.loc[303] = [304, 63, 1, 'typical', 145, 233, 1, 2, 150, 0, 2.3, 3, 0.0, 'f
```

```
data.shape
```

```
(304, 15)
```

```
data.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
...
299    False
300    False
301    False
302    False
303    False
Length: 304, dtype: bool
```

```
new_data = data.drop(data.columns[0], axis = 1)
new_data.duplicated().sum()
```

```
1
```

```
new_data.drop_duplicates(inplace=True)
```

```
new_data.duplicated().sum()
```

```
0
```

## ▼ Renaming Column

```
data.rename(columns={data.columns[0]: 'No'}, inplace=True)
```

```
data.columns
```

```
Index(['No', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs', 'RestECG',
       'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal', 'AHD'],
      dtype='object')
```

## ▼ Data Integration

```
data1 = data.iloc[0:50, 0:5]  
data1
```

18	19	48	0	nonanginal	130
19	20	49	1	nontypical	130
20	21	64	1	typical	110
21	22	58	0	typical	150
22	23	58	1	nontypical	120
23	24	58	1	nonanginal	132
24	25	60	1	asymptomatic	130
25	26	50	0	nonanginal	120
26	27	58	0	nonanginal	120
27	28	66	0	typical	150
28	29	43	1	asymptomatic	150
29	30	40	1	asymptomatic	110
30	31	69	0	typical	140
31	32	60	1	asymptomatic	117
32	33	64	1	nonanginal	140
33	34	59	1	asymptomatic	135
34	35	44	1	nonanginal	130
35	36	42	1	asymptomatic	140
36	37	43	1	asymptomatic	120
37	38	57	1	asymptomatic	150
38	39	55	1	asymptomatic	132
39	40	61	1	nonanginal	150
40	41	65	0	asymptomatic	150
41	42	40	1	typical	140
42	43	71	0	nontypical	160
43	44	50	1	nonanginal	150

```
data2 = data.iloc[0:50, 5:10]
data2
```



18	275	0	0	139	0
19	266	0	0	171	0
20	211	0	2	144	1
21	283	1	2	162	0
22	284	0	2	160	0
23	224	0	2	173	0
24	206	0	2	132	1
25	219	0	0	158	0
26	340	0	0	172	0
27	226	0	0	114	0
28	247	0	0	171	0
29	167	0	2	114	1
30	239	0	0	151	0
31	230	1	0	160	1
32	335	0	0	158	0
33	234	0	0	161	0
34	233	0	0	179	1
35	226	0	0	178	0
36	177	0	2	120	1
37	276	0	2	112	1
38	353	0	0	132	1
39	243	1	0	137	1
40	225	0	2	114	0
41	199	0	0	178	1
42	302	0	0	162	0
43	212	1	0	157	0
44	330	0	2	169	0
45	230	0	2	165	0
46	175	0	0	123	0
47	243	0	2	128	0
48	417	1	2	157	0
49	197	1	2	152	0

```
data_concat = pd.concat([data1, data2])
```

```
data_concat
```

	No	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng
0	1.0	63.0	1.0	typical	145.0	NaN	NaN	NaN	NaN	NaN
1	2.0	67.0	1.0	asymptomatic	160.0	NaN	NaN	NaN	NaN	NaN
2	3.0	67.0	1.0	asymptomatic	120.0	NaN	NaN	NaN	NaN	NaN
3	4.0	37.0	1.0	nonanginal	130.0	NaN	NaN	NaN	NaN	NaN
4	5.0	41.0	0.0	nontypical	130.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
45	NaN	NaN	NaN	NaN	NaN	230.0	0.0	2.0	165.0	0.0
46	NaN	NaN	NaN	NaN	NaN	175.0	0.0	0.0	123.0	0.0
47	NaN	NaN	NaN	NaN	NaN	243.0	0.0	2.0	128.0	0.0
48	NaN	NaN	NaN	NaN	NaN	417.0	1.0	2.0	157.0	0.0
49	NaN	NaN	NaN	NaN	NaN	197.0	1.0	2.0	152.0	0.0

100 rows × 10 columns

```
data_merge = pd.merge(data1, data2, left_index=True, right_index=True)
```

```
data_merge
```

18	19	48	0	nonanginal	130	275	0	0	139	0
19	20	49	1	nontypical	130	266	0	0	171	0
20	21	64	1	typical	110	211	0	2	144	1
21	22	58	0	typical	150	283	1	2	162	0
22	23	58	1	nontypical	120	284	0	2	160	0
23	24	58	1	nonanginal	132	224	0	2	173	0
24	25	60	1	asymptomatic	130	206	0	2	132	1
25	26	50	0	nonanginal	120	219	0	0	158	0
26	27	58	0	nonanginal	120	340	0	0	172	0
27	28	66	0	typical	150	226	0	0	114	0
28	29	43	1	asymptomatic	150	247	0	0	171	0
29	30	40	1	asymptomatic	110	167	0	2	114	1
30	31	69	0	typical	140	239	0	0	151	0
31	32	60	1	asymptomatic	117	230	1	0	160	1
32	33	64	1	nonanginal	140	335	0	0	158	0
33	34	59	1	asymptomatic	135	234	0	0	161	0
34	35	44	1	nonanginal	130	233	0	0	179	1
35	36	42	1	asymptomatic	140	226	0	0	178	0
36	37	43	1	asymptomatic	120	177	0	2	120	1
37	38	57	1	asymptomatic	150	276	0	2	112	1
38	39	55	1	asymptomatic	132	353	0	0	132	1
39	40	61	1	nonanginal	150	243	1	0	137	1
40	41	65	0	asymptomatic	150	225	0	2	114	0
41	42	40	1	typical	140	199	0	0	178	1
42	43	71	0	nontypical	160	302	0	0	162	0
43	44	59	1	nonanginal	150	212	1	0	157	0
44	45	61	0	asymptomatic	130	330	0	2	169	0
45	46	58	1	nonanginal	112	230	0	2	165	0
46	47	51	1	nonanginal	110	175	0	0	123	0
47	48	50	1	asymptomatic	150	243	0	2	128	0
48	49	65	0	nonanginal	140	417	1	2	157	0
49	50	53	1	nonanginal	130	197	1	2	152	0

## ▼ Data Transformation

modifying structure or content of data

```
data['ChestPain'] = data['ChestPain'].replace('typical', 0)
data['ChestPain'] = data['ChestPain'].replace('asymptomatic', 1)
data['ChestPain'] = data['ChestPain'].replace('nonanginal', 2)
data['ChestPain'] = data['ChestPain'].replace('nontypical', 3)
```

```
data['ChestPain']
```

```
0      0
1      1
2      1
3      2
4      3
..
299    1
300    1
301    3
302    2
303    0
Name: ChestPain, Length: 304, dtype: int64
```

## ▼ Error Correction

```
# adding some error in dataset
```

```
data.loc[305] = [304, 63, 1, 'typical', 145, 233, 1, 2, 150, 0, 2.3, 3, '?', 'f']
data.loc[306] = [304, 63, 1, 'typical', 145, 233, 1, 2, 150, 0, 2.3, 3, '?', 'f']
```

```
data['Ca'].unique()
```

```
array([0.0, 3.0, 2.0, 1.0, '?'], dtype=object)
```

```
data['Ca'] = data['Ca'].replace('?', data['Ca'].mode()[0])
```

```
data['Ca'].unique()
```

```
array([0., 3., 2., 1.])
```

```
data['Ca'].value_counts()
```

```
0.0    183
1.0     65
2.0     38
3.0     20
Name: Ca, dtype: int64
```

## Model Building

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn import tree
```

```
feature_cols = ['Age', 'ChestPain', 'RestBP', 'Fbs', 'RestECG', 'MaxHR', 'Slope']
```

```
X = data[feature_cols]
X
```

	Age	ChestPain	RestBP	Fbs	RestECG	MaxHR	Slope
0	63	0	145	1	2	150	3
1	67	1	160	0	2	108	2
2	67	1	120	0	2	129	2
3	37	2	130	0	0	187	3
4	41	3	130	0	2	172	1
...	...	...	...	...	...	...	...
301	57	3	130	0	2	174	2
302	38	2	138	0	0	173	1
303	63	0	145	1	2	150	3
305	63	typical	145	1	2	150	3
306	63	typical	145	1	2	150	3

306 rows × 7 columns

```
y = data[['AHD']]
y
```