

Iris Dataset

```
In [173... import pandas as pd
import numpy as np
```

```
In [174... df=pd.read_csv("Iris.csv")
df
```

Out[174]:

	Id	SL	SW	PL	PW	Species	
	0	1	5.1	3.5	1.4	0.2	Iris-setosa
	1	2	4.9	3.0	1.4	0.2	Iris-setosa
	2	3	4.7	3.2	1.3	0.2	Iris-setosa
	3	4	4.6	3.1	1.5	0.2	Iris-setosa
	4	5	5.0	3.6	1.4	0.2	Iris-setosa

	145	146	6.7	3.0	5.2	2.3	Iris-virginica
	146	147	6.3	2.5	5.0	1.9	Iris-virginica
	147	148	6.5	3.0	5.2	2.0	Iris-virginica
	148	149	6.2	3.4	5.4	2.3	Iris-virginica
	149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
In [175... df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Id           150 non-null    int64
1   SL           150 non-null    float64
2   SW           150 non-null    float64
3   PL           150 non-null    float64
4   PW           150 non-null    float64
5   Species      150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
In [176... df.isnull()
```

Out[176]:

	Id	SL	SW	PL	PW	Species
	0	False	False	False	False	False
	1	False	False	False	False	False
	2	False	False	False	False	False
	3	False	False	False	False	False
	4	False	False	False	False	False

	145	False	False	False	False	False
	146	False	False	False	False	False
	147	False	False	False	False	False
	148	False	False	False	False	False
	149	False	False	False	False	False

150 rows × 6 columns

```
In [177... df.isnull().sum()
```

```
Out[177]: Id      0
          SL      0
          SW      0
          PL      0
          PW      0
          Species  0
          dtype: int64
```

```
In [178... df.describe()
```

```
Out[178]:
```

	Id	SL	SW	PL	PW
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
In [179... X=df.iloc[:,1:5]
          Y=df.iloc[:, -1]
```

```
In [180... X
```

```
Out[180]:
```

	SL	SW	PL	PW
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
In [181... Y
```

```
Out[181]: 0      Iris-setosa
          1      Iris-setosa
          2      Iris-setosa
          3      Iris-setosa
          4      Iris-setosa
          ...
          145    Iris-virginica
          146    Iris-virginica
          147    Iris-virginica
          148    Iris-virginica
          149    Iris-virginica
          Name: Species, Length: 150, dtype: object
```

```
In [182... from sklearn.model_selection import train_test_split
```

```
In [183... X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=42)
```

```
In [184... X_train
```

Out[184]:

	SL	SW	PL	PW
4	5.0	3.6	1.4	0.2
32	5.2	4.1	1.5	0.1
142	5.8	2.7	5.1	1.9
85	6.0	3.4	4.5	1.6
86	6.7	3.1	4.7	1.5
...
71	6.1	2.8	4.0	1.3
106	4.9	2.5	4.5	1.7
14	5.8	4.0	1.2	0.2
92	5.8	2.6	4.0	1.2
102	7.1	3.0	5.9	2.1

112 rows × 4 columns

In [185... `x_train.shape`

Out[185]: `(112, 4)`

In [186... `x_test`

Out[186]:

	SL	SW	PL	PW
73	6.1	2.8	4.7	1.2
18	5.7	3.8	1.7	0.3
118	7.7	2.6	6.9	2.3
78	6.0	2.9	4.5	1.5
76	6.8	2.8	4.8	1.4
31	5.4	3.4	1.5	0.4
64	5.6	2.9	3.6	1.3
141	6.9	3.1	5.1	2.3
68	6.2	2.2	4.5	1.5
82	5.8	2.7	3.9	1.2
110	6.5	3.2	5.1	2.0
12	4.8	3.0	1.4	0.1
36	5.5	3.5	1.3	0.2
9	4.9	3.1	1.5	0.1
19	5.1	3.8	1.5	0.3
56	6.3	3.3	4.7	1.6
104	6.5	3.0	5.8	2.2
69	5.6	2.5	3.9	1.1
55	5.7	2.8	4.5	1.3
132	6.4	2.8	5.6	2.2
29	4.7	3.2	1.6	0.2
127	6.1	3.0	4.9	1.8
26	5.0	3.4	1.6	0.4
128	6.4	2.8	5.6	2.1
131	7.9	3.8	6.4	2.0
145	6.7	3.0	5.2	2.3
108	6.7	2.5	5.8	1.8
143	6.8	3.2	5.9	2.3
45	4.8	3.0	1.4	0.3
30	4.8	3.1	1.6	0.2
22	4.6	3.6	1.0	0.2
15	5.7	4.4	1.5	0.4
65	6.7	3.1	4.4	1.4
11	4.8	3.4	1.6	0.2
42	4.4	3.2	1.3	0.2
146	6.3	2.5	5.0	1.9
51	6.4	3.2	4.5	1.5
27	5.2	3.5	1.5	0.2

In [187... X_test.shape

Out[187]: (38, 4)

In [188... Y_train

```
Out[188]: 4      Iris-setosa
          32      Iris-setosa
          142     Iris-virginica
          85      Iris-versicolor
          86      Iris-versicolor
          ...
          71      Iris-versicolor
          106     Iris-virginica
          14      Iris-setosa
          92      Iris-versicolor
          102     Iris-virginica
          Name: Species, Length: 112, dtype: object
```

```
In [189... Y_train.shape
```

```
Out[189]: (112,)
```

```
In [190... Y_test
```

```
Out[190]: 73      Iris-versicolor
          18      Iris-setosa
          118     Iris-virginica
          78      Iris-versicolor
          76      Iris-versicolor
          31      Iris-setosa
          64      Iris-versicolor
          141     Iris-virginica
          68      Iris-versicolor
          82      Iris-versicolor
          110     Iris-virginica
          12      Iris-setosa
          36      Iris-setosa
          9       Iris-setosa
          19      Iris-setosa
          56      Iris-versicolor
          104     Iris-virginica
          69      Iris-versicolor
          55      Iris-versicolor
          132     Iris-virginica
          29      Iris-setosa
          127     Iris-virginica
          26      Iris-setosa
          128     Iris-virginica
          131     Iris-virginica
          145     Iris-virginica
          108     Iris-virginica
          143     Iris-virginica
          45      Iris-setosa
          30      Iris-setosa
          22      Iris-setosa
          15      Iris-setosa
          65      Iris-versicolor
          11      Iris-setosa
          42      Iris-setosa
          146     Iris-virginica
          51      Iris-versicolor
          27      Iris-setosa
          Name: Species, dtype: object
```

```
In [191... Y_test.shape
```

```
Out[191]: (38,)
```

```
In [192... from sklearn.tree import DecisionTreeClassifier
```

```
In [193... dt=DecisionTreeClassifier()
```

```
In [194... dt.fit(X_train,Y_train)
```

```
Out[194]: ▾ DecisionTreeClassifier
          DecisionTreeClassifier()
```

```
In [195... Y_pred=dt.predict(X_test)
```

```
In [196... Y_pred
```

```
Out[196]: array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
                'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
                'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
                'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
                'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica',
                'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
                'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
                'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
                'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
                'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
                'Iris-virginica', 'Iris-versicolor', 'Iris-setosa'], dtype=object)
```

```
In [197... dt.score(X_test,Y_test)
```

```
Out[197]: 1.0
```

```
In [198... from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score,confusion_matrix
```

```
In [199... accuracy_score(Y_test,Y_pred)
```

```
Out[199]: 1.0
```

```
In [200... precision_score(Y_test, Y_pred, average='micro')
```

```
Out[200]: 1.0
```

```
In [201... recall_score(Y_test, Y_pred, average='micro')
```

```
Out[201]: 1.0
```

```
In [202... f1_score(Y_test, Y_pred, average='micro')
```

```
Out[202]: 1.0
```

```
In [203... confusion_matrix(Y_test, Y_pred)
```

```
Out[203]: array([[15,  0,  0],
                 [ 0, 11,  0],
                 [ 0,  0, 12]], dtype=int64)
```

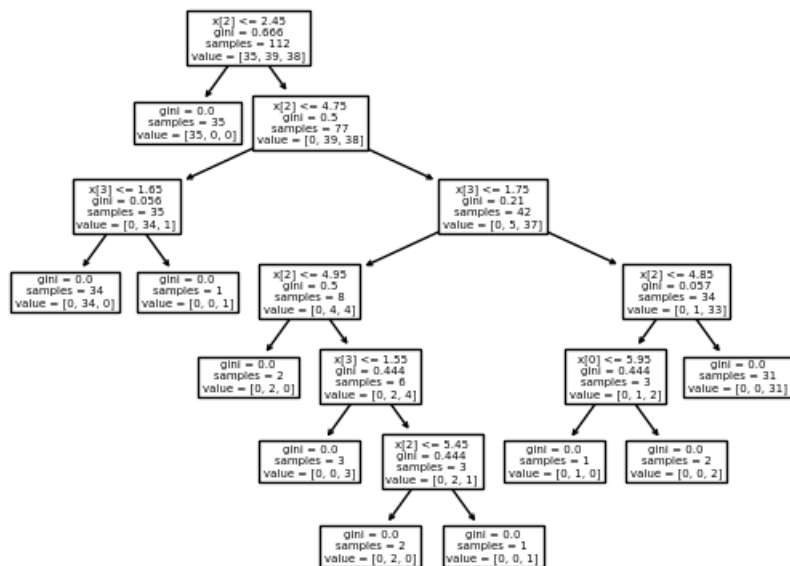
```
In [204... from sklearn import tree
```

```
In [205... clf=tree.DecisionTreeClassifier()
```

```
In [206... clf=clf.fit(X_train,Y_train)
```

```
In [207... tree.plot_tree(clf)
```

```
Out[207]: [Text(0.3076923076923077, 0.9285714285714286, 'x[2] <= 2.45\ngini = 0.666\nsamples = 112\nvalue = [35, 39, 38]'),
          Text(0.23076923076923078, 0.7857142857142857, 'gini = 0.0\nsamples = 35\nvalue = [35, 0, 0]'),
          Text(0.38461538461538464, 0.7857142857142857, 'x[2] <= 4.75\ngini = 0.5\nsamples = 77\nvalue = [0, 39, 38]'),
          Text(0.15384615384615385, 0.6428571428571429, 'x[3] <= 1.65\ngini = 0.056\nsamples = 35\nvalue = [0, 34, 1]'),
          Text(0.07692307692307693, 0.5, 'gini = 0.0\nsamples = 34\nvalue = [0, 34, 0]'),
          Text(0.23076923076923078, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
          Text(0.6153846153846154, 0.6428571428571429, 'x[3] <= 1.75\ngini = 0.21\nsamples = 42\nvalue = [0, 5, 37]'),
          Text(0.38461538461538464, 0.5, 'x[2] <= 4.95\ngini = 0.5\nsamples = 8\nvalue = [0, 4, 4]'),
          Text(0.3076923076923077, 0.35714285714285715, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
          Text(0.46153846153846156, 0.35714285714285715, 'x[3] <= 1.55\ngini = 0.444\nsamples = 6\nvalue = [0, 2, 4]'),
          Text(0.38461538461538464, 0.21428571428571427, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
          Text(0.5384615384615384, 0.21428571428571427, 'x[2] <= 5.45\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]'),
          Text(0.46153846153846156, 0.07142857142857142, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
          Text(0.6153846153846154, 0.07142857142857142, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
          Text(0.8461538461538461, 0.5, 'x[2] <= 4.85\ngini = 0.057\nsamples = 34\nvalue = [0, 1, 33]'),
          Text(0.7692307692307693, 0.35714285714285715, 'x[0] <= 5.95\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
          Text(0.6923076923076923, 0.21428571428571427, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
          Text(0.8461538461538461, 0.21428571428571427, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
          Text(0.9230769230769231, 0.35714285714285715, 'gini = 0.0\nsamples = 31\nvalue = [0, 0, 31]')]
```



```
In [208... user_data=np.array([[3.6,2.23,5.0,1.7],[4.9, 3.0, 1.4, 0.2]])
species_pred=dt.predict(user_data)
species_pred
```

C:\Users\Personal\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
warnings.warn(

```
Out[208]: array(['Iris-versicolor', 'Iris-setosa'], dtype=object)
```

Heart_Disease_Prediction Dataset

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: df=pd.read_csv("Heart_Disease_Prediction.csv")
df
```

```
Out[2]:
```

	index	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
...
265	265	52	1	3	172	199	1	0	162	0	0.5	1	0	7	Absence
266	266	44	1	2	120	263	0	0	173	0	0.0	1	0	7	Absence
267	267	56	0	2	140	294	0	2	153	0	1.3	2	0	3	Absence
268	268	57	1	4	140	192	0	0	148	0	0.4	2	0	6	Absence
269	269	67	1	4	160	286	0	2	108	1	1.5	2	3	3	Presence

270 rows × 15 columns

```
In [3]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   index                                270 non-null    int64
1   Age                                  270 non-null    int64
2   Sex                                  270 non-null    int64
3   Chest pain type                      270 non-null    int64
4   BP                                    270 non-null    int64
5   Cholesterol                          270 non-null    int64
6   FBS over 120                         270 non-null    int64
7   EKG results                          270 non-null    int64
8   Max HR                              270 non-null    int64
9   Exercise angina                      270 non-null    int64
10  ST depression                        270 non-null    float64
11  Slope of ST                          270 non-null    int64
12  Number of vessels fluro              270 non-null    int64
13  Thallium                             270 non-null    int64
14  Heart Disease                        270 non-null    object
dtypes: float64(1), int64(13), object(1)
memory usage: 31.8+ KB

```

In [4]: `df.isnull()`

Out[4]:

	index	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
265	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
266	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
267	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
268	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
269	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

270 rows × 15 columns

In [5]: `df.isnull().sum()`

```

Out[5]:
index          0
Age            0
Sex            0
Chest pain type 0
BP             0
Cholesterol    0
FBS over 120   0
EKG results    0
Max HR         0
Exercise angina 0
ST depression  0
Slope of ST    0
Number of vessels fluro 0
Thallium       0
Heart Disease  0
dtype: int64

```

In [6]: `df.describe()`

Out[6]:

	index	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000
mean	134.500000	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630
std	78.086491	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952
min	0.000000	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000
25%	67.250000	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000
50%	134.500000	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000
75%	201.750000	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000
max	269.000000	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000

In [7]: `X=df.iloc[:,1:14]`
`Y=df.iloc[:, -1]`

In [8]: `X`

Out[8]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3
...
265	52	1	3	172	199	1	0	162	0	0.5	1	0	7
266	44	1	2	120	263	0	0	173	0	0.0	1	0	7
267	56	0	2	140	294	0	2	153	0	1.3	2	0	3
268	57	1	4	140	192	0	0	148	0	0.4	2	0	6
269	67	1	4	160	286	0	2	108	1	1.5	2	3	3

270 rows × 13 columns

In [9]: `Y`

Out[9]:

```

0      Presence
1      Absence
2      Presence
3      Absence
4      Absence
...
265     Absence
266     Absence
267     Absence
268     Absence
269     Presence
Name: Heart Disease, Length: 270, dtype: object

```

In [10]: `from sklearn.model_selection import train_test_split`

In [11]: `X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=42)`

In [12]: `X_train`

Out[12]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
182	41	1	4	110	172	0	2	158	0	0.0	1	0	7
73	76	0	3	140	197	0	1	116	0	1.1	2	0	3
18	64	1	1	110	211	0	2	144	1	1.8	2	0	3
101	67	1	4	120	237	0	0	71	0	1.0	2	0	3
195	55	0	2	135	250	0	2	161	0	1.4	2	0	3
...
20	67	1	4	120	229	0	2	129	1	2.6	2	2	7
188	62	0	4	140	394	0	2	157	0	1.2	2	0	3
71	57	0	4	120	354	0	0	163	1	0.6	1	0	3
106	51	1	3	100	222	0	0	143	1	1.2	2	0	3
102	49	0	4	130	269	0	0	163	0	0.0	1	0	3

202 rows × 13 columns

In [13]: `X_train.shape`

Out[13]: (202, 13)

In [14]: `X_test`

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium
30	57	1	3	128	229	0	2	150	0	0.4	2	1	7
116	46	1	4	120	249	0	2	144	0	0.8	1	0	7
79	56	1	2	120	236	0	0	178	0	0.8	1	0	3
127	52	0	3	136	196	0	2	169	0	0.1	2	0	3
196	58	0	4	100	248	0	2	122	0	1.0	2	0	3
...
10	59	1	4	135	234	0	0	161	0	0.5	2	0	7
233	52	1	4	128	255	0	0	161	1	0.0	1	1	7
66	51	1	4	140	261	0	2	186	1	0.0	1	0	3
253	51	1	3	110	175	0	0	123	0	0.6	1	0	3
225	41	1	2	135	203	0	0	132	0	0.0	2	0	6

68 rows × 13 columns

In [15]: `X_test.shape`

Out[15]: (68, 13)

In [16]: `Y_train`

Out[16]: 182 Presence
73 Absence
18 Absence
101 Presence
195 Absence
...
20 Presence
188 Absence
71 Absence
106 Absence
102 Absence
Name: Heart Disease, Length: 202, dtype: object

In [17]: `Y_train.shape`

Out[17]: (202,)

```
In [18]: Y_test
```

```
Out[18]: 30      Presence
          116     Presence
          79      Absence
          127     Absence
          196     Absence
          ...
          10      Absence
          233     Presence
          66      Absence
          253     Absence
          225     Absence
          Name: Heart Disease, Length: 68, dtype: object
```

```
In [19]: Y_test.shape
```

```
Out[19]: (68,)
```

```
In [20]: from sklearn.tree import DecisionTreeClassifier
```

```
In [21]: dt=DecisionTreeClassifier(criterion="gini")
```

```
In [22]: dt.fit(X_train,Y_train)
```

```
Out[22]: ▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [23]: Y_pred=dt.predict(X_test)
```

```
In [24]: Y_pred
```

```
Out[24]: array(['Absence', 'Presence', 'Absence', 'Absence', 'Presence',
                'Presence', 'Absence', 'Absence', 'Absence', 'Absence', 'Presence',
                'Absence', 'Presence', 'Presence', 'Presence', 'Absence',
                'Presence', 'Presence', 'Presence', 'Absence', 'Presence',
                'Absence', 'Absence', 'Absence', 'Presence', 'Presence', 'Absence',
                'Presence', 'Absence', 'Absence', 'Absence', 'Absence', 'Presence',
                'Absence', 'Presence', 'Presence', 'Presence', 'Absence',
                'Presence', 'Absence', 'Absence', 'Absence', 'Presence', 'Absence',
                'Absence', 'Absence', 'Absence', 'Absence', 'Absence', 'Absence',
                'Presence', 'Presence', 'Absence', 'Presence', 'Absence',
                'Presence', 'Absence', 'Absence', 'Absence', 'Absence', 'Presence',
                'Absence', 'Presence', 'Absence', 'Absence', 'Absence', 'Absence',
                'Absence', 'Presence', 'Presence', 'Presence', 'Presence', 'Absence',
                'Absence', 'Presence'], dtype=object)
```

```
In [25]: dt.score(X_test,Y_test)
```

Out[25]: 0.75

```
In [26]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
```

```
In [27]: accuracy_score(Y_test,Y_pred)
```

```
Out[27]: 0.75
```

```
In [28]: precision_score(Y_test, Y_pred, average='micro')
```

```
Out[28]: 0.75
```

```
In [29]: recall_score(Y_test, Y_pred, average='micro')
```

```
Out[29]: 0.75
```

```
In [30]: f1_score(Y_test, Y_pred, average='micro')
```

```
Out[30]: 0.75
```

```
In [31]: confusion_matrix(Y_test, Y_pred)
```

```
Out[31]: array([[31, 9],
               [ 8, 20]], dtype=int64)
```

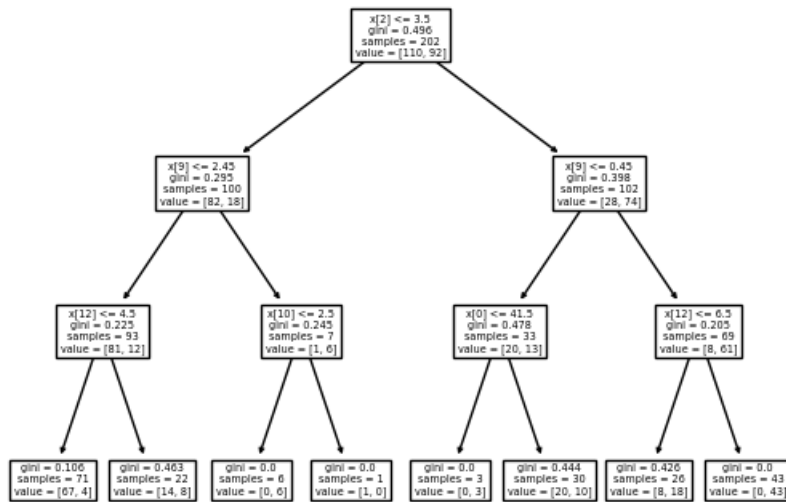
```
In [32]: from sklearn import tree
```

```
In [41]: clf=tree.DecisionTreeClassifier(criterion="gini",max_depth=3)
```

```
In [42]: clf=clf.fit(X_train,Y_train)
```

```
In [43]: tree.plot_tree(clf)
```

```
Out[43]: [Text(0.5, 0.875, 'x[2] <= 3.5\ngini = 0.496\nsamples = 202\nvalue = [110, 92]'),
Text(0.25, 0.625, 'x[9] <= 2.45\ngini = 0.295\nsamples = 100\nvalue = [82, 18]'),
Text(0.125, 0.375, 'x[12] <= 4.5\ngini = 0.225\nsamples = 93\nvalue = [81, 12]'),
Text(0.0625, 0.125, 'gini = 0.106\nsamples = 71\nvalue = [67, 4]'),
Text(0.1875, 0.125, 'gini = 0.463\nsamples = 22\nvalue = [14, 8]'),
Text(0.375, 0.375, 'x[10] <= 2.5\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),
Text(0.3125, 0.125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
Text(0.4375, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.75, 0.625, 'x[9] <= 0.45\ngini = 0.398\nsamples = 102\nvalue = [28, 74]'),
Text(0.625, 0.375, 'x[0] <= 41.5\ngini = 0.478\nsamples = 33\nvalue = [20, 13]'),
Text(0.5625, 0.125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.6875, 0.125, 'gini = 0.444\nsamples = 30\nvalue = [20, 10]'),
Text(0.875, 0.375, 'x[12] <= 6.5\ngini = 0.205\nsamples = 69\nvalue = [8, 61]'),
Text(0.8125, 0.125, 'gini = 0.426\nsamples = 26\nvalue = [8, 18]'),
Text(0.9375, 0.125, 'gini = 0.0\nsamples = 43\nvalue = [0, 43]')]
```



```
In [44]: user_data=np.array([[59,1,1,170,288,0,2,159,0,0.2,2,0,7],
                             [53,1,4,142,226,0,2,111,1,0,1,0,7]])
disease_pred=dt.predict(user_data)
disease_pred
```

```
C:\Users\Personal\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

```
Out[44]: array(['Presence', 'Absence'], dtype=object)
```