DECISION TREE

In [45]:

```
import pandas as pd
import numpy as np

raw_data = pd.read_csv('kyphosis.csv')
print(raw_data.columns)
print(raw_data.head())
```

```
Index(['rownames', 'Kyphosis', 'Age', 'Number', 'Start'], dtype='object')
   rownames Kyphosis  Age  Number  Start
0         1   absent   71       3      5
1         2   absent  158       3     14
2         3  present  128       4      5
3         4   absent    2       5      1
4         5   absent    1       4     15
```

In [46]:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

x = raw_data.drop('Kyphosis', axis = 1)
y = raw_data['Kyphosis']
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.3, random_state = 42)
```

In [47]:

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

ypred = model.predict(x_test)
```

model evaluation

In [48]:

```
# accuracy
print(f'Accuracy : {accuracy_score(y_test, ypred)}')
```

```
Accuracy : 0.76
```

In [49]:

```
# confusion matrix
print(f'Confusion matrix : \n {confusion_matrix(y_test, ypred)}')
```

```
Confusion matrix : 
 [[18  1]
 [ 5  1]]
```

In [50]:

```
# classification report
print(f'Classification Report : \n {classification_report(y_test, ypred)}')
```

```
Classification Report : 
               precision    recall  f1-score   support

      absent       0.78      0.95      0.86        19
     present       0.50      0.17      0.25         6

    accuracy                           0.76        25
   macro avg       0.64      0.56      0.55        25
weighted avg       0.71      0.76      0.71        25
```

Random forest

In [51]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [52]:

```
rf_class = RandomForestClassifier(max_depth = 10, min_samples_split= 5, min_samples_leaf = 2)
rf_class.fit(x_train, y_train)

y_pred = rf_class.predict(x_test)
```

In [53]:

```
# accuracy
print(f'Accuracy : {accuracy_score(y_test, y_pred)}')
```

```
Accuracy : 0.8
```

In [54]:

```
# confusion matrix
print(f'Confusion matrix : \n {confusion_matrix(y_test, y_pred)}')
```

```
Confusion matrix : 
 [[19  0]
 [ 5  1]]
```

In [55]:

```
# classification report
print(f'Classification Report : \n {classification_report(y_test, y_pred)}')
```

```
Classification Report : 
               precision    recall  f1-score   support

      absent       0.79      1.00      0.88        19
     present       1.00      0.17      0.29         6

    accuracy                           0.80        25
   macro avg       0.90      0.58      0.58        25
weighted avg       0.84      0.80      0.74        25
```

In [ ]:

In [ ]: