In [2]:

```
#DECISION TREE FOR BREAST CANCER DATASET
```

In [40]:

```
#Decision tree
import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.tree import plot_tree
```

In [46]:

```
cdata = pd.read_csv("C:\\Users\\G.Madhu mitha\\Downloads\\breast_cancer.csv")
cdata
```

Out[46]:

	id_number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin
0	1000025	5	1	1	1	2	1	3
1	1002945	5	4	4	5	7	10	3
2	1015425	3	1	1	1	2	2	3
3	1016277	6	8	8	1	3	4	3
4	1017023	4	1	1	3	2	1	3
...	...	...	...	...	...	...	...	...
694	776715	3	1	1	1	3	2	1
695	841769	2	1	1	1	2	1	1
696	888820	5	10	10	3	7	3	8
697	897471	4	8	6	4	3	4	10
698	897471	4	8	8	5	4	5	10

699 rows × 11 columns

In [49]:

```
shape=cdata.shape
shape
```

Out[49]:

```
(699, 11)
```

In [28]:

```
#Splitting the data
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=42)
```

In [29]:

```
X_train
```

Out[29]:

```
array([[1.374e+01, 1.791e+01, 8.812e+01, ..., 6.019e-02, 2.350e-01,
        7.014e-02],
       [1.337e+01, 1.639e+01, 8.610e+01, ..., 8.978e-02, 2.048e-01,
        7.628e-02],
       [1.469e+01, 1.398e+01, 9.822e+01, ..., 1.108e-01, 2.827e-01,
        9.208e-02],
       ...,
       [1.429e+01, 1.682e+01, 9.030e+01, ..., 3.333e-02, 2.458e-01,
        6.120e-02],
       [1.398e+01, 1.962e+01, 9.112e+01, ..., 1.827e-01, 3.179e-01,
        1.055e-01],
       [1.218e+01, 2.052e+01, 7.722e+01, ..., 7.431e-02, 2.694e-01,
        6.878e-02]])
```

In [30]:

```
X_test
```

Out[30]:

```
array([[1.247e+01, 1.860e+01, 8.109e+01, ..., 1.015e-01, 3.014e-01,
        8.750e-02],
       [1.894e+01, 2.131e+01, 1.236e+02, ..., 1.789e-01, 2.551e-01,
        6.589e-02],
       [1.546e+01, 1.948e+01, 1.017e+02, ..., 1.514e-01, 2.837e-01,
        8.019e-02],
       ...,
       [9.904e+00, 1.806e+01, 6.460e+01, ..., 9.910e-02, 2.614e-01,
        1.162e-01],
       [1.382e+01, 2.449e+01, 9.233e+01, ..., 1.521e-01, 3.651e-01,
        1.183e-01],
       [1.289e+01, 1.411e+01, 8.495e+01, ..., 1.561e-01, 2.639e-01,
        1.178e-01]])
```

In [31]:

```
y_train
```

Out[31]:

```
array([1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1,
       0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1,
       0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0,
       1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1,
       1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       0, 1])
```

In [32]:

```
y_test
```

Out[32]:

```
array([1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1])
```

In [33]:

```
 = DecisionTreeClassifier(criterion='gini', max_depth=4, random_state=42)
```

In [34]:

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

Out[34]:

```
DecisionTreeClassifier(max_depth=4, random_state=42)
```

In [35]:

```
path = model.cost_complexity_pruning_path(X_train, y_train)
ccp_alphas, impurities = path.ccp_alphas, path.impurities
trees = []
for ccp_alpha in ccp_alphas:
    model = DecisionTreeClassifier(criterion='gini', max_depth=4, random_state=42, ccp_a
    model.fit(X_train, y_train)
    trees.append(model)
```

In [36]:

```
val_accs = []
for model in trees:
    val_preds = model.predict(X_test)
    val_acc = accuracy_score(y_test, val_preds)
    val_accs.append(val_acc)
best_tree_idx = np.argmax(val_accs)
best_tree = trees[best_tree_idx]
```

In [39]:

```
test_preds = best_tree.predict(X_test)
accuracy = accuracy_score(y_test, test_preds)
print("Accuracy: ",accuracy)
precision = precision_score(y_test, test_preds)
print("Precision: ",precision)
recall = recall_score(y_test, test_preds)
print("Recall: ",recall)
f1 = f1_score(y_test, test_preds)
print("F1 score: ",f1)
```

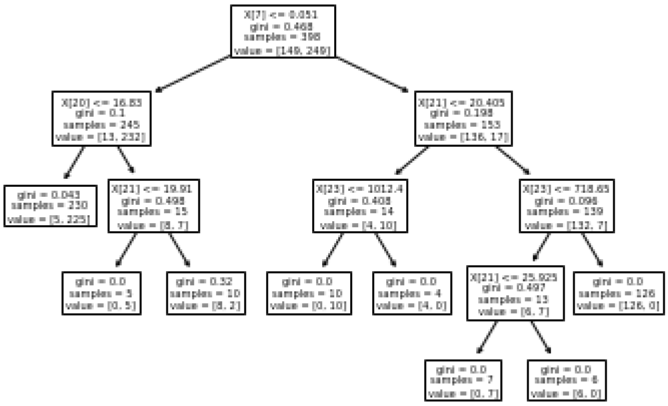
```
Accuracy:  0.9649122807017544
Precision:  0.9722222222222222
Recall:  0.9722222222222222
F1 score:  0.9722222222222222
```

In [38]:

```
plot_tree(best_tree)
```

Out[38]:

```
[Text(141.64615384615385, 195.696, 'X[7] <= 0.051\ngini = 0.468\nsamples
= 398\nvalue = [149, 249]'),
 Text(51.50769230769231, 152.208, 'X[20] <= 16.83\ngini = 0.1\nsamples =
245\nvalue = [13, 232]'),
 Text(25.753846153846155, 108.72, 'gini = 0.043\nsamples = 230\nvalue =
[5, 225]'),
 Text(77.26153846153846, 108.72, 'X[21] <= 19.91\ngini = 0.498\nsamples =
15\nvalue = [8, 7]'),
 Text(51.50769230769231, 65.232, 'gini = 0.0\nsamples = 5\nvalue = [0,
5]'),
 Text(103.01538461538462, 65.232, 'gini = 0.32\nsamples = 10\nvalue = [8,
2]'),
 Text(231.7846153846154, 152.208, 'X[21] <= 20.405\ngini = 0.198\nsamples
= 153\nvalue = [136, 17]'),
 Text(180.27692307692308, 108.72, 'X[23] <= 1012.4\ngini = 0.408\nsamples
= 14\nvalue = [4, 10]'),
 Text(154.52307692307693, 65.232, 'gini = 0.0\nsamples = 10\nvalue = [0,
10]'),
 Text(206.03076923076924, 65.232, 'gini = 0.0\nsamples = 4\nvalue = [4,
0]'),
 Text(283.2923076923077, 108.72, 'X[23] <= 718.65\ngini = 0.096\nsamples
= 139\nvalue = [132, 7]'),
 Text(257.53846153846155, 65.232, 'X[21] <= 25.925\ngini = 0.497\nsamples
= 13\nvalue = [6, 7]'),
 Text(231.7846153846154, 21.744, 'gini = 0.0\nsamples = 7\nvalue = [0,
7]'),
 Text(283.2923076923077, 21.744, 'gini = 0.0\nsamples = 6\nvalue = [6,
0]'),
 Text(309.04615384615386, 65.232, 'gini = 0.0\nsamples = 126\nvalue = [12
6, 0]')]
```



In [ ]:

```
#WORD DONE BY G.MADHUMITHA
```