

In [12]:

```
from sklearn.datasets import make_classification
from sklearn import tree
from sklearn.model_selection import train_test_split
```

In [13]:

```
X, t = make_classification(100, 5, n_classes=2, shuffle=True, random_state=10)
X_train, X_test, t_train, t_test = train_test_split(
    X, t, test_size=0.3, shuffle=True, random_state=1)
```

In [8]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)
```

In [14]:

```
model = tree.DecisionTreeClassifier()
model = model.fit(X_train, t_train)
```

In [15]:

```
predicted_value = model.predict(X_test)
print(predicted_value)
```

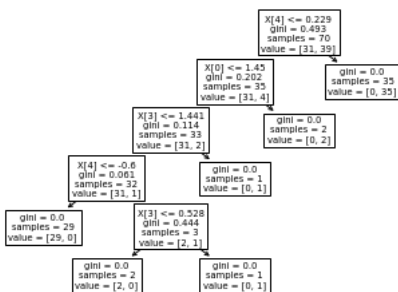
```
[0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 1 1 1 0]
```

In [16]:

```
tree.plot_tree(model)
```

Out[16]:

```
[Text(239.14285714285714, 199.32, 'X[4] <= 0.229\ngini = 0.493\nsamples = 70\nvalue = [31, 39]'),
Text(191.31428571428572, 163.07999999999998, 'X[0] <= 1.45\ngini = 0.202\nsamples = 35\nvalue = [31, 4]'),
Text(143.4857142857143, 126.83999999999999, 'X[3] <= 1.441\ngini = 0.114\nsamples = 33\nvalue = [31, 2]'),
Text(95.65714285714286, 90.6, 'X[4] <= -0.6\ngini = 0.061\nsamples = 32\nvalue = [31, 1]'),
Text(47.82857142857143, 54.359999999999985, 'gini = 0.0\nsamples = 29\nvalue = [29, 0]'),
Text(143.4857142857143, 54.359999999999985, 'X[3] <= 0.528\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(95.65714285714286, 18.119999999999976, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(191.31428571428572, 18.119999999999976, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(191.31428571428572, 90.6, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(239.14285714285714, 126.83999999999999, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(286.9714285714286, 163.07999999999998, 'gini = 0.0\nsamples = 35\nvalue = [0, 35]')]
```



In [17]:

```
zeroes = 0
ones = 0
for i in range(0, len(t_train)):
    if t_train[i] == 0:
        zeroes += 1
    else:
        ones += 1
```

In [18]:

```
print(zeroes)
print(ones)
```

```
31
39
```

In [19]:

```
val = 1 - ((zeroes/70)*(zeroes/70) + (ones/70)*(ones/70))
print("Gini :", val)
```

```
Gini : 0.4934693877551021
```

In [20]:

```
match = 0
UnMatch = 0
```

In [21]:

```
for i in range(30):
    if predicted_value[i] == t_test[i]:
        match += 1
    else:
        UnMatch += 1
```

In [22]:

```
accuracy = match/30
print("Accuracy is: ", accuracy)
```

Accuracy is: 0.8666666666666667

In []: