In [2]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import make_classification
```

In [4]:
```
x,y=make_classification(n_features=5,n_redundant=0,
                        n_informative=5,n_clusters_per_class=1)
```

In [5]:
```
df=pd.DataFrame(x,columns=['col1','col2','col3','col4','col5'])
```

In [6]:
```
df
```

Out[6]:

	col1	col2	col3	col4	col5
0	0.630779	1.448208	0.941771	-1.455243	-0.280989
1	0.410830	-1.556666	0.050209	-1.814057	0.519052
2	-1.540291	-0.206125	3.502919	-0.450205	1.565273
3	2.096366	0.218187	2.130244	-1.871511	1.154351
4	-1.146170	1.392093	0.425851	1.735075	-0.301708
...	...	...	...	...	...
95	-2.972610	-2.147737	-2.247455	-2.976066	3.146960
96	2.240225	-2.756495	3.478257	-2.654683	0.902704
97	-0.222252	-1.108487	0.396636	-1.215891	3.027722
98	2.312145	-0.369648	2.926730	0.726767	1.634659
99	-2.867050	-2.507944	-1.733411	-2.709481	2.555341

100 rows × 5 columns

In [7]:
```
x
```

```
       [-8.89389312e-01, -1.09894653e+00,  9.14821331e-01,
        -8.02322835e-01,  2.32514110e+00],
       [ 2.58677916e+00, -1.80885080e+00,  1.88322947e-02,
        -2.38309634e+00,  1.17562456e+00],
       [ 1.64005748e+00, -3.11438786e+00,  1.96866946e+00,
        -1.60171890e+00,  1.98675492e+00],
       [-1.18752396e+00, -4.70919462e-01, -6.03135493e-01,
        -2.39317511e+00,  2.80858185e+00],
       [ 2.38105429e+00, -1.38968359e+00,  6.41426222e-01,
        -2.06893265e+00,  2.06747337e+00],

       [ 1.11504943e-02, -1.81647457e-01,  2.55353399e-01,
        -6.78925846e-01,  6.45197641e-01],
       [-9.60354977e-01, -1.50279891e+00,  1.04264809e+00,
        -1.02331619e+00, -2.26039990e-01],
       [-2.58341413e+00, -2.25050287e+00, -2.85441107e+00,
        -3.66396355e+00,  1.76497734e+00],
       [-3.01066000e+00, -2.11506289e+00,  1.07307977e+00,
        -1.61082884e+00,  1.50900564e+00],
       [-3.50366889e-01,  1.79176645e+00,  4.27064584e+00,
```

In [9]:
```
df['target']=y
```

In [10]:
```
y
```

Out[10]:
```
array([0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1,
       0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1])
```

In [83]:
```
#from sklearn.model_selection import train_test_split
```

In [85]:
```
#x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [89]:
```
#x_test.shape
```

In [11]:
```
df
```

Out[11]:

	col1	col2	col3	col4	col5	target
0	0.630779	1.448208	0.941771	-1.455243	-0.280989	0
1	0.410830	-1.556666	0.050209	-1.814057	0.519052	0
2	-1.540291	-0.206125	3.502919	-0.450205	1.565273	1
3	2.096366	0.218187	2.130244	-1.871511	1.154351	0
4	-1.146170	1.392093	0.425851	1.735075	-0.301708	0
...	...	...	...	...	...	...
95	-2.972610	-2.147737	-2.247455	-2.976066	3.146960	1
96	2.240225	-2.756495	3.478257	-2.654683	0.902704	0
97	-0.222252	-1.108487	0.396636	-1.215891	3.027722	1
98	2.312145	-0.369648	2.926730	0.726767	1.634659	0
99	-2.867050	-2.507944	-1.733411	-2.709481	2.555341	1

100 rows × 6 columns

In [12]:
```
df.shape
```

Out[12]:
```
(100, 6)
```

In [13]:
```
df.describe()
```

Out[13]:

	col1	col2	col3	col4	col5	target
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	-0.217273	-0.934874	0.867478	-1.232877	1.054425	0.500000
std	1.827953	1.185598	1.435814	1.703789	1.157741	0.502519
min	-4.067846	-3.409669	-2.854411	-5.121569	-1.587915	0.000000
25%	-1.566089	-1.817883	-0.008769	-2.545824	0.441268	0.000000
50%	-0.215917	-1.067414	0.928296	-1.210079	1.006864	0.500000
75%	1.018992	-0.200006	1.848240	-0.083697	1.668817	1.000000
max	5.190239	1.791766	4.270646	2.305320	3.853080	1.000000

In [14]:
```
#rows
```

In [17]:
```
def sample_rows(df,percent):
    return df.sample(int(percent*df.shape[0]),replace=True)
```

In [61]:
```
df1=sample_rows(df,0.2)
```

In [62]:
```
df1
```

Out[62]:

	col1	col2	col3	col4	col5	target
13	-0.620036	-0.256508	3.681323	0.000046	-0.717912	1
25	-1.187524	-0.470919	-0.603135	-2.393175	2.808582	1
35	-1.048709	-1.203271	0.703654	-0.052110	-1.187669	1
61	1.115461	-1.474103	-0.688489	0.344932	1.741754	0
64	0.703844	0.490278	2.287098	-0.233926	0.034431	0
47	-1.435702	-1.967460	1.542792	-0.540756	1.255426	1
69	-1.989563	-0.399381	-0.791770	-2.649196	2.128462	1
98	2.312145	-0.369648	2.926730	0.726767	1.634659	0
49	-1.873927	-1.302895	0.496048	-1.671690	1.133324	1
88	2.658478	0.742565	0.081130	-0.659919	1.093026	0
70	-1.824587	-0.508452	3.237107	-1.552525	-0.299422	0
32	0.923046	-1.738692	1.464596	-2.872023	1.200902	0
89	-0.697854	-0.992987	1.857542	-1.592445	0.627273	1
14	-0.776155	-0.253194	2.810633	1.063000	0.800006	1
73	2.035331	-0.474713	-0.498837	1.242075	1.421553	0
43	0.560771	-0.407857	3.328741	-2.390933	0.211989	0
29	-2.583414	-2.250503	-2.854411	-3.663964	1.764977	1
30	-3.010660	-2.115063	1.073080	-1.610829	1.509006	1
22	-0.889389	-1.098947	0.914821	-0.802323	2.325141	1
57	3.486667	-0.493357	0.498534	0.148575	1.454346	0

In [63]:
```
df1.shape
```

Out[63]:
```
(20, 6)
```

In [64]:
```
df2=sample_rows(df,0.2)
```

In [65]:
```
df2
```

Out[65]:

	col1	col2	col3	col4	col5	target
29	-2.583414	-2.250503	-2.854411	-3.663964	1.764977	1
39	1.813860	-1.457932	1.055468	-0.132909	0.953043	0
99	-2.867050	-2.507944	-1.733411	-2.709481	2.555341	1
92	-2.512780	-0.279227	1.198327	-2.808435	1.775605	1
38	-4.067846	-1.264625	-1.537569	-4.729491	0.973132	1
52	-0.842741	-1.439331	1.808316	0.257283	0.380747	1
49	-1.873927	-1.302895	0.496048	-1.671690	1.133324	1
11	-0.531307	-2.266932	0.259518	-0.391977	0.716866	1
62	1.688753	1.476900	-0.041840	1.135373	1.312912	0
66	0.594057	-2.420382	1.344552	-1.515076	0.526110	0
22	-0.889389	-1.098947	0.914821	-0.802323	2.325141	1
1	0.410830	-1.556666	0.050209	-1.814057	0.519052	0
62	1.688753	1.476900	-0.041840	1.135373	1.312912	0
50	5.190239	-1.108562	-0.287753	1.353989	0.917915	0
30	-3.010660	-2.115063	1.073080	-1.610829	1.509006	1
58	-3.847802	-2.988380	-0.535181	-4.083800	0.509744	1
10	-2.353748	0.215573	0.794943	-4.256807	-0.706140	0
77	-0.127024	-1.356630	1.445276	0.277285	0.761517	1
89	-0.697854	-0.992987	1.857542	-1.592445	0.627273	1
73	2.035331	-0.474713	-0.498837	1.242075	1.421553	0

In [66]:
```
df2.shape
```

Out[66]:
```
(20, 6)
```

In [67]:
```
df3=sample_rows(df,0.2)
```

In [68]:
```
df3
```

Out[68]:

	col1	col2	col3	col4	col5	target
23	2.586779	-1.808851	0.018832	-2.383096	1.175625	0
15	-2.284465	-2.624479	-0.668061	-0.366578	0.570548	1
51	-0.228301	-0.784220	0.283200	-0.616130	1.194642	1
59	1.741651	-2.104256	1.284468	-1.321281	2.927966	0
33	-1.979184	-1.035882	1.845140	-2.253072	1.248839	1
43	0.560771	-0.407857	3.328741	-2.390933	0.211989	0
85	-0.798660	0.080747	2.263777	0.304528	0.859543	1
32	0.923046	-1.738692	1.464596	-2.872023	1.200902	0
83	-2.084350	-1.402551	-2.005406	-3.560831	0.791396	1
98	2.312145	-0.369648	2.926730	0.726767	1.634659	0
54	0.737752	-1.902913	0.077432	-3.061150	0.421945	0
95	-2.972610	-2.147737	-2.247455	-2.976066	3.146960	1
37	-0.770094	-2.176952	-0.185587	-0.477982	-0.750859	1
49	-1.873927	-1.302895	0.496048	-1.671690	1.133324	1
94	0.986836	-3.211069	0.657087	-2.920357	0.606947	0
73	2.035331	-0.474713	-0.498837	1.242075	1.421553	0
96	2.240225	-2.756495	3.478257	-2.654683	0.902704	0
81	-0.336107	-0.765091	1.257872	-0.723729	-0.389066	0
53	2.207093	-1.844981	-0.639568	0.703398	2.573596	0
53	2.207093	-1.844981	-0.639568	0.703398	2.573596	0

In [69]:
```
df3.shape
```

Out[69]:
```
(20, 6)
```

In [70]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [71]:
```
clf1=DecisionTreeClassifier()
clf2=DecisionTreeClassifier()
clf3=DecisionTreeClassifier()
```

In [72]:
```
clf1.fit(df1.iloc[:,0:5],df1.iloc[:,-1])
```

Out[72]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [73]:
```
clf2.fit(df2.iloc[:,0:5],df2.iloc[:,-1])
```

Out[73]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

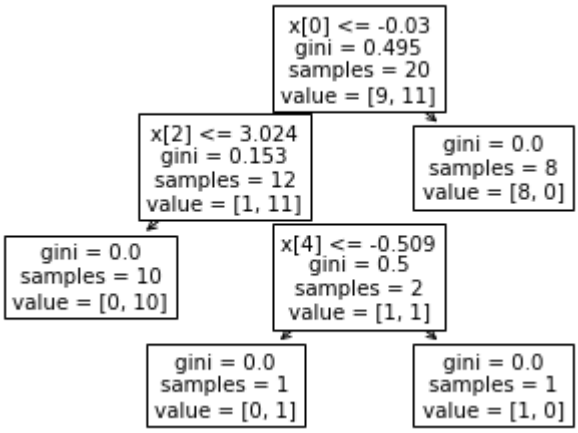
In [74]:
```
clf3.fit(df3.iloc[:,0:5],df3.iloc[:,-1])
```

Out[74]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [75]:
```
from sklearn.tree import plot_tree
```

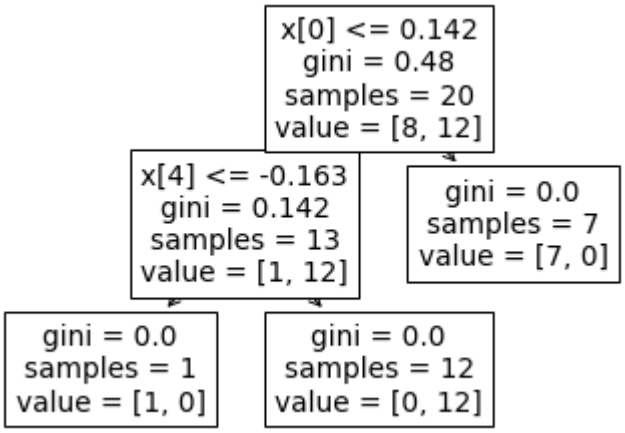
In [76]:
```
plot_tree(clf1)
```

Out[76]:
```
[Text(0.6, 0.875, 'x[0] <= -0.03\ngini = 0.495\nsamples = 20\nvalue = [9, 1
1]'),
 Text(0.4, 0.625, 'x[2] <= 3.024\ngini = 0.153\nsamples = 12\nvalue = [1, 1
1]'),
 Text(0.2, 0.375, 'gini = 0.0\nsamples = 10\nvalue = [0, 10]'),
 Text(0.6, 0.375, 'x[4] <= -0.509\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.4, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8, 0.625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]')]
```



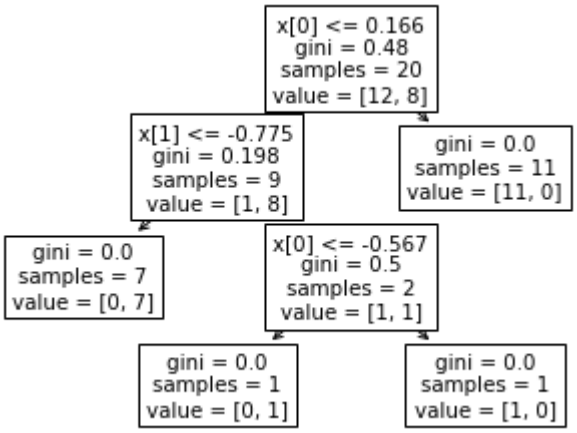
In [77]:
```
plot_tree(clf2)
```

Out[77]:
```
[Text(0.6, 0.8333333333333334, 'x[0] <= 0.142\ngini = 0.48\nsamples = 20\nvalue
= [8, 12]'),
 Text(0.4, 0.5, 'x[4] <= -0.163\ngini = 0.142\nsamples = 13\nvalue = [1, 12]'),
 Text(0.2, 0.16666666666666666, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6, 0.16666666666666666, 'gini = 0.0\nsamples = 12\nvalue = [0, 12]'),
 Text(0.8, 0.5, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]')]
```



In [78]:
```
plot_tree(clf3)
```

Out[78]:
```
[Text(0.6, 0.875, 'x[0] <= 0.166\ngini = 0.48\nsamples = 20\nvalue = [12, 8]'),
 Text(0.4, 0.625, 'x[1] <= -0.775\ngini = 0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(0.2, 0.375, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.6, 0.375, 'x[0] <= -0.567\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.4, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8, 0.625, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]')]
```



In [79]:
```
#predicting the values
```

In [90]:
```
clf1.predict(np.array([-1.048709,-1.203271,0.703654,-0.052110,-1.187669])
             .reshape(1,5))
```

```
C:\Users\91733\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:40
9: UserWarning: X does not have valid feature names, but DecisionTreeClassifier
was fitted with feature names
  warnings.warn(
```

Out[90]:
```
array([1])
```

In [81]:
```
clf2.predict(np.array([-1.048709,-1.203271,0.703654,-0.052110,-1.187669])
             .reshape(1,5))
```

```
C:\Users\91733\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:40
9: UserWarning: X does not have valid feature names, but DecisionTreeClassifier
was fitted with feature names
  warnings.warn(
```

Out[81]:
```
array([0])
```

In [82]:
```
clf3.predict(np.array([-1.048709,-1.203271,0.703654,-0.052110,-1.187669])
             .reshape(1,5))
```

```
C:\Users\91733\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:40
9: UserWarning: X does not have valid feature names, but DecisionTreeClassifier
was fitted with feature names
  warnings.warn(
```

Out[82]:
```
array([1])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: