In [2]:

```
import pandas as pd
import numpy as np
import matplotlib as mpl
```

In [3]:

```
%matplotlib inline
```

In [4]:

```
# read data.csv
df = pd.read_csv('data.csv')
```

In [5]:

```
df
```

Out[5]:

	编号	脾气	身高	身材	经济	年龄差	结果
0	1	差	168	佳	有	小	合适
1	2	好	158	佳	没有	小	合适
2	3	一般	170	良	有	小	不合适
3	4	一般	160	佳	没有	大	不合适
4	5	一般	162	不好	没有	大	不合适
5	6	差	163	良	有	小	合适
6	7	差	164	佳	有	大	不合适
7	8	好	168	良	没有	小	合适
8	9	好	153	不好	有	大	不合适
9	10	差	160	良	有	大	合适

In [6]:

```
df.head()
```

Out[6]:

	编号	脾气	身高	身材	经济	年龄差	结果
0	1	差	168	佳	有	小	合适
1	2	好	158	佳	没有	小	合适
2	3	一般	170	良	有	小	不合适
3	4	一般	160	佳	没有	大	不合适
4	5	一般	162	不好	没有	大	不合适

请分别采用 （a）信息增益ID3算法； （b）信息增益率C4.5算法； （c）基尼系数CART算法； 判断“某女朋友”是否合适？ 该“某女朋友”特征为：脾气差、身高163cm之上、身材良、有经济条件、年龄差大。

In [7]:

```
df.head
```

Out[7]:

```
<bound method NDFrame.head of    编号  脾气   身高  身材  经济 年龄差   结果
0   1   差  168   佳   有   小   合适
1   2   好  158   佳  没有   小   合适
2   3  一般  170   良   有   小  不合适
3   4  一般  160   佳  没有   大  不合适
4   5  一般  162  不好  没有   大  不合适
5   6   差  163   良   有   小   合适
6   7   差  164   佳   有   大  不合适
7   8   好  168   良  没有   小   合适
8   9   好  153  不好   有   大  不合适
9  10   差  160   良   有   大   合适>
```

In [8]:

```
import math
import sklearn
```

In [9]:

```
# read data.csv
df = pd.read_csv('data.csv')
df
```

Out[9]:

	编号	脾气	身高	身材	经济	年龄差	结果
0	1	差	168	佳	有	小	合适
1	2	好	158	佳	没有	小	合适
2	3	一般	170	良	有	小	不合适
3	4	一般	160	佳	没有	大	不合适
4	5	一般	162	不好	没有	大	不合适
5	6	差	163	良	有	小	合适
6	7	差	164	佳	有	大	不合适
7	8	好	168	良	没有	小	合适
8	9	好	153	不好	有	大	不合适
9	10	差	160	良	有	大	合适

In [10]:

```
df['脾气']=df['脾气'].map({'好':2,'一般':1,'差':0})
# 大于163 为1 小于等于163 为0
df['身高']=df['身高'].map(lambda x:1 if x>163 else 0)
df['身材']=df['身材'].map({'佳':2,'良':1,'不好':0})
df['经济']=df['经济'].map({'有':1,'没有':0})
df['年龄差']=df['年龄差'].map({'小':1,'大':0})

df['结果']=df['结果'].map({'合适':1,'不合适':0})
```

In [11]:

```
# X 为第2-6列
x = df.iloc[:, 1:6]
print(x)
y = df.iloc[:,6]
print(y)

predict_df = pd.DataFrame([[0, 1, 1, 1, 0]], columns=['脾气', '身高', '身材', '经济', '年龄差'])
print(predict_df)
```

```
   脾气  身高  身材  经济  年龄差
0   0   1   2   1    1
1   2   0   2   0    1
2   1   1   1   1    1
3   1   0   2   0    0
4   1   0   0   0    0
5   0   0   1   1    1
6   0   1   2   1    0
7   2   1   1   0    1
8   2   0   0   1    0
9   0   0   1   1    0
0    1
1    1
2    0
3    0
4    0
5    1
6    0
7    1
8    0
9    1
Name: 结果, dtype: int64
   脾气  身高  身材  经济  年龄差
0   0   1   1   1    0
```

ID3 Algorithm

In [12]:

```
from sklearn.tree import DecisionTreeClassifier
id3 = DecisionTreeClassifier(criterion='entropy')
id3 = id3.fit(x, y)

# 该“某女朋友”特征为：脾气差、身高163cm之上、身材良、有经济条件、年龄差大。
# 请问该女朋友是否合适？

print(id3.predict(predict_df))
```

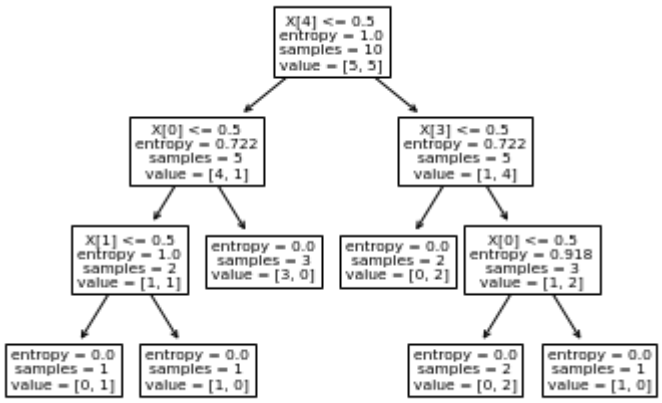
```
[0]
```

In [13]:

```
from sklearn import tree
tree.plot_tree(id3)
```

Out[13]:

```
[Text(0.5, 0.875, 'X[4] <= 0.5\nentropy = 1.0\nsamples = 10\nvalue = [5, 5]'),
 Text(0.3, 0.625, 'X[0] <= 0.5\nentropy = 0.722\nsamples = 5\nvalue = [4, 1]'),
 Text(0.2, 0.375, 'X[1] <= 0.5\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
 Text(0.1, 0.125, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3, 0.125, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4, 0.375, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.7, 0.625, 'X[3] <= 0.5\nentropy = 0.722\nsamples = 5\nvalue = [1, 4]'),
 Text(0.6, 0.375, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8, 0.375, 'X[0] <= 0.5\nentropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.7, 0.125, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9, 0.125, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]')]
```



C4.5 Algorithm

In [14]:

```
# 信息增益率C4.5算法;

from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()
model = model.fit(x, y)

print(model.predict(predict_df))
```

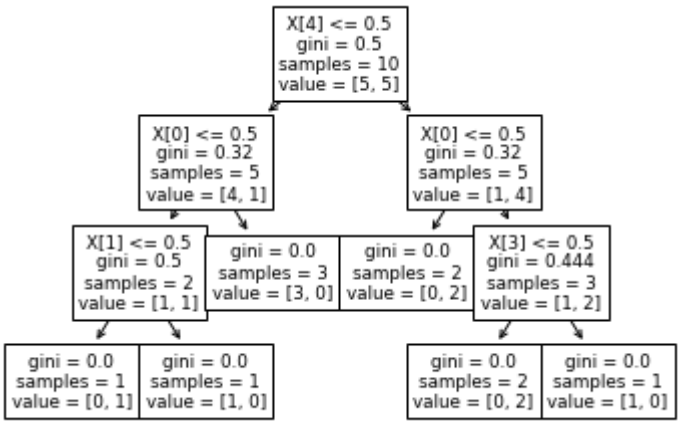
[0]

In [17]:

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

Out[17]:

```
[Text(0.5, 0.875, 'X[4] <= 0.5\ngini = 0.5\nsamples = 10\nvalue = [5, 5]'),
 Text(0.3, 0.625, 'X[0] <= 0.5\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.2, 0.375, 'X[1] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.1, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4, 0.375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.7, 0.625, 'X[0] <= 0.5\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(0.6, 0.375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8, 0.375, 'X[3] <= 0.5\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.7, 0.125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]')]
```



CART Algorithm

In [15]:

```
gini = tree.DecisionTreeClassifier(criterion='gini')
gini = gini.fit(x, y)

print(gini.predict(predict_df))
```

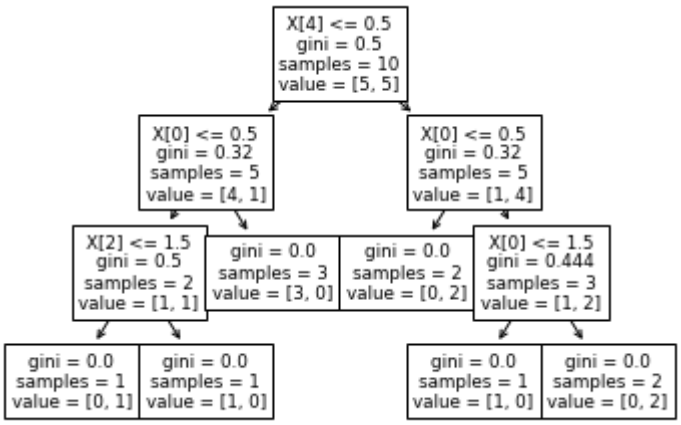
```
[1]
```

In [16]:

```
tree.plot_tree(gini)
```

Out[16]:

```
[Text(0.5, 0.875, 'X[4] <= 0.5\ngini = 0.5\nsamples = 10\nvalue = [5, 5]'),
 Text(0.3, 0.625, 'X[0] <= 0.5\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.2, 0.375, 'X[2] <= 1.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.1, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4, 0.375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.7, 0.625, 'X[0] <= 0.5\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(0.6, 0.375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8, 0.375, 'X[0] <= 1.5\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.7, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9, 0.125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]')]
```



In [ ]:

In [ ]:

In [ ]: