In [1]:
```
import pandas as pd
```

In [2]:
```
data=pd.read_csv('salary.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	company	job	degree	salary
0	google	salesexecutive	bachelor	0
1	google	salesexecutive	master	0
2	google	businessmanager	bachelor	1
3	google	businessmanager	master	1
4	google	computerprogrammer	bachelor	0

In [4]:
```
from sklearn.preprocessing import LabelEncoder
```

In [5]:
```
LE=LabelEncoder()
```

In [6]:
```
data.head()
```

Out[6]:

	company	job	degree	salary
0	google	salesexecutive	bachelor	0
1	google	salesexecutive	master	0
2	google	businessmanager	bachelor	1
3	google	businessmanager	master	1
4	google	computerprogrammer	bachelor	0

In [15]:
```
x=data[['company','job','degree']]
```

In [16]:
```
y=data[['salary']]
```

In [17]:
```
data["company"]=LE.fit_transform(data["company"])
data["degree"]=LE.fit_transform(data["degree"])
data["job"]=LE.fit_transform(data["job"])
```

In [18]:
```
data
```

Out[18]:

	company	job	degree	salary
0	2	2	0	0
1	2	2	1	0
2	2	0	0	1
3	2	0	1	1
4	2	1	0	0
5	2	1	1	1
6	0	2	0	0
7	0	0	0	0
8	0	0	1	0
9	0	2	1	1
10	1	2	0	1
11	1	0	0	1
12	1	0	1	1
13	1	2	0	1
14	1	2	1	1
15	1	0	0	1
16	1	0	0	1
17	1	1	1	1

In [19]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [20]:
```
model=DecisionTreeClassifier()
```

In [21]:
```
model.fit(x,y)
```

Out[21]:
```
DecisionTreeClassifier()
```

In [22]:
```
model.predict([[1,2,1]])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
s not have valid feature names, but DecisionTreeClassifier was fitted with feature
names
  warnings.warn(
```

Out[22]:
```
array([1], dtype=int64)
```

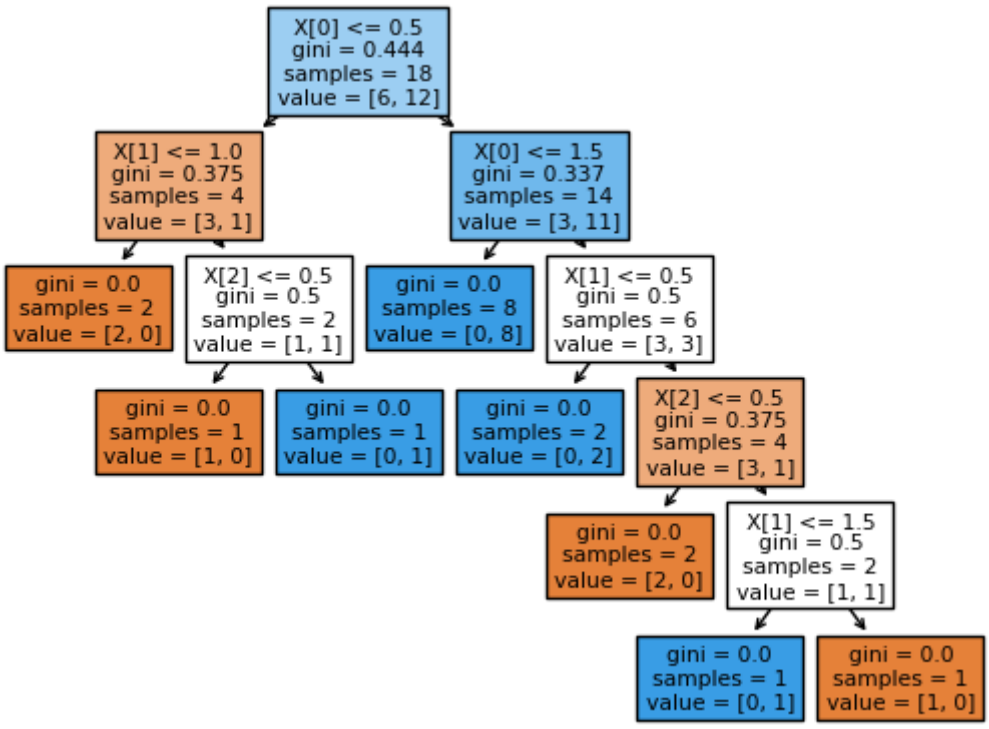
In [23]:
```
import matplotlib.pyplot as plt
% matplotlib inline
```

```
UsageError: Line magic function `%` not found.
```

In [37]:
```
from sklearn import tree
```

In [40]:
```
tree.plot_tree(model,filled=True)
```

Out[40]:
```
[Text(0.36363636363636365, 0.9166666666666666, 'X[0] <= 0.5\ngini = 0.444\nsamples
= 18\nvalue = [6, 12]'),
 Text(0.18181818181818182, 0.75, 'X[1] <= 1.0\ngini = 0.375\nsamples = 4\nvalue =
[3, 1]'),
 Text(0.09090909090909091, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.2727272727272727, 0.5833333333333334, 'X[2] <= 0.5\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.18181818181818182, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.36363636363636365, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5454545454545454, 0.75, 'X[0] <= 1.5\ngini = 0.337\nsamples = 14\nvalue =
[3, 11]'),
 Text(0.45454545454545453, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]'),
 Text(0.6363636363636364, 0.5833333333333334, 'X[1] <= 0.5\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
 Text(0.5454545454545454, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.7272727272727273, 0.4166666666666667, 'X[2] <= 0.5\ngini = 0.375\nsamples
= 4\nvalue = [3, 1]'),
 Text(0.6363636363636364, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8181818181818182, 0.25, 'X[1] <= 1.5\ngini = 0.5\nsamples = 2\nvalue = [1,
1]'),
 Text(0.7272727272727273, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9090909090909091, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]')]
```



using test_train

In [50]:
```
from sklearn.model_selection import train_test_split
```

In [54]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [55]:
```
model.fit(x_train,y_train)
```

Out[55]:
```
DecisionTreeClassifier()
```

In [60]:
```
y_pred=model.predict(x_test)
```

In [61]:
```
y_pred
```

Out[61]:
```
array([1, 1, 1, 0, 1, 0], dtype=int64)
```

In [62]:
```
from sklearn.metrics import accuracy_score
```

In [71]:
```
score=accuracy_score(y_pred,y_test)
```

In [72]:
```
score
```

Out[72]:
```
0.8333333333333334
```

In [73]:
```
from sklearn.metrics import mean_squared_error
```

In [74]:
```
mean_squared_error(y_pred,y_test)
```

Out[74]:
```
0.16666666666666666
```

In [ ]:
```

```