Scikit-learnを使用してPythonで性別分類モデルを構築する

この問題では、人の[身長(cm)、体重(kg)、靴のサイズ(EU size)]に関するデータがあります。 このデータを使用して、新しい人の性別を予測する分類器を作成します。 分類は複数の分類モデルを使用して行う必要があり、最も正確な分類モデルを特定する必要があります。

In [1]:
```
X = [[181, 80, 44], [177, 70, 43], [160, 60, 38],
     [154, 54, 37],[166, 65, 40], [190, 90, 47], [175, 64, 39],
     [177, 70, 40], [159, 55, 37], [171, 75, 42],
     [181, 85, 43], [168, 75, 41], [168, 77, 41]]

Y = ['male', 'male', 'female', 'female', 'male', 'male',
    'female','female','female', 'male', 'male',
     'female', 'female']
```

In [2]:
```
import pandas as pd

df = pd.DataFrame(X,columns=['Height(cm)','Weight(kg)','Shoe size(EU)'])
df
```

Out[2]:

	Height(cm)	Weight(kg)	Shoe size(EU)
0	181	80	44
1	177	70	43
2	160	60	38
3	154	54	37
4	166	65	40
5	190	90	47
6	175	64	39
7	177	70	40
8	159	55	37
9	171	75	42
10	181	85	43
11	168	75	41
12	168	77	41

In [3]:
```
df['Shoe size(cm)']=round(df['Shoe size(EU)']/3*2,1)
df
```

Out[3]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)
0	181	80	44	29.3
1	177	70	43	28.7
2	160	60	38	25.3
3	154	54	37	24.7
4	166	65	40	26.7
5	190	90	47	31.3
6	175	64	39	26.0
7	177	70	40	26.7
8	159	55	37	24.7
9	171	75	42	28.0
10	181	85	43	28.7
11	168	75	41	27.3
12	168	77	41	27.3

In [4]:
```
df['Gender']=Y
df
```

Out[4]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
0	181	80	44	29.3	male
1	177	70	43	28.7	male
2	160	60	38	25.3	female
3	154	54	37	24.7	female
4	166	65	40	26.7	male

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
5	190	90	47	31.3	male
6	175	64	39	26.0	female
7	177	70	40	26.7	female
8	159	55	37	24.7	female
9	171	75	42	28.0	male
10	181	85	43	28.7	male
11	168	75	41	27.3	female
12	168	77	41	27.3	female

In [5]:
```
test_data = [[190, 70, 43],[154, 75, 38],[181,65,40]]
test_labels = ['male','female','male']
```

In [6]:
```
df_test = pd.DataFrame(test_data,columns=['Height(cm)','Weight(kg)','Shoe size(EU)'])
df_test
```

Out[6]:

	Height(cm)	Weight(kg)	Shoe size(EU)
0	190	70	43
1	154	75	38
2	181	65	40

In [7]:
```
df_test['Shoe size(cm)']=round(df_test['Shoe size(EU)']/3*2,1)
df_test
```

Out[7]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)
0	190	70	43	28.7
1	154	75	38	25.3
2	181	65	40	26.7

分類器の作成

- 決定木
- ランダムフォレスト

[身長(cm)、体重(kg)、靴のサイズ(EU size)]は特徴量とも言います。

- ジェンダーラベル：教師データ

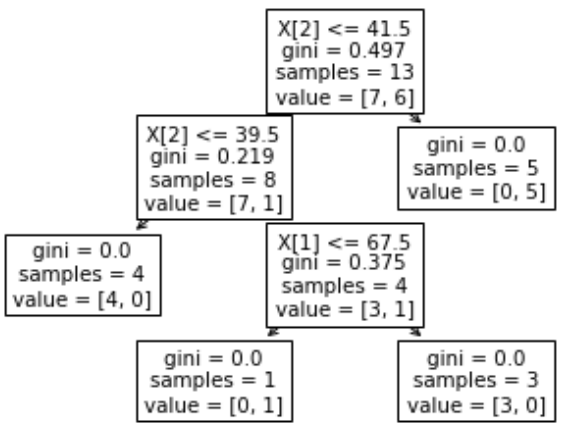
In [8]:
```
from sklearn import tree
```

In [9]:
```
#  学習データを用いて決定木モデルの作成  DecisionTreeClassifier
dtc_clf = tree.DecisionTreeClassifier()
dtc_clf = dtc_clf.fit(X,Y)
print(dtc_clf.get_params())
```

```
{'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impu
rity_decrease': 0.0, 'min_impurity_split': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'presor
t': 'deprecated', 'random_state': None, 'splitter': 'best'}
```

In [11]:
```
tree.plot_tree(dtc_clf)
```

Out[11]:
```
[Text(200.88000000000002, 190.26, 'X[2] <= 41.5¥ngini = 0.497¥nsamples = 13¥nvalue = [7, 6]'),
 Text(133.92000000000002, 135.9, 'X[2] <= 39.5¥ngini = 0.219¥nsamples = 8¥nvalue = [7, 1]'),
 Text(66.96000000000001, 81.53999999999999, 'gini = 0.0¥nsamples = 4¥nvalue = [4, 0]'),
 Text(200.88000000000002, 81.53999999999999, 'X[1] <= 67.5¥ngini = 0.375¥nsamples = 4¥nvalue = [3, 1]'),
 Text(133.92000000000002, 27.180000000000007, 'gini = 0.0¥nsamples = 1¥nvalue = [0, 1]'),
 Text(267.84000000000003, 27.180000000000007, 'gini = 0.0¥nsamples = 3¥nvalue = [3, 0]'),
 Text(267.84000000000003, 135.9, 'gini = 0.0¥nsamples = 5¥nvalue = [0, 5]')]
```



- 決定木のルールと決定木プロットは上記の様になります。
- $X[2]$とは靴のサイズです。この特徴量は一番情報量が持ちます。木のトップになります。
- $X[2] > 41.5$ならば、'male'という結果が決定されます（5人）
- $X[2] <= 39.5$ならば、'female'は4人という結果が決定されます
- $X[1]$とは体重です。$X[2] > 39.5$かつ$X[1] <= 67.5kg$であれば、'male'は一人いる、そうではないは'female'は3人いる

In [13]:
```
df[df['Shoe size(EU)']>41.5]
```

Out[13]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
0	181	80	44	29.3	male
1	177	70	43	28.7	male
5	190	90	47	31.3	male
9	171	75	42	28.0	male
10	181	85	43	28.7	male

In [16]:
```
df[df['Shoe size(EU)']<=41.5]
```

Out[16]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
2	160	60	38	25.3	female
3	154	54	37	24.7	female
4	166	65	40	26.7	male
6	175	64	39	26.0	female
7	177	70	40	26.7	female
8	159	55	37	24.7	female
11	168	75	41	27.3	female
12	168	77	41	27.3	female

In [18]:
```
df[(df['Shoe size(EU)']<=41.5) & (df['Shoe size(EU)']<=39.5)]
```

Out[18]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
2	160	60	38	25.3	female
3	154	54	37	24.7	female
6	175	64	39	26.0	female
8	159	55	37	24.7	female

In [22]:
```
df[(df['Weight(kg)']<67.5) & (df['Shoe size(EU)']>39.5)]
```

Out[22]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
4	166	65	40	26.7	male

In [26]:
```
df[(df['Weight(kg)']>67.5) & (df['Shoe size(EU)']>39.5) &  (df['Shoe size(EU)']<=41.5)]
```

Out[26]:

	Height(cm)	Weight(kg)	Shoe size(EU)	Shoe size(cm)	Gender
7	177	70	40	26.7	female
11	168	75	41	27.3	female
12	168	77	41	27.3	female

In [27]:
```
# 決定木モデルを用いてtest_dataのラベルを予測する
dtc_prediction = dtc_clf.predict(test_data)
print(dtc_prediction)
```

```
['male' 'female' 'male']
```

In [28]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [29]:
```
#  学習データを用いてランダムフォレストの作成     RandomForestClassifier
rfc_clf = RandomForestClassifier()
rfc_clf.fit(X,Y)
print(rfc_clf.get_params())
```

```
{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_impurity_split': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': None, 'verbose': 0, 'warm_start': False}
```

In [30]:
```
# ランダムフォレストを用いてtest_dataのラベルを予測する
rfc_prediction = rfc_clf.predict(test_data)
print (rfc_prediction)
```

```
['male' 'female' 'female']
```

In [31]:
```
import numpy as np
from sklearn.metrics import accuracy_score
```

In [32]:
```
# 精度スコアの計算  accuracy scores
dtc_tree_acc = accuracy_score(dtc_prediction,test_labels)
rfc_acc = accuracy_score(rfc_prediction,test_labels)

print("決定木の精度：",round(dtc_tree_acc,3))
print("ランダムフォレストの精度：",round(rfc_acc,3))
```

```
決定木の精度： 1.0
ランダムフォレストの精度： 0.667
```

In [ ]:
```

```