In [14]:

```
import pandas as pd
import numpy as np
from sklearn import tree
```

In [15]:

```
df=pd.read_csv("weight_height_dataset.csv")
```

In [16]:

```
df
```

Out[16]:

| | Height(cm) | Weight(kg) | Class |
|---|---|---|---|
| **0** | 171.408421 | 69.037935 | Normal |
| **1** | 153.935688 | 47.797508 | Underweight |
| **2** | 176.573961 | 78.871438 | Overweight |
| **3** | 170.663093 | 70.263714 | Normal |
| **4** | 164.009912 | 68.730922 | Normal |
| **...** | ... | ... | ... |
| **145** | 181.933161 | 85.660306 | Overweight |
| **146** | 166.007758 | 73.997699 | Normal |
| **147** | 158.383396 | 55.464065 | Underweight |
| **148** | 174.596901 | 86.130276 | Overweight |
| **149** | 176.323440 | 89.020962 | Overweight |

150 rows × 3 columns

In [19]:

```
X=df.drop('Class',axis=1)
y=df['Class']
```

In [20]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,y,test_size = 0.2,random_s
```

In [21]:

```
X_test,Y_test
```

Out[21]:

```
(     Height(cm)  Weight(kg)
 73   167.994192   76.505154
 18   156.647261   51.335016
 118  171.333920   70.912901
 78   169.968486   73.902363
 76   171.389720   68.779720
 31   177.740527   83.352176
 64   179.098169   82.629365
 141  174.604070   92.592633
 68   160.648719   66.327005
 82   161.586887   58.771700
 110  168.890980   70.716720
 12   176.299598   89.709293
 36   159.924211   56.153378
 9    179.198932   89.625512
 19   163.490092   56.483632
 56   165.336540   58.206047
 104  162.299134   69.270385
 69   166.183559   63.980637
 55   163.741888   70.113080
 132  179.322071   96.839584
 29   164.582065   52.487563
 127  165.924112   66.632277
 26   164.021128   73.497982
 128  161.806830   62.963657
 131  175.382512   84.698945
 145  181.933161   85.660306
 108  180.538624   84.070203
 143  156.201879   49.355527
 45   164.180631   76.222786
 30   172.341812   78.373656,
 73          Normal
 18     Underweight
 118         Normal
 78          Normal
 76          Normal
 31      Overweight
 64      Overweight
 141     Overweight
 68          Normal
 82     Underweight
 110         Normal
 12      Overweight
 36     Underweight
 9       Overweight
 19     Underweight
 56     Underweight
 104         Normal
 69          Normal
 55          Normal
 132     Overweight
 29     Underweight
 127         Normal
 26          Normal
 128    Underweight
 131     Overweight
 145     Overweight
 108     Overweight
 143    Underweight
 45          Normal
30      Overweight
Name: Class, dtype: object)
```

In [22]:

```
clf = tree.DecisionTreeClassifier(criterion="entropy",max_depth=5, min_samp
clf=clf.fit(X_train,Y_train)
prediction = clf.predict(X_test)
```

In [23]:

```
prediction
```

Out[23]:

```
array(['Normal', 'Underweight', 'Normal', 'Normal', 'Normal',
       'Overweight', 'Overweight', 'Overweight', 'Normal', 'Underweight',
       'Normal', 'Overweight', 'Underweight', 'Overweight', 'Underweight',
       'Underweight', 'Normal', 'Normal', 'Normal', 'Overweight',
       'Underweight', 'Normal', 'Normal', 'Normal', 'Overweight',
       'Overweight', 'Overweight', 'Underweight', 'Normal', 'Normal'],
      dtype=object)
```

In [25]:

```
from sklearn.metrics import accuracy_score
```

In [28]:

```
print("Train data accuracy:",accuracy_score(y_true =Y_train, y_pred=clf.pre
print("Test data accuracy:",accuracy_score(y_true =Y_test, y_pred=predictic
```

```
Train data accuracy: 0.95
Test data accuracy: 0.9333333333333333
```

In [29]:

```
from sklearn import metrics
cf=metrics.confusion_matrix(Y_test,prediction)
cf
```

Out[29]:

```
array([[12,  0,  0],
       [ 1,  9,  0],
       [ 1,  0,  7]], dtype=int64)
```

In [32]:

```
print("Precision",metrics.precision_score(Y_test,prediction,average=None))
```

```
Precision [0.85714286 1.         1.        ]
```

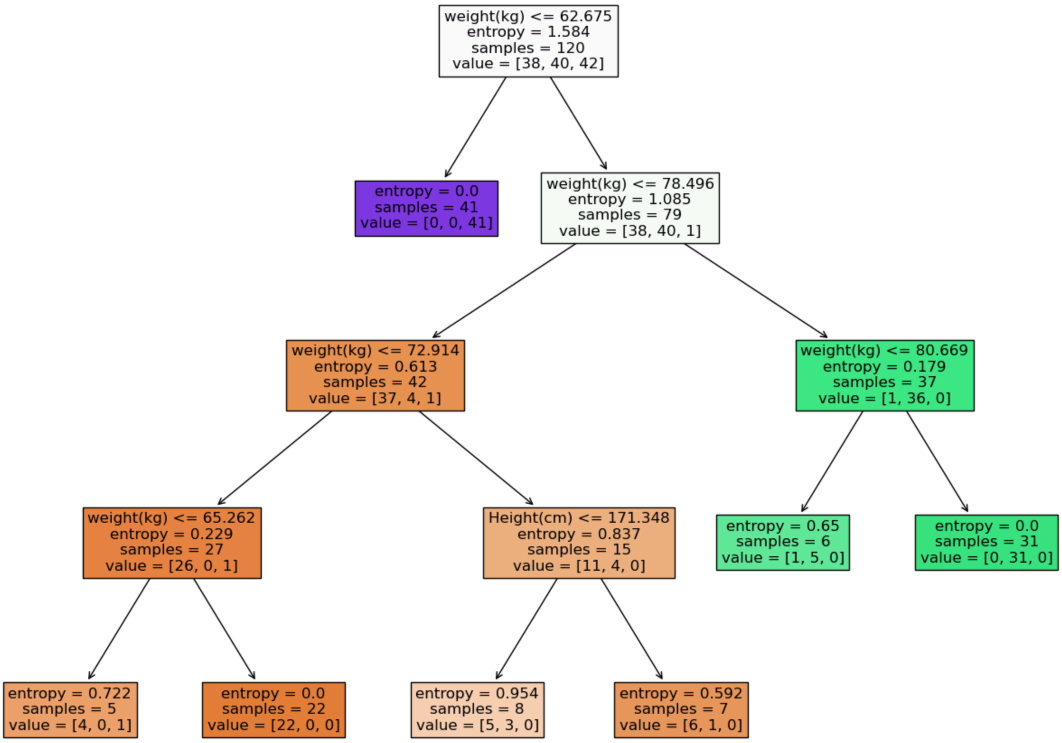
In [33]:

```
print("Recall",metrics.recall_score(Y_test,prediction,average=None))
```

```
Recall [1.    0.9   0.875]
```

In [38]:
```
from sklearn.tree import plot_tree

import matplotlib.pyplot as plt
fig = plt.figure(figsize=(16,12))
a = plot_tree(clf, feature_names=['Height(cm)','weight(kg)'], fontsize=12,
```



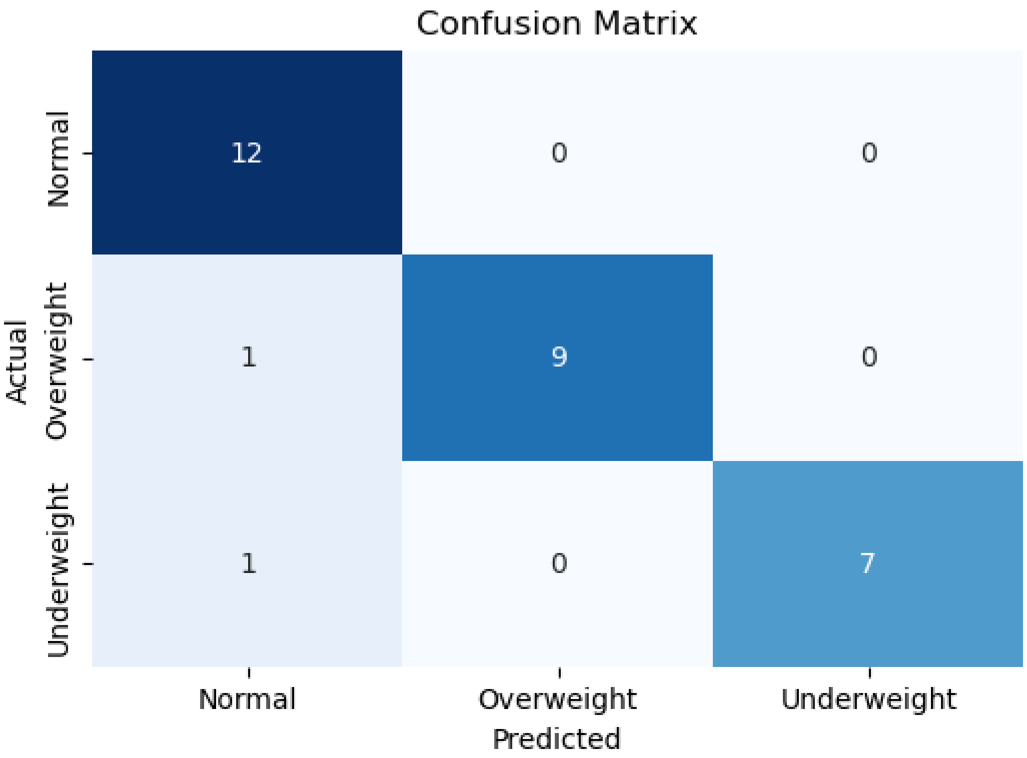
In [39]:
```
import seaborn as sns
```

In [43]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

confusion_matrix = np.array([[12, 0, 0],
                             [1, 9, 0],
                             [1, 0, 7]])

plt.figure(figsize=(6, 4))
sns.heatmap(confusion_matrix, annot=True, cmap='Blues', fmt='d', cbar=False
            xticklabels=['Normal', 'Overweight', 'Underweight'],
            yticklabels=['Normal', 'Overweight', 'Underweight'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



In [44]:

```
clf.classes_
```

Out[44]:

```
array(['Normal', 'Overweight', 'Underweight'], dtype=object)
```

In [ ]: