In [25]:

```
#Load the required libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.naive_bayes import GaussianNB
from scipy.stats import randint

from sklearn.tree import export_graphviz, DecisionTreeClassifier, plot_tree
from IPython.display import Image
import graphviz
```

In [26]:

```
#Load the data
obesity = pd.read_csv('ObesityDataSet.csv')
obesity.head()
```

Out[26]:

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC
0	Female	21.0	1.62	64.0	yes	no	2.0	3.0	Sometimes
1	Female	21.0	1.52	56.0	yes	no	3.0	3.0	Sometimes
2	Male	23.0	1.80	77.0	yes	no	2.0	3.0	Sometimes
3	Male	27.0	1.80	87.0	no	no	3.0	3.0	Sometimes
4	Male	22.0	1.78	89.8	no	no	2.0	1.0	Sometimes

NAIVE BAYES MODEL

In [27]:
```
#Summary of the dataset
obesity.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Gender                          2111 non-null   object 
 1   Age                             2111 non-null   float64
 2   Height                          2111 non-null   float64
 3   Weight                          2111 non-null   float64
 4   family_history_with_overweight  2111 non-null   object 
 5   FAVC                            2111 non-null   object 
 6   FCVC                            2111 non-null   float64
 7   NCP                             2111 non-null   float64
 8   CAEC                            2111 non-null   object 
 9   SMOKE                           2111 non-null   object 
 10  CH2O                            2111 non-null   float64
 11  SCC                             2111 non-null   object 
 12  FAF                             2111 non-null   float64
 13  TUE                             2111 non-null   float64
 14  CALC                            2111 non-null   object 
 15  MTRANS                          2111 non-null   object 
 16  NObesity                        2111 non-null   object 
dtypes: float64(8), object(9)
memory usage: 280.5+ KB
```

In [28]:
```
#Address categorical data features with one hot encoding
pre_obesity = pd.get_dummies(obesity,columns=['Gender', 'family_history_with_overweight', 'FAVC', 'CAEC', 'SMOKE', 'SCC', 'CALC', 'MTRANS'],drop_first=False)
pre_obesity.head()
```

Out[28]:

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE	NObesity	Gender_Female	...
0	21.0	1.62	64.0	2.0	3.0	2.0	0.0	1.0	Normal_Weight	1	...
1	21.0	1.52	56.0	3.0	3.0	3.0	3.0	0.0	Normal_Weight	1	...
2	23.0	1.80	77.0	2.0	3.0	2.0	2.0	1.0	Normal_Weight	0	...
3	27.0	1.80	87.0	3.0	3.0	2.0	2.0	0.0	Overweight_Level_I	0	...
4	22.0	1.78	89.8	2.0	1.0	2.0	0.0	0.0	Overweight_Level_II	0	...

5 rows × 32 columns

In [29]:
```
#Separate the independent and dependent variables
X = pre_obesity.drop('NObesity', axis=1)
y = pre_obesity['NObesity']
```

In [30]:
```
#Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=100)
```

In [31]:

```
#Use the Gaussian Naive Bayes algorithm to train the model
model = GaussianNB()

model.fit(X_train, y_train);
```

In [32]:

```
#Check the evaluation metrics of the model
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_pred, y_test)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred,average='macro')
print("Accuracy:", accuracy*100)
print("Precision:", precision*100)
print("Recall:", recall*100)
```

```
Accuracy: 55.95408895265423
Precision: 58.46020331952657
Recall: 55.02774189904681
```

DECISION TREE MODEL

In [33]:

```
#Separate the independent and dependent variables
obesity_x = obesity.drop('NObesity', axis=1)
obesity_y = obesity['NObesity']
```

In [34]:

```
#Address categorical data features with one hot encoding
obesity_x_encoded = pd.get_dummies(obesity_x, drop_first= True)
obesity_x_encoded.head()
```

Out[34]:

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE	Gender_Male	family_history_with_overwe
0	21.0	1.62	64.0	2.0	3.0	2.0	0.0	1.0	0	
1	21.0	1.52	56.0	3.0	3.0	3.0	3.0	0.0	0	
2	23.0	1.80	77.0	2.0	3.0	2.0	2.0	1.0	1	
3	27.0	1.80	87.0	3.0	3.0	2.0	2.0	0.0	1	
4	22.0	1.78	89.8	2.0	1.0	2.0	0.0	0.0	1	

5 rows × 23 columns

In [35]:

```
#Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(obesity_x_encoded, obesity
_y, test_size=0.3)
```

In [36]:

```
#Train the model with the decision tree classifier
dtree_entropy = DecisionTreeClassifier(criterion = 'entropy',
random_state = 100, max_depth=3, min_samples_leaf=5)
dtree_entropy.fit(x_train, y_train)
```

Out[36]:

```
▾                          DecisionTreeClassifier

DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=5,
                       random_state=100)
```

In [37]:

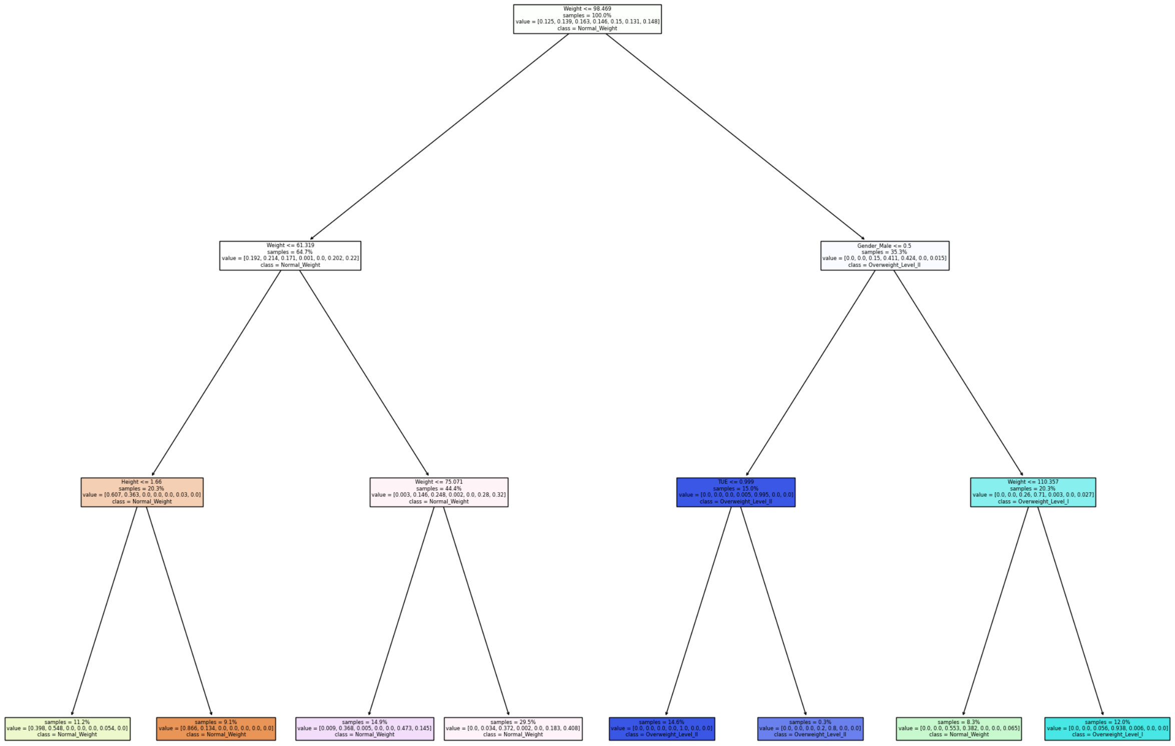
```
#Predict the test dataset values using the model above
y_pred_en = dtree_entropy.predict(x_test)
y_pred_en
```

```
Out[37]: array(['Overweight_Level_I', 'Normal_Weight', 'Obesity_Type_III',
                'Insufficient_Weight', 'Obesity_Type_I', 'Overweight_Level_II',
                'Obesity_Type_I', 'Obesity_Type_III', 'Obesity_Type_III',
                'Obesity_Type_III', 'Overweight_Level_I', 'Obesity_Type_I',
                'Overweight_Level_I', 'Overweight_Level_I', 'Overweight_Level_I',
                'Insufficient_Weight', 'Insufficient_Weight', 'Normal_Weight',
                'Overweight_Level_II', 'Obesity_Type_I', 'Obesity_Type_II',
                'Overweight_Level_II', 'Insufficient_Weight', 'Obesity_Type_II',
                'Overweight_Level_II', 'Obesity_Type_III', 'Obesity_Type_II',
                'Overweight_Level_II', 'Obesity_Type_II', 'Overweight_Level_I',
                'Overweight_Level_I', 'Overweight_Level_II', 'Obesity_Type_III',
                'Obesity_Type_III', 'Obesity_Type_III', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II', 'Obesity_Type_I',
                'Obesity_Type_I', 'Obesity_Type_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_I', 'Obesity_Type_I',
                'Normal_Weight', 'Obesity_Type_III', 'Obesity_Type_II',
                'Overweight_Level_I', 'Insufficient_Weight', 'Insufficient_Weight',
                'Obesity_Type_II', 'Overweight_Level_II', 'Overweight_Level_II',
                'Normal_Weight', 'Overweight_Level_I', 'Obesity_Type_I',
                'Overweight_Level_II', 'Insufficient_Weight', 'Obesity_Type_III',
                'Insufficient_Weight', 'Normal_Weight', 'Obesity_Type_III',
                'Overweight_Level_II', 'Normal_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Obesity_Type_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_I', 'Overweight_Level_II',
                'Obesity_Type_II', 'Overweight_Level_II', 'Overweight_Level_I',
                'Insufficient_Weight', 'Normal_Weight', 'Obesity_Type_II',
                'Insufficient_Weight', 'Insufficient_Weight', 'Obesity_Type_III',
                'Obesity_Type_III', 'Obesity_Type_II', 'Overweight_Level_I',
                'Overweight_Level_I', 'Obesity_Type_III', 'Obesity_Type_III',
                'Obesity_Type_III', 'Obesity_Type_I', 'Overweight_Level_I',
                'Normal_Weight', 'Insufficient_Weight', 'Obesity_Type_II',
                'Overweight_Level_II', 'Obesity_Type_I', 'Insufficient_Weight',
                'Obesity_Type_III', 'Obesity_Type_III', 'Obesity_Type_III',
                'Overweight_Level_II', 'Overweight_Level_I', 'Obesity_Type_III',
                'Normal_Weight', 'Overweight_Level_II', 'Obesity_Type_II',
                'Insufficient_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II', 'Normal_Weight',
                'Overweight_Level_I', 'Overweight_Level_I', 'Obesity_Type_III',
                'Normal_Weight', 'Obesity_Type_III', 'Obesity_Type_I',
                'Obesity_Type_I', 'Obesity_Type_III', 'Obesity_Type_III',
                'Overweight_Level_II', 'Overweight_Level_II', 'Normal_Weight',
                'Overweight_Level_II', 'Obesity_Type_I', 'Overweight_Level_II',
                'Normal_Weight', 'Obesity_Type_I', 'Overweight_Level_II',
                'Overweight_Level_II', 'Obesity_Type_III', 'Overweight_Level_I',
                'Normal_Weight', 'Obesity_Type_III', 'Overweight_Level_II',
                'Overweight_Level_II', 'Insufficient_Weight',
                'Overweight_Level_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II',
                'Insufficient_Weight', 'Overweight_Level_II', 'Normal_Weight',
                'Overweight_Level_I', 'Obesity_Type_III', 'Insufficient_Weight',
                'Overweight_Level_II', 'Overweight_Level_II', 'Overweight_Level_I',
                'Overweight_Level_II', 'Insufficient_Weight',
                'Overweight_Level_II', 'Obesity_Type_II', 'Obesity_Type_I',
                'Overweight_Level_I', 'Obesity_Type_III', 'Normal_Weight',
                'Overweight_Level_II', 'Insufficient_Weight', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_I', 'Insufficient_Weight',
                'Obesity_Type_I', 'Overweight_Level_I', 'Overweight_Level_II',
                'Obesity_Type_II', 'Overweight_Level_I', 'Obesity_Type_III',
                'Overweight_Level_I', 'Obesity_Type_III', 'Insufficient_Weight',
                'Insufficient_Weight', 'Insufficient_Weight', 'Obesity_Type_I',
                'Overweight_Level_I', 'Overweight_Level_I', 'Overweight_Level_I',
                'Overweight_Level_I', 'Obesity_Type_III', 'Overweight_Level_I',
                'Obesity_Type_III', 'Overweight_Level_II', 'Normal_Weight',
                'Normal_Weight', 'Obesity_Type_III', 'Insufficient_Weight',
                'Insufficient_Weight', 'Obesity_Type_II', 'Obesity_Type_III',
                'Obesity_Type_III', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_III', 'Overweight_Level_II', 'Obesity_Type_III',
                'Obesity_Type_III', 'Obesity_Type_III', 'Normal_Weight',
                'Overweight_Level_II', 'Overweight_Level_II', 'Obesity_Type_II',
                'Overweight_Level_I', 'Overweight_Level_I', 'Overweight_Level_I',
                'Overweight_Level_II', 'Normal_Weight', 'Obesity_Type_III',
                'Overweight_Level_II', 'Overweight_Level_II', 'Obesity_Type_I',
                'Insufficient_Weight', 'Overweight_Level_I', 'Insufficient_Weight',
                'Normal_Weight', 'Overweight_Level_II', 'Normal_Weight',
                'Obesity_Type_II', 'Insufficient_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Insufficient_Weight', 'Obesity_Type_III',
                'Overweight_Level_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II', 'Obesity_Type_III',
                'Obesity_Type_II', 'Insufficient_Weight', 'Overweight_Level_I',
                'Overweight_Level_II', 'Obesity_Type_III', 'Obesity_Type_III',
                'Overweight_Level_I', 'Obesity_Type_I', 'Obesity_Type_III',
                'Obesity_Type_II', 'Obesity_Type_I', 'Insufficient_Weight',
                'Normal_Weight', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_III', 'Obesity_Type_I', 'Obesity_Type_I',
                'Normal_Weight', 'Normal_Weight', 'Normal_Weight',
                'Overweight_Level_I', 'Obesity_Type_III', 'Overweight_Level_II',
                'Obesity_Type_III', 'Obesity_Type_II', 'Normal_Weight',
                'Obesity_Type_III', 'Obesity_Type_III', 'Obesity_Type_I',
                'Obesity_Type_III', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_II', 'Obesity_Type_I', 'Overweight_Level_II',
                'Insufficient_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II', 'Overweight_Level_I',
                'Normal_Weight', 'Obesity_Type_III', 'Overweight_Level_II',
                'Overweight_Level_II', 'Normal_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Obesity_Type_III', 'Insufficient_Weight',
                'Obesity_Type_III', 'Obesity_Type_I', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_I', 'Overweight_Level_I',
                'Obesity_Type_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Insufficient_Weight', 'Obesity_Type_III', 'Overweight_Level_I',
                'Obesity_Type_I', 'Obesity_Type_II', 'Obesity_Type_III',
                'Obesity_Type_III', 'Normal_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II', 'Obesity_Type_II',
                'Insufficient_Weight', 'Overweight_Level_II', 'Normal_Weight',
                'Overweight_Level_I', 'Overweight_Level_I', 'Obesity_Type_III',
                'Overweight_Level_I', 'Obesity_Type_II', 'Obesity_Type_II',
                'Obesity_Type_III', 'Normal_Weight', 'Overweight_Level_II',
                'Normal_Weight', 'Insufficient_Weight', 'Obesity_Type_III',
                'Obesity_Type_III', 'Normal_Weight', 'Overweight_Level_II',
                'Normal_Weight', 'Obesity_Type_III', 'Overweight_Level_I',
                'Obesity_Type_II', 'Obesity_Type_III', 'Normal_Weight',
                'Overweight_Level_II', 'Obesity_Type_III', 'Overweight_Level_I',
                'Obesity_Type_I', 'Obesity_Type_III', 'Obesity_Type_III',
                'Overweight_Level_II', 'Overweight_Level_II', 'Normal_Weight',
                'Obesity_Type_III', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_II', 'Overweight_Level_II', 'Obesity_Type_II',
                'Overweight_Level_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_III', 'Overweight_Level_II', 'Insufficient_Weight',
                'Obesity_Type_III', 'Obesity_Type_III', 'Insufficient_Weight',
                'Overweight_Level_I', 'Overweight_Level_I', 'Obesity_Type_II',
                'Normal_Weight', 'Overweight_Level_II', 'Overweight_Level_I',
                'Obesity_Type_II', 'Obesity_Type_II', 'Obesity_Type_II',
                'Obesity_Type_III', 'Obesity_Type_II', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_I', 'Normal_Weight',
                'Overweight_Level_II', 'Overweight_Level_I', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_II', 'Overweight_Level_I',
                'Obesity_Type_II', 'Obesity_Type_II', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_I', 'Overweight_Level_II',
                'Overweight_Level_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Normal_Weight', 'Obesity_Type_II', 'Obesity_Type_II',
                'Obesity_Type_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_I', 'Overweight_Level_II',
                'Insufficient_Weight', 'Obesity_Type_II', 'Obesity_Type_II',
                'Overweight_Level_II', 'Insufficient_Weight',
                'Overweight_Level_II', 'Obesity_Type_III', 'Overweight_Level_I',
                'Obesity_Type_II', 'Overweight_Level_II', 'Obesity_Type_III',
                'Overweight_Level_II', 'Obesity_Type_III', 'Overweight_Level_II',
                'Overweight_Level_I', 'Insufficient_Weight', 'Obesity_Type_III',
                'Overweight_Level_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_I', 'Obesity_Type_II', 'Obesity_Type_III',
                'Obesity_Type_I', 'Obesity_Type_II', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_I', 'Obesity_Type_I',
                'Obesity_Type_III', 'Overweight_Level_I', 'Obesity_Type_III',
                'Obesity_Type_I', 'Insufficient_Weight', 'Normal_Weight',
                'Obesity_Type_III', 'Overweight_Level_II', 'Insufficient_Weight',
                'Obesity_Type_II', 'Obesity_Type_III', 'Obesity_Type_III',
                'Obesity_Type_I', 'Insufficient_Weight', 'Insufficient_Weight',
                'Overweight_Level_I', 'Obesity_Type_II', 'Overweight_Level_II',
                'Normal_Weight', 'Normal_Weight', 'Overweight_Level_II',
                'Obesity_Type_III', 'Normal_Weight', 'Obesity_Type_I',
                'Obesity_Type_I', 'Overweight_Level_I', 'Insufficient_Weight',
                'Overweight_Level_II', 'Obesity_Type_II', 'Normal_Weight',
                'Overweight_Level_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Obesity_Type_II', 'Overweight_Level_II', 'Insufficient_Weight',
                'Overweight_Level_I', 'Normal_Weight', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_I', 'Obesity_Type_I',
                'Insufficient_Weight', 'Overweight_Level_I', 'Obesity_Type_II',
                'Obesity_Type_I', 'Overweight_Level_I', 'Obesity_Type_II',
                'Obesity_Type_I', 'Normal_Weight', 'Insufficient_Weight',
                'Insufficient_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Normal_Weight', 'Overweight_Level_I',
                'Obesity_Type_II', 'Overweight_Level_II', 'Normal_Weight',
                'Overweight_Level_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_I', 'Normal_Weight',
                'Obesity_Type_III', 'Obesity_Type_III', 'Overweight_Level_I',
                'Overweight_Level_II', 'Overweight_Level_I', 'Obesity_Type_III',
                'Obesity_Type_II', 'Obesity_Type_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Obesity_Type_II', 'Insufficient_Weight',
                'Obesity_Type_II', 'Obesity_Type_I', 'Overweight_Level_II',
                'Normal_Weight', 'Overweight_Level_I', 'Overweight_Level_II',
                'Obesity_Type_II', 'Normal_Weight', 'Overweight_Level_I',
                'Obesity_Type_III', 'Overweight_Level_II', 'Overweight_Level_I',
                'Obesity_Type_III', 'Overweight_Level_II', 'Obesity_Type_II',
                'Overweight_Level_I', 'Overweight_Level_II', 'Overweight_Level_II',
                'Normal_Weight', 'Overweight_Level_II', 'Obesity_Type_III',
                'Overweight_Level_I', 'Overweight_Level_II', 'Obesity_Type_III',
                'Overweight_Level_II', 'Normal_Weight', 'Obesity_Type_III',
                'Obesity_Type_III', 'Overweight_Level_II', 'Overweight_Level_I',
                'Overweight_Level_II', 'Obesity_Type_III', 'Normal_Weight',
                'Normal_Weight', 'Overweight_Level_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II',
                'Insufficient_Weight', 'Overweight_Level_II',
                'Overweight_Level_II', 'Insufficient_Weight', 'Overweight_Level_I',
                'Overweight_Level_II', 'Obesity_Type_I', 'Overweight_Level_II',
                'Insufficient_Weight', 'Obesity_Type_I', 'Obesity_Type_II',
                'Insufficient_Weight', 'Normal_Weight', 'Obesity_Type_II',
                'Obesity_Type_I', 'Obesity_Type_II', 'Obesity_Type_II',
                'Overweight_Level_II', 'Overweight_Level_II',
                'Overweight_Level_II', 'Overweight_Level_II', 'Normal_Weight',
                'Obesity_Type_I', 'Obesity_Type_I', 'Overweight_Level_II',
                'Normal_Weight', 'Insufficient_Weight', 'Overweight_Level_II',
                'Overweight_Level_I', 'Obesity_Type_I', 'Overweight_Level_I',
                'Overweight_Level_II', 'Obesity_Type_II', 'Overweight_Level_II',
                'Obesity_Type_III', 'Obesity_Type_II', 'Obesity_Type_II',
                'Overweight_Level_I', 'Normal_Weight', 'Obesity_Type_III',
                'Overweight_Level_I', 'Overweight_Level_II', 'Insufficient_Weight',
                'Overweight_Level_I', 'Overweight_Level_I', 'Obesity_Type_III',
                'Obesity_Type_III', 'Overweight_Level_I', 'Overweight_Level_I',
                'Normal_Weight', 'Insufficient_Weight', 'Overweight_Level_II',
                'Obesity_Type_II', 'Normal_Weight', 'Obesity_Type_II',
                'Overweight_Level_I', 'Normal_Weight', 'Obesity_Type_II',
                'Obesity_Type_I', 'Obesity_Type_I', 'Overweight_Level_II',
                'Overweight_Level_II', 'Obesity_Type_III', 'Overweight_Level_II',
                'Overweight_Level_II', 'Obesity_Type_I', 'Overweight_Level_I',
                'Overweight_Level_II', 'Insufficient_Weight', 'Obesity_Type_I',
                'Overweight_Level_I', 'Insufficient_Weight', 'Obesity_Type_III',
                'Overweight_Level_II', 'Overweight_Level_I', 'Obesity_Type_I',
                'Obesity_Type_III', 'Overweight_Level_II', 'Normal_Weight',
                'Overweight_Level_II', 'Obesity_Type_I', 'Insufficient_Weight',
                'Overweight_Level_II', 'Overweight_Level_II', 'Obesity_Type_II',
                'Overweight_Level_II', 'Obesity_Type_II', 'Obesity_Type_III',
                'Obesity_Type_II', 'Overweight_Level_II', 'Overweight_Level_I',
                'Insufficient_Weight', 'Overweight_Level_I', 'Obesity_Type_III',
                'Obesity_Type_II', 'Normal_Weight', 'Obesity_Type_III',
                'Overweight_Level_II', 'Insufficient_Weight', 'Overweight_Level_I',
                'Obesity_Type_III', 'Overweight_Level_II', 'Overweight_Level_II',
                'Insufficient_Weight'], dtype=object)
```

In [38]:
```
#Check the evaluation metrics of the model
accuracy = accuracy_score(y_test,y_pred_en)
precision = precision_score(y_test,y_pred_en,average='macro')
recall = recall_score(y_test,y_pred_en, average='macro')
print("Accuracy:", accuracy*100)
print("Precision:", precision*100)
print("Recall:", recall*100)
```

```
Accuracy: 63.722397476340696
Precision: 69.15626708764749
Recall: 64.60225390512039
```

In [39]:
```
#Visualize the decision tree
fig = plt.figure(figsize=((25,20)))
plot_tree(dtree_entropy,
            feature_names = obesity_x_encoded.columns,
            class_names=obesity_y,
            impurity=False,
            proportion=True,
            filled=True)
fig.savefig('test.png')
```


Weight <= 98.469
samples = 100.0%
value = [0.125, 0.139, 0.163, 0.146, 0.15, 0.131, 0.148]
class = Normal_Weight
Weight <= 61.319
samples = 64.7%
value = [0.192, 0.214, 0.171, 0.001, 0.0, 0.202, 0.22]
class = Normal_Weight
Gender_Male <= 0.5
samples = 35.3%
value = [0.0, 0.0, 0.15, 0.411, 0.424, 0.0, 0.015]
class = Overweight_Level_II
Height <= 1.66
samples = 20.3%
value = [0.607, 0.363, 0.0, 0.0, 0.0, 0.03, 0.0]
class = Normal_Weight
Weight <= 75.071
samples = 44.4%
value = [0.003, 0.146, 0.248, 0.002, 0.0, 0.28, 0.32]
class = Normal_Weight
TUE <= 0.999
samples = 15.0%
value = [0.0, 0.0, 0.0, 0.005, 0.995, 0.0, 0.0]
class = Overweight_Level_II
Weight <= 110.357
samples = 20.3%
value = [0.0, 0.0, 0.26, 0.71, 0.003, 0.0, 0.027]
class = Overweight_Level_I
samples = 11.2%
value = [0.398, 0.548, 0.0, 0.0, 0.0, 0.054, 0.0]
class = Normal_Weight
samples = 9.1%
value = [0.866, 0.134, 0.0, 0.0, 0.0, 0.0, 0.0]
class = Normal_Weight
samples = 14.9%
value = [0.009, 0.368, 0.005, 0.0, 0.0, 0.473, 0.145]
class = Normal_Weight
samples = 29.5%
value = [0.0, 0.034, 0.372, 0.002, 0.0, 0.183, 0.408]
class = Normal_Weight
samples = 14.6%
value = [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]
class = Overweight_Level_II
samples = 0.3%
value = [0.0, 0.0, 0.0, 0.2, 0.8, 0.0, 0.0]
class = Overweight_Level_II
samples = 8.3%
value = [0.0, 0.0, 0.553, 0.382, 0.0, 0.0, 0.065]
class = Normal_Weight
samples = 12.0%
value = [0.0, 0.0, 0.056, 0.938, 0.006, 0.0, 0.0]
class = Overweight_Level_I

RANDOM FOREST MODEL

In [40]:
```
#Map categorical values to numerical values
obesity['Gender'] = obesity['Gender'].map({'Female': 0, 'Male':1})
obesity['family_history_with_overweight'] = obesity['family_history_with_overweight'].map({'no': 0, 'yes':1})
obesity['FAVC'] = obesity['FAVC'].map({'no': 0, 'yes':1})
obesity['CAEC'] = obesity['CAEC'].map({'no': 0, 'Sometimes':1, 'Frequently':2, 'Always':3})
obesity['SMOKE'] = obesity['SMOKE'].map({'no': 0, 'yes':1})
obesity['SCC'] = obesity['SCC'].map({'no': 0, 'yes':1})
obesity['CALC'] = obesity['CALC'].map({'no': 0, 'Sometimes':1, 'Frequently':2, 'Always':3})
obesity['MTRANS'] = obesity['MTRANS'].map({'Walking': 0, 'Public_Transportation':1, 'Bike':2, 'Motorbike':3, 'Automobile':4})
```

In [41]:
```
#View the first 5 rows of the dataset to ensure that the mapping was applied correctly
obesity.head()
```

Out[41]:

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMC
0	0	21.0	1.62	64.0	1	0	2.0	3.0	1	
1	0	21.0	1.52	56.0	1	0	3.0	3.0	1	
2	1	23.0	1.80	77.0	1	0	2.0	3.0	1	
3	1	27.0	1.80	87.0	0	0	3.0	3.0	1	
4	1	22.0	1.78	89.8	0	0	2.0	1.0	1	

In [42]:
```
# Split the data into features (X) and target (y)
X = obesity.drop('NObesity', axis=1)
y = obesity['NObesity']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

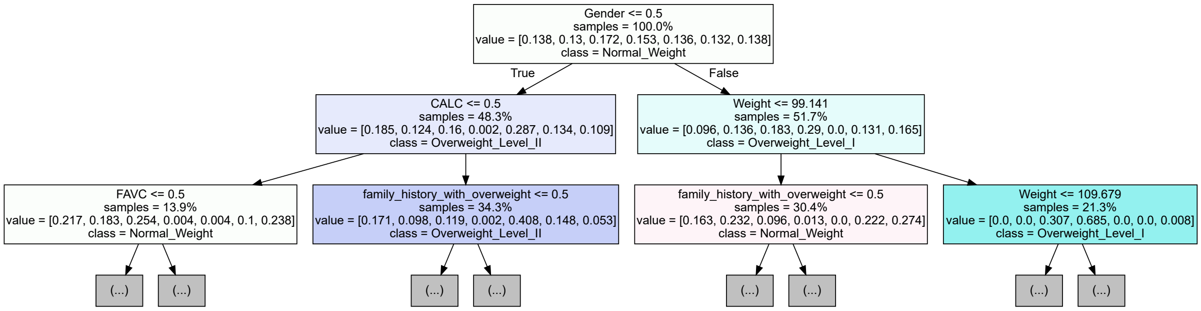
In [43]:
```
#Train the model using the training sets
rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(X_train, y_train)
```

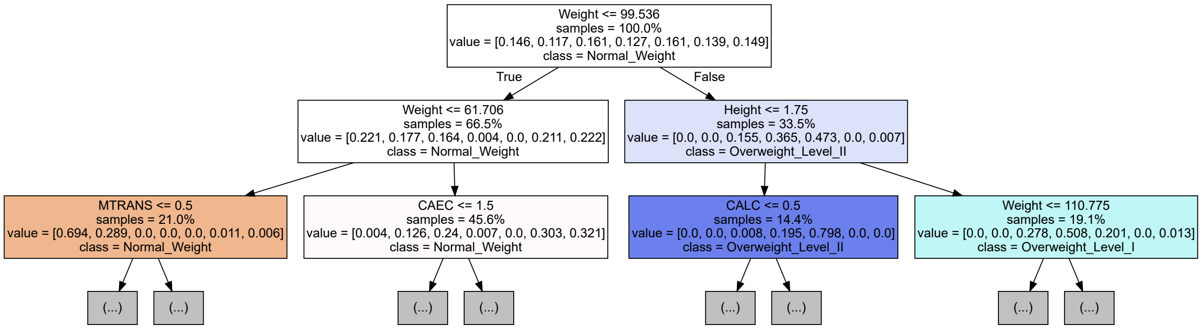
Out[43]:
```
▾          RandomForestClassifier
RandomForestClassifier(random_state=0)
```

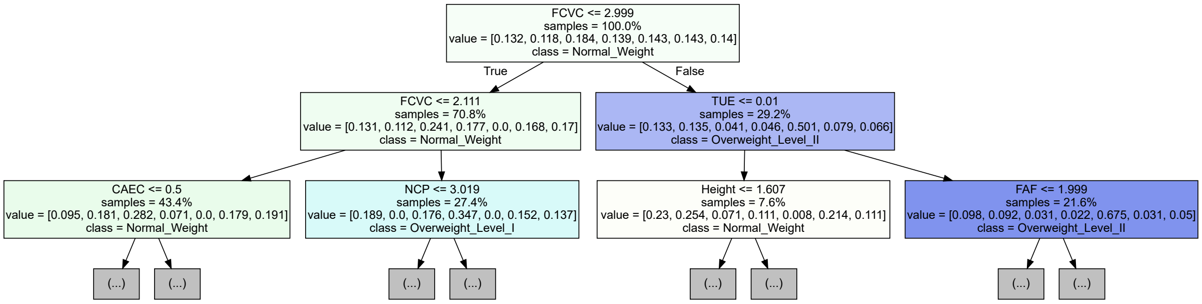
In [44]:
```
#Check the evaluation metrics of the model
y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
print("Accuracy:", accuracy*100)
print("Precision:", precision*100)
print("Recall:", recall*100)
```

```
Accuracy: 94.56264775413712
Precision: 94.8605341519006
Recall: 94.60728771693958
```

In [45]:
```
#Visualize the first 3 trees
for i in range(3):
    tree = rf.estimators_[i]
    dot_data = export_graphviz(tree,
                               feature_names=X_train.columns,
                               class_names=y,
                               filled=True,
                               max_depth=2,
                               impurity=False,
                               proportion=True)
    graph = graphviz.Source(dot_data)
    display(graph)
```


Gender <= 0.5
samples = 100.0%
value = [0.138, 0.13, 0.172, 0.153, 0.136, 0.132, 0.138]
class = Normal_Weight
True
False
CALC <= 0.5
samples = 48.3%
value = [0.185, 0.124, 0.16, 0.002, 0.287, 0.134, 0.109]
class = Overweight_Level_II
Weight <= 99.141
samples = 51.7%
value = [0.096, 0.136, 0.183, 0.29, 0.0, 0.131, 0.165]
class = Overweight_Level_I
FAVC <= 0.5
samples = 13.9%
value = [0.217, 0.183, 0.254, 0.004, 0.004, 0.1, 0.238]
class = Normal_Weight
family_history_with_overweight <= 0.5
samples = 34.3%
value = [0.171, 0.098, 0.119, 0.002, 0.408, 0.148, 0.053]
class = Overweight_Level_II
family_history_with_overweight <= 0.5
samples = 30.4%
value = [0.163, 0.232, 0.096, 0.013, 0.0, 0.222, 0.274]
class = Normal_Weight
Weight <= 109.679
samples = 21.3%
value = [0.0, 0.0, 0.307, 0.685, 0.0, 0.0, 0.008]
class = Overweight_Level_I
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)


Weight <= 99.536
samples = 100.0%
value = [0.146, 0.117, 0.161, 0.127, 0.161, 0.139, 0.149]
class = Normal_Weight
True
False
Weight <= 61.706
samples = 66.5%
value = [0.221, 0.177, 0.164, 0.004, 0.0, 0.211, 0.222]
class = Normal_Weight
Height <= 1.75
samples = 33.5%
value = [0.0, 0.0, 0.155, 0.365, 0.473, 0.0, 0.007]
class = Overweight_Level_II
MTRANS <= 0.5
samples = 21.0%
value = [0.694, 0.289, 0.0, 0.0, 0.0, 0.011, 0.006]
class = Normal_Weight
CAEC <= 1.5
samples = 45.6%
value = [0.004, 0.126, 0.24, 0.007, 0.0, 0.303, 0.321]
class = Normal_Weight
CALC <= 0.5
samples = 14.4%
value = [0.0, 0.0, 0.008, 0.195, 0.798, 0.0, 0.0]
class = Overweight_Level_II
Weight <= 110.775
samples = 19.1%
value = [0.0, 0.0, 0.278, 0.508, 0.201, 0.0, 0.013]
class = Overweight_Level_I
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)


FCVC <= 2.999
samples = 100.0%
value = [0.132, 0.118, 0.184, 0.139, 0.143, 0.143, 0.14]
class = Normal_Weight
True
False
FCVC <= 2.111
samples = 70.8%
value = [0.131, 0.112, 0.241, 0.177, 0.0, 0.168, 0.17]
class = Normal_Weight
TUE <= 0.01
samples = 29.2%
value = [0.133, 0.135, 0.041, 0.046, 0.501, 0.079, 0.066]
class = Overweight_Level_II
CAEC <= 0.5
samples = 43.4%
value = [0.095, 0.181, 0.282, 0.071, 0.0, 0.179, 0.191]
class = Normal_Weight
NCP <= 3.019
samples = 27.4%
value = [0.189, 0.0, 0.176, 0.347, 0.0, 0.152, 0.137]
class = Overweight_Level_I
Height <= 1.607
samples = 7.6%
value = [0.23, 0.254, 0.071, 0.111, 0.008, 0.214, 0.111]
class = Normal_Weight
FAF <= 1.999
samples = 21.6%
value = [0.098, 0.092, 0.031, 0.022, 0.675, 0.031, 0.05]
class = Overweight_Level_II
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)