Random Forest

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df1=pd.read_csv(r"C:\Users\user\Downloads\C6_bmi.csv")
df1
```

Out[3]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [11]:
```
df.columns
```

Out[11]:
```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Gender  10 non-null     object
 1   Height  10 non-null     int64 
 2   Weight  10 non-null     int64 
 3   Index   10 non-null     int64 
dtypes: int64(3), object(1)
memory usage: 448.0+ bytes
```

In [4]:

```
df=df1.head(10)
df
```

Out[4]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
5	Male	189	104	3
6	Male	147	92	5
7	Male	154	111	5
8	Male	174	90	3
9	Female	169	103	4

In [6]:

```
df['Index'].value_counts()
```

Out[6]:

```
3    4
4    3
5    2
2    1
Name: Index, dtype: int64
```

In [21]:

```
x=df[[ 'Height', 'Weight']]
y=df['Index']
```

In [22]:

```
g1={"g":{'g':1,'g':2}}
df=df.replace(g1)
print(df)
```

```
   Gender  Height  Weight  Index
0    Male     174      96      4
1    Male     189      87      2
2  Female     185     110      4
3  Female     195     104      3
4    Male     149      61      3
5    Male     189     104      3
6    Male     147      92      5
7    Male     154     111      5
8    Male     174      90      3
9  Female     169     103      4
```

In [23]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)
```

In [24]:

```
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[24]:

```
RandomForestClassifier()
```

In [25]:

```
parameters = { 'max_depth':[1,2,3,4,5],
    'min_samples_leaf':[5,10,15,20,25],
              'n_estimators':[10,20,30,40,50]
}
```

In [26]:

```
from sklearn.model_selection import GridSearchCV

grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="a
grid_search.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
666: UserWarning: The least populated class in y has only 1 members, which is
less than n_splits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[26]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [27]:

```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=5, min_samples_leaf=20, n_estimators=10)
```

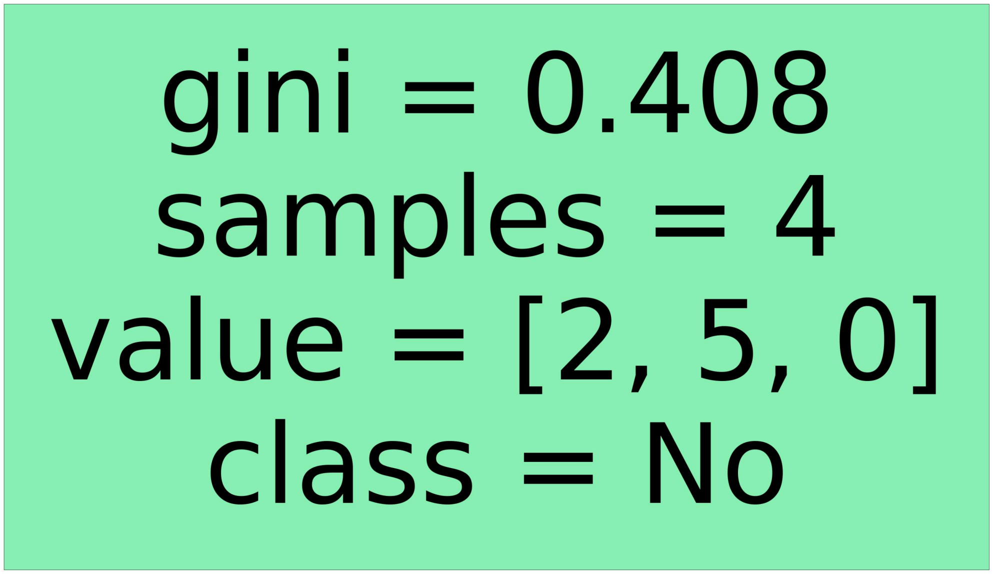
In [28]:

```
from sklearn.tree import plot_tree

plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['Yes','N
```

Out[28]:

```
[Text(2232.0, 1087.2, 'gini = 0.408\nsamples = 4\nvalue = [2, 5, 0]\nclass =
No')]
```


gini = 0.408
samples = 4
value = [2, 5, 0]
class = No