Random Forest

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [9]:
```
df1=pd.read_csv(r"C:\Users\user\Downloads\C4_framingham.csv")
df1
```

Out[9]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.0	0	0
1	0	46	2.0	0	0.0	0.0	0	0
2	1	48	1.0	1	20.0	0.0	0	0
3	0	61	3.0	1	30.0	0.0	0	1
4	0	46	3.0	1	23.0	0.0	0	0
...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
4235	0	48	2.0	1	20.0	NaN	0	0
4236	0	44	1.0	1	15.0	0.0	0	0
4237	0	52	2.0	0	0.0	0.0	0	0

4238 rows × 16 columns

In [11]:
```
df=df1.head(10)
df
```

Out[11]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	di
0	1	39	4.0	0	0.0	0.0	0	0	
1	0	46	2.0	0	0.0	0.0	0	0	
2	1	48	1.0	1	20.0	0.0	0	0	
3	0	61	3.0	1	30.0	0.0	0	1	
4	0	46	3.0	1	23.0	0.0	0	0	
5	0	43	2.0	0	0.0	0.0	0	1	
6	0	63	1.0	0	0.0	0.0	0	0	
7	0	45	2.0	1	20.0	0.0	0	0	
8	1	52	1.0	0	0.0	0.0	0	1	
9	1	43	1.0	1	30.0	0.0	0	1	

In [12]:
```
df['currentSmoker'].value_counts()
```

Out[12]:
```
0    5
1    5
Name: currentSmoker, dtype: int64
```

In [13]:
```
x=df.drop('currentSmoker',axis=1)
y=df['currentSmoker']
```

In [14]:
```
g1={"g":{'g':1,'g':2}}
df=df.replace(g1)
print(df)
```

```
   male  age  education  currentSmoker  cigsPerDay  BPMeds  prevalentStroke
\
0     1   39        4.0              0         0.0     0.0                0
1     0   46        2.0              0         0.0     0.0                0
2     1   48        1.0              1        20.0     0.0                0
3     0   61        3.0              1        30.0     0.0                0
4     0   46        3.0              1        23.0     0.0                0
5     0   43        2.0              0         0.0     0.0                0
6     0   63        1.0              0         0.0     0.0                0
7     0   45        2.0              1        20.0     0.0                0
8     1   52        1.0              0         0.0     0.0                0
9     1   43        1.0              1        30.0     0.0                0

   prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  heartRate  glucose
\
0             0         0    195.0  106.0   70.0  26.97       80.0     77.0
1             0         0    250.0  121.0   81.0  28.73       95.0     76.0
2             0         0    245.0  127.5   80.0  25.34       75.0     70.0
3             1         0    225.0  150.0   95.0  28.58       65.0    103.0
4             0         0    285.0  130.0   84.0  23.10       85.0     85.0
5             1         0    228.0  180.0  110.0  30.30       77.0     99.0
6             0         0    205.0  138.0   71.0  33.11       60.0     85.0
7             0         0    313.0  100.0   71.0  21.68       79.0     78.0
8             1         0    260.0  141.5   89.0  26.36       76.0     79.0
9             1         0    225.0  162.0  107.0  23.61       93.0     88.0

   TenYearCHD
0           0
1           0
2           0
3           1
4           0
5           0
6           1
7           0
8           0
9           0
```

In [15]:
```
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 [16]:
```
from sklearn.ensemble import RandomForestClassifier

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

Out[16]:
```
RandomForestClassifier()
```

In [17]:
```
parameters = { 'max_depth':[1,2,3,4,5],
    'min_samples_leaf':[5,10,15,20,25],
              'n_estimators':[10,20,30,40,50]
}
```

In [18]:
```
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)
```

Out[18]:
```
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 [19]:
```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=1, min_samples_leaf=5, n_estimators=20)
```

In [20]:
```
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[20]:
```
[Text(2232.0, 1087.2, 'gini = 0.408\nsamples = 3\nvalue = [2, 5]\nclass = N
o')]
```