In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.tree import plot_tree
```

In [2]:
```
df=pd.read_csv("C4_framingham - C4_framingham.csv")
df
```

Out[2]:

| | 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 [3]:
```
df1=df.fillna(value=0)
df1
```

Out[3]:

| cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totChol | sysBP | diaBP | BMI | hea |
|---|---|---|---|---|---|---|---|---|---|
| 0.0 | 0.0 | 0 | 0 | 0 | 195.0 | 106.0 | 70.0 | 26.97 | |
| 0.0 | 0.0 | 0 | 0 | 0 | 250.0 | 121.0 | 81.0 | 28.73 | |
| 20.0 | 0.0 | 0 | 0 | 0 | 245.0 | 127.5 | 80.0 | 25.34 | |
| 30.0 | 0.0 | 0 | 1 | 0 | 225.0 | 150.0 | 95.0 | 28.58 | |
| 23.0 | 0.0 | 0 | 0 | 0 | 285.0 | 130.0 | 84.0 | 23.10 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 1.0 | 0.0 | 0 | 1 | 0 | 313.0 | 179.0 | 92.0 | 25.97 | |
| 43.0 | 0.0 | 0 | 0 | 0 | 207.0 | 126.5 | 80.0 | 19.71 | |
| 20.0 | 0.0 | 0 | 0 | 0 | 248.0 | 131.0 | 72.0 | 22.00 | |
| 15.0 | 0.0 | 0 | 0 | 0 | 210.0 | 126.5 | 87.0 | 19.16 | |
| 0.0 | 0.0 | 0 | 0 | 0 | 269.0 | 133.5 | 83.0 | 21.47 | |

In [4]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4238 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4238 non-null   float64
 5   BPMeds           4238 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4238 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4238 non-null   float64
 13  heartRate        4238 non-null   float64
 14  glucose          4238 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [6]:
```
df1.columns
```

Out[6]:
```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
```

In [7]:
```
df2=df1[['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD']]
df2
```

Out[7]:

| r | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totChol | sysBP | diaBP | BMI | hea |
|---|---|---|---|---|---|---|---|---|---|---|
| ) | 0.0 | 0.0 | 0 | 0 | 0 | 195.0 | 106.0 | 70.0 | 26.97 | |
| ) | 0.0 | 0.0 | 0 | 0 | 0 | 250.0 | 121.0 | 81.0 | 28.73 | |
| l | 20.0 | 0.0 | 0 | 0 | 0 | 245.0 | 127.5 | 80.0 | 25.34 | |
| l | 30.0 | 0.0 | 0 | 1 | 0 | 225.0 | 150.0 | 95.0 | 28.58 | |
| l | 23.0 | 0.0 | 0 | 0 | 0 | 285.0 | 130.0 | 84.0 | 23.10 | |
| . | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| l | 1.0 | 0.0 | 0 | 1 | 0 | 313.0 | 179.0 | 92.0 | 25.97 | |
| l | 43.0 | 0.0 | 0 | 0 | 0 | 207.0 | 126.5 | 80.0 | 19.71 | |
| l | 20.0 | 0.0 | 0 | 0 | 0 | 248.0 | 131.0 | 72.0 | 22.00 | |
| l | 15.0 | 0.0 | 0 | 0 | 0 | 210.0 | 126.5 | 87.0 | 19.16 | |
| ) | 0.0 | 0.0 | 0 | 0 | 0 | 269.0 | 133.5 | 83.0 | 21.47 | |

In [8]:
```
df2[ 'TenYearCHD'].value_counts()
```

Out[8]:
```
0    3594
1     644
Name: TenYearCHD, dtype: int64
```

In [9]:
```
x=df2.drop( 'TenYearCHD',axis=1)
y=df2[ 'TenYearCHD']
```

In [10]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.70)
```

In [11]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[11]:
```
RandomForestClassifier()
```

In [12]:
```
parameters = {'max_depth':[1,2,3,4,5],
              'min_samples_leaf':[5,10,15,20,25],
               'n_estimators':[10,20,30,40,50]}
```

In [13]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='ac
grid_search.fit(x_train,y_train)
```

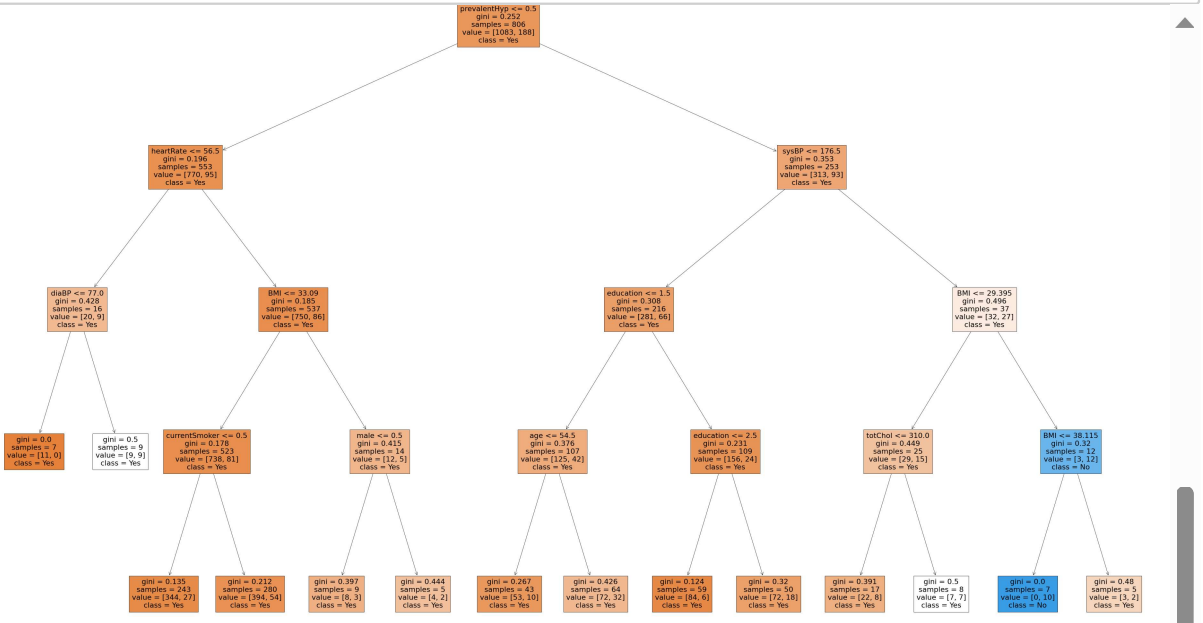
Out[13]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [14]:
```
grid_search.best_score_
```

Out[14]:
```
0.847365423661665
```

In [15]:
```
rfc_best =grid_search.best_estimator_
```

In [16]:
```
plt.figure(figsize=(80,50))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','N
```




In [ ]:
```
```

In [ ]:
```
```