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

In [2]:
```
from sklearn.linear_model import LogisticRegression
```

In [3]:
```
df=pd.read_csv("framingham.csv").dropna()
df
```

Out[3]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabe
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	
...	...	...	...	...	...	...	...	...	
4231	1	58	3.0	0	0.0	0.0	0	1	
4232	1	68	1.0	0	0.0	0.0	0	1	
4233	1	50	1.0	1	1.0	0.0	0	1	
4234	1	51	3.0	1	43.0	0.0	0	0	
4237	0	52	2.0	0	0.0	0.0	0	0	

3656 rows × 16 columns

In [4]:
```
df.dropna(inplace=True)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3656 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             3656 non-null   int64  
 1   age              3656 non-null   int64  
 2   education        3656 non-null   float64
 3   currentSmoker    3656 non-null   int64  
 4   cigsPerDay       3656 non-null   float64
 5   BPMeds           3656 non-null   float64
 6   prevalentStroke  3656 non-null   int64  
 7   prevalentHyp     3656 non-null   int64  
 8   diabetes         3656 non-null   int64  
 9   totChol          3656 non-null   float64
```

```
 10  sysBP           3656 non-null   float64
 11  diaBP           3656 non-null   float64
 12  BMI             3656 non-null   float64
 13  heartRate       3656 non-null   float64
 14  glucose         3656 non-null   float64
 15  TenYearCHD      3656 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 485.6 KB
```

In [6]:
```
feature_matrix = df[['male','age','education','currentSmoker','cigsPerDay','BPMeds','pr
target_vector = df['TenYearCHD']
```

In [7]:
```
feature_matrix.shape
```

Out[7]: (3656, 15)

In [8]:
```
target_vector.shape
```

Out[8]: (3656,)

In [9]:
```
from sklearn.preprocessing import StandardScaler
```

In [10]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [11]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[11]: LogisticRegression()

In [12]:
```
feature_matrix.shape
```

Out[12]: (3656, 15)

In [13]:
```
target_vector.shape
```

Out[13]: (3656,)

In [14]:
```
from sklearn.preprocessing import StandardScaler
```

In [15]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [16]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[16]: LogisticRegression()

In [17]:
```
observation=df[['male','age','education','currentSmoker','cigsPerDay','BPMeds','prevale
```

In [18]:
```
prediction = logr.predict(observation)
prediction
```

Out[18]:
```
array([1, 1, 1, ..., 1, 1, 1], dtype=int64)
```

In [19]:
```
logr.classes_
```

Out[19]:
```
array([0, 1], dtype=int64)
```

In [20]:
```
logr.predict_proba(observation)[0][1]
```

Out[20]:
```
1.0
```

In [21]:
```
df['TenYearCHD'].value_counts()
```

Out[21]:
```
0    3099
1     557
Name: TenYearCHD, dtype: int64
```

In [22]:
```
x=df.drop('TenYearCHD', axis=1)
y=df['TenYearCHD']
```

In [23]:
```
g1={'TenYearCHD':{"0":1, "1":2}}
df=df.replace(g1)
df
```

Out[23]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabe
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	
...	...	...	...	...	...	...	...	...	
4231	1	58	3.0	0	0.0	0.0	0	1	
4232	1	68	1.0	0	0.0	0.0	0	1	
4233	1	50	1.0	1	1.0	0.0	0	1	
4234	1	51	3.0	1	43.0	0.0	0	0	
4237	0	52	2.0	0	0.0	0.0	0	0	

3656 rows × 16 columns

In [24]:
```
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 [25]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[25]:
```
RandomForestClassifier()
```

In [26]:
```
parameters = {'max_depth':[1,2,3,4,5],'min_samples_leaf':[5,10,15,20,25],'n_estimators'
```

In [27]:
```
from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(estimator=rfc,param_grid= parameters,cv=2,scoring = "accurac
grid_search.fit(x_train,y_train)
```

Out[27]:
```
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 [28]:
```
grid_search.best_score_
```

Out[28]:
```
0.8475973050234558
```

In [29]:
```
rfc_best = grid_search.best_estimator_
```

In [30]:
```
from sklearn.tree import plot_tree
plt.figure(figsize = (80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names = ['Yes','No'],fi
```

Out[30]:
```
[Text(2287.7999999999997, 1956.96, 'prevalentHyp <= 0.5\ngini = 0.261\nsamples = 1614\nv
alue = [2165, 394]\nclass = Yes'),
 Text(1227.6, 1522.0800000000002, 'totChol <= 204.5\ngini = 0.193\nsamples = 1110\nvalue
= [1569, 191]\nclass = Yes'),
 Text(669.5999999999999, 1087.2, 'diaBP <= 90.5\ngini = 0.108\nsamples = 293\nvalue = [4
30, 26]\nclass = Yes'),
 Text(446.4, 652.3200000000002, 'heartRate <= 89.5\ngini = 0.099\nsamples = 282\nvalue =
[419, 23]\nclass = Yes'),
 Text(223.2, 217.44000000000005, 'gini = 0.083\nsamples = 251\nvalue = [377, 17]\nclass
= Yes'),
 Text(669.5999999999999, 217.44000000000005, 'gini = 0.219\nsamples = 31\nvalue = [42,
6]\nclass = Yes'),
 Text(892.8, 652.3200000000002, 'gini = 0.337\nsamples = 11\nvalue = [11, 3]\nclass = Ye
s'),
 Text(1785.6, 1087.2, 'education <= 1.5\ngini = 0.221\nsamples = 817\nvalue = [1139, 16
5]\nclass = Yes'),
 Text(1339.1999999999998, 652.3200000000002, 'male <= 0.5\ngini = 0.287\nsamples = 346\n
value = [453, 95]\nclass = Yes'),
 Text(1116.0, 217.44000000000005, 'gini = 0.196\nsamples = 179\nvalue = [250, 31]\nclass
= Yes'),
```

```
 Text(1562.3999999999999, 217.44000000000005, 'gini = 0.364\nsamples = 167\nvalue = [20
3, 64]\nclass = Yes'),
 Text(2232.0, 652.3200000000002, 'BMI <= 21.98\ngini = 0.168\nsamples = 471\nvalue = [68
6, 70]\nclass = Yes'),
 Text(2008.8, 217.44000000000005, 'gini = 0.028\nsamples = 85\nvalue = [138, 2]\nclass =
Yes'),
 Text(2455.2, 217.44000000000005, 'gini = 0.196\nsamples = 386\nvalue = [548, 68]\nclass
= Yes'),
 Text(3348.0, 1522.0800000000002, 'diaBP <= 72.75\ngini = 0.379\nsamples = 504\nvalue =
[596, 203]\nclass = Yes'),
 Text(3124.7999999999997, 1087.2, 'gini = 0.346\nsamples = 12\nvalue = [4, 14]\nclass =
No'),
 Text(3571.2, 1087.2, 'sysBP <= 151.75\ngini = 0.367\nsamples = 492\nvalue = [592, 189]
\nclass = Yes'),
 Text(3124.7999999999997, 652.3200000000002, 'glucose <= 116.5\ngini = 0.297\nsamples =
251\nvalue = [320, 71]\nclass = Yes'),
 Text(2901.6, 217.44000000000005, 'gini = 0.284\nsamples = 240\nvalue = [314, 65]\nclass
= Yes'),
 Text(3348.0, 217.44000000000005, 'gini = 0.5\nsamples = 11\nvalue = [6, 6]\nclass = Ye
s'),
 Text(4017.6, 652.3200000000002, 'heartRate <= 68.5\ngini = 0.422\nsamples = 241\nvalue
= [272, 118]\nclass = Yes'),
 Text(3794.3999999999996, 217.44000000000005, 'gini = 0.5\nsamples = 58\nvalue = [48, 4
6]\nclass = Yes'),
 Text(4240.8, 217.44000000000005, 'gini = 0.368\nsamples = 183\nvalue = [224, 72]\nclass
= Yes')]
```