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("C4 framingham csv").dropna()

df
```

Out[3]:

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

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
```

In [6]:
```
feature_matrix = df[['male','age','education','currentSmoker','cigsPerDay','BPM
                     'diabetes','totChol','sysBP','diaBP','BMI','heartRate','g
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'
                'diabetes','totChol','sysBP','diaBP','BMI','heartRate','g
```

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
```

Random Forest

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

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

In [22]:
```
x=df[['male','age','education','currentSmoker','cigsPerDay','BPMeds','prevalen
                'diabetes','totChol','sysBP','diaBP','BMI','heartRate','g
y=df['TenYearCHD']
```

In [31]:
```
g1={'TenYearCHD':{'1':1, '0':2}}
df=df.replace(g1)
df
```

Out[31]:

	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
...	...	...	...	...	...	...	...	...
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 [32]:
```
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 [33]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[33]:
```
RandomForestClassifier()
```

In [34]:
```
parameters = {'max_depth':[1,2,3,4,5],'min_samples_leaf':[5,10,15,20,25],
              'n_estimators': [10,20,30,40,50]
              }
```

In [35]:
```
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[35]:
```
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 [36]:
```
grid_search.best_score_
```

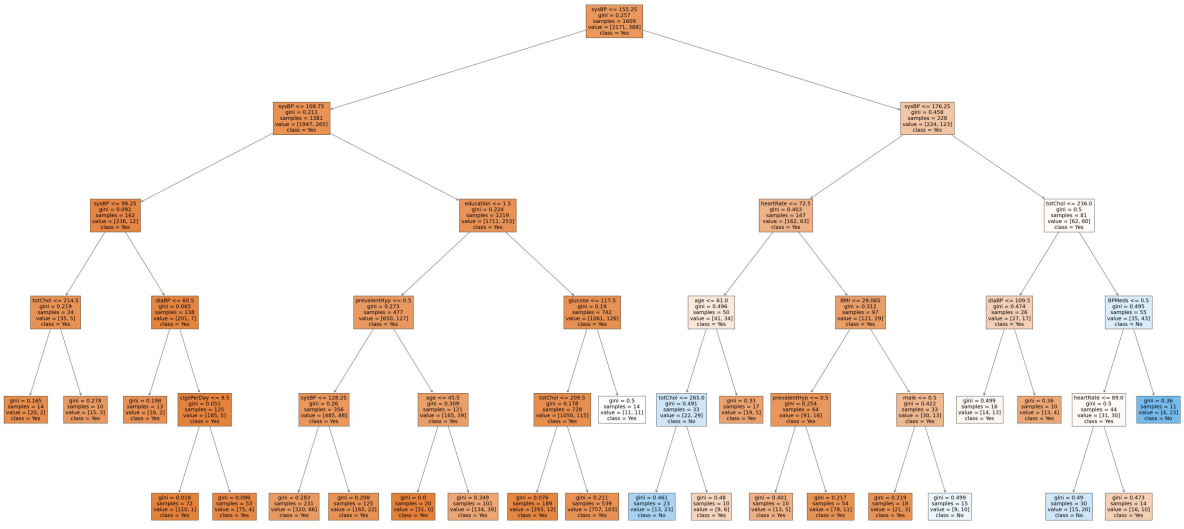
Out[36]:
```
0.8456432637802971
```

In [37]:
```
rfc_best = grid_search.best_estimator_
```

In [38]:
```
from sklearn.tree import plot_tree
plt.figure(figsize = (80,40,))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','
```

Out[38]:
```
[Text(2315.7, 1993.2, 'sysBP <= 155.25\ngini = 0.257\nsamples = 1609\nvalue =
[2171, 388]\nclass = Yes'),
 Text(1143.8999999999999, 1630.8000000000002, 'sysBP <= 108.75\ngini = 0.211
\nsamples = 1381\nvalue = [1947, 265]\nclass = Yes'),
 Text(446.4, 1268.4, 'sysBP <= 98.25\ngini = 0.092\nsamples = 162\nvalue = [2
36, 12]\nclass = Yes'),
 Text(223.2, 906.0, 'totChol <= 214.5\ngini = 0.219\nsamples = 24\nvalue = [3
5, 5]\nclass = Yes'),
 Text(111.6, 543.5999999999999, 'gini = 0.165\nsamples = 14\nvalue = [20, 2]
\nclass = Yes'),
 Text(334.79999999999995, 543.5999999999999, 'gini = 0.278\nsamples = 10\nval
ue = [15, 3]\nclass = Yes'),
 Text(669.5999999999999, 906.0, 'diaBP <= 60.5\ngini = 0.065\nsamples = 138\n
value = [201, 7]\nclass = Yes'),
 Text(558.0, 543.5999999999999, 'gini = 0.198\nsamples = 13\nvalue = [16, 2]
\nclass = Yes'),
 Text(781.1999999999999, 543.5999999999999, 'cigsPerDay <= 9.5\ngini = 0.051
\nsamples = 125\nvalue = [185, 5]\nclass = Yes'),
 Text(669.5999999999999, 181.19999999999982, 'gini = 0.018\nsamples = 72\nval
ue = [110, 1]\nclass = Yes'),
 Text(892.8, 181.19999999999982, 'gini = 0.096\nsamples = 53\nvalue = [75, 4]
\nclass = Yes'),
 Text(1841.3999999999999, 1268.4, 'education <= 1.5\ngini = 0.224\nsamples =
1219\nvalue = [1711, 253]\nclass = Yes'),
 Text(1450.8, 906.0, 'prevalentHyp <= 0.5\ngini = 0.273\nsamples = 477\nvalue
= [650, 127]\nclass = Yes'),
 Text(1227.6, 543.5999999999999, 'sysBP <= 128.25\ngini = 0.26\nsamples = 356
\nvalue = [485, 88]\nclass = Yes'),
 Text(1116.0, 181.19999999999982, 'gini = 0.283\nsamples = 231\nvalue = [320,
66]\nclass = Yes'),
 Text(1339.1999999999998, 181.19999999999982, 'gini = 0.208\nsamples = 125\nv
alue = [165, 22]\nclass = Yes'),
 Text(1674.0, 543.5999999999999, 'age <= 45.5\ngini = 0.309\nsamples = 121\nv
alue = [165, 39]\nclass = Yes'),
 Text(1562.3999999999999, 181.19999999999982, 'gini = 0.0\nsamples = 20\nvalu
e = [31, 0]\nclass = Yes'),
 Text(1785.6, 181.19999999999982, 'gini = 0.349\nsamples = 101\nvalue = [134,
39]\nclass = Yes'),
 Text(2232.0, 906.0, 'glucose <= 117.5\ngini = 0.19\nsamples = 742\nvalue =
[1061, 126]\nclass = Yes'),
 Text(2120.4, 543.5999999999999, 'totChol <= 209.5\ngini = 0.178\nsamples = 7
28\nvalue = [1050, 115]\nclass = Yes'),
 Text(2008.8, 181.19999999999982, 'gini = 0.076\nsamples = 189\nvalue = [293,
12]\nclass = Yes'),
 Text(2232.0, 181.19999999999982, 'gini = 0.211\nsamples = 539\nvalue = [757,
103]\nclass = Yes'),
 Text(2343.6, 543.5999999999999, 'gini = 0.5\nsamples = 14\nvalue = [11, 11]
\nclass = Yes'),
 Text(3487.5, 1630.8000000000002, 'sysBP <= 176.25\ngini = 0.458\nsamples = 2
28\nvalue = [224, 123]\nclass = Yes'),
 Text(2957.3999999999996, 1268.4, 'heartRate <= 72.5\ngini = 0.403\nsamples =
147\nvalue = [162, 63]\nclass = Yes'),
 Text(2678.3999999999996, 906.0, 'age <= 61.0\ngini = 0.496\nsamples = 50\nva
lue = [41, 34]\nclass = Yes'),
 Text(2566.7999999999997, 543.5999999999999, 'totChol <= 265.0\ngini = 0.491
\nsamples = 33\nvalue = [22, 29]\nclass = No'),
 Text(2455.2, 181.19999999999982, 'gini = 0.461\nsamples = 23\nvalue = [13, 2
```

```
3]\nclass = No'),
 Text(2678.3999999999996, 181.19999999999982, 'gini = 0.48\nsamples = 10\nval
ue = [9, 6]\nclass = Yes'),
 Text(2790.0, 543.5999999999999, 'gini = 0.33\nsamples = 17\nvalue = [19, 5]
\nclass = Yes'),
 Text(3236.3999999999996, 906.0, 'BMI <= 29.065\ngini = 0.312\nsamples = 97\n
value = [121, 29]\nclass = Yes'),
 Text(3013.2, 543.5999999999999, 'prevalentHyp <= 0.5\ngini = 0.254\nsamples
= 64\nvalue = [91, 16]\nclass = Yes'),
 Text(2901.6, 181.19999999999982, 'gini = 0.401\nsamples = 10\nvalue = [13,
5]\nclass = Yes'),
 Text(3124.7999999999997, 181.19999999999982, 'gini = 0.217\nsamples = 54\nva
lue = [78, 11]\nclass = Yes'),
 Text(3459.6, 543.5999999999999, 'male <= 0.5\ngini = 0.422\nsamples = 33\nva
lue = [30, 13]\nclass = Yes'),
 Text(3348.0, 181.19999999999982, 'gini = 0.219\nsamples = 18\nvalue = [21,
3]\nclass = Yes'),
 Text(3571.2, 181.19999999999982, 'gini = 0.499\nsamples = 15\nvalue = [9, 1
0]\nclass = No'),
 Text(4017.6, 1268.4, 'totChol <= 236.0\ngini = 0.5\nsamples = 81\nvalue = [6
2, 60]\nclass = Yes'),
 Text(3794.3999999999996, 906.0, 'diaBP <= 109.5\ngini = 0.474\nsamples = 26
\nvalue = [27, 17]\nclass = Yes'),
 Text(3682.7999999999997, 543.5999999999999, 'gini = 0.499\nsamples = 16\nval
ue = [14, 13]\nclass = Yes'),
 Text(3906.0, 543.5999999999999, 'gini = 0.36\nsamples = 10\nvalue = [13, 4]
\nclass = Yes'),
 Text(4240.8, 906.0, 'BPMeds <= 0.5\ngini = 0.495\nsamples = 55\nvalue = [35,
43]\nclass = No'),
 Text(4129.2, 543.5999999999999, 'heartRate <= 89.0\ngini = 0.5\nsamples = 44
\nvalue = [31, 30]\nclass = Yes'),
 Text(4017.6, 181.19999999999982, 'gini = 0.49\nsamples = 30\nvalue = [15, 2
0]\nclass = No'),
 Text(4240.8, 181.19999999999982, 'gini = 0.473\nsamples = 14\nvalue = [16, 1
0]\nclass = Yes'),
 Text(4352.4, 543.5999999999999, 'gini = 0.36\nsamples = 11\nvalue = [4, 13]
\nclass = No')]
```



In [ ]: