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

In [54]:
```
data=pd.read_csv(r"C:\Users\user\Desktop\vicky\C4_framingham (1).csv")
```

In [94]:
```
data.fillna(value=1)
```

Out[94]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0
...	...	...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0
4235	0	48	2.0	1	20.0	1.0	0	0	0	248.0
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0

4238 rows × 16 columns

In [95]:
```
data.head()
```

Out[95]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sys
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	10
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	12
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	12
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	15
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	13

In [105]:
```
data.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        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   object 
dtypes: float64(9), int64(6), object(1)
memory usage: 529.9+ KB
```

In [107]:
```
data1=data[['male','age','diabetes','TenYearCHD']]
```

In [108]:
```
data1['TenYearCHD'].value_counts()
```

Out[108]:
```
No     3594
Yes     644
Name: TenYearCHD, dtype: int64
```

In [109]:
```
x=data1.drop('TenYearCHD',axis=1)
y=data1['TenYearCHD']
```

In [ ]:
```

```

In [110]:
```
g1={"TenYearCHD":{1:"Yes",0:"No",}}
data1=data1.replace(g1)
print(data1)
```

```
      male  age  diabetes TenYearCHD
0        1   39         0         No
1        0   46         0         No
2        1   48         0         No
3        0   61         0        Yes
4        0   46         0         No
...    ...  ...       ...        ...
4233     1   50         0        Yes
4234     1   51         0         No
4235     0   48         0         No
4236     0   44         0         No
4237     0   52         0         No

[4238 rows x 4 columns]
```

In [111]:
```
from sklearn.model_selection import train_test_split
```

In [112]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [113]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [114]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[114]:
```
RandomForestClassifier()
```

In [115]:
```
parameters = {'max_depth':[1,2,3,4,5],
              'min_samples_leaf':[5,10,15,20,25],
              'n_estimators':[10,20,30,40,50]

}
```

In [116]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[116]:
```
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 [117]:
```
grid_search.best_score_
```

Out[117]:
```
0.844571813890762
```

In [118]:
```
from sklearn.tree import plot_tree
```

In [119]:
```
rfc_best=grid_search.best_estimator_
```

In [120]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True
```

Out[120]:
```
[Text(2232.0, 1902.6000000000001, 'age <= 60.5\ngini = 0.267\nsamples = 1879\nvalue = [2495,
471]\nclass = Yes'),
 Text(1116.0, 1359.0, 'age <= 48.5\ngini = 0.218\nsamples = 1623\nvalue = [2239, 318]\nclass
= Yes'),
 Text(558.0, 815.4000000000001, 'male <= 0.5\ngini = 0.138\nsamples = 901\nvalue = [1320, 10
6]\nclass = Yes'),
 Text(279.0, 271.79999999999995, 'gini = 0.12\nsamples = 505\nvalue = [741, 51]\nclass = Ye
s'),
 Text(837.0, 271.79999999999995, 'gini = 0.158\nsamples = 396\nvalue = [579, 55]\nclass = Ye
s'),
 Text(1674.0, 815.4000000000001, 'age <= 49.5\ngini = 0.305\nsamples = 722\nvalue = [919, 21
2]\nclass = Yes'),
 Text(1395.0, 271.79999999999995, 'gini = 0.38\nsamples = 60\nvalue = [70, 24]\nclass = Ye
s'),
 Text(1953.0, 271.79999999999995, 'gini = 0.297\nsamples = 662\nvalue = [849, 188]\nclass = Y
es'),
 Text(3348.0, 1359.0, 'male <= 0.5\ngini = 0.468\nsamples = 256\nvalue = [256, 153]\nclass =
Yes'),
 Text(2790.0, 815.4000000000001, 'diabetes <= 0.5\ngini = 0.468\nsamples = 144\nvalue = [147,
88]\nclass = Yes'),
 Text(2511.0, 271.79999999999995, 'gini = 0.46\nsamples = 136\nvalue = [141, 79]\nclass = Ye
s'),
 Text(3069.0, 271.79999999999995, 'gini = 0.48\nsamples = 8\nvalue = [6, 9]\nclass = No'),
 Text(3906.0, 815.4000000000001, 'diabetes <= 0.5\ngini = 0.468\nsamples = 112\nvalue = [109,
65]\nclass = Yes'),
 Text(3627.0, 271.79999999999995, 'gini = 0.442\nsamples = 103\nvalue = [108, 53]\nclass = Ye
s'),
 Text(4185.0, 271.79999999999995, 'gini = 0.142\nsamples = 9\nvalue = [1, 12]\nclass = No')]
```

In [ ]:
```

```