Random Forest

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(r"C:\Users\user\Downloads\fra.csv")
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	
...	...	...	...	...	...	...	...	...	
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 [4]:

```
df.columns
```

Out[4]:

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

In [13]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 15 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  TenYearCHD       4238 non-null   int64  
dtypes: float64(8), int64(7)
memory usage: 496.8 KB
```

In [14]:

```
d=df[['male','age','currentSmoker','prevalentStroke', 'prevalentHyp', 'diabetes','TenYearCHD']]
```

In [15]:

```
d['TenYearCHD'].value_counts()
```

Out[15]:

```
0    3594
1     644
Name: TenYearCHD, dtype: int64
```

In [16]:

```
x=d.drop('TenYearCHD',axis=1)
y=d['TenYearCHD']
```

In [17]:

```
TenYearCHD1={"TenYearCHD":{'TenYearCHD':0,'TenYearCHD':1}}
d=d.replace('TenYearCHD')
print(d)
```

```
      male  age  currentSmoker  prevalentStroke  prevalentHyp  diabetes  \
0        1   39              0                0             0         0   
1        0   46              0                0             0         0   
2        1   48              1                0             0         0   
3        0   61              1                0             1         0   
4        0   46              1                0             0         0   
...    ...  ...            ...              ...           ...       ...   
4233     1   50              1                0             1         0   
4234     1   51              1                0             0         0   
4235     0   48              1                0             0         0   
4236     0   44              1                0             0         0   
4237     0   52              0                0             0         0   

      TenYearCHD  
0              0  
1              0  
2              0  
3              1  
4              0  
...          ...  
4233           1  
4234           0  
4235           0  
4236           0  
4237           0  

[4238 rows x 7 columns]
```

In [18]:

```
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 [19]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [20]:

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

Out[20]:

```
RandomForestClassifier()
```

In [21]:

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

In [22]:

```
from sklearn.model_selection import GridSearchCV
```

In [23]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[23]:

```
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 [24]:

```
grid_search.best_score_
```

Out[24]:

```
0.8472690492245448
```

In [25]:

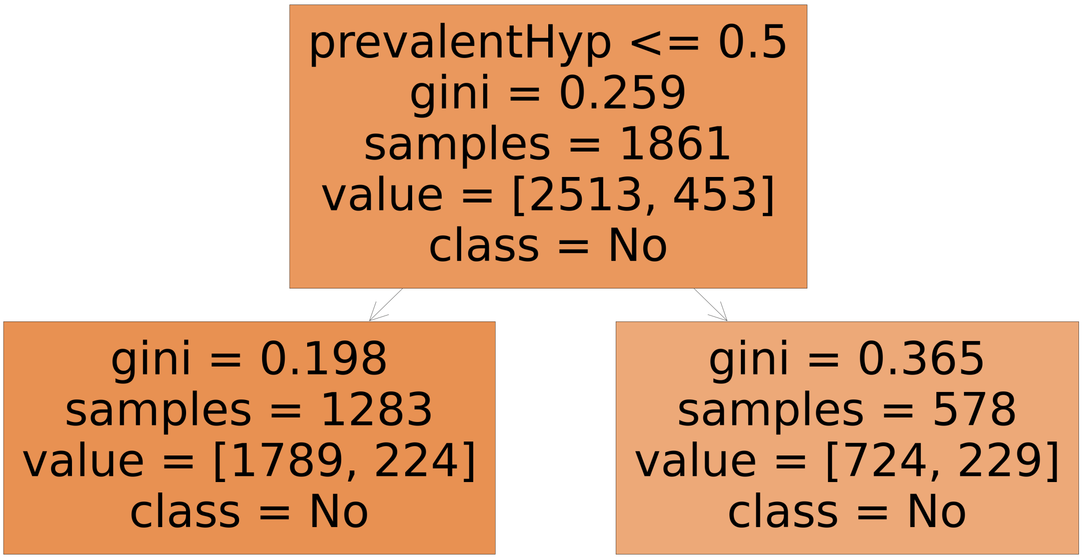
```
rfc_best=grid_search.best_estimator_
```

In [26]:

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

Out[26]:

```
[Text(2232.0, 1630.8000000000002, 'prevalentHyp <= 0.5\ngini = 0.259\nsamples = 18
61\nvalue = [2513, 453]\nclass = No'),
 Text(1116.0, 543.5999999999999, 'gini = 0.198\nsamples = 1283\nvalue = [1789, 22
4]\nclass = No'),
 Text(3348.0, 543.5999999999999, 'gini = 0.365\nsamples = 578\nvalue = [724, 229]
\nclass = No')]
```



In [ ]: