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]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\bmi.csv")
d
```

Out[3]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [4]:

```
d.columns
```

Out[4]:

```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

In [5]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Gender  500 non-null    object
 1   Height  500 non-null    int64 
 2   Weight  500 non-null    int64 
 3   Index   500 non-null    int64 
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

In [6]:

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

Out[6]:

```
Female    255
Male      245
Name: Gender, dtype: int64
```

In [7]:

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

In [8]:

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

```
     Gender  Height  Weight  Index
0      Male     174      96      4
1      Male     189      87      2
2    Female     185     110      4
3    Female     195     104      3
4      Male     149      61      3
..      ...     ...     ...    ...
495  Female     150     153      5
496  Female     184     121      4
497  Female     141     136      5
498    Male     150      95      5
499    Male     173     131      5

[500 rows x 4 columns]
```

In [9]:

```
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 [10]:

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

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]:

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

In [14]:

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

Out[14]:

```
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 [15]:

```
grid_search.best_score_
```

Out[15]:

```
0.5828571428571429
```

In [16]:

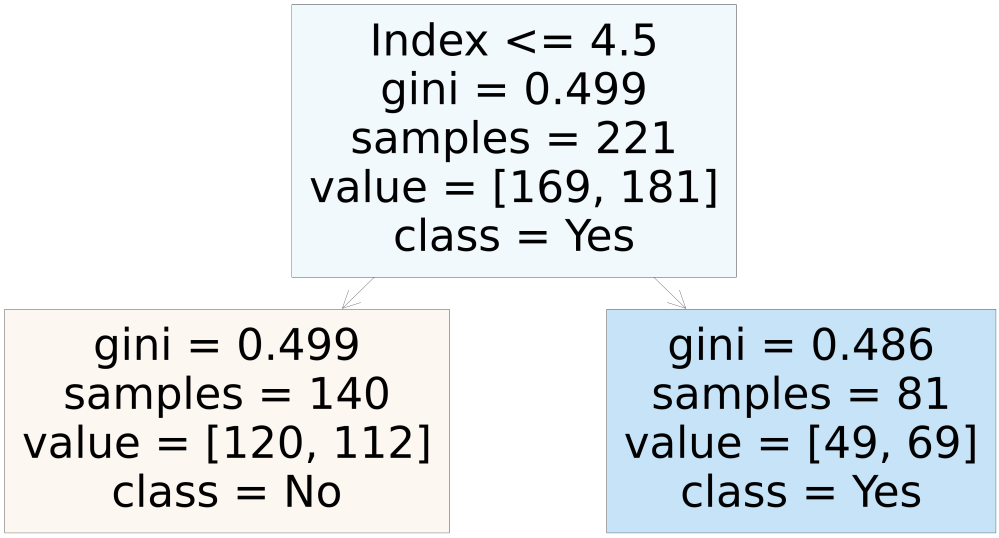
```
rfc_best=grid_search.best_estimator_
```

In [17]:

```
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
```

Out[17]:

```
[Text(2232.0, 1630.8000000000002, 'Index <= 4.5\ngini = 0.499\nsamples = 2
21\nvalue = [169, 181]\nclass = Yes'),
 Text(1116.0, 543.5999999999999, 'gini = 0.499\nsamples = 140\nvalue = [12
0, 112]\nclass = No'),
 Text(3348.0, 543.5999999999999, 'gini = 0.486\nsamples = 81\nvalue = [49,
69]\nclass = Yes')]
```



In [ ]: