RANDOM FOREST - 6

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

In [2]:
```
df=pd.read_csv(r"C:\Users\BHOOMISH\Downloads\C6_bmi.csv")
df
```

Out[2]:

	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 [3]:
```
df.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 [4]:
```
df=df.dropna()
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Gender    0
Height    0
Weight    0
Index     0
dtype: int64
```

In [6]:
```
df.describe()
```

Out[6]:

	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

In [8]:
```
df['Gender'].value_counts()
```

Out[8]:
```
Female    255
Male      245
Name: Gender, dtype: int64
```

In [9]:
```
g1={"Gender":{'Female':1,'Male':2}}
df=df.replace(g1)
print(df)
```

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

[500 rows x 4 columns]
```

In [10]:
```
x=df.drop("Gender",axis=1)
y=df["Gender"]
```

In [11]:
```
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 [12]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[12]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [13]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [14]:
```
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[14]:
```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

In [15]:
```
grid_search.best_score_
```

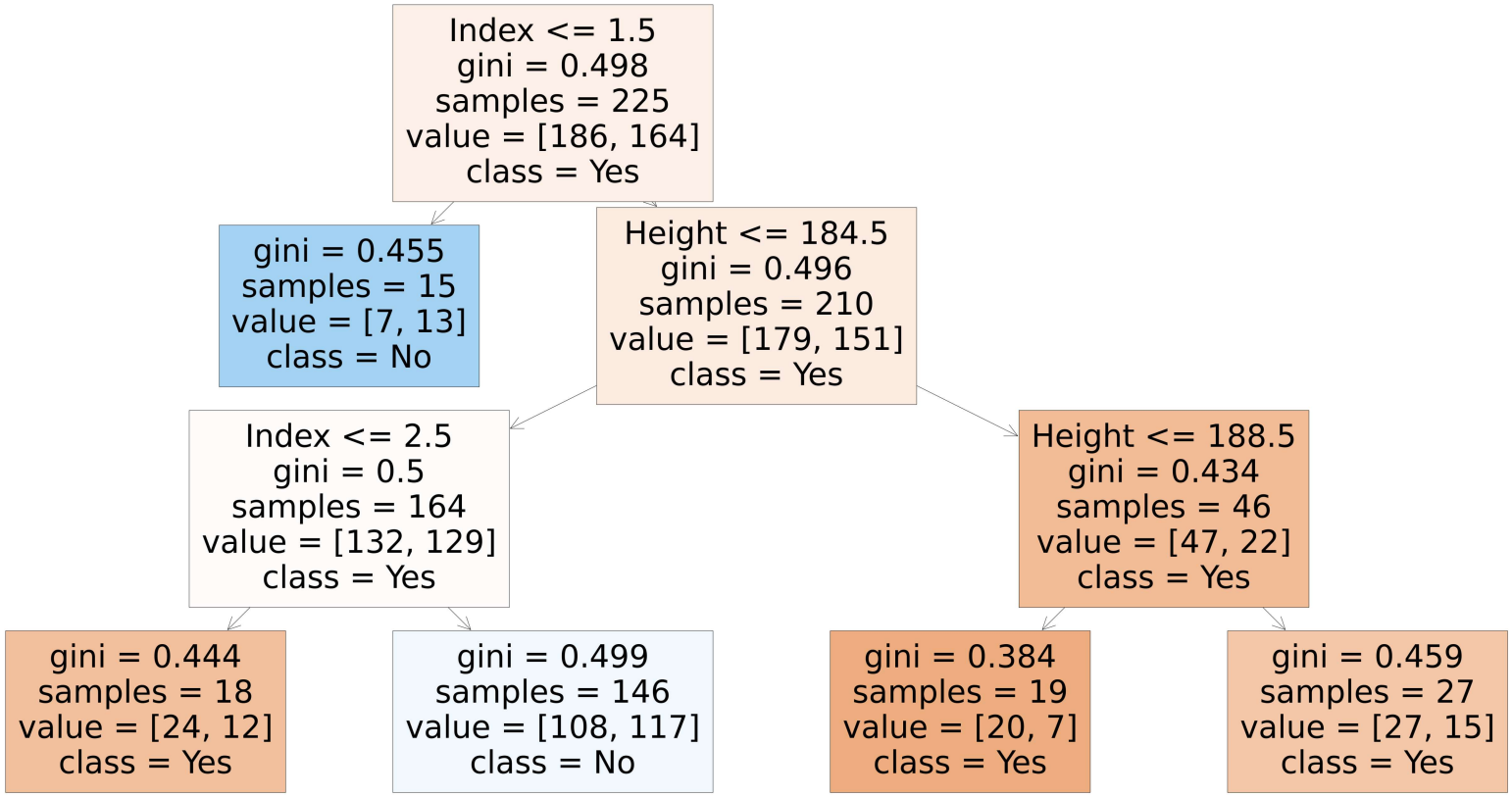
Out[15]:
```
0.5485714285714286
```

In [16]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [17]:
```
rfc_best=grid_search.best_estimator_
```

In [18]:
```
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'],filled=True)
```

Out[18]:
```
[Text(0.375, 0.875, 'Index <= 1.5\ngini = 0.498\nsamples = 225\nvalue = [186, 164]\nclass = Yes'),
 Text(0.25, 0.625, 'gini = 0.455\nsamples = 15\nvalue = [7, 13]\nclass = No'),
 Text(0.5, 0.625, 'Height <= 184.5\ngini = 0.496\nsamples = 210\nvalue = [179, 151]\nclass = Yes'),
 Text(0.25, 0.375, 'Index <= 2.5\ngini = 0.5\nsamples = 164\nvalue = [132, 129]\nclass = Yes'),
 Text(0.125, 0.125, 'gini = 0.444\nsamples = 18\nvalue = [24, 12]\nclass = Yes'),
 Text(0.375, 0.125, 'gini = 0.499\nsamples = 146\nvalue = [108, 117]\nclass = No'),
 Text(0.75, 0.375, 'Height <= 188.5\ngini = 0.434\nsamples = 46\nvalue = [47, 22]\nclass = Yes'),
 Text(0.625, 0.125, 'gini = 0.384\nsamples = 19\nvalue = [20, 7]\nclass = Yes'),
 Text(0.875, 0.125, 'gini = 0.459\nsamples = 27\nvalue = [27, 15]\nclass = Yes')]
```



In [ ]: