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

In [2]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\bm.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['Gender'].value_counts()
```

Out[3]:
```
Female    255
Male      245
Name: Gender, dtype: int64
```

In [4]:
```
df['Gender'].value_counts()
```

Out[4]:
```
Female    255
Male      245
Name: Gender, dtype: int64
```

In [5]:
```
x=df.drop('Gender',axis=1)
y=df['Gender']
```

In [6]:
```
g1={"Gender":{'Male':1,'Female':2}}
df=df.replace(g1)
print(df)
```

```
     Gender  Height  Weight  Index
0         1     174      96      4
```

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

[500 rows x 4 columns]
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.70)
```

In [8]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[8]:
```
RandomForestClassifier()
```

In [9]:
```
parameters= {
    "max_depth":[1,2,3,4,5],
    "min_samples_leaf":[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [10]:
```
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[10]:
```
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 [11]:
```
grid_search.best_score_
```

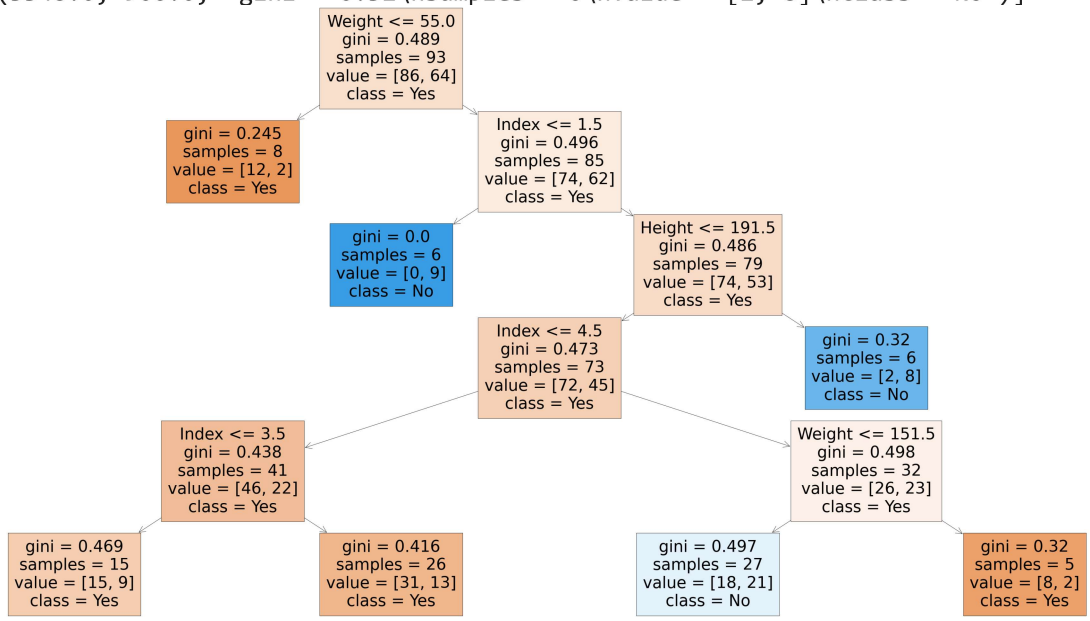
Out[11]:
```
0.5866666666666667
```

In [12]:
```
rfc_best=grid_search.best_estimator_
```

In [13]:
```
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'],fill
```

Out[13]:
```
[Text(1674.0, 1993.2, 'Weight <= 55.0\ngini = 0.489\nsamples = 93\nvalue = [86, 64]\ncla
ss = Yes'),
 Text(1116.0, 1630.8000000000002, 'gini = 0.245\nsamples = 8\nvalue = [12, 2]\nclass = Y
es'),
 Text(2232.0, 1630.8000000000002, 'Index <= 1.5\ngini = 0.496\nsamples = 85\nvalue = [7
```

```
4, 62]\nclass = Yes'),
 Text(1674.0, 1268.4, 'gini = 0.0\nsamples = 6\nvalue = [0, 9]\nclass = No'),
 Text(2790.0, 1268.4, 'Height <= 191.5\ngini = 0.486\nsamples = 79\nvalue = [74, 53]\ncl
ass = Yes'),
 Text(2232.0, 906.0, 'Index <= 4.5\ngini = 0.473\nsamples = 73\nvalue = [72, 45]\nclass
= Yes'),
 Text(1116.0, 543.5999999999999, 'Index <= 3.5\ngini = 0.438\nsamples = 41\nvalue = [46,
22]\nclass = Yes'),
 Text(558.0, 181.19999999999982, 'gini = 0.469\nsamples = 15\nvalue = [15, 9]\nclass = Y
es'),
 Text(1674.0, 181.19999999999982, 'gini = 0.416\nsamples = 26\nvalue = [31, 13]\nclass =
Yes'),
 Text(3348.0, 543.5999999999999, 'Weight <= 151.5\ngini = 0.498\nsamples = 32\nvalue =
[26, 23]\nclass = Yes'),
 Text(2790.0, 181.19999999999982, 'gini = 0.497\nsamples = 27\nvalue = [18, 21]\nclass =
No'),
 Text(3906.0, 181.19999999999982, 'gini = 0.32\nsamples = 5\nvalue = [8, 2]\nclass = Ye
s'),
 Text(3348.0, 906.0, 'gini = 0.32\nsamples = 6\nvalue = [2, 8]\nclass = No')]
```



In [ ]: