

```
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"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()
```

```
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(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_
```

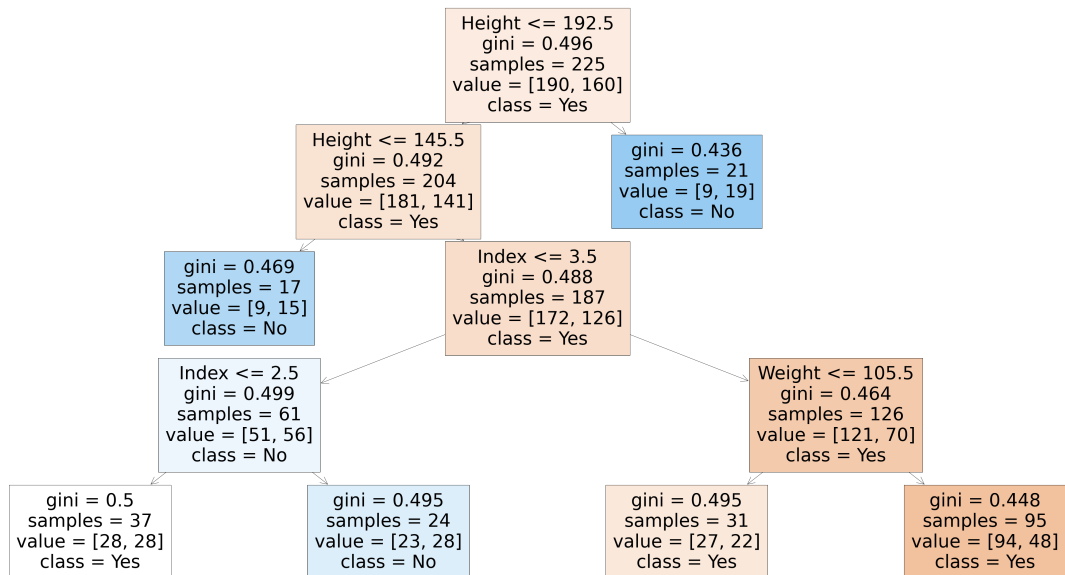
```
Out[15]: 0.5428571428571429
```

```
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'])
```

```
Out[18]: [Text(2232.0, 1956.96, 'Height <= 192.5\ngini = 0.496\nsamples = 225\nvalue = [190, 160]\n\nclass = Yes'),
Text(1674.0, 1522.0800000000002, 'Height <= 145.5\ngini = 0.492\nsamples = 204\nvalue = [181, 141]\n\nclass = Yes'),
Text(1116.0, 1087.2, 'gini = 0.469\nsamples = 17\nvalue = [9, 15]\n\nclass = No'),
Text(2232.0, 1087.2, 'Index <= 3.5\ngini = 0.488\nsamples = 187\nvalue = [172, 126]\n\nclass = Yes'),
Text(1116.0, 652.3200000000002, 'Index <= 2.5\ngini = 0.499\nsamples = 61\nvalue = [51, 56]\n\nclass = No'),
Text(558.0, 217.44000000000005, 'gini = 0.5\nsamples = 37\nvalue = [28, 28]\n\nclass = Yes'),
Text(1674.0, 217.44000000000005, 'gini = 0.495\nsamples = 24\nvalue = [23, 28]\n\nclass = No'),
Text(3348.0, 652.3200000000002, 'Weight <= 105.5\ngini = 0.464\nsamples = 126\nvalue = [121, 70]\n\nclass = Yes'),
Text(2790.0, 217.44000000000005, 'gini = 0.495\nsamples = 31\nvalue = [27, 22]\n\nclass = Yes'),
Text(3906.0, 217.44000000000005, 'gini = 0.448\nsamples = 95\nvalue = [94, 48]\n\nclass = Yes'),
Text(2790.0, 1522.0800000000002, 'gini = 0.436\nsamples = 21\nvalue = [9, 19]\n\nclass = No')]
```



In []: