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

In [2]:

In [3]:
```
df = pd.read_csv("C6_bmi.csv")
```

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]:

```
<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 [5]:

Out[5]:
```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

In [6]:
```
f_m=df[['Height', 'Weight', 'Index']]
```

```
In [7]:

Out[7]: (500, 3)

In [8]:

Out[8]: (500,)

In [9]:

In [10]:

In [11]: logr=LogisticRegression()

Out[11]: LogisticRegression()

In [12]:

In [13]: prediction=logr.predict(observation)

Out[13]: array(['Male'], dtype=object)

In [14]:

Out[14]: array(['Female', 'Male'], dtype=object)

In [15]:

Out[15]: 0.4428979082451251

In [16]:

Out[16]: 0.5571020917548749
```

RANDOM FOREST

```
In [17]:

Out[17]: Female    255
         Male      245
         Name: Gender, dtype: int64

In [18]: x=df[['Height', 'Weight', 'Index']]
```

In [19]:
```
g1={"Gender":{'Male':1,'Female':2}}
df=df.replace(g1)
```

Out[19]:

	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 × 4 columns

In [20]:
```
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 [21]:
```
rfc=RandomForestClassifier()
```

Out[21]:
```
RandomForestClassifier()
```

In [22]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
```

In [23]:
```
from sklearn.model_selection import GridSearchCV
grid_search =GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="ac
```

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')
```

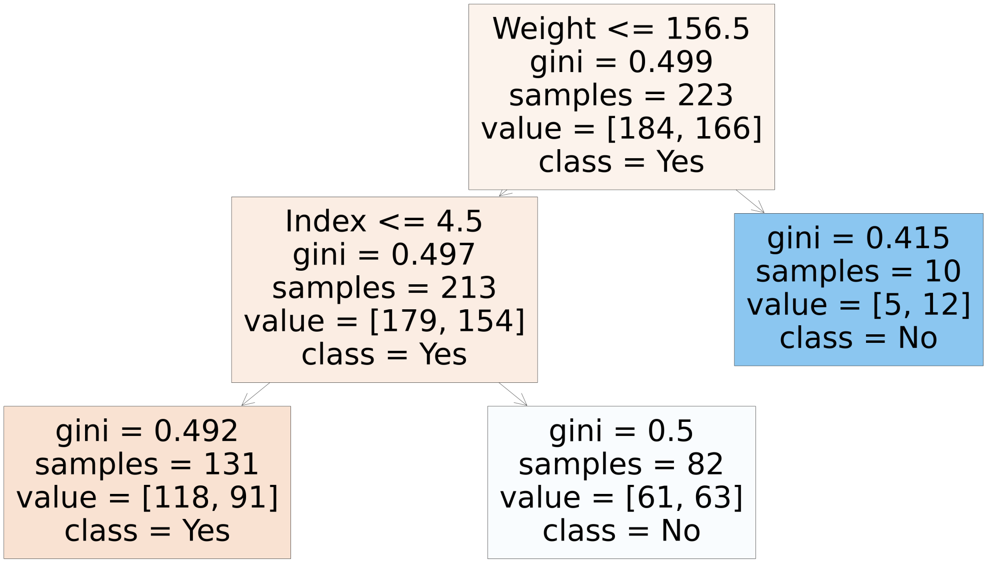
In [24]:

Out[24]:
```
0.5714285714285714
```

In [25]:

```
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=['Yes','
```

```
Out[26]: [Text(2678.3999999999996, 1812.0, 'Weight <= 156.5\ngini = 0.499\nsamples = 2
         23\nvalue = [184, 166]\nclass = Yes'),
          Text(1785.6, 1087.2, 'Index <= 4.5\ngini = 0.497\nsamples = 213\nvalue = [17
         9, 154]\nclass = Yes'),
          Text(892.8, 362.39999999999986, 'gini = 0.492\nsamples = 131\nvalue = [118,
         91]\nclass = Yes'),
          Text(2678.3999999999996, 362.39999999999986, 'gini = 0.5\nsamples = 82\nvalu
         e = [61, 63]\nclass = No'),
          Text(3571.2, 1087.2, 'gini = 0.415\nsamples = 10\nvalue = [5, 12]\nclass = N
         o')]
```


Weight <= 156.5
gini = 0.499
samples = 223
value = [184, 166]
class = Yes
Index <= 4.5
gini = 0.497
samples = 213
value = [179, 154]
class = Yes
gini = 0.415
samples = 10
value = [5, 12]
class = No
gini = 0.492
samples = 131
value = [118, 91]
class = Yes
gini = 0.5
samples = 82
value = [61, 63]
class = No