**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

In [38]:

**from** sklearn.linear\_model **import** LogisticRegression

In [39]:

df **=** pd**.**read\_csv(r"C:\Users\user\Desktop\C6\_bmi.csv")

df

Out[39]:

|  | **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 [40]:

df**.**dropna(inplace**=True**)

In [41]:

df**.**info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 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: 19.5+ KB

In [42]:

df**.**columns

Out[42]:

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

In [43]:

feature\_matrix **=** df[['Height', 'Weight', 'Index']]

target\_vector **=** df['Gender']

In [44]:

feature\_matrix**.**shape

Out[44]:

(500, 3)

In [45]:

target\_vector**.**shape

Out[45]:

(500,)

In [46]:

**from** sklearn.preprocessing **import** StandardScaler

In [47]:

fs **=** StandardScaler()**.**fit\_transform(feature\_matrix)

In [48]:

logr**=** LogisticRegression()

logr**.**fit(fs,target\_vector)

Out[48]:

LogisticRegression()

In [49]:

observation**=**df[['Height', 'Weight', 'Index']]

In [50]:

prediction **=** logr**.**predict(observation)

prediction

Out[50]:

array(['Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Male', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Male', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Male', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Male', 'Female', 'Female', 'Female', 'Female',

'Female', 'Female', 'Female'], dtype=object)

In [51]:

logr**.**classes\_

Out[51]:

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

In [52]:

logr**.**predict\_proba(observation)[0][1]

Out[52]:

0.013560340101419254

In [54]:

df['Gender']**.**value\_counts()

Out[54]:

Female 255

Male 245

Name: Gender, dtype: int64

In [55]:

x**=**df[['Height', 'Weight', 'Index']]

y**=**df['Gender']

In [56]:

g1**=**{'Gender':{'True':1,"False":2}}

df**=**df**.**replace(g1)

df

Out[56]:

|  | **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 [57]:

**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 [58]:

**from** sklearn.ensemble **import** RandomForestClassifier

rfc **=** RandomForestClassifier()

rfc**.**fit(x\_train,y\_train)

Out[58]:

RandomForestClassifier()

In [59]:

parameters **=**{'max\_depth':[1, 2, 3, 4,5],'min\_samples\_leaf':[5,10,15,20,25],

'n\_estimators':[10,20,30,40,50]}

In [60]:

**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[60]:

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 [61]:

grid\_search**.**best\_score\_

Out[61]:

0.5428571428571428

In [62]:

rfc\_best **=**grid\_search**.**best\_estimator\_

In [63]:

**from** sklearn.tree **import** plot\_tree

plt**.**figure(figsize**=**(89,40))

plot\_tree(rfc\_best**.**estimators\_[5], feature\_names**=**x**.**columns, class\_names**=**['Yes','No'], filled**=True**)

Out[63]:

[Text(2483.1, 1956.96, 'Height <= 145.5\ngini = 0.497\nsamples = 220\nvalue = [189, 161]\nclass = Yes'),

Text(2031.6272727272726, 1522.0800000000002, 'gini = 0.418\nsamples = 24\nvalue = [11, 26]\nclass = No'),

Text(2934.572727272727, 1522.0800000000002, 'Height <= 187.5\ngini = 0.491\nsamples = 196\nvalue = [178, 135]\nclass = Yes'),

Text(1805.890909090909, 1087.2, 'Weight <= 102.5\ngini = 0.479\nsamples = 154\nvalue = [150, 99]\nclass = Yes'),

Text(902.9454545454545, 652.3200000000002, 'Weight <= 62.5\ngini = 0.444\nsamples = 69\nvalue = [72, 36]\nclass = Yes'),

Text(451.47272727272724, 217.44000000000005, 'gini = 0.498\nsamples = 23\nvalue = [16, 18]\nclass = No'),

Text(1354.4181818181817, 217.44000000000005, 'gini = 0.368\nsamples = 46\nvalue = [56, 18]\nclass = Yes'),

Text(2708.8363636363633, 652.3200000000002, 'Index <= 4.5\ngini = 0.494\nsamples = 85\nvalue = [78, 63]\nclass = Yes'),

Text(2257.363636363636, 217.44000000000005, 'gini = 0.488\nsamples = 25\nvalue = [16, 22]\nclass = No'),

Text(3160.3090909090906, 217.44000000000005, 'gini = 0.479\nsamples = 60\nvalue = [62, 41]\nclass = Yes'),

Text(4063.254545454545, 1087.2, 'Height <= 192.5\ngini = 0.492\nsamples = 42\nvalue = [28, 36]\nclass = No'),

Text(3611.781818181818, 652.3200000000002, 'gini = 0.498\nsamples = 21\nvalue = [15, 17]\nclass = No'),

Text(4514.727272727272, 652.3200000000002, 'gini = 0.482\nsamples = 21\nvalue = [13, 19]\nclass = No')]

In [ ]:

