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"1_ionosphere.csv")
df
```

Out[2]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.03760	...	-0.511
0	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	...	-0.265
1	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	...	-0.402
2	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	...	0.906
3	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	...	-0.651
4	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	...	-0.015
...	...	...	...	...	...	...	...	...	...	...	...	
345	1	0	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567	0.90441	-0.04622	...	-0.042
346	1	0	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920	0.94590	0.01606	...	0.013
347	1	0	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431	0.95584	0.02446	...	0.031
348	1	0	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646	0.85746	0.00110	...	-0.020
349	1	0	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260	0.88928	-0.09139	...	-0.151

350 rows × 35 columns

In [3]:
```
df['g'].value_counts()
```

Out[3]:
```
g    224
b    126
Name: g, dtype: int64
```

In [4]:
```
x=df.drop('g',axis=1)
y=df['g']
```

In [5]:
```
g1={"g":{'g':1,'b':2}}
df=df.replace(g1)
print(df)
```

```
     1  0  0.99539  -0.05889  0.85243  0.02306  0.83398  -0.37708      1.1
\
0    1  0  1.00000  -0.18829  0.93035 -0.36156 -0.10868  -0.93597  1.00000
1    1  0  1.00000  -0.03365  1.00000  0.00485  1.00000  -0.12062  0.88965
2    1  0  1.00000  -0.45161  1.00000  1.00000  0.71216  -1.00000  0.00000
3    1  0  1.00000  -0.02401  0.94140  0.06531  0.92106  -0.23255  0.77152
4    1  0  0.02337  -0.00592 -0.09924 -0.11949 -0.00763  -0.11824  0.14706
..  .. ..      ...       ...      ...      ...      ...       ...      ...
345  1  0  0.83508   0.08298  0.73739 -0.14706  0.84349  -0.05567  0.90441
346  1  0  0.95113   0.00419  0.95183 -0.02723  0.93438  -0.01920  0.94590
347  1  0  0.94701  -0.00034  0.93207 -0.03227  0.95177  -0.03431  0.95584
348  1  0  0.90608  -0.01657  0.98122 -0.01989  0.95691  -0.03646  0.85746
349  1  0  0.84710   0.13533  0.73638 -0.06151  0.87873   0.08260  0.88928

     0.03760  ...  -0.51171  0.41078  -0.46168  0.21266  -0.34090  0.42267
\
0   -0.04549  ...  -0.26569 -0.20468  -0.18401 -0.19040  -0.11593 -0.16626
1    0.01198  ...  -0.40220  0.58984  -0.22145  0.43100  -0.17365  0.60436
2    0.00000  ...   0.90695  0.51613   1.00000  1.00000  -0.20099  0.25682
```

In [6]:
```
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 [7]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[7]:
```
RandomForestClassifier()
```

In [8]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]}
```

In [9]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[9]:
```
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 [10]:
```
grid_search.best_score_
```

Out[10]:
```
0.9344262295081968
```

In [11]:
```
rfc_best=grid_search.best_estimator_
```

In [12]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','N
```

Out[12]:
```
[Text(1674.0, 1993.2, '-0.46168 <= -0.956\ngini = 0.476\nsamples = 162\nvalue
= [95, 149]\nclass = No'),
 Text(744.0, 1630.8000000000002, '0.85243 <= 0.5\ngini = 0.053\nsamples = 21
\nvalue = [36, 1]\nclass = Yes'),
 Text(372.0, 1268.4, 'gini = 0.0\nsamples = 15\nvalue = [28, 0]\nclass = Ye
s'),
 Text(1116.0, 1268.4, 'gini = 0.198\nsamples = 6\nvalue = [8, 1]\nclass = Ye
s'),
 Text(2604.0, 1630.8000000000002, '0.85243 <= 0.232\ngini = 0.408\nsamples =
141\nvalue = [59, 148]\nclass = No'),
 Text(1860.0, 1268.4, '0.60536 <= 0.094\ngini = 0.105\nsamples = 25\nvalue =
[34, 2]\nclass = Yes'),
 Text(1488.0, 906.0, 'gini = 0.0\nsamples = 20\nvalue = [30, 0]\nclass = Ye
s'),
 Text(2232.0, 906.0, 'gini = 0.444\nsamples = 5\nvalue = [4, 2]\nclass = Ye
s'),
 Text(3348.0, 1268.4, '-0.44945 <= -0.608\ngini = 0.25\nsamples = 116\nvalue
= [25, 146]\nclass = No'),
 Text(2976.0, 906.0, 'gini = 0.0\nsamples = 8\nvalue = [12, 0]\nclass = Ye
s'),
 Text(3720.0, 906.0, '-0.45300 <= 0.951\ngini = 0.15\nsamples = 108\nvalue =
[13, 146]\nclass = No'),
 Text(3348.0, 543.5999999999999, '0.56811 <= 0.998\ngini = 0.089\nsamples = 1
02\nvalue = [7, 143]\nclass = No'),
 Text(2976.0, 181.19999999999982, 'gini = 0.029\nsamples = 93\nvalue = [2, 13
5]\nclass = No'),
 Text(3720.0, 181.19999999999982, 'gini = 0.473\nsamples = 9\nvalue = [5, 8]
\nclass = No'),
 Text(4092.0, 543.5999999999999, 'gini = 0.444\nsamples = 6\nvalue = [6, 3]\n
class = Yes')]
```