RANDOM FOREST - 1

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

In [3]:
```
df=pd.read_csv(r"C:\Users\BHOOMISH\Downloads\C1_ionosphere.CSV")
df
```

Out[3]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.03760	...	-0.51171	0.41078	-0.46168	0.21266	-0.340
0	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	...	-0.26569	-0.20468	-0.18401	-0.19040	-0.115
1	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	...	-0.40220	0.58984	-0.22145	0.43100	-0.173
2	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	...	0.90695	0.51613	1.00000	1.00000	-0.20(
3	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	...	-0.65158	0.13290	-0.53206	0.02431	-0.62'
4	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	...	-0.01535	-0.03240	0.09223	-0.07859	0.007
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
345	1	0	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567	0.90441	-0.04622	...	-0.04202	0.83479	0.00123	1.00000	0.128
346	1	0	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920	0.94590	0.01606	...	0.01361	0.93522	0.04925	0.93159	0.08'
347	1	0	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431	0.95584	0.02446	...	0.03193	0.92489	0.02542	0.92120	0.02'
348	1	0	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646	0.85746	0.00110	...	-0.02099	0.89147	-0.07760	0.82983	-0.17'
349	1	0	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260	0.88928	-0.09139	...	-0.15114	0.81147	-0.04822	0.78207	-0.007

350 rows × 35 columns

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

Out[4]:
```
g    224
b    126
Name: g, dtype: int64
```

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

In [6]:
```
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
3   -0.16399  ...  -0.65158  0.13290  -0.53206  0.02431  -0.62197 -0.05707
4    0.06637  ...  -0.01535 -0.03240   0.09223 -0.07859   0.00732  0.00000
```

In [7]:
```
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 [8]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [9]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[9]:
```
RandomForestClassifier
RandomForestClassifier()
```

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

In [11]:
```
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[11]:
```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

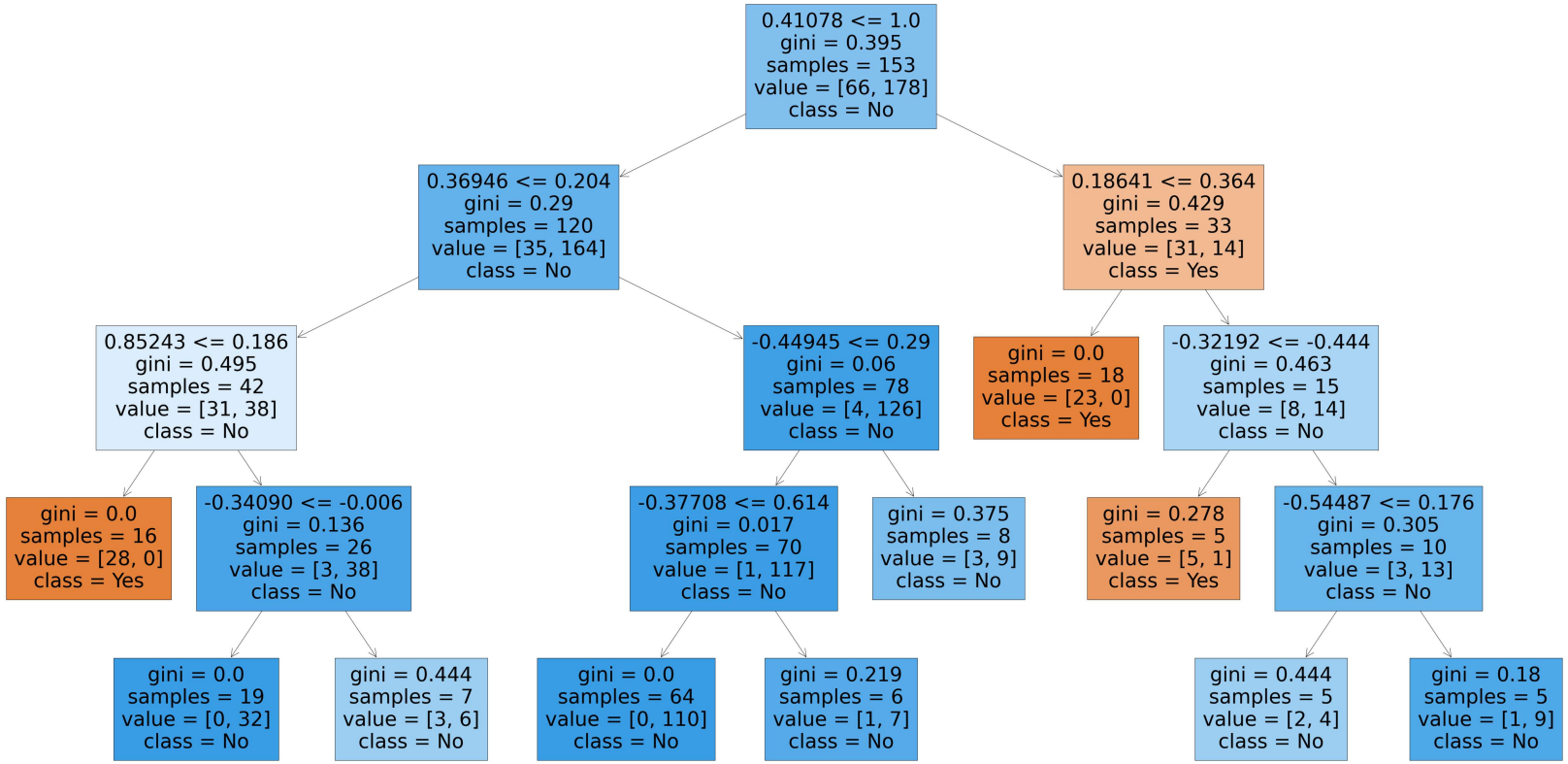
In [12]:
```
grid_search.best_score_
```

Out[12]:
```
0.9344262295081968
```

In [13]:
```
rfc_best=grid_search.best_estimator_
```

In [14]:
```
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'],filled=True)
```

Out[14]:
```
[Text(0.5333333333333333, 0.9, '0.41078 <= 1.0\ngini = 0.395\nsamples = 153\nvalue = [66, 178]\nclass = N
o'),
 Text(0.3333333333333333, 0.7, '0.36946 <= 0.204\ngini = 0.29\nsamples = 120\nvalue = [35, 164]\nclass = N
o'),
 Text(0.13333333333333333, 0.5, '0.85243 <= 0.186\ngini = 0.495\nsamples = 42\nvalue = [31, 38]\nclass = N
o'),
 Text(0.06666666666666667, 0.3, 'gini = 0.0\nsamples = 16\nvalue = [28, 0]\nclass = Yes'),
 Text(0.2, 0.3, '-0.34090 <= -0.006\ngini = 0.136\nsamples = 26\nvalue = [3, 38]\nclass = No'),
 Text(0.13333333333333333, 0.1, 'gini = 0.0\nsamples = 19\nvalue = [0, 32]\nclass = No'),
 Text(0.26666666666666666, 0.1, 'gini = 0.444\nsamples = 7\nvalue = [3, 6]\nclass = No'),
 Text(0.5333333333333333, 0.5, '-0.44945 <= 0.29\ngini = 0.06\nsamples = 78\nvalue = [4, 126]\nclass = No'),
 Text(0.4666666666666667, 0.3, '-0.37708 <= 0.614\ngini = 0.017\nsamples = 70\nvalue = [1, 117]\nclass = N
o'),
 Text(0.4, 0.1, 'gini = 0.0\nsamples = 64\nvalue = [0, 110]\nclass = No'),
 Text(0.5333333333333333, 0.1, 'gini = 0.219\nsamples = 6\nvalue = [1, 7]\nclass = No'),
 Text(0.6, 0.3, 'gini = 0.375\nsamples = 8\nvalue = [3, 9]\nclass = No'),
 Text(0.7333333333333333, 0.7, '0.18641 <= 0.364\ngini = 0.429\nsamples = 33\nvalue = [31, 14]\nclass = Ye
s'),
 Text(0.6666666666666666, 0.5, 'gini = 0.0\nsamples = 18\nvalue = [23, 0]\nclass = Yes'),
 Text(0.8, 0.5, '-0.32192 <= -0.444\ngini = 0.463\nsamples = 15\nvalue = [8, 14]\nclass = No'),
 Text(0.7333333333333333, 0.3, 'gini = 0.278\nsamples = 5\nvalue = [5, 1]\nclass = Yes'),
 Text(0.8666666666666667, 0.3, '-0.54487 <= 0.176\ngini = 0.305\nsamples = 10\nvalue = [3, 13]\nclass = N
o'),
 Text(0.8, 0.1, 'gini = 0.444\nsamples = 5\nvalue = [2, 4]\nclass = No'),
 Text(0.9333333333333333, 0.1, 'gini = 0.18\nsamples = 5\nvalue = [1, 9]\nclass = No')]
```


0.41078 <= 1.0
gini = 0.395
samples = 153
value = [66, 178]
class = No
0.36946 <= 0.204
gini = 0.29
samples = 120
value = [35, 164]
class = No
0.18641 <= 0.364
gini = 0.429
samples = 33
value = [31, 14]
class = Yes
0.85243 <= 0.186
gini = 0.495
samples = 42
value = [31, 38]
class = No
-0.44945 <= 0.29
gini = 0.06
samples = 78
value = [4, 126]
class = No
gini = 0.0
samples = 18
value = [23, 0]
class = Yes
-0.32192 <= -0.444
gini = 0.463
samples = 15
value = [8, 14]
class = No
gini = 0.0
samples = 16
value = [28, 0]
class = Yes
-0.34090 <= -0.006
gini = 0.136
samples = 26
value = [3, 38]
class = No
-0.37708 <= 0.614
gini = 0.017
samples = 70
value = [1, 117]
class = No
gini = 0.375
samples = 8
value = [3, 9]
class = No
gini = 0.278
samples = 5
value = [5, 1]
class = Yes
-0.54487 <= 0.176
gini = 0.305
samples = 10
value = [3, 13]
class = No
gini = 0.0
samples = 19
value = [0, 32]
class = No
gini = 0.444
samples = 7
value = [3, 6]
class = No
gini = 0.0
samples = 64
value = [0, 110]
class = No
gini = 0.219
samples = 6
value = [1, 7]
class = No
gini = 0.444
samples = 5
value = [2, 4]
class = No
gini = 0.18
samples = 5
value = [1, 9]
class = No