In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
```

DataSet C1

In [2]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\1_ionosphere.csv")
a
```

Out[2]:

	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
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
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
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
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
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
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
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
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
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
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

350 rows × 35 columns

In [3]:

```
a['g'].value_counts()
```

Out[3]:

```
g    224
b    126
Name: g, dtype: int64
```

In [4]:

```
x=a.drop('g',axis=1)
y=a['g']
```

In [5]:

```
g1={"g":{'g':1,'b':2}}
a=a.replace(g1)
print(a)
```

```
     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
..       ...  ...       ...      ...       ...      ...       ...      ...
345 -0.04622  ...  -0.04202  0.83479   0.00123  1.00000   0.12815  0.86660
346  0.01606  ...   0.01361  0.93522   0.04925  0.93159   0.08168  0.94066
347  0.02446  ...   0.03193  0.92489   0.02542  0.92120   0.02242  0.92459
348  0.00110  ...  -0.02099  0.89147  -0.07760  0.82983  -0.17238  0.96022
349 -0.09139  ...  -0.15114  0.81147  -0.04822  0.78207  -0.00703  0.75747

     -0.54487  0.18641  -0.45300  g
0    -0.06288 -0.13738  -0.02447  2
1    -0.24180  0.56045  -0.38238  1
2     1.00000 -0.32382   1.00000  2
3    -0.59573 -0.04608  -0.65697  1
4     0.00000 -0.00039   0.12011  2
..        ...      ...       ... ..
345  -0.10714  0.90546  -0.04307  1
346  -0.00035  0.91483   0.04712  1
347   0.00442  0.92697  -0.00577  1
348  -0.03757  0.87403  -0.16243  1
349  -0.06678  0.85764  -0.06151  1

[350 rows x 35 columns]
```

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="accuracy")
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]:

```
0.9385245901639344
```

In [11]:

```
rfc_best=grid_search.best_estimator_
```

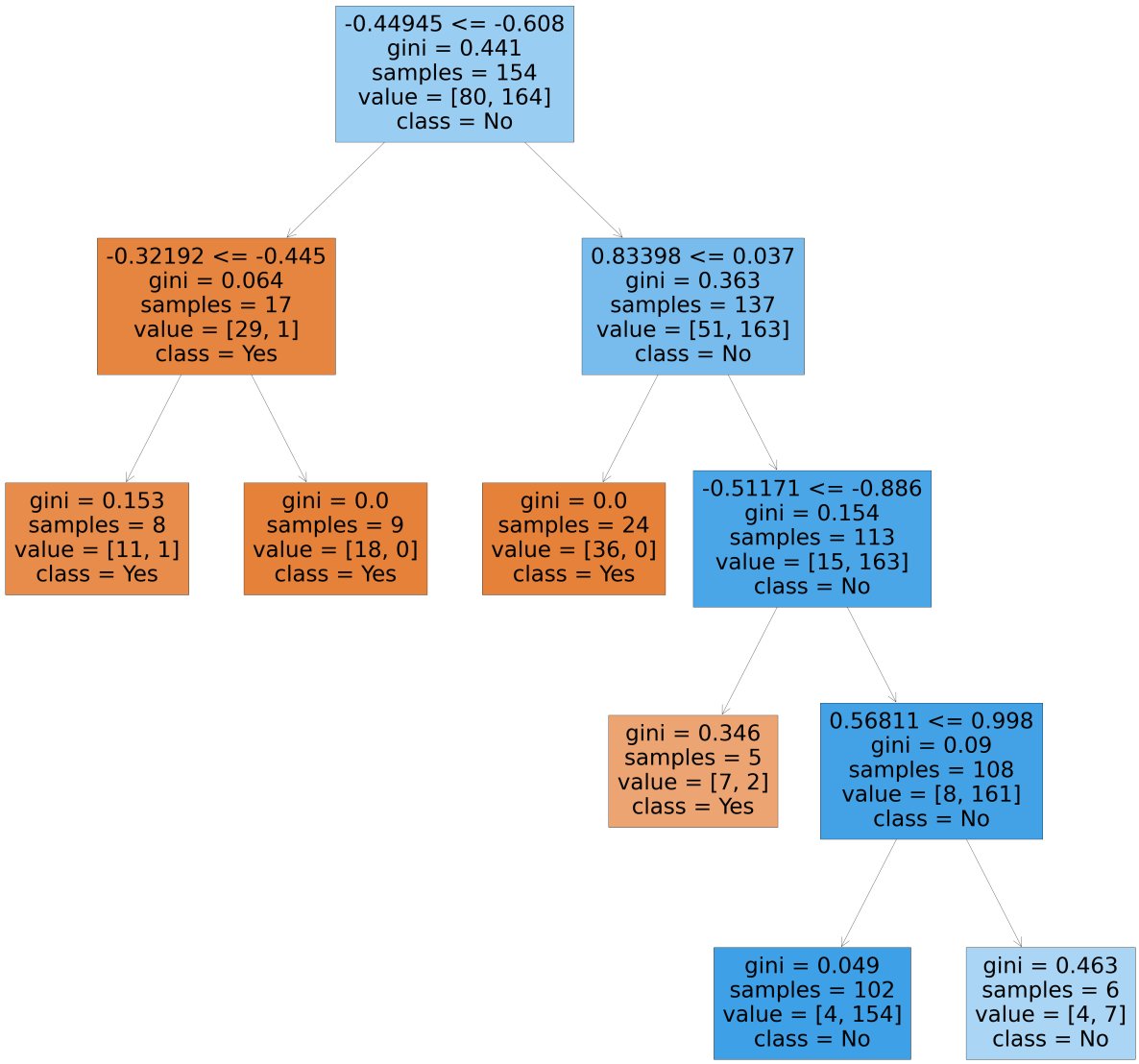
In [12]:

```
from sklearn.tree import plot_tree

plt.figure(figsize=(80,80))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[12]:

```
[Text(1785.6, 3913.92, '-0.44945 <= -0.608\ngini = 0.441\nsamples = 154\nvalue = [80, 164]\nclass = No'),
 Text(892.8, 3044.1600000000003, '-0.32192 <= -0.445\ngini = 0.064\nsamples = 17\nvalue = [29, 1]\nclass
= Yes'),
 Text(446.4, 2174.4, 'gini = 0.153\nsamples = 8\nvalue = [11, 1]\nclass = Yes'),
 Text(1339.1999999999998, 2174.4, 'gini = 0.0\nsamples = 9\nvalue = [18, 0]\nclass = Yes'),
 Text(2678.3999999999996, 3044.1600000000003, '0.83398 <= 0.037\ngini = 0.363\nsamples = 137\nvalue = [5
1, 163]\nclass = No'),
 Text(2232.0, 2174.4, 'gini = 0.0\nsamples = 24\nvalue = [36, 0]\nclass = Yes'),
 Text(3124.7999999999997, 2174.4, '-0.51171 <= -0.886\ngini = 0.154\nsamples = 113\nvalue = [15, 163]\ncl
ass = No'),
 Text(2678.3999999999996, 1304.6400000000003, 'gini = 0.346\nsamples = 5\nvalue = [7, 2]\nclass = Yes'),
 Text(3571.2, 1304.6400000000003, '0.56811 <= 0.998\ngini = 0.09\nsamples = 108\nvalue = [8, 161]\nclass
= No'),
 Text(3124.7999999999997, 434.8800000000001, 'gini = 0.049\nsamples = 102\nvalue = [4, 154]\nclass = N
o'),
 Text(4017.6, 434.8800000000001, 'gini = 0.463\nsamples = 6\nvalue = [4, 7]\nclass = No')]
```



DataSet C2

In [13]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C2_test.gender_submission.csv")
a
```

Out[13]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 11 columns

In [14]:

```
a=a.fillna(value=54)
a
```

Out[14]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	54	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	54	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	54	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	54	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	54	S
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	54.0	0	0	A.5. 3236	8.0500	54	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	54	S
416	1308	3	Ware, Mr. Frederick	male	54.0	0	0	359309	8.0500	54	S
417	1309	3	Peter, Master. Michael J	male	54.0	1	1	2668	22.3583	54	C

418 rows × 11 columns

In [15]:

```
a=a[['PassengerId','Pclass','Age','SibSp','Parch','Fare','Embarked']]
a
```

Out[15]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Embarked
0	892	3	34.5	0	0	7.8292	Q
1	893	3	47.0	1	0	7.0000	S
2	894	2	62.0	0	0	9.6875	Q
3	895	3	27.0	0	0	8.6625	S
4	896	3	22.0	1	1	12.2875	S
...	...	...	...	...	...	...	...
413	1305	3	54.0	0	0	8.0500	S
414	1306	1	39.0	0	0	108.9000	C
415	1307	3	38.5	0	0	7.2500	S
416	1308	3	54.0	0	0	8.0500	S
417	1309	3	54.0	1	1	22.3583	C

418 rows × 7 columns

In [16]:

```
a['Embarked'].value_counts()
```

Out[16]:

```
S    270
C    102
Q     46
Name: Embarked, dtype: int64
```

In [17]:

```
x=a.drop('Embarked',axis=1)
y=a['Embarked']
```

In [18]:

```
g1={"Embarked":{'S':1,'C':2,'Q':3}}
a=a.replace(g1)
print(a)
```

```
     PassengerId  Pclass   Age  SibSp  Parch      Fare  Embarked
0            892       3  34.5      0      0    7.8292         3
1            893       3  47.0      1      0    7.0000         1
2            894       2  62.0      0      0    9.6875         3
3            895       3  27.0      0      0    8.6625         1
4            896       3  22.0      1      1   12.2875         1
..           ...     ...   ...    ...    ...       ...       ...
413         1305       3  54.0      0      0    8.0500         1
414         1306       1  39.0      0      0  108.9000         2
415         1307       3  38.5      0      0    7.2500         1
416         1308       3  54.0      0      0    8.0500         1
417         1309       3  54.0      1      1   22.3583         2

[418 rows x 7 columns]
```

In [19]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [20]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[20]:

```
RandomForestClassifier()
```

In [21]:

```
parameters={'max_depth':[1,8,3,9,6],
            'min_samples_leaf':[5,8,13,46,25],
            'n_estimators':[10,60,14,53,50]
            }
```

In [22]:

```
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[22]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 8, 3, 9, 6],
                         'min_samples_leaf': [5, 8, 13, 46, 25],
                         'n_estimators': [10, 60, 14, 53, 50]},
             scoring='accuracy')
```

In [23]:

```
grid_search.best_score_
```

Out[23]:

```
0.7397260273972603
```

In [24]:

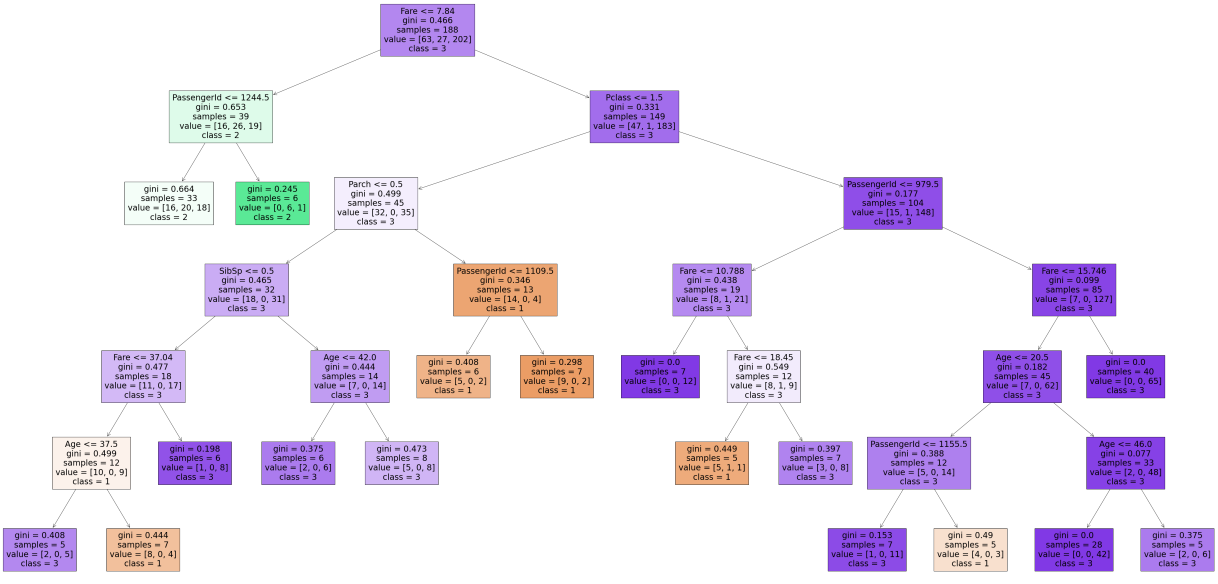
```
rfc_best=grid_search.best_estimator_
```

In [25]:

```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[4],feature_names=x.columns,class_names=['1','2','3'],filled=True)
```

Out[25]:

```
[Text(1581.0, 2019.0857142857144, 'Fare <= 7.84\ngini = 0.466\nsamples = 188\nvalue = [63, 27, 202]\nclas
s = 3'),
 Text(837.0, 1708.457142857143, 'PassengerId <= 1244.5\ngini = 0.653\nsamples = 39\nvalue = [16, 26, 19]
\nclass = 2'),
 Text(651.0, 1397.8285714285716, 'gini = 0.664\nsamples = 33\nvalue = [16, 20, 18]\nclass = 2'),
 Text(1023.0, 1397.8285714285716, 'gini = 0.245\nsamples = 6\nvalue = [0, 6, 1]\nclass = 2'),
 Text(2325.0, 1708.457142857143, 'Pclass <= 1.5\ngini = 0.331\nsamples = 149\nvalue = [47, 1, 183]\nclass
= 3'),
 Text(1395.0, 1397.8285714285716, 'Parch <= 0.5\ngini = 0.499\nsamples = 45\nvalue = [32, 0, 35]\nclass =
3'),
 Text(930.0, 1087.2, 'SibSp <= 0.5\ngini = 0.465\nsamples = 32\nvalue = [18, 0, 31]\nclass = 3'),
 Text(558.0, 776.5714285714287, 'Fare <= 37.04\ngini = 0.477\nsamples = 18\nvalue = [11, 0, 17]\nclass =
3'),
 Text(372.0, 465.9428571428573, 'Age <= 37.5\ngini = 0.499\nsamples = 12\nvalue = [10, 0, 9]\nclass =
1'),
 Text(186.0, 155.3142857142857, 'gini = 0.408\nsamples = 5\nvalue = [2, 0, 5]\nclass = 3'),
 Text(558.0, 155.3142857142857, 'gini = 0.444\nsamples = 7\nvalue = [8, 0, 4]\nclass = 1'),
 Text(744.0, 465.9428571428573, 'gini = 0.198\nsamples = 6\nvalue = [1, 0, 8]\nclass = 3'),
 Text(1302.0, 776.5714285714287, 'Age <= 42.0\ngini = 0.444\nsamples = 14\nvalue = [7, 0, 14]\nclass =
3'),
 Text(1116.0, 465.9428571428573, 'gini = 0.375\nsamples = 6\nvalue = [2, 0, 6]\nclass = 3'),
 Text(1488.0, 465.9428571428573, 'gini = 0.473\nsamples = 8\nvalue = [5, 0, 8]\nclass = 3'),
 Text(1860.0, 1087.2, 'PassengerId <= 1109.5\ngini = 0.346\nsamples = 13\nvalue = [14, 0, 4]\nclass =
1'),
 Text(1674.0, 776.5714285714287, 'gini = 0.408\nsamples = 6\nvalue = [5, 0, 2]\nclass = 1'),
 Text(2046.0, 776.5714285714287, 'gini = 0.298\nsamples = 7\nvalue = [9, 0, 2]\nclass = 1'),
 Text(3255.0, 1397.8285714285716, 'PassengerId <= 979.5\ngini = 0.177\nsamples = 104\nvalue = [15, 1, 14
8]\nclass = 3'),
 Text(2604.0, 1087.2, 'Fare <= 10.788\ngini = 0.438\nsamples = 19\nvalue = [8, 1, 21]\nclass = 3'),
 Text(2418.0, 776.5714285714287, 'gini = 0.0\nsamples = 7\nvalue = [0, 0, 12]\nclass = 3'),
 Text(2790.0, 776.5714285714287, 'Fare <= 18.45\ngini = 0.549\nsamples = 12\nvalue = [8, 1, 9]\nclass =
3'),
 Text(2604.0, 465.9428571428573, 'gini = 0.449\nsamples = 5\nvalue = [5, 1, 1]\nclass = 1'),
 Text(2976.0, 465.9428571428573, 'gini = 0.397\nsamples = 7\nvalue = [3, 0, 8]\nclass = 3'),
 Text(3906.0, 1087.2, 'Fare <= 15.746\ngini = 0.099\nsamples = 85\nvalue = [7, 0, 127]\nclass = 3'),
 Text(3720.0, 776.5714285714287, 'Age <= 20.5\ngini = 0.182\nsamples = 45\nvalue = [7, 0, 62]\nclass =
3'),
 Text(3348.0, 465.9428571428573, 'PassengerId <= 1155.5\ngini = 0.388\nsamples = 12\nvalue = [5, 0, 14]\n
class = 3'),
 Text(3162.0, 155.3142857142857, 'gini = 0.153\nsamples = 7\nvalue = [1, 0, 11]\nclass = 3'),
 Text(3534.0, 155.3142857142857, 'gini = 0.49\nsamples = 5\nvalue = [4, 0, 3]\nclass = 1'),
 Text(4092.0, 465.9428571428573, 'Age <= 46.0\ngini = 0.077\nsamples = 33\nvalue = [2, 0, 48]\nclass =
3'),
 Text(3906.0, 155.3142857142857, 'gini = 0.0\nsamples = 28\nvalue = [0, 0, 42]\nclass = 3'),
 Text(4278.0, 155.3142857142857, 'gini = 0.375\nsamples = 5\nvalue = [2, 0, 6]\nclass = 3'),
 Text(4092.0, 776.5714285714287, 'gini = 0.0\nsamples = 40\nvalue = [0, 0, 65]\nclass = 3')]
```



DataSet C3

In [26]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C3_bot_detection_data.csv")
a
```

Out[26]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Hashtags
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinston	2020-05-11 15:29:50	NaN
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston	2022-11-26 05:18:10	both live
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfurt	2022-08-08 03:16:54	phone ahead
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezberg	2021-08-14 22:27:05	ever quickly new I
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachoville	2020-04-13 21:24:21	foreign mention
...	...	...	...	...	...	...	...	...	...	...	...
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Lake Kimberlyburgh	2023-04-20 11:06:26	teach quality ten education any
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury	2022-10-18 03:57:35	add walk among believe
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort	2020-07-08 03:54:08	onto admit artist first
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Stephenside	2022-03-22 12:13:44	star
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Novakberg	2022-12-03 06:11:07	home

50000 rows × 11 columns

In [27]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   User ID         50000 non-null  int64 
 1   Username        50000 non-null  object
 2   Tweet           50000 non-null  object
 3   Retweet Count   50000 non-null  int64 
 4   Mention Count   50000 non-null  int64 
 5   Follower Count  50000 non-null  int64 
 6   Verified        50000 non-null  bool  
 7   Bot Label       50000 non-null  int64 
 8   Location        50000 non-null  object
 9   Created At      50000 non-null  object
 10  Hashtags        41659 non-null  object
dtypes: bool(1), int64(5), object(5)
memory usage: 3.9+ MB
```

In [28]:

```
a=a.dropna(axis=1)
a
```

Out[28]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinston	2020-05-11 15:29:50
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston	2022-11-26 05:18:10
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harrisonfurt	2022-08-08 03:16:54
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martinezberg	2021-08-14 22:27:05
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camachoville	2020-04-13 21:24:21
...	...	...	...	...	...	...	...	...	...	...
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Lake Kimberlyburgh	2023-04-20 11:06:26
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury	2022-10-18 03:57:35
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort	2020-07-08 03:54:08
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Stephenside	2022-03-22 12:13:44
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Novakberg	2022-12-03 06:11:07

50000 rows × 10 columns

In [29]:

```
a.columns
```

Out[29]:

```
Index(['User ID', 'Username', 'Tweet', 'Retweet Count', 'Mention Count',
       'Follower Count', 'Verified', 'Bot Label', 'Location', 'Created At'],
      dtype='object')
```

In [30]:

```
b=a[['User ID','Retweet Count', 'Mention Count',
       'Follower Count',  'Bot Label','Verified']]
b
```

Out[30]:

	User ID	Retweet Count	Mention Count	Follower Count	Bot Label	Verified
0	132131	85	1	2353	1	False
1	289683	55	5	9617	0	True
2	779715	6	2	4363	0	True
3	696168	54	5	2242	1	True
4	704441	26	3	8438	1	False
...	...	...	...	...	...	...
49995	491196	64	0	9911	1	True
49996	739297	18	5	9900	1	False
49997	674475	43	3	6313	1	True
49998	167081	45	1	6343	0	False
49999	311204	91	4	4006	0	False

50000 rows × 6 columns

In [31]:

```
b['Verified'].value_counts()
```

Out[31]:

```
True     25004
False    24996
Name: Verified, dtype: int64
```

In [32]:

```
x=b.drop('Verified',axis=1)
y=b['Verified']
```

In [33]:

```
g1={"Verified":{'True':11,'False':22}}
b=b.replace(g1)
print(b)
```

```
       User ID  Retweet Count  Mention Count  Follower Count  Bot Label  \
0       132131             85              1            2353          1
1       289683             55              5            9617          0
2       779715              6              2            4363          0
3       696168             54              5            2242          1
4       704441             26              3            8438          1
...        ...            ...            ...             ...        ...
49995   491196             64              0            9911          1
49996   739297             18              5            9900          1
49997   674475             43              3            6313          1
49998   167081             45              1            6343          0
49999   311204             91              4            4006          0

       Verified
0         False
1          True
2          True
3          True
4         False
...         ...
49995      True
49996     False
49997      True
49998     False
49999     False

[50000 rows x 6 columns]
```

In [34]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [35]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[35]:

```
RandomForestClassifier()
```

In [36]:

```
parameters={'max_depth':[3,1,7,2,9],
            'min_samples_leaf':[25,30,27,12,54],
            'n_estimators':[25,34,78,15,98]
            }
```

In [37]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[37]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [3, 1, 7, 2, 9],
                         'min_samples_leaf': [25, 30, 27, 12, 54],
                         'n_estimators': [25, 34, 78, 15, 98]},
             scoring='accuracy')
```

In [38]:

```
grid_search.best_score_
```

Out[38]:

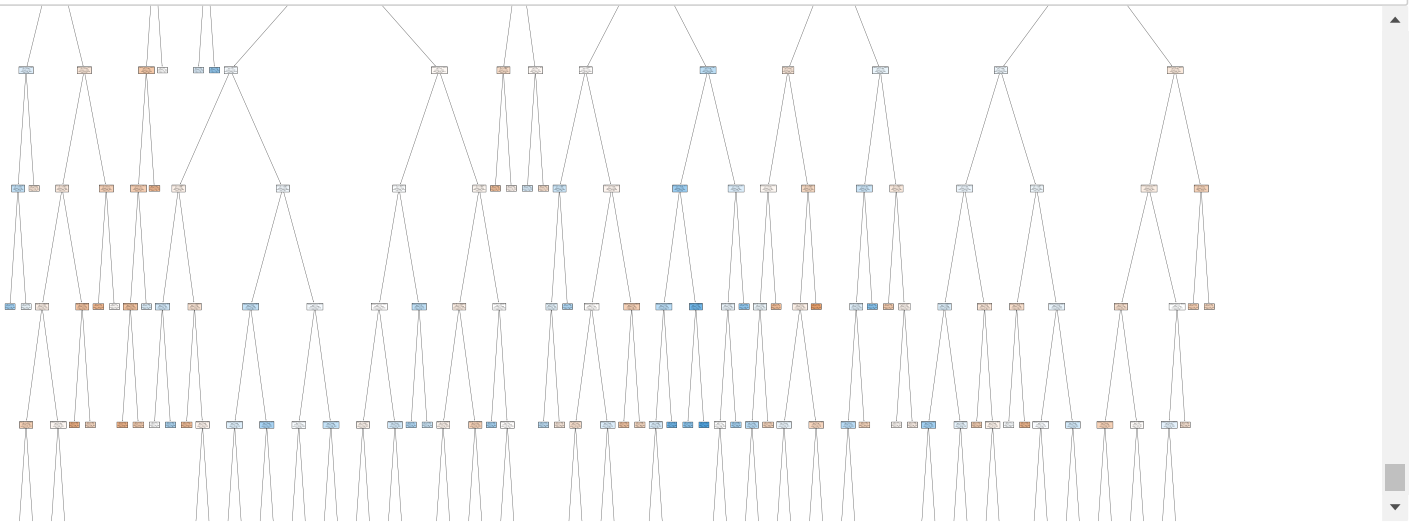
```
0.5080571428571429
```

In [39]:

```
rfc_best=grid_search.best_estimator_
```

In [40]:

```
plt.figure(figsize=(80,80))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```



DataSet C4

In [41]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C4_framingham.csv")
a
```

Out[41]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0
...	...	...	...	...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0

4238 rows × 16 columns

In [42]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [43]:

```
a.columns
```

Out[43]:

```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
```

In [44]:

```
a=a[['male', 'age',  'currentSmoker',
       'prevalentStroke', 'prevalentHyp', 'diabetes',  'sysBP',
       'diaBP',  'TenYearCHD']]
a
```

Out[44]:

	male	age	currentSmoker	prevalentStroke	prevalentHyp	diabetes	sysBP	diaBP	TenYearCHD
0	1	39	0	0	0	0	106.0	70.0	0
1	0	46	0	0	0	0	121.0	81.0	0
2	1	48	1	0	0	0	127.5	80.0	0
3	0	61	1	0	1	0	150.0	95.0	1
4	0	46	1	0	0	0	130.0	84.0	0
...	...	...	...	...	...	...	...	...	...
4233	1	50	1	0	1	0	179.0	92.0	1
4234	1	51	1	0	0	0	126.5	80.0	0
4235	0	48	1	0	0	0	131.0	72.0	0
4236	0	44	1	0	0	0	126.5	87.0	0
4237	0	52	0	0	0	0	133.5	83.0	0

4238 rows × 9 columns

In [45]:

```
a['TenYearCHD'].value_counts()
```

Out[45]:

```
0    3594
1     644
Name: TenYearCHD, dtype: int64
```

In [46]:

```
x=a.drop('TenYearCHD',axis=1)
y=a['TenYearCHD']
```

In [47]:

```
g1={"TenYearCHD":{'0':1,'1':2}}
a=a.replace(g1)
print(a)
```

```
      male  age  currentSmoker  prevalentStroke  prevalentHyp  diabetes  \
0        1   39              0                0             0         0
1        0   46              0                0             0         0
2        1   48              1                0             0         0
3        0   61              1                0             1         0
4        0   46              1                0             0         0
...    ...  ...            ...              ...           ...       ...
4233     1   50              1                0             1         0
4234     1   51              1                0             0         0
4235     0   48              1                0             0         0
4236     0   44              1                0             0         0
4237     0   52              0                0             0         0

      sysBP  diaBP  TenYearCHD
0     106.0   70.0           0
1     121.0   81.0           0
2     127.5   80.0           0
3     150.0   95.0           1
4     130.0   84.0           0
...     ...    ...         ...
4233  179.0   92.0           1
4234  126.5   80.0           0
4235  131.0   72.0           0
4236  126.5   87.0           0
4237  133.5   83.0           0

[4238 rows x 9 columns]
```

In [48]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [49]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[49]:

```
RandomForestClassifier()
```

In [50]:

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

In [51]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[51]:

```
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 [52]:

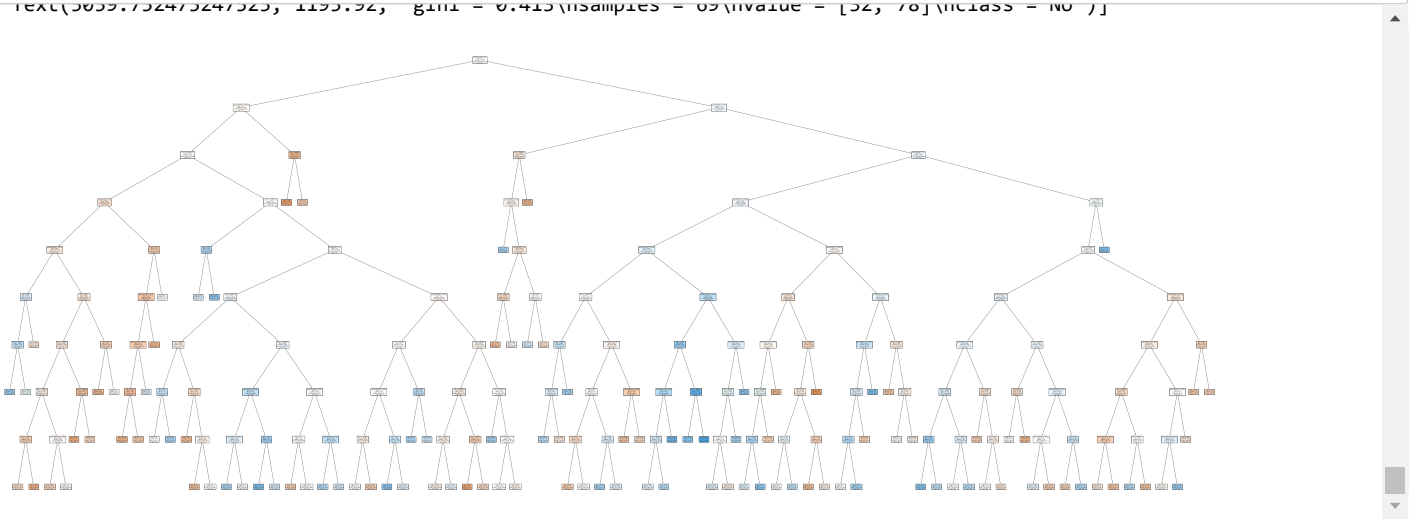
```
grid_search.best_score_
```

Out[52]:

```
0.8506405933917734
```

In [53]:

```
plt.figure(figsize=(100,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```



DataSet C5

In [54]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C5_health care diabetes.csv")
a
```

Out[54]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [55]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [56]:

```
a['Outcome'].value_counts()
```

Out[56]:

```
0    500
1    268
Name: Outcome, dtype: int64
```

In [57]:

```
x=a.drop('Outcome',axis=1)
y=a['Outcome']
```

In [58]:

```
g1={"Outcome":{'0':41,'1':62}}
a=a.replace(g1)
print(a)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
```

In [59]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [60]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[60]:

```
RandomForestClassifier()
```

In [61]:

```
parameters={'max_depth':[3,1,7,2,9],
            'min_samples_leaf':[25,30,27,12,54],
            'n_estimators':[25,34,78,15,98]
            }
```

In [62]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[62]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [3, 1, 7, 2, 9],
                         'min_samples_leaf': [25, 30, 27, 12, 54],
                         'n_estimators': [25, 34, 78, 15, 98]},
             scoring='accuracy')
```

In [63]:

```
grid_search.best_score_
```

Out[63]:

```
0.7597722354768907
```

In [64]:

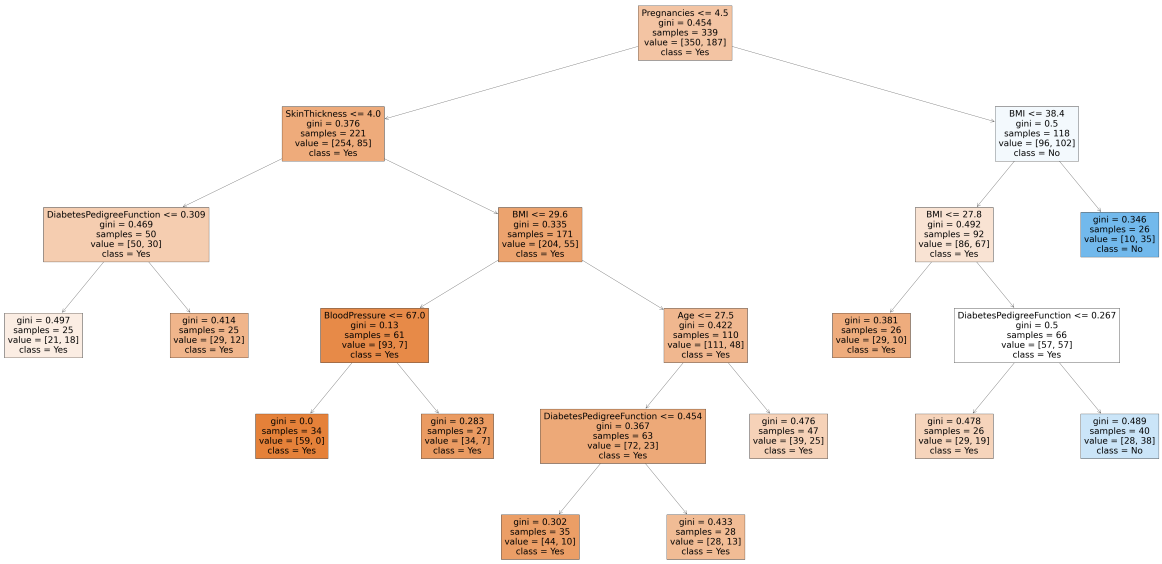
```
rfc_best=grid_search.best_estimator_
```

In [65]:

```
plt.figure(figsize=(100,50))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[65]:

```
[Text(3255.0, 2491.5, 'Pregnancies <= 4.5\ngini = 0.454\nsamples = 339\nvalue = [350, 187]\nclass = Ye
s'),
 Text(1674.0, 2038.5, 'SkinThickness <= 4.0\ngini = 0.376\nsamples = 221\nvalue = [254, 85]\nclass = Ye
s'),
 Text(744.0, 1585.5, 'DiabetesPedigreeFunction <= 0.309\ngini = 0.469\nsamples = 50\nvalue = [50, 30]\ncl
ass = Yes'),
 Text(372.0, 1132.5, 'gini = 0.497\nsamples = 25\nvalue = [21, 18]\nclass = Yes'),
 Text(1116.0, 1132.5, 'gini = 0.414\nsamples = 25\nvalue = [29, 12]\nclass = Yes'),
 Text(2604.0, 1585.5, 'BMI <= 29.6\ngini = 0.335\nsamples = 171\nvalue = [204, 55]\nclass = Yes'),
 Text(1860.0, 1132.5, 'BloodPressure <= 67.0\ngini = 0.13\nsamples = 61\nvalue = [93, 7]\nclass = Yes'),
 Text(1488.0, 679.5, 'gini = 0.0\nsamples = 34\nvalue = [59, 0]\nclass = Yes'),
 Text(2232.0, 679.5, 'gini = 0.283\nsamples = 27\nvalue = [34, 7]\nclass = Yes'),
 Text(3348.0, 1132.5, 'Age <= 27.5\ngini = 0.422\nsamples = 110\nvalue = [111, 48]\nclass = Yes'),
 Text(2976.0, 679.5, 'DiabetesPedigreeFunction <= 0.454\ngini = 0.367\nsamples = 63\nvalue = [72, 23]\ncl
ass = Yes'),
 Text(2604.0, 226.5, 'gini = 0.302\nsamples = 35\nvalue = [44, 10]\nclass = Yes'),
 Text(3348.0, 226.5, 'gini = 0.433\nsamples = 28\nvalue = [28, 13]\nclass = Yes'),
 Text(3720.0, 679.5, 'gini = 0.476\nsamples = 47\nvalue = [39, 25]\nclass = Yes'),
 Text(4836.0, 2038.5, 'BMI <= 38.4\ngini = 0.5\nsamples = 118\nvalue = [96, 102]\nclass = No'),
 Text(4464.0, 1585.5, 'BMI <= 27.8\ngini = 0.492\nsamples = 92\nvalue = [86, 67]\nclass = Yes'),
 Text(4092.0, 1132.5, 'gini = 0.381\nsamples = 26\nvalue = [29, 10]\nclass = Yes'),
 Text(4836.0, 1132.5, 'DiabetesPedigreeFunction <= 0.267\ngini = 0.5\nsamples = 66\nvalue = [57, 57]\ncla
ss = Yes'),
 Text(4464.0, 679.5, 'gini = 0.478\nsamples = 26\nvalue = [29, 19]\nclass = Yes'),
 Text(5208.0, 679.5, 'gini = 0.489\nsamples = 40\nvalue = [28, 38]\nclass = No'),
 Text(5208.0, 1585.5, 'gini = 0.346\nsamples = 26\nvalue = [10, 35]\nclass = No')]
```



DataSet C6

In [66]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C6_bmi.csv")
a
```

Out[66]:

	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 [67]:

```
a.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 [68]:

```
a['Gender'].value_counts()
```

Out[68]:

```
Female    255
Male      245
Name: Gender, dtype: int64
```

In [69]:

```
x=a.drop('Gender',axis=1)
y=a['Gender']
```

In [70]:

```
g1={"Gender":{'Female':51,'Male':72}}
a=a.replace(g1)
print(a)
```

```
     Gender  Height  Weight  Index
0        72     174      96      4
1        72     189      87      2
2        51     185     110      4
3        51     195     104      3
4        72     149      61      3
..      ...     ...     ...    ...
495      51     150     153      5
496      51     184     121      4
497      51     141     136      5
498      72     150      95      5
499      72     173     131      5

[500 rows x 4 columns]
```

In [71]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [72]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[72]:

```
RandomForestClassifier()
```

In [73]:

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

In [74]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[74]:

```
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 [75]:

```
grid_search.best_score_
```

Out[75]:

```
0.5457142857142857
```

In [76]:

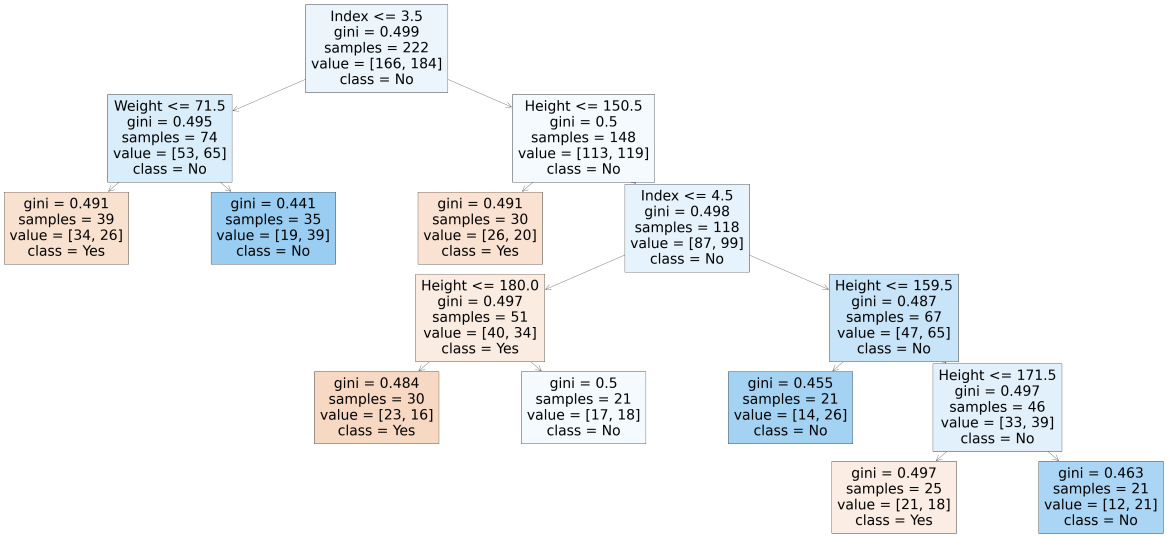
```
rfc_best=grid_search.best_estimator_
```

In [77]:

```
plt.figure(figsize=(90,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[77]:

```
[Text(1674.0, 1993.2, 'Index <= 3.5\ngini = 0.499\nsamples = 222\nvalue = [166, 184]\nclass = No'),
 Text(837.0, 1630.8000000000002, 'Weight <= 71.5\ngini = 0.495\nsamples = 74\nvalue = [53, 65]\nclass = N
o'),
 Text(418.5, 1268.4, 'gini = 0.491\nsamples = 39\nvalue = [34, 26]\nclass = Yes'),
 Text(1255.5, 1268.4, 'gini = 0.441\nsamples = 35\nvalue = [19, 39]\nclass = No'),
 Text(2511.0, 1630.8000000000002, 'Height <= 150.5\ngini = 0.5\nsamples = 148\nvalue = [113, 119]\nclass
= No'),
 Text(2092.5, 1268.4, 'gini = 0.491\nsamples = 30\nvalue = [26, 20]\nclass = Yes'),
 Text(2929.5, 1268.4, 'Index <= 4.5\ngini = 0.498\nsamples = 118\nvalue = [87, 99]\nclass = No'),
 Text(2092.5, 906.0, 'Height <= 180.0\ngini = 0.497\nsamples = 51\nvalue = [40, 34]\nclass = Yes'),
 Text(1674.0, 543.5999999999999, 'gini = 0.484\nsamples = 30\nvalue = [23, 16]\nclass = Yes'),
 Text(2511.0, 543.5999999999999, 'gini = 0.5\nsamples = 21\nvalue = [17, 18]\nclass = No'),
 Text(3766.5, 906.0, 'Height <= 159.5\ngini = 0.487\nsamples = 67\nvalue = [47, 65]\nclass = No'),
 Text(3348.0, 543.5999999999999, 'gini = 0.455\nsamples = 21\nvalue = [14, 26]\nclass = No'),
 Text(4185.0, 543.5999999999999, 'Height <= 171.5\ngini = 0.497\nsamples = 46\nvalue = [33, 39]\nclass =
No'),
 Text(3766.5, 181.19999999999982, 'gini = 0.497\nsamples = 25\nvalue = [21, 18]\nclass = Yes'),
 Text(4603.5, 181.19999999999982, 'gini = 0.463\nsamples = 21\nvalue = [12, 21]\nclass = No')]
```



DataSet C7

In [78]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\c7_used_cars.csv")
a
```

Out[78]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Make
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2.0	VW
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2.0	VW
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2.0	VW
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2.0	VW
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1.5	VW
...	...	...	...	...	...	...	...	...	...	...	...
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1.0	Audi
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1.0	Audi
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1.0	Audi
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1.4	Audi
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1.4	Audi

99187 rows × 11 columns

In [79]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99187 entries, 0 to 99186
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    99187 non-null  int64  
 1   model         99187 non-null  object 
 2   year          99187 non-null  int64  
 3   price         99187 non-null  int64  
 4   transmission  99187 non-null  object 
 5   mileage       99187 non-null  int64  
 6   fuelType      99187 non-null  object 
 7   tax           99187 non-null  int64  
 8   mpg           99187 non-null  float64
 9   engineSize    99187 non-null  float64
 10  Make          99187 non-null  object 
dtypes: float64(2), int64(5), object(4)
memory usage: 8.3+ MB
```

In [80]:

```
a.columns
```

Out[80]:

```
Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
       'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
      dtype='object')
```

In [81]:

```
a=a[['Unnamed: 0','year','price','mileage',
       'fuelType', 'tax', 'mpg', 'engineSize']]
a=a.head(10000)
a
```

Out[81]:

	Unnamed: 0	year	price	mileage	fuelType	tax	mpg	engineSize
0	0	2019	25000	13904	Diesel	145	49.6	2.0
1	1	2019	26883	4562	Diesel	145	49.6	2.0
2	2	2019	20000	7414	Diesel	145	50.4	2.0
3	3	2019	33492	4825	Petrol	145	32.5	2.0
4	4	2019	22900	6500	Petrol	150	39.8	1.5
...	...	...	...	...	...	...	...	...
9995	9995	2015	8700	59650	Petrol	20	60.1	1.2
9996	9996	2009	2495	41637	Petrol	145	48.7	1.2
9997	9997	2017	11000	14809	Petrol	150	60.1	1.2
9998	9998	2016	12000	15441	Petrol	20	61.4	1.4
9999	9999	2017	9100	38291	Petrol	150	60.1	1.2

10000 rows × 8 columns

In [82]:

```
a['fuelType'].value_counts()
```

Out[82]:

```
Petrol    6580
Diesel    3224
Hybrid     145
Other       51
Name: fuelType, dtype: int64
```

In [83]:

```
x=a.drop('fuelType',axis=1)
y=a['fuelType']
```

In [84]:

```
g1={"fuelType":{'Petrol':21,'Diesel':72,'Hybrid':45,'Other':67,'Electric':86}}
a=a.replace(g1)
print(a)
```

```
      Unnamed: 0  year  price  mileage  fuelType  tax   mpg  engineSize
0              0  2019  25000    13904        72  145  49.6         2.0
1              1  2019  26883     4562        72  145  49.6         2.0
2              2  2019  20000     7414        72  145  50.4         2.0
3              3  2019  33492     4825        21  145  32.5         2.0
4              4  2019  22900     6500        21  150  39.8         1.5
...          ...   ...    ...      ...       ...  ...   ...         ...
9995        9995  2015   8700    59650        21   20  60.1         1.2
9996        9996  2009   2495    41637        21  145  48.7         1.2
9997        9997  2017  11000    14809        21  150  60.1         1.2
9998        9998  2016  12000    15441        21   20  61.4         1.4
9999        9999  2017   9100    38291        21  150  60.1         1.2

[10000 rows x 8 columns]
```

In [85]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [86]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[86]:

```
RandomForestClassifier()
```

In [87]:

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

In [88]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[88]:

```
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 [89]:

```
grid_search.best_score_
```

Out[89]:

```
0.9854285714285714
```

In [93]:

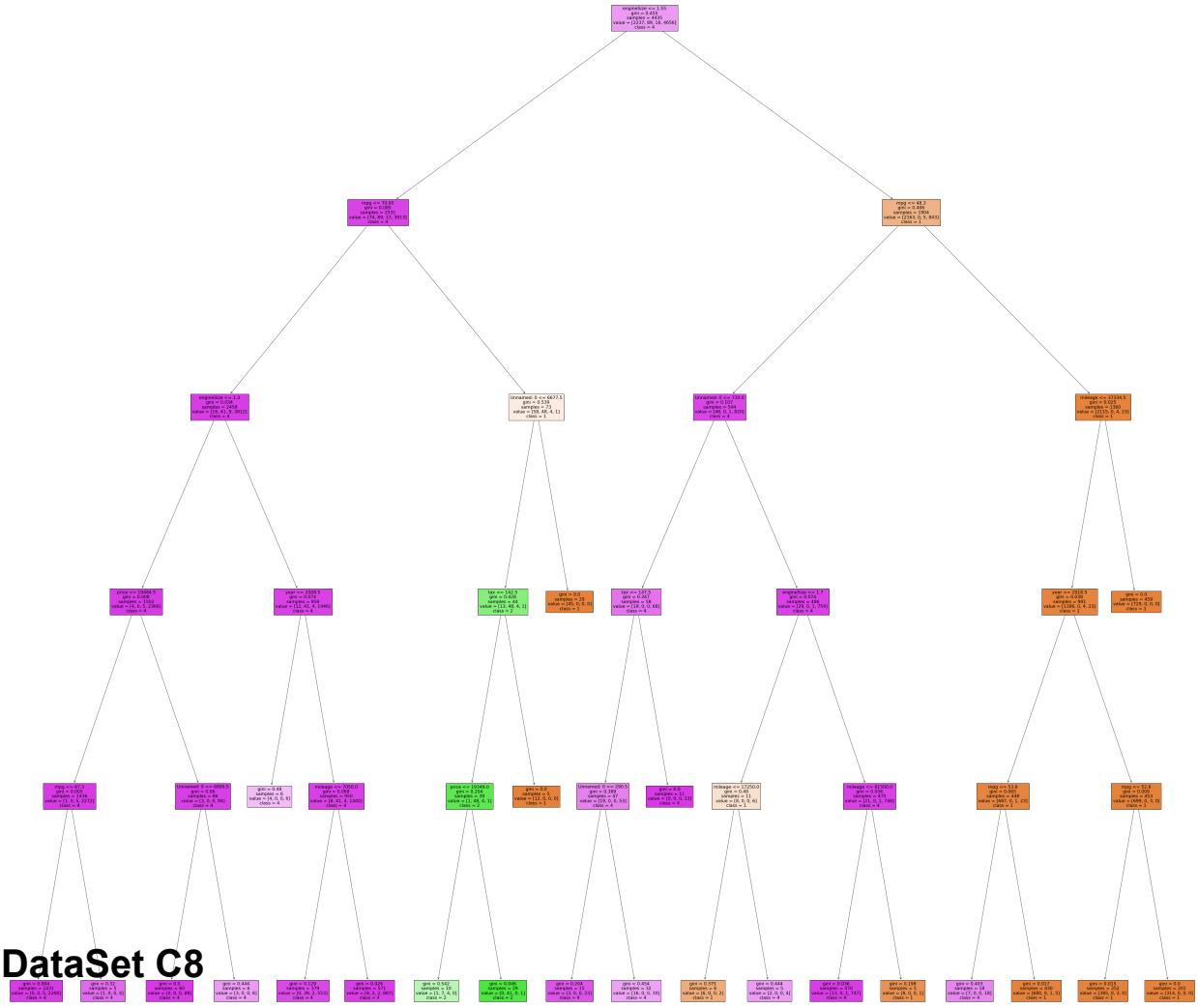
```
rfc_best=grid_search.best_estimator_
```

In [96]:

```
plt.figure(figsize=(80,80))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['1','2','3','4','5'],filled=True)
```

Out[96]:

```
[Text(2387.0, 3986.4, 'engineSize <= 1.55\ngini = 0.455\nsamples = 4435\nvalue = [2237, 89, 18, 4656]\ncl
ass = 4'),
 Text(1395.0, 3261.6000000000004, 'mpg <= 70.65\ngini = 0.085\nsamples = 2531\nvalue = [74, 89, 13, 3813]
\nclass = 4'),
 Text(806.0, 2536.8, 'engineSize <= 1.3\ngini = 0.034\nsamples = 2458\nvalue = [16, 41, 9, 3812]\nclass =
4'),
 Text(496.0, 1812.0, 'price <= 19484.5\ngini = 0.008\nsamples = 1502\nvalue = [4, 0, 5, 2366]\nclass =
4'),
 Text(248.0, 1087.1999999999998, 'mpg <= 67.3\ngini = 0.005\nsamples = 1436\nvalue = [1, 0, 5, 2272]\ncla
ss = 4'),
 Text(124.0, 362.39999999999964, 'gini = 0.004\nsamples = 1431\nvalue = [0, 0, 5, 2268]\nclass = 4'),
 Text(372.0, 362.39999999999964, 'gini = 0.32\nsamples = 5\nvalue = [1, 0, 0, 4]\nclass = 4'),
 Text(744.0, 1087.1999999999998, 'Unnamed: 0 <= 6800.5\ngini = 0.06\nsamples = 66\nvalue = [3, 0, 0, 94]
\nclass = 4'),
 Text(620.0, 362.39999999999964, 'gini = 0.0\nsamples = 60\nvalue = [0, 0, 0, 88]\nclass = 4'),
 Text(868.0, 362.39999999999964, 'gini = 0.444\nsamples = 6\nvalue = [3, 0, 0, 6]\nclass = 4'),
 Text(1116.0, 1812.0, 'year <= 2008.5\ngini = 0.074\nsamples = 956\nvalue = [12, 41, 4, 1446]\nclass =
4'),
 Text(992.0, 1087.1999999999998, 'gini = 0.48\nsamples = 6\nvalue = [4, 0, 0, 6]\nclass = 4'),
 Text(1240.0, 1087.1999999999998, 'mileage <= 7050.0\ngini = 0.069\nsamples = 950\nvalue = [8, 41, 4, 144
0]\nclass = 4'),
 Text(1116.0, 362.39999999999964, 'gini = 0.129\nsamples = 379\nvalue = [0, 39, 2, 553]\nclass = 4'),
 Text(1364.0, 362.39999999999964, 'gini = 0.026\nsamples = 571\nvalue = [8, 2, 2, 887]\nclass = 4'),
 Text(1984.0, 2536.8, 'Unnamed: 0 <= 6677.5\ngini = 0.539\nsamples = 73\nvalue = [58, 48, 4, 1]\nclass =
1'),
 Text(1860.0, 1812.0, 'tax <= 142.5\ngini = 0.428\nsamples = 44\nvalue = [13, 48, 4, 1]\nclass = 2'),
 Text(1736.0, 1087.1999999999998, 'price <= 19349.0\ngini = 0.204\nsamples = 39\nvalue = [1, 48, 4, 1]\nc
lass = 2'),
 Text(1612.0, 362.39999999999964, 'gini = 0.542\nsamples = 10\nvalue = [1, 7, 4, 0]\nclass = 2'),
 Text(1860.0, 362.39999999999964, 'gini = 0.046\nsamples = 29\nvalue = [0, 41, 0, 1]\nclass = 2'),
 Text(1984.0, 1087.1999999999998, 'gini = 0.0\nsamples = 5\nvalue = [12, 0, 0, 0]\nclass = 1'),
 Text(2108.0, 1812.0, 'gini = 0.0\nsamples = 29\nvalue = [45, 0, 0, 0]\nclass = 1'),
 Text(3379.0, 3261.6000000000004, 'mpg <= 48.3\ngini = 0.406\nsamples = 1904\nvalue = [2163, 0, 5, 843]\n
class = 1'),
 Text(2666.0, 2536.8, 'Unnamed: 0 <= 730.0\ngini = 0.107\nsamples = 544\nvalue = [48, 0, 1, 820]\nclass =
4'),
 Text(2356.0, 1812.0, 'tax <= 147.5\ngini = 0.347\nsamples = 58\nvalue = [19, 0, 0, 66]\nclass = 4'),
 Text(2232.0, 1087.1999999999998, 'Unnamed: 0 <= 290.5\ngini = 0.389\nsamples = 47\nvalue = [19, 0, 0, 5
3]\nclass = 4'),
 Text(2108.0, 362.39999999999964, 'gini = 0.204\nsamples = 15\nvalue = [3, 0, 0, 23]\nclass = 4'),
 Text(2356.0, 362.39999999999964, 'gini = 0.454\nsamples = 32\nvalue = [16, 0, 0, 30]\nclass = 4'),
 Text(2480.0, 1087.1999999999998, 'gini = 0.0\nsamples = 11\nvalue = [0, 0, 0, 13]\nclass = 4'),
 Text(2976.0, 1812.0, 'engineSize <= 1.7\ngini = 0.074\nsamples = 486\nvalue = [29, 0, 1, 754]\nclass =
4'),
 Text(2728.0, 1087.1999999999998, 'mileage <= 17250.0\ngini = 0.49\nsamples = 11\nvalue = [8, 0, 0, 6]\nc
lass = 1'),
 Text(2604.0, 362.39999999999964, 'gini = 0.375\nsamples = 6\nvalue = [6, 0, 0, 2]\nclass = 1'),
 Text(2852.0, 362.39999999999964, 'gini = 0.444\nsamples = 5\nvalue = [2, 0, 0, 4]\nclass = 4'),
 Text(3224.0, 1087.1999999999998, 'mileage <= 81500.0\ngini = 0.056\nsamples = 475\nvalue = [21, 0, 1, 74
8]\nclass = 4'),
 Text(3100.0, 362.39999999999964, 'gini = 0.036\nsamples = 470\nvalue = [13, 0, 1, 747]\nclass = 4'),
 Text(3348.0, 362.39999999999964, 'gini = 0.198\nsamples = 5\nvalue = [8, 0, 0, 1]\nclass = 1'),
 Text(4092.0, 2536.8, 'mileage <= 37334.5\ngini = 0.025\nsamples = 1360\nvalue = [2115, 0, 4, 23]\nclass
= 1'),
 Text(3968.0, 1812.0, 'year <= 2018.5\ngini = 0.038\nsamples = 901\nvalue = [1386, 0, 4, 23]\nclass =
1'),
 Text(3720.0, 1087.1999999999998, 'mpg <= 53.8\ngini = 0.065\nsamples = 448\nvalue = [687, 0, 1, 23]\ncla
ss = 1'),
 Text(3596.0, 362.39999999999964, 'gini = 0.403\nsamples = 18\nvalue = [7, 0, 0, 18]\nclass = 4'),
 Text(3844.0, 362.39999999999964, 'gini = 0.017\nsamples = 430\nvalue = [680, 0, 1, 5]\nclass = 1'),
 Text(4216.0, 1087.1999999999998, 'mpg <= 52.8\ngini = 0.009\nsamples = 453\nvalue = [699, 0, 3, 0]\nclas
s = 1'),
 Text(4092.0, 362.39999999999964, 'gini = 0.015\nsamples = 252\nvalue = [385, 0, 3, 0]\nclass = 1'),
 Text(4340.0, 362.39999999999964, 'gini = 0.0\nsamples = 201\nvalue = [314, 0, 0, 0]\nclass = 1'),
 Text(4216.0, 1812.0, 'gini = 0.0\nsamples = 459\nvalue = [729, 0, 0, 0]\nclass = 1')]
```



DataSet C8

In [97]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C8_loan-train.csv")
a
```

Out[97]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	
...	...	...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	

614 rows × 13 columns

In [98]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [99]:

```
a=a.fillna(value=76)
a
```

Out[99]:

ried	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
No	0	Graduate	No	5849	0.0	76.0	360.0	1.0
Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0
Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0
Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0
No	0	Graduate	No	6000	0.0	141.0	360.0	1.0
...	...	...	...	...	...	...	...	...
No	0	Graduate	No	2900	0.0	71.0	360.0	1.0
Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0
Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0
Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0
No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0

In [100]:

```
a.columns
```

Out[100]:

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [101]:

```
b=a[['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History','Married']]
b
```

Out[101]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Married
0	5849	0.0	76.0	360.0	1.0	No
1	4583	1508.0	128.0	360.0	1.0	Yes
2	3000	0.0	66.0	360.0	1.0	Yes
3	2583	2358.0	120.0	360.0	1.0	Yes
4	6000	0.0	141.0	360.0	1.0	No
...	...	...	...	...	...	...
609	2900	0.0	71.0	360.0	1.0	No
610	4106	0.0	40.0	180.0	1.0	Yes
611	8072	240.0	253.0	360.0	1.0	Yes
612	7583	0.0	187.0	360.0	1.0	Yes
613	4583	0.0	133.0	360.0	0.0	No

614 rows × 6 columns

In [104]:

```
b['Married'].value_counts()
```

Out[104]:

```
Yes    398
No     213
76       3
Name: Married, dtype: int64
```

In [105]:

```
x=b.drop('Married',axis=1)
y=b['Married']
```

In [106]:

```
g1={"Married":{'Yes':71,'No':28}}
b=b.replace(g1)
print(b)
```

```
     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5849                0.0        76.0             360.0
1               4583             1508.0       128.0             360.0
2               3000                0.0        66.0             360.0
3               2583             2358.0       120.0             360.0
4               6000                0.0       141.0             360.0
..               ...                ...         ...               ...
609             2900                0.0        71.0             360.0
610             4106                0.0        40.0             180.0
611             8072              240.0       253.0             360.0
612             7583                0.0       187.0             360.0
613             4583                0.0       133.0             360.0

     Credit_History  Married
0               1.0       28
1               1.0       71
2               1.0       71
3               1.0       71
4               1.0       28
..              ...      ...
609             1.0       28
610             1.0       71
611             1.0       71
612             1.0       71
613             0.0       28

[614 rows x 6 columns]
```

In [107]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [108]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[108]:

```
RandomForestClassifier()
```

In [109]:

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

In [110]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[110]:

```
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 [111]:

```
grid_search.best_score_
```

Out[111]:

```
0.7086502934144752
```

In [112]:

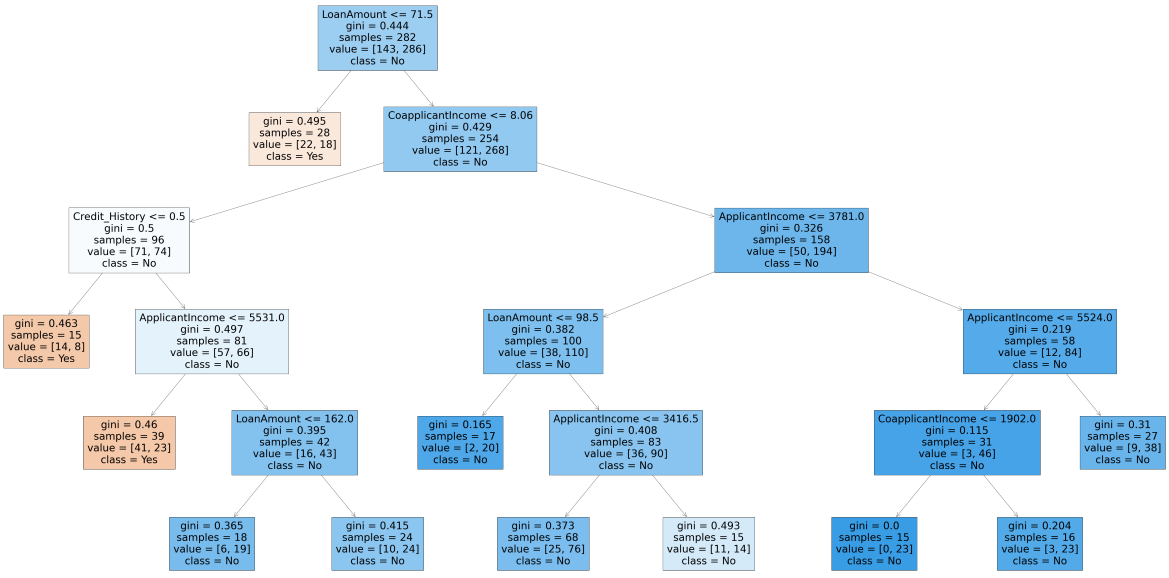
```
rfc_best=grid_search.best_estimator_
```

In [113]:

```
plt.figure(figsize=(100,50))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[113]:

```
[Text(1860.0, 2491.5, 'LoanAmount <= 71.5\ngini = 0.444\nsamples = 282\nvalue = [143, 286]\nclass = No'),
 Text(1488.0, 2038.5, 'gini = 0.495\nsamples = 28\nvalue = [22, 18]\nclass = Yes'),
 Text(2232.0, 2038.5, 'CoapplicantIncome <= 8.06\ngini = 0.429\nsamples = 254\nvalue = [121, 268]\nclass
= No'),
 Text(744.0, 1585.5, 'Credit_History <= 0.5\ngini = 0.5\nsamples = 96\nvalue = [71, 74]\nclass = No'),
 Text(372.0, 1132.5, 'gini = 0.463\nsamples = 15\nvalue = [14, 8]\nclass = Yes'),
 Text(1116.0, 1132.5, 'ApplicantIncome <= 5531.0\ngini = 0.497\nsamples = 81\nvalue = [57, 66]\nclass = N
o'),
 Text(744.0, 679.5, 'gini = 0.46\nsamples = 39\nvalue = [41, 23]\nclass = Yes'),
 Text(1488.0, 679.5, 'LoanAmount <= 162.0\ngini = 0.395\nsamples = 42\nvalue = [16, 43]\nclass = No'),
 Text(1116.0, 226.5, 'gini = 0.365\nsamples = 18\nvalue = [6, 19]\nclass = No'),
 Text(1860.0, 226.5, 'gini = 0.415\nsamples = 24\nvalue = [10, 24]\nclass = No'),
 Text(3720.0, 1585.5, 'ApplicantIncome <= 3781.0\ngini = 0.326\nsamples = 158\nvalue = [50, 194]\nclass =
No'),
 Text(2604.0, 1132.5, 'LoanAmount <= 98.5\ngini = 0.382\nsamples = 100\nvalue = [38, 110]\nclass = No'),
 Text(2232.0, 679.5, 'gini = 0.165\nsamples = 17\nvalue = [2, 20]\nclass = No'),
 Text(2976.0, 679.5, 'ApplicantIncome <= 3416.5\ngini = 0.408\nsamples = 83\nvalue = [36, 90]\nclass = N
o'),
 Text(2604.0, 226.5, 'gini = 0.373\nsamples = 68\nvalue = [25, 76]\nclass = No'),
 Text(3348.0, 226.5, 'gini = 0.493\nsamples = 15\nvalue = [11, 14]\nclass = No'),
 Text(4836.0, 1132.5, 'ApplicantIncome <= 5524.0\ngini = 0.219\nsamples = 58\nvalue = [12, 84]\nclass = N
o'),
 Text(4464.0, 679.5, 'CoapplicantIncome <= 1902.0\ngini = 0.115\nsamples = 31\nvalue = [3, 46]\nclass = N
o'),
 Text(4092.0, 226.5, 'gini = 0.0\nsamples = 15\nvalue = [0, 23]\nclass = No'),
 Text(4836.0, 226.5, 'gini = 0.204\nsamples = 16\nvalue = [3, 23]\nclass = No'),
 Text(5208.0, 679.5, 'gini = 0.31\nsamples = 27\nvalue = [9, 38]\nclass = No')]
```



DataSet C9

In [114]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C9_Data.csv")
a
```

Out[114]:

	row_id	user_id	timestamp	gate_id
0	0	18	2022-07-29 09:08:54	7
1	1	18	2022-07-29 09:09:54	9
2	2	18	2022-07-29 09:09:54	9
3	3	18	2022-07-29 09:10:06	5
4	4	18	2022-07-29 09:10:08	5
...	...	...	...	...
37513	37513	6	2022-12-31 20:38:56	11
37514	37514	6	2022-12-31 20:39:22	6
37515	37515	6	2022-12-31 20:39:23	6
37516	37516	6	2022-12-31 20:39:31	9
37517	37517	6	2022-12-31 20:39:31	9

37518 rows × 4 columns

In [115]:

```
a=a.head(1000)
a
```

Out[115]:

	row_id	user_id	timestamp	gate_id
0	0	18	2022-07-29 09:08:54	7
1	1	18	2022-07-29 09:09:54	9
2	2	18	2022-07-29 09:09:54	9
3	3	18	2022-07-29 09:10:06	5
4	4	18	2022-07-29 09:10:08	5
...	...	...	...	...
995	995	3	2022-08-01 18:33:29	11
996	996	3	2022-08-01 18:33:48	4
997	997	3	2022-08-01 18:33:49	4
998	998	55	2022-08-01 18:35:45	7
999	999	55	2022-08-01 18:36:41	3

1000 rows × 4 columns

In [116]:

```
a.columns
```

Out[116]:

```
Index(['row_id', 'user_id', 'timestamp', 'gate_id'], dtype='object')
```

In [117]:

```
b=a[['row_id', 'user_id',  'gate_id']]
b
```

Out[117]:

	row_id	user_id	gate_id
0	0	18	7
1	1	18	9
2	2	18	9
3	3	18	5
4	4	18	5
...	...	...	...
995	995	3	11
996	996	3	4
997	997	3	4
998	998	55	7
999	999	55	3

1000 rows × 3 columns

In [118]:

```
b['gate_id'].value_counts()
```

Out[118]:

```
 4     207
 3     157
 5     136
 10    112
 11    102
 7      88
 9      70
 6      49
 13     34
 12     32
 15      7
-1       6
Name: gate_id, dtype: int64
```

In [119]:

```
x=b.drop('gate_id',axis=1)
y=b['gate_id']
```

In [140]:

```
g1={"gate_id":{'10':71,'11':28,'4':32,'3':21,'5':23,'7':34,'9':56,'6':43,'13':25,'12':21,'15':6,'-1':3}}
b=b.replace(g1)
print(b)
```

```
     row_id  user_id  gate_id
0         0       18        7
1         1       18        9
2         2       18        9
3         3       18        5
4         4       18        5
..      ...      ...      ...
995     995        3       11
996     996        3        4
997     997        3        4
998     998       55        7
999     999       55        3

[1000 rows x 3 columns]
```

In [141]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [142]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[142]:

```
RandomForestClassifier()
```

In [143]:

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

In [144]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[144]:

```
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 [145]:

```
grid_search.best_score_
```

Out[145]:

```
0.24714285714285716
```

In [146]:

```
rfc_best=grid_search.best_estimator_
```

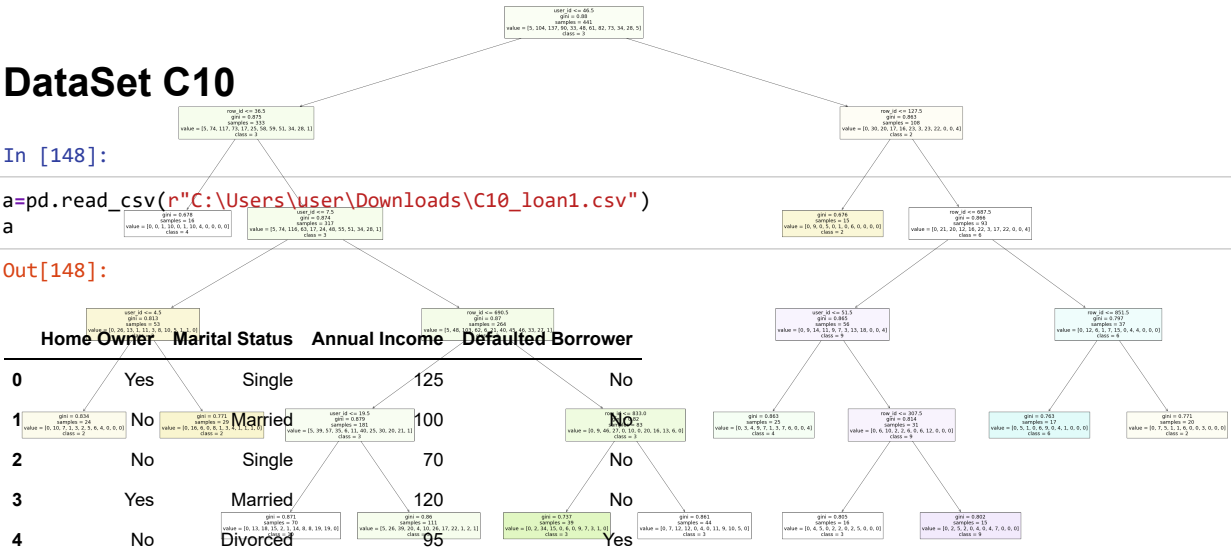
In [147]:

```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['1','2','3','4','5','6','7','8','9','10','11','1
```

Out[147]:

```
[Text(2046.0, 1993.2, 'user_id <= 46.5\ngini = 0.88\nsamples = 441\nvalue = [5, 104, 137, 90, 33, 48, 61,
82, 73, 34, 28, 5]\nclass = 3'),
 Text(868.0, 1630.8000000000002, 'row_id <= 36.5\ngini = 0.875\nsamples = 333\nvalue = [5, 74, 117, 73, 1
7, 25, 58, 59, 51, 34, 28, 1]\nclass = 3'),
 Text(620.0, 1268.4, 'gini = 0.678\nsamples = 16\nvalue = [0, 0, 1, 10, 0, 1, 10, 4, 0, 0, 0, 0]\nclass =
4'),
 Text(1116.0, 1268.4, 'user_id <= 7.5\ngini = 0.874\nsamples = 317\nvalue = [5, 74, 116, 63, 17, 24, 48,
55, 51, 34, 28, 1]\nclass = 3'),
 Text(496.0, 906.0, 'user_id <= 4.5\ngini = 0.813\nsamples = 53\nvalue = [0, 26, 13, 1, 11, 3, 8, 10, 5,
1, 1, 0]\nclass = 2'),
 Text(248.0, 543.5999999999999, 'gini = 0.834\nsamples = 24\nvalue = [0, 10, 7, 1, 3, 2, 5, 6, 4, 0, 0,
0]\nclass = 2'),
 Text(744.0, 543.5999999999999, 'gini = 0.771\nsamples = 29\nvalue = [0, 16, 6, 0, 8, 1, 3, 4, 1, 1, 1,
0]\nclass = 2'),
 Text(1736.0, 906.0, 'row_id <= 690.5\ngini = 0.87\nsamples = 264\nvalue = [5, 48, 103, 62, 6, 21, 40, 4
5, 46, 33, 27, 1]\nclass = 3'),
 Text(1240.0, 543.5999999999999, 'user_id <= 19.5\ngini = 0.879\nsamples = 181\nvalue = [5, 39, 57, 35,
6, 11, 40, 25, 30, 20, 21, 1]\nclass = 3'),
 Text(992.0, 181.19999999999982, 'gini = 0.871\nsamples = 70\nvalue = [0, 13, 18, 15, 2, 1, 14, 8, 8, 19,
19, 0]\nclass = 10'),
 Text(1488.0, 181.19999999999982, 'gini = 0.86\nsamples = 111\nvalue = [5, 26, 39, 20, 4, 10, 26, 17, 22,
1, 2, 1]\nclass = 3'),
 Text(2232.0, 543.5999999999999, 'row_id <= 833.0\ngini = 0.82\nsamples = 83\nvalue = [0, 9, 46, 27, 0, 1
0, 0, 20, 16, 13, 6, 0]\nclass = 3'),
 Text(1984.0, 181.19999999999982, 'gini = 0.737\nsamples = 39\nvalue = [0, 2, 34, 15, 0, 6, 0, 9, 7, 3,
1, 0]\nclass = 3'),
 Text(2480.0, 181.19999999999982, 'gini = 0.861\nsamples = 44\nvalue = [0, 7, 12, 12, 0, 4, 0, 11, 9, 10,
5, 0]\nclass = 3'),
 Text(3224.0, 1630.8000000000002, 'row_id <= 127.5\ngini = 0.863\nsamples = 108\nvalue = [0, 30, 20, 17,
16, 23, 3, 23, 22, 0, 0, 4]\nclass = 2'),
 Text(2976.0, 1268.4, 'gini = 0.676\nsamples = 15\nvalue = [0, 9, 0, 5, 0, 1, 0, 6, 0, 0, 0, 0]\nclass =
2'),
 Text(3472.0, 1268.4, 'row_id <= 687.5\ngini = 0.866\nsamples = 93\nvalue = [0, 21, 20, 12, 16, 22, 3, 1
7, 22, 0, 0, 4]\nclass = 6'),
 Text(2976.0, 906.0, 'user_id <= 51.5\ngini = 0.865\nsamples = 56\nvalue = [0, 9, 14, 11, 9, 7, 3, 13, 1
8, 0, 0, 4]\nclass = 9'),
 Text(2728.0, 543.5999999999999, 'gini = 0.863\nsamples = 25\nvalue = [0, 3, 4, 9, 7, 1, 3, 7, 6, 0, 0,
4]\nclass = 4'),
 Text(3224.0, 543.5999999999999, 'row_id <= 307.5\ngini = 0.814\nsamples = 31\nvalue = [0, 6, 10, 2, 2,
6, 0, 6, 12, 0, 0, 0]\nclass = 9'),
 Text(2976.0, 181.19999999999982, 'gini = 0.805\nsamples = 16\nvalue = [0, 4, 5, 0, 2, 2, 0, 2, 5, 0, 0,
0]\nclass = 3'),
 Text(3472.0, 181.19999999999982, 'gini = 0.802\nsamples = 15\nvalue = [0, 2, 5, 2, 0, 4, 0, 4, 7, 0, 0,
0]\nclass = 9'),
 Text(3968.0, 906.0, 'row_id <= 851.5\ngini = 0.797\nsamples = 37\nvalue = [0, 12, 6, 1, 7, 15, 0, 4, 4,
0, 0, 0]\nclass = 6'),
 Text(3720.0, 543.5999999999999, 'gini = 0.763\nsamples = 17\nvalue = [0, 5, 1, 0, 6, 9, 0, 4, 1, 0, 0,
0]\nclass = 6'),
 Text(4216.0, 543.5999999999999, 'gini = 0.771\nsamples = 20\nvalue = [0, 7, 5, 1, 1, 6, 0, 0, 3, 0, 0,
0]\nclass = 2')]
```

DataSet C10



In [148]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_loan1.csv")
a
```

Out[148]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

In [149]:

```
b=a[['Annual Income','Defaulted Borrower']]
b
```

Out[149]:

	Annual Income	Defaulted Borrower
0	125	No
1	100	No
2	70	No
3	120	No
4	95	Yes
5	60	No
6	220	No
7	85	Yes
8	75	No
9	90	Yes

In [150]:

```
b['Defaulted Borrower'].value_counts()
```

Out[150]:

```
No     7
Yes    3
Name: Defaulted Borrower, dtype: int64
```

In [151]:

```
x=b.drop('Defaulted Borrower',axis=1)
y=b['Defaulted Borrower']
```

In [152]:

```
g1={"Defaulted Borrower":{'Yes':71,'No':28}}
b=b.replace(g1)
print(b)
```

```
   Annual Income  Defaulted Borrower
0            125                  28
1            100                  28
2             70                  28
3            120                  28
4             95                  71
5             60                  28
6            220                  28
7             85                  71
8             75                  28
9             90                  71
```

In [153]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [154]:

```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[154]:

```
RandomForestClassifier()
```

In [155]:

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

In [156]:

```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:666: UserWarning: The least
populated class in y has only 1 members, which is less than n_splits=2.
  warnings.warn(("The least populated class in y has only %d"
```

Out[156]:

```
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 [157]:

```
grid_search.best_score_
```

Out[157]:

```
0.875
```

In [158]:

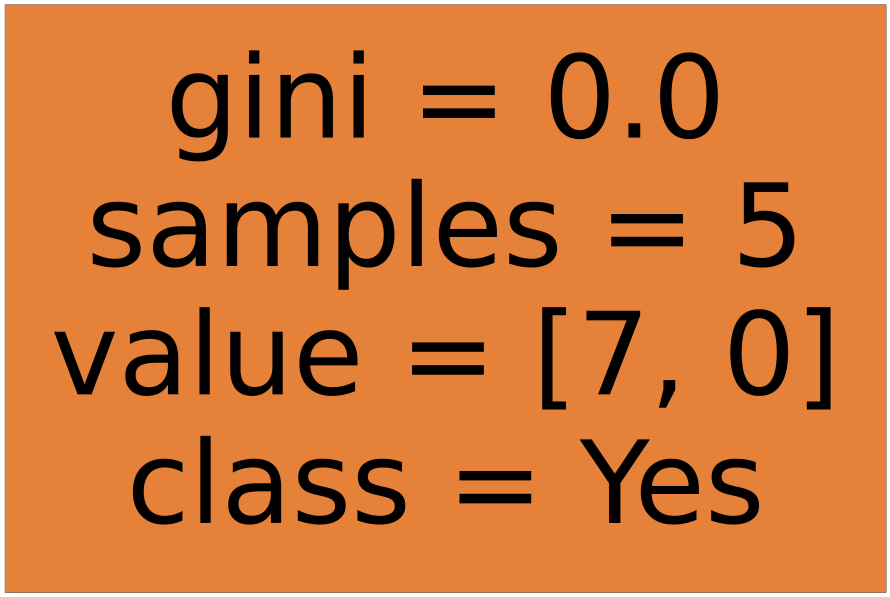
```
rfc_best=grid_search.best_estimator_
```

In [159]:

```
plt.figure(figsize=(100,50))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True)
```

Out[159]:

```
[Text(2790.0, 1359.0, 'gini = 0.0\nsamples = 5\nvalue = [7, 0]\nclass = Yes')]
```



In [ ]: