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

In [2]:
```
df1=pd.read_csv('C3_Bot_detection_data.csv')
df1
```

Out[2]:

	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

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Hashtags
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 [3]:
```
df1.head()
```

Out[3]:

	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

In [4]:
```
df1.tail()
```

Out[4]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Hashtags
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

In [5]:
```
df1.describe()
```

Out[5]:

	User ID	Retweet Count	Mention Count	Follower Count	Bot Label
count	50000.000000	50000.00000	50000.000000	50000.000000	50000.000000
mean	548890.680540	50.00560	2.513760	4988.602380	0.500360
std	259756.681425	29.18116	1.708563	2878.742898	0.500005
min	100025.000000	0.00000	0.000000	0.000000	0.000000
25%	323524.250000	25.00000	1.000000	2487.750000	0.000000
50%	548147.000000	50.00000	3.000000	4991.500000	1.000000
75%	772983.000000	75.00000	4.000000	7471.000000	1.000000
max	999995.000000	100.00000	5.000000	10000.000000	1.000000

In [6]:
```
df1.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 [7]:
```
df1["Hashtags"].value_counts()
```

Out[7]:
```
area                                  21
big                                   20
treat                                 19
ground                                18
watch                                 18
                                      ..
probably term drug spring              1
president conference field process     1
market live mouth sit wide             1
your five                              1
onto admit artist first                1
Name: Hashtags, Length: 34247, dtype: int64
```

In [8]:

```
Verified={"Verified":{True:1,False:0}}
df1=df1.replace(Verified)
df1
```

Out[8]:

	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	0	1	Adkinston	2020-05-11 15:29:50	NaN
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	1	0	Sanderston	2022-11-26 05:18:10	both live
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	1	0	Harrisonfurt	2022-08-08 03:16:54	phone ahead
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	1	1	Martinezberg	2021-08-14 22:27:05	ever quickly new I
4	704441	noah87	Animal sign six data good or.	26	3	8438	0	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	1	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	0	1	Greenbury	2022-10-18 03:57:35	add walk among believe
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	1	1	Deborahfort	2020-07-08 03:54:08	onto admit artist first

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Hashtags
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	0	0	Stephenside	2022-03-22 12:13:44	star
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	0	0	Novakberg	2022-12-03 06:11:07	home

50000 rows × 11 columns

In [9]:
```
df1=df1.dropna()
```

In [10]:
```
list(df1)
```

Out[10]:
```
['User ID',
 'Username',
 'Tweet',
 'Retweet Count',
 'Mention Count',
 'Follower Count',
 'Verified',
 'Bot Label',
 'Location',
 'Created At',
 'Hashtags']
```

In [11]:
```
x=df1[[
    'User ID',
 'Retweet Count',
 'Mention Count',
 'Follower Count',
 'Bot Label',
]]
y=df1['Verified']
```

In [12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.40)
```

In [13]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[13]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [14]:
```
rf=RandomForestClassifier()
```

In [15]:
```
params={'max_depth':[1,2,3,4,5],
        'min_samples_leaf':[2,4,6,8,10],
       }
```

In [16]:
```
from sklearn.model_selection import GridSearchCV
gs=grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring='accuracy')
gs.fit(x_train,y_train)
```

Out[16]:
```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

In [17]:
```
gs.best_score_
```

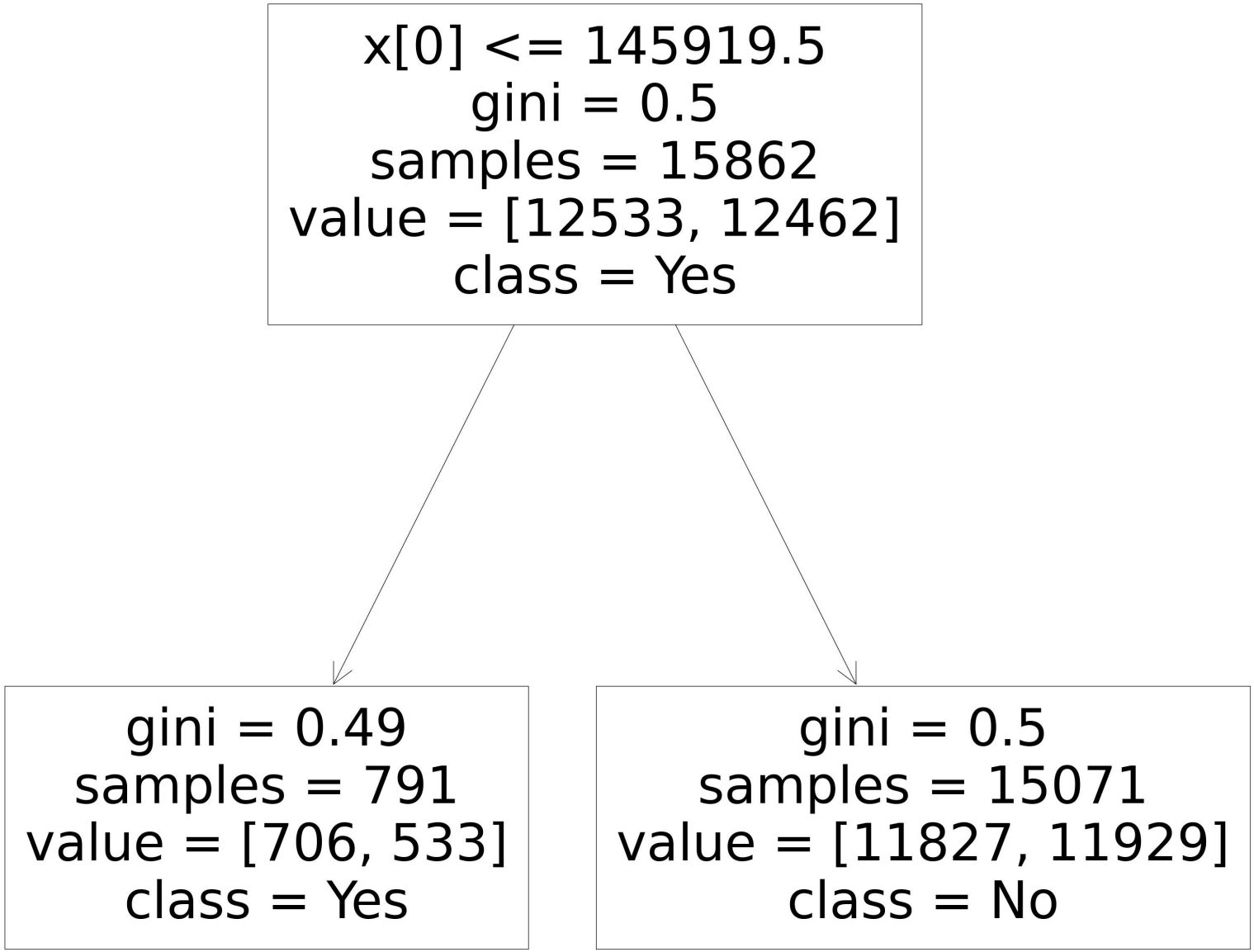
Out[17]:
```
0.5073016499796085
```

In [18]:
```
rf_best=gs.best_estimator_
rf_best
```

Out[18]:
```
RandomForestClassifier
RandomForestClassifier(max_depth=1, min_samples_leaf=8)
```

In [19]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(40,40))
plot_tree(rf_best.estimators_[4],feature_names=None,class_names=['Yes','No'])
```

Out[19]:
```
[Text(0.5, 0.75, 'x[0] <= 145919.5\ngini = 0.5\nsamples = 15862\nvalue = [12533, 12462]\nclass = Yes'),
 Text(0.25, 0.25, 'gini = 0.49\nsamples = 791\nvalue = [706, 533]\nclass = Yes'),
 Text(0.75, 0.25, 'gini = 0.5\nsamples = 15071\nvalue = [11827, 11929]\nclass = No')]
```


x[0] <= 145919.5
gini = 0.5
samples = 15862
value = [12533, 12462]
class = Yes
gini = 0.49
samples = 791
value = [706, 533]
class = Yes
gini = 0.5
samples = 15071
value = [11827, 11929]
class = No

In [20]:
```
df2=pd.read_csv("C10_Loan1.csv")
df2
```

Out[20]:

	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 [21]:
```
df2.head()
```

Out[21]:

	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

In [22]:
```
df2.tail()
```

Out[22]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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 [23]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Home Owner          10 non-null     object
 1   Marital Status      10 non-null     object
 2   Annual Income       10 non-null     int64 
 3   Defaulted Borrower  10 non-null     object
dtypes: int64(1), object(3)
memory usage: 452.0+ bytes
```

In [24]:
```
df2.describe()
```

Out[24]:

	Annual Income
count	10.000000
mean	104.000000
std	45.631373
min	60.000000
25%	77.500000
50%	92.500000
75%	115.000000
max	220.000000

In [25]:
```
list(df2)
```

Out[25]:
```
['Home Owner', 'Marital Status', 'Annual Income', 'Defaulted Borrower']
```

In [31]:
```
HomeOwner={'Home Owner':{'Yes':1,'No':0}}
df2=df2.replace(HomeOwner)
MaritalStatus={"Marital Status":{'Single':1,"Married":2,"Divorced":0}}
df2=df2.replace(MaritalStatus)
DefaultedBorrower={"Defaulted Borrower":{"Yes":1,"No":0}}
df2=df2.replace(DefaultedBorrower)
```

In [32]:
```
df2
```

Out[32]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	1	1	125	0
1	0	2	100	0
2	0	1	70	0
3	1	2	120	0
4	0	0	95	1
5	0	2	60	0
6	1	0	220	0
7	0	1	85	1
8	0	2	75	0
9	0	1	90	1

In [33]:
```
list(df2)
```

Out[33]:
```
['Home Owner', 'Marital Status', 'Annual Income', 'Defaulted Borrower']
```

In [76]:
```
X=df2[['Home Owner', 'Annual Income', 'Defaulted Borrower']]
y=df2[ 'Marital Status']
```

In [77]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.60)
```

In [78]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

Out[78]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [106]:
```
params={'max_depth':[1,2,3,4,5],
        'min_samples_leaf':[1,2,3,4,5]
        ,'n_estimators':[10,25,30,50,100,200]
       }
```

In [107]:
```
from sklearn.model_selection import GridSearchCV
gs=grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring='accuracy')
gs.fit(X_train,y_train)
```

```
C:\Users\Ajay\anaconda3\Lib\site-packages\sklearn\model_selection\_split.py:725: UserWarning: The least
populated class in y has only 1 members, which is less than n_splits=2.
  warnings.warn(
```

Out[107]:
```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
  ▸ RandomForestClassifier
```

In [108]:
```
gs.best_score_
```

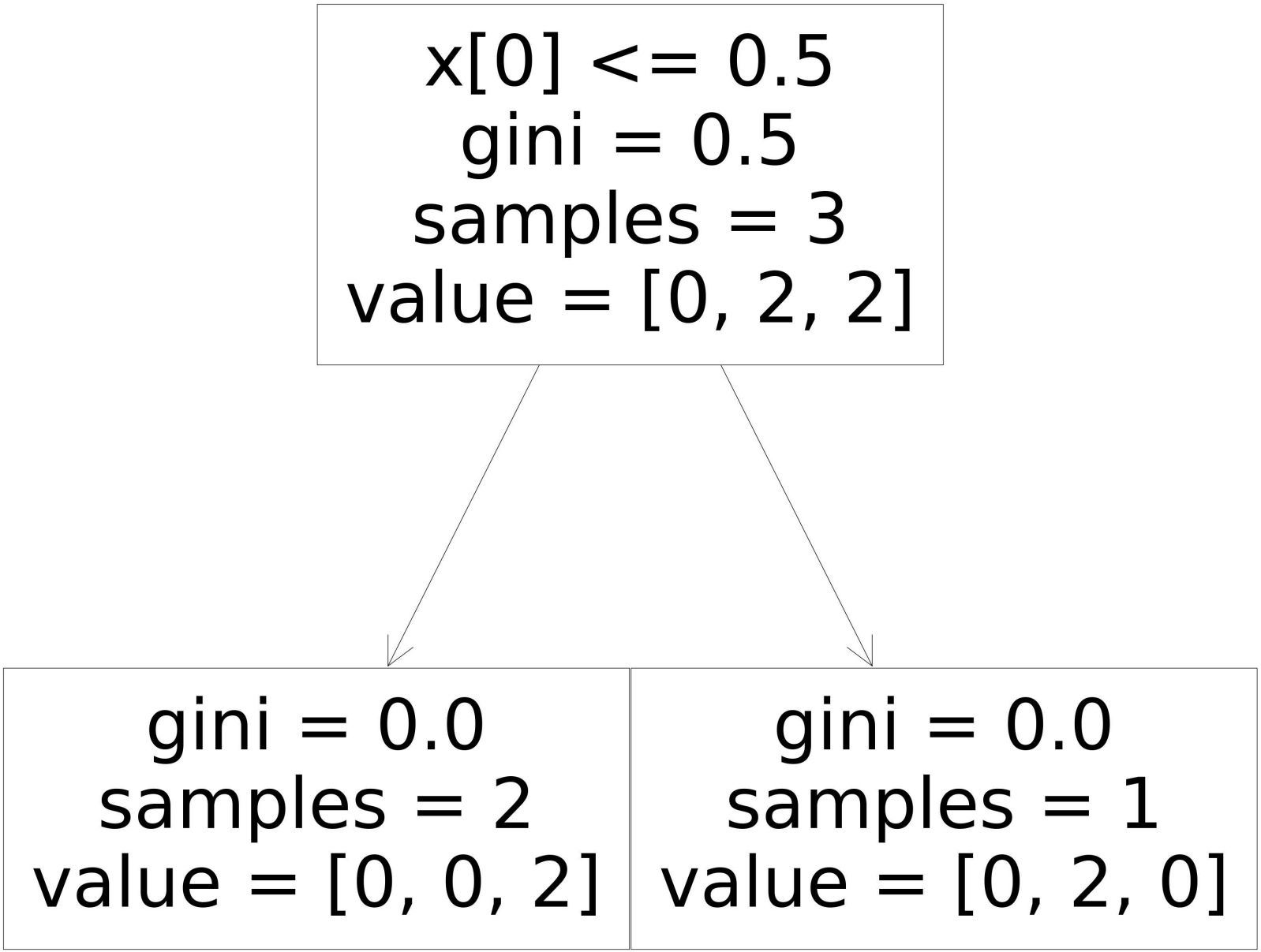
Out[108]:
```
0.5
```

In [109]:
```
rf_best=gs.best_estimator_
rf_best
```

Out[109]:
```
▾ RandomForestClassifier
RandomForestClassifier(max_depth=1, n_estimators=30)
```

In [110]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(40,40))
plot_tree(rf_best.estimators_[2],feature_names=None)
```

Out[110]:
```
[Text(0.5, 0.75, 'x[0] <= 0.5\ngini = 0.5\nsamples = 3\nvalue = [0, 2, 2]'),
 Text(0.25, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(0.75, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 2, 0]')]
```


x[0] <= 0.5
gini = 0.5
samples = 3
value = [0, 2, 2]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 1
value = [0, 2, 0]

In [ ]:

```

```