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

In [2]:
```
df = pd.read_csv(r"C:\Users\user\Downloads\C3_bot_detection_data (2).csv")[0:5000]
df
```

Out[2]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Ha
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	b
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	
...	...	...	...	...	...	...	...	...	...	...	
4995	741163	smccullough	Ago common foreign every he TV off seat never ...	69	2	9694	True	1	Louisburgh	2020-11-03 01:32:49	
4996	389863	brian57	Store hope blue civil base son improve action.	61	1	6733	True	1	Ericberg	2021-02-25 12:38:17	r
4997	510860	davidjenkins	Exist major fall include so sing last wish card.	4	2	8664	False	1	Penaview	2022-04-08 08:08:13	
4998	413100	llong	News size return authority close administratio...	59	2	2796	False	0	East Samanthafort	2020-04-20 20:45:34	ac so
4999	670684	uadams	Outside rich fact where begin later.	45	2	1956	False	0	Laurietown	2021-03-30 22:33:47	ir

5000 rows × 11 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   User ID         5000 non-null   int64 
 1   Username        5000 non-null   object
 2   Tweet           5000 non-null   object
 3   Retweet Count   5000 non-null   int64 
 4   Mention Count   5000 non-null   int64 
 5   Follower Count  5000 non-null   int64 
 6   Verified        5000 non-null   bool  
 7   Bot Label       5000 non-null   int64 
 8   Location        5000 non-null   object
 9   Created At      5000 non-null   object
 10  Hashtags        4166 non-null   object
dtypes: bool(1), int64(5), object(5)
memory usage: 395.6+ KB
```

In [4]:
```
df.columns
```

Out[4]:
```
Index(['User ID', 'Username', 'Tweet', 'Retweet Count', 'Mention Count',
       'Follower Count', 'Verified', 'Bot Label', 'Location', 'Created At',
       'Hashtags'],
      dtype='object')
```

In [5]:
```
df1 = df[['User ID','Retweet Count','Mention Count','Follower Count','Bot Label','Verified']]
df1
```

Out[5]:

	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
...	...	...	...	...	...	...
4995	741163	69	2	9694	1	True
4996	389863	61	1	6733	1	True
4997	510860	4	2	8664	1	False
4998	413100	59	2	2796	0	False
4999	670684	45	2	1956	0	False

5000 rows × 6 columns

In [6]:
```
x = df1[['User ID','Retweet Count','Mention Count','Follower Count','Bot Label']]
y = df1['Verified']
```

In [7]:
```
from sklearn.model_selection import train_test_split
```

In [8]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)
```

In [9]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [10]:
```
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[10]:
```
RandomForestClassifier()
```

In [11]:
```
parameters = {
    'max_depth':[11,12,13,14,15],
    'min_samples_leaf':[15,20,25,30,35],
    'n_estimators':[10,20,30,40,50]
}
```

In [12]:
```
from sklearn.model_selection import GridSearchCV
```

In [13]:
```
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='accuracy')
grid_search.fit(x_train,y_train)
```

Out[13]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [11, 12, 13, 14, 15],
                         'min_samples_leaf': [15, 20, 25, 30, 35],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [14]:
```
grid_search.best_score_
```

Out[14]:
```
0.5291428571428571
```

In [15]:
```
from sklearn.tree import plot_tree
```

In [16]:
```
rfc_best= grid_search.best_estimator_
```

In [18]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=["Yes","No"],filled=Tru
```

```
128\nvalue = [110, 104]\nclass = Yes'),
 Text(2568.328767123288, 752.6769230769232, 'Follower Count <= 6617.5\ngini = 0.491\nsamp
les = 91\nvalue = [66, 86]\nclass = No'),
 Text(2507.178082191781, 585.4153846153847, 'gini = 0.49\nsamples = 45\nvalue = [41, 31]
\nclass = Yes'),
 Text(2629.479452054795, 585.4153846153847, 'gini = 0.43\nsamples = 46\nvalue = [25, 55]
\nclass = No'),
 Text(2690.6301369863013, 752.6769230769232, 'gini = 0.412\nsamples = 37\nvalue = [44, 1
8]\nclass = Yes'),
 Text(2629.479452054795, 1087.2, 'gini = 0.375\nsamples = 35\nvalue = [12, 36]\nclass = N
o'),
 Text(3810.452054794521, 1421.7230769230769, 'Follower Count <= 8249.0\ngini = 0.5\nsampl
es = 769\nvalue = [612, 587]\nclass = Yes'),
 Text(3462.6575342465753, 1254.4615384615386, 'Retweet Count <= 60.5\ngini = 0.499\nsampl
es = 584\nvalue = [429, 469]\nclass = No'),
 Text(3195.123287671233, 1087.2, 'Retweet Count <= 49.5\ngini = 0.5\nsamples = 373\nvalue
= [283, 285]\nclass = No'),
 Text(3026.958904109589, 919.9384615384615, 'Retweet Count <= 35.5\ngini = 0.496\nsamples
= 300\nvalue = [207, 247]\nclass = No'),
```

In [ ]: