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

In [2]:
```
from sklearn.linear_model import LogisticRegression
```

In [3]:
```
df=pd.read_csv("C3 bot csv").dropna()

df
```

Out[3]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sa
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Ha
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Mar

In [4]:
```
df.dropna(inplace=True)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 41659 entries, 1 to 49999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   User ID         41659 non-null  int64 
 1   Username        41659 non-null  object
 2   Tweet           41659 non-null  object
 3   Retweet Count   41659 non-null  int64 
 4   Mention Count   41659 non-null  int64 
 5   Follower Count  41659 non-null  int64 
 6   Verified        41659 non-null  bool  
 7   Bot Label       41659 non-null  int64 
 8   Location        41659 non-null  object
 9   Created At      41659 non-null  object
 10  Hashtags        41659 non-null  object
dtypes: bool(1), int64(5), object(5)
memory usage: 3.5+ MB
```

In [6]:
```
feature_matrix = df[['User ID','Retweet Count','Mention Count','Follower Count
target_vector = df['Verified']
```

In [7]:
```
feature_matrix.shape
```

Out[7]:
```
(41659, 5)
```

In [8]:
```
target_vector.shape
```

Out[8]:
```
(41659,)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler
```

In [10]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [11]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[11]:
```
LogisticRegression()
```

In [12]:
```
feature_matrix.shape
```

Out[12]:
```
(41659, 5)
```

In [13]:
```
target_vector.shape
```

Out[13]:
```
(41659,)
```

In [14]:
```
from sklearn.preprocessing import StandardScaler
```

In [15]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [16]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

Out[16]:
```
LogisticRegression()
```

In [17]:
```
observation=df[['User ID','Retweet Count','Mention Count','Follower Count','Bo
```

In [18]:
```
prediction = logr.predict(observation)
prediction
```

Out[18]:
```
array([ True,  True,  True, ...,  True,  True,  True])
```

In [19]:
```
logr.classes_
```

Out[19]:
```
array([False,  True])
```

In [20]:
```
logr.predict_proba(observation)[0][1]
```

Out[20]:
```
1.0
```

Random Forest

In [21]:
```
df['Verified'].value_counts()
```

Out[21]:
```
True     20845
False    20814
Name: Verified, dtype: int64
```

In [24]:
```
x=df[['User ID','Retweet Count','Mention Count','Follower Count','Bot Label']]
y=df['Verified']
```

In [25]:
```
g1={'Verified':{'True':1, 'False':2}}
df=df.replace(g1)
df
```

Out[25]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Lo
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sand
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	Harris
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Martine
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Camac
5	570928	james00	See wonder travel this suffer less yard office...	41	4	3792	True	1	Che
...	...	...	...	...	...	...	...	...	
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Kimberly
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Gree
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Debor
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	Steph
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Nova

41659 rows × 11 columns

In [26]:
```
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 [27]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[27]:
```
RandomForestClassifier()
```

In [28]:
```
parameters = {'max_depth':[1,2,3,4,5],'min_samples_leaf':[5,10,15,20,25],
              'n_estimators': [10,20,30,40,50]
              }
```

In [29]:
```
from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="a
grid_search.fit(x_train,y_train)
```

Out[29]:
```
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')
```

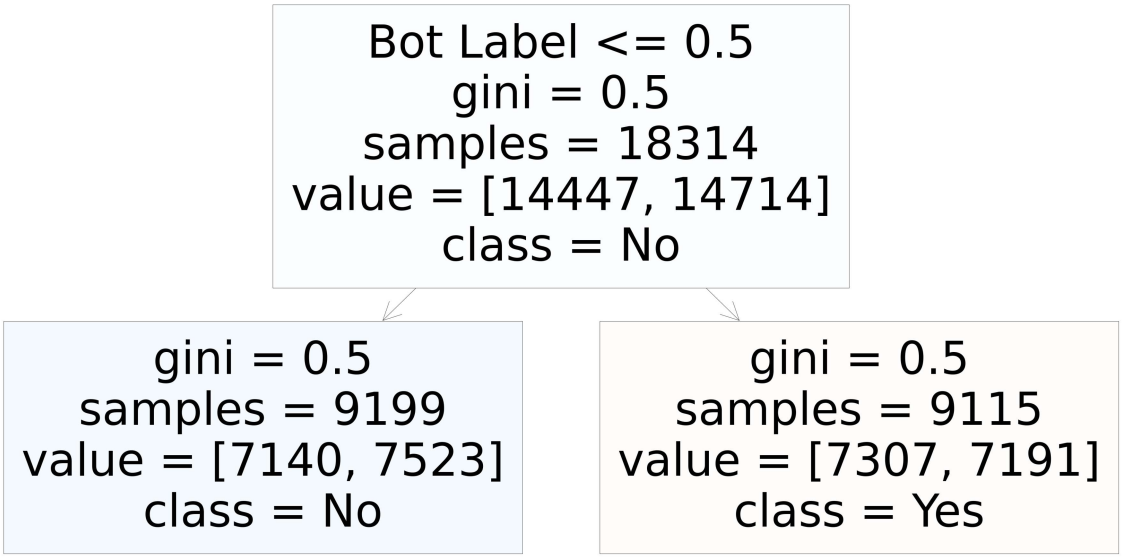
In [31]:
```
grid_search.best_score_
```

Out[31]:
```
0.5056067336441086
```

In [32]:
```
rfc_best = grid_search.best_estimator_
```

In [33]:
```
from sklearn.tree import plot_tree
plt.figure(figsize = (80,40,))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','
```

Out[33]:
```
[Text(2232.0, 1630.8000000000002, 'Bot Label <= 0.5\ngini = 0.5\nsamples = 18
314\nvalue = [14447, 14714]\nclass = No'),
 Text(1116.0, 543.5999999999999, 'gini = 0.5\nsamples = 9199\nvalue = [7140,
7523]\nclass = No'),
 Text(3348.0, 543.5999999999999, 'gini = 0.5\nsamples = 9115\nvalue = [7307,
7191]\nclass = Yes')]
```



In [ ]: