

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

```
In [5]: df=pd.read_csv(r"C:\Users\Admin\Downloads\C1_ionosphere - C1_ionosphere.csv")
df
```

Out[5]:

	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1.1	0.0376	...	-0.51171	0.41078	-0.4611
0	1	0	1.00000	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1.00000	-0.04549	...	-0.26569	-0.20468	-0.1841
1	1	0	1.00000	-0.03365	1.00000	0.00485	1.00000	-0.12062	0.88965	0.01198	...	-0.40220	0.58984	-0.2211
2	1	0	1.00000	-0.45161	1.00000	1.00000	0.71216	-1.00000	0.00000	0.00000	...	0.90695	0.51613	1.0000
3	1	0	1.00000	-0.02401	0.94140	0.06531	0.92106	-0.23255	0.77152	-0.16399	...	-0.65158	0.13290	-0.5321
4	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	...	-0.01535	-0.03240	0.0921
...
345	1	0	0.83508	0.08298	0.73739	-0.14706	0.84349	-0.05567	0.90441	-0.04622	...	-0.04202	0.83479	0.0011
346	1	0	0.95113	0.00419	0.95183	-0.02723	0.93438	-0.01920	0.94590	0.01606	...	0.01361	0.93522	0.0491
347	1	0	0.94701	-0.00034	0.93207	-0.03227	0.95177	-0.03431	0.95584	0.02446	...	0.03193	0.92489	0.0251
348	1	0	0.90608	-0.01657	0.98122	-0.01989	0.95691	-0.03646	0.85746	0.00110	...	-0.02099	0.89147	-0.0771
349	1	0	0.84710	0.13533	0.73638	-0.06151	0.87873	0.08260	0.88928	-0.09139	...	-0.15114	0.81147	-0.0481

350 rows × 35 columns



```
In [7]: df['g'].value_counts()
```

```
Out[7]: g    224
b     126
Name: g, dtype: int64
```

```
In [8]: x=df.drop('g',axis=1)
y=df['g']
```

```
In [9]: 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

      0.0376  ... -0.51171  0.41078 -0.46168  0.21266 -0.3409  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  -0.453  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 [10]: from sklearn.model_selection import train_test_split
```

```
In [11]: x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

```
In [12]: from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
Out[12]: RandomForestClassifier()
```

```
In [13]: parameters={'max_depth':[1,53,67,53,57],
                    'min_samples_leaf':[5,34,67,87,45],
                    'n_estimators':[23,45,67,23,56]}
```

```
In [14]: 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)
```

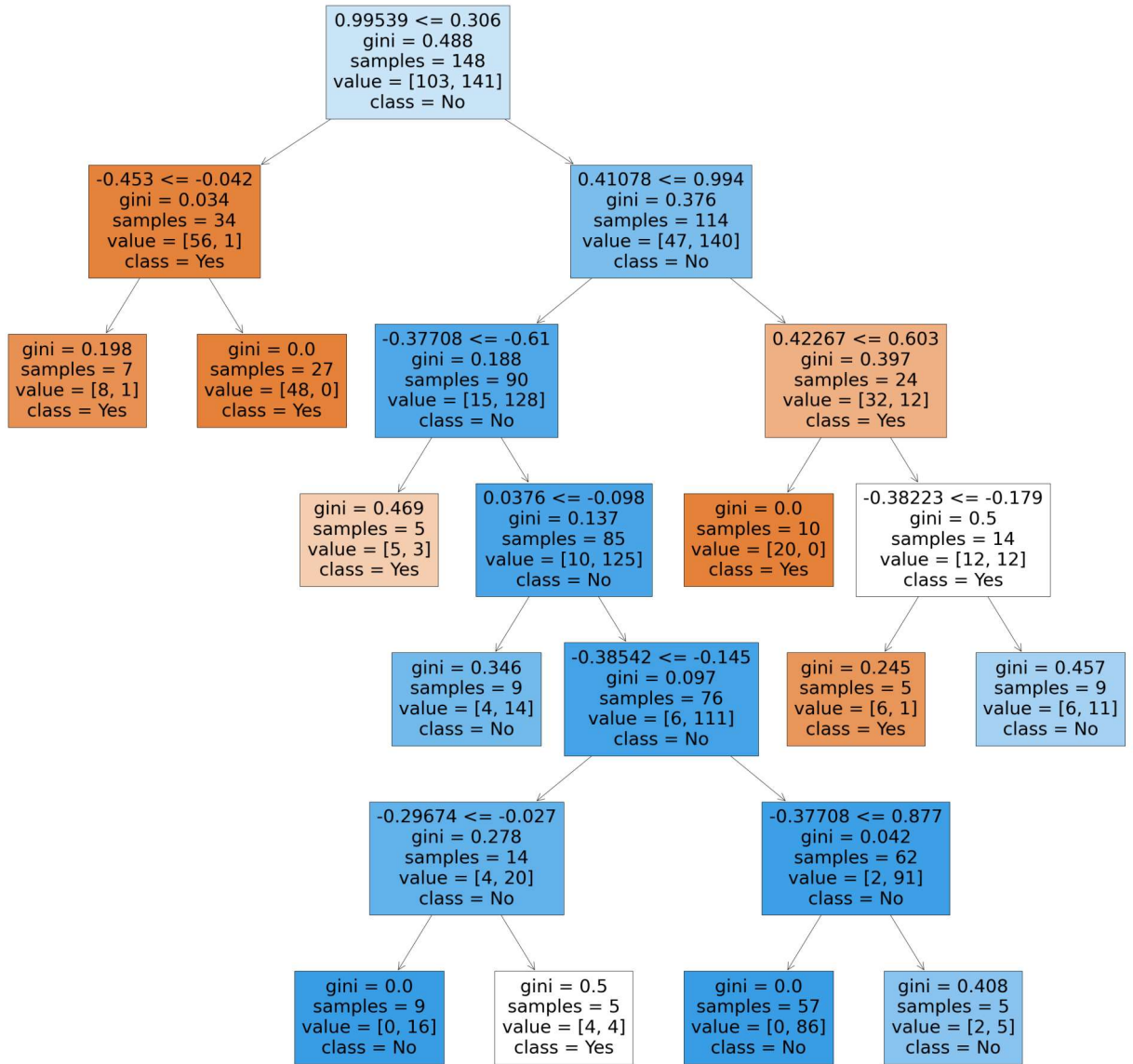
```
Out[14]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
                    param_grid={'max_depth': [1, 53, 67, 53, 57],
                                'min_samples_leaf': [5, 34, 67, 87, 45],
                                'n_estimators': [23, 45, 67, 23, 56]},
                    scoring='accuracy')
```

```
In [15]: grid_search.best_score_
```

```
Out[15]: 0.9139344262295082
```

```
In [16]: rfc_best=grid_search.best_estimator_
from sklearn.tree import plot_tree
plt.figure(figsize=(50,49))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'],filled=True
```

```
Out[16]: [Text(1046.25, 2473.38, '0.99539 <= 0.306\ngini = 0.488\nsamples = 148\nvalue = [103, 141]\nclass = No'),
Text(465.0, 2092.8599999999997, '-0.453 <= -0.042\ngini = 0.034\nsamples = 34\nvalue = [56, 1]\nclass = Yes'),
Text(232.5, 1712.34, 'gini = 0.198\nsamples = 7\nvalue = [8, 1]\nclass = Yes'),
Text(697.5, 1712.34, 'gini = 0.0\nsamples = 27\nvalue = [48, 0]\nclass = Yes'),
Text(1627.5, 2092.8599999999997, '0.41078 <= 0.994\ngini = 0.376\nsamples = 114\nvalue = [47, 140]\nclass = No'),
Text(1162.5, 1712.34, '-0.37708 <= -0.61\ngini = 0.188\nsamples = 90\nvalue = [15, 128]\nclass = No'),
Text(930.0, 1331.82, 'gini = 0.469\nsamples = 5\nvalue = [5, 3]\nclass = Yes'),
Text(1395.0, 1331.82, '0.0376 <= -0.098\ngini = 0.137\nsamples = 85\nvalue = [10, 125]\nclass = No'),
Text(1162.5, 951.3, 'gini = 0.346\nsamples = 9\nvalue = [4, 14]\nclass = No'),
Text(1627.5, 951.3, '-0.38542 <= -0.145\ngini = 0.097\nsamples = 76\nvalue = [6, 111]\nclass = No'),
Text(1162.5, 570.7800000000002, '-0.29674 <= -0.027\ngini = 0.278\nsamples = 14\nvalue = [4, 20]\nclass = No'),
Text(930.0, 190.25999999999997, 'gini = 0.0\nsamples = 9\nvalue = [0, 16]\nclass = No'),
Text(1395.0, 190.25999999999997, 'gini = 0.5\nsamples = 5\nvalue = [4, 4]\nclass = Yes'),
Text(2092.5, 570.7800000000002, '-0.37708 <= 0.877\ngini = 0.042\nsamples = 62\nvalue = [2, 91]\nclass = No'),
Text(1860.0, 190.25999999999997, 'gini = 0.0\nsamples = 57\nvalue = [0, 86]\nclass = No'),
Text(2325.0, 190.25999999999997, 'gini = 0.408\nsamples = 5\nvalue = [2, 5]\nclass = No'),
Text(2092.5, 1712.34, '0.42267 <= 0.603\ngini = 0.397\nsamples = 24\nvalue = [32, 12]\nclass = Yes'),
Text(1860.0, 1331.82, 'gini = 0.0\nsamples = 10\nvalue = [20, 0]\nclass = Yes'),
Text(2325.0, 1331.82, '-0.38223 <= -0.179\ngini = 0.5\nsamples = 14\nvalue = [12, 12]\nclass = Yes'),
Text(2092.5, 951.3, 'gini = 0.245\nsamples = 5\nvalue = [6, 1]\nclass = Yes'),
Text(2557.5, 951.3, 'gini = 0.457\nsamples = 9\nvalue = [6, 11]\nclass = No')]
```



C2

```
In [18]: df1=pd.read_csv(r"C:\Users\Admin\Downloads\C2_test.gender_submission - C2_test.gender_submission.csv")
df1
```

```
Out[18]:
```

	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 [19]: d=df1.dropna()
d
```

```
Out[19]:
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
12	904	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	21228	82.2667	B45	S
14	906	1	Chaffee, Mrs. Herbert Fuller (Carrie Constance...	female	47.0	1	0	W.E.P. 5734	61.1750	E31	S
24	916	1	Ryerson, Mrs. Arthur Larned (Emily Maria Borie)	female	48.0	1	3	PC 17608	262.3750	B57 B59 B63 B66	C
26	918	1	Ostby, Miss. Helene Ragnhild	female	22.0	0	1	113509	61.9792	B36	C
28	920	1	Brady, Mr. John Bertram	male	41.0	0	0	113054	30.5000	A21	S
...
404	1296	1	Frauenthal, Mr. Isaac Gerald	male	43.0	1	0	17765	27.7208	D40	C
405	1297	2	Nourney, Mr. Alfred (Baron von Drachstedt)"	male	20.0	0	0	SC/PARIS 2166	13.8625	D38	C
407	1299	1	Widener, Mr. George Dunton	male	50.0	1	1	113503	211.5000	C80	C
411	1303	1	Minahan, Mrs. William Edward (Lillian E Thorpe)	female	37.0	1	0	19928	90.0000	C78	Q
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C

87 rows × 11 columns

```
In [20]: d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87 entries, 12 to 414
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     87 non-null    int64
1   Pclass          87 non-null    int64
2   Name            87 non-null    object
3   Sex             87 non-null    object
4   Age             87 non-null    float64
5   SibSp           87 non-null    int64
6   Parch           87 non-null    int64
7   Ticket          87 non-null    object
8   Fare            87 non-null    float64
9   Cabin           87 non-null    object
10  Embarked        87 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 8.2+ KB
```

```
In [21]: a=d[['PassengerId', 'Pclass', 'SibSp', 'Parch', 'Embarked']]
a
```

Out[21]:

	PassengerId	Pclass	SibSp	Parch	Embarked
12	904	1	1	0	S
14	906	1	1	0	S
24	916	1	1	3	C
26	918	1	0	1	C
28	920	1	0	0	S
...
404	1296	1	1	0	C
405	1297	2	0	0	C
407	1299	1	1	1	C
411	1303	1	1	0	Q
414	1306	1	0	0	C

87 rows × 5 columns

```
In [22]: a['Embarked'].value_counts()
```

Out[22]: C 47
S 39
Q 1
Name: Embarked, dtype: int64

```
In [23]: x=a.drop('Embarked',axis=1)
y=a['Embarked']
```

```
In [24]: g1={"Embarked":{'S':1,'C':2,'Q':5}}
a=a.replace(g1)
print(a)
```

	PassengerId	Pclass	SibSp	Parch	Embarked
12	904	1	1	0	1
14	906	1	1	0	1
24	916	1	1	3	2
26	918	1	0	1	2
28	920	1	0	0	1
..
404	1296	1	1	0	2
405	1297	2	0	0	2
407	1299	1	1	1	2
411	1303	1	1	0	5
414	1306	1	0	0	2

[87 rows x 5 columns]

```
In [25]: 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 [26]: from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[26]: RandomForestClassifier()


```
In [27]: parameters={'max_depth':[1,53,67,53,57],  
                  'min_samples_leaf':[5,34,67,87,45],  
                  'n_estimators':[23,45,67,23,56]}
```

```
In [28]: 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)
```

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\model_selection_split.py:670: 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[28]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),  
                    param_grid={'max_depth': [1, 53, 67, 53, 57],  
                                'min_samples_leaf': [5, 34, 67, 87, 45],  
                                'n_estimators': [23, 45, 67, 23, 56]},  
                    scoring='accuracy')
```

```
In [29]: grid_search.best_score_
```

```
Out[29]: 0.5833333333333333
```

```
In [31]: rfc_best=grid_search.best_estimator_
```

```
In [33]: from sklearn.tree import plot_tree  
plt.figure(figsize=(50,49))  
plot_tree(rfc_best.estimators_[3],feature_names=x.columns,class_names=['Yes','No'],filled=True
```

C3

```
In [35]: a1=pd.read_csv(r"C:\Users\Admin\Downloads\C3_bot_detection_data - C3_bot_detection_data.csv")
a1
```

Out[35]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Has
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adkinson	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	bc
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	i n
...	
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	edu
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury	2022-10-18 03:57:35	ad ; t
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort	2020-07-08 03:54:08	art
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 × 11 columns



In [36]:

a2=a1.dropna()
a2

Out[36]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Location	Created At	Has
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	Sanderston	2022-11-26 05:18:10	bc
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	f n
5	570928	james00	See wonder travel this suffer less yard office...	41	4	3792	True	1	West Cheyenne	2023-05-07 22:24:47	ε re pt i
...	
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	edu
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	Greenbury	2022-10-18 03:57:35	ad : t
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	Deborahfort	2020-07-08 03:54:08	art
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	

41659 rows × 11 columns

In [37]: a2.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 [44]: a3=a2[['User ID', 'Retweet Count', 'Mention Count', 'Follower Count', 'Bot Label', 'Verified']]
a3

Out[44]:

	User ID	Retweet Count	Mention Count	Follower Count	Bot Label	Verified
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
5	570928	41	4	3792	1	True
...
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

41659 rows × 6 columns

In [45]: a3['Verified'].value_counts()

Out[45]: True 20845
False 20814
Name: Verified, dtype: int64In [46]: x=a3.drop('Verified',axis=1)
y=a3['Verified']

```
In [47]: g1={"Verified":{"True":4,'False':2}}  
a3=a3.replace(g1)  
print(a3)
```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-47-598d56ee55a5> in <module>
      1 g1={"Verified":{'True':4,'False':2}}
----> 2 a3=a3.replace(g1)
      3 print(a3)

~\anaconda3\lib\site-packages\pandas\core\frame.py in replace(self, to_replace, value, inplace, limit, regex, method)
    4164         method="pad",
    4165     ):
-> 4166         return super().replace(
    4167             to_replace=to_replace,
    4168             value=value,

~\anaconda3\lib\site-packages\pandas\core\generic.py in replace(self, to_replace, value, inplace, limit, regex, method)
    6643         to_replace, value = keys, values
    6644
-> 6645         return self.replace(
    6646             to_replace, value, inplace=inplace, limit=limit, regex=regex
    6647         )

~\anaconda3\lib\site-packages\pandas\core\frame.py in replace(self, to_replace, value, inplace, limit, regex, method)
    4164         method="pad",
    4165     ):
-> 4166         return super().replace(
    4167             to_replace=to_replace,
    4168             value=value,

~\anaconda3\lib\site-packages\pandas\core\generic.py in replace(self, to_replace, value, inplace, limit, regex, method)
    6660                                     # object conversion is handled in
    6661                                     # series.replace which is called recursively
-> 6662         res[c] = res[c].replace(
    6663             to_replace=src,
    6664             value=value[c],

~\anaconda3\lib\site-packages\pandas\core\series.py in replace(self, to_replace, value, inplace, limit, regex, method)
    4170         method="pad",
    4171     ):
-> 4172         return super().replace(
    4173             to_replace=to_replace,
    4174             value=value,

~\anaconda3\lib\site-packages\pandas\core\generic.py in replace(self, to_replace, value, inplace, limit, regex, method)
    6693         )
    6694
-> 6695         new_data = self._data.replace_list(
    6696             src_list=to_replace,
    6697             dest_list=value,

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in replace_list(self, src_list, dest_list, inplace, regex)
    611         return _compare_or_regex_search(values, s, regex)
    612
--> 613         masks = [comp(s, regex) for i, s in enumerate(src_list)]
    614
    615         result_blocks = []

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in <listcomp>(.0)
    611         return _compare_or_regex_search(values, s, regex)

```

```

612
--> 613         masks = [comp(s, regex) for i, s in enumerate(src_list)]
614
615         result_blocks = []

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in comp(s, regex)
609             maybe_convert_objects(values), s.asm8, regex
610         )
--> 611         return _compare_or_regex_search(values, s, regex)
612
613         masks = [comp(s, regex) for i, s in enumerate(src_list)]

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in _compare_or_regex_search
(a, b, regex)
1933         type_names[1] = f"ndarray(dtype={b.dtype})"
1934
-> 1935         raise TypeError(
1936             f"Cannot compare types {repr(type_names[0])} and {repr(type_names[1])}"
1937         )

```

TypeError: Cannot compare types 'ndarray(dtype=bool)' and 'str'

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':[1,53,67,53,65],
 'min_samples_leaf':[5,34,67,97,45],
 'n_estimators':[5,45,67,23,56]}`

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, 53, 67, 53, 65],
 'min_samples_leaf': [5, 34, 67, 97, 45],
 'n_estimators': [5, 45, 67, 23, 56]},
 scoring='accuracy')`

In [52]: `grid_search.best_score_`

Out[52]: `0.5042007920561823`

In [53]: `rfc_best=grid_search.best_estimato`

```

-----
AttributeError                                Traceback (most recent call last)
<ipython-input-53-1624dda19930> in <module>
----> 1 rfc_best=grid_search.best_estimato

AttributeError: 'GridSearchCV' object has no attribute 'best_estimato'

```

In []: `plt.figure(figsize=(34,21))
plot_tree(rfc_best.estimators_[4],feature_names=x.columns,class_names=['Yes','No'],filled=True)`

In []: