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
```