

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

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

```
In [4]: df = pd.read_csv(r"C:\Users\user\Downloads\C3_bot_detection_data.csv")  
df
```

Out[4]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	Loc
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	Adki
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	Camact
...
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	Stephe
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	Nova

50000 rows × 11 columns

```
In [5]: df.columns
```

```
Out[5]: Index(['User ID', 'Username', 'Tweet', 'Retweet Count', 'Mention Count',  
             'Follower Count', 'Verified', 'Bot Label', 'Location', 'Created At',  
             'Hashtags'],  
            dtype='object')
```

```
In [23]: feature_matrix = df[['User ID', 'Retweet Count', 'Mention Count',  
                             'Follower Count', 'Bot Label']]  
target_vector = df['Verified']
```

```
In [24]: feature_matrix.shape
```

```
Out[24]: (50000, 5)
```

```
In [25]: from sklearn.preprocessing import StandardScaler
```

```
In [26]: fs = StandardScaler().fit_transform(feature_matrix)
```

```
In [27]: logs = LogisticRegression()  
logs.fit(fs, target_vector)
```

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

```
In [28]: observation = [[1.4, 1.5, 1.6, 2, 2.1]]  
prediction = logs.predict(observation)
```

```
In [29]: print(prediction)
```

```
[ True]
```

```
In [30]: logs.classes_
```

```
Out[30]: array([False,  True])
```

```
In [31]: logs.predict_proba(observation)[0][0]
```

```
Out[31]: 0.48896843942135226
```

```
In [ ]:
```