In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

DATA SET 1 C3

In [2]:

```
from sklearn.linear_model import LogisticRegression
```

In [3]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C3_bot_detection_data.csv")
a
```

Out[3]:

	User ID	Username	Tweet	Retweet Count	Mention Count	Follower Count	Verified	Bot Label	
0	132131	flong	Station activity person against natural majori...	85	1	2353	False	1	
1	289683	hinesstephanie	Authority research natural life material staff...	55	5	9617	True	0	S
2	779715	roberttran	Manage whose quickly especially foot none to g...	6	2	4363	True	0	H
3	696168	pmason	Just cover eight opportunity strong policy which.	54	5	2242	True	1	Ma
4	704441	noah87	Animal sign six data good or.	26	3	8438	False	1	Cai
...	...	...	...	...	...	...	...	...	
49995	491196	uberg	Want but put card direction know miss former h...	64	0	9911	True	1	Kiml
49996	739297	jessicamunoz	Provide whole maybe agree church respond most ...	18	5	9900	False	1	(
49997	674475	lynncunningham	Bring different everyone international capital...	43	3	6313	True	1	D
49998	167081	richardthompson	Than about single generation itself seek sell ...	45	1	6343	False	0	St
49999	311204	daniel29	Here morning class various room human true bec...	91	4	4006	False	0	N

50000 rows × 11 columns

In [49]:

```
f=a[['User ID','Retweet Count','Mention Count','Follower Count','Bot Label']]
f1=a['Verified']
```

In [50]:

```
f.shape
```

Out[50]:

```
(50000, 5)
```

In [51]:

```
f1.shape
```

Out[51]:

```
(50000,)
```

In [52]:

```
from sklearn.preprocessing import StandardScaler
```

In [53]:

```
b=StandardScaler().fit_transform(f)
```

In [54]:

```
c=LogisticRegression()
c.fit(b,f1)
```

Out[54]:

```
LogisticRegression()
```

In [55]:

```
d=[[12,22,32,42,52]]
```

In [56]:

```
e=c.predict(d)
print(e)
```

```
[ True]
```

In [43]:

```
c.classes_
```

Out[43]:

```
array([False,  True])
```

In [57]:

```
c.predict_proba(d)[0][0]
```

Out[57]:

```
0.3646324586423182
```

In [58]:

```
c.predict_proba(d)[0][1]
```

Out[58]:

```
0.6353675413576818
```

DATA SET 2 C4

In [62]:

```
P=pd.read_csv(r"C:\Users\user\Downloads\C4_framingham.csv")
P
```

Out[62]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentH
0	1	39	4.0	0	0.0	0.0	0	
1	0	46	2.0	0	0.0	0.0	0	
2	1	48	1.0	1	20.0	0.0	0	
3	0	61	3.0	1	30.0	0.0	0	
4	0	46	3.0	1	23.0	0.0	0	
...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.0	0	
4234	1	51	3.0	1	43.0	0.0	0	
4235	0	48	2.0	1	20.0	NaN	0	
4236	0	44	1.0	1	15.0	0.0	0	
4237	0	52	2.0	0	0.0	0.0	0	

4238 rows × 16 columns

In [65]:

```
t=P.fillna(value=40)
t
```

Out[65]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentH
0	1	39	4.0	0	0.0	0.0	0	
1	0	46	2.0	0	0.0	0.0	0	
2	1	48	1.0	1	20.0	0.0	0	
3	0	61	3.0	1	30.0	0.0	0	
4	0	46	3.0	1	23.0	0.0	0	
...	...	...	...	...	...	...	...	
4233	1	50	1.0	1	1.0	0.0	0	
4234	1	51	3.0	1	43.0	0.0	0	
4235	0	48	2.0	1	20.0	40.0	0	
4236	0	44	1.0	1	15.0	0.0	0	
4237	0	52	2.0	0	0.0	0.0	0	

4238 rows × 16 columns

In [83]:

```
Q=t[['male','age','education','currentSmoker','totChol']]
Q1=t['TenYearCHD']
```

In [84]:

```
Q.shape
```

Out[84]:

```
(4238, 5)
```

In [85]:

```
Q1.shape
```

Out[85]:

```
(4238,)
```

In [86]:

```
from sklearn.preprocessing import StandardScaler
```

In [88]:

```
b=StandardScaler().fit_transform(Q)
```

In [89]:

```
c=LogisticRegression()
c.fit(b,Q1)
```

Out[89]:

```
LogisticRegression()
```

In [90]:

```
d=[[12,22,32,42,52]]
```

In [91]:

```
e=c.predict(d)
print(e)
```

```
[1]
```

In [92]:

```
c.classes_
```

Out[92]:

```
array([0, 1], dtype=int64)
```

In [93]:

```
c.predict_proba(d)[0][0]
```

Out[93]:

```
1.0309531006669204e-12
```

In [94]:

```
c.predict_proba(d)[0][1]
```

Out[94]:

```
0.999999999998969
```

DATA SET 3 C5

In [95]:

```
v=pd.read_csv(r"C:\Users\user\Downloads\C5_health care diabetes.csv")
v
```

Out[95]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFun
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

768 rows × 9 columns

In [97]:

```
o=v.iloc[:,0:8]
o1=v.iloc[:,-1]
```

In [98]:

```
o.shape
```

Out[98]:

```
(768, 8)
```

In [99]:

```
o1.shape
```

Out[99]:

```
(768,)
```

In [100]:

```
from sklearn.preprocessing import StandardScaler
```

In [101]:

```
b=StandardScaler().fit_transform(o)
```

In [102]:

```
c=LogisticRegression()
c.fit(b,o1)
```

Out[102]:

```
LogisticRegression()
```

In [103]:

```
d=[[12,22,32,42,52,89,65,76]]
```

In [104]:

```
e=c.predict(d)
print(e)
```

```
[1]
```

In [105]:

```
c.classes_
```

Out[105]:

```
array([0, 1], dtype=int64)
```

In [106]:

```
c.predict_proba(d)[0][0]
```

Out[106]:

```
0.0
```

In [107]:

```
c.predict_proba(d)[0][1]
```

Out[107]:

```
1.0
```

DATASET C6

In [108]:

```
l=pd.read_csv(r"C:\Users\user\Downloads\C6_bmi.csv")
l
```

Out[108]:

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

In [141]:

```
k=l.iloc[:,1:4]
k1=l.iloc[:,0]
```

In [142]:

```
k.shape
```

Out[142]:

```
(500, 3)
```

In [143]:

```
k1.shape
```

Out[143]:

```
(500,)
```

In [144]:

```
from sklearn.preprocessing import StandardScaler
```

In [145]:

```
b=StandardScaler().fit_transform(k)
```

In [146]:

```
c=LogisticRegression()
c.fit(b,k1)
```

Out[146]:

```
LogisticRegression()
```

In [147]:

```
d=[[12,22,36]]
```

In [148]:

```
e=c.predict(d)
print(e)
```

```
['Male']
```

In [149]:

```
c.classes_
```

Out[149]:

```
array(['Female', 'Male'], dtype=object)
```

In [150]:

```
c.predict_proba(d)[0][0]
```

Out[150]:

```
0.032630220489630046
```

In [151]:

```
c.predict_proba(d)[0][1]
```

Out[151]:

```
0.96736977951037
```

DATASET C7

In [156]:

```
j=pd.read_csv(r"C:\Users\user\Downloads\c7_used_cars.csv")
j
```

Out[156]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSiz
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1
...	...	...	...	...	...	...	...	...	...	
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1

99187 rows × 11 columns

In [157]:

```
r=j[['Unnamed: 0','year','price','mileage','tax','mpg','engineSize']]
d=j['Make']
```

In [158]:

```
r.shape
```

Out[158]:

```
(99187, 7)
```

In [159]:

```
d.shape
```

Out[159]:

```
(99187,)
```

In [160]:

```
b=StandardScaler().fit_transform(r)
```

In [161]:

```
c=LogisticRegression()
c.fit(b,d)
```

Out[161]:

```
LogisticRegression()
```

In [162]:

```
d=[[12,22,32,42,52,98,56]]
```

In [163]:

```
e=c.predict(d)
print(e)
```

```
['BMW']
```

In [164]:

```
c.classes_
```

Out[164]:

```
array(['Audi', 'BMW', 'VW', 'ford', 'hyundi', 'merc', 'skoda', 'toyota',
       'vauxhall'], dtype=object)
```

In [165]:

```
c.predict_proba(d)[0][0]
```

Out[165]:

```
1.601360239301771e-54
```

In [166]:

```
c.predict_proba(d)[0][1]
```

Out[166]:

```
0.9999999999684899
```

DATASET C8 TRAIN

In [167]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C8_loan-train.csv")
a
```

Out[167]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [208]:

```
e=a.fillna(value=70)
e
```

Out[208]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [209]:

```
t=e[['ApplicantIncome','CoapplicantIncome','Loan_Amount_Term','Property_Area']]
t
```

Out[209]:

	ApplicantIncome	CoapplicantIncome	Loan_Amount_Term	Property_Area
0	5849	0.0	360.0	Urban
1	4583	1508.0	360.0	Rural
2	3000	0.0	360.0	Urban
3	2583	2358.0	360.0	Urban
4	6000	0.0	360.0	Urban
...	...	...	...	...
609	2900	0.0	360.0	Rural
610	4106	0.0	180.0	Rural
611	8072	240.0	360.0	Urban
612	7583	0.0	360.0	Urban
613	4583	0.0	360.0	Semiurban

614 rows × 4 columns

In [210]:

```
r=t.iloc[:,0:3]
r1=t.iloc[:,-1]
```

In [211]:

```
r.shape
```

Out[211]:

```
(614, 3)
```

In [212]:

```
r1.shape
```

Out[212]:

```
(614,)
```

In [213]:

```
b=StandardScaler().fit_transform(r)
```

In [214]:

```
c=LogisticRegression()
c.fit(b,r1)
```

Out[214]:

```
LogisticRegression()
```

In [215]:

```
d=[[12,22,32]]
```

In [216]:

```
e=c.predict(d)
print(e)
```

```
['Semiurban']
```

In [217]:

```
c.classes_
```

Out[217]:

```
array(['Rural', 'Semiurban', 'Urban'], dtype=object)
```

In [218]:

```
c.predict_proba(d)[0][0]
```

Out[218]:

```
0.36379658643513385
```

In [219]:

```
c.predict_proba(d)[0][1]
```

Out[219]:

```
0.6353265137357376
```

DATASET C8 TEST

In [220]:

```
m1=pd.read_csv(r"C:\Users\user\Downloads\C8_loan-test.csv")
m1
```

Out[220]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001015	Male	Yes	0	Graduate	No	5720	
1	LP001022	Male	Yes	1	Graduate	No	3076	
2	LP001031	Male	Yes	2	Graduate	No	5000	
3	LP001035	Male	Yes	2	Graduate	No	2340	
4	LP001051	Male	No	0	Not Graduate	No	3276	
...	...	...	...	...	...	...	...	
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	
363	LP002975	Male	Yes	0	Graduate	No	4158	
364	LP002980	Male	No	0	Graduate	No	3250	
365	LP002986	Male	Yes	0	Graduate	No	5000	
366	LP002989	Male	No	0	Graduate	Yes	9200	

367 rows × 12 columns

In [221]:

```
m1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            367 non-null    object 
 1   Gender             356 non-null    object 
 2   Married            367 non-null    object 
 3   Dependents         357 non-null    object 
 4   Education          367 non-null    object 
 5   Self_Employed      344 non-null    object 
 6   ApplicantIncome    367 non-null    int64  
 7   CoapplicantIncome  367 non-null    int64  
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   361 non-null    float64
 10  Credit_History     338 non-null    float64
 11  Property_Area      367 non-null    object 
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB
```

In [222]:

```
m2=m1.fillna(40)
m2
```

Out[222]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001015	Male	Yes	0	Graduate	No	5720	
1	LP001022	Male	Yes	1	Graduate	No	3076	
2	LP001031	Male	Yes	2	Graduate	No	5000	
3	LP001035	Male	Yes	2	Graduate	No	2340	
4	LP001051	Male	No	0	Not Graduate	No	3276	
...	...	...	...	...	...	...	...	
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	
363	LP002975	Male	Yes	0	Graduate	No	4158	
364	LP002980	Male	No	0	Graduate	No	3250	
365	LP002986	Male	Yes	0	Graduate	No	5000	
366	LP002989	Male	No	0	Graduate	Yes	9200	

367 rows × 12 columns

In [223]:

```
y=m2[['ApplicantIncome','CoapplicantIncome','Loan_Amount_Term','Property_Area']]
y
```

Out[223]:

	ApplicantIncome	CoapplicantIncome	Loan_Amount_Term	Property_Area
0	5720	0	360.0	Urban
1	3076	1500	360.0	Urban
2	5000	1800	360.0	Urban
3	2340	2546	360.0	Urban
4	3276	0	360.0	Urban
...	...	...	...	...
362	4009	1777	360.0	Urban
363	4158	709	360.0	Urban
364	3250	1993	360.0	Semiurban
365	5000	2393	360.0	Rural
366	9200	0	180.0	Rural

367 rows × 4 columns

In [225]:

```
r=y.iloc[:,0:3]
r1=y.iloc[:,-1]
```

In [226]:

```
r.shape
```

Out[226]:

```
(367, 3)
```

In [227]:

```
r1.shape
```

Out[227]:

```
(367,)
```

In [228]:

```
b=StandardScaler().fit_transform(r)
```

In [229]:

```
c=LogisticRegression()
c.fit(b,r1)
```

Out[229]:

```
LogisticRegression()
```

In [230]:

```
d=[[12,22,32]]
```

In [231]:

```
e=c.predict(d)
print(e)
```

```
['Rural']
```

In [232]:

```
c.classes_
```

Out[232]:

```
array(['Rural', 'Semiurban', 'Urban'], dtype=object)
```

In [233]:

```
c.predict_proba(d)[0][0]
```

Out[233]:

```
0.9241639552899842
```

In [234]:

```
c.predict_proba(d)[0][1]
```

Out[234]:

```
0.0537634314275885
```

DATASET C9

In [235]:

```
n1=pd.read_csv(r"C:\Users\user\Downloads\C9_Data.csv")
n1
```

Out[235]:

	row_id	user_id	timestamp	gate_id
0	0	18	2022-07-29 09:08:54	7
1	1	18	2022-07-29 09:09:54	9
2	2	18	2022-07-29 09:09:54	9
3	3	18	2022-07-29 09:10:06	5
4	4	18	2022-07-29 09:10:08	5
...	...	...	...	...
37513	37513	6	2022-12-31 20:38:56	11
37514	37514	6	2022-12-31 20:39:22	6
37515	37515	6	2022-12-31 20:39:23	6
37516	37516	6	2022-12-31 20:39:31	9
37517	37517	6	2022-12-31 20:39:31	9

37518 rows × 4 columns

In [254]:

```
x=n1[['row_id', 'user_id']]
y=n1['gate_id']
```

In [255]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [256]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[256]:

```
LinearRegression()
```

In [257]:

```
print(lr.intercept_)
```

```
7.31553712314673
```

In [258]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[258]:

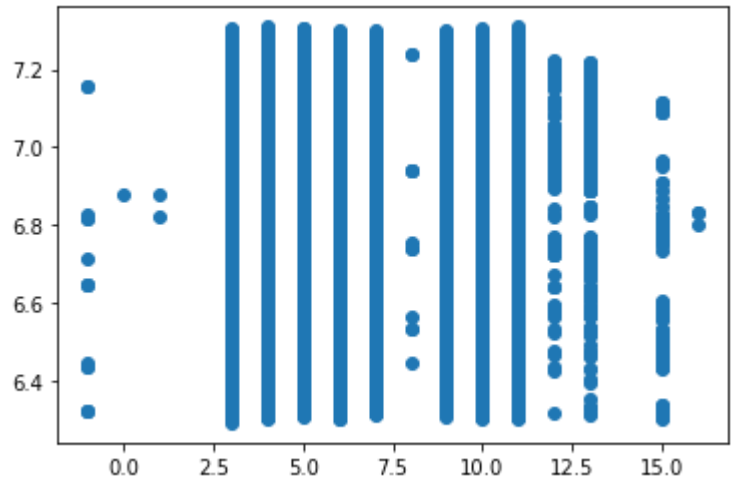
	Co-efficient
row_id	-0.000006
user_id	-0.014525

In [259]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[259]:

```
<matplotlib.collections.PathCollection at 0x1e5ee3c99d0>
```



In [260]:

```
print(lr.score(x_test,y_test))
```

```
0.001672361377533682
```

In [261]:

```
lr.score(x_train,y_train)
```

Out[261]:

```
0.0068756859455977315
```

In [236]:

```
n1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37518 entries, 0 to 37517
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   row_id     37518 non-null  int64 
 1   user_id    37518 non-null  int64 
 2   timestamp  37518 non-null  object
 3   gate_id    37518 non-null  int64 
dtypes: int64(3), object(1)
memory usage: 1.1+ MB
```

In [243]:

```
p=n1.iloc[:,0:2]
p1=n1.iloc[:,-1]
```

In [244]:

```
p.shape
```

Out[244]:

```
(37518, 2)
```

In [245]:

```
p1.shape
```

Out[245]:

```
(37518,)
```

In [246]:

```
b=StandardScaler().fit_transform(p)
```

In [247]:

```
c=LogisticRegression()
c.fit(b,p1)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.
py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sc
ikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression (https://scikit-learn.org/stable/modules/linear_model.html#logisti
c-regression)
  n_iter_i = _check_optimize_result(
```

Out[247]:

```
LogisticRegression()
```

In [248]:

```
d=[[12,22]]
```

In [249]:

```
e=c.predict(d)
print(e)
```

```
[-1]
```

In [250]:

```
c.classes_
```

Out[250]:

```
array([-1,  0,  1,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 1
6],
      dtype=int64)
```

In [251]:

```
c.predict_proba(d)[0][0]
```

Out[251]:

```
0.9656133937999277
```

In [252]:

```
c.predict_proba(d)[0][1]
```

Out[252]:

```
1.1207500772129741e-13
```