In [93]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [94]:
```
from sklearn.linear_model import LogisticRegression
```

In [95]:
```
df=pd.read_csv("Data.csv").dropna()
df
```

Out[95]:

| | 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 [96]:
```
df.dropna(inplace=True)
```

In [97]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 1.4+ MB
```

In [98]:
```
feature_matrix = df[['row_id','user_id']]
target_vector = df['gate_id']
```

In [99]:
```
feature_matrix.shape
```

Out[99]:
```
(37518, 2)
```

In [100…
```
target_vector.shape
```

Out[100…
```
(37518,)
```

In [101…
```
from sklearn.preprocessing import StandardScaler
```

In [102…
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [103…
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: Conver
genceWarning: 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 in:
    https://scikit-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-regression
  n_iter_i = _check_optimize_result(
```

Out[103…
```
LogisticRegression()
```

In [104…
```
feature_matrix.shape
```

Out[104…
```
(37518, 2)
```

In [105…
```
target_vector.shape
```

Out[105…
```
(37518,)
```

In [106…
```
from sklearn.preprocessing import StandardScaler
```

In [107…
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [108…
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: Conver
genceWarning: 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 in:
    https://scikit-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-regression
  n_iter_i = _check_optimize_result(
```

Out[108…
```
LogisticRegression()
```

In [109…
```
observation=df[['row_id','user_id']]
```

In [110…
```
prediction = logr.predict(observation)
prediction
```

Out[110…
```
array([-1, -1, -1, ..., 16, 16, 16], dtype=int64)
```

In [111…
```
logr.classes_
```

Out[111…
```
array([-1,  0,  1,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16],
      dtype=int64)
```

In [112…
```
logr.predict_proba(observation)[0][1]
```

Out[112…
```
1.7263815682078809e-09
```

In [113…
```
from sklearn.linear_model import Ridge,Lasso
```

In [114…
```
x = df[['row_id','user_id']]
y = df['gate_id']
```

In [115…
```
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 [116…
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
rr.score(x_train,y_train)
```

Out[116…
```
0.0045668615621417
```

In [117…
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

Out[117…
```
LinearRegression()
```

In [118…
```
lr.intercept_
```

Out[118…
```
7.255120438415789
```

In [119…
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[119…

| | Co-efficient |
|---|---|
| **row_id** | -0.000006 |
| **user_id** | -0.011816 |

In [120…
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[120… `<matplotlib.collections.PathCollection at 0x1747faf4580>`