

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
In [3]: df=pd.read_csv(r"C9_Data.csv")
df
```

Out[3]:

	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 [4]: df.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 [5]: df=df.dropna()
```

In [6]: `df.describe()`

Out[6]:

	row_id	user_id	gate_id
count	37518.000000	37518.000000	37518.000000
mean	18758.500000	28.219015	6.819607
std	10830.658036	17.854464	3.197746
min	0.000000	0.000000	-1.000000
25%	9379.250000	12.000000	4.000000
50%	18758.500000	29.000000	6.000000
75%	28137.750000	47.000000	10.000000
max	37517.000000	57.000000	16.000000

In [7]: `df.columns`

Out[7]: Index(['row_id', 'user_id', 'timestamp', 'gate_id'], dtype='object')

In [8]: `x=df[['row_id', 'user_id']]`
`y=df['gate_id']`
`x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)`
`lr=LogisticRegression()`
`lr.fit(x_train,y_train)`

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 in:
<https://scikit-learn.org/stable/modules/preprocessing.html> (<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 (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
`n_iter_i = _check_optimize_result(`

Out[8]: `LogisticRegression()`

In [9]: `lr.predict(x_test)`

Out[9]: array([4, 4, 4, ..., 4, 4, 4], dtype=int64)

In [10]: `lr.score(x_test,y_test)`

Out[10]: 0.22032693674484718

```
In [11]: from sklearn.preprocessing import StandardScaler
fs=StandardScaler().fit_transform(x)
logr=LogisticRegression()
logr.fit(fs,y)
```

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 in:
<https://scikit-learn.org/stable/modules/preprocessing.html> (<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 (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
n_iter_i = _check_optimize_result(

Out[11]: LogisticRegression()

```
In [13]: o=[[34,52]]
prediction=logr.predict(o)
print(prediction)

[-1]
```

```
In [14]: logr.classes_
```

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

```
In [15]: logr.predict_proba(o)[0][0]
```

Out[15]: 0.99962911956057

```
In [16]: logr.predict_proba(o)[0][1]
```

Out[16]: 1.0517042278090056e-30

```
In [ ]:
```