

```
In [66]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [67]: from sklearn.linear_model import LogisticRegression
```

```
In [143]: df=pd.read_csv(r"C:\Users\Admin\Downloads\C9_Data - C9_Data.csv")
df
```

Out[143]:

	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 [144]: 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 [145]: df1=df.dropna()
```

```
In [152]: d1=df1.iloc[:,0:2]
d2=df1.iloc[:, -1]
```

```
In [153]: d1.shape
```

```
Out[153]: (37518, 2)
```

```
In [154]: d2.shape
```

```
Out[154]: (37518,)
```

```
In [155]: from sklearn.preprocessing import StandardScaler
```

```
In [156]: a=StandardScaler().fit_transform(d1)
```

```
In [157]: lr=LogisticRegression()  
lr.fit(a,d2)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:  
762: 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[157]: LogisticRegression()
```

```
In [158]: obs=[[10,12]]
```

```
In [159]: pdt=lr.predict(obs)  
print(pdt)
```

```
[15]
```

```
In [160]: lr.classes_
```

```
Out[160]: array([-1,  0,  1,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16],  
          dtype=int64)
```

```
In [161]: lr.predict_proba(obs)[0][0]
```

```
Out[161]: 0.41718404568394063
```

```
In [162]: lr.predict_proba(obs)[0][1]
```

```
Out[162]: 1.7613109507315681e-09
```

In []:

```
In [163]: from sklearn.model_selection import train_test_split
d1_train,d1_test,d2_train,d2_test = train_test_split(d1,d2,test_size=0.3)
```

```
In [164]: from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(d1_train,d2_train)
```

Out[164]: LinearRegression()

```
In [165]: print(lr.intercept_)
```

7.290875105954916

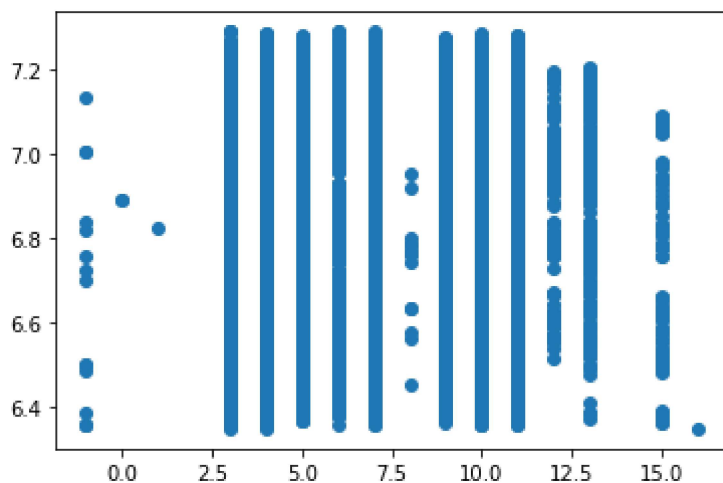
```
In [167]: coeff = pd.DataFrame(lr.coef_,d1.columns,columns=["Co-efficient"])
coeff
```

Out[167]:

	Co-efficient
row_id	-0.000006
user_id	-0.012787

```
In [168]: prediction= lr.predict(d1_test)
plt.scatter(d2_test,prediction)
```

Out[168]: <matplotlib.collections.PathCollection at 0x2a933587fa0>



```
In [169]: print(lr.score(d1_test,d2_test))
```

0.0056897674619974525

```
In [171]: print(lr.score(d1_train,d2_train))
```

0.00543828955052883

In []: