In [1]:
```python
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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

In [4]:
```python
# To Import Dataset
sd=pd.read_csv(r"c:\Users\user\Downloads\C9_Data.csv")
sd
```

Out[4]:

	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 [5]:
```
sd.dropna()
sd
```

Out[5]:
```
       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 [6]:
```
sd.fillna(20)
```

Out[6]:
```
       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 [8]:
```
feature_matrix = sd[['row_id','user_id']]
target_vector=sd['gate_id']
```

```
In [9]: feature_matrix.shape

Out[9]: (37518, 2)

In [10]: target_vector.shape

Out[10]: (37518,)

In [11]: from sklearn.preprocessing import StandardScaler

In [12]: fs=StandardScaler().fit_transform(feature_matrix)

In [13]: logr= LogisticRegression()
         logr.fit(fs,target_vector)

         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://sciki
         t-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-regres
         sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
         ession)
           n_iter_i = _check_optimize_result(

Out[13]: LogisticRegression()

In [14]: observation =[[1.2,2.3,3.3]]

In [15]: logr.classes_

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

In [17]:
```
logr.predict_proba(observation)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-17-7c5bc94db2a6> in <module>
----> 1 logr.predict_proba(observation)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py
in predict_proba(self, X)
   1469             return super()._predict_proba_lr(X)
   1470         else:
-> 1471             decision = self.decision_function(X)
   1472             if decision.ndim == 1:
   1473                 # Workaround for multi_class="multinomial" and binary
outcomes

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_base.py in d
ecision_function(self, X)
    286         n_features = self.coef_.shape[1]
    287         if X.shape[1] != n_features:
--> 288             raise ValueError("X has %d features per sample; expecting
%d"
    289                              % (X.shape[1], n_features))
    290

ValueError: X has 3 features per sample; expecting 2
```

In [ ]:
```

```