

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

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

```
In [3]: df=pd.read_csv("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=df.dropna()
df
```

```
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

	row_id	user_id	timestamp	gate_id
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]: `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 [6]: `df.columns`

Out[6]: `Index(['row_id', 'user_id', 'timestamp', 'gate_id'], dtype='object')`

In [7]: `feature_matrix=df[['row_id', 'user_id']]
target_vector=df['gate_id']`

In [8]: `feature_matrix.shape`

Out[8]: `(37518, 2)`

In [9]: `target_vector.shape`

Out[9]: `(37518,)`

In [10]: `from sklearn.preprocessing import StandardScaler`

In [11]: `fs=StandardScaler().fit_transform(feature_matrix)`

In [12]: `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>
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[12]: LogisticRegression()
```

```
In [13]: observation=[[1,2]]
```

```
In [14]: prediction=logr.predict(observation)
print(prediction)
```

```
[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 [16]: logr.predict_proba(observation)[0][0]
```

```
Out[16]: 0.005365176788164149
```

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

```
Out[17]: array([[5.36517679e-03, 2.43221075e-05, 9.36568351e-05, 2.22025633e-01,
                 2.19695882e-01, 7.52352405e-02, 5.84513730e-02, 7.17956781e-02,
                 2.68284044e-03, 7.98655513e-02, 1.24425419e-01, 1.07054385e-01,
                 2.51118120e-03, 7.57336969e-03, 2.68214159e-05, 2.29125763e-02,
                 2.60893089e-04]])
```

```
In [18]: x=df[['row_id', 'user_id']]
y=df['gate_id']
```

```
In [19]: 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 [20]: from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

```
Out[20]: LinearRegression()
```

```
In [21]: lr.intercept_
```

```
Out[21]: 7.286168664592197
```

```
In [22]: coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[22]:

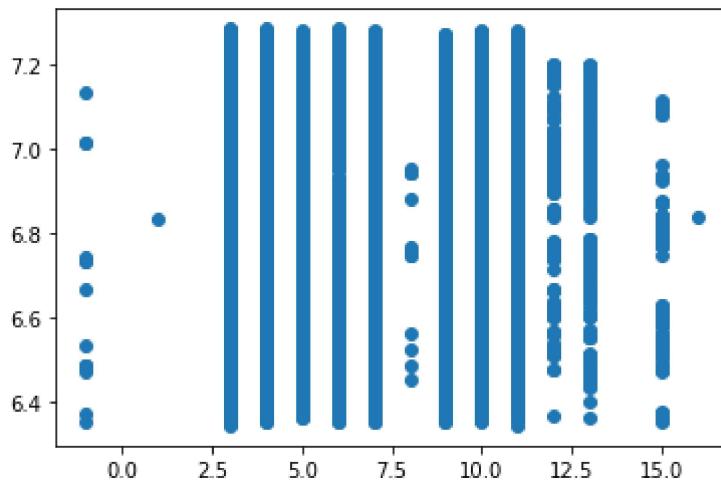
Co-efficient

row_id	-0.000005
user_id	-0.013397

In [23]:

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

Out[23]: <matplotlib.collections.PathCollection at 0x19b81fe1730>



In [24]:

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

Out[24]: 0.0046982671522645125

In [25]:

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

Out[25]: 0.005821756323376892

In [26]:

```
df['gate_id'].value_counts()
```

```
4      8170
3      5351
10     4767
5      4619
11     4090
9      3390
7      3026
6      1800
13     1201
12     698
15     298
-1      48
8      48
1       5
16      4
0       2
14      1
Name: gate_id, dtype: int64
```

```
In [27]: x=df[['row_id', 'user_id']]
y=df[ 'gate_id']
```

```
In [28]: #g1={ 'TenYearCHD':{'Audi':1, 'BMW':2, 'VW':3, 'ford':4, 'hyundai':5, 'merc':6, 'skoda':7,
#df=df.replace(g1)
#df
```

```
In [29]: from sklearn.model_selection import train_test_split
```

```
In [30]: x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

```
In [31]: from sklearn.ensemble import RandomForestClassifier
```

```
In [32]: rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
Out[32]: RandomForestClassifier()
```

```
In [33]: parameters={'max_depth':[1,2,3,4,5],
'min_samples_leaf':[5,10,15,20,25],
'n_estimators':[10,20,30,40,50]
}
```

```
In [34]: from sklearn.model_selection import GridSearchCV
grid_search =GridSearchCV(estimator=rfc,param_grid=parameters, cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
Out[34]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
param_grid={'max_depth': [1, 2, 3, 4, 5],
'min_samples_leaf': [5, 10, 15, 20, 25],
'n_estimators': [10, 20, 30, 40, 50]},
scoring='accuracy')
```

```
In [35]: grid_search.best_score_
```

```
Out[35]: 0.22336455715482445
```

```
In [36]: rfc_best=grid_search.best_estimator_
```

```
In [37]: from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['a','b','c','d',
```

```
Out[37]: [Text(2291.785714285714, 1993.2, 'user_id <= 49.5\ngini = 0.871\nsamples = 16565\nvalue = [44, 3, 6, 3815, 5631, 3324, 1280, 2115, 29, 2227\nn3326, 2964, 464, 829, 204, 1]\nclass = e'),  
Text(1275.4285714285713, 1630.8000000000002, 'user_id <= 16.0\ngini = 0.871\nsamples = 13796\nvalue = [29, 3, 6, 2742, 4699, 3044, 950, 1718, 29, 1997\nn2804, 2513, 435, 774, 160, 1]\nclass = e'),  
Text(637.7142857142857, 1268.4, 'row_id <= 16313.5\ngini = 0.855\nsamples = 5490\nvalue = [7, 0, 0, 828, 2037, 1592, 332, 678, 4, 736, 1203\nn1070, 53, 190, 28, 0]\nclass = e'),  
Text(318.85714285714283, 906.0, 'row_id <= 1382.0\ngini = 0.85\nsamples = 2159\nvalue = [4, 0, 0, 330, 861, 646, 155, 280, 2, 201, 426\nn424, 27, 95, 0, 0]\nclass = e'),  
Text(159.42857142857142, 543.5999999999999, 'user_id <= 7.5\ngini = 0.833\nsamples = 176\nvalue = [0, 0, 0, 59, 74, 49, 16, 17, 0, 8, 20, 37, 2\nn5, 0, 0]\nclass = e'),  
Text(79.71428571428571, 181.19999999999982, 'gini = 0.836\nsamples = 93\nvalue = [0, 0, 0, 42, 32, 19, 14, 10, 0, 7, 9, 24, 1\nn1, 0, 0]\nclass = d'),  
Text(239.1428571428571, 181.19999999999982, 'gini = 0.798\nsamples = 83\nvalue = [0, 0, 0, 17, 42, 30, 2, 7, 0, 1, 11, 13, 1\nn4, 0, 0]\nclass = e'),  
Text(478.2857142857142, 543.5999999999999, 'row_id <= 16133.5\ngini = 0.85\nsamples = 1983\nvalue = [4, 0, 0, 271, 787, 597, 139, 263, 2, 193, 406\nn387, 25, 90, 0, 0]\nclass = e'),  
Text(398.57142857142856, 181.19999999999982, 'gini = 0.849\nsamples = 1949\nvalue = [4, 0, 0, 270, 778, 587, 125, 257, 2, 192, 398\nn376, 25, 88, 0, 0]\nclass = e'),  
Text(558.0, 181.19999999999982, 'gini = 0.843\nsamples = 34\nvalue = [0, 0, 0, 1, 9, 10, 14, 6, 0, 1, 8, 11, 0, 2\nn0, 0]\nclass = g'),  
Text(956.5714285714284, 906.0, 'row_id <= 29100.5\ngini = 0.857\nsamples = 3331\nvalue = [3, 0, 0, 498, 1176, 946, 177, 398, 2, 535, 777\nn646, 26, 95, 28, 0]\nclass = e'),  
Text(797.1428571428571, 543.5999999999999, 'user_id <= 13.0\ngini = 0.857\nsamples = 2044\nvalue = [3, 0, 0, 282, 701, 593, 105, 269, 0, 378, 464\nn387, 15, 56, 0, 0]\nclass = e'),  
Text(717.4285714285713, 181.19999999999982, 'gini = 0.864\nsamples = 1567\nvalue = [3, 0, 0, 279, 501, 408, 94, 215, 0, 234, 366\nn305, 15, 56, 0, 0]\nclass = e'),  
Text(876.8571428571428, 181.19999999999982, 'gini = 0.811\nsamples = 477\nvalue = [0, 0, 0, 3, 200, 185, 11, 54, 0, 144, 98, 82\nn0, 0, 0, 0]\nclass = e'),  
Text(1116.0, 543.5999999999999, 'row_id <= 31469.0\ngini = 0.855\nsamples = 1287\nvalue = [0, 0, 0, 216, 475, 353, 72, 129, 2, 157, 313\nn259, 11, 39, 28, 0]\nclass = e'),  
Text(1036.2857142857142, 181.19999999999982, 'gini = 0.857\nsamples = 428\nvalue = [0, 0, 0, 99, 146, 117, 34, 45, 0, 37, 119, 101\nn4, 11, 5, 0]\nclass = e'),  
Text(1195.7142857142856, 181.19999999999982, 'gini = 0.852\nsamples = 859\nvalue = [0, 0, 0, 117, 329, 236, 38, 84, 2, 120, 194, 158\nn7, 28, 23, 0]\nclass = e'),  
Text(1913.1428571428569, 1268.4, 'row_id <= 16230.0\ngini = 0.878\nsamples = 8306\nvalue = [22, 3, 6, 1914, 2662, 1452, 618, 1040, 25, 1261\nn1601, 1443, 382, 584, 132, 1]\nclass = e'),  
Text(1594.2857142857142, 906.0, 'user_id <= 19.5\ngini = 0.888\nsamples = 3742\nvalue = [13, 3, 2, 767, 1085, 726, 300, 459, 10, 568, 651\nn588, 306, 439, 48, 0]\nclass = e'),  
Text(1434.8571428571427, 543.5999999999999, 'user_id <= 18.5\ngini = 0.866\nsamples = 826\nvalue = [0, 0, 0, 133, 150, 93, 33, 69, 0, 71, 113, 103\nn243, 291, 0, 0]\nclass = n'),  
Text(1355.142857142857, 181.19999999999982, 'gini = 0.864\nsamples = 445\nvalue = [0, 0, 0, 24, 83, 92, 23, 29, 0, 63, 72, 39\nn118, 157, 0, 0]\nclass = n'),  
Text(1514.5714285714284, 181.19999999999982, 'gini = 0.84\nsamples = 381\nvalue = [0, 0, 0, 109, 67, 1, 10, 40, 0, 8, 41, 64, 125\nn134, 0, 0]\nclass = n'),  
Text(1753.7142857142856, 543.5999999999999, 'user_id <= 48.5\ngini = 0.876\nsamples = 2916\nvalue = [13, 3, 2, 634, 935, 633, 267, 390, 10, 497, 538\nn485, 63, 148, 48, 0]\nclass = e'),  
Text(1673.9999999999998, 181.19999999999982, 'gini = 0.874\nsamples = 2676\nvalue = [13, 3, 2, 624, 877, 569, 250, 378, 10, 404, 492\nn436, 63, 148, 15, 0]\nclass = e'),  
Text(1833.4285714285713, 181.19999999999982, 'gini = 0.848\nsamples = 240\nvalue = [0, 0, 0, 10, 58, 64, 17, 12, 0, 93, 46, 49, 0\nn0, 33, 0]\nclass = j'),  
Text(2232.0, 906.0, 'user_id <= 34.5\ngini = 0.866\nsamples = 4564\nvalue = [9, 0, 4, 147, 1577, 726, 318, 581, 15, 693, 950\nn855, 76, 145, 84, 1]\nclass = e'),  
Text(2072.5714285714284, 543.5999999999999, 'user_id <= 32.5\ngini = 0.862\nsamples = 2326\nvalue = [2, 0, 4, 741, 776, 209, 185, 342, 8, 307, 453\nn451, 61, 84, 51, 1]\nclass = e'),  
Text(1992.8571428571427, 181.19999999999982, 'gini = 0.858\nsamples = 1934\nvalue = [2, 0, 4, 607, 702, 183, 97, 278, 4, 234, 367\nn386, 61, 84, 51, 1]\nclass = e'),  
Text(2152.285714285714, 181.19999999999982, 'gini = 0.86\nsamples = 392\nvalue = [0, 0,
```

```

0, 134, 74, 26, 88, 64, 4, 73, 86, 65, 0\n0, 0, 0]\nclass = d'),
Text(2391.428571428571, 543.5999999999999, 'row_id <= 37176.5\ngini = 0.861\nsamples =
2238\nvalue = [7, 0, 0, 406, 801, 517, 133, 239, 7, 386, 497\n404, 15, 61, 33, 0]\nclass
= e'),
Text(2311.7142857142853, 181.19999999999982, 'gini = 0.861\nsamples = 2218\nvalue = [7,
0, 0, 404, 782, 514, 130, 238, 7, 386, 496\n401, 15, 61, 33, 0]\nclass = e'),
Text(2471.142857142857, 181.19999999999982, 'gini = 0.615\nsamples = 20\nvalue = [0, 0,
0, 2, 19, 3, 3, 1, 0, 0, 1, 3, 0, 0\n0, 0]\nclass = e'),
Text(3308.142857142857, 1630.800000000002, 'row_id <= 23332.5\ngini = 0.847\nsamples =
2769\nvalue = [15, 0, 0, 1073, 932, 280, 330, 397, 0, 230, 522\n451, 29, 55, 44, 0]\ncla
ss = d'),
Text(2790.0, 1268.4, 'row_id <= 2370.0\ngini = 0.839\nsamples = 1786\nvalue = [7, 0, 0,
735, 593, 154, 264, 263, 0, 91, 337\n263, 27, 48, 0, 0]\nclass = d'),
Text(2630.5714285714284, 906.0, 'user_id <= 51.5\ngini = 0.787\nsamples = 156\nvalue =
[0, 0, 0, 84, 64, 9, 22, 13, 0, 1, 16, 34, 1\n6, 0, 0]\nclass = d'),
Text(2550.8571428571427, 543.5999999999999, 'gini = 0.77\nsamples = 20\nvalue = [0, 0,
0, 8, 8, 2, 9, 0, 0, 0, 2, 2, 0, 0\n0, 0]\nclass = g'),
Text(2710.285714285714, 543.5999999999999, 'user_id <= 54.5\ngini = 0.78\nsamples = 136
\nvalue = [0, 0, 0, 76, 56, 7, 13, 13, 0, 1, 14, 32, 1\n6, 0, 0]\nclass = d'),
Text(2630.5714285714284, 181.19999999999982, 'gini = 0.765\nsamples = 88\nvalue = [0,
0, 0, 47, 42, 5, 1, 10, 0, 1, 7, 19, 1, 6\n0, 0]\nclass = d'),
Text(2790.0, 181.19999999999982, 'gini = 0.779\nsamples = 48\nvalue = [0, 0, 0, 29, 14,
2, 12, 3, 0, 0, 7, 13, 0, 0\n0, 0]\nclass = d'),
Text(2949.428571428571, 906.0, 'row_id <= 3140.0\ngini = 0.842\nsamples = 1630\nvalue =
[7, 0, 0, 651, 529, 145, 242, 250, 0, 90, 321\n229, 26, 42, 0, 0]\nclass = d'),
Text(2869.7142857142853, 543.5999999999999, 'gini = 0.852\nsamples = 49\nvalue = [0, 0,
0, 10, 13, 3, 0, 8, 0, 13, 17, 5, 2, 5\n0, 0]\nclass = k'),
Text(3029.142857142857, 543.5999999999999, 'user_id <= 55.5\ngini = 0.84\nsamples = 158
1\nvalue = [7, 0, 0, 641, 516, 142, 242, 242, 0, 77, 304\n224, 24, 37, 0, 0]\nclass =
d),
Text(2949.428571428571, 181.19999999999982, 'gini = 0.843\nsamples = 1353\nvalue = [1,
0, 0, 545, 418, 131, 239, 200, 0, 63, 261\n176, 24, 37, 0, 0]\nclass = d'),
Text(3108.8571428571427, 181.19999999999982, 'gini = 0.807\nsamples = 228\nvalue = [6,
0, 0, 96, 98, 11, 3, 42, 0, 14, 43, 48, 0\n0, 0, 0]\nclass = e'),
Text(3826.2857142857138, 1268.4, 'row_id <= 26824.5\ngini = 0.856\nsamples = 983\nvalue
= [8, 0, 0, 338, 339, 126, 66, 134, 0, 139, 185\n188, 2, 7, 44, 0]\nclass = e'),
Text(3507.428571428571, 906.0, 'user_id <= 54.5\ngini = 0.866\nsamples = 263\nvalue =
[0, 0, 0, 77, 65, 65, 34, 33, 0, 43, 61, 50, 0\n1, 0, 0]\nclass = d'),
Text(3347.999999999995, 543.5999999999999, 'row_id <= 25471.5\ngini = 0.859\nsamples =
178\nvalue = [0, 0, 0, 52, 50, 42, 12, 22, 0, 43, 37, 29, 0\n1, 0, 0]\nclass = d'),
Text(3268.285714285714, 181.19999999999982, 'gini = 0.848\nsamples = 99\nvalue = [0, 0,
0, 42, 24, 19, 9, 16, 0, 21, 18, 10, 0\n1, 0, 0]\nclass = d'),
Text(3427.7142857142853, 181.19999999999982, 'gini = 0.844\nsamples = 79\nvalue = [0,
0, 0, 10, 26, 23, 3, 6, 0, 22, 19, 19, 0\n0, 0, 0]\nclass = e'),
Text(3666.8571428571427, 543.5999999999999, 'row_id <= 26416.5\ngini = 0.849\nsamples =
85\nvalue = [0, 0, 0, 25, 15, 23, 22, 11, 0, 0, 24, 21, 0\n0, 0, 0]\nclass = d'),
Text(3587.142857142857, 181.19999999999982, 'gini = 0.841\nsamples = 65\nvalue = [0, 0,
0, 22, 14, 21, 10, 8, 0, 0, 16, 12, 0\n0, 0, 0]\nclass = d'),
Text(3746.5714285714284, 181.19999999999982, 'gini = 0.784\nsamples = 20\nvalue = [0,
0, 0, 3, 1, 2, 12, 3, 0, 0, 8, 9, 0, 0\n0, 0]\nclass = g'),
Text(4145.142857142857, 906.0, 'row_id <= 29778.5\ngini = 0.845\nsamples = 720\nvalue =
[8, 0, 0, 261, 274, 61, 32, 101, 0, 96, 124, 138\n2, 6, 44, 0]\nclass = e'),
Text(3985.7142857142853, 543.5999999999999, 'row_id <= 28935.0\ngini = 0.79\nsamples =
163\nvalue = [3, 0, 0, 65, 85, 8, 3, 20, 0, 16, 25, 29, 0\n1, 0, 0]\nclass = e'),
Text(3905.999999999995, 181.19999999999982, 'gini = 0.82\nsamples = 100\nvalue = [3,
0, 0, 40, 37, 7, 2, 16, 0, 10, 19, 12, 0\n1, 0, 0]\nclass = d'),
Text(4065.428571428571, 181.19999999999982, 'gini = 0.716\nsamples = 63\nvalue = [0, 0,
0, 25, 48, 1, 1, 4, 0, 6, 6, 17, 0, 0\n0, 0]\nclass = e'),
Text(4304.571428571428, 543.5999999999999, 'row_id <= 34739.5\ngini = 0.856\nsamples =
557\nvalue = [5, 0, 0, 196, 189, 53, 29, 81, 0, 80, 99, 109\n2, 5, 44, 0]\nclass = d'),
Text(4224.857142857142, 181.19999999999982, 'gini = 0.852\nsamples = 353\nvalue = [0,
0, 0, 143, 109, 30, 29, 48, 0, 47, 67, 62\n2, 5, 24, 0]\nclass = d'),
Text(4384.285714285714, 181.19999999999982, 'gini = 0.853\nsamples = 204\nvalue = [5,
0, 0, 53, 80, 23, 0, 33, 0, 33, 32, 47, 0\n0, 20, 0]\nclass = e')]

```

