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

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

In [3]:
```
df=pd.read_csv("Data.csv").dropna()
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.dropna(inplace=True)
```

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

In [7]:
```
feature_matrix.shape
```

Out[7]:
```
(37518, 2)
```

In [8]:
```
target_vector.shape
```

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

In [9]:
```
from sklearn.preprocessing import StandardScaler
```

In [10]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [11]:
```
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[11]:
```
LogisticRegression()
```

In [12]:
```
feature_matrix.shape
```

Out[12]:
```
(37518, 2)
```

In [13]:
```
target_vector.shape
```

Out[13]:
```
(37518,)
```

In [14]:
```
from sklearn.preprocessing import StandardScaler
```

In [15]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [16]:
```
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[16]:
```
LogisticRegression()
```

In [17]:
```
observation=df[['row_id','user_id']]
```

In [18]:
```
prediction = logr.predict(observation)
prediction
```

Out[18]:
```
array([-1, -1, -1, ..., 16, 16, 16], dtype=int64)
```

In [19]:
```
logr.classes_
```

Out[19]:
```
array([-1,  0,  1,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16],
      dtype=int64)
```

In [20]:
```
logr.predict_proba(observation)[0][1]
```

Out[20]:
```
1.7263815682078809e-09
```

In [21]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [22]:
```
x = df[['row_id','user_id']]
y = df['gate_id']
```

In [23]:
```
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 [24]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
rr.score(x_train,y_train)
```

Out[24]:
```
0.005545384866390224
```

In [25]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

Out[25]:
```
LinearRegression()
```

In [26]:
```
lr.intercept_
```

Out[26]:
```
7.29500831419383
```

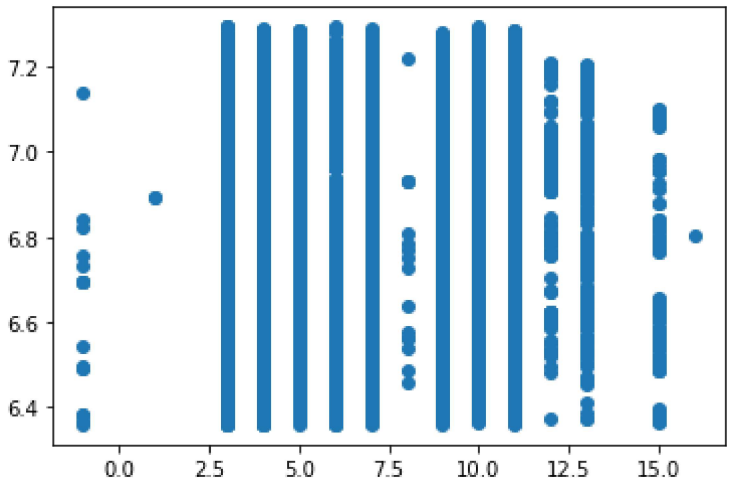
In [27]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[27]:

	Co-efficient
row_id	-0.000006
user_id	-0.012988

In [28]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[28]:
```
<matplotlib.collections.PathCollection at 0x1fa8049b9a0>
```



In [29]:
```
df['gate_id'].value_counts()
```

Out[29]:
```
 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 [38]:
```
x=df[['row_id','user_id']]
y=df['gate_id']
```

In [39]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

In [40]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[40]:
```
RandomForestClassifier()
```

In [41]:
```
parameters = {'max_depth':[1,2,3,4,5],'min_samples_leaf':[5,10,15,20,25],'n_estimators'
```

In [42]:
```
from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(estimator=rfc,param_grid= parameters,cv=2,scoring = "accurac
grid_search.fit(x_train,y_train)
```

Out[42]:
```
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 [43]:
```
grid_search.best_score_
```

Out[43]:
```
0.22355494631025818
```

In [44]:
```
rfc_best = grid_search.best_estimator_
```

In [46]:
```
from sklearn.tree import plot_tree
plt.figure(figsize = (80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names = ['Yes','No','Ye
```

Out[46]:
```
[Text(1915.2972972972973, 1993.2, 'row_id <= 16935.5\ngini = 0.871\nsamples = 16517\nval
ue = [39, 1, 1, 3724, 5730, 3363, 1235, 2178, 37, 2368\n3232, 2824, 501, 832, 197, 0]\nc
lass = Yes'),
 Text(633.4054054054054, 1630.8000000000002, 'row_id <= 96.5\ngini = 0.876\nsamples = 74
24\nvalue = [19, 1, 0, 1688, 2510, 1515, 640, 992, 8, 944, 1293\n1164, 377, 588, 54, 0]
\nclass = Yes'),
 Text(241.2972972972973, 1268.4, 'user_id <= 27.5\ngini = 0.799\nsamples = 38\nvalue =
[0, 0, 0, 5, 1, 15, 0, 9, 0, 20, 9, 3, 1, 0\n1, 0]\nclass = No'),
 Text(120.64864864864865, 906.0, 'gini = 0.744\nsamples = 23\nvalue = [0, 0, 0, 2, 1, 9,
0, 4, 0, 17, 4, 1, 1, 0\n1, 0]\nclass = No'),
 Text(361.94594594594594, 906.0, 'gini = 0.812\nsamples = 15\nvalue = [0, 0, 0, 3, 0, 6,
0, 5, 0, 3, 5, 2, 0, 0\n0, 0]\nclass = No'),
 Text(1025.5135135135135, 1268.4, 'user_id <= 52.5\ngini = 0.876\nsamples = 7386\nvalue
= [19, 1, 0, 1683, 2509, 1500, 640, 983, 8, 924, 1284\n1161, 376, 588, 53, 0]\nclass = Y
es'),
 Text(603.2432432432432, 906.0, 'user_id <= 19.5\ngini = 0.878\nsamples = 6252\nvalue =
[14, 1, 0, 1178, 2075, 1416, 472, 839, 8, 869, 1081\n994, 361, 555, 53, 0]\nclass = Ye
s'),
 Text(361.94594594594594, 543.5999999999999, 'user_id <= 16.5\ngini = 0.879\nsamples = 3
077\nvalue = [4, 0, 0, 495, 1028, 741, 182, 406, 1, 354, 512\n462, 298, 419, 0, 0]\nclas
s = Yes'),
```

```
 Text(241.2972972972973, 181.19999999999982, 'gini = 0.86\nsamples = 2232\nvalue = [4,
0, 0, 376, 828, 647, 152, 330, 1, 279, 411\n389, 47, 119, 0, 0]\nclass = Yes'),
 Text(482.5945945945946, 181.19999999999982, 'gini = 0.86\nsamples = 845\nvalue = [0, 0,
0, 119, 200, 94, 30, 76, 0, 75, 101, 73\n251, 300, 0, 0]\nclass = No'),
 Text(844.5405405405405, 543.5999999999999, 'row_id <= 6864.0\ngini = 0.873\nsamples = 3
175\nvalue = [10, 1, 0, 683, 1047, 675, 290, 433, 7, 515, 569\n532, 63, 136, 53, 0]\ncla
ss = Yes'),
 Text(723.8918918918919, 181.19999999999982, 'gini = 0.875\nsamples = 1255\nvalue = [2,
0, 0, 313, 410, 236, 142, 166, 0, 145, 205\n224, 47, 64, 28, 0]\nclass = Yes'),
 Text(965.1891891891892, 181.19999999999982, 'gini = 0.87\nsamples = 1920\nvalue = [8,
1, 0, 370, 637, 439, 148, 267, 7, 370, 364\n308, 16, 72, 25, 0]\nclass = Yes'),
 Text(1447.7837837837837, 906.0, 'row_id <= 16422.0\ngini = 0.826\nsamples = 1134\nvalue
= [5, 0, 0, 505, 434, 84, 168, 144, 0, 55, 203, 167\n15, 33, 0, 0]\nclass = No'),
 Text(1327.135135135135, 543.5999999999999, 'row_id <= 231.0\ngini = 0.827\nsamples = 11
05\nvalue = [5, 0, 0, 497, 406, 84, 164, 144, 0, 55, 197, 163\n15, 33, 0, 0]\nclass = N
o'),
 Text(1206.4864864864865, 181.19999999999982, 'gini = 0.544\nsamples = 16\nvalue = [0,
0, 0, 15, 2, 0, 1, 3, 0, 0, 1, 1, 0, 0\n0, 0]\nclass = No'),
 Text(1447.7837837837837, 181.19999999999982, 'gini = 0.829\nsamples = 1089\nvalue = [5,
0, 0, 482, 404, 84, 163, 141, 0, 55, 196, 162\n15, 33, 0, 0]\nclass = No'),
 Text(1568.4324324324325, 543.5999999999999, 'gini = 0.634\nsamples = 29\nvalue = [0, 0,
0, 8, 28, 0, 4, 0, 0, 0, 6, 4, 0, 0\n0, 0]\nclass = Yes'),
 Text(3197.189189189189, 1630.8000000000002, 'user_id <= 16.0\ngini = 0.865\nsamples = 9
093\nvalue = [20, 0, 1, 2036, 3220, 1848, 595, 1186, 29, 1424\n1939, 1660, 124, 244, 14
3, 0]\nclass = Yes'),
 Text(2533.6216216216217, 1268.4, 'user_id <= 11.5\ngini = 0.853\nsamples = 3173\nvalue
= [4, 0, 0, 384, 1161, 900, 149, 386, 4, 570, 726\n602, 26, 75, 26, 0]\nclass = Yes'),
 Text(2051.027027027027, 906.0, 'row_id <= 17565.5\ngini = 0.863\nsamples = 2038\nvalue
= [4, 0, 0, 322, 705, 497, 134, 253, 0, 306, 503\n361, 26, 74, 26, 0]\nclass = Yes'),
 Text(1809.7297297297296, 543.5999999999999, 'row_id <= 17428.5\ngini = 0.843\nsamples =
50\nvalue = [0, 0, 0, 16, 6, 1, 8, 9, 0, 17, 9, 7, 0, 1\n0, 0]\nclass = No'),
 Text(1689.081081081081, 181.19999999999982, 'gini = 0.835\nsamples = 35\nvalue = [0, 0,
0, 8, 4, 0, 6, 5, 0, 15, 8, 7, 0, 1\n0, 0]\nclass = No'),
 Text(1930.3783783783783, 181.19999999999982, 'gini = 0.765\nsamples = 15\nvalue = [0,
0, 0, 8, 2, 1, 2, 4, 0, 2, 1, 0, 0, 0\n0, 0]\nclass = No'),
 Text(2292.324324324324, 543.5999999999999, 'row_id <= 18138.0\ngini = 0.861\nsamples =
1988\nvalue = [4, 0, 0, 306, 699, 496, 126, 244, 0, 289, 494\n354, 26, 73, 26, 0]\nclass
= Yes'),
 Text(2171.675675675676, 181.19999999999982, 'gini = 0.766\nsamples = 58\nvalue = [0, 0,
0, 8, 37, 11, 3, 8, 0, 2, 7, 7, 2, 1\n0, 0]\nclass = Yes'),
 Text(2412.972972972973, 181.19999999999982, 'gini = 0.863\nsamples = 1930\nvalue = [4,
0, 0, 298, 662, 485, 123, 236, 0, 287, 487\n347, 24, 72, 26, 0]\nclass = Yes'),
 Text(3016.2162162162163, 906.0, 'user_id <= 13.0\ngini = 0.825\nsamples = 1135\nvalue =
[0, 0, 0, 62, 456, 403, 15, 133, 4, 264, 223, 241\n0, 1, 0, 0]\nclass = Yes'),
 Text(2774.9189189189187, 543.5999999999999, 'row_id <= 27965.5\ngini = 0.833\nsamples =
466\nvalue = [0, 0, 0, 52, 195, 143, 4, 52, 4, 96, 89, 110\n0, 1, 0, 0]\nclass = Yes'),
 Text(2654.27027027027, 181.19999999999982, 'gini = 0.83\nsamples = 273\nvalue = [0, 0,
0, 39, 116, 73, 2, 20, 0, 67, 60, 60, 0\n0, 0, 0]\nclass = Yes'),
 Text(2895.5675675675675, 181.19999999999982, 'gini = 0.827\nsamples = 193\nvalue = [0,
0, 0, 13, 79, 70, 2, 32, 4, 29, 29, 50, 0\n1, 0, 0]\nclass = Yes'),
 Text(3257.5135135135133, 543.5999999999999, 'row_id <= 29325.0\ngini = 0.815\nsamples =
669\nvalue = [0, 0, 0, 10, 261, 260, 11, 81, 0, 168, 134, 131\n0, 0, 0, 0]\nclass = Ye
s'),
 Text(3136.864864864865, 181.19999999999982, 'gini = 0.807\nsamples = 458\nvalue = [0,
0, 0, 1, 192, 155, 10, 41, 0, 154, 85, 85\n0, 0, 0, 0]\nclass = Yes'),
 Text(3378.162162162162, 181.19999999999982, 'gini = 0.8\nsamples = 211\nvalue = [0, 0,
0, 9, 69, 105, 1, 40, 0, 14, 49, 46, 0\n0, 0, 0]\nclass = No'),
 Text(3860.7567567567567, 1268.4, 'row_id <= 16983.0\ngini = 0.865\nsamples = 5920\nvalu
e = [16, 0, 1, 1652, 2059, 948, 446, 800, 25, 854, 1213\n1058, 98, 169, 117, 0]\nclass =
Yes'),
 Text(3740.108108108108, 906.0, 'gini = 0.365\nsamples = 18\nvalue = [0, 0, 0, 0, 25, 0,
0, 0, 0, 3, 0, 4, 0, 0\n0, 0]\nclass = Yes'),
 Text(3981.4054054054054, 906.0, 'row_id <= 25235.0\ngini = 0.865\nsamples = 5902\nvalue
= [16, 0, 1, 1652, 2034, 948, 446, 800, 25, 851, 1213\n1054, 98, 169, 117, 0]\nclass = Y
es'),
```

```
 Text(3740.108108108108, 543.5999999999999, 'row_id <= 17090.5\ngini = 0.871\nsamples =
2368\nvalue = [2, 0, 0, 676, 741, 369, 212, 327, 6, 306, 533\n432, 58, 110, 40, 0]\nclas
s = Yes'),
 Text(3619.459459459459, 181.19999999999982, 'gini = 0.825\nsamples = 38\nvalue = [0, 0,
0, 14, 3, 10, 1, 11, 0, 16, 14, 0, 0\n0, 2, 0]\nclass = No'),
 Text(3860.7567567567567, 181.19999999999982, 'gini = 0.87\nsamples = 2330\nvalue = [2,
0, 0, 662, 738, 359, 211, 316, 6, 290, 519\n432, 58, 110, 38, 0]\nclass = Yes'),
 Text(4222.7027027027025, 543.5999999999999, 'user_id <= 34.5\ngini = 0.86\nsamples = 35
34\nvalue = [14, 0, 1, 976, 1293, 579, 234, 473, 19, 545, 680\n622, 40, 59, 77, 0]\nclas
s = Yes'),
 Text(4102.054054054054, 181.19999999999982, 'gini = 0.858\nsamples = 1383\nvalue = [5,
0, 1, 481, 452, 170, 117, 211, 10, 173, 262\n255, 15, 18, 29, 0]\nclass = No'),
 Text(4343.351351351352, 181.19999999999982, 'gini = 0.858\nsamples = 2151\nvalue = [9,
0, 0, 495, 841, 409, 117, 262, 9, 372, 418\n367, 25, 41, 48, 0]\nclass = Yes')]
```