dicz4i1qy

August 2, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

[2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[3]:
```
df=pd.read_csv("/content/drive/MyDrive/mydatasets/C9_Data.csv")
df
```

[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 x 4 columns]
```

[4]:
```
df.head()
```

[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

1 Data Cleaning and Data Preprocessing

[5]:
```
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
```

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

[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

[7]:
```
df.columns
```

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

[8]:
```
feature_matrix = df[['row_id', 'user_id']]
target_vector = df[['gate_id']]
```

[9]:
```
fs = StandardScaler().fit_transform(feature_matrix)
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458:
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(
```

[9]:
```
LogisticRegression()
```

[10]:
```
observation=[[1,2]]
prediction = logr.predict(observation)
print(prediction)
```

```
[3]
```

[11]:
```
logr.classes_
```

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

[12]:
```
logr.predict_proba(observation)
```

[12]:
```
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]])
```

[13]:
```
x = df[['row_id', 'user_id']]
y = df['gate_id']
```

[14]:
```
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)
```

[15]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[15]:
```
LinearRegression()
```

[16]:
```
lr.intercept_
```

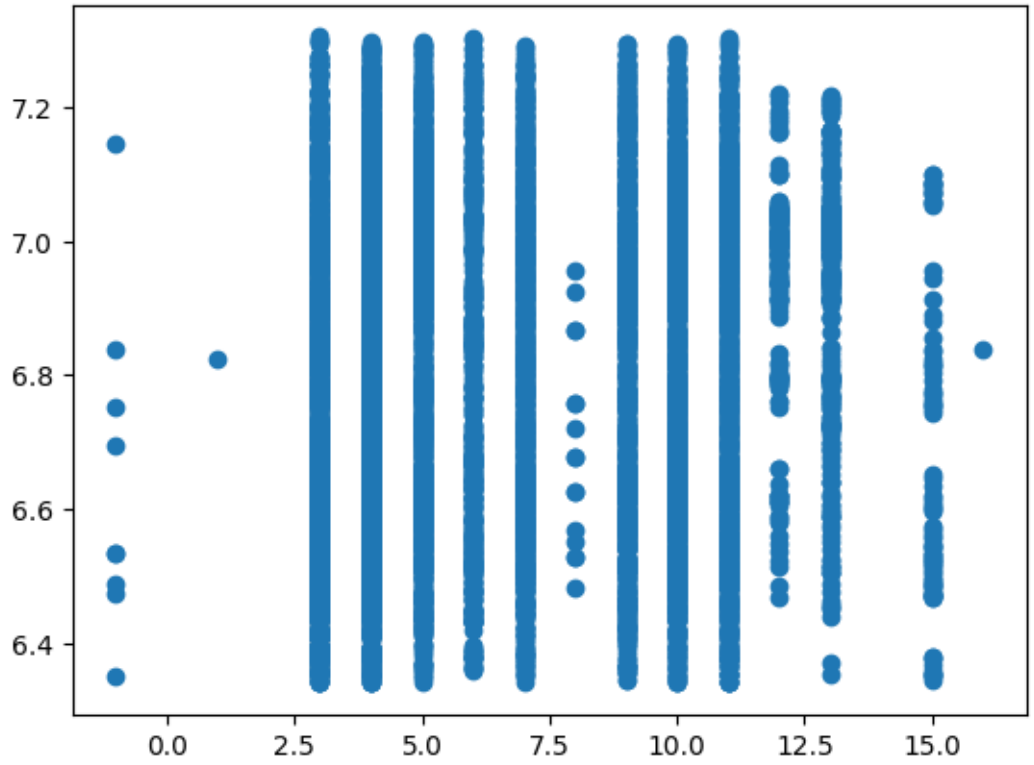
[16]:
```
7.305089401895409
```

[17]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

[17]:
```
          Co-efficient
row_id       -0.000006
user_id      -0.013217
```

[18]:
```
prediction =lr.predict(x_test)
plt.scatter(y_test,prediction)
```

[18]:
```
<matplotlib.collections.PathCollection at 0x7b60dec42740>
```



[19]:
```
lr.score(x_test,y_test)
```

[19]:
```
0.004886334761457278
```

[20]:
```
lr.score(x_train,y_train)
```

[20]:
```
0.005770412773131284
```

Random Forest

[31]:
```
df['gate_id'].value_counts()
```

[31]:
```
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
```

[32]:
```
x=df[['row_id', 'user_id']]
y=df[ 'gate_id']
```

[33]:
```
from sklearn.model_selection import train_test_split
```

[34]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

[35]:
```
from sklearn.ensemble import RandomForestClassifier
```

[36]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

[36]:
```
RandomForestClassifier()
```

[37]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf':[5,10,15,20,25],
            'n_estimators':[10,20,30,40,50]
}
```

[38]:
```
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)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_split.py:700:
UserWarning: The least populated class in y has only 1 members, which is less
than n_splits=2.
  warnings.warn(
```

[38]:
```
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')
```

[39]:
```
grid_search.best_score_
```

[39]:
```
0.2229076231817836
```

[40]:
```
rfc_best=grid_search.best_estimator_
```

[41]:
```
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','e','f','g','h','i','j','k','l','m','n','o','p','q'],f
```

[41]:
```
[Text(0.5145833333333333, 0.9166666666666666, 'user_id <= 49.5\ngini =
0.87\nsamples = 16506\nvalue = [49, 0, 1, 3693, 5796, 3334, 1240, 2055, 31,
2371\n3296, 2861, 479, 857, 1, 196, 2]\nclass = e'),
 Text(0.26666666666666666, 0.75, 'user_id <= 16.0\ngini = 0.872\nsamples =
13771\nvalue = [34, 0, 1, 2652, 4746, 3059, 951, 1715, 31, 2130\n2757, 2393,
461, 803, 1, 168, 2]\nclass = e'),
 Text(0.13333333333333333, 0.5833333333333334, 'user_id <= 13.0\ngini =
0.858\nsamples = 5474\nvalue = [6, 0, 0, 827, 1915, 1576, 308, 663, 7, 818,
1193\n1045, 60, 192, 1, 38, 0]\nclass = e'),
 Text(0.06666666666666667, 0.4166666666666667, 'row_id <= 25248.0\ngini =
0.865\nsamples = 4392\nvalue = [6, 0, 0, 804, 1491, 1068, 277, 518, 7, 616,
990\n830, 60, 192, 1, 38, 0]\nclass = e'),
 Text(0.03333333333333333, 0.25, 'row_id <= 19839.0\ngini = 0.863\nsamples =
2875\nvalue = [6, 0, 0, 532, 1007, 663, 185, 375, 3, 363, 636\n550, 47, 136, 0,
0, 0]\nclass = e'),
 Text(0.016666666666666666, 0.08333333333333333, 'gini = 0.868\nsamples =
2201\nvalue = [4, 0, 0, 416, 734, 532, 139, 295, 3, 303, 463\n426, 44, 123, 0,
0, 0]\nclass = e'),
 Text(0.05, 0.08333333333333333, 'gini = 0.844\nsamples = 674\nvalue = [2, 0, 0,
116, 273, 131, 46, 80, 0, 60, 173, 124\n3, 13, 0, 0, 0]\nclass = e'),
 Text(0.1, 0.25, 'row_id <= 25353.0\ngini = 0.865\nsamples = 1517\nvalue = [0,
0, 0, 272, 484, 405, 92, 143, 4, 253, 354\n280, 13, 56, 1, 38, 0]\nclass = e'),
 Text(0.08333333333333333, 0.08333333333333333, 'gini = 0.64\nsamples =
14\nvalue = [0, 0, 0, 0, 0, 8, 0, 3, 0, 10, 1, 0, 0, 0\n0, 0, 0]\nclass = j'),
 Text(0.11666666666666667, 0.08333333333333333, 'gini = 0.865\nsamples =
1503\nvalue = [0, 0, 0, 272, 484, 397, 92, 140, 4, 243, 353\n280, 13, 56, 1, 38,
0]\nclass = e'),
 Text(0.2, 0.4166666666666667, 'row_id <= 15355.5\ngini = 0.808\nsamples =
1082\nvalue = [0, 0, 0, 23, 424, 508, 31, 145, 0, 202, 203, 215\n0, 0, 0, 0,
```

0]\nclass = f'),
 Text(0.16666666666666666, 0.25, 'row_id <= 4727.0\ngini = 0.765\nsamples =
346\nvalue = [0, 0, 0, 3, 159, 205, 20, 41, 0, 19, 59, 72, 0\n0, 0, 0, 0]\nclass
= f'),
 Text(0.15, 0.08333333333333333, 'gini = 0.788\nsamples = 107\nvalue = [0, 0, 0,
0, 37, 68, 12, 7, 0, 14, 34, 23, 0\n0, 0, 0, 0]\nclass = f'),
 Text(0.18333333333333332, 0.08333333333333333, 'gini = 0.741\nsamples =
239\nvalue = [0, 0, 0, 3, 122, 137, 8, 34, 0, 5, 25, 49, 0\n0, 0, 0, 0]\nclass =
f'),
 Text(0.23333333333333334, 0.25, 'row_id <= 29384.5\ngini = 0.82\nsamples =
736\nvalue = [0, 0, 0, 20, 265, 303, 11, 104, 0, 183, 144, 143\n0, 0, 0, 0,
0]\nclass = f'),
 Text(0.21666666666666667, 0.08333333333333333, 'gini = 0.819\nsamples =
499\nvalue = [0, 0, 0, 12, 171, 201, 8, 76, 0, 154, 87, 87\n0, 0, 0, 0,
0]\nclass = f'),
 Text(0.25, 0.08333333333333333, 'gini = 0.808\nsamples = 237\nvalue = [0, 0, 0,
8, 94, 102, 3, 28, 0, 29, 57, 56, 0\n0, 0, 0, 0]\nclass = f'),
 Text(0.4, 0.5833333333333334, 'user_id <= 19.5\ngini = 0.877\nsamples =
8297\nvalue = [28, 0, 1, 1825, 2831, 1483, 643, 1052, 24, 1312\n1564, 1348, 401,
611, 0, 130, 2]\nclass = e'),
 Text(0.3333333333333333, 0.4166666666666667, 'user_id <= 17.5\ngini =
0.882\nsamples = 1772\nvalue = [0, 0, 0, 394, 515, 166, 103, 231, 2, 172,
306\n247, 296, 398, 0, 0, 0]\nclass = e'),
 Text(0.3, 0.25, 'row_id <= 30141.0\ngini = 0.857\nsamples = 293\nvalue = [0, 0,
0, 90, 98, 34, 46, 47, 1, 33, 66, 55, 0\n0, 0, 0, 0]\nclass = e'),
 Text(0.2833333333333333, 0.08333333333333333, 'gini = 0.848\nsamples =
140\nvalue = [0, 0, 0, 33, 61, 24, 25, 30, 0, 24, 21, 13, 0\n0, 0, 0, 0]\nclass
= e'),
 Text(0.31666666666666665, 0.08333333333333333, 'gini = 0.837\nsamples =
153\nvalue = [0, 0, 0, 57, 37, 10, 21, 17, 1, 9, 45, 42, 0\n0, 0, 0, 0]\nclass =
d'),
 Text(0.36666666666666664, 0.25, 'user_id <= 18.5\ngini = 0.878\nsamples =
1479\nvalue = [0, 0, 0, 304, 417, 132, 57, 184, 1, 139, 240\n192, 296, 398, 0,
0, 0]\nclass = e'),
 Text(0.35, 0.08333333333333333, 'gini = 0.873\nsamples = 714\nvalue = [0, 0, 0,
63, 182, 119, 21, 72, 1, 113, 126, 70\n149, 237, 0, 0, 0]\nclass = n'),
 Text(0.38333333333333336, 0.08333333333333333, 'gini = 0.86\nsamples =
765\nvalue = [0, 0, 0, 241, 235, 13, 36, 112, 0, 26, 114, 122\n147, 161, 0, 0,
0]\nclass = d'),
 Text(0.4666666666666667, 0.4166666666666667, 'user_id <= 48.5\ngini =
0.869\nsamples = 6525\nvalue = [28, 0, 1, 1431, 2316, 1317, 540, 821, 22,
1140\n1258, 1101, 105, 213, 0, 130, 2]\nclass = e'),
 Text(0.43333333333333335, 0.25, 'user_id <= 35.5\ngini = 0.868\nsamples =
5977\nvalue = [28, 0, 1, 1410, 2112, 1154, 519, 792, 17, 959\n1168, 1032, 105,
213, 0, 52, 2]\nclass = e'),
 Text(0.4166666666666667, 0.08333333333333333, 'gini = 0.86\nsamples =
2913\nvalue = [6, 0, 1, 817, 1048, 452, 281, 380, 10, 566, 519\n498, 12, 15, 0,

52, 2]\nclass = e'),
 Text(0.45, 0.08333333333333333, 'gini = 0.871\nsamples = 3064\nvalue = [22, 0,
0, 593, 1064, 702, 238, 412, 7, 393, 649\n534, 93, 198, 0, 0, 0]\nclass = e'),
 Text(0.5, 0.25, 'row_id <= 32338.0\ngini = 0.836\nsamples = 548\nvalue = [0, 0,
0, 21, 204, 163, 21, 29, 5, 181, 90, 69\n0, 0, 0, 78, 0]\nclass = e'),
 Text(0.48333333333333334, 0.08333333333333333, 'gini = 0.837\nsamples =
497\nvalue = [0, 0, 0, 19, 168, 158, 21, 19, 2, 159, 75, 67\n0, 0, 0, 78,
0]\nclass = e'),
 Text(0.5166666666666667, 0.08333333333333333, 'gini = 0.762\nsamples =
51\nvalue = [0, 0, 0, 2, 36, 5, 0, 10, 3, 22, 15, 2, 0, 0\n0, 0, 0]\nclass =
e'),
 Text(0.7625, 0.75, 'row_id <= 18667.0\ngini = 0.84\nsamples = 2735\nvalue =
[15, 0, 0, 1041, 1050, 275, 289, 340, 0, 241, 539\n468, 18, 54, 0, 28, 0]\nclass
= e'),
 Text(0.6583333333333333, 0.5833333333333334, 'user_id <= 55.5\ngini =
0.831\nsamples = 1415\nvalue = [7, 0, 0, 603, 556, 141, 179, 189, 0, 90,
253\n241, 17, 23, 0, 0, 0]\nclass = d'),
 Text(0.6, 0.4166666666666667, 'user_id <= 54.5\ngini = 0.832\nsamples =
1209\nvalue = [1, 0, 0, 526, 459, 134, 177, 158, 0, 77, 196\n184, 17, 23, 0, 0,
0]\nclass = d'),
 Text(0.5666666666666667, 0.25, 'row_id <= 128.0\ngini = 0.819\nsamples =
798\nvalue = [0, 0, 0, 344, 361, 83, 84, 102, 0, 50, 113, 123\n17, 21, 0, 0,
0]\nclass = e'),
 Text(0.55, 0.08333333333333333, 'gini = 0.397\nsamples = 6\nvalue = [0, 0, 0,
8, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0\n0, 0, 0]\nclass = d'),
 Text(0.5833333333333334, 0.08333333333333333, 'gini = 0.82\nsamples =
792\nvalue = [0, 0, 0, 336, 361, 83, 84, 102, 0, 50, 110, 123\n17, 21, 0, 0,
0]\nclass = e'),
 Text(0.6333333333333333, 0.25, 'row_id <= 307.5\ngini = 0.84\nsamples =
411\nvalue = [1, 0, 0, 182, 98, 51, 93, 56, 0, 27, 83, 61, 0\n2, 0, 0, 0]\nclass
= d'),
 Text(0.6166666666666667, 0.08333333333333333, 'gini = 0.449\nsamples = 7\nvalue
= [0, 0, 0, 10, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0\n0, 0, 0]\nclass = d'),
 Text(0.65, 0.08333333333333333, 'gini = 0.843\nsamples = 404\nvalue = [1, 0, 0,
172, 96, 51, 93, 54, 0, 27, 83, 61, 0\n2, 0, 0, 0]\nclass = d'),
 Text(0.7166666666666667, 0.4166666666666667, 'row_id <= 16549.5\ngini =
0.809\nsamples = 206\nvalue = [6, 0, 0, 77, 97, 7, 2, 31, 0, 13, 57, 57, 0\n0,
0, 0, 0]\nclass = e'),
 Text(0.7, 0.25, 'row_id <= 4992.5\ngini = 0.81\nsamples = 181\nvalue = [6, 0,
0, 57, 91, 7, 2, 29, 0, 13, 54, 45, 0\n0, 0, 0, 0]\nclass = e'),
 Text(0.6833333333333333, 0.08333333333333333, 'gini = 0.714\nsamples =
21\nvalue = [0, 0, 0, 5, 8, 0, 1, 0, 0, 0, 5, 14, 0, 0\n0, 0, 0]\nclass = l'),
 Text(0.7166666666666667, 0.08333333333333333, 'gini = 0.809\nsamples =
160\nvalue = [6, 0, 0, 52, 83, 7, 1, 29, 0, 13, 49, 31, 0\n0, 0, 0, 0]\nclass =
e'),
 Text(0.7333333333333333, 0.25, 'gini = 0.679\nsamples = 25\nvalue = [0, 0, 0,
20, 6, 0, 0, 2, 0, 0, 3, 12, 0, 0\n0, 0, 0]\nclass = d'),

```
 Text(0.8666666666666667, 0.5833333333333334, 'user_id <= 54.5\ngini =
0.847\nsamples = 1320\nvalue = [8, 0, 0, 438, 494, 134, 110, 151, 0, 151,
286\n227, 1, 31, 0, 28, 0]\nclass = e'),
 Text(0.8, 0.4166666666666667, 'row_id <= 34838.0\ngini = 0.838\nsamples =
689\nvalue = [0, 0, 0, 198, 290, 80, 26, 90, 0, 124, 138, 126\n0, 8, 0, 0,
0]\nclass = e'),
 Text(0.7666666666666667, 0.25, 'row_id <= 31928.5\ngini = 0.84\nsamples =
602\nvalue = [0, 0, 0, 186, 242, 72, 26, 78, 0, 91, 126, 114\n0, 8, 0, 0,
0]\nclass = e'),
 Text(0.75, 0.08333333333333333, 'gini = 0.84\nsamples = 550\nvalue = [0, 0, 0,
154, 229, 72, 23, 70, 0, 85, 115, 99\n0, 8, 0, 0, 0]\nclass = e'),
 Text(0.7833333333333333, 0.08333333333333333, 'gini = 0.787\nsamples =
52\nvalue = [0, 0, 0, 32, 13, 0, 3, 8, 0, 6, 11, 15, 0, 0\n0, 0, 0]\nclass =
d'),
 Text(0.8333333333333334, 0.25, 'user_id <= 51.5\ngini = 0.785\nsamples =
87\nvalue = [0, 0, 0, 12, 48, 8, 0, 12, 0, 33, 12, 12, 0\n0, 0, 0, 0]\nclass =
e'),
 Text(0.8166666666666667, 0.08333333333333333, 'gini = 0.749\nsamples =
49\nvalue = [0, 0, 0, 10, 33, 2, 0, 8, 0, 3, 10, 11, 0, 0\n0, 0, 0]\nclass =
e'),
 Text(0.85, 0.08333333333333333, 'gini = 0.671\nsamples = 38\nvalue = [0, 0, 0,
2, 15, 6, 0, 4, 0, 30, 2, 1, 0, 0\n0, 0, 0]\nclass = j'),
 Text(0.9333333333333333, 0.4166666666666667, 'row_id <= 32691.5\ngini =
0.846\nsamples = 631\nvalue = [8, 0, 0, 240, 204, 54, 84, 61, 0, 27, 148,
101\n1, 23, 0, 28, 0]\nclass = d'),
 Text(0.9, 0.25, 'row_id <= 21761.5\ngini = 0.836\nsamples = 445\nvalue = [3, 0,
0, 168, 166, 30, 64, 41, 0, 23, 100, 69\n1, 22, 0, 3, 0]\nclass = d'),
 Text(0.8833333333333333, 0.08333333333333333, 'gini = 0.86\nsamples =
124\nvalue = [0, 0, 0, 44, 29, 12, 17, 13, 0, 13, 36, 16, 0\n15, 0, 0, 0]\nclass
= d'),
 Text(0.9166666666666666, 0.08333333333333333, 'gini = 0.818\nsamples =
321\nvalue = [3, 0, 0, 124, 137, 18, 47, 28, 0, 10, 64, 53\n1, 7, 0, 3,
0]\nclass = e'),
 Text(0.9666666666666667, 0.25, 'row_id <= 33559.5\ngini = 0.856\nsamples =
186\nvalue = [5, 0, 0, 72, 38, 24, 20, 20, 0, 4, 48, 32, 0\n1, 0, 25, 0]\nclass
= d'),
 Text(0.95, 0.08333333333333333, 'gini = 0.844\nsamples = 25\nvalue = [0, 0, 0,
4, 4, 9, 4, 5, 0, 0, 10, 2, 0, 1\n0, 2, 0]\nclass = k'),
 Text(0.9833333333333333, 0.08333333333333333, 'gini = 0.847\nsamples =
161\nvalue = [5, 0, 0, 68, 34, 15, 16, 15, 0, 4, 38, 30, 0\n0, 0, 23, 0]\nclass
= d')]
```