

```
In [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
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.tree import plot_tree
```

```
In [2]: df=pd.read_csv("C9_Data - C9_Data.csv")
df
```

Out[2]:

	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 [3]: df1=df.fillna(value=0)
df1
```

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]: df1.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 [5]: df1.columns
```

Out[5]: Index(['row_id', 'user_id', 'timestamp', 'gate_id'], dtype='object')

```
In [19]: df2=df1[['row_id', 'user_id', 'gate_id']]
df2
```

Out[19]:

	row_id	user_id	gate_id
0	0	18	7
1	1	18	9
2	2	18	9
3	3	18	5
4	4	18	5
...
37513	37513	6	11
37514	37514	6	6
37515	37515	6	6
37516	37516	6	9
37517	37517	6	9

37518 rows × 3 columns

```
In [20]: df2['user_id'].value_counts()
```

```
Out[20]: 37    2262
          55    2238
          6     2013
          12    1953
          19    1793
          15    1756
          18    1578
          47    1341
          53    1311
          1     1299
          33    1285
          11    1281
          49    1275
          0     1250
          39    1144
          32    1076
          54    1070
          9     1034
          50     994
          29     990
          3     989
          48     743
          14     696
          17     677
          27     603
          35     601
          46     502
          57     497
          24     416
          42     359
          26     316
          34     284
          23     261
          25     247
          40     242
          31     191
          56     137
          43     124
          41     124
          20     115
          22      96
          28      64
          45      57
          7       49
          36      48
          2       39
          8       29
          10      17
          38      13
          30      10
          5       10
          21       5
          52       5
          44       4
          51       3
```

```
4      2
Name: user_id, dtype: int64
```

```
In [21]: x=df2.drop('user_id',axis=1)
        y=df2['user_id']
```

```
In [22]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.70)
```

```
In [23]: rfc=RandomForestClassifier()
        rfc.fit(x_train,y_train)
```

```
Out[23]: RandomForestClassifier()
```

```
In [24]: parameters = {'max_depth':[1,2,3,4,5],
                        'min_samples_leaf':[5,10,15,20,25],
                        'n_estimators':[10,20,30,40,50]}
```

```
In [25]: grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='acc
        grid_search.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:
666: UserWarning: The least populated class in y has only 1 members, which is
less than n_splits=2.
      warnings.warn("The least populated class in y has only %d"
```

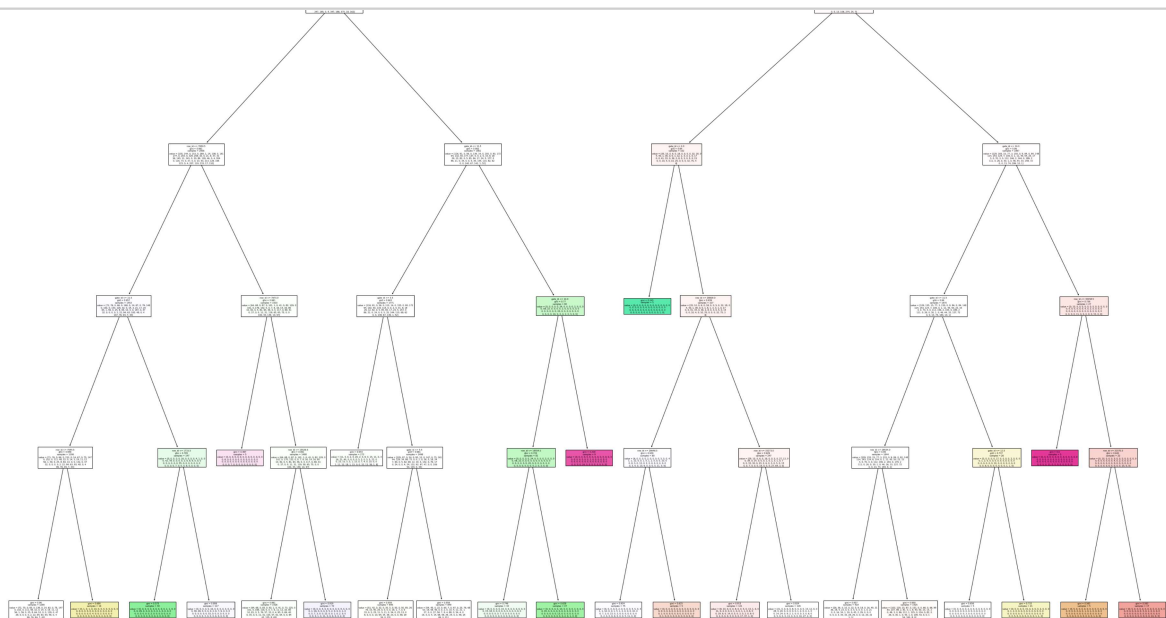
```
Out[25]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
                  param_grid={'max_depth': [1, 2, 3, 4, 5],
                              'min_samples_leaf': [5, 10, 15, 20, 25],
                              'n_estimators': [10, 20, 30, 40, 50]},
                  scoring='accuracy')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

```
In [26]: grid_search.best_score_
```

```
Out[26]: 0.10093274582683323
```

```
In [27]: rfc_best =grid_search.best_estimator_
```

```
In [28]: plt.figure(figsize=(50,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,filled=True)
```



In []:

In []: