

■
GROUP ASSIGNMENT:
ANALYTICS ON COMPLEX DATA

PREDICTING CREDIT CARD CHURNING CUSTOMERS

PRESENTED BY
ชญาบันก์ มนະຄິຈາບນກ 6510503298
ศຸກກີຕຕໍ່ ວົງສົ່ຕ 6510503816



DATASET

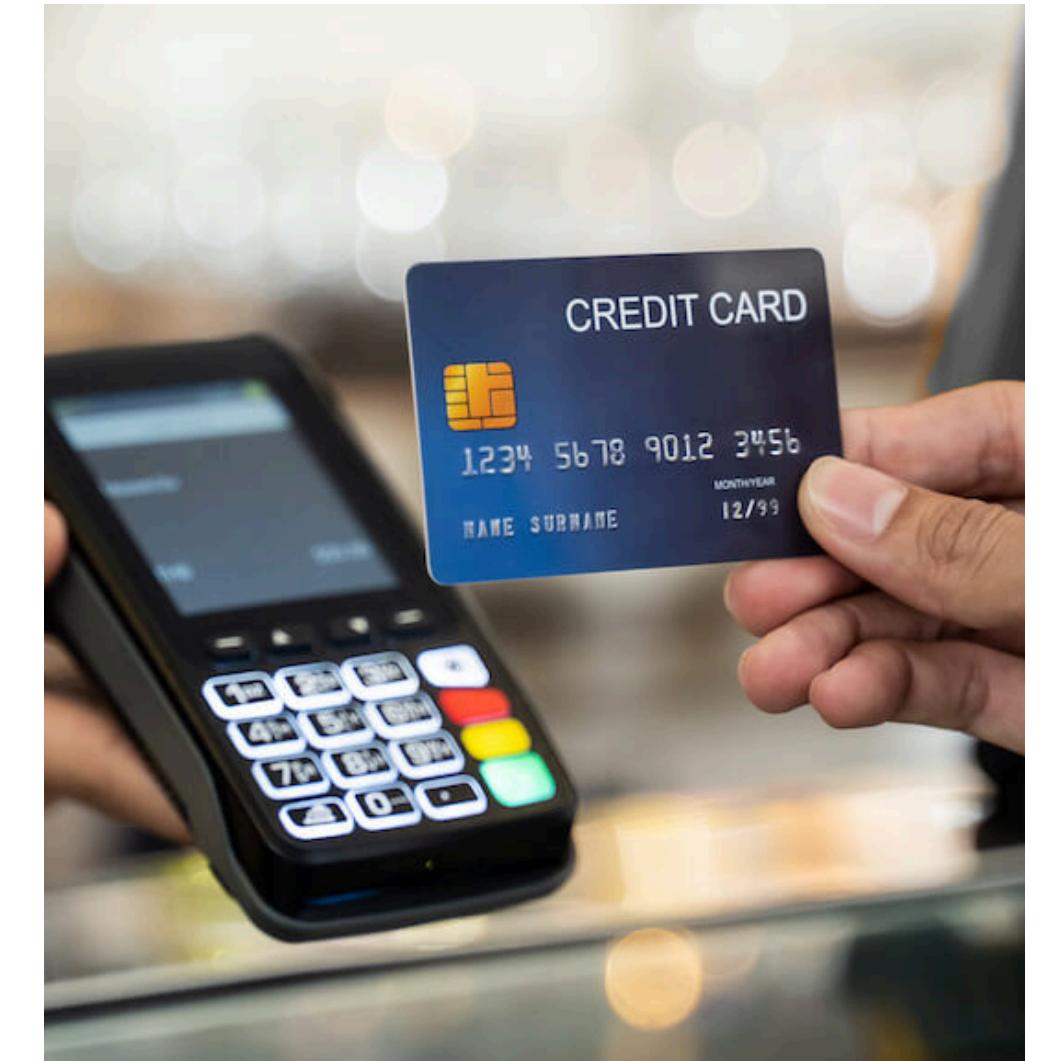


BankChurners.csv มีตัวแปรต่างๆ ซึ่งเกี่ยวข้องกับข้อมูลของผู้ใช้งานบัตรเครดิต ก็ทั้งที่ยังใช้อยู่ และเลิกใช้ไปมีข้อมูลต่างๆดังนี้

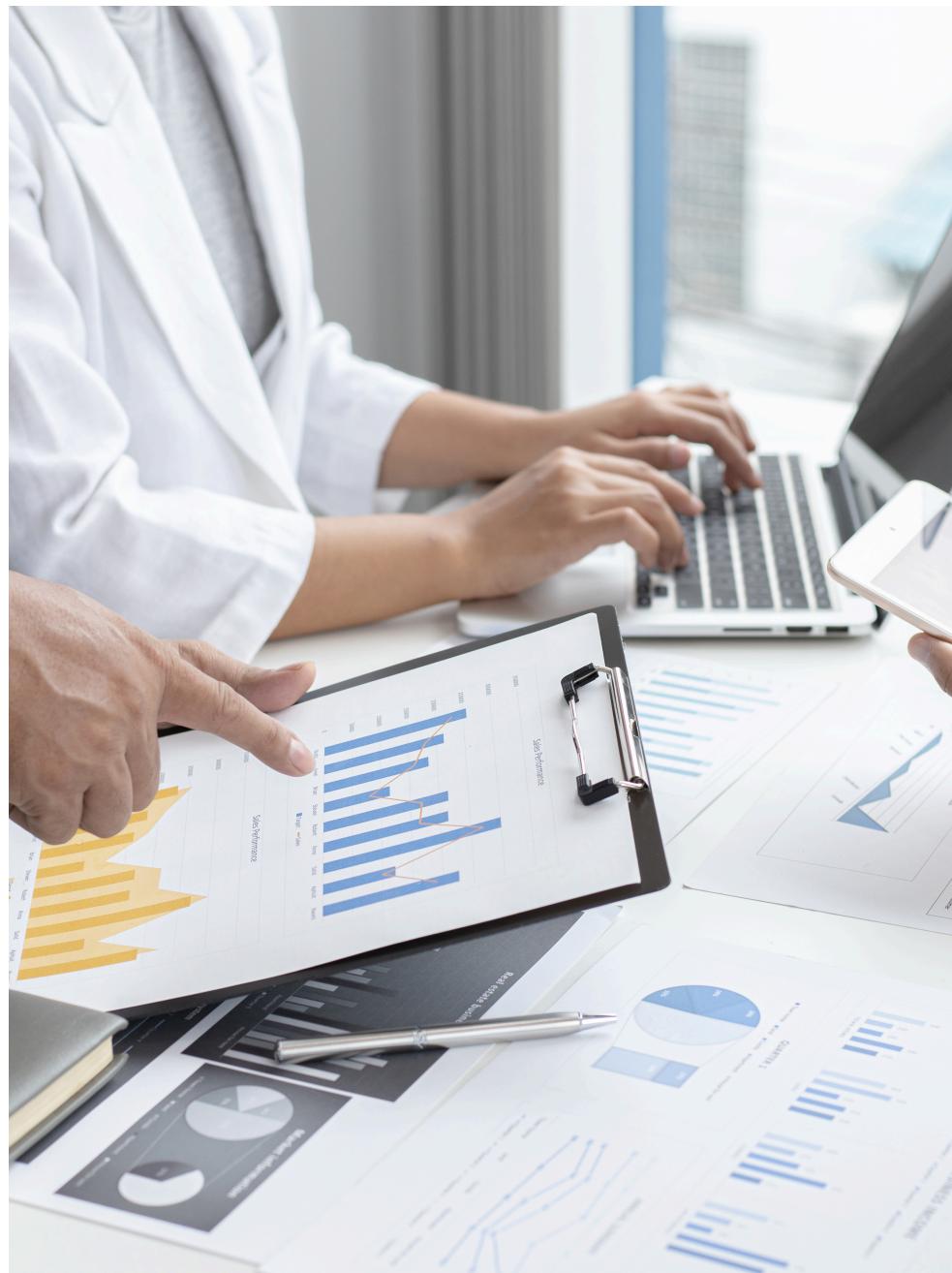
- Attrition_Flag คือ customer activity ว่ายังเปิดบัตรเครดิตกับธนาคารอยู่ หรือไม่
- Customer_Age คือ อายุของลูกค้า
- Gender คือ เพศของลูกค้า
- Dependent_count คือ จำนวน dependent หรือบุคคลที่ໄວ่ใจได้
- Education_Level คือ ระดับการศึกษาของลูกค้า
- Marital_Status คือสถานะของลูกค้า
- Income_Category คือรายรับของลูกค้า จะเป็นช่วงรายรับ เช่น \$60K - \$80K
- Card_Category คือระดับของบัตรเครดิต เช่น Gold Silver
- Months_on_book คือระยะเวลาที่ลูกค้าเปิดบัญชีกับธนาคาร

DATASET

- Total_Relationship_Count คือจำนวน product ที่ลูกค้ามี
- Months_Inactive_12_mon คือจำนวนเดือนที่ inactive ใน 12 เดือนล่าสุด
- Contacts_Count_12_mon คือจำนวน contact ใน 12 เดือนล่าสุด
- Credit_Limit คือ credit limit ของบัตรเครดิตลูกค้า
- Total_Revolving_Bal คือจำนวนเงิน revolving balance ของลูกค้า
- Avg_Open_To_Buy คือจำนวนวงเงินที่ลูกค้าเหลือโดยเฉลี่ย 12 เดือน
- Total_Amt_Chng_Q4_Q1 คืออัตราส่วนการใช้จ่ายระหว่าง Q4/Q1
- Total_Trans_Amt คือจำนวนเงินที่ทำ transaction ในรอบ 12 เดือนที่ผ่านมา
- Total_Trans_Ct คือ จำนวนครั้งที่ทำ transaction ในรอบ 12 เดือนที่ผ่านมา
- Total_Ct_Chng_Q4_Q1 คืออัตราส่วน transaction ระหว่าง Q4/Q1
- Avg_Utilization_Ratio คือเปอร์เซ็นต์การใช้งานบัตรเครดิตต่อวงเงินทั้งหมด



JUSTIFY HOW COMPLEX IS YOUR DATA?

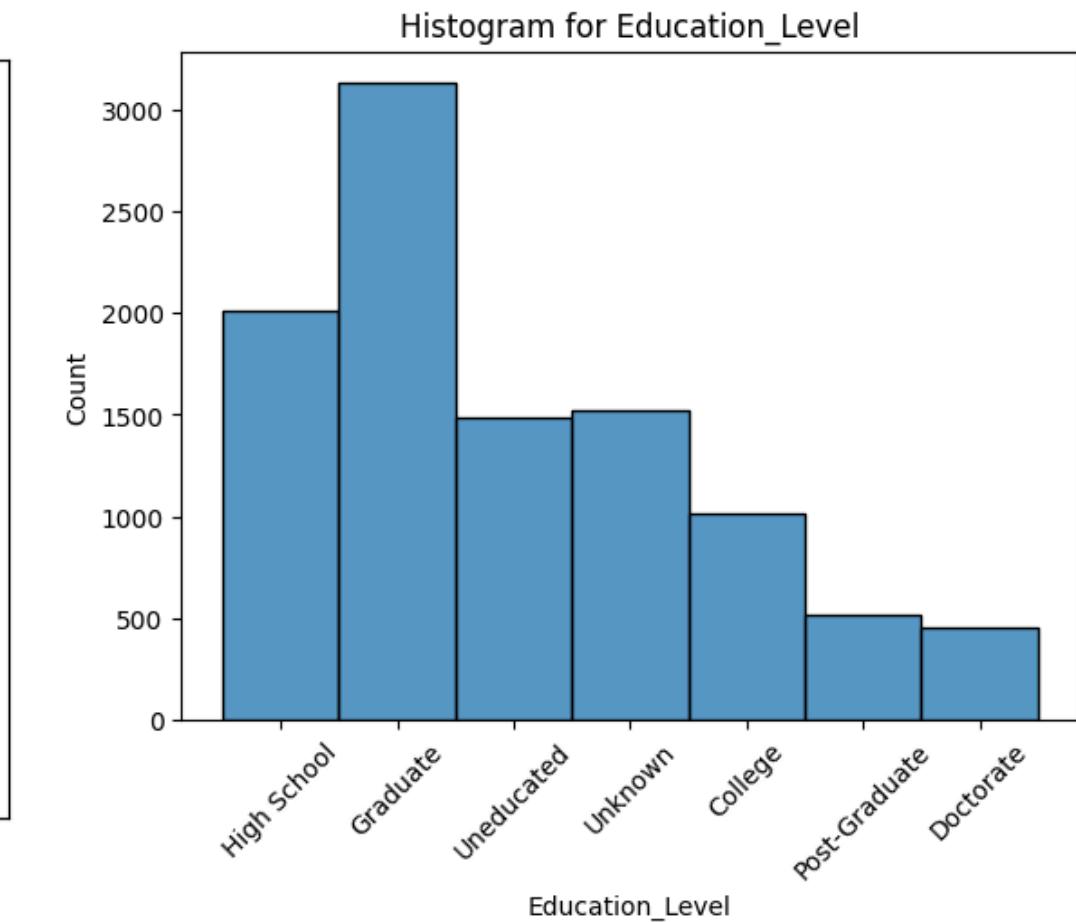
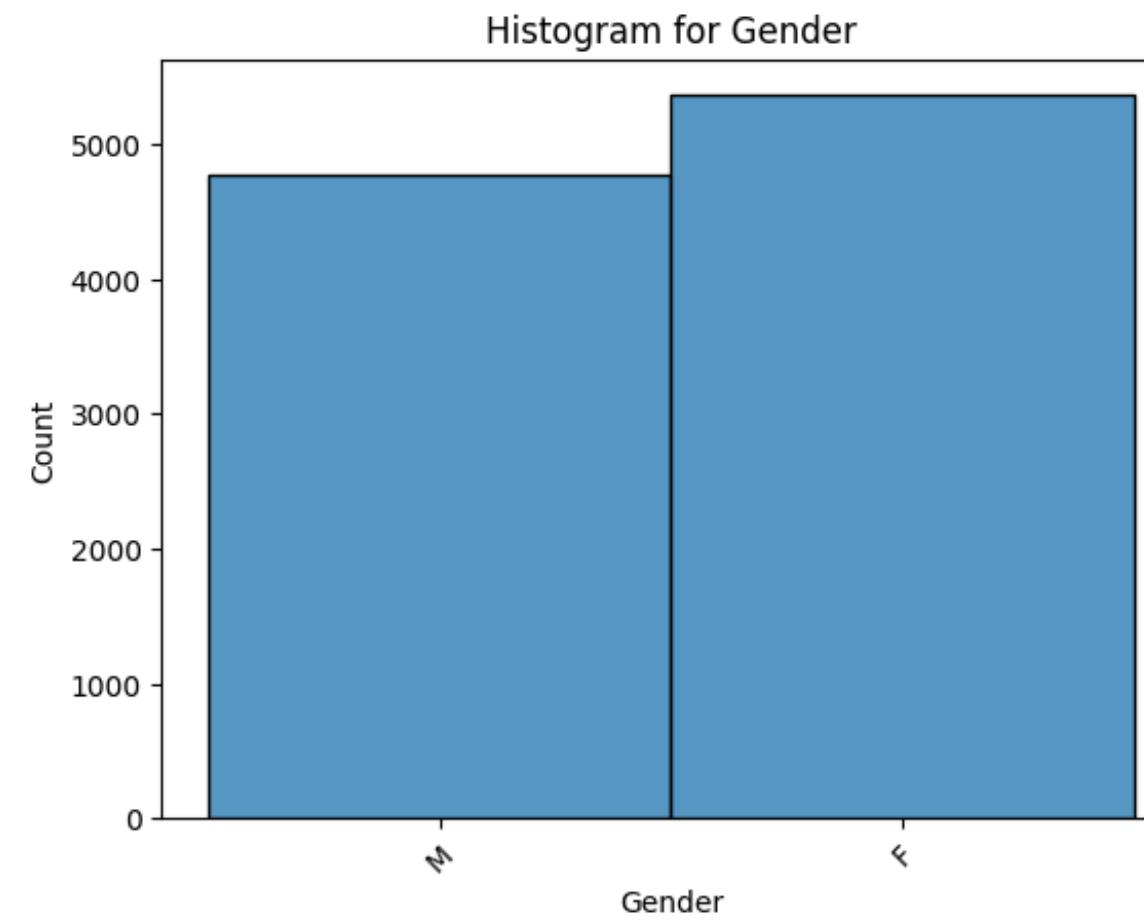
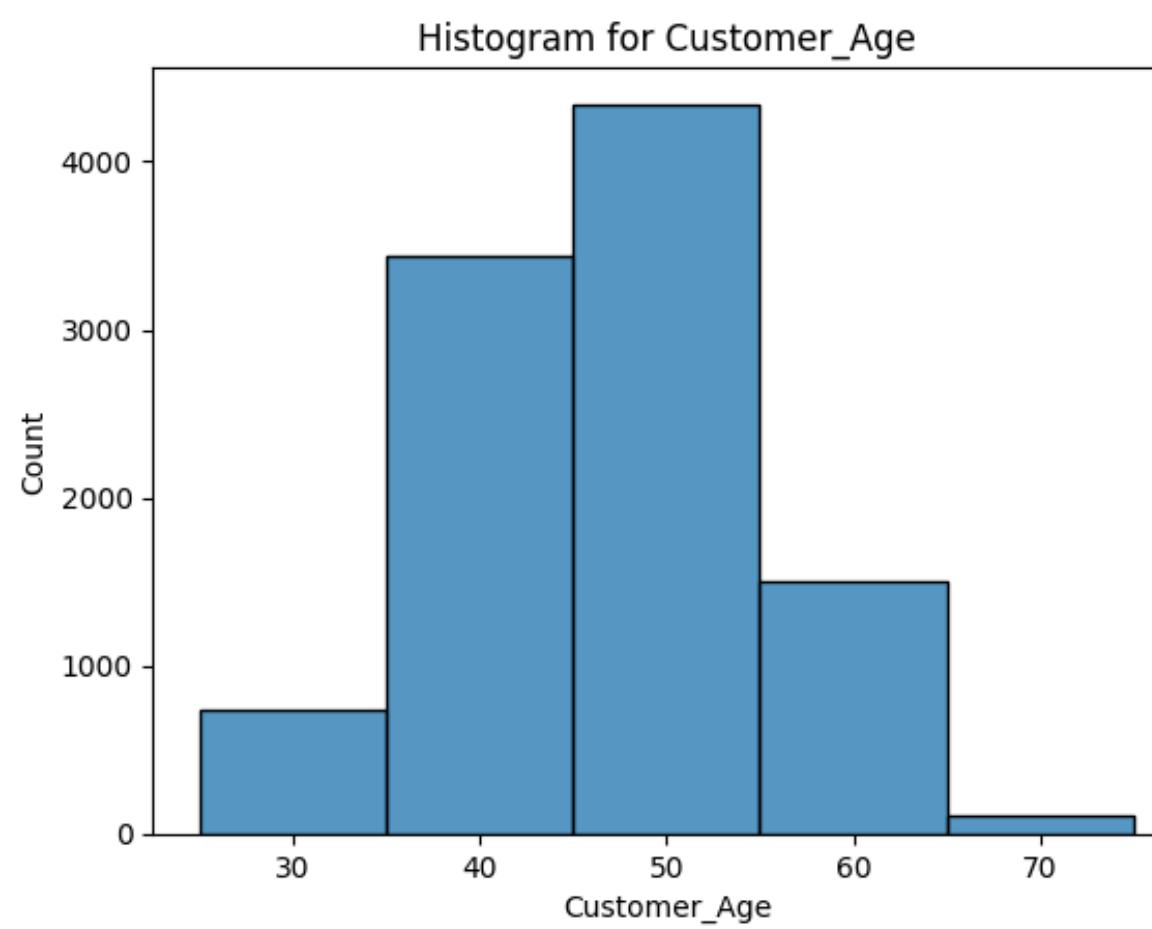


- BankChurners.csv มีตัวแปรต่างๆมากมายที่มีส่วนเกี่ยวข้องในการนำมาทำเป็น prediction model ดังนั้น เราจึงต้องจัดการ feature ให้พร้อมกับการทำ BalancedRandomForestClassifier
- ใน column Attrition_Flag ที่เป็น อัตราการ Churn ของลูกค้า มี churn rate อยู่เพียง 16% ซึ่งทำให้เป็น imbalanced data ที่ต้องจัดการก่อนนำมาทำ prediction model เพื่อเพิ่มความแม่นยำในการทำนาย

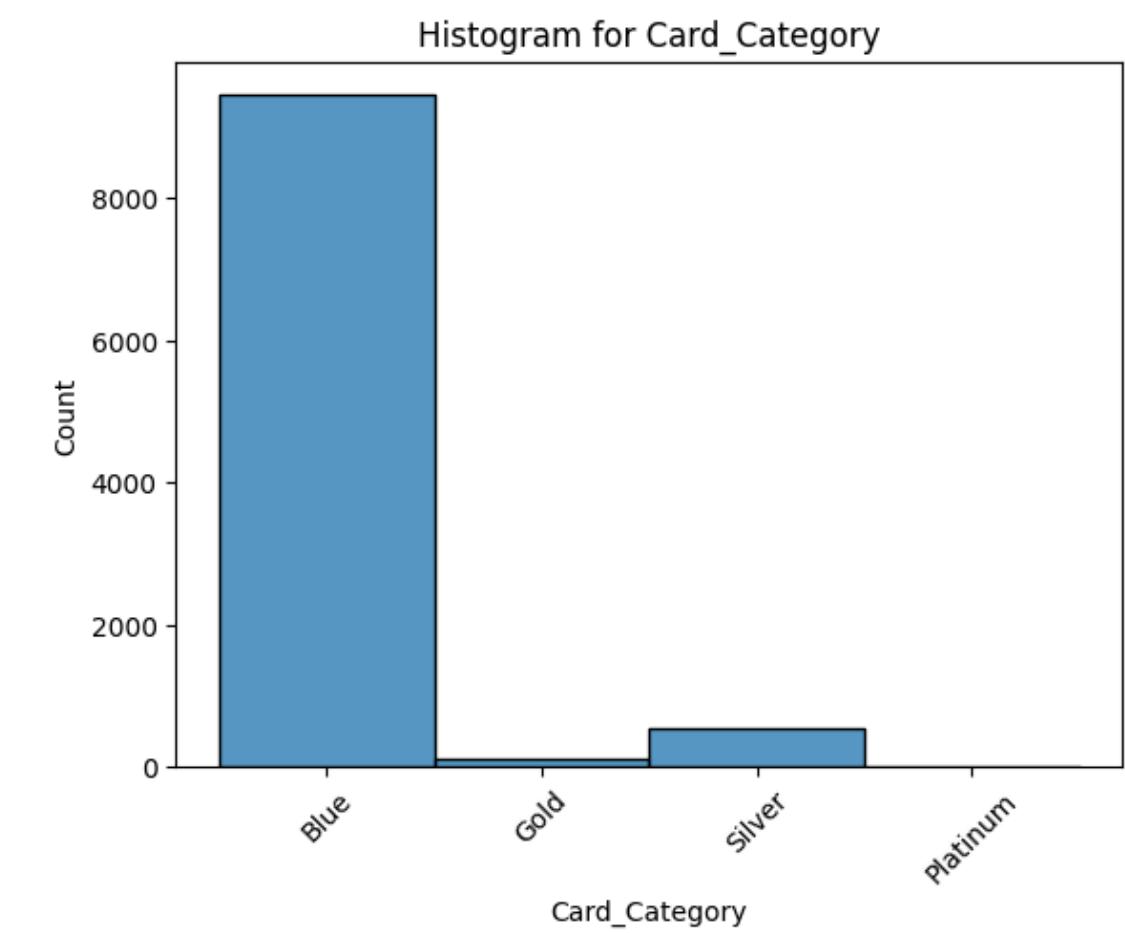
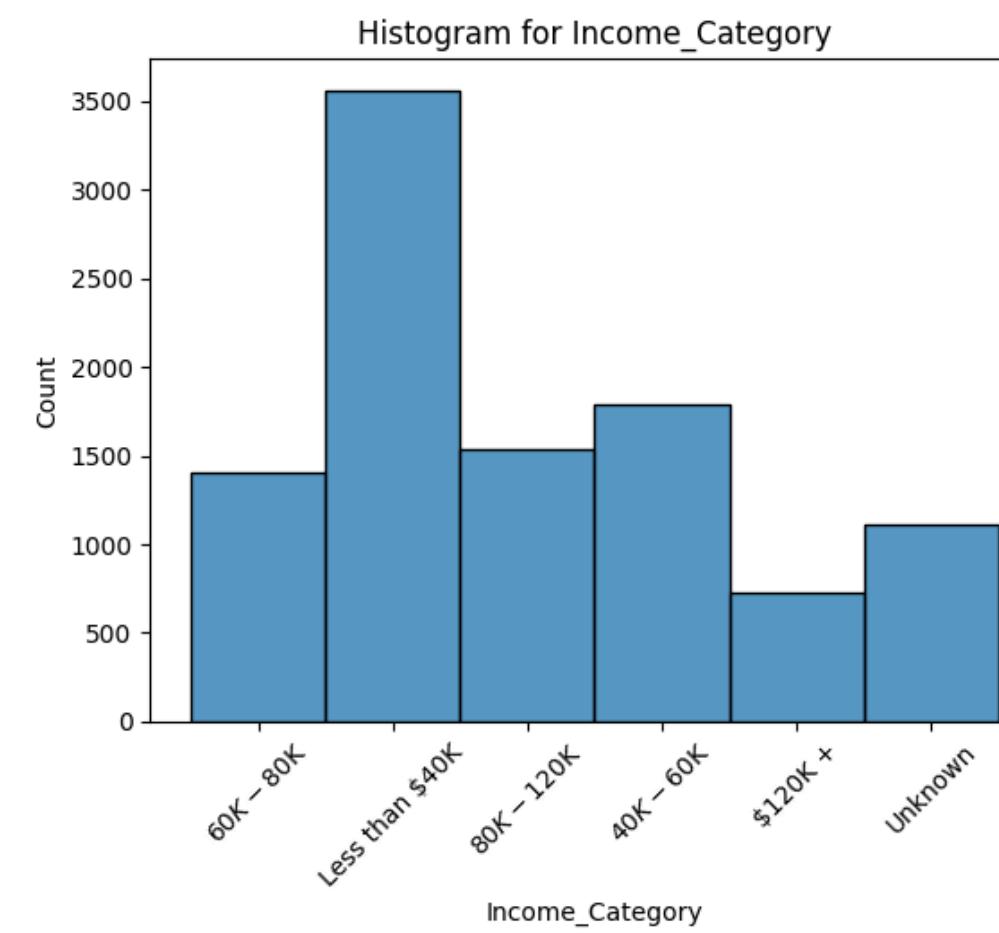
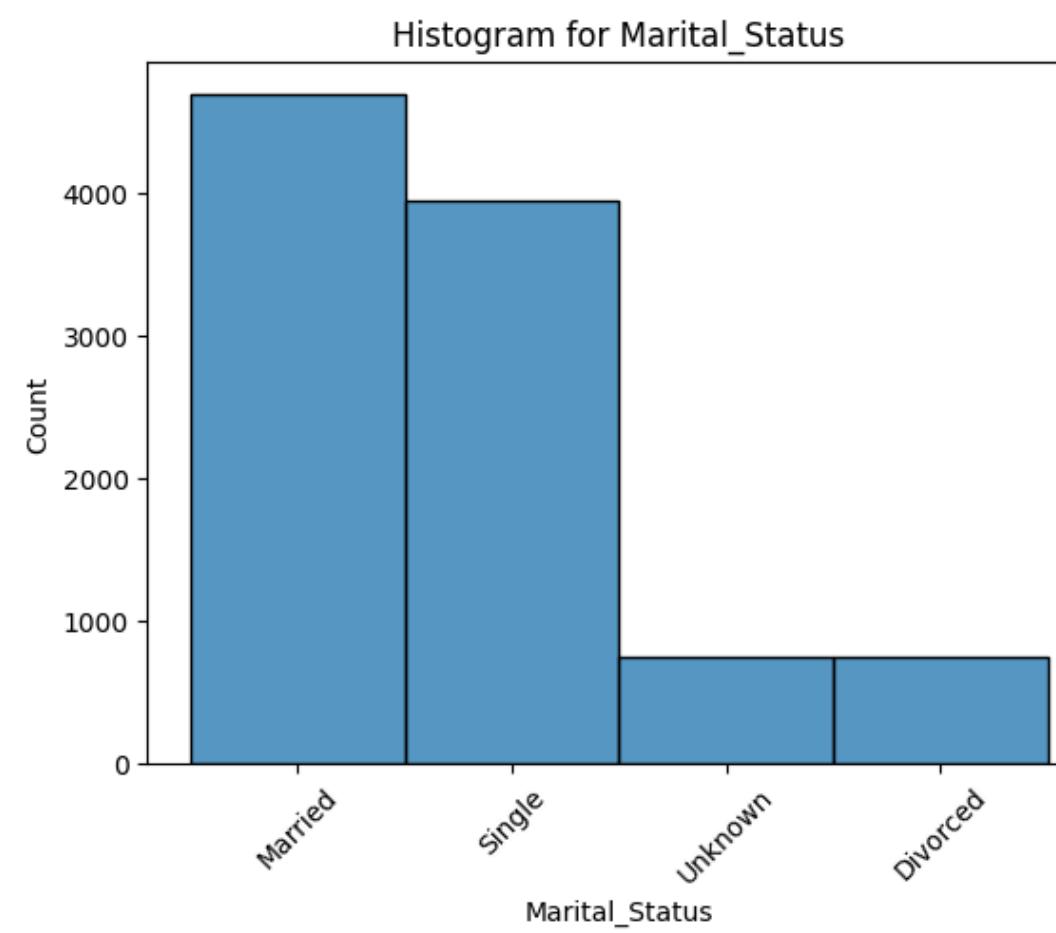
COMPLEX DATA EXPLORATION

```
1 df.info()
2 <class 'pandas.core.frame.DataFrame'>
3 RangeIndex: 10127 entries, 0 to 10126
4 Data columns (total 23 columns):
5 #   Column                                         Non-Null Count  Dtype  
6 --- 
7 0   CLIENTNUM                                     10127 non-null   int64  
8 1   Attrition_Flag                                10127 non-null   object  
9 2   Customer_Age                                  10127 non-null   int64  
10 3   Gender                                       10127 non-null   object  
11 4   Dependent_count                             10127 non-null   int64  
12 5   Education_Level                            10127 non-null   object  
13 6   Marital_Status                             10127 non-null   object  
14 7   Income_Category                           10127 non-null   object  
15 8   Card_Category                             10127 non-null   object  
16 9   Months_on_book                           10127 non-null   int64  
17 10  Total_Relationship_Count                 10127 non-null   int64  
18 11  Months_Inactive_12_mon                   10127 non-null   int64  
19 12  Contacts_Count_12_mon                   10127 non-null   int64  
20 13  Credit_Limit                             10127 non-null   float64 
21 14  Total_Revolving_Bal                     10127 non-null   int64  
22 15  Avg_Open_To_Buy                         10127 non-null   float64 
23 16  Total_Amt_Chng_Q4_Q1                   10127 non-null   float64 
24 17  Total_Trans_Amt                        10127 non-null   int64  
25 18  Total_Trans_Ct                          10127 non-null   int64  
26 19  Total_Ct_Chng_Q4_Q1                   10127 non-null   float64 
27 20  Avg_Utilization_Ratio                  10127 non-null   float64 
28 21  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1 10127 non-null   float64 
29 22  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2 10127 non-null   float64 
30 dtypes: float64(7), int64(10), object(6)
31 memory usage: 1.8+ MB
```

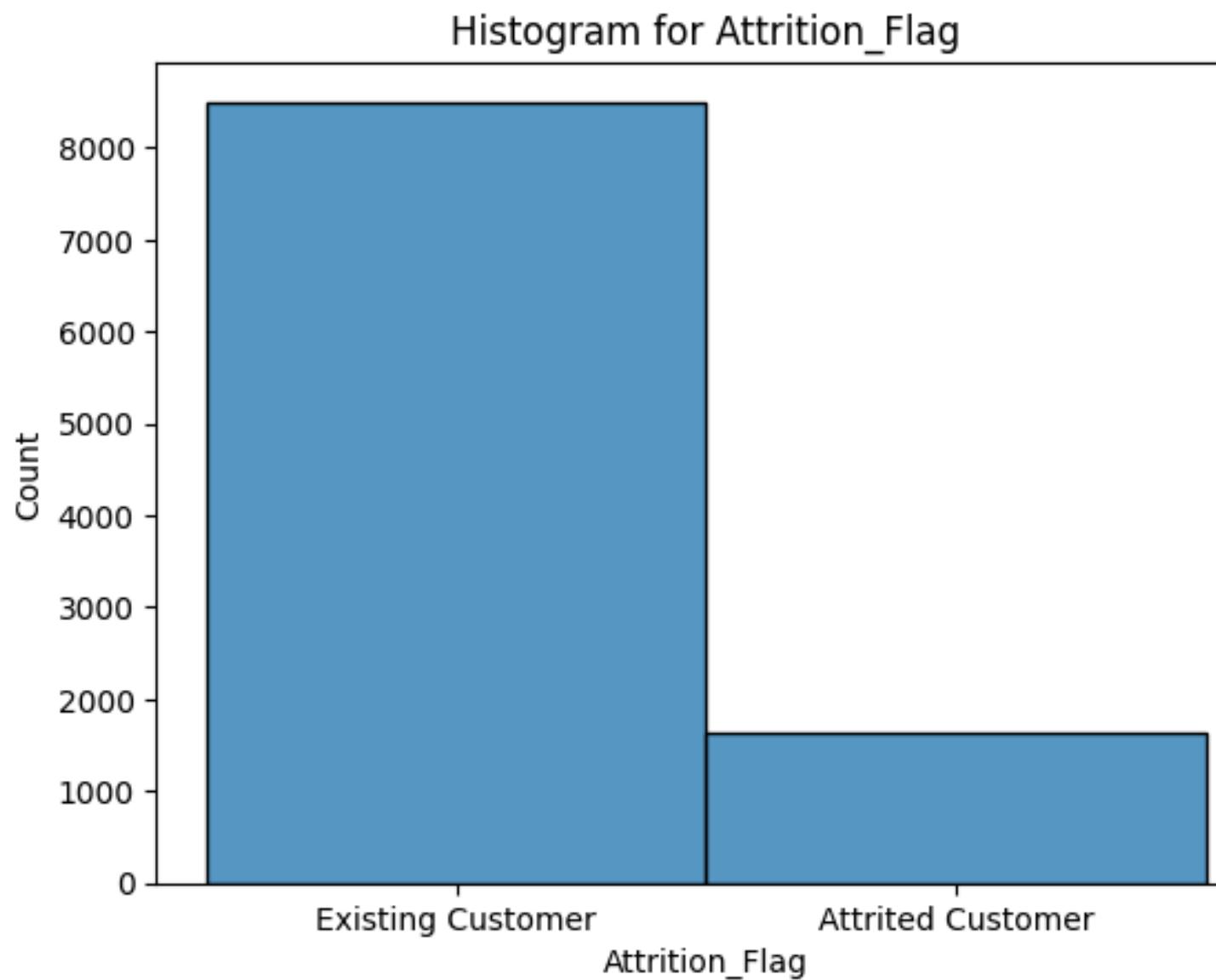
COMPLEX DATA EXPLORATION



COMPLEX DATA EXPLORATION



COMPLEX DATA EXPLORATION



```
[ ] 1 y = df['Attrition_Flag']
2 df = df.drop('Attrition_Flag', axis=1)

[ ] 1 y.value_counts()
```

count

Attrition_Flag	count
Existing Customer	8500
Attrited Customer	1627

dtype: int64

จาก `y.value_counts()` พบว่า dataset นี้เป็นแบบ imbalanced โดยมี Imbalance Ratio ของ Churn rate = $8500/1627 = 5.22$



COMPLEX DATA PRE-PROCESSING

```
[23] 1 categorical_columns = df.select_dtypes("object").columns
      2 print(categorical_columns)
      ↗ Index(['Gender', 'Education_Level', 'Marital_Status', 'Income_Category',
              'Card_Category'],
              dtype='object')

[24] 1 df = pd.get_dummies(df, columns=categorical_columns)
      2 df.head()
```

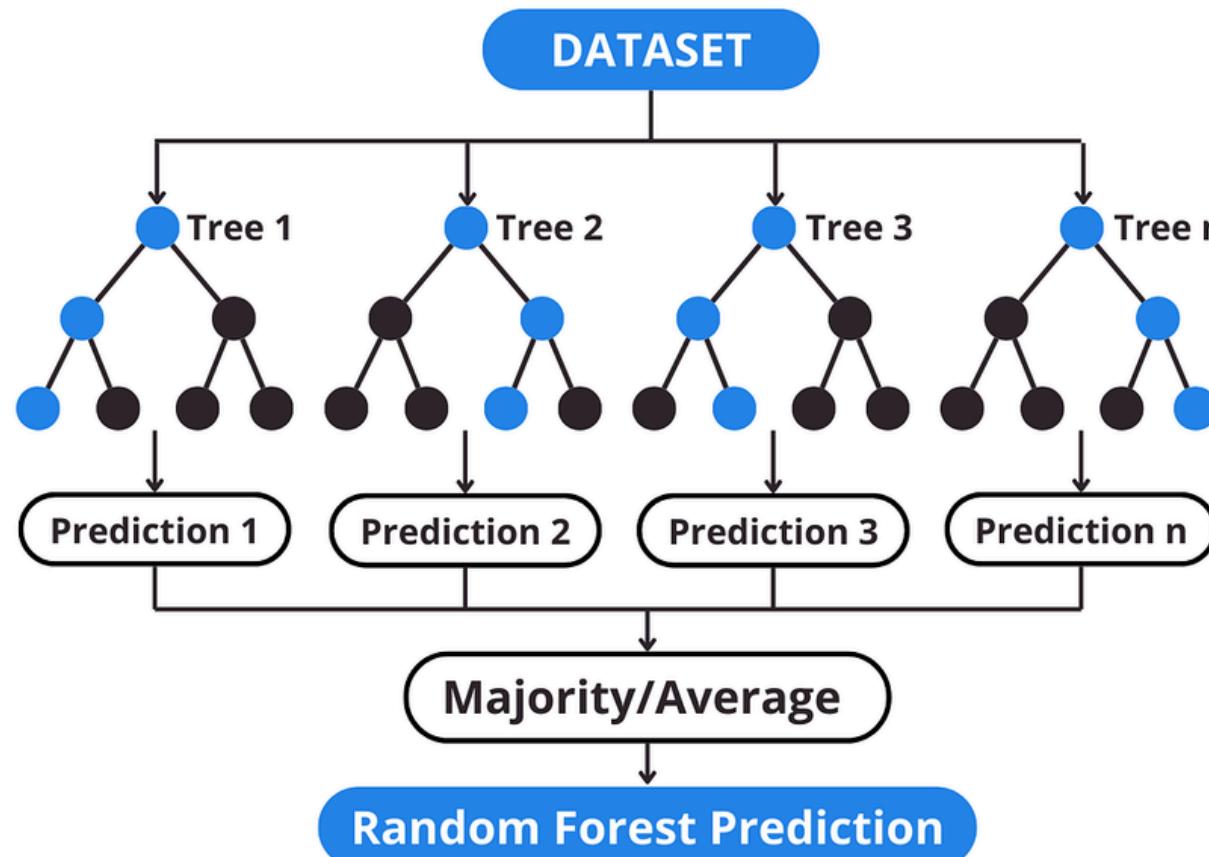
```
1 df.drop(['CLIENTNUM'], axis=1, inplace=True)
2 df.drop(['Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon'],
          axis=1, inplace=True)
3 df.drop(['Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon'],
          axis=1, inplace=True)
4 df.head()
```

	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revolving_Bal	Avg_Open_To_Buy	Total_Amt_Chng_Q4_Q1
0	45	3	39	5	1	3	12691.0	777	11914.0	1.335
1	49	5	44	6	1	2	8256.0	864	7392.0	1.541
2	51	3	36	4	1	0	3418.0	0	3418.0	2.594
3	40	4	34	3	4	1	3313.0	2517	796.0	1.405
4	40	3	21	5	1	0	4716.0	0	4716.0	2.175

Category_>\$40K	Income_Category_>\$60K	Income_Category_>\$80K	Income_Category_Less than \$40K	Income_Category_Unknown	Card_Category_Blue	Card_Category_Gold	Card_Category_Platinum	Card_Category_Silver
- \$60K	- \$80K	- \$120K	than \$40K					
False	True	False	False	False	True	False	False	False
False	False	False	True	False	True	False	False	False
False	False	True	False	False	True	False	False	False
False	False	False	True	False	True	False	False	False
False	True	False	False	False	True	False	False	False

ANALYTIC TECHNIQUE

BALANCEDRANDOMFORESTCLASSIFIER



สำหรับเทคโนโลยี `BalancedRandomForestClassifier` บันทึกของการที่ทำการ balance dataset ก่อนด้วยการ resampling โดยสามารถตั้งค่าการ balance ได้ด้วย parameter `sampling_strategy` หลังจากทำ sampling พังก์ชันจะทำการสร้างต้นไม้ตามจำนวน parameter `n_estimators` จากนั้นในการ predict แต่ละต้นไม้จะให้ผลลัพธ์จากการ predict มารวมกันและทำการเลือกผลลัพธ์สุดท้ายด้วย Majority voting

MODEL PARAMETER

```
model = BalancedRandomForestClassifier(n_estimators=200,  
                                         criterion='entropy',  
                                         max_depth=None,  
                                         min_samples_split=2,  
                                         sampling_strategy='not majority',  
                                         replacement=True,  
                                         bootstrap=False)
```

N_ESTIMATORS

จำนวน Tree ที่ใช้ใน Random Forest ซึ่งกำหนดให้เป็น 200

CRITERION

มาตราที่จะใช้ในการแบ่ง Tree ซึ่งเลือกได้ว่าจะเป็น Gini หรือ Entropy

MAX_DEPTH

ความลึกสูงสุดของต้นไม้

MIN_SAMPLES_SPLIT

จำนวน Sample ที่ต้องใช้อย่างน้อย ก่อนที่จะ Split Tree

SAMPLING_STRATEGY

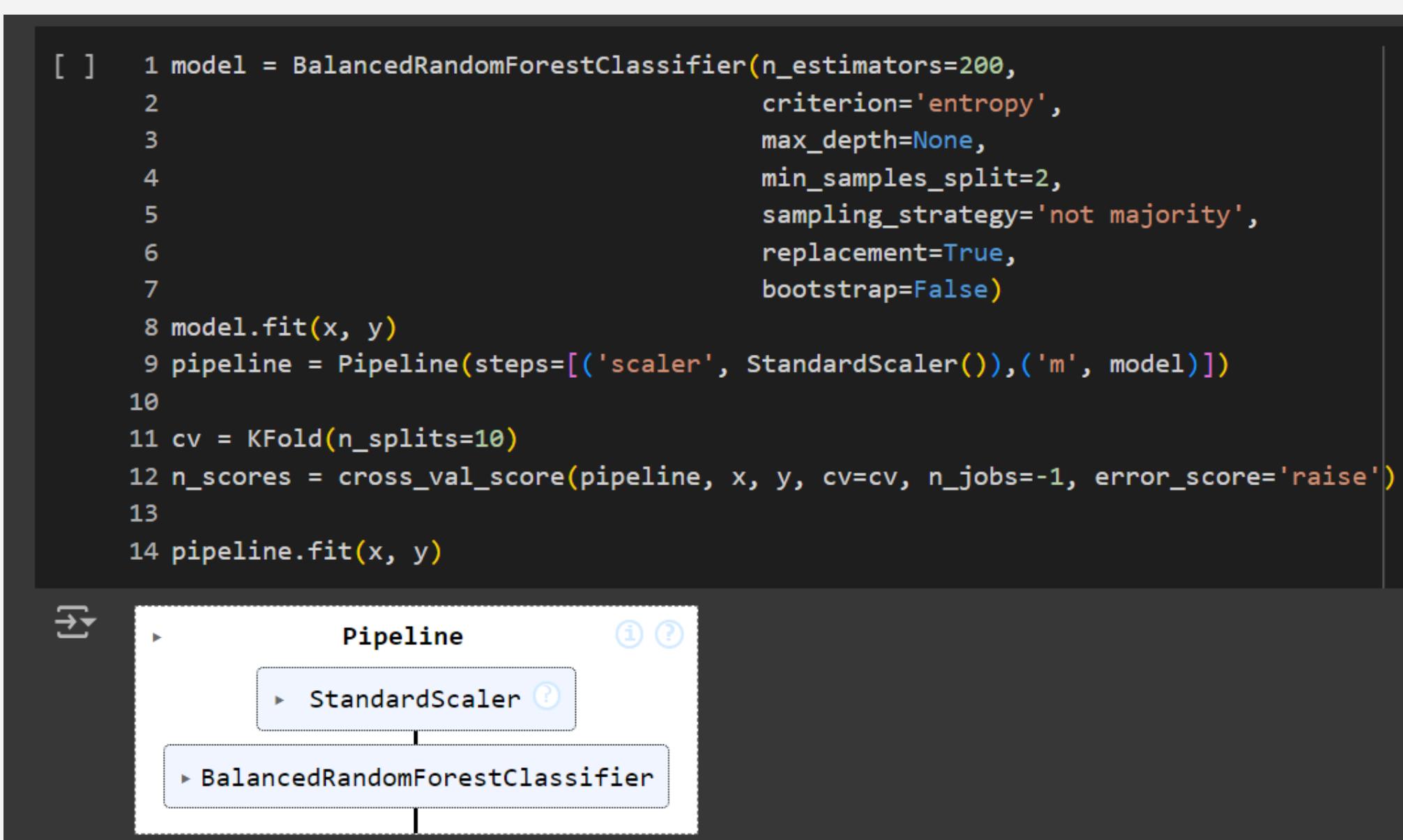
วิธีในการ Sample ค่ามาใช้ใน Tree โดยเลือก 'not majority' เพื่อบำคลาสที่ไม่ใช่ Majority Class มา Sample

REPLACEMENT & BOOTSTRAP

เป็น Default Parameter ที่จะนำ Minority Class มา Sample

DEMO WITH TOOL OR LIBRARIES OR CODING

```
[ ] 1 model = BalancedRandomForestClassifier(n_estimators=200,
2                                     criterion='entropy',
3                                     max_depth=None,
4                                     min_samples_split=2,
5                                     sampling_strategy='not majority',
6                                     replacement=True,
7                                     bootstrap=False)
8 model.fit(x, y)
9 pipeline = Pipeline(steps=[('scaler', StandardScaler()), ('m', model)])
10
11 cv = KFold(n_splits=10)
12 n_scores = cross_val_score(pipeline, x, y, cv=cv, n_jobs=-1, error_score='raise')
13
14 pipeline.fit(x, y)
```



The image shows a Jupyter Notebook cell with the following Python code:

```
[ ] 1 model = BalancedRandomForestClassifier(n_estimators=200,
2                                     criterion='entropy',
3                                     max_depth=None,
4                                     min_samples_split=2,
5                                     sampling_strategy='not majority',
6                                     replacement=True,
7                                     bootstrap=False)
8 model.fit(x, y)
9 pipeline = Pipeline(steps=[('scaler', StandardScaler()), ('m', model)])
10
11 cv = KFold(n_splits=10)
12 n_scores = cross_val_score(pipeline, x, y, cv=cv, n_jobs=-1, error_score='raise')
13
14 pipeline.fit(x, y)
```

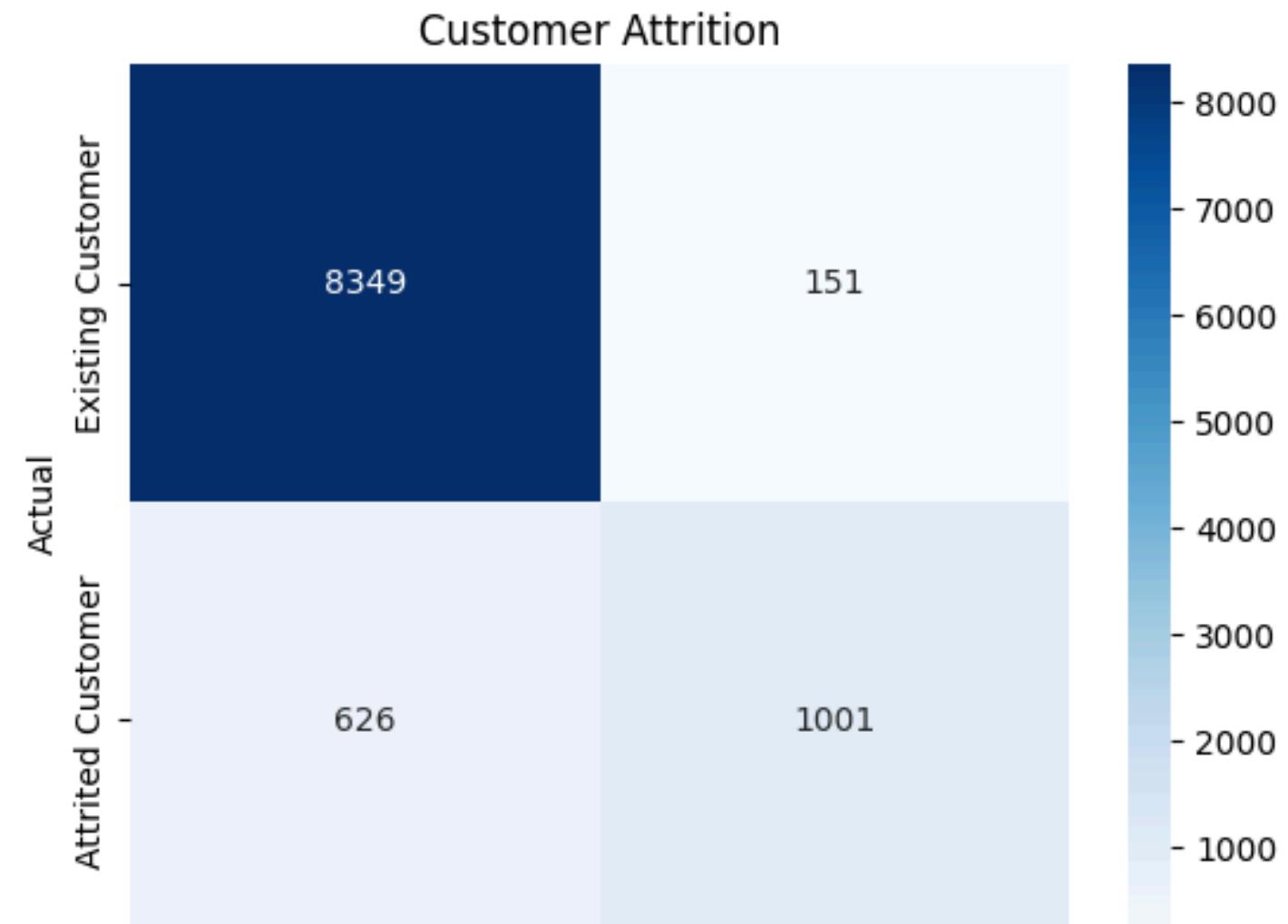
Below the code, a screenshot of a machine learning pipeline interface is displayed. It shows a 'Pipeline' component with two steps: 'StandardScaler' and 'BalancedRandomForestClassifier'. The 'StandardScaler' step is highlighted with a blue border.

DEMO WITH TOOL OR LIBRARIES OR CODING

```
✓ 58 [23] 1 y_pred = cross_val_predict(pipeline, x, y, cv=cv)
      2 print(y_pred[0:20])
→ ['Existing Customer' 'Existing Customer' 'Existing Customer'
   'Existing Customer' 'Existing Customer' 'Existing Customer'
   'Attrited Customer' 'Existing Customer']

✓ 0 [27] 1 n_scores.mean()
      2
→ 0.9256310259121537
```

MODEL RESULTS



จาก Confusion Matrix เราจะได้ผลลัพธ์ดังนี้

- Class Existing Customer หรือกลุ่มลูกค้าที่ยังคงอยู่
 - มีค่า Precision อยู่ที่ 0.93 หรือ 93% หมายความว่า Model ของเรามาตรฐานได้ว่าลูกค้าคนใดจะยังคงใช้บริการของเราต่อ
 - มีค่า Recall อยู่ที่ 0.98 หรือ 98% หมายความว่า Model ของเรามาตรฐานได้ว่าลูกค้าคนใดจะใช้บริการต่อ 98% จากลูกค้าทั้งหมดที่ยังใช้บริการต่อ
- Class Attrited Customer หรือกลุ่มลูกค้าเลิกใช้บริการ
 - มีค่า Precision อยู่ที่ 0.87 หรือ 87% หมายความว่า Model ของเรามาตรฐานได้ว่าลูกค้าคนใดจะเลิกใช้บริการ 87%
 - มีค่า Recall อยู่ที่ 0.62 หรือ 62% หมายความว่า Model ของเรามาตรฐานได้ว่าลูกค้าคนใดจะเลิกใช้บริการ 62% จากลูกค้าทั้งหมดที่จะเลิกใช้บริการ

	precision	recall	f1-score	support
Attrited Customer	0.87	0.62	0.72	1627
Existing Customer	0.93	0.98	0.96	8500
accuracy			0.92	10127
macro avg	0.90	0.80	0.84	10127
weighted avg	0.92	0.92	0.92	10127

EVALUATION

	precision	recall	f1-score	support
Attrited Customer	0.61	0.70	0.65	1627
Existing Customer	0.94	0.91	0.93	8500
accuracy			0.88	10127
macro avg	0.78	0.81	0.79	10127
weighted avg	0.89	0.88	0.88	10127

DECISION TREE

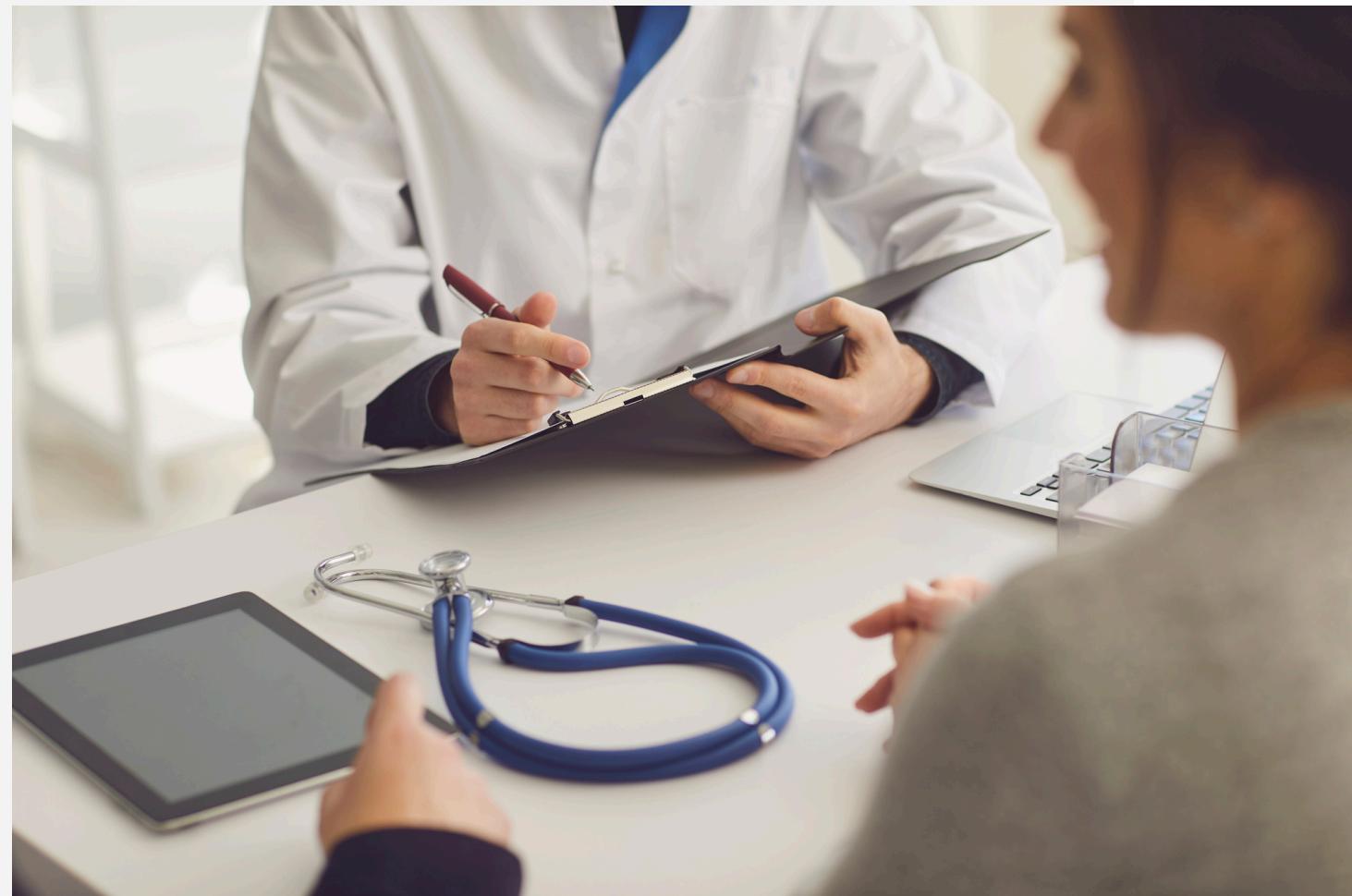
	precision	recall	f1-score	support
Attrited Customer	0.88	0.62	0.73	1627
Existing Customer	0.93	0.98	0.96	8500
accuracy			0.92	10127
macro avg	0.90	0.80	0.84	10127
weighted avg	0.92	0.92	0.92	10127

BALANCED RANDOM FOREST

ทำการทดสอบโดยการเปรียบเทียบกับ Decision Tree จะเห็นได้ว่า Balanced Random Forest มี Performance ที่ดีกว่าอย่างเห็นได้ชัด โดยมี Accuracy 92% ส่วน Decision Tree ปกติมี Accuracy 88%, F1-score ของ Class ที่เราสนใจอย่าง Attrited Customer (Churn) เพิ่มขึ้นจาก 65% เป็น 73% โดยมี Precision ที่เพิ่มขึ้นถึง 27% (61% -> 88%) แต่มี Trade-off ที่ Recall ลดลงไป 8% (70% -> 62%) ส่วน Class Existing Customer ที่ F1-score, Precision และ Recall ที่ดีขึ้น เช่นกัน จึงสามารถสรุปได้ว่า การทำ Balanced Random Forest สามารถ Perform ได้ดีกับ Imbalanced Dataset

POSSIBLE APPLICATIONS

Random Forest มีข้อดีที่ช่วยในการจัดการกับ Data ที่มีขนาดใหญ่ มีความยืดหยุ่นสูง ถูกรบกวนได้น้อยจากปัญหาต่างๆ ไม่ว่าจะเป็น Imbalance Dataset, Noise หรือ Overfitting เนื่องจากหลักการ Ensemble Learning (Bagging) ใช้ผลโหวตส่วนใหญ่ในการ Predict ผลลัพธ์ และโดยเฉพาะอย่างยิ่งกับ Balanced Random Forest ที่ทำการ Resample Dataset ก่อนอีก ยิ่งทำให้จัดการกับ Imbalanced Dataset ได้ดียิ่งขึ้นไปอีก จึงสามารถนำไปประยุกต์ใช้ได้กับ Dataset หลายประเภท เช่น



- การนำมายความเสี่ยงที่จะเป็นโรคหายากต่างๆ เนื่องจากโรคหายากนี้มีจำนวนข้อมูลผู้ป่วยน้อย ทำให้ Dataset เกิดความ Imbalance การใช้ Balanced Random Forest จึงสามารถช่วยทำให้ผลการทำนายแม่นยำขึ้นได้
- การนำมายความเสี่ยงที่เครื่องจักรในโรงงานจะชำรุด เนื่องจากในโรงงานอุตสาหกรรม การชำรุดของเครื่องจักรเกิดขึ้นไม่บ่อย แต่หากชำรุดแล้วจะเกิดผลเสียกับระบบอย่างมาก การนำ Balanced Random Forest ไปช่วยทำนายจึงจะช่วยลดความเสี่ยงที่จะเกิดขึ้นได้เป็นอย่างดี

REFERENCES

DATASET

Credit Card Customers

<https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers>

KNOWLEDGE

BalancedRandomForestClassifier

<https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html#r2d8f3e873ec3-1>

Random Forest Tuning - Tree Depth And Number Of Trees

<https://stackoverflow.com/questions/34997134/random-forest-tuning-tree-depth-and-number-of-trees>

THANK YOU