Predict คนลาออกจากงาน

Absurdism

ภูมิภัทร พินธิระ 6410301032
อนันต์โชติ ธนโชคธัญสิริ 6410301033

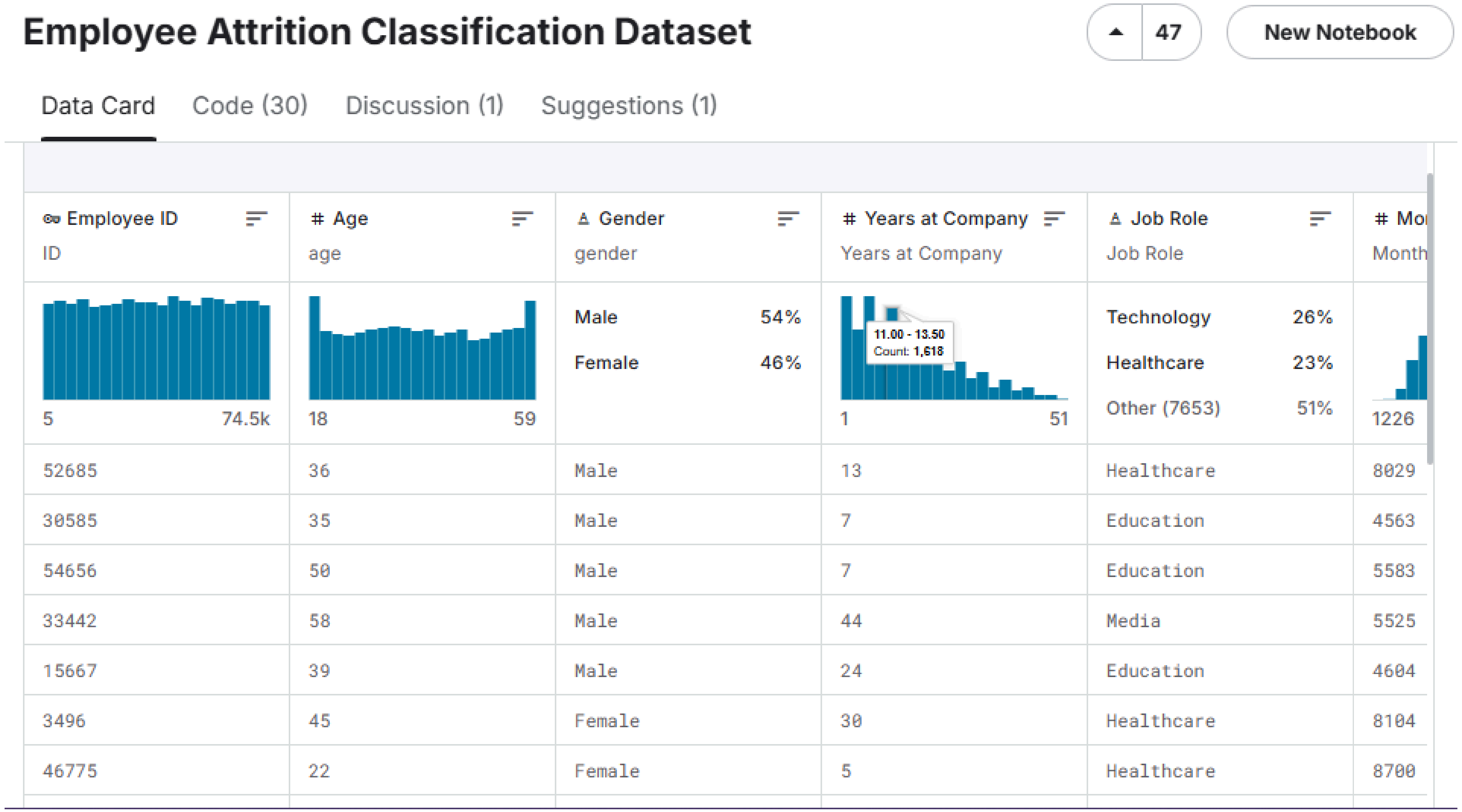
Problem

- พนักงานเป็นฟันเฟืองหนึ่งขององค์กร
- การลาออกของพนักงานส่งผลต่อการทำงาน, ประสิทธิภาพขององค์กร
- ฉะนั้นแต่ละองค์กรควรที่จะรับรู้ว่าปัจจัย (factor) ใดที่จะส่งผลต่อการลาออกของพนักงาน เพื่อที่จะได้เตรียมการในการแก้ไขปัญหา

Goal

- ใช้เพื่อวิเคราะห์ว่าปัจจัยใดที่ส่งผลต่อการลาออกของพนักงาน

Data Set



Employee Attrition Classification Dataset

47 | New Notebook

Data Card | Code (30) | Discussion (1) | Suggestions (1)

Employee ID	Age	Gender	Years at Company	Job Role	Mo
ID	age	gender	Years at Company	Job Role	Month
5 – 74.5k	18 – 59	Male 54% Female 46%	1 – 51 11.00 - 13.50 Count: 1,618	Technology 26% Healthcare 23% Other (7653) 51%	1226
52685	36	Male	13	Healthcare	8029
30585	35	Male	7	Education	4563
54656	50	Male	7	Education	5583
33442	58	Male	44	Media	5525
15667	39	Male	24	Education	4604
3496	45	Female	30	Healthcare	8104
46775	22	Female	5	Healthcare	8700

https://www.kaggle.com/datasets/stealthtechnologies/employee-attrition-dataset/data

Data Understanding

```
RangeIndex: 74498 entries, 0 to 74497
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       74498 non-null  int64
 1   Gender                    74498 non-null  object
 2   Years at Company          74498 non-null  int64
 3   Job Role                  74498 non-null  object
 4   Monthly Income            74498 non-null  int64
 5   Work-Life Balance         74498 non-null  object
 6   Job Satisfaction          74498 non-null  object
 7   Performance Rating        74498 non-null  object
 8   Number of Promotions      74498 non-null  int64
 9   Overtime                  74498 non-null  object
 10  Distance from Home        74498 non-null  int64
 11  Education Level           74498 non-null  object
 12  Marital Status            74498 non-null  object
 13  Number of Dependents      74498 non-null  int64
 14  Job Level                 74498 non-null  object
 15  Company Size              74498 non-null  object
 16  Company Tenure            74498 non-null  int64
 17  Remote Work               74498 non-null  object
 18  Leadership Opportunities  74498 non-null  object
 19  Innovation Opportunities  74498 non-null  object
 20  Company Reputation        74498 non-null  object
 21  Employee Recognition      74498 non-null  object
 22  Attrition                 74498 non-null  object
dtypes: int64(7), object(16)
```

```
data['Attrition'].value_counts()
```

[7] ✓ 0.0s

```
Attrition
Stayed    39128
Left      35370
Name: count, dtype: int64
```

```
data['Age'].describe()
```

✓ 0.0s

```
count    74498.000000
mean        38.529746
std         12.083456
min         18.000000
25%         28.000000
50%         39.000000
75%         49.000000
max         59.000000
Name: Age, dtype: float64
```

```
data['Gender'].value_counts()
```

✓ 0.0s

```
Gender
Male      40826
Female    33672
Name: count, dtype: int64
```

```
data['Years at Company'].describe()
```

[10] ✓ 0.0s

```
count    74498.000000
mean        15.721603
std         11.223744
min          1.000000
25%          7.000000
50%         13.000000
75%         23.000000
max         51.000000
Name: Years at Company, dtype: float64
```

```
data['Years at Company'].value_counts()
```

[11] ✓ 0.0s

```
Years at Company
5     3084
1     3056
2     3039
8     3015
10    2987
9     2965
3     2961
6     2952
7     2933
4     2903
11    2893
12    2758
13    2547
14    2349
15    2281
16    2145
17    1994
18    1889
19    1754
20    1729
21    1606
23    1557
22    1537
24    1385
...
```

```python
data['Job Role'].value_counts()
```

2] ✓ 0.0s

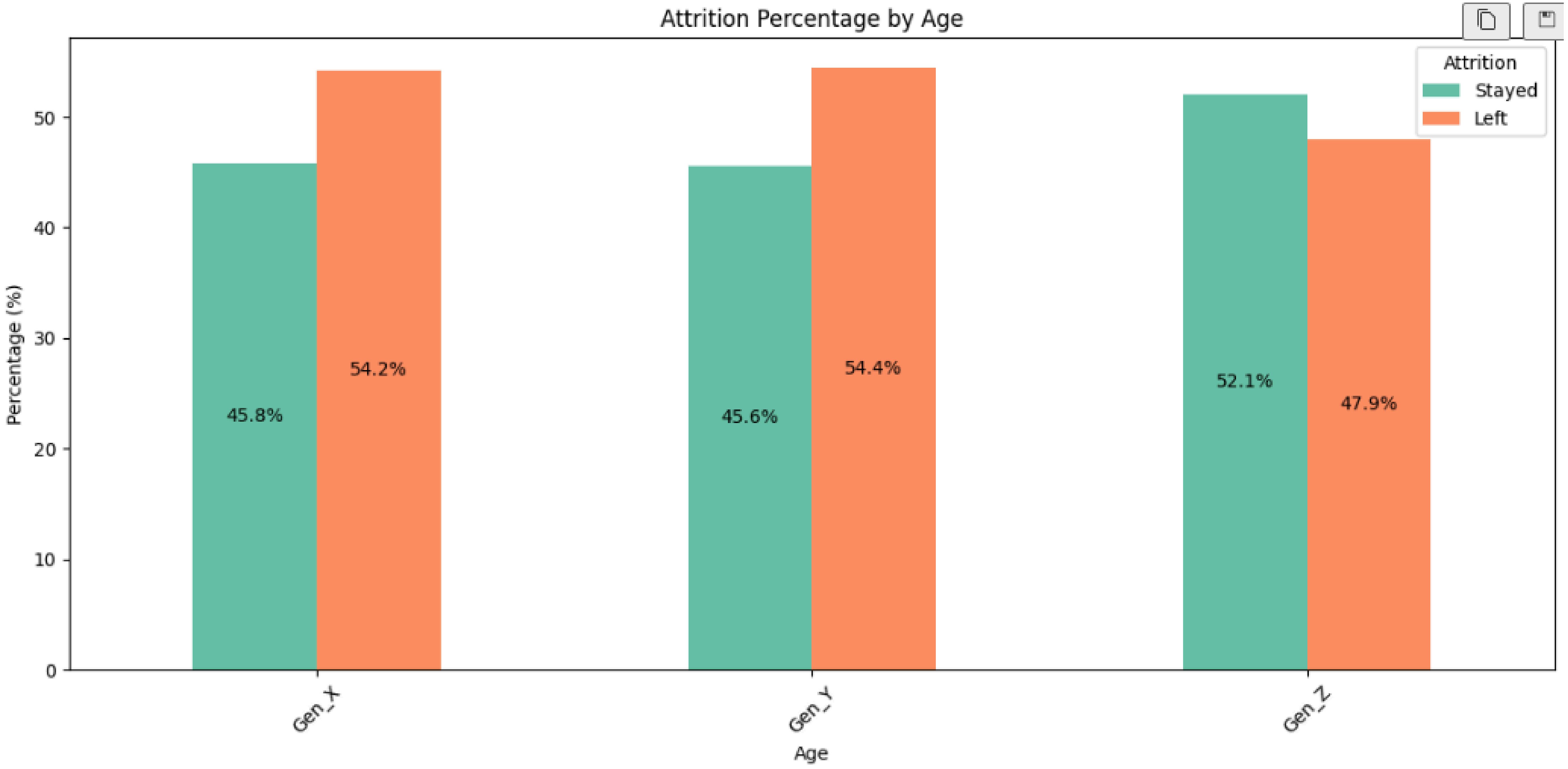
```
Job Role
Technology    19322
Healthcare    17074
Education     15658
Media         11996
Finance       10448
Name: count, dtype: int64
```

Feature Engineering

```python
def classify_birth_year_group(gen):
    birth_year = 2024 - gen
    if birth_year >= 2013:
        return 'Gen_Alpha'
    elif 1995 <= birth_year <= 2012:
        return 'Gen_Z'
    elif 1980 <= birth_year <= 1994:
        return 'Gen_Y'
    elif 1965 <= birth_year <= 1979:
        return 'Gen_X'
    else:
        return 'Baby_Boomer'


data['Generation'] = data['Age'].apply(classify_birth_year_group)

data

# X11 = Prepro_Data[['Generation']]
```

✓ 0.1s

Years at	Monthly	Work-


Attrition Percentage by Age
Attrition
Stayed
Left
Percentage (%)
0
10
20
30
40
50
45.8%
54.2%
45.6%
54.4%
52.1%
47.9%
Gen_X
Gen_Y
Gen_Z
Age

Data Preparation

```python
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
continuous_cols = ['Age', 'Years at Company', 'Monthly Income', 'Number of Promotions', 'Distance from Home', 'Company Tenure']
scaler = StandardScaler()
data[continuous_cols] = scaler.fit_transform(data[continuous_cols])
```

Python

```
(count    7.449800e+04
 mean     2.114517e-16
 std      1.000007e+00
 min     -1.699008e+00
 25%     -8.714242e-01
 50%      3.891746e-02
 75%      8.665008e-01
 max      1.694084e+00
 Name: Age, dtype: float64,
 Age
 -0.374874    1875
  0.121676    1861
  1.197534    1843
  0.535467    1842
 -0.043841    1834
  1.363051    1824
  0.369951    1822
 -0.705908    1818
  0.038917    1813
  0.700984    1806
 -0.292116    1806
 -0.209358    1795
  0.618226    1793
 -0.457633    1791
 -1.616249    1789
...
 -1.285216    1711
  1.114776    1708
  0.783742    1702
 -1.699008    1702
```

```
label_encoder = LabelEncoder()
data['Attrition'] = label_encoder.fit_transform(data['Attrition'])
```

```
data['Attrition'].value_counts()
```

```
Attrition
1    39128
0    35370
Name: count, dtype: int64
```

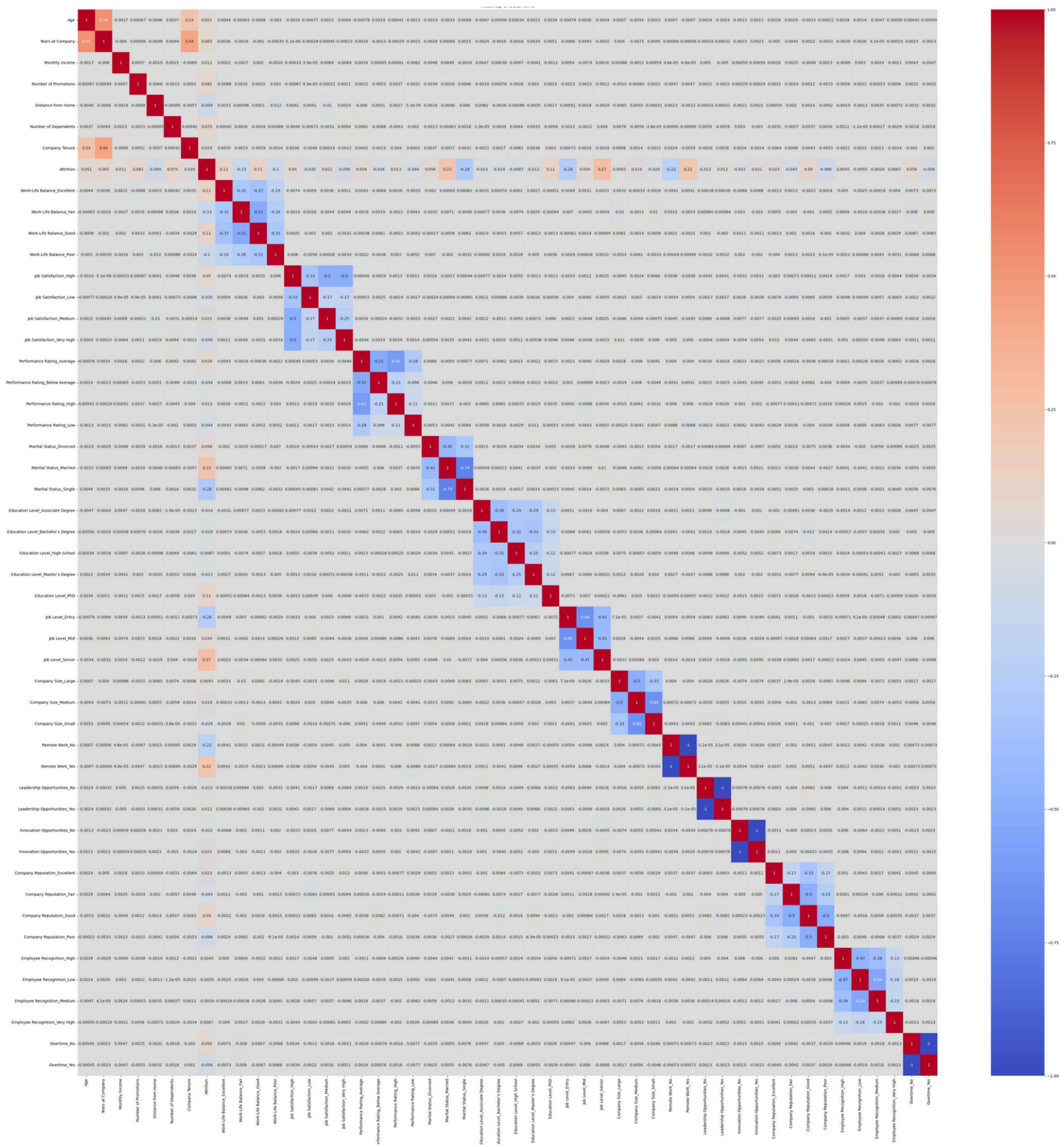
1 คือ อยู่ต่อ

0 คือ ลาออก

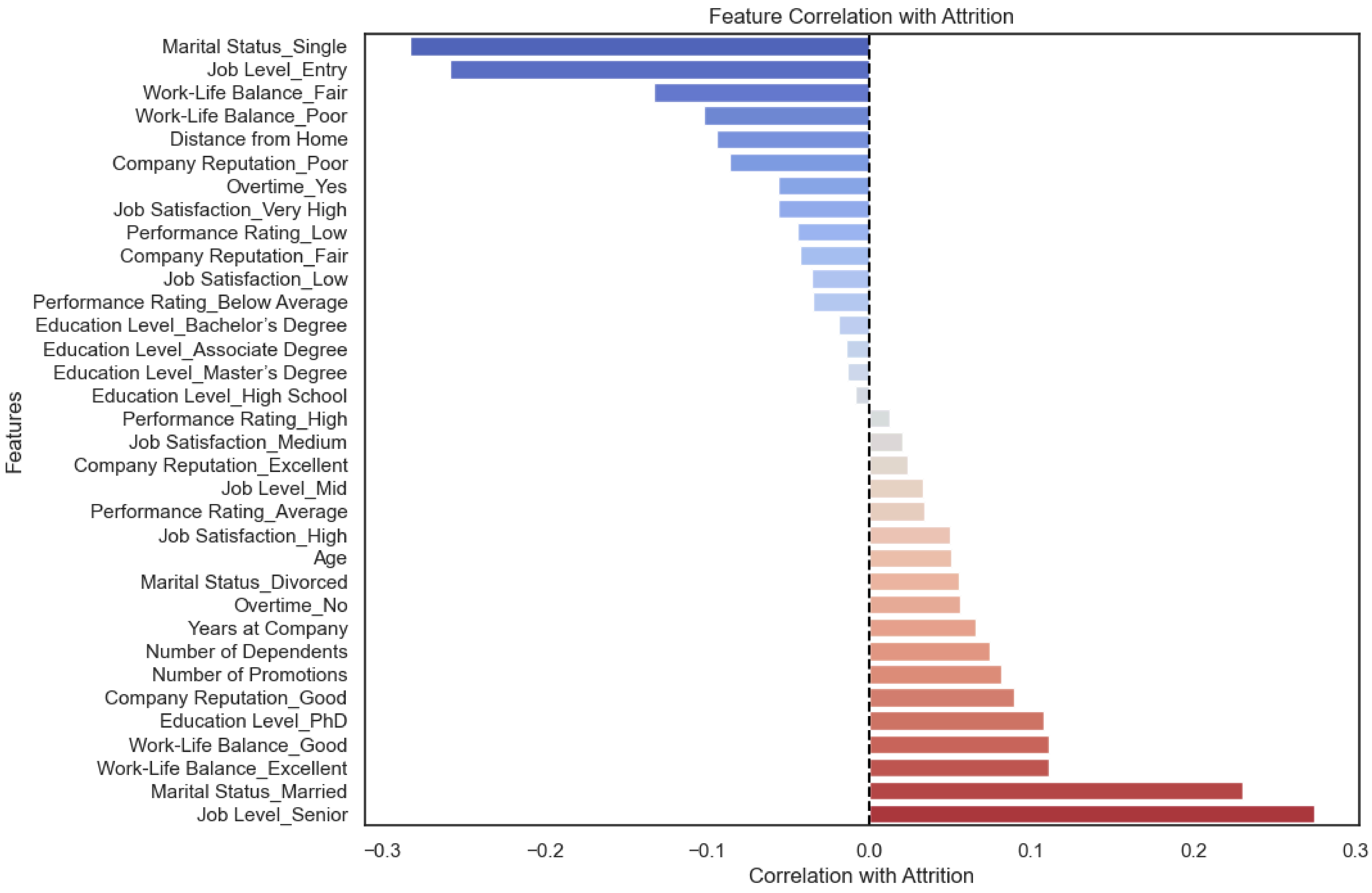
```python
# คอลัมน์ที่เป็นตัวแปรประเภท categorical ที่ต้องการทำ One-Hot Encoding
categorical_cols = ['Gender', 'Job Role', 'Work-Life Balance', 'Job Satisfaction',
                    'Performance Rating', 'Marital Status', 'Education Level',
                    'Job Level', 'Company Size', 'Remote Work',
                    'Leadership Opportunities', 'Innovation Opportunities',
                    'Company Reputation', 'Employee Recognition', 'Overtime']

# ทำ One-Hot Encoding โดยใช้ pd.get_dummies()
data = pd.get_dummies(data, columns=categorical_cols)

# แสดงผลลัพธ์
print(data.head())
```

```
57
Index(['Age', 'Years at Company', 'Monthly Income', 'Number of Promotions',
       'Distance from Home', 'Number of Dependents', 'Company Tenure',
       'Attrition', 'Gender_Female', 'Gender_Male', 'Job Role_Education',
       'Job Role_Finance', 'Job Role_Healthcare', 'Job Role_Media',
       'Job Role_Technology', 'Work-Life Balance_Excellent',
       'Work-Life Balance_Fair', 'Work-Life Balance_Good',
       'Work-Life Balance_Poor', 'Job Satisfaction_High',
       'Job Satisfaction_Low', 'Job Satisfaction_Medium',
       'Job Satisfaction_Very High', 'Performance Rating_Average',
       'Performance Rating_Below Average', 'Performance Rating_High',
       'Performance Rating_Low', 'Marital Status_Divorced',
       'Marital Status_Married', 'Marital Status_Single',
       'Education Level_Associate Degree', 'Education Level_Bachelor’s Degree',
       'Education Level_High School', 'Education Level_Master’s Degree',
       'Education Level_PhD', 'Job Level_Entry', 'Job Level_Mid',
       'Job Level_Senior', 'Company Size_Large', 'Company Size_Medium',
       'Company Size_Small', 'Remote Work_No', 'Remote Work_Yes',
       'Leadership Opportunities_No', 'Leadership Opportunities_Yes',
       'Innovation Opportunities_No', 'Innovation Opportunities_Yes',
       'Company Reputation_Excellent', 'Company Reputation_Fair',
       'Company Reputation_Good', 'Company Reputation_Poor',
       'Employee Recognition_High', 'Employee Recognition_Low',
       'Employee Recognition_Medium', 'Employee Recognition_Very High',
       'Overtime_No', 'Overtime_Yes'],
      dtype='object')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74498 entries, 0 to 74497
Data columns (total 57 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Age                                74498 non-null  float64
 1   Years at Company                   74498 non-null  float64
 2   Monthly Income                     74498 non-null  float64
 3   Number of Promotions               74498 non-null  float64
 4   Distance from Home                 74498 non-null  float64
 5   Number of Dependents               74498 non-null  int64  
 6   Company Tenure                     74498 non-null  float64
 7   Attrition                          74498 non-null  int64  
 8   Gender_Female                      74498 non-null  bool   
 9   Gender_Male                        74498 non-null  bool   
 10  Job Role_Education                 74498 non-null  bool   
 11  Job Role_Finance                   74498 non-null  bool   
 12  Job Role_Healthcare                74498 non-null  bool   
 13  Job Role_Media                     74498 non-null  bool   
 14  Job Role_Technology                74498 non-null  bool   
 15  Work-Life Balance_Excellent        74498 non-null  bool   
 16  Work-Life Balance_Fair             74498 non-null  bool   
 17  Work-Life Balance_Good             74498 non-null  bool   
 18  Work-Life Balance_Poor             74498 non-null  bool   
 19  Job Satisfaction_High              74498 non-null  bool   
...
 56  Overtime_Yes                       74498 non-null  bool   
dtypes: bool(49), float64(6), int64(2)
```



ถ้า Correalation Matrix มีค่า 0.03 ขึ้นไป
แสดงค่านั้นว่ามีผลต่อการลาออกมาก



Feature Correlation with Attrition
Marital Status_Single
Job Level_Entry
Work-Life Balance_Fair
Work-Life Balance_Poor
Distance from Home
Company Reputation_Poor
Overtime_Yes
Job Satisfaction_Very High
Performance Rating_Low
Company Reputation_Fair
Job Satisfaction_Low
Performance Rating_Below Average
Education Level_Bachelor's Degree
Education Level_Associate Degree
Education Level_Master's Degree
Education Level_High School
Performance Rating_High
Job Satisfaction_Medium
Company Reputation_Excellent
Job Level_Mid
Performance Rating_Average
Job Satisfaction_High
Age
Marital Status_Divorced
Overtime_No
Years at Company
Number of Dependents
Number of Promotions
Company Reputation_Good
Education Level_PhD
Work-Life Balance_Good
Work-Life Balance_Excellent
Marital Status_Married
Job Level_Senior
Features
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Correlation with Attrition

```python
data = data.drop((['Gender_Female', 'Gender_Male' , 'Job Role_Education' ,  'Job Role_Finance', 'Job Role_Healthcare', 'Job Role_Media',
        'Job Role_Technology']), axis=1)
print(data.columns)
```

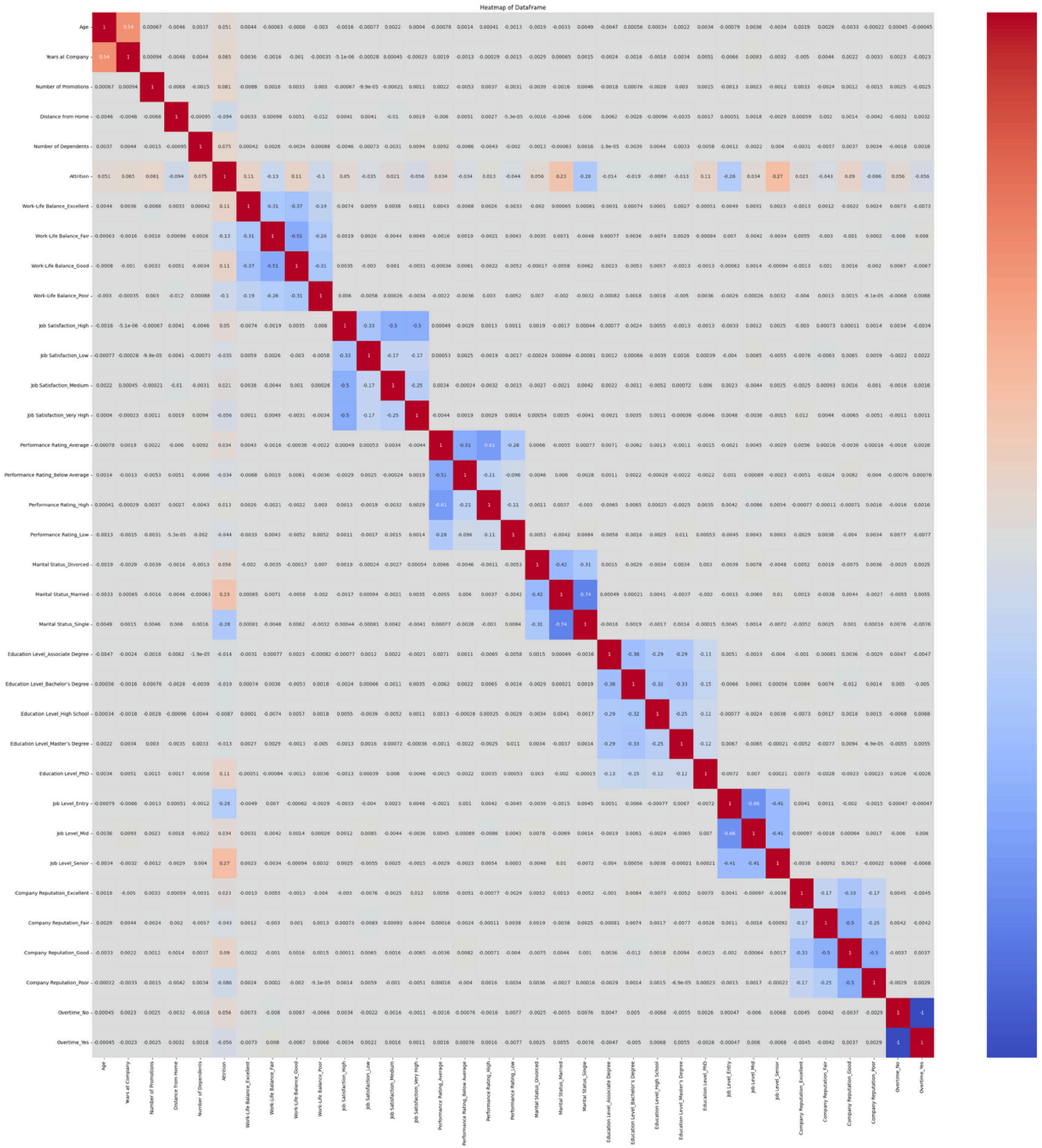
Python

```python
data = data.drop((['Employee Recognition_Medium' , 'Employee Recognition_Low', 'Employee Recognition_High' ,  'Employee Recognition_Very
        'Innovation Opportunities_Yes' , 'Leadership Opportunities_No', 'Leadership Opportunities_Yes' , 'Company Size_Medium', 'Company S
        'Remote Work_No', 'Remote Work_Yes' , 'Company Tenure' , 'Monthly Income' , 'Company Size_Large']), axis=1)
```

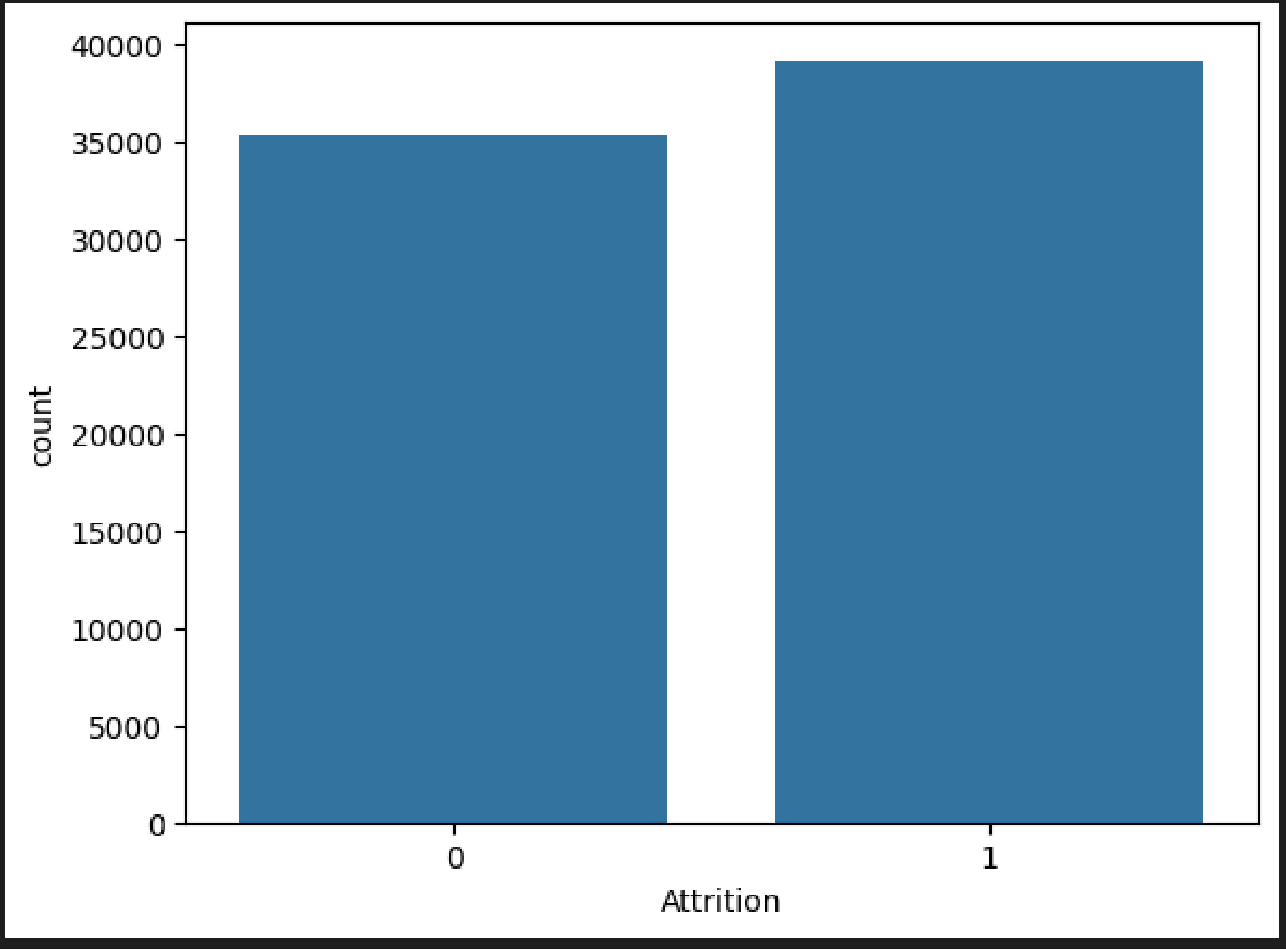
Python

```
print(len(data.columns))
print(data.columns)
```

```
35
Index(['Age', 'Years at Company', 'Number of Promotions', 'Distance from Home',
       'Number of Dependents', 'Attrition', 'Work-Life Balance_Excellent',
       'Work-Life Balance_Fair', 'Work-Life Balance_Good',
       'Work-Life Balance_Poor', 'Job Satisfaction_High',
       'Job Satisfaction_Low', 'Job Satisfaction_Medium',
       'Job Satisfaction_Very High', 'Performance Rating_Average',
       'Performance Rating_Below Average', 'Performance Rating_High',
       'Performance Rating_Low', 'Marital Status_Divorced',
       'Marital Status_Married', 'Marital Status_Single',
       'Education Level_Associate Degree', 'Education Level_Bachelor’s Degree',
       'Education Level_High School', 'Education Level_Master’s Degree',
       'Education Level_PhD', 'Job Level_Entry', 'Job Level_Mid',
       'Job Level_Senior', 'Company Reputation_Excellent',
       'Company Reputation_Fair', 'Company Reputation_Good',
       'Company Reputation_Poor', 'Overtime_No', 'Overtime_Yes'],
      dtype='object')
```


Heatmap of DataFrame
Age
Years at Company
Number of Promotions
Distance from Home
Number of Dependents
Attrition
Work-Life Balance_Excellent
Work-Life Balance_Fair
Work-Life Balance_Good
Work-Life Balance_Poor
Job Satisfaction_High
Job Satisfaction_Low
Job Satisfaction_Medium
Job Satisfaction_Very High
Performance Rating_Average
Performance Rating_Below Average
Performance Rating_High
Performance Rating_Low
Marital Status_Divorced
Marital Status_Married
Marital Status_Single
Education Level_Associate Degree
Education Level_Bachelor's Degree
Education Level_High School
Education Level_Master's Degree
Education Level_PhD
Job Level_Entry
Job Level_Mid
Job Level_Senior
Company Reputation_Excellent
Company Reputation_Fair
Company Reputation_Good
Company Reputation_Poor
Overtime_No
Overtime_Yes

```
sns.countplot(x='Attrition', data=data)
plt.show()
```

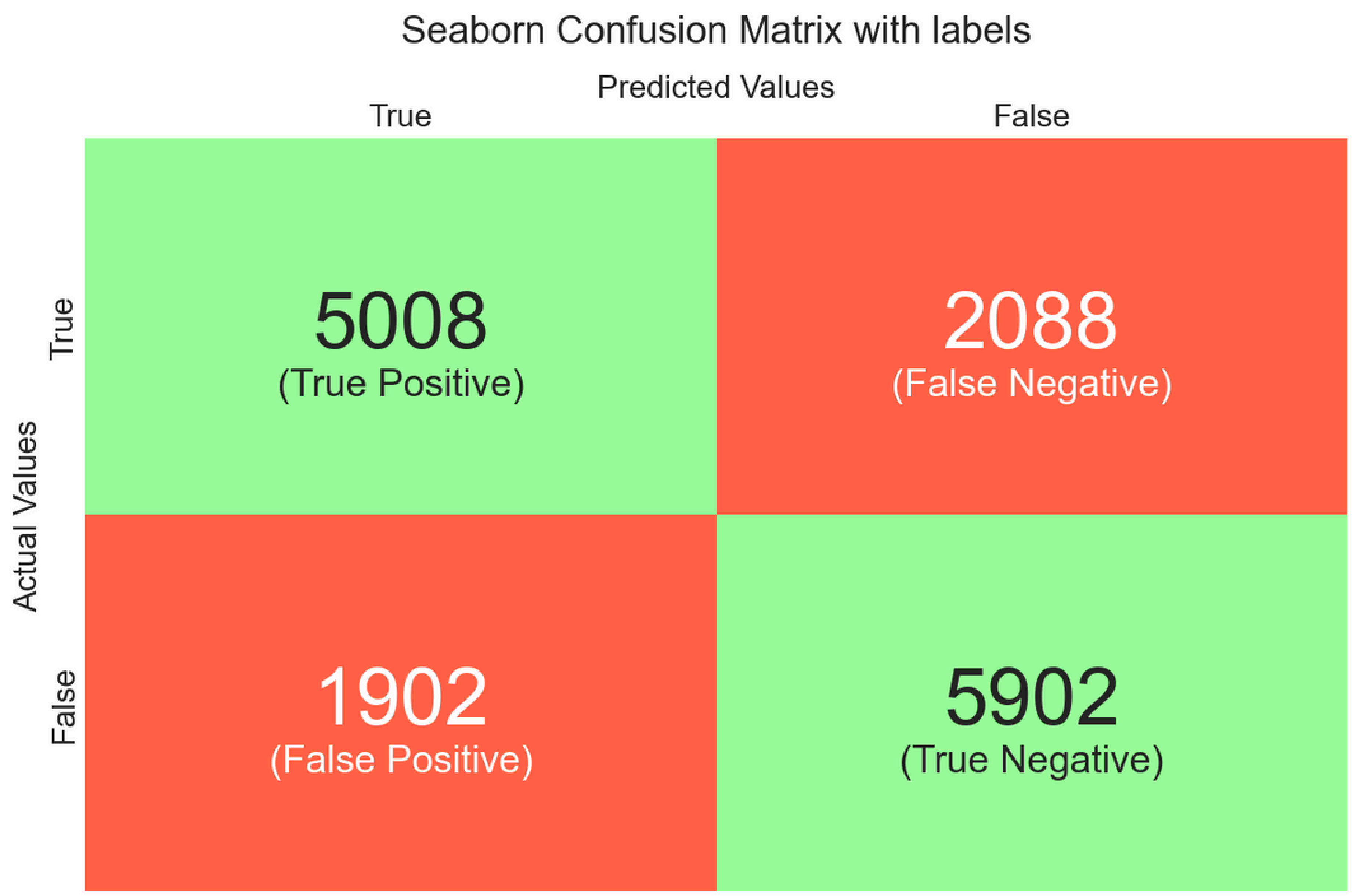


```python
# แบ่งข้อมูล
X = data.drop(columns=['Attrition'])  # Features (ทั้งหมดยกเว้น Attrition)
y = data['Attrition']                 # Target column

# แบ่งข้อมูล
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Train set size: {X_train.shape}, Test set size: {X_test.shape}")
```

```
Train set size: (59598, 34), Test set size: (14900, 34)
```

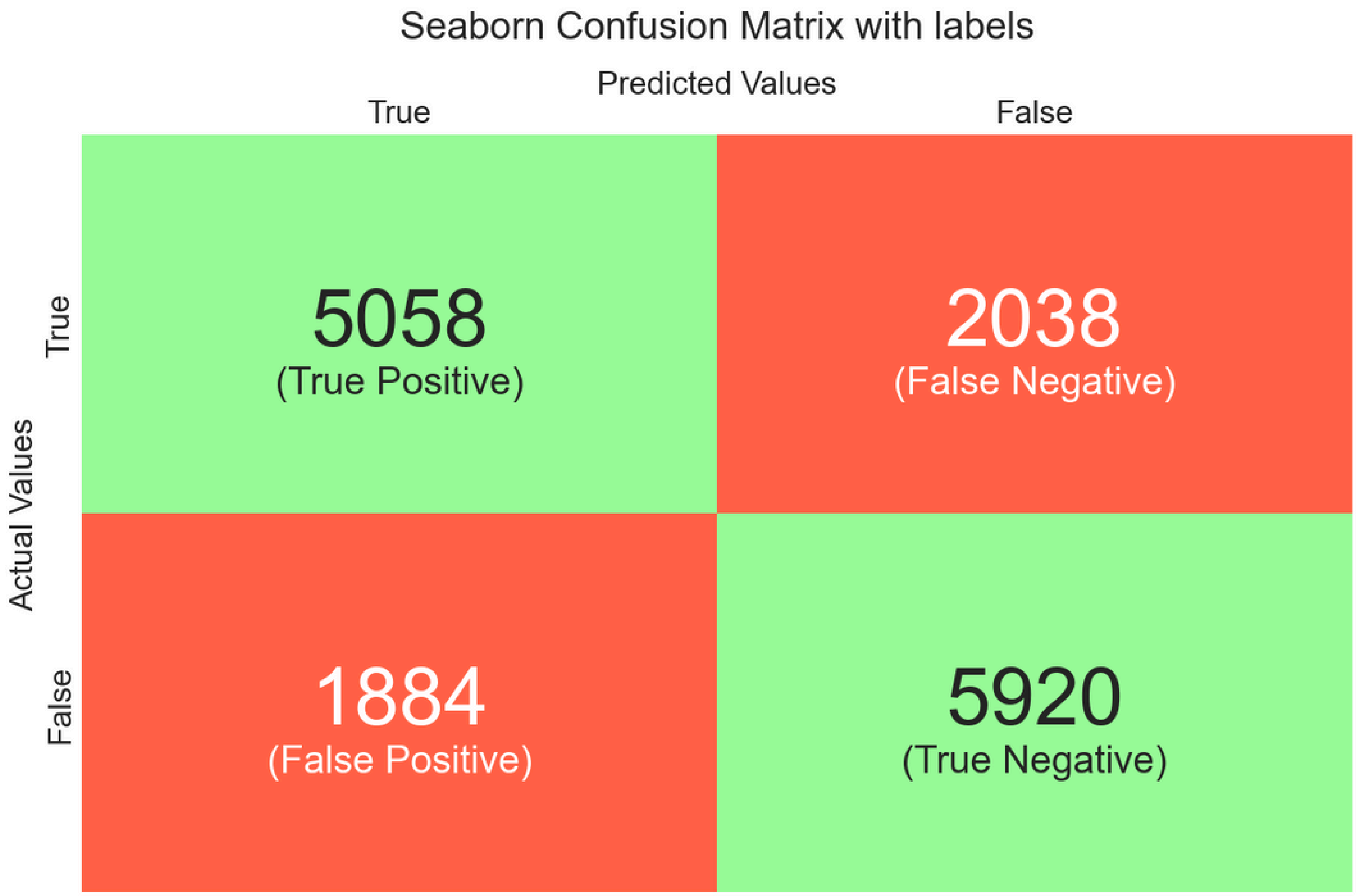
```python
model = LogisticRegression(random_state=42)

# ฝึกโมเดล
model.fit(X_train, y_train)

# ทดสอบโมเดล
y_pred = model.predict(X_test)

# ประเมินโมเดล
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

cftmt_lo = confusion_matrix(y_test, y_pred)
print(cftmt_lo)
custom_plt_confution_matrix(cftmt_lo)
```

```
Accuracy: 0.7322147651006712
              precision    recall  f1-score   support

           0       0.72      0.71      0.72      7096
           1       0.74      0.76      0.75      7804

    accuracy                           0.73     14900
   macro avg       0.73      0.73      0.73     14900
weighted avg       0.73      0.73      0.73     14900
```



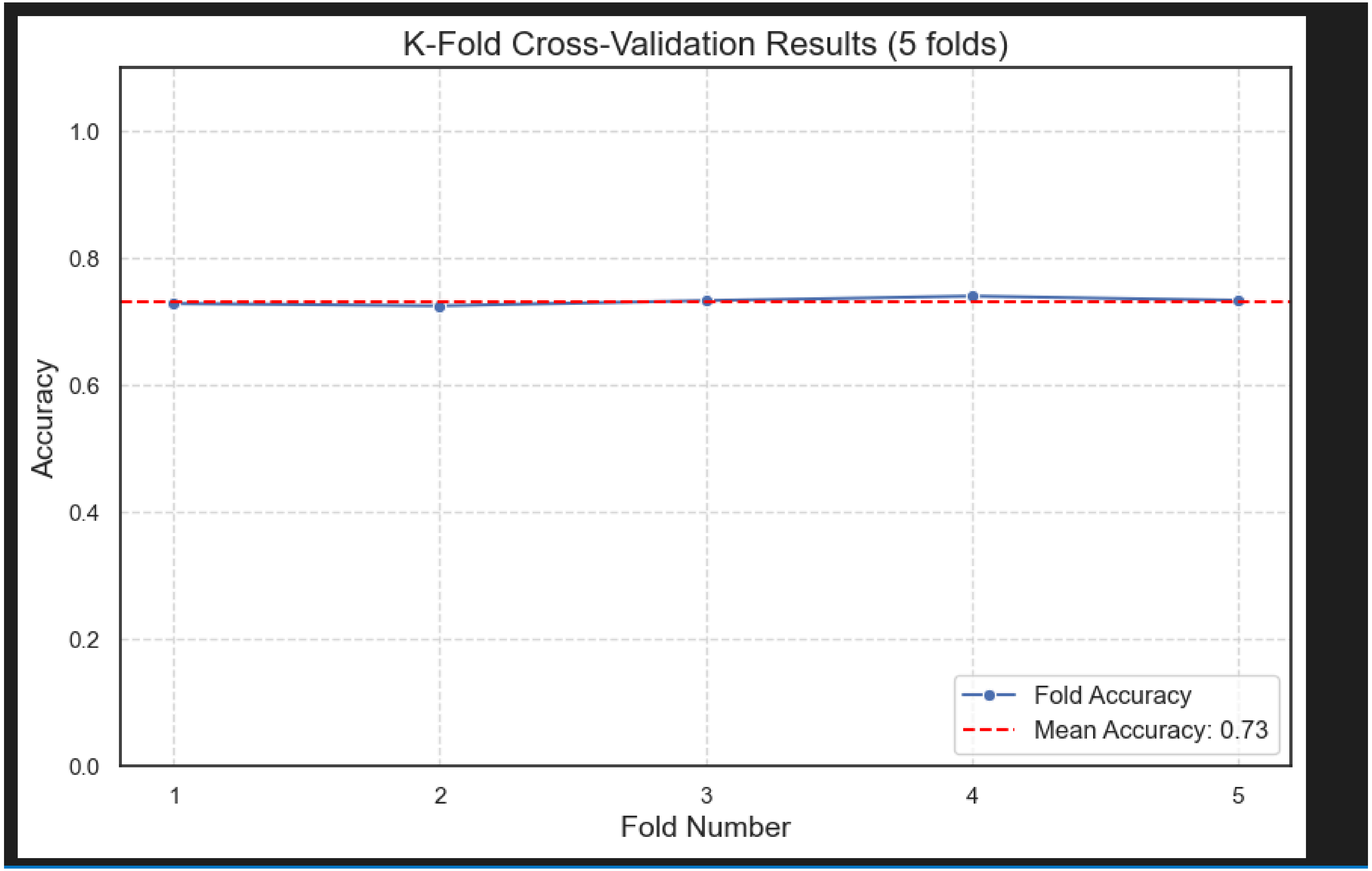
```python
gb_model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1,
 max_depth=3, random_state=42)
# เทรนโมเดลด้วยข้อมูล train
gb_model.fit(X_train, y_train)

# ทำนายผลด้วยข้อมูล test
y_pred_gb = gb_model.predict(X_test)

# ประเมินผลโมเดล
print("Accuracy:", accuracy_score(y_test, y_pred_gb))
print("\nClassification Report:\n", classification_report(y_test, y_pred_gb))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred_gb))

gBoostCunMatrix = confusion_matrix(y_test, y_pred_gb)
```

```
Accuracy: 0.7367785234899329

Classification Report:
               precision    recall  f1-score   support

           0       0.73      0.71      0.72      7096
           1       0.74      0.76      0.75      7804

    accuracy                           0.74     14900
   macro avg       0.74      0.74      0.74     14900
weighted avg       0.74      0.74      0.74     14900
```



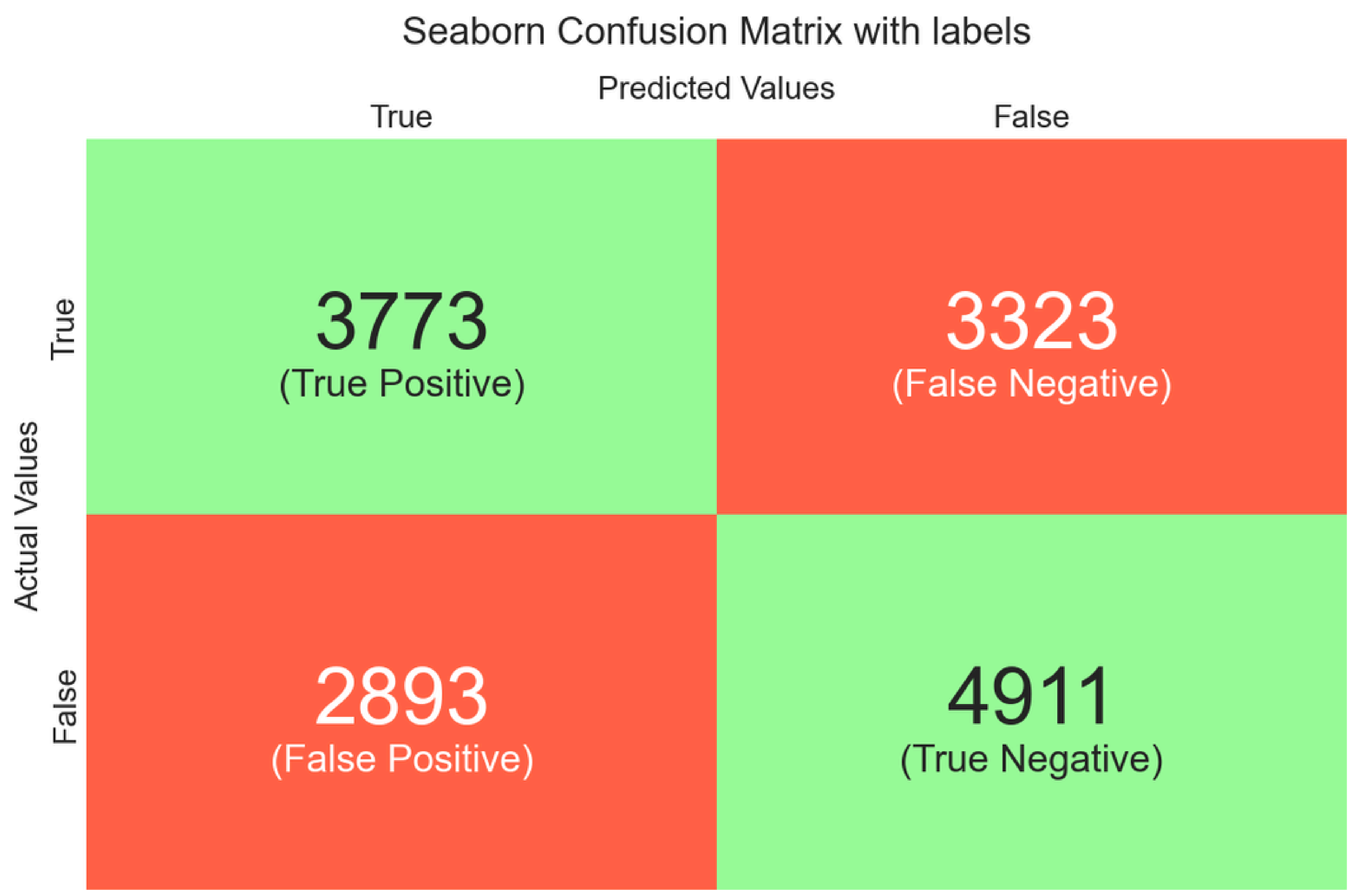
```python
k = 5

# สร้างโมเดล Gradient Boosting
gb_model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)

# สร้าง K-Fold Cross-Validator (Stratified เพื่อให้คลาสสมดุลในแต่ละ Fold)
kfold = StratifiedKFold(n_splits=k, shuffle=True, random_state=42)

# ใช้ cross_val_score ประเมินโมเดล
scores = cross_val_score(gb_model, X, y, cv=kfold, scoring='accuracy')

# แสดงผลคะแนน
print(f"K-Fold Cross-Validation Results ({k} folds):")
print(f"Scores: {scores}")
print(f"Mean Accuracy: {scores.mean():.2f}")
print(f"Standard Deviation: {scores.std():.2f}")
```

```
K-Fold Cross-Validation Results (5 folds):
Scores: [0.72865772 0.72516779 0.73328859 0.74045238 0.73380764]
Mean Accuracy: 0.73
Standard Deviation: 0.01
```


K-Fold Cross-Validation Results (5 folds)
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
Fold Number
Fold Accuracy
Mean Accuracy: 0.73

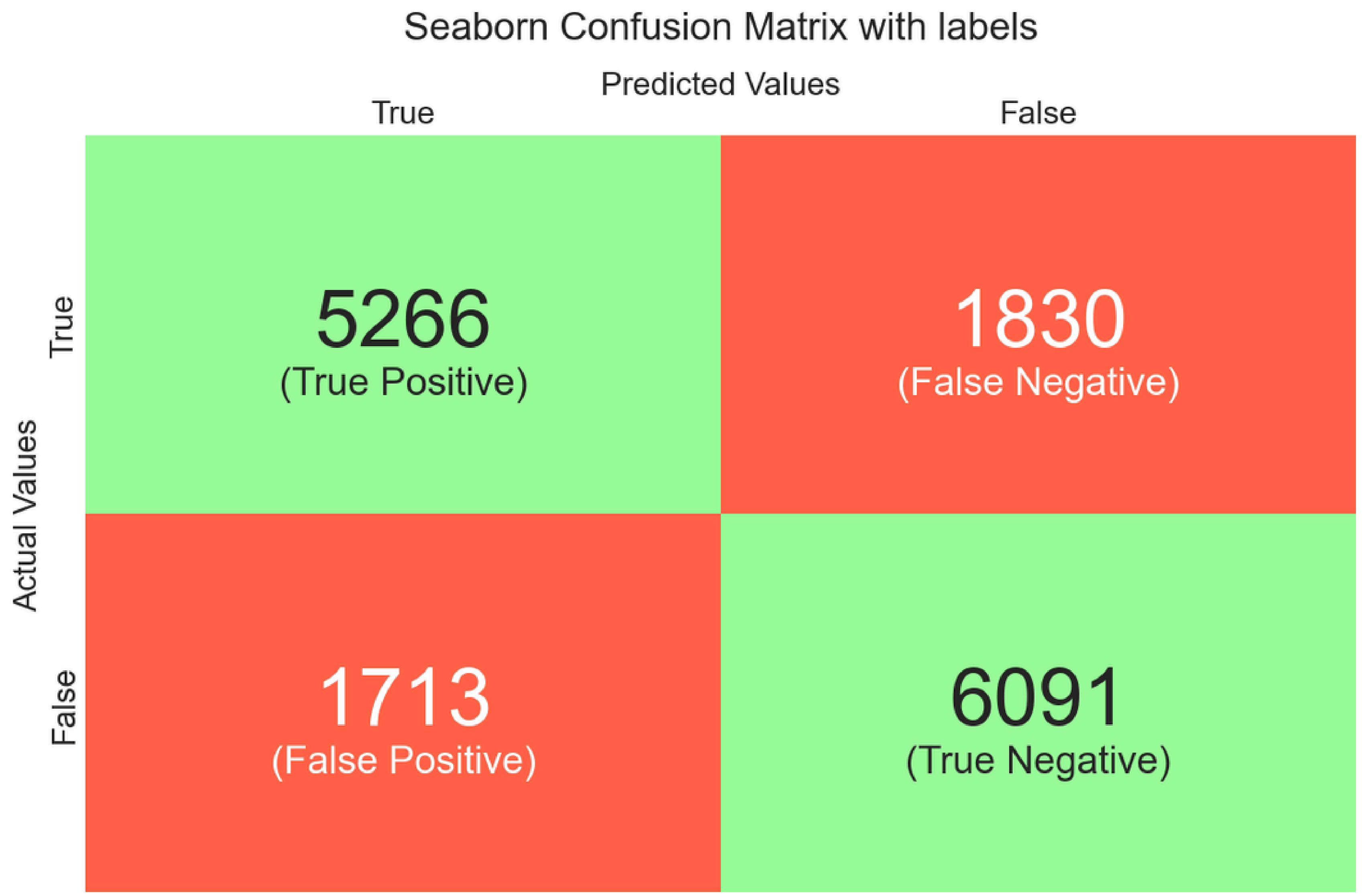
```python
model = LogisticRegression(random_state=42)

# ฝึกโมเดล
model.fit(X_train, y_train)

# ทดสอบโมเดล
y_pred = model.predict(X_test)

# ประเมินโมเดล
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
cftmt_lo = confusion_matrix(y_test, y_pred)
print(cftmt_lo)
custom_plt_confution_matrix(cftmt_lo)
```

```
Accuracy: 0.5828187919463087
              precision    recall  f1-score   support

           0       0.57      0.53      0.55      7096
           1       0.60      0.63      0.61      7804

    accuracy                           0.58     14900
   macro avg       0.58      0.58      0.58     14900
weighted avg       0.58      0.58      0.58     14900
```



```python
gb_model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3,
random_state=42)
# เทรนโมเดลด้วยข้อมูล train
gb_model.fit(X_train, y_train)

# ทำนายผลด้วยข้อมูล test
y_pred_gb = gb_model.predict(X_test)

# ประเมินผลโมเดล
print("Accuracy:", accuracy_score(y_test, y_pred_gb))
print("\nClassification Report:\n", classification_report(y_test, y_pred_gb))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred_gb))
gBoostConMatrix = confusion_matrix(y_test, y_pred_gb)
```

```
Accuracy: 0.7622147651006711

Classification Report:
               precision    recall  f1-score   support

           0       0.75      0.74      0.75      7096
           1       0.77      0.78      0.77      7804

    accuracy                           0.76     14900
   macro avg       0.76      0.76      0.76     14900
weighted avg       0.76      0.76      0.76     14900


Confusion Matrix:
 [[5266 1830]
 [1713 6091]]
```



Logistic Regression

เปรียบเทียบความแตกต่าง

เปรียบเทียบ	แบบไม่ทำอะไรเลย	แบบทำ Standardization & Drop features
Accuracy	58.2%	73.2%
Precision (0,1)	(0.57, 0.60)	(0.72, 0.74)
Recall (0,1)	(0.53, 0.63)	(0.71, 0.76)
Confusion Matrix (TP, FN, FP, TN)	(3773, 3323, 2893, 4911)	(5008, 2088, 1902, 5902)

Gradient Boosting

เปรียบเทียบความแตกต่าง

Metric	Gradient Boosting (ไม่ทำอะไรเลย)	Gradient Boosting (ทำ Feature Engineering)
Accuracy	76.2%	73.6%
Precision (0,1)	(0.75, 0.77)	(0.73, 0.74)
Recall (0,1)	(0.75, 0.78)	(0.71, 0.76)
Confusion Matrix (TP, FN, FP, TN)	(5266, 1830, 1713, 6091)	(5058, 2038, 1884, 5920)

data = data.drop((['Marital Status_Divorced', 'Marital Status_Married', 'Marital Status_Single']), axis=1)
print(data.columns)

Logistic Regression

```
gb_model.fit(X_train, y_train)

# ทำนายผลด้วยข้อมูล test
y_pred_gb = gb_model.predict(X_test)

# ประเมินผลโมเดล
print("Accuracy:", accuracy_score(y_test, y_pred_gb))
print("\nClassification Report:\n", classification_report(y_test, y_pred_gb))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred_gb))

gBoostCunMatrix = confusion_matrix(y_test, y_pred_gb)
```

[42] ✓ 7.3s Python

```
Accuracy: 0.6989261744966443

Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.69      0.69      7096
           1       0.72      0.71      0.71      7804

    accuracy                           0.70     14900
   macro avg       0.70      0.70      0.70     14900
weighted avg       0.70      0.70      0.70     14900


Confusion Matrix:
 [[4907 2189]
 [2297 5507]]
```

Gradient Boosting

```
Accuracy: 0.6989261744966443

Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.69      0.69      7096
           1       0.72      0.71      0.71      7804

    accuracy                           0.70     14900
   macro avg       0.70      0.70      0.70     14900
weighted avg       0.70      0.70      0.70     14900


Confusion Matrix:
 [[4907 2189]
 [2297 5507]]
```

หลังจากดรอป Feature ที่มีผลต่อ y มากที่สุดออก

มีค่า Accuracy: 0.6 น้อยลง

```
data = data.drop((['Gender_Female', 'Gender_Male' , 'Job Role_Education' ,
 'Job Role_Finance', 'Job Role_Healthcare', 'Job Role_Media',
       'Job Role_Technology']), axis=1)
print(data.columns)
```

```
Accuracy: 0.7331543624161074
              precision    recall  f1-score   support

           0       0.73      0.71      0.72      7096
           1       0.74      0.76      0.75      7804

    accuracy                           0.73     14900
   macro avg       0.73      0.73      0.73     14900
weighted avg       0.73      0.73      0.73     14900

[[5009 2087]
 [1889 5915]]
```

Logistic Regression

```
Accuracy: 0.7377852348993289

Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.71      0.72      7096
           1       0.74      0.76      0.75      7804

    accuracy                           0.74     14900
   macro avg       0.74      0.74      0.74     14900
weighted avg       0.74      0.74      0.74     14900


Confusion Matrix:
 [[5040 2056]
 [1851 5953]]
```

Gradient Boosting

หลังจากดรอป Feature ที่มีผลต่อ y น้อย

มีค่า Accuracy: 0.73 ไม่ต่างกันมาก

Note

1. ถ้า Correalation Matrix มีค่า 0.03 ขึ้นไปให้ใช้ค่านั้น (ว่ามีผลต่อการลาออก)
2. [1 คืออยู่, 0 คือลาออก]
3. (ตอนนี้เรา predict ว่าปัจจัยใดที่มีผลต่อการลาออกได้แล้ว) [27 Jan 2025]