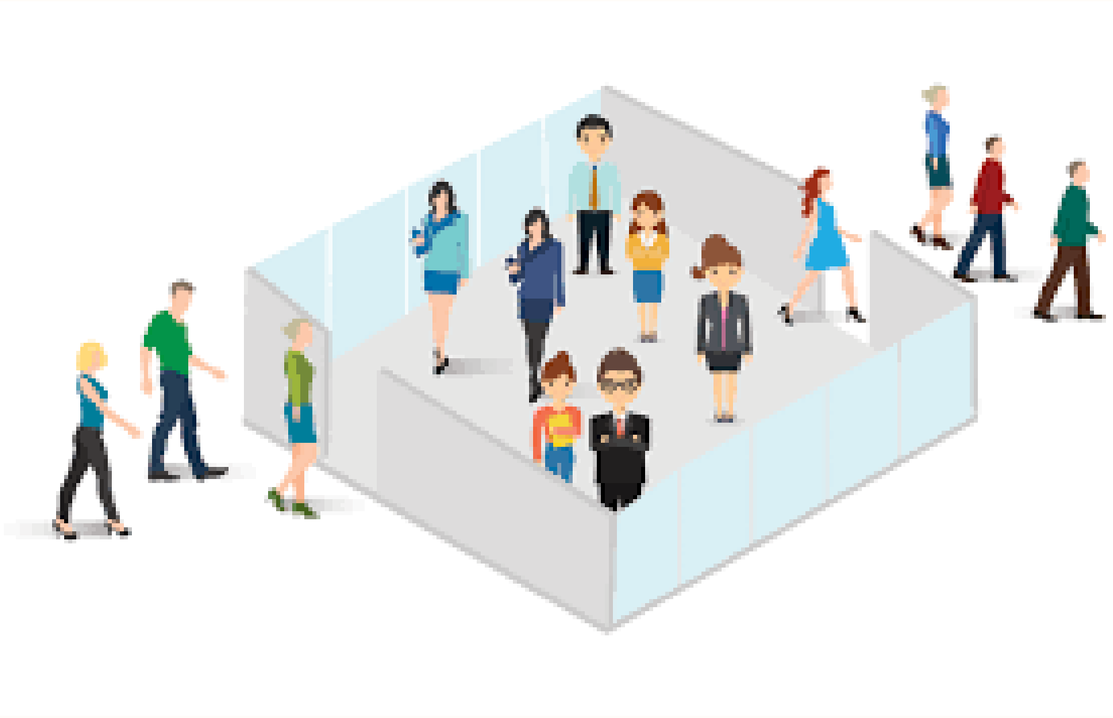
# FINAL PROJECT
# BANK CHURNERS

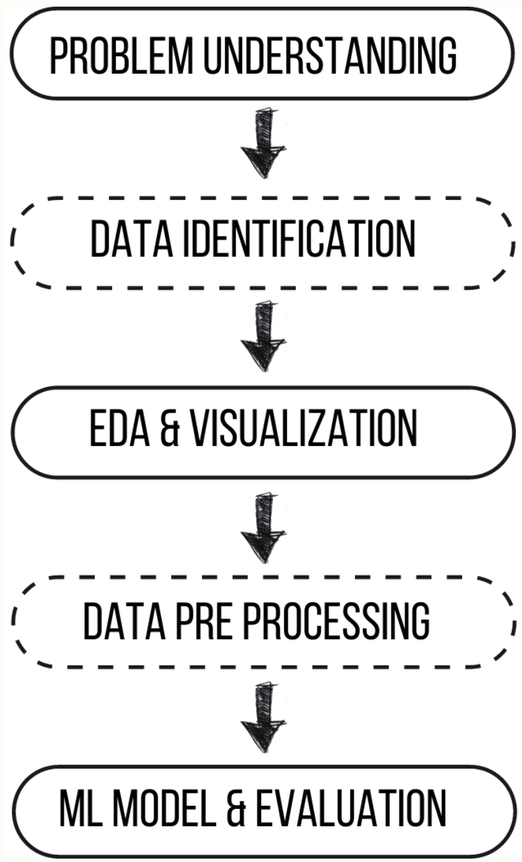


HEADER – ABDILLAH HASYIM

# INTRODUCTION

Final project diambil dari dataset pada Kaggle dengan tema Bank Churners. Pada kasus ini kita menggunakan Model Machine Learning yang berbeda untuk mengetahui model mana yang optimal untuk melakukan prediksi.

FLOW PENGERJAAN



# PROBLEM UNDERSTANDING

Seorang manajer bank merasa terganggu dengan pelanggan yang semakin banyak meninggalkan layanan kartu kredit. Manajer bank ingin mengetahui pelanggan mana saja yang akan churn sehingga pihak bank dapat melakukan pendekatan dengan pelanggan agar merubah pikiran untuk tetap menggunakan layanan kartu kredit.

Tujuan: Mengetahui model Machine Learning yang optimal untuk memprediksi pelanggan mana saja yang akan churn.

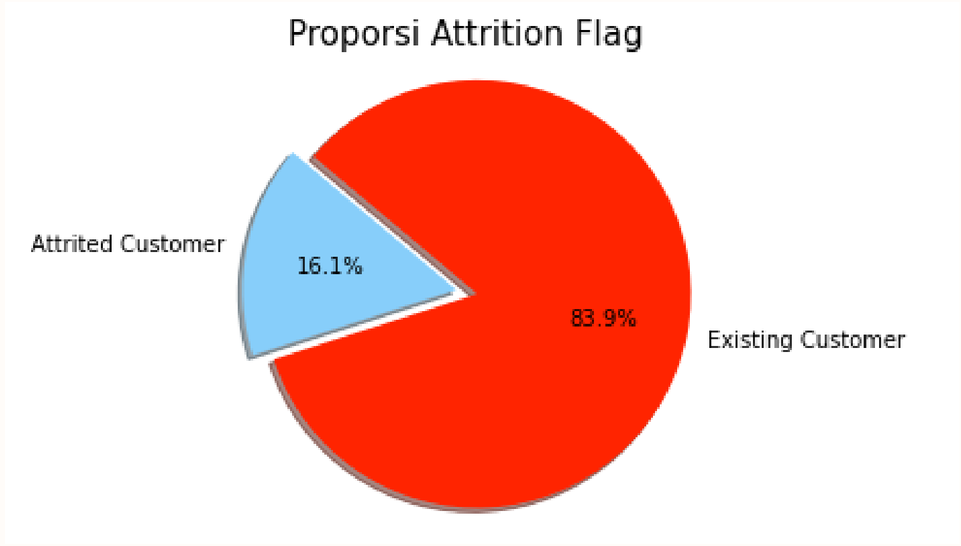
Hipotesis:
H0 = Tidak terdapat pengaruh terhadap churn rate setelah dilakukan pendekatan atau treatment terhadap pelanggan yang akan churn.
Ha = Terdapat pengaruh terhadap churn rate setelah dilakukan pendekatan atau treatment terhadap pelanggan yang akan churn.

# DATA IDENTIFICATION

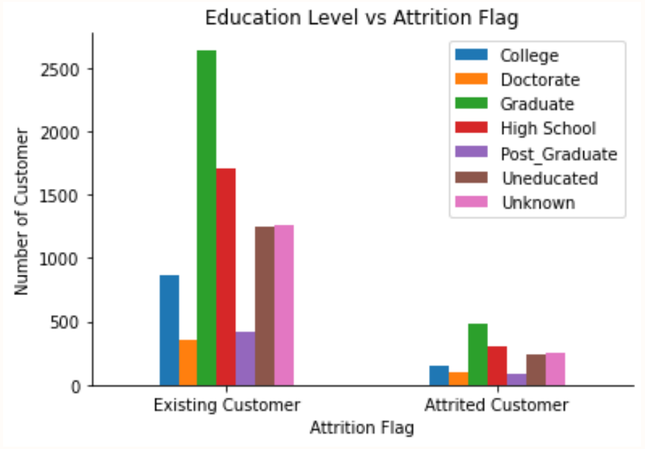
| FEATURE | DESCRIPTION |
|---|---|
| **CLIENTNUM** | Nomor ID Klien |
| **Customer_Age** | Usia Konsumen |
| **Gender** | Gender |
| **Dependent_count** | Tanggungan Dari Pengguna Kartu Kredit |
| **Education_level** | Education Level |
| **Marital_status** | Status Pernikahan |
| **Income_Category** | Income Category |
| **Card_Category** | Card Category |
| **Months_on_book** | Berapa kali transaksi tercatat dalam 1 bulan. |
| **Total_Relationship_Count** | Jumlah produk yang dimiliki pelanggan (card, akun) |
| **Months_Inactive_12_mon** | Customer yang tidak aktif selama 1 tahun |
| **Contacts_Count_12_mon** | Berapa kali pihak bank menghubungi customer dalam penawaran produk |
| **Credit_Limit** | Limit kredit |
| **Total_Revolving_Bal** | Total Revolving Balance |
| **Avg_Open_To_Buy** | Jumlah yang digunakan yang belum dilunasi |
| **Total_Amt_Chng_Q4_Q1** | Perubahan jumlah transaksi pada Q4 dan Q1 (Dalam satuan mata uang) |
| **Total_Trans_Amt** | Total transaction amount dalam 12 bulan terakhir. |
| **Total_Trans_Ct** | Total transaction count dalam 12 bulan terakhir. |
| **Total_Ct_Chng_Q4_Q1** | Perubahan jumlah transaksi pada Q4 dan Q1 (Frekuensi Transaksi) |
| **Avg_Utilization_Ratio** | Rasio rata-rata penggunaan kartu |
| **Attrition_Flag** | Status Customer (Existing & Churn |

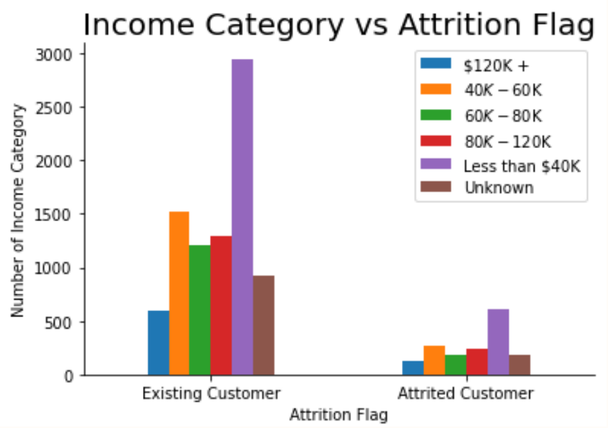
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 23 columns):
```

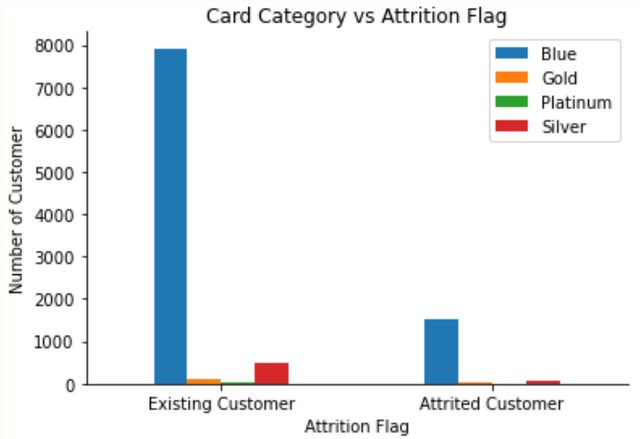
# EDA & VISUALIZATION (Categoric)

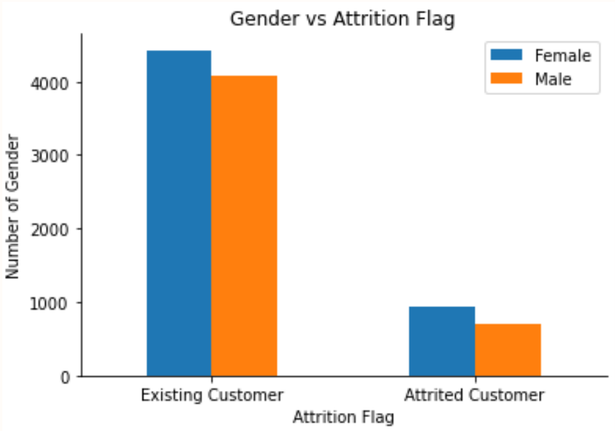



Proporsi data Imbalanced dengan perbandingan existing customer 83,9% dan attrited customer 16,1%



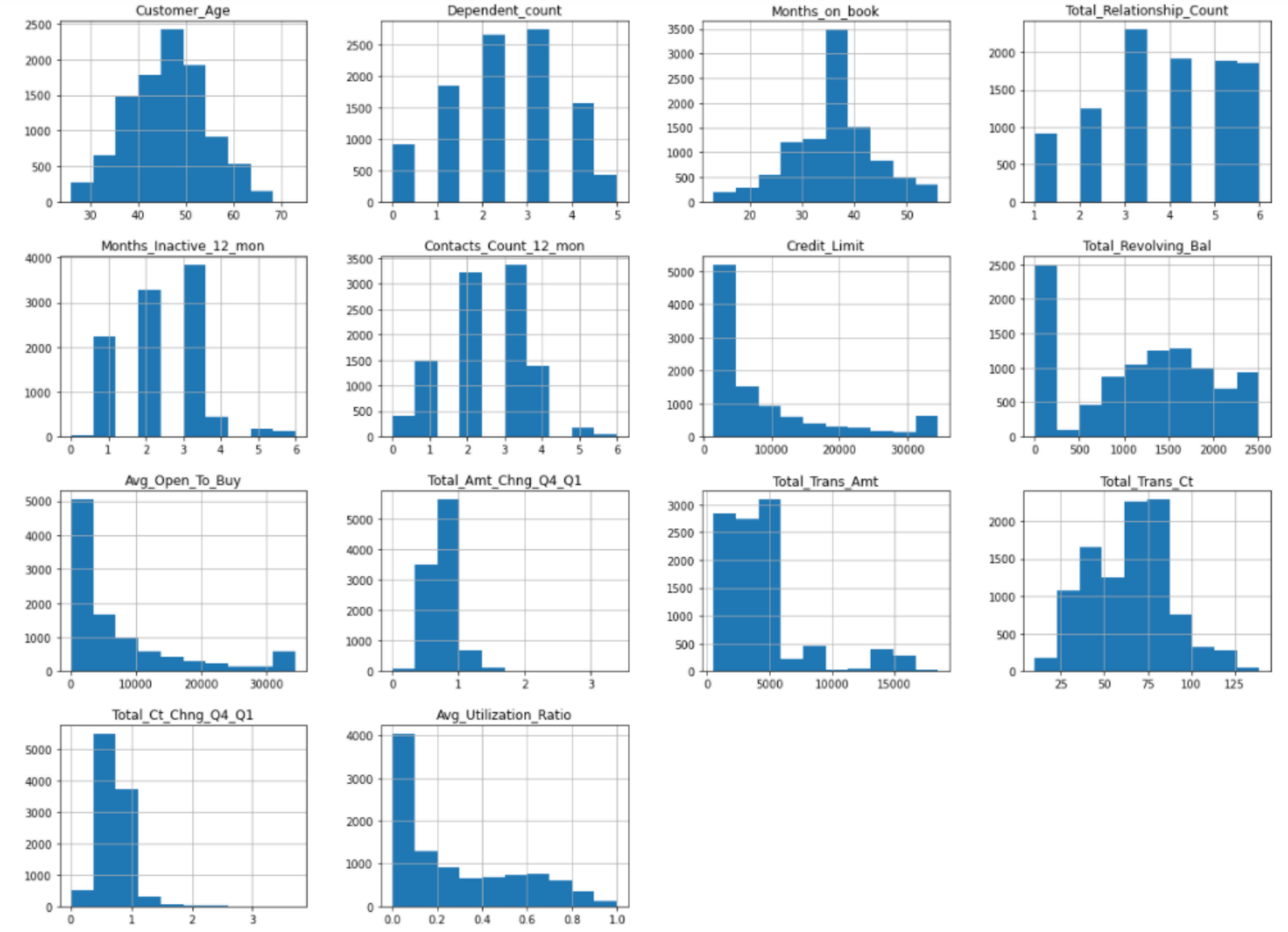




Pada education level jumlah terbanyak pada attrited customer adalah Graduate. Jumlah terbanyak attrited customer pada Income Category berada pada level Less than $40K. Jumlah terbanyak attrited customer pada Card Category berada pada kategori Blue, dan jumlah attrited customer terbanyak pada Gender Female.

# EDA & VISUALIZATION (Numeric)

## Distribution Summary

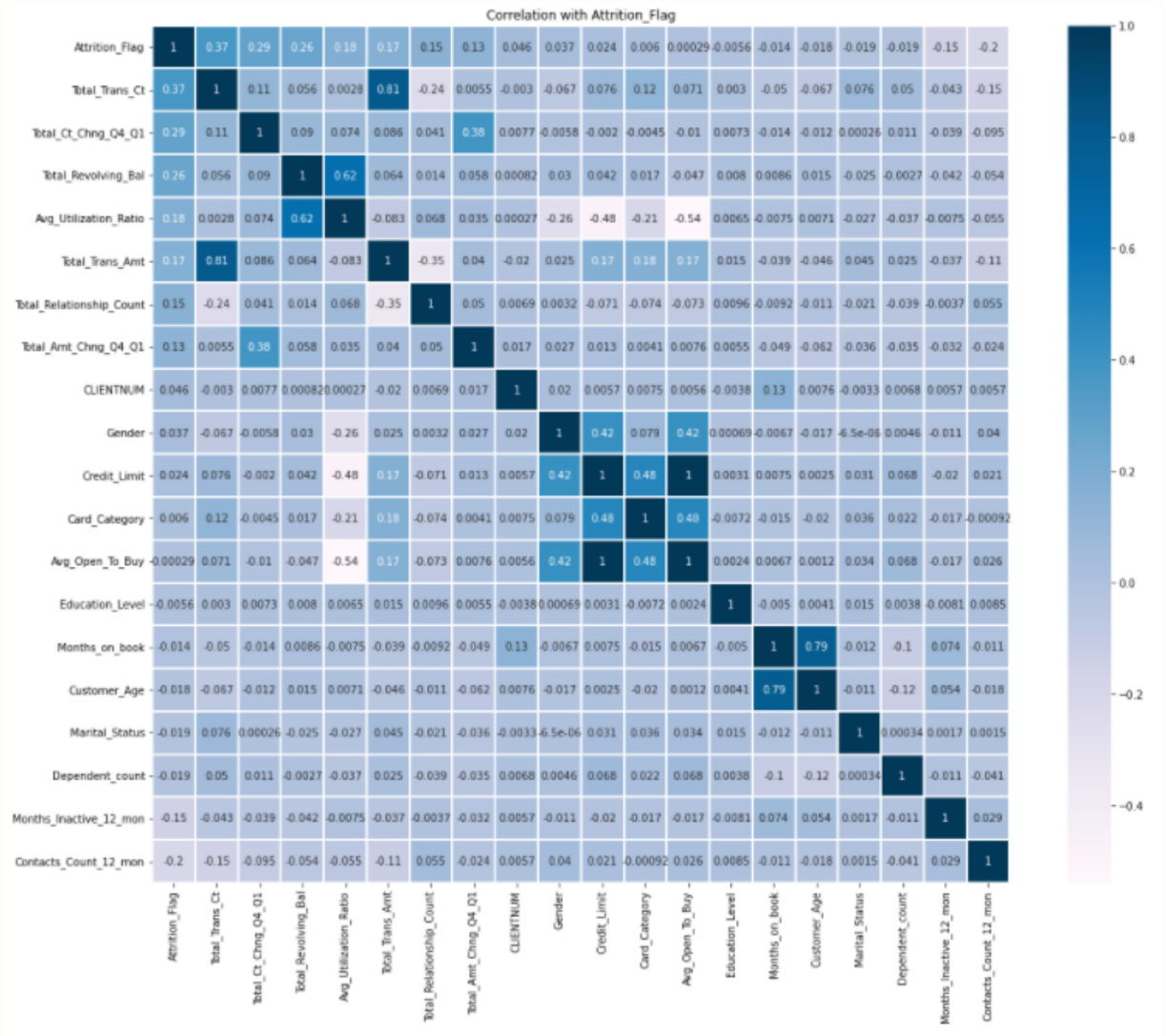



- Sebaran data pada customer age dan months on book terlihat berdistribusi normal.
- Untuk Credit Limit, Average Open to Buy dan Average Utilization Ratio terlihat data berdistribusi secara skew.

# DATA PRE-PROCESSING

- Tidak terdapat Null Values.
- Tidak terdapat duplicate value.
- Melakukan label encoder terhadap Attrition Flag, Education Level, Card Category, Gender dan Marital Status.

# CORRELATION MATRIX




Pada correlation matrix terdapat gejala multikolinearitas yaitu antara feature Credit Limit dan Average Open to Buy karena memiliki nilai r = 1.

# FEATURE SELECTION

## SelectKBest

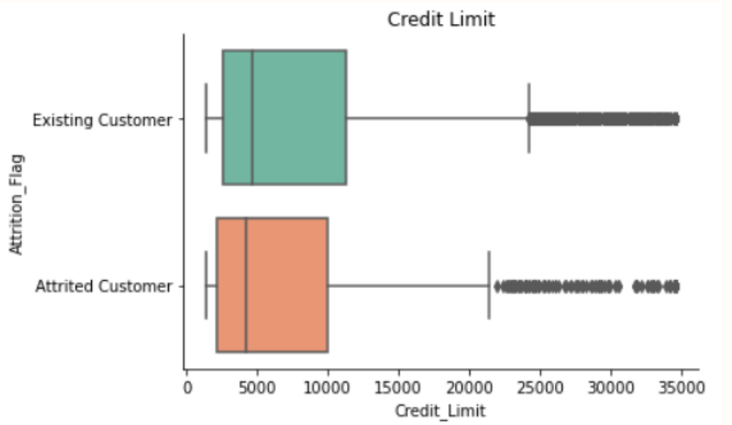
```
fs = SelectKBest(score_func=chi2, k=4)

X = df_read.iloc[:, 9:]
y = df_read['Attrition_Flag']

X_selected = fs.fit_transform(X, y)
print(X_selected.shape)
```

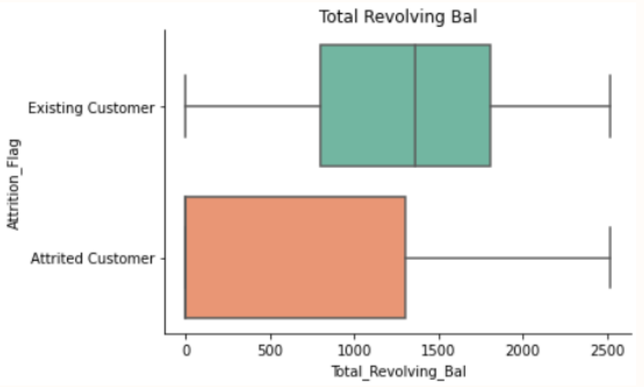
```
(10127, 4)
```

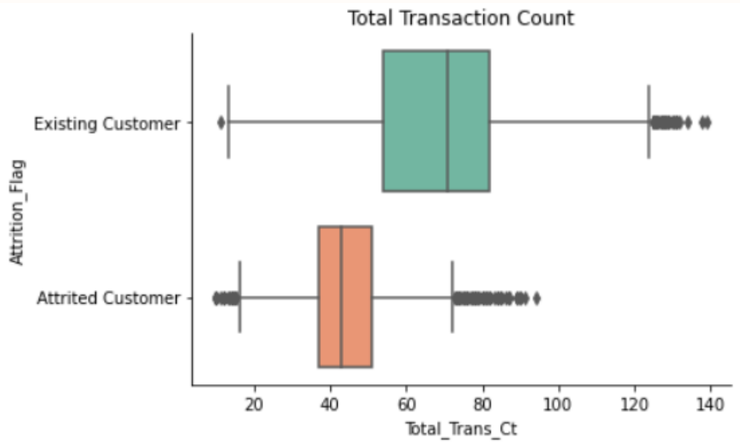
```
fs.get_feature_names_out()
```

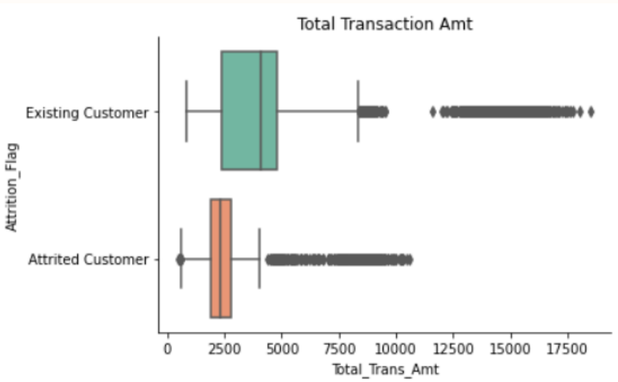
```
array(['Credit_Limit', 'Total_Revolving_Bal', 'Total_Trans_Amt',
       'Total_Trans_Ct'], dtype=object)
```

Feature Importance yang didapat dengan metode SelectKBest yaitu Credit Limit, Total Revolving Balance, Total Transaction Amount dan Total Transaction Count



# MACHINE LEARNING MODELLING

Pemisahan antara variabel X dan variabel Y → Feature Scaling pada Variabel X → Train Test Split → Machine Learning Model & Evaluation

Dividing Predictor and Target

```
X = df_read.iloc[:, 9:]
y = df_read['Attrition_Flag']
```

Scaling with StandardScaler

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Train Test Split

```
from sklearn.model_selection import StratifiedKFold, train_test_split, RandomizedSearchCV
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=500, stratify = y)

print('Proportional class distribution in train data:\n', y_train.value_counts() / y_train.count(), '\n')
print('Proportional class distribution in test data:\n', y_test.value_counts() / y_test.count())
```

```
Proportional class distribution in train data:
 1    0.839306
0    0.160694
Name: Attrition_Flag, dtype: float64

Proportional class distribution in test data:
 1    0.839421
0    0.160579
Name: Attrition_Flag, dtype: float64
```

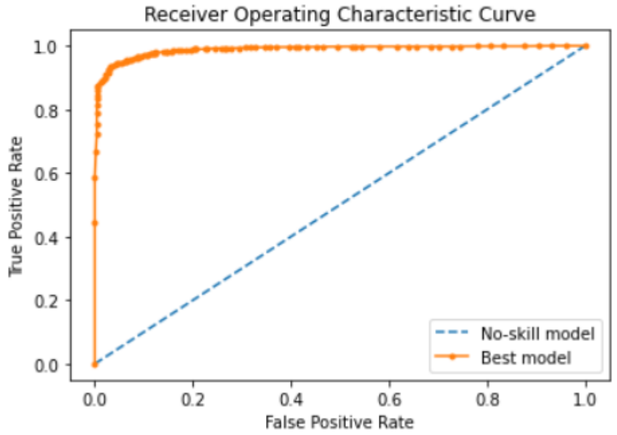
Comparing Random Forest & Logistic Regression

```
# Comparing Random Forest and Logistic Regression
## Modeling
rf_model = RandomForestClassifier(random_state = 500)
logr_model = LogisticRegression(random_state = 500)
```

# MACHINE LEARNING EVALUATION

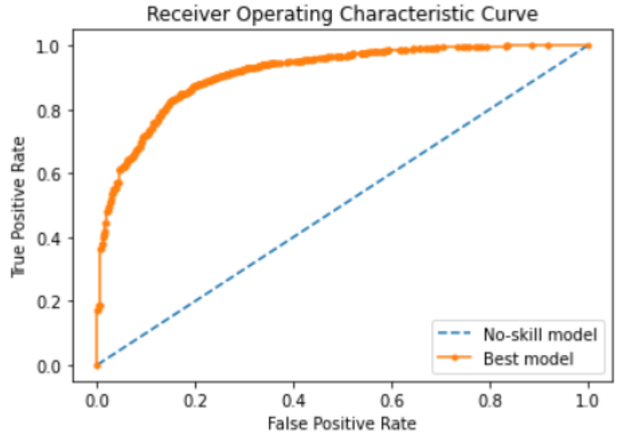
## Model Random Forest

```
Random Forest Accuracy: 0.9605133267522211
Random Forest Precision: 0.8319672131147541
Random Forest Recall: 0.9144144144144144
Random Forest F-Score: 0.871244635193133
Random Forest AUC: 0.9883844972399124
```




## Model Logistic Regression

```
Logistic Regression Accuracy: 0.8927278710102007
Logistic Regression Precision: 0.5409836065573771
Logistic Regression Recall: 0.7213114754098361
Logistic Regression F-Score: 0.6182669789227166
Logistic Regression AUC: 0.913985033191741
```



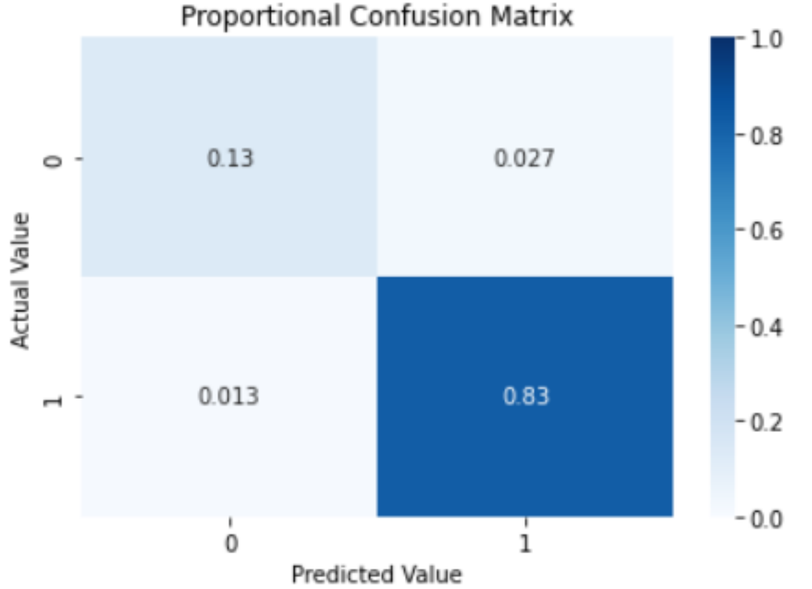

| MODEL | ACCURACY | PRECISION | RECALL |
|---|---|---|---|
| RANDOM FOREST | 96,05 | 83,20 | 91,44 |
| LOGISTIC REGRESSION | 89,27 | 54,10 | 72,13 |

- Random Forest menunjukkan hasil evaluasi yang lebih baik dibandingkan Logistic Regression.
- Accuracy tidak menjadi focus.
- Fokus: Recall

# MACHINE LEARNING EVALUATION

Model Random Forest

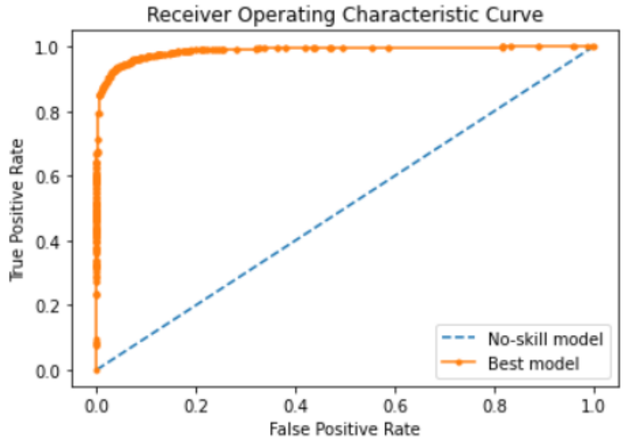
```
Random Forest Confusion Matrix:
[[ 406   82]
 [  38 2513]]
```




- Nilai TP = 406
- Nilai TN = 2513
- Nilai FN = 82
- Nilai FP = 38

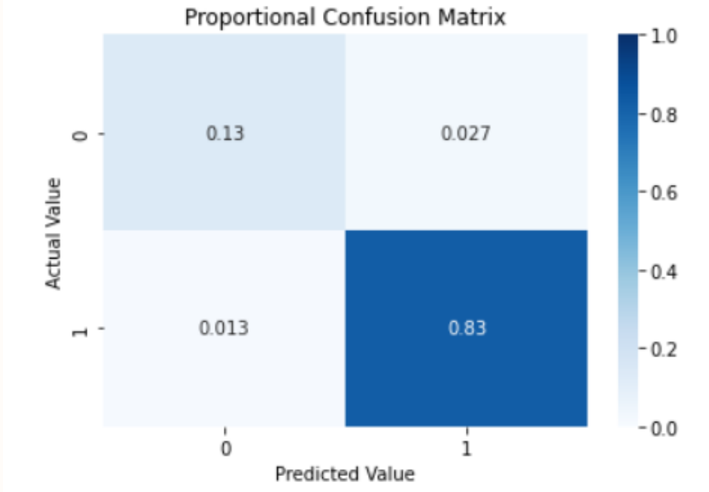
# HYPERPARAMETER TUNING RANDOMIZED SEARCH

```
Accuracy: 0.9605133267522211
F-Score: 0.8728813559322034
AUC: 0.9880655930493346
Precision: 0.8442622950819673
Recall: 0.9035087719298246
```



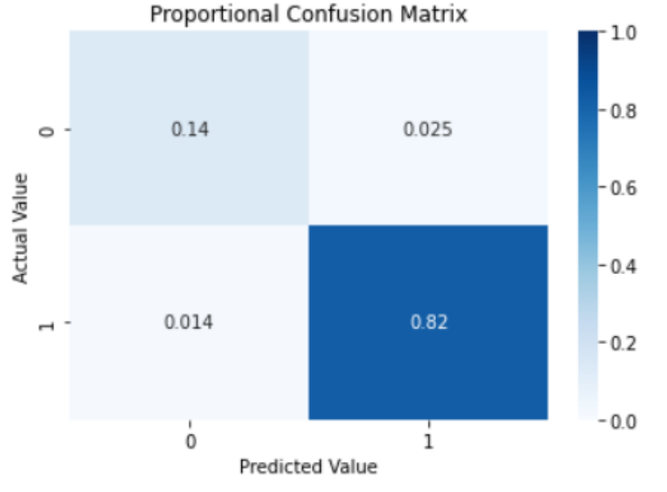

## BEFORE

```
Random Forest Confusion Matrix:
[[ 406   82]
 [  38 2513]]
```




## AFTER

```
Confusion Matrix:
[[ 412   76]
 [  44 2507]]
```




| RANDOM FOREST | ACCURACY | PRECISION | RECALL |
|---|---|---|---|
| BEFORE | 96,05 | 83,20 | 91,44 |
| AFTER | 96,05 | 84,42 | 90,35 |

- Nilai TP = 406
- Nilai TN = 2513
- Nilai FP = 82
- Nilai FN = 38

- Nilai TP = 412
- Nilai TN = 2507
- Nilai FP = 76
- Nilai FN = 44

BEST PARAMETER: n_estimator = 100, min_sample_split = 4, min_samples_leaf = 1, max_features = 0.7, max_depth = 10

# KESIMPULAN

- Model Machine Learning yang digunakan dalam memprediksi customer mana yang akan churn adalah Model Random Forest karena memiliki nilai Accuracy sebesar 96,05%, nilai Precision sebesar 83,20%, dan Recall sebesar 91,4%. Jauh lebih baik dibandingkan dengan Logistic Regression yang hanya memiliki Accuracy sebesar 89,27%, Precision 54,10% dan Recall 72,13%.
- Perlu dilakukan optimasi dengan hyperparameter tuning-RandomizedSearchCV agar model Random Forest Classifier lebih optimal dalam memprediksi customer yang churn.
- Fokus parameter evaluation terletak pada recall. Dengan recall maka customer yang churn lebih optimal untuk diklasifikasi.
- Meskipun menggunakan recall ada kemungkinan existing customer dapat diklasifikasikan sebagai customer yang churn, ini tidak mengapa. Karena pihak bank akan tetap memberikan service yang optimal, yang dapat mengakibatkan meningkatnya loyalitas dari existing customer.

# TERIMAKASIH