Final Project

Credit Card Customer Analysis

By Mutual Team

Business and Data Understanding

Credit Bank Customer

Deskripsi Dataset

Source : Kaggle

Tantangan Manajer Bank adalah untuk menganalis pelanggan yang akan churn/ meninggalkan layanan kartu kredit dari banknya.

Tujuan untuk menganalisis dan membuat modeling untuk mengetahui pelanggan yang akan churn/ masih aktif bertransaksi menggunakan kartu kredit.

Pendekatan Analisis Credit Bank Customer

Descriptive Analytics

Banyak pelanggan yang tidak aktif dalam bertransaksi menggunakan kartu kredit pada bank

Diagnostic Analytics

Banyak pelanggan yang mungkin kurang puas dengan pelayanan dari bank dan juga penawaran yang ditawarkan kurang sesuai dengan keinginan pelanggan. Selain itu, Bank juga mungkin memiliki masalah keuangan sehingga pelanggan menjadi kurang puas dengan pelayanan bank

Predictive Analytics

Terdapat 2 kemungkinan yang didapatkan yaitu pelanggan akan tetap berlangganan dan juga pelanggan berhenti berlangganan dari bank tersebut

Prescriptive Analytics

model prediksi dibuat dengan melakukan pre-processing, dan juga menggunakan model machine learning pipeline yang dirasa dapat memberikan prediksi yang maksimal

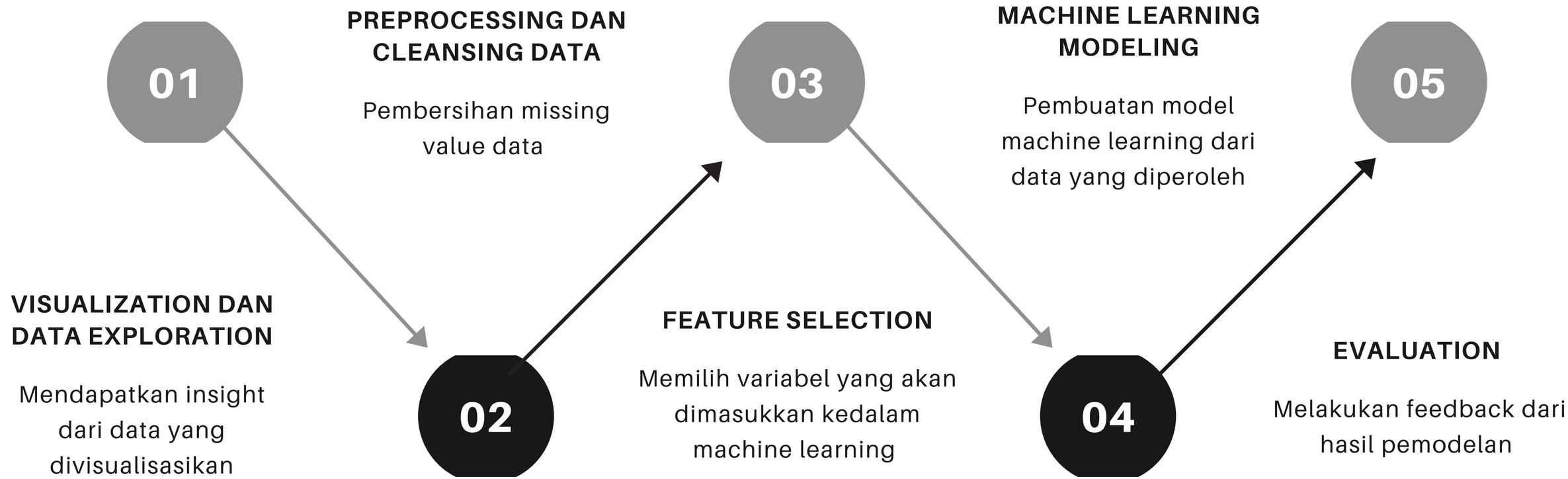
Deskripsi Variabel

Pada Dataset Credit Card Customer Analysis terdapat 23 kolom: 8 Kolom Demografi, 6 Kolom Balance, 4 kolom Data Transaksi, dll

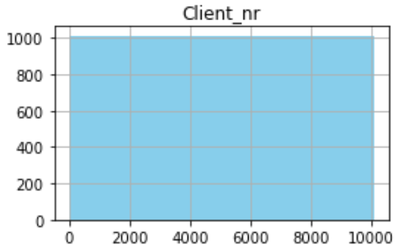
No	Nama Kolom	Tipe Data	Skala	Deskripsi
1	Clientnum	Integer	Ratio	Nomor unik ID pelanggan
2	Attrition_Flag	String	Nominal	0 = Tidak aktif 1 = aktif
3	Customer_Age	Integer	Ratio	Umur Pelanggan
4	Gender	String	Ordinal	F = Female M = Male
5	Dependent_count	Integer	Nominal	Jumlah tanggungan yang dimiliki
6	Education_Level	String	Ordinal	Kualifikasi Pendidikan
7	Marital_Status	String	Nominal	Married, Single, Divorced, Unknown
8	Months_on_book	Integer	Ratio	Periode dalam bulan yang berkaitan dengan bank
9	Months_Inactive_12_mon	Integer	Ratio	Jumlah bulan yang sudah tidak aktif menggunakan kartu kredit dalam 12 bulan terakhir

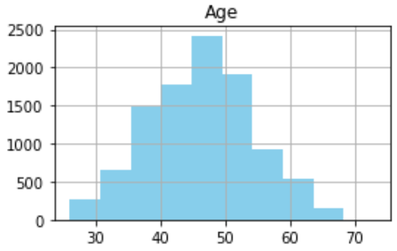
Activities that should be done

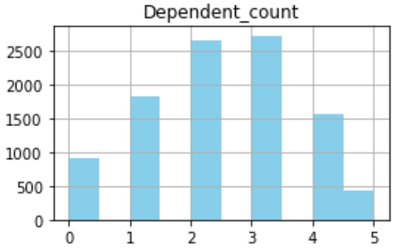
STEP ACTIVITIES SHOULD BE DONE

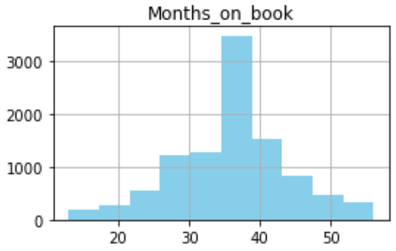


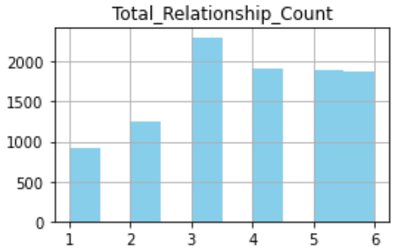
Exploration and Data Visualization + Feature Selection

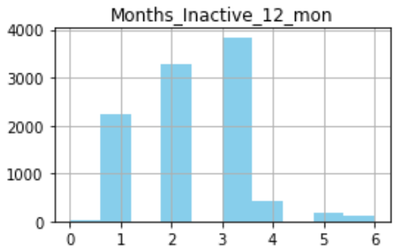


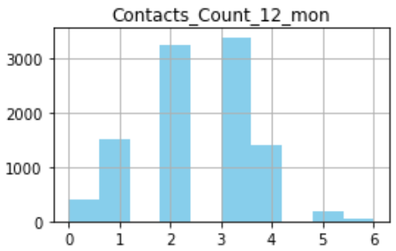


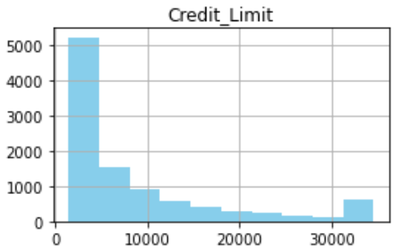


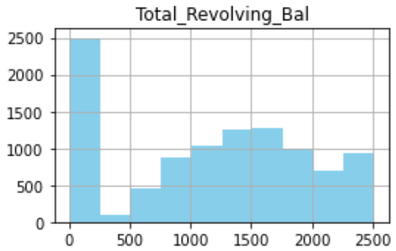


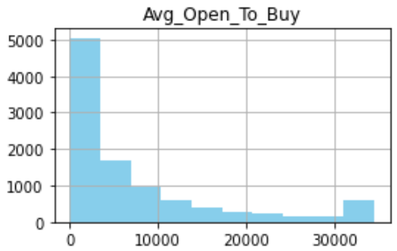


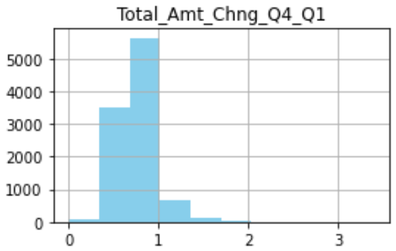


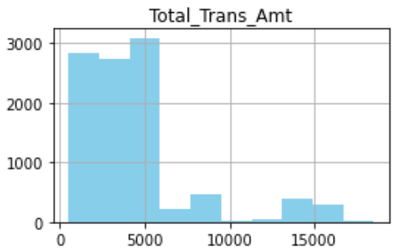


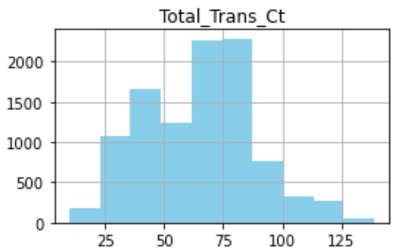


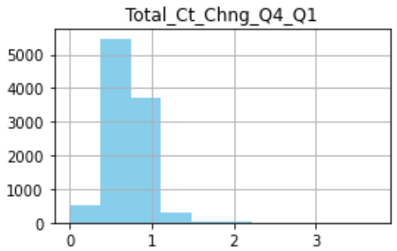


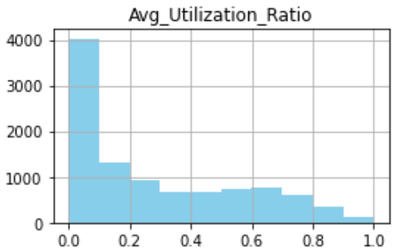


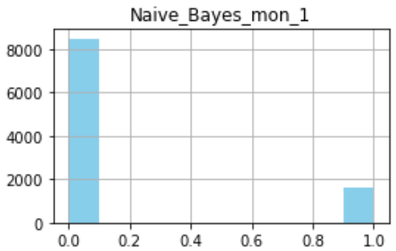


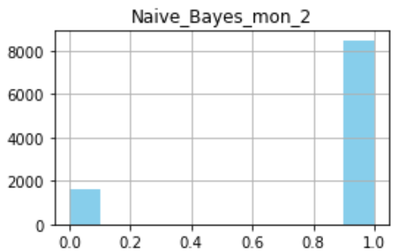












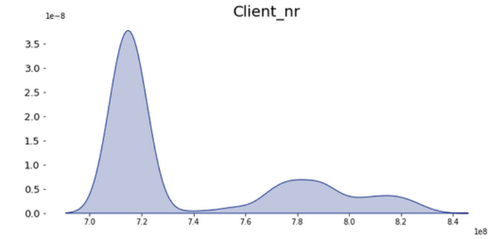
Histogram dari setiap variabel

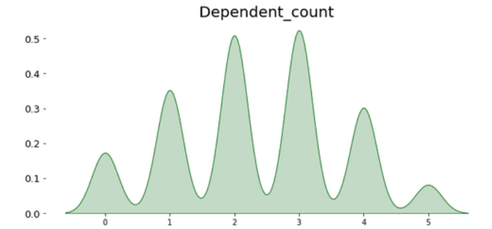
- Age, dependent count dan Months on book, memiliki distribusi data yang mendekati simetris
- Credit limit, Avg open to buy, Avg utilization ratio memiliki distribusi data yang skew ke kanan
- Variabel yang lain memiliki distribusi data yang tidak simetris

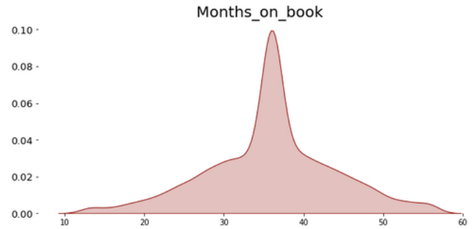
Distribusi dari setiap variabel

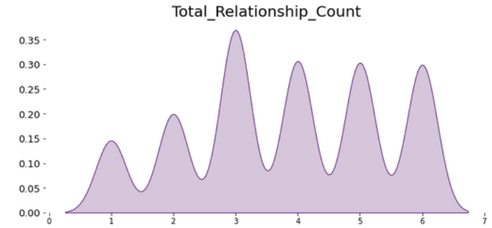
Pada KDE plot dapat dilihat simetris datanya tidak jauh berbeda dari histogram plot

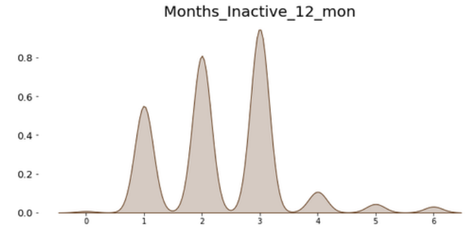
- Age dan months on book memiliki Data yang Simetris
- Variabel yang lain memiliki data yang tidak simetris

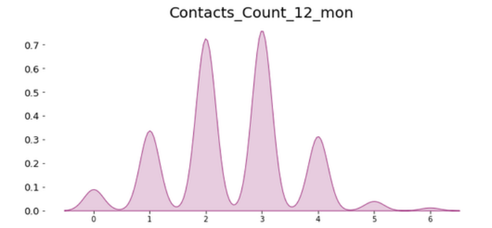


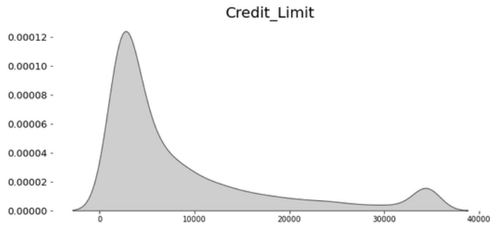


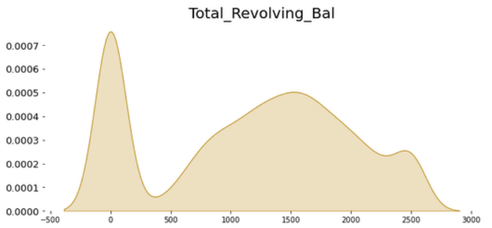


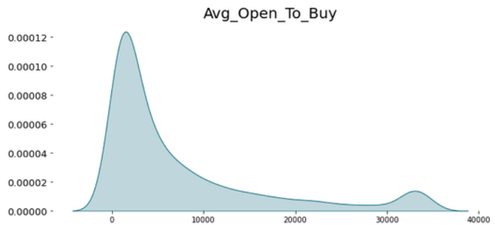


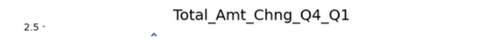


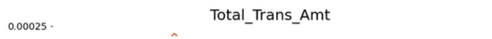


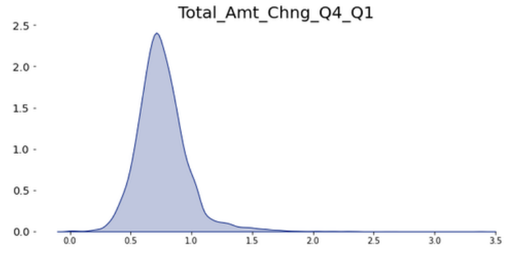


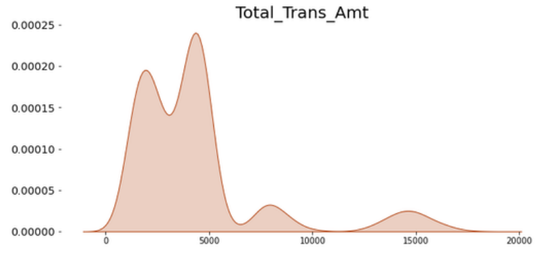


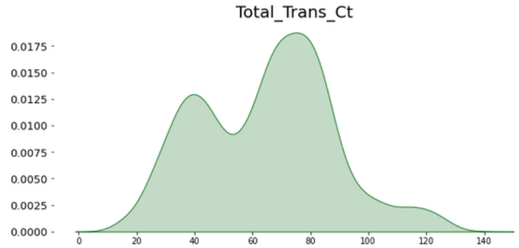


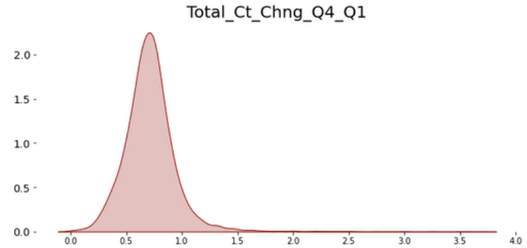




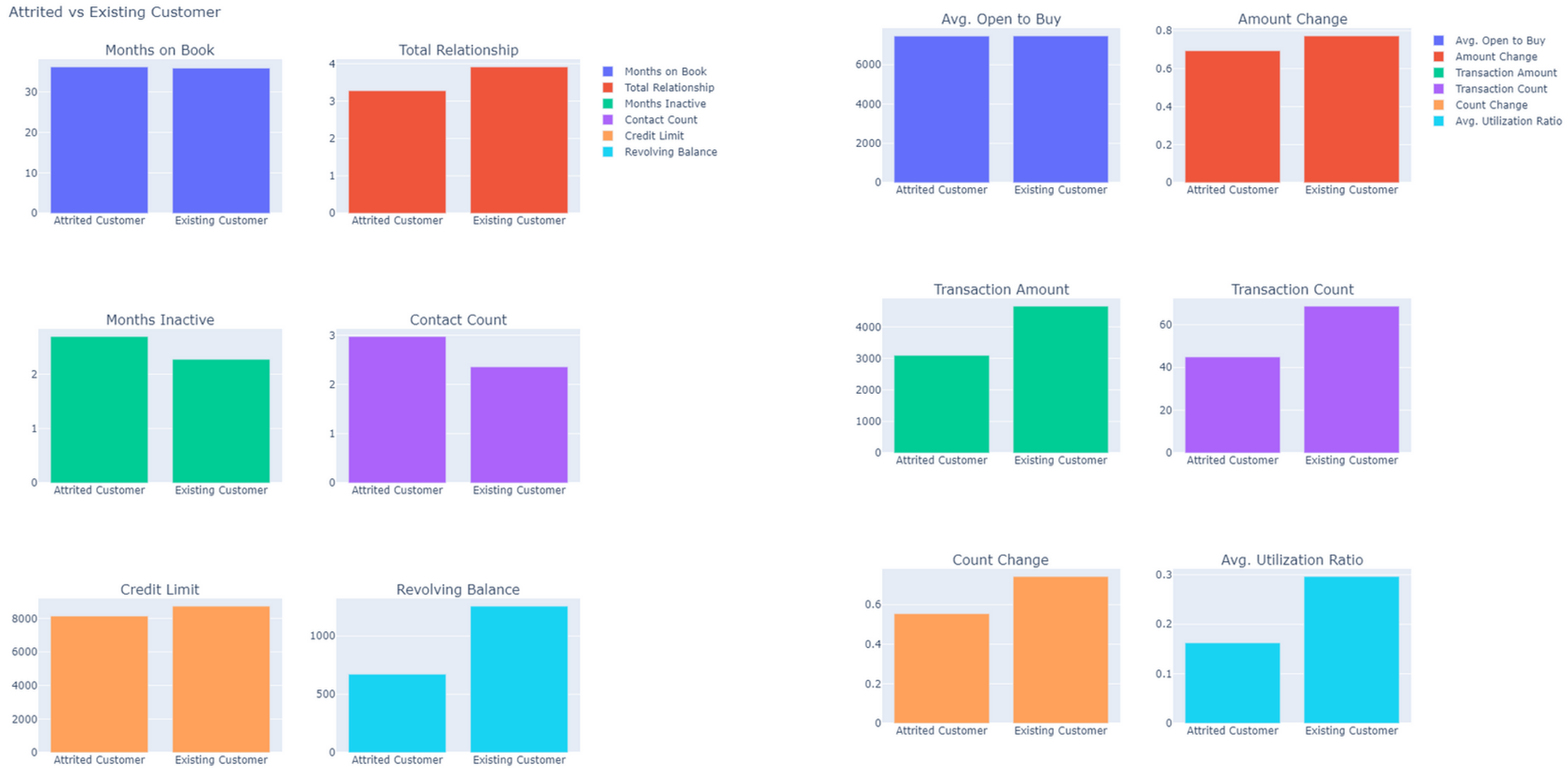






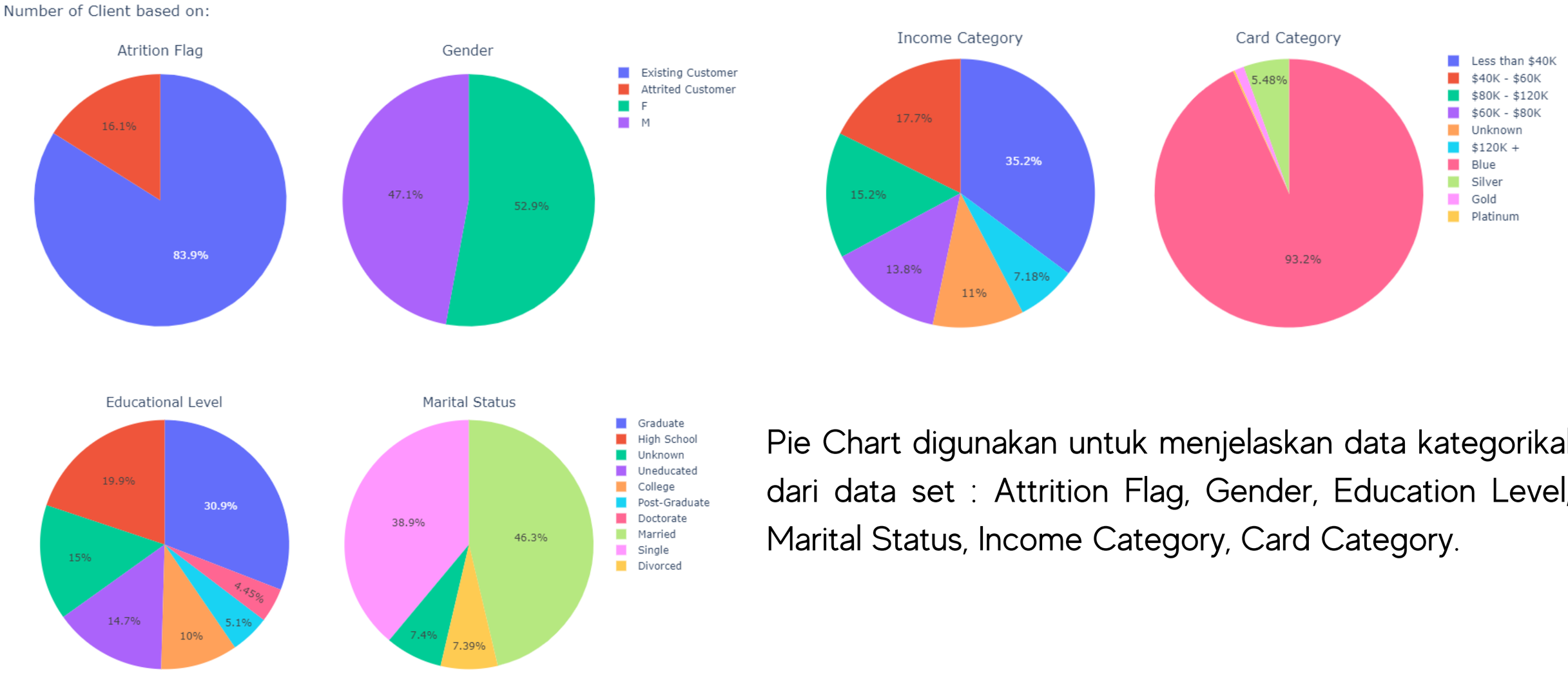


Plot dari variabel pada dataframe



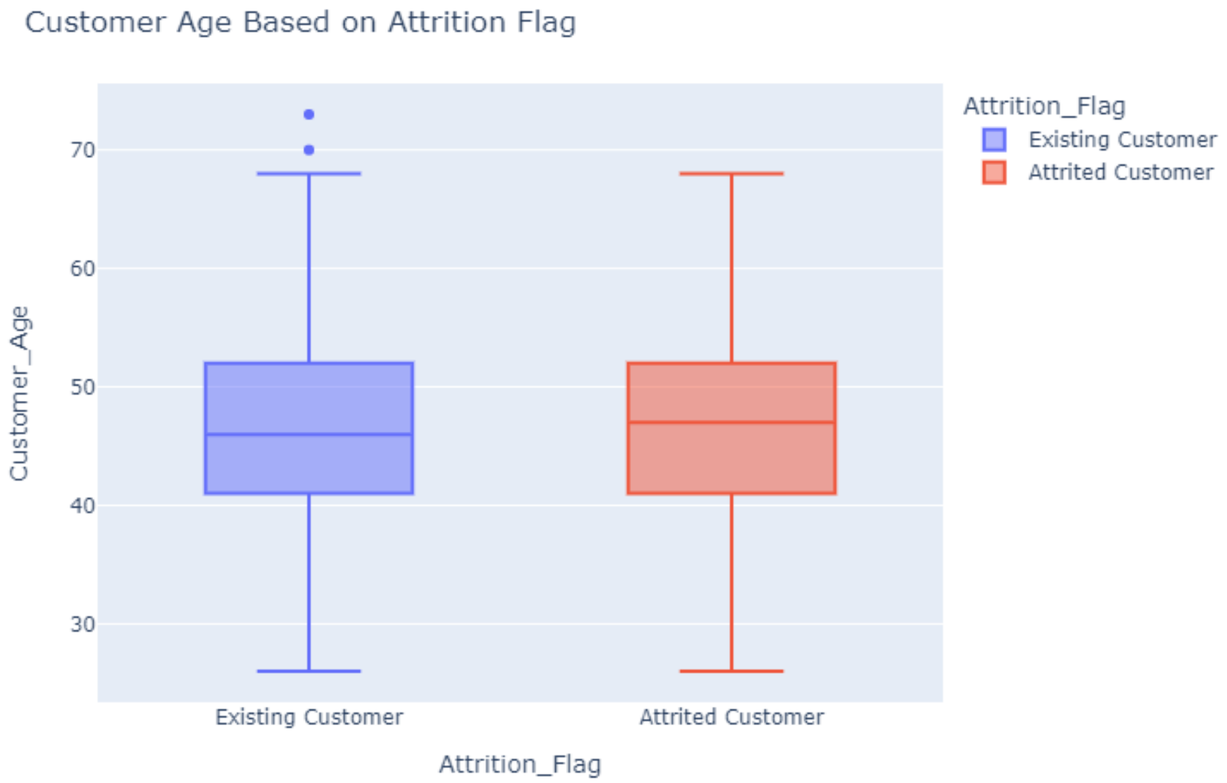
- Bar Chart digunakan untuk data Numerik dan terdapat beberapa feature yang memiliki perbedaan signifikan yaitu: Revolving Balanced, Transaction Amount, Transaction Count, Count Change, dan Avg_Utilation Ratio

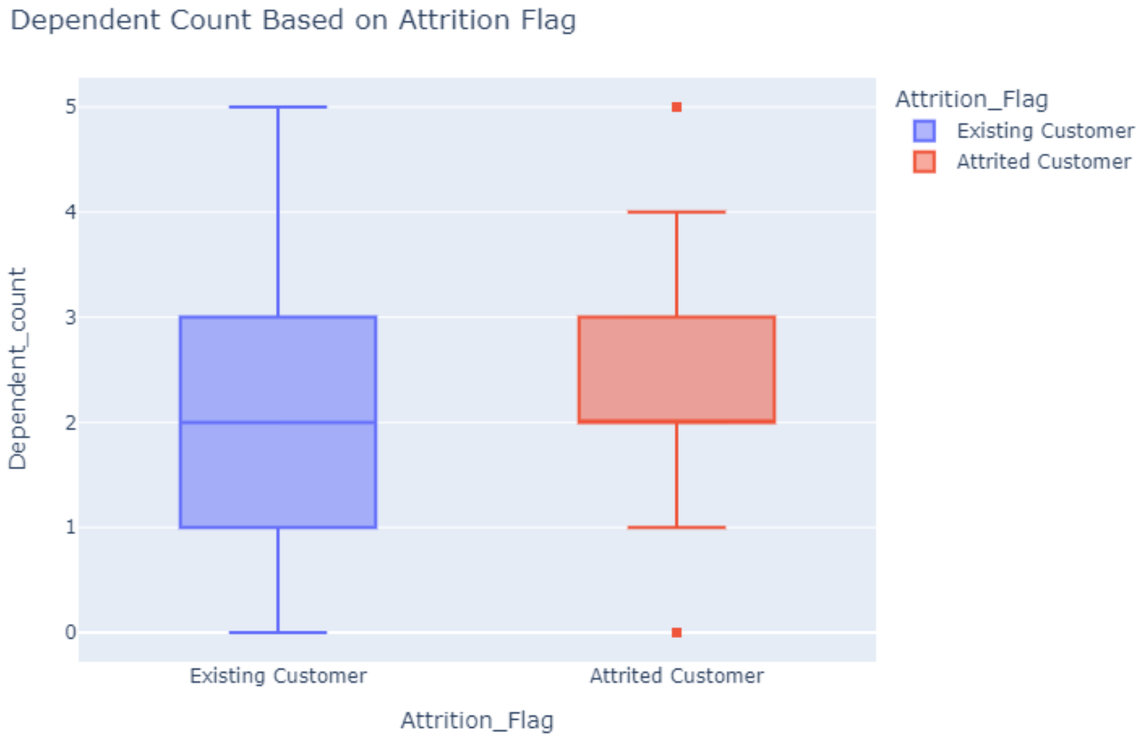
Plot dari variabel pada dataframe



Pie Chart digunakan untuk menjelaskan data kategorikal dari data set : Attrition Flag, Gender, Education Level, Marital Status, Income Category, Card Category.

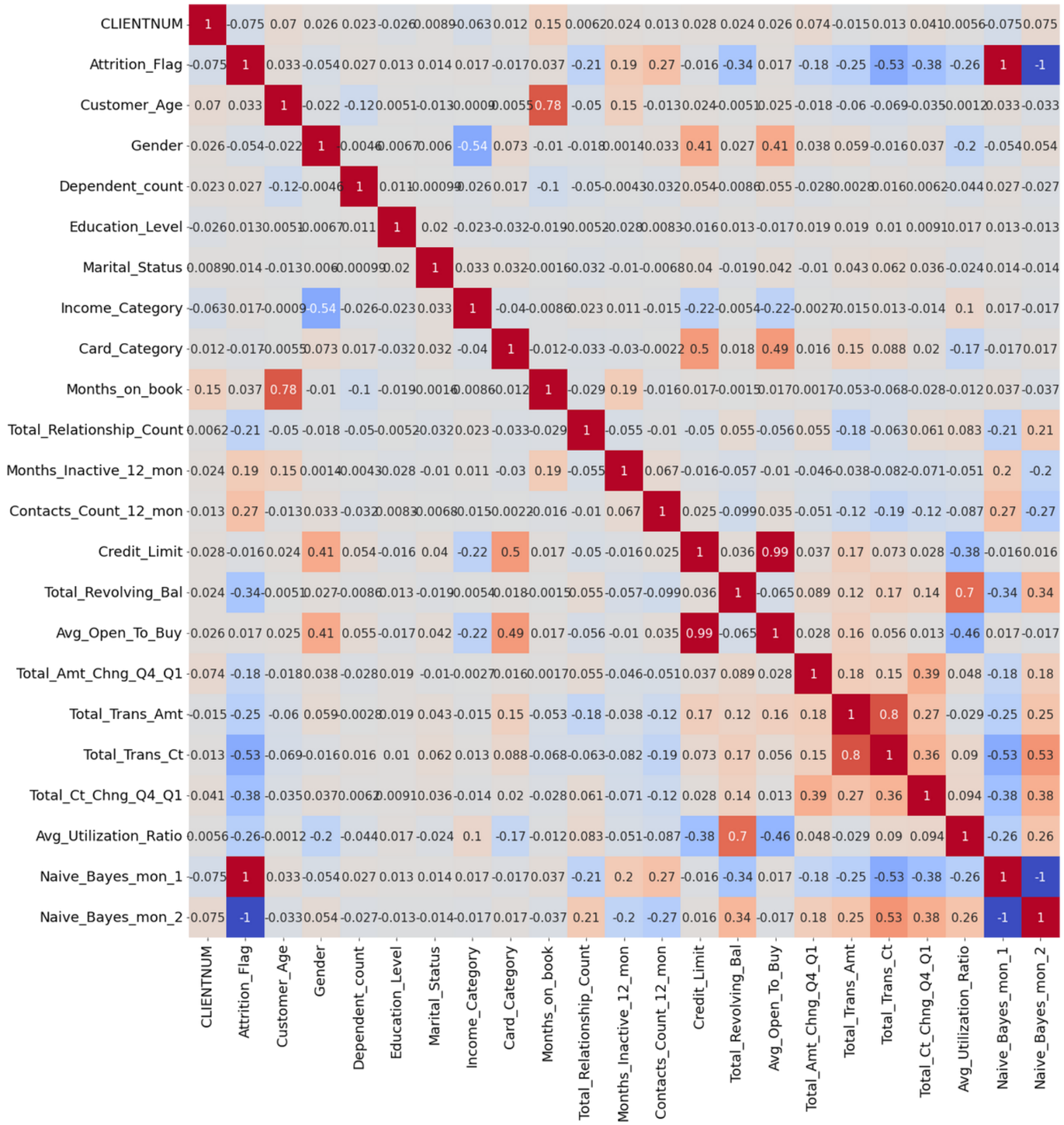
Boxplot Umur dan Dependent Customer berdasarkan Attrition Flag





- Terlihat bahwa tidak terdapat perbedaan signifikan pada umur customer baik yang Attrited maupun Existing

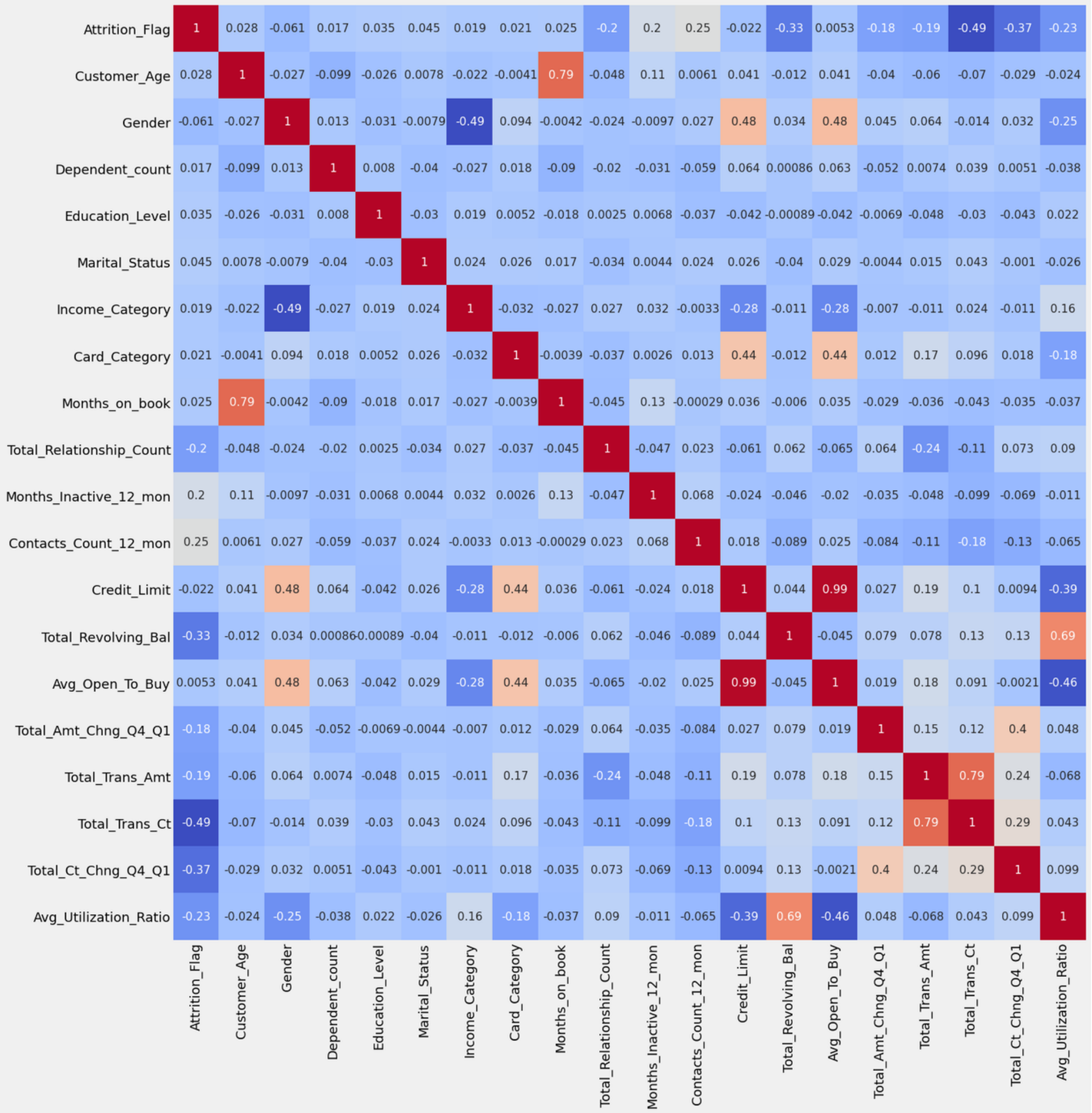
- Terlihat bahwa pada customer yang Attrited memiliki dependent count yang cukup tinggi dibandingkan customer yang Existing



Matriks korelasi

- Matriks korelasi dibuat berdasarkan variable yang memiliki kecenderungan korelasi yang cukup dengan Attrition
- Dapat dilihat bahwa Naive Bayes dan juga Clientnum memiliki korelasi cukup tinggi namun tidak berhubungan dengan attrition flag

Attrition_Flag	1.000000
Naive_Bayes_mon_1	0.999981
Contacts_Count_12_mon	0.269807
Months_Inactive_12_mon	0.193742
Months_on_book	0.037054
Customer_Age	0.033421
Dependent_count	0.026685
Avg_Open_To_Buy	0.017446
Income_Category	0.017053
Marital_Status	0.013694
Education_Level	0.013345
Credit_Limit	-0.016429
Card_Category	-0.017162
Gender	-0.053689
CLIENTNUM	-0.074770
Total_Amt_Chng_Q4_Q1	-0.176736
Total_Relationship_Count	-0.205691
Total_Trans_Amt	-0.248432
Avg_Utilization_Ratio	-0.257584
Total_Revolving_Bal	-0.336493
Total_Ct_Chng_Q4_Q1	-0.383022
Total_Trans_Ct	-0.529942
Naive_Bayes_mon_2	-0.999981



Matriks korelasi

- Karena hal tersebut maka dilakukan drop pada variabel clientnum dan juga naive bayes

Attrition_Flag	1.000000
Contacts_Count_12_mon	0.272232
Months_Inactive_12_mon	0.208682
Income_Category	0.026106
Education_Level	0.022673
Avg_Open_To_Buy	0.021975
Customer_Age	0.020120
Months_on_book	0.016948
Marital_Status	0.008034
Dependent_count	0.006197
Card_Category	0.004344
Credit_Limit	-0.003337
Gender	-0.054969
Total_Amt_Chng_Q4_Q1	-0.166294
Total_Relationship_Count	-0.178401
Total_Trans_Amt	-0.185691
Avg_Utilization_Ratio	-0.244988
Total_Revolving_Bal	-0.325302
Total_Ct_Chng_Q4_Q1	-0.364361
Total_Trans_Ct	-0.500273

Variabel yang dipilih berdasarkan urutan correlation matrix

Wrapper Method

- Wrapper method feature selection digunakan untuk pemilihan variabel pada dataset
- Variabel setelah melakukan wrapper method feature selection :

```
Index(['Customer_Age', 'Income_Category', 'Card_Category',
       'Total_Relationship_Count', 'Contacts_Count_12_mon', 'Credit_Limit',
       'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt', 'Total_Trans_Ct',
       'Total_Ct_Chng_Q4_Q1'],
      dtype='object')
```

Data Preprocessing

- **Memeriksa duplicate**

```
df.duplicated().any()
```

```
False
```

- **Menghapus variabel yang mengandung unknown**

```
df = df[~df['Education'].str.contains('Unknown')]
df = df[~df['Marital_Status'].str.contains('Unknown')]
df = df[~df['Income_Category'].str.contains('Unknown')]
```

Unknown dihapus karena jumlah datanya pada data kategorikal sedikit.

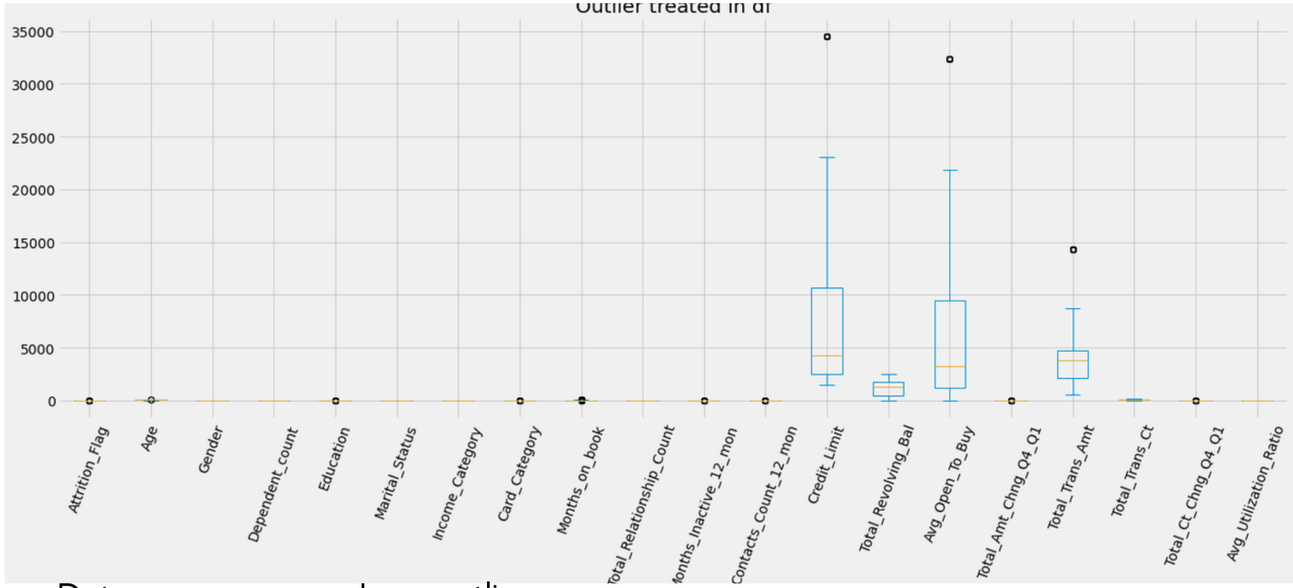
- **Menghapus nilai yang mengandung outlier**

```
def treat_outlier(x):
    q5 = np.percentile(x,5)
    q25= np.percentile(x,25)
    q75= np.percentile(x,75)
    up_trend=np.percentile(x,95)
    IQR = q75-q25
    low_level = q25-(1.5*IQR)
    up_level = q75+(1.5*IQR)

    return x.apply(lambda y: up_trend if y> up_level else
y).apply(lambda y: q5 if y < low_level else y)
```

```
outlier_list = ['Credit_Limit','Avg_Open_To_Buy','Total_Trans_Amt']
for i in df[outlier_list]:
    df[i]=treat_outlier(df[i])
```

```
plt.style.use('fivethirtyeight')
outlier= df.plot(kind='box',figsize=(20,7));
plt.xticks(rotation=70);
plt.title('Outlier treated in df');
```



Data yang mengandung outlier :

- Credit_Limit
- Total_Revolving_Bal
- Avg_Open_To_Buy
- Total_Trans_Amt

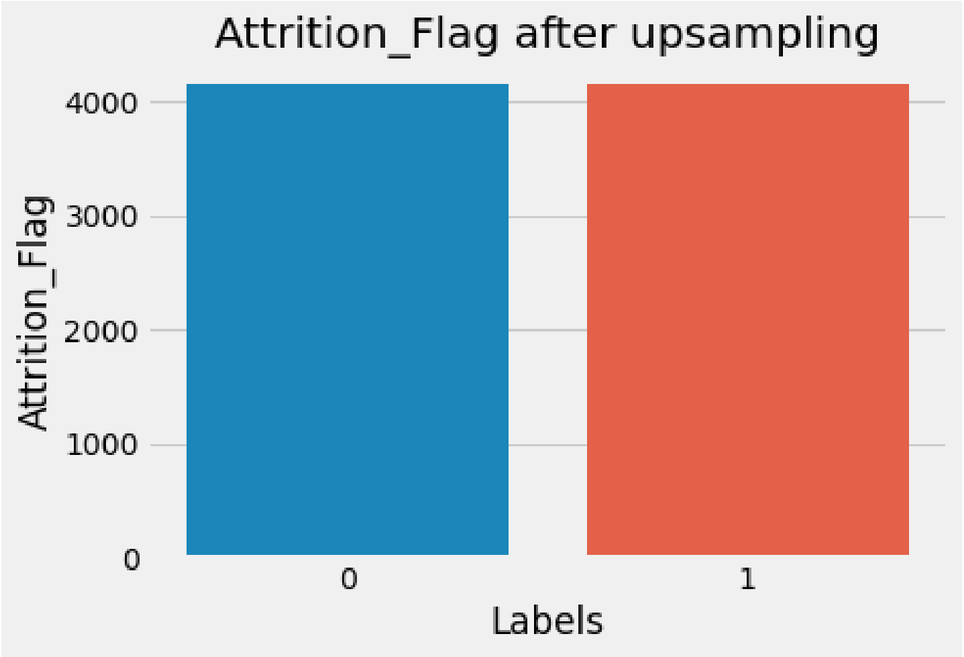
- **Melakukan Oversampling dengan SMOTE**

```
from sklearn.model_selection import train_test_split
X_train, X_test, train_labels, test_labels =
train_test_split(X,y,test_size=0.30,random_state=0)

from imblearn.over_sampling import SMOTE
OS_SMOTE = SMOTE()

X_train, train_labels = OS_SMOTE.fit_resample(X_train,train_labels)

oversample_plot = train_labels.value_counts().reset_index()
oversample_plot.columns = ['Labels','Attrition_Flag']
print(oversample_plot)
sns.barplot(x='Labels',y='Attrition_Flag',data=oversample_plot);
plt.title('Attrition_Flag after upsampling');
```



Melakukan oversampling SMOTE untuk menyeimbangkan data Existing customer dan Attrited customer pada variabel Attrition Flag.

Data Modelling

Data Modelling

- Pada Tahap Pemodelan, dilakukan beberapa treatment terlebih dahulu yaitu :

  1. Melakukan penanganan terhadap imbalanced data
  2. Penanganan outlier pada data

- Kemudian, dilakukan pemodelan dengan memecah data menjadi :

  1. Data Train Sebesar 70% dari Total data
  2. Data Test sebesar 30% dari Total data

- Dengan menggunakan Hyperparameter GridSearchCv dilakukan pemodelan dengan model :

  - SVM
  - XGBOOST
  - KNN
  - Random Forest
  - Logistic Regression

Hasil Evaluasi Model dengan GridSearchCv dengan Feature Selection Correlation Matrix

Model	Hyperparameter	Accuracy	Precision		Recall		F1 - Score		ROC AUC	MSE
			0	1	0	1	0	1		
Random Forest	Tanpa Parameter	86%	89%	83%	90%	81%	89%	82%	86%	12%
	GridSearchCv	88%	90%	85%	90%	85%	90%	85%	87%	11%
KNN	Tanpa Parameter	84%	90%	79%	86%	81%	88%	80%	83%	15%
	GridSearchCv	86%	89%	81%	87%	83%	88%	82%	85%	14%
Logistic Regression	Tanpa Parameter	84%	90%	79%	86%	81%	88%	80%	83%	15%
	GridSearchCv	83%	88%	76%	83%	82%	85%	79%	82%	17%
SVM	Tanpa Parameter	84%	89%	78%	89%	83%	89%	80%	85%	13%
	GridSearchCv	83%	88%	76%	83%	82%	85%	79%	82%	17%
XGBOOST	Tanpa Parameter	86%	90%	81%	89%	82%	90%	82%	84%	15%
	GridSearchCv	87%	89%	84%	90%	83%	89%	84%	86%	12%

Pada Feature Selection dengan Correlation Matrix, model yang paling bagus direkomendasikan adalah Random Forest dengan GridSearchCv dengan Accuracy 88%, Precision 85%, Recall 85%, F1 - Score 85%, ROC-AUC 87% dan estimasi nilai eror pada model Random Forest paling kecil 11%

Hasil Evaluasi Model dengan GridSearchCv dengan Feature Selection Wrapper Method

Model	Hyperparameter	Accuracy	Precision		Recall		F1 - Score		ROC AUC	MSE
			0	1	0	1	0	1		
Random Forest	Tanpa Parameter	93%	97%	87%	91%	95%	94%	91%	93%	7%
	GridSearchCv	95%	97%	92%	94%	96%	96%	94%	95%	5%
KNN	Tanpa Parameter	81%	90%	72%	78%	86%	84%	78%	82%	18%
	GridSearchCv	83%	86%	76%	83%	84%	86%	80%	83%	17%
Logistic Regressio	Tanpa Parameter	81%	90%	72%	78%	86%	84%	78%	82%	18%
	GridSearchCv	83%	87%	76%	84%	81%	85%	79%	82%	16%
SVM	Tanpa Parameter	84%	90%	77%	83%	86%	86%	81%	84%	15%
	GridSearchCv	83%	87%	76%	84%	81%	85%	79%	82%	16%
XGBOOST	Tanpa Parameter	93%	97%	88%	92%	95%	94%	92%	93%	6%
	GridSearchCv	94%	96%	91%	94%	94%	95%	93%	94%	16%

Pada Feature Selection dengan Wrapper Method, model yang paling bagus direkomendasikan adalah Random Forest dengan GridSearchCv dengan Accuracy 95%, Precision 92%, Recall 96%, F1 - Score 94%, ROC-AUC 96% dan estimasi nilai eror pada model Random Forest paling kecil 5%

Kesimpulan

Pada tahap Evaluasi model, ada 5 model yang digunakan dan nilai dari persentasinya 85% - 95% dengan tingkat eror yang cukup kecil 5%.

Rekomendasi

Memfollow up pelanggan yang sudah keluar maupun yang masih aktif dengan memberikan penawaran menarik seperti bebas uang iuran pada tahun pertama, paket liburan, dll.

Thank You!