Proyek Data Analyst

Business Decision Research With Python

DQLab sport center adalah toko yang menjual berbagai kebutuhan olahraga seperti Jaket, Baju, Tas, dan Sepatu. Toko ini mulai berjualan sejak tahun 2013, sehingga sudah memiliki pelanggan tetap sejak lama, dan tetap berusaha untuk mendapatkan pelanggan baru sampai saat ini.

Di awal tahun 2019, manajer toko tersebut merekrut junior DA untuk membantu memecahkan masalah yang ada di tokonya, yaitu menurunnya pelanggan yang membeli kembali ke tokonya. Junior DA tersebut pun diberi kepercayaan mengolah data transaksi toko tersebut. Manajer toko mendefinisikan bahwa customer termasuk sudah bukan disebut pelanggan lagi (churn) ketika dia sudah tidak bertransaksi ke tokonya lagi sampai dengan 6 bulan terakhir dari update data terakhir yang tersedia.

Manajer toko pun memberikan data transaksi dari tahun 2013 sampai dengan 2019 dalam bentuk csv (comma separated value) dengan data_retail.csv dengan jumlah baris 100.000 baris data.

Berikut tampilan datanya:

	no	Row_Num	Customer_ID	Product	First_Transaction	Last_Transaction	Average_Transaction_Amount	Count_Transaction
0	1	1	29531	Jaket	1466304274396	1538718482608	1467681	22
1	2	2	29531	Sepatu	1406077331494	1545735761270	1269337	41
2	3	3	141526	Tas	1493349147000	1548322802000	310915	30
3	4	4	141526	Jaket	1493362372547	1547643603911	722632	27
4	5	5	37545	Sepatu	1429178498531	1542891221530	1775036	25

Berdasarkan data diatas, field yang ada adalah :

- No
- Row_Num
- Customer_ID
- Product
- First_Transaction
- Last_Transaction
- Average_Transaction_Amount
- Count_Transaction

1. Data Preparation

a) Inspection Data

Melakukan inspeksi data dengan menampilkan 5 data teratas dan info dataset.

```
Lima data teratas:
   no  Row_Num  ...  Average_Transaction_Amount Count_Transaction
0   1        1  ...                     1467681                22
1   2        2  ...                     1269337                41
2   3        3  ...                      310915                30
3   4        4  ...                      722632                27
4   5        5  ...                     1775036                25

[5 rows x 8 columns]

Info dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 8 columns):
no                          100000 non-null int64
Row_Num                     100000 non-null int64
Customer_ID                 100000 non-null int64
Product                     100000 non-null object
First_Transaction           100000 non-null int64
Last_Transaction            100000 non-null int64
Average_Transaction_Amount  100000 non-null int64
Count_Transaction           100000 non-null int64
dtypes: int64(7), object(1)
memory usage: 6.1+ MB
None
```

b) Data Cleansing

Berdasarkan info dataset diatas, terdapat dua kolom yang menunjukkan terjadinya transaksi tidak bertipe datetime, maka ubahlah kedua kolom tersebut ke tipe data datetime.

```
Info dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 8 columns):
no                          100000 non-null int64
Row_Num                     100000 non-null int64
Customer_ID                 100000 non-null int64
Product                     100000 non-null object
First_Transaction           100000 non-null datetime64[ns]
Last_Transaction            100000 non-null datetime64[ns]
Average_Transaction_Amount  100000 non-null int64
Count_Transaction           100000 non-null int64
dtypes: datetime64[ns](2), int64(5), object(1)
memory usage: 6.1+ MB
None
```

c) Churn Customer

Untuk menentukan churn customer, perlu diketahui dahulu transaksi terakhir yang dilakukan dan mengklasifikasikan customer yang berstatus churn dan yang tidak.

```
2019-02-01 23:57:57.286000013
Lima data teratas:
   no  Row_Num  ...  Count_Transaction is_churn
0   1        1  ...                 22    False
1   2        2  ...                 41    False
2   3        3  ...                 30    False
3   4        4  ...                 27    False
4   5        5  ...                 25    False

[5 rows x 9 columns]

Info dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
no                          100000 non-null int64
Row_Num                     100000 non-null int64
Customer_ID                 100000 non-null int64
Product                     100000 non-null object
First_Transaction           100000 non-null datetime64[ns]
Last_Transaction            100000 non-null datetime64[ns]
Average_Transaction_Amount  100000 non-null int64
Count_Transaction           100000 non-null int64
is_churn                    100000 non-null bool
dtypes: bool(1), datetime64[ns](2), int64(5), object(1)
memory usage: 6.2+ MB
None
```

d) Menghapus Kolom Yang Tidak Diperlukan

Hapus kolom no dan Row_Num.

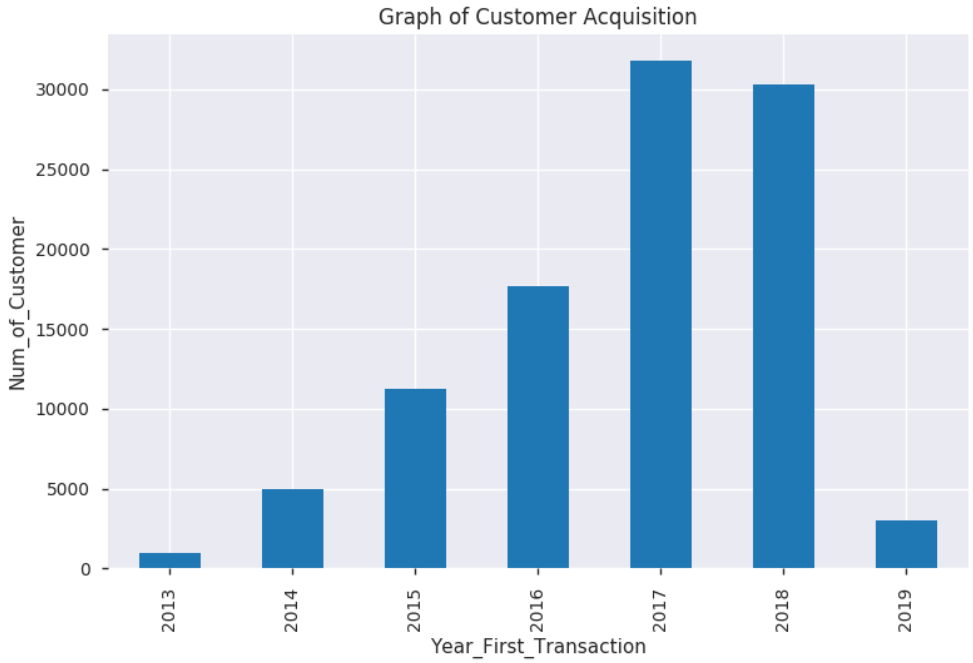
```
   Customer_ID Product  ... Average_Transaction_Amount Count_Transaction
0        29531   Jaket  ...                    1467681                22
1        29531  Sepatu  ...                    1269337                41
2       141526     Tas  ...                     310915                30
3       141526   Jaket  ...                     722632                27
4        37545  Sepatu  ...                    1775036                25

[5 rows x 6 columns]
```

2. Data Visualization

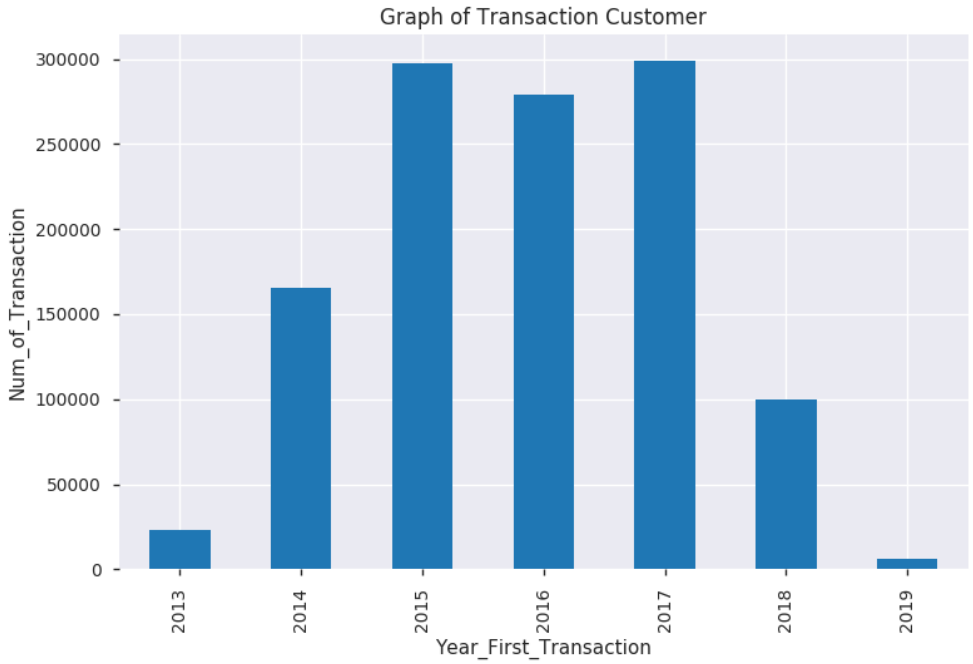
a) Customer Acquisition by Year

Membuat visualisasi data berupa trend of customer acquisition by year dengan meggunakan bar chart.



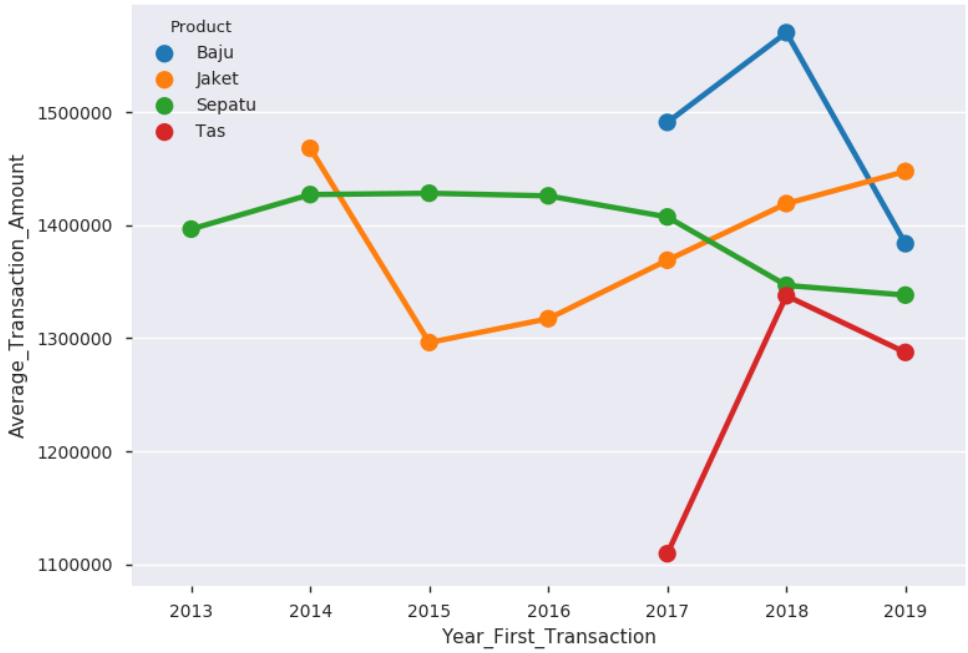
b) Transaction by Year

Visualisasi trend jumlah transaksi per tahunnya dengan menggunakan bar chart.



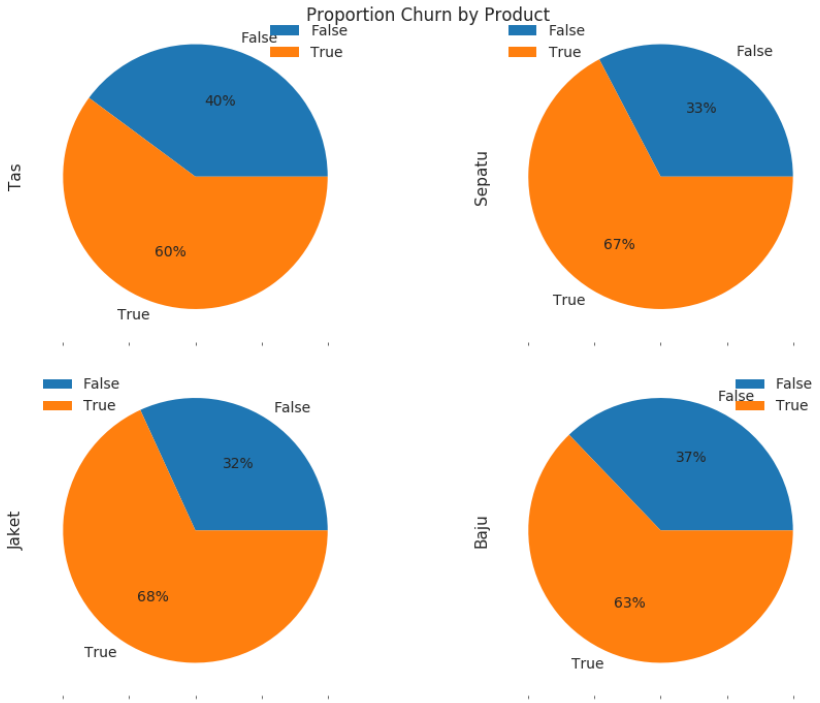
c) Average Transaction Amount by Year

Dengan menggunakan seaborn pointplot, akan dibuat visualisasi tren dari tahun ke tahun rata-rata jumlah transaksi untuk tiap-tiap produknya.



d) Proporsi Churned Customer Untuk Tiap Produk

Dari sisi churned customer, khususnya untuk melihat seberapa besar proporsi churned customer untuk tiap-tiap produk dapat diketahui insight-nya melalui pie chart.

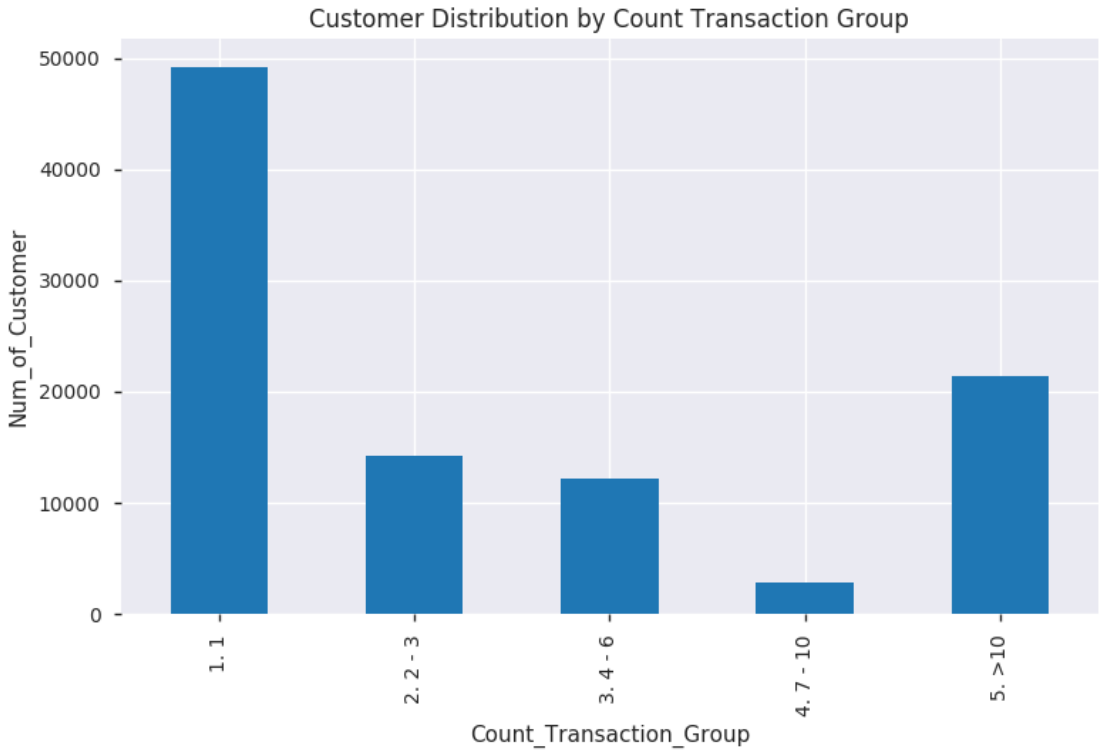


e) Distribusi Kategorisasi Count Transaction

Selanjutnya akan dilakukan visualisasi dari distribusi kategorisasi count transaction. Kategorisasi ini dilakukan dengan mengelompokkan jumlah transaksi seperti yang diperlihatkan oleh tabel berikut:

Rentang jumlah transaksi	Kategori
s/d 1	1. 1
2 s/d 3	2. 2 - 3
4 s/d 6	3. 4 - 6
7 s/d 10	4. 7 - 10
> 10	5. > 10

Setelah menambahkan kolom baru untuk kategori ini dengan nama Count_Transaction_Group, selajutnya dilakukan visualisasi dengan bar chart.

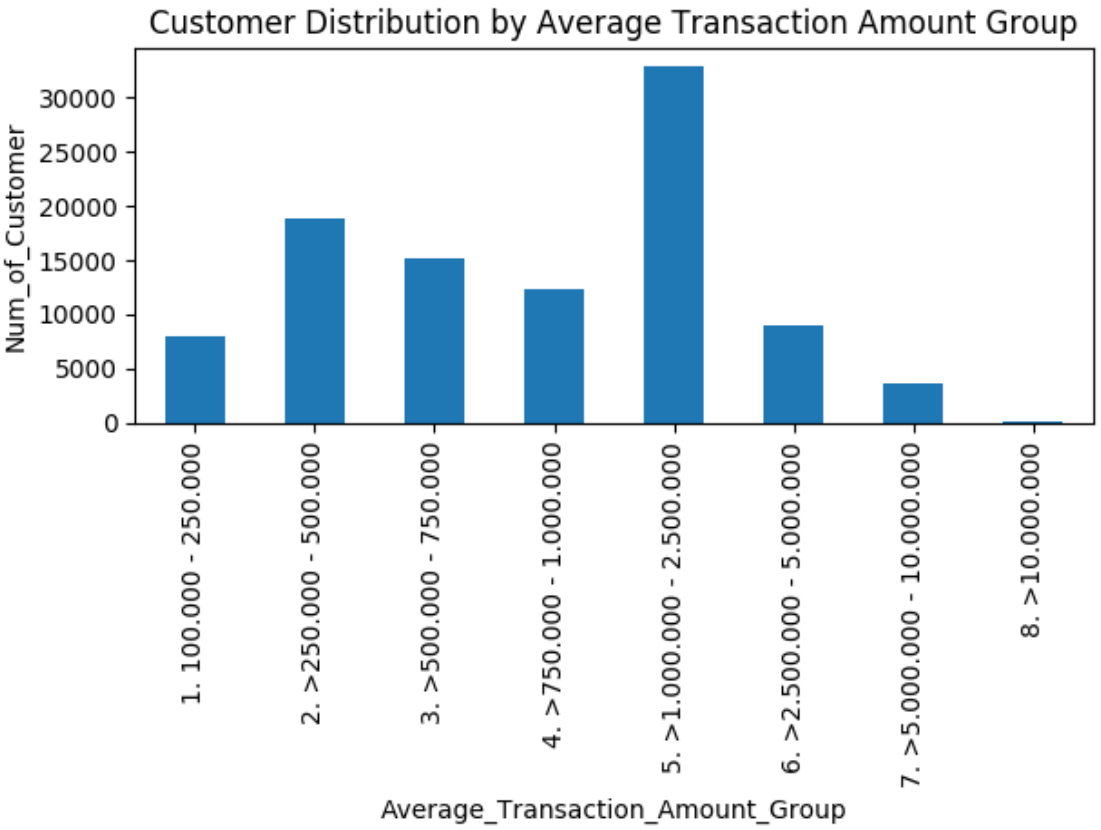


f) Distribusi Kategorisasi Average Transaction Amount

Selanjutnya, akan melakukan visualisasi dari distribusi kategorisasi average transaction amount. Kategorisasi ini dilakukan dengan mengelompokkan rata-rata besar transaksi seperti yang diperlihatkan oleh tabel berikut:

Rentang rata-rata besar transaksi	Kategori
100.000 s/d 250.000	1. 100.000 - 250.000
>250.000 s/d 500.000	2. >250.000 - 500.000
>500.000 s/d 750.000	3. >500.000 - 750.000
>750.000 s/d 1.000.000	4. >750.000 - 1.000.000
>1.000.000 s/d 2.500.000	5. >1.000.000 - 2.500.000
>2.500.000 s/d 5.000.000	6. >2.500.000 - 5.000.000
>5.000.000 s/d 10.000.000	7. >5.000.000 - 10.000.000
>10.000.000	8. >10.000.000

Setelah ditambahkan kolom baru untuk kategori ini dengan nama Average_Transaction_Amount_Group, selanjutnya dilakukan visualisasi dengan bar chart.



3. Modelling

a) Feature columns and Target

Dibagian ini, perlu menentukan feature columns dari dataset yang dimiliki, di sini dipilih kolom Average_Transaction_Amount, Count_Transaction, dan Year_Diff. Akan tetapi, kolom terakhir belum ada. Sehingga akan dicreate dahulu kolom Year_Diff ini dan kemudian assign dataset dengan feature columns ini sebagai variabel independent X.

Untuk target tentunya persoalan costumer dengan kondisi churn atau tidak, assign dataset untuk target ini ke dalam variabe dependent y.

```
import pandas as pd
df = pd.read_csv('https://storage.googleapis.com/dqlab-dataset/data_retail.csv', sep=';')
df['First_Transaction'] = pd.to_datetime(df['First_Transaction']/1000, unit='s', origin='1970-01-01')
df['Last_Transaction'] = pd.to_datetime(df['Last_Transaction']/1000, unit='s', origin='1970-01-01')
df['Year_First_Transaction'] = df['First_Transaction'].dt.year
df['Year_Last_Transaction'] = df['Last_Transaction'].dt.year
df.loc[df['Last_Transaction'] <= '2018-08-01', 'is_churn'] = True
df.loc[df['Last_Transaction'] > '2018-08-01', 'is_churn'] = False

# Feature column: Year_Diff
df['Year_Diff'] = df['Year_Last_Transaction'] - df['Year_First_Transaction']
# Nama-nama feature columns
feature_columns = ['Average_Transaction_Amount', 'Count_Transaction', 'Year_Diff']

# Features variable
X = df[feature_columns]

# Target variable
y = df['is_churn']
```

b) Split X dan Y ke Dalam Bagian Training

Setelah variabel independent X dan variabel dependent y selesai dilakukan, selanjutnya memecahkan X dan y ke dalam bagian training dan testing. Bagian testing 25% dari jumlah entri data.

```
import pandas as pd
df = pd.read_csv('https://storage.googleapis.com/dqlab-dataset/data_retail.csv', sep=';')
df['First_Transaction'] = pd.to_datetime(df['First_Transaction']/1000, unit='s', origin='1970-01-01')
df['Last_Transaction'] = pd.to_datetime(df['Last_Transaction']/1000, unit='s', origin='1970-01-01')
df['Year_First_Transaction'] = df['First_Transaction'].dt.year
df['Year_Last_Transaction'] = df['Last_Transaction'].dt.year
df.loc[df['Last_Transaction'] <= '2018-08-01', 'is_churn'] = True
df.loc[df['Last_Transaction'] > '2018-08-01', 'is_churn'] = False

df['Year_Diff']=df['Year_Last_Transaction']-df['Year_First_Transaction']
feature_columns = ['Average_Transaction_Amount', 'Count_Transaction', 'Year_Diff']

X = df[feature_columns]
y = df['is_churn']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

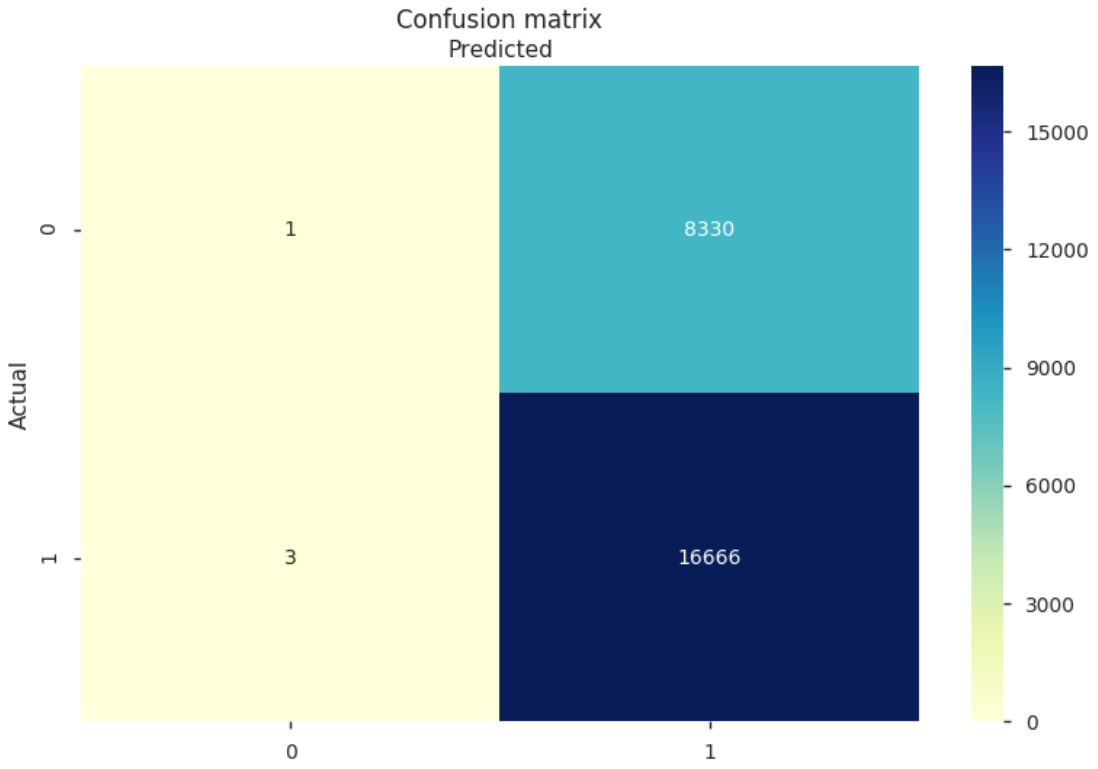
c) Train, Predict and Evaluate

Langkah selanjutnya akan membuat model menggunakan Logistic Regression, inisialisasilah model, fit, dan kemudian evaluasi model dengan menggunakan confusion matrix.

```
Confusion Matrix:
 [[    1  8330]
 [    3 16666]]
```

d) Visualisasi Confusion Matrix

Confusion matrix yang telah dihitung sebelumnya dapat divisualisasikan dengan menggunakan heatmap dari seaborn. Untuk itu akan ditampilkan visualisasi dari confusion matrix ini.



e) Accuracy, Precision, and Recall

Kemudian, perlu dihitung nilai accuracy, precission dan recall berdasarkan nilai target sesungguhnya dan nilai target hasil prediksi.

```
Accuracy : 0.66668
Precision: 0.66668
Recall   : 0.66668
```

Platform : DQLab