DQLab

www.dqlab.id

Bussines Decision Research

By Muhammad Rizky Halawi

Bussines Decision Research

This is last project in DQLab for program Data Analyst Career. The purpose of this project is to assess that it understands concepts of basics of using python to being able to create data visualizations and making some basic stats method test. This project is practice to analyze data of some cases.

Goals

Goal 1

To Analyze data for bussines decision research data

Goal 2

To understand some coding test like data preparation, visualization and stats modelin test

Goal 3

To know Accuracy, Precision, and Recall on this Data

DQLab

www.dqlab.id

Type of Data Analytics Test

Theoritical test → **Basic Theory**

Coding Test →

- **Data preparation**
- **Data Visualization**
- **Basic Stats Method Test**

Content Table

Theoritical

1

Skill for Data Analyst

Business Understanding

Data Cleansing and Algorithm skills

Data Storytelling and Visualization

DQLab
www.dqlab.id

Data Preparation 2

Importing Data and Inspection

```
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):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   no                          100000 non-null  int64
 1   Row_Num                     100000 non-null  int64
 2   Customer_ID                 100000 non-null  int64
 3   Product                     100000 non-null  object
 4   First_Transaction           100000 non-null  int64
 5   Last_Transaction            100000 non-null  int64
 6   Average_Transaction_Amount  100000 non-null  int64
 7   Count_Transaction           100000 non-null  int64
dtypes: int64(7), object(1)
memory usage: 6.1+ MB
None
```

Data Cleansing

```
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):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   no                          100000 non-null  int64
 1   Row_Num                     100000 non-null  int64
 2   Customer_ID                 100000 non-null  int64
 3   Product                     100000 non-null  object
 4   First_Transaction           100000 non-null  datetime64[ns]
 5   Last_Transaction            100000 non-null  datetime64[ns]
 6   Average_Transaction_Amount  100000 non-null  int64
 7   Count_Transaction           100000 non-null  int64
dtypes: datetime64[ns](2), int64(5), object(1)
memory usage: 6.1+ MB
None
```

Churn Customer

Code

```
# Pengecekan transaksaksi terakhir dalam dataset
print(max(df['Last_Transaction']))

# Klasifikasikan customer yang berstatus churn atau tidak dengan boolean
df.loc[df['Last_Transaction'] <= '2018-08-01', 'is_churn'] = True
df.loc[df['Last_Transaction'] > '2018-08-01', 'is_churn'] = False

print('Lima data teratas:')
print(df.head())

print('\nInfo dataset:')
print(df.info())
```

Output

```
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):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   no                          100000 non-null  int64
 1   Row_Num                     100000 non-null  int64
 2   Customer_ID                 100000 non-null  int64
 3   Product                     100000 non-null  object
 4   First_Transaction           100000 non-null  datetime64[ns]
 5   Last_Transaction            100000 non-null  datetime64[ns]
 6   Average_Transaction_Amount  100000 non-null  int64
 7   Count_Transaction           100000 non-null  int64
 8   is_churn                    100000 non-null  object
dtypes: datetime64[ns](2), int64(5), object(2)
memory usage: 6.9+ MB
None
```

To find Churn Customers:

- Find Last Transaction
- Classified Customer with Churn Status

DQLab

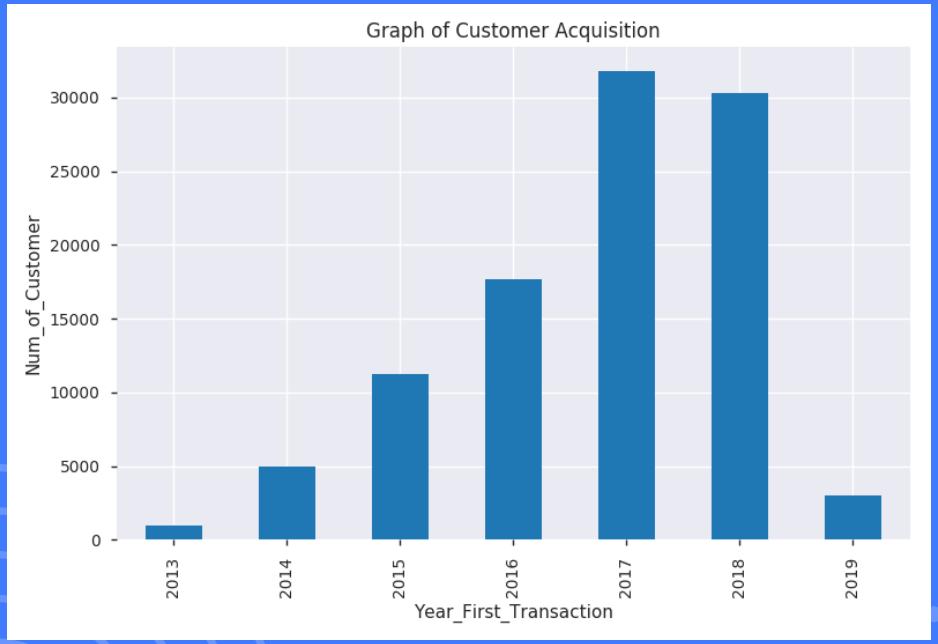
www.dqlab.id

3

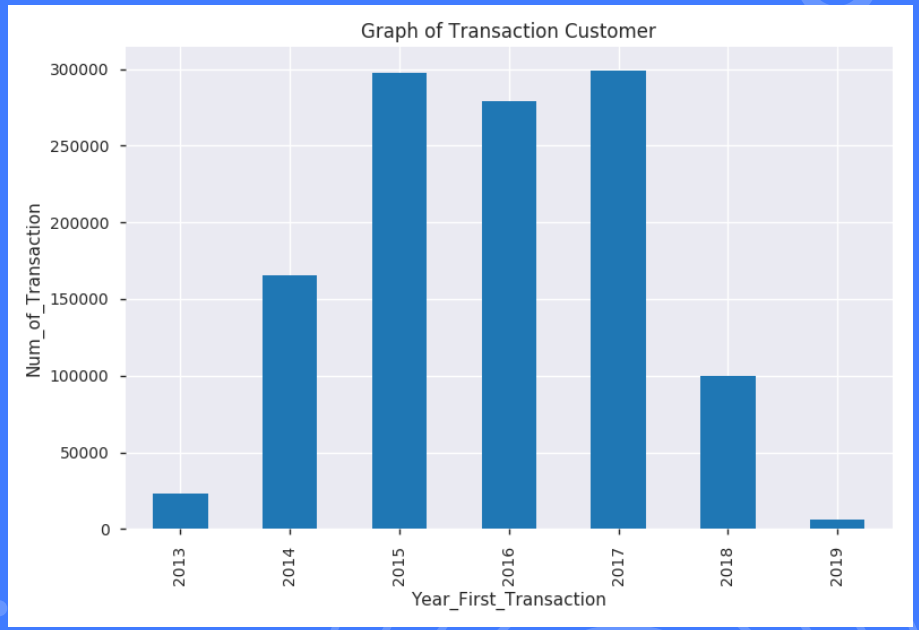
Data Visualization

Bussiness Decision Research Data Visualization

Customer acquisition by year

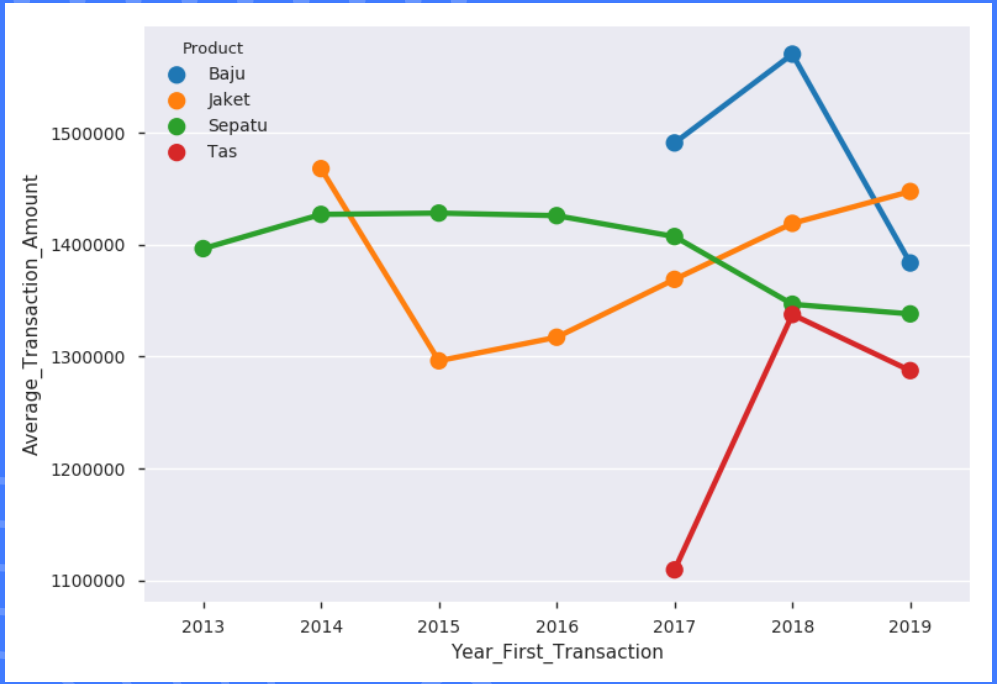


Transaction by year

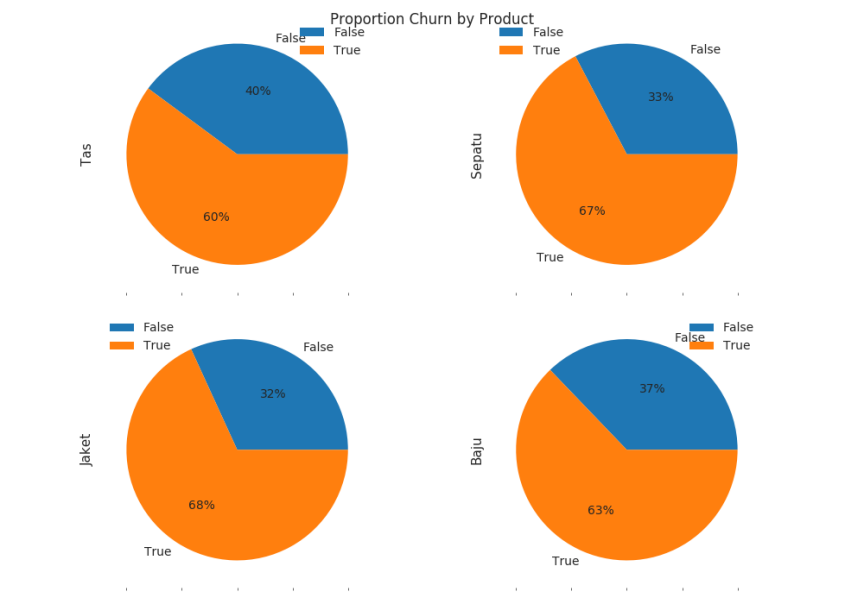


Bussiness Decision Research Data Visualization

Average transaction amount by year

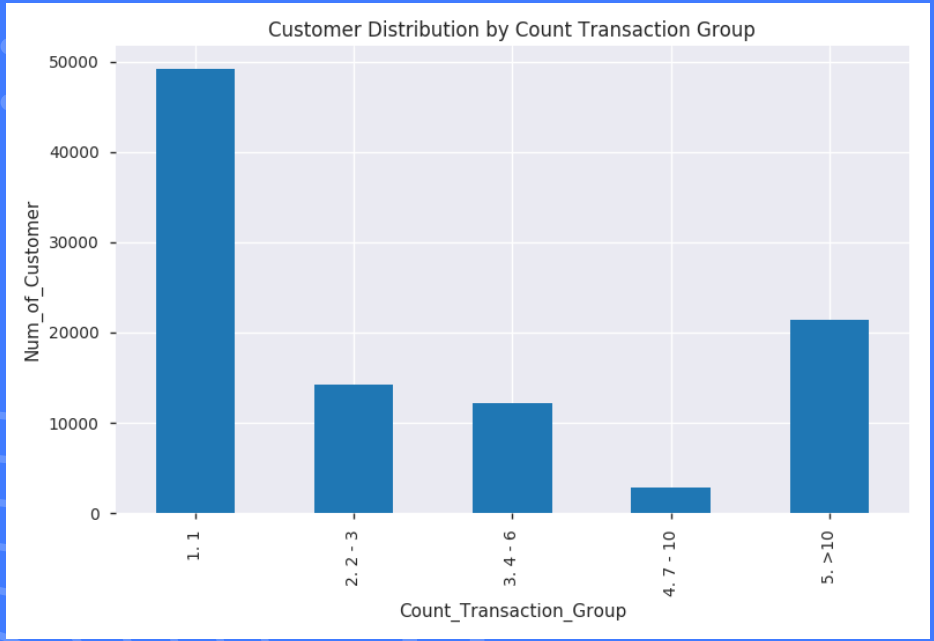


Proportion Churn by Product

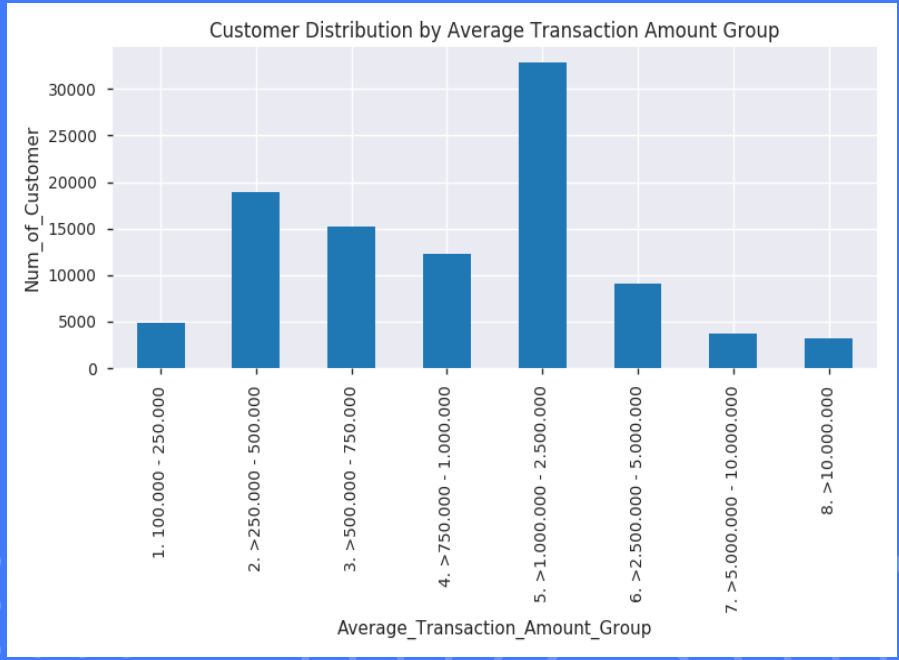


Bussiness Decision Research Data Visualization

Customer Distribution by Count Transaction



Customer Distribution by Average Transaction Amount Group



Modeling

4

Type of Modeling

Feature columns

```
# 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']
```

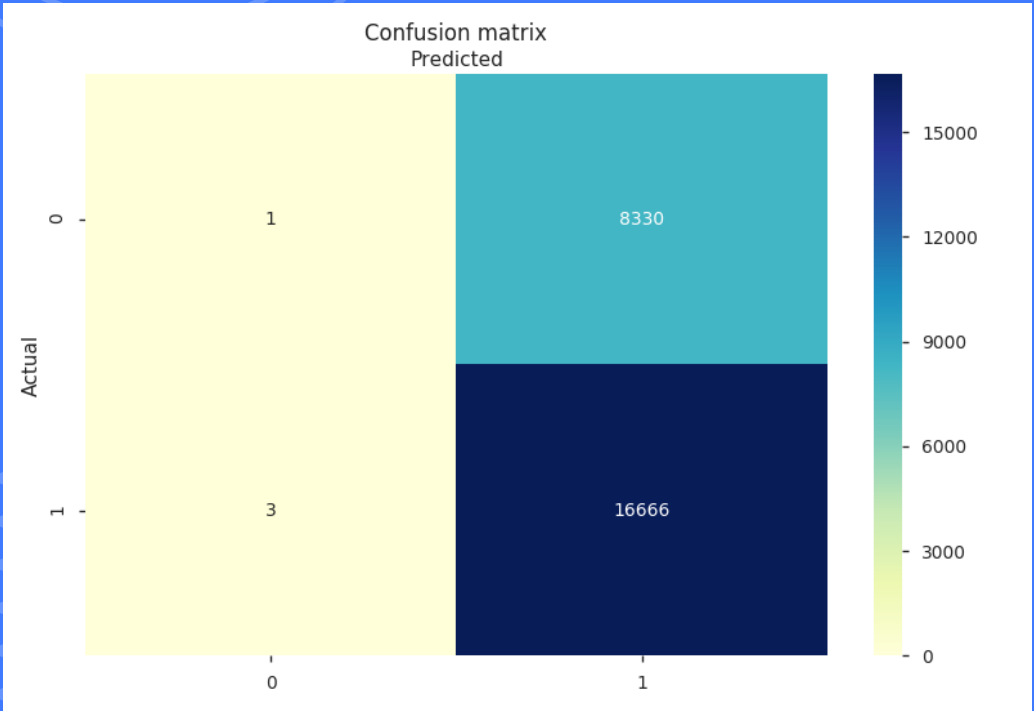
Train, Predict and Evaluate

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix

# Inisiasi model logreg
logreg = LogisticRegression()

# fit the model with data
logreg.fit(X_train, y_train)

# Predict model
y_pred=logreg.predict(X_test)

# Evaluasi model menggunakan confusion matrix
cnf_matrix = confusion_matrix(y_test, y_pred)
print('Confusion Matrix:\n', cnf_matrix)
```

Split X and y in training and testing

```
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)
```

Confusion Matrix



Code

```
# import required modules
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

plt.clf()
# name  of classes
class_names = [0, 1]
fig, ax = plt.subplots()

tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)

# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix),
annot=True, cmap='YlGnBu', fmt='g')
ax.xaxis.set_label_position('top')
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.tight_layout()
plt.show()
```

Accuracy, Precision, dan Recall this Data

DQLab
www.dqlab.id

```
from sklearn.metrics import accuracy_score, precision_score, recall_score

#Menghitung Accuracy, Precision, dan Recall
print('Accuracy :', accuracy_score(y_test, y_pred))
print('Precision:', precision_score(y_test, y_pred, average='micro'))
print('Recall   :', recall_score(y_test, y_pred, average='micro'))
```

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

DQLab

Conclusion

- The important step for data preparation in cleansing
- Do the 6 type of Visualization of this data from Acquisition until distribution of the Customer
- Type of stats modeling on this data is make Feature Column, train, predict and Evaluate, and Etc.

Tools of This Project

Thank you

DOLab

CONTACT ME

https://www.kaggle.com/rizkyhalawi

+62812131463091

rizkyhalawi@gmail.com

DRAMAGA, BOGOR, INDONESIA