Intermediate Assignment Python

(Syihab Agung Satriotomo)

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Analytical Objective

What is the business requirement / goal of analysis?

By understanding the dataset bank_promotion_analysis we can conclude that we want to know Spending pattern among Revoshop customers who are users of RevoBank credit card by making correlation between column :

- AVG_TXN_AMT_L6M as average sales amount per transaction
- TXN_CNT_L6M as number of transaction occurred in the past 6 months
- PROMO_TXN_CNT_L6M as Number of transaction occurred in the past 6 months in response to promo on merchant

So we could make Customer segmentation & Promo sensitivity on each customer who are used promo the last 6 month so we make could find ways to reduce the cost of the promotion.

Data Cleaning

Info and missing value :

From 111133 and 23 columns in the dataset

```
[660] dfp_eda.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 111133 entries, 0 to 112632
Data columns (total 24 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   ACCOUNT_ID              111133 non-null  object
 1   MCC                     111133 non-null  object
 2   MERCHANT_NAME           111133 non-null  object
 3   AVG_TXN_AMT_L6M         111133 non-null  float64
 4   AVG_TXN_AMT_LTM         111133 non-null  float64
 5   AVG_PROMO_TXN_AMT_L6M   111133 non-null  float64
 6   AVG_PROMO_TXN_AMT_LTM   111133 non-null  float64
 7   TXN_CNT_L6M             111133 non-null  int64
 8   TXN_CNT_LTM             111133 non-null  int64
 9   PROMO_TXN_CNT_L6M       111133 non-null  int64
 10  PROMO_TXN_CNT_LTM       111133 non-null  int64
 11  LAST_TXN_DAY            111133 non-null  int64
 12  CNT_PROMO_L6M           111133 non-null  int64
 13  CNT_PROMO_L12M          111133 non-null  int64
 14  CUST_VALUE_GROUP        111133 non-null  object
 15  MAPP_ACTIVE_GROUP       111133 non-null  object
 16  PROXY_INCOME            111133 non-null  int64
 17  MOB                     111133 non-null  int64
 18  FLAG_FEMALE             111133 non-null  int64
 19  PROMO_CHANNEL           111133 non-null  object
 20  BIRTH_DATE              111133 non-null  datetime64[ns]
 21  promo_experience        108268 non-null  object
 22  PROMO_SENSITIVE_LTM     111133 non-null  float64
 23  PROMO_SENSITIVE_L6M     111133 non-null  float64
dtypes: datetime64[ns](1), float64(6), int64(10), object(7)
memory usage: 21.2+ MB
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Data Cleaning

Data Cleaning Steps and Considerations : String manipulation

1. Change ACCOUNT_ID column into string data type so it wouldn't calculate when it counted
2. Change PROMO_CHANNEL column into string data type so it wouldn't calculate when it counted

```
[604] # Change ACCOUNT_ID dat atype into string
      dfp_dc['ACCOUNT_ID'] = dfp_dc['ACCOUNT_ID'].astype(str)

      # Check the datatype of ACCOUNT_ID after replaced
      dfp_dc['ACCOUNT_ID'].dtype

      dtype('O')

[605] # Change PROMO_CHANNEL data type into string
      dfp_dc['PROMO_CHANNEL'] = dfp_dc['PROMO_CHANNEL'].astype(str)

      # Check the datatype of ACCOUNT_ID after replaced
      dfp_dc['PROMO_CHANNEL'].dtype

      dtype('O')
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Data Cleaning

Data Cleaning Steps and Considerations : Time Series Manipulation

Change BIRTH_DATE column into datetime data type because it has year, month, and day of each customer born

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

```
# convert BIRTH_DATE column into date time data type
dfp_dc['BIRTH_DATE'] = pd.to_datetime(dfp_dc['BIRTH_DATE'])
dfp_dc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112634 entries, 0 to 112633
Data columns (total 24 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   ACCOUNT_ID              112634 non-null  object
 1   MCC                     112634 non-null  object
 2   MERCHANT_NAME           112634 non-null  object
 3   AVG_TXN_AMT_L6M         112634 non-null  float64
 4   AVG_TXN_AMT_LTM         112634 non-null  float64
 5   AVG_PROMO_TXN_AMT_L6M   112634 non-null  float64
 6   AVG_PROMO_TXN_AMT_LTM   112634 non-null  float64
 7   TXN_CNT_L6M             112634 non-null  int64
 8   TXN_CNT_LTM             112634 non-null  int64
 9   PROMO_TXN_CNT_L6M       112634 non-null  int64
 10  PROMO_TXN_CNT_LTM       112634 non-null  int64
 11  LAST_TXN_DAY            112634 non-null  int64
 12  CNT_PROMO_L6M           112634 non-null  int64
 13  CNT_PROMO_L12M          112634 non-null  int64
 14  CUST_VALUE_GROUP        112634 non-null  object
 15  MAPP_ACTIVE_GROUP       112634 non-null  object
 16  HOMEOWNER_STATUS        112634 non-null  int64
 17  HOME_VALUE              112634 non-null  int64
 18  PROXY_INCOME            112634 non-null  int64
 19  PCT_INCOME_RETIREMENT   112634 non-null  int64
 20  MOB                     112634 non-null  int64
 21  FLAG_FEMALE             112634 non-null  int64
 22  PROMO_CHANNEL           112634 non-null  object
 23  BIRTH_DATE              112634 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(4), int64(13), object(6)
memory usage: 20.6+ MB
```

Data Cleaning

Data Cleaning Steps and Considerations : Checking value & typo each column

Checking each column values one by one

```
# First we need to know the index of the columns
dfp_dc.columns
```

```
Index(['ACCOUNT_ID', 'MCC', 'MERCHANT_NAME', 'AVG_TXN_AMT_L6M',
       'AVG_TXN_AMT_LTM', 'AVG_PROMO_TXN_AMT_L6M', 'AVG_PROMO_TXN_AMT_LTM',
       'TXN_CNT_L6M', 'TXN_CNT_LTM', 'PROMO_TXN_CNT_L6M', 'PROMO_TXN_CNT_LTM',
       'LAST_TXN_DAY', 'CNT_PROMO_L6M', 'CNT_PROMO_L12M', 'CUST_VALUE_GROUP',
       'MAPP_ACTIVE_GROUP', 'HOMEOWNER_STATUS', 'HOME_VALUE', 'PROXY_INCOME',
       'PCT_INCOME_RETIREMENT', 'MOB', 'FLAG_FEMALE', 'PROMO_CHANNEL',
       'BIRTH_DATE'],
      dtype='object')
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Data Cleaning

Data Cleaning Steps and Considerations : Replacing string

On Column MERHCANT_NAME there is “REVOSHOP” and “REVOSH MKTPLC”

Those two are the same “REVOSHOP” so we change “REVOSH MKTPLC” into “REVOSHOP” and also remove and also exclude merchant other than REVOSHOP since we only look on REVOSHOP partnership

```
[182] dfp_eda[dfp_eda.columns[2]].value_counts()

REVOSHOP           111029
TOKTOKLIVE           1403
EL CORTE INGLES         1
Name: MERCHANT_NAME, dtype: int64
```

```
[611] # change REVOSH MKTPLC into REVOSH
      dfp_dc['MERCHANT_NAME'].replace (["REVOSH MKTPLC"],'REVOSHOP', inplace = True)
```

```
[640] dfp_dc[dfp_dc.columns[2]].value_counts()

REVOSHOP           111133
TOKTOKLIVE           1500
EL CORTE INGLES         1
Name: MERCHANT_NAME, dtype: int64

dfp = dfp_dc[dfp_dc['MERCHANT_NAME'].isin(['REVOSHOP'])]
dfp[dfp_dc.columns[2]].value_counts()

REVOSHOP     111133
Name: MERCHANT_NAME, dtype: int64
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Data Cleaning

Data Cleaning Steps and Considerations : treat missing and irrelevant values

- In column “AVG_PROMO_TXN_AMT_L6M, has non-null but missing values that identify as -1 so we make -1 become 0 so it wouldn’t affect our calculate
- .HOMEOWNER_STATUS column did exclude since it has no relation to any other column
- PCT_INCOME_RETIREMENT has been excluded because there is a lot missing data value on it

Data Cleaning

Data Cleaning Steps and Considerations : confirm column was dropped

```
[161] # confirm that above column was dropped
      dfp_dc.columns

      Index(['ACCOUNT_ID', 'MCC', 'MERCHANT_NAME', 'AVG_TXN_AMT_L6M',
             'AVG_TXN_AMT_LTM', 'AVG_PROMO_TXN_AMT_L6M', 'AVG_PROMO_TXN_AMT_LTM',
             'TXN_CNT_L6M', 'TXN_CNT_LTM', 'PROMO_TXN_CNT_L6M', 'PROMO_TXN_CNT_LTM',
             'LAST_TXN_DAY', 'CNT_PROMO_L6M', 'CNT_PROMO_L12M', 'CUST_VALUE_GROUP',
             'MAPP_ACTIVE_GROUP', 'PROXY_INCOME', 'MOB', 'FLAG_FEMALE',
             'PROMO_CHANNEL', 'BIRTH_DATE'],
            dtype='object')
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Data Cleaning

Data Cleaning Steps and Considerations : Data Manipulation

- We make a new group column for “PROMO_TXN_CNT_L6M” to know how much our customer making transaction with respond on merchant

```
[164] # Here we also decided to assign it into different column: promo_experience using .loc[row,column]
      dfp_dc.loc[(a == 1) | (a == 2) | (a == 3)  , 'promo_experience'] = 'Junior'
      dfp_dc.loc[(a == 4) | (a == 5) | (a == 6) | (a=='7'), 'promo_experience'] = 'Middle'
      dfp_dc.loc[(a == 8) | (a == 9) | (a == 10), 'promo_experience'] = 'Senior'
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Data Cleaning

Data Cleaning Steps and Considerations : Check Duplicate

Based on Account id it has unique number so one customer has each one account id and we checking duplicate that we don't found any duplicated data

```
[648] # Check duplcate data
      dfp[dfp['ACCOUNT_ID'].duplicated()]
```

ACCOUNT_ID	MCC	MERCHANT_NAME	AVG_TXN_

0 rows × 22 columns

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Exploratory Data Analysis

What are your business questions that you can formulate and what are the answers? From Descriptive Statistics - Numerical Columns we know that insight

- Average sales amount per transactions attributed to each account over the past 6 months is 127.48 euro, the median 110 euro and the maximum 2000 euro without promo
- Average sales amount per transactions attributed to each account in the past 6 months in response to promo on merchant is 121.75 euro, the median 100 euro
- Maximum count of number of transaction in the past 6 months in response to promo on merchant are 10 transaction
- Maximum of Number of e-mail/SMS about promo received by each account in the past 6 months are 21 per customer

Exploratory Data Analysis

What are your business questions that you can formulate and what are the answers? From Descriptive Statistics - Non Numeric Columns we know that insight

- We have 112433 **user**
- Most **selling merchant** are REVOSHOP
- Most **profitable and creditworthy customers** are Customer Group "**E**"
- Most Frequent **Active users** are 104550 times
- Most **Account Holder** are **Female**
- Most **Promo are received via SMS**
- Most **Promo transaction** are in **Junior** level which mean **most of customer are using promo last six month** just **once in a while**

Exploratory Data Analysis

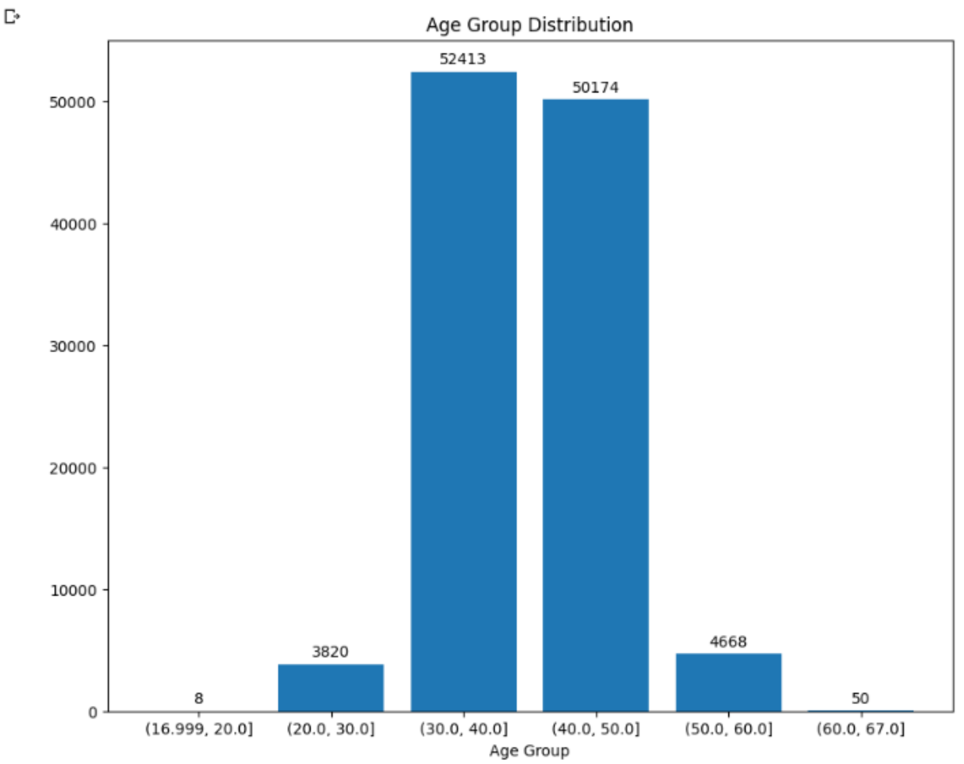
- What are your business questions that you can formulate and what are the answers? From DescriptMost of Account holder are Born in 1983-09-13
- Our Older Account Holder are Born in 1956-03-18
- Youngest Account Holder are Born in 2005-09-20

ive Statistics - Date Column we know that insight

Exploratory Data Analysis

Insight : from From customer Age group we know account holder on RevoBank based on dataset that from Age 30-50 years old with average is 40.5 years

Recommendation : Probably we could make some promotion into youth generation since they are the lowest based age group distribution



https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

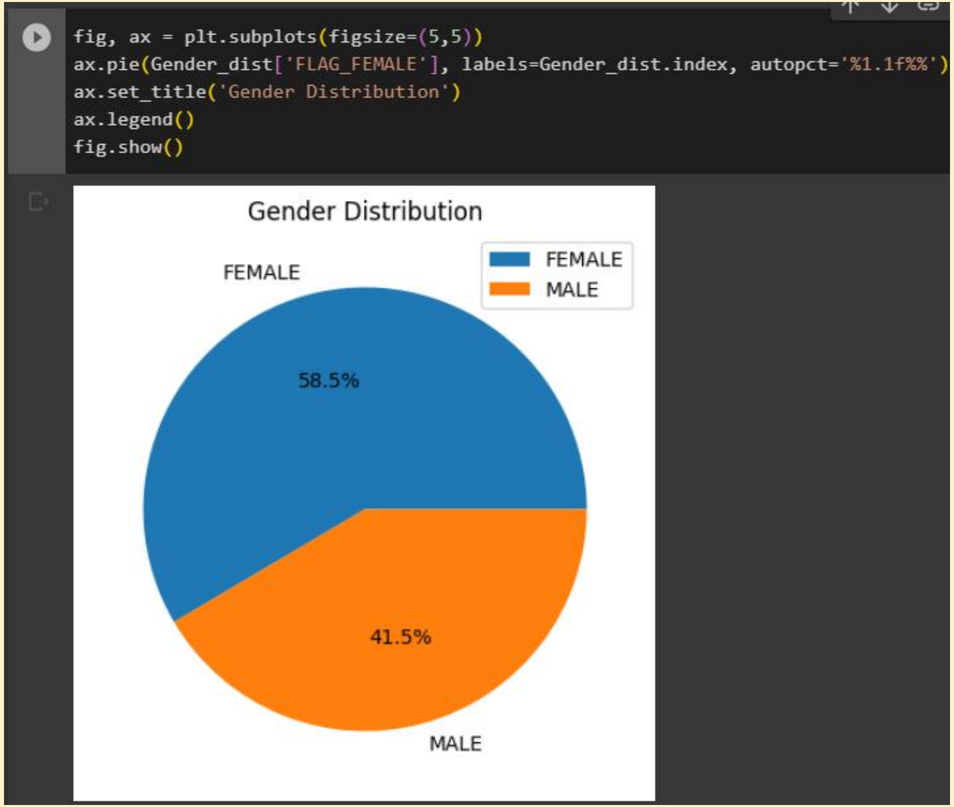
Exploratory Data Analysis

By Gender Distribution as we know most Account Holder on our RevoBank was Female indicate

1 : Female

0 : Male

Recommendation: as we know since female more **tend to try new promotion-things** this could be recommendation to offer as account holder to female rather than Male



Exploratory Data Analysis

Based on MOB distribution by Age Group, age group 17-20 years account lifetime around 76 month on average, and other age has lifetime 101 months on average

```
# MOB Distribution of Customers by Age Group
dfp_eda['MOB'].groupby(dfp_eda['AGE_GROUP']).mean()
```

```
AGE_GROUP
(16.999, 20.0]     76.25
(20.0, 30.0]      101.39
(30.0, 40.0]      101.27
(40.0, 50.0]      101.69
(50.0, 60.0]      101.94
(60.0, 67.0]      101.42
Name: MOB, dtype: float64
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Exploratory Data Analysis

Insights : Total sales have been generated in RevoShop over the past 6 months

```
[680] # calculate total sales last 6 months period
      Total_Sales_L6M = (dfp_eda['AVG_TXN_AMT_L6M']*dfp_eda['TXN_CNT_L6M']).sum()
      Total_Sales_L6M

      58077848.9
```

As we know based on data **58,077,849** euros are total sales RevoShop to RevoBank over **past 6 months**, and **35,208,915** euros are **Total RevoShop Sales in response to Promotion over past 6 months**

Recommendation : **this should be awareness almost total Revoshop sales in response to Promotion is reaching total sales without Promotion**

```
# calculate total sales in response to promotion during in the last 6 months
Total_Sales_Promo_L6M = (dfp_eda['AVG_PROMO_TXN_AMT_L6M']*dfp_eda['PROMO_TXN_CNT_L6M']).sum()
Total_Sales_Promo_L6M

35208915.9
```

Exploratory Data Analysis

Insights :

- in the last 6 months, there is 61% of total sales is attributed to promotion
- in the lifetime of account, there is 59% of total sales is Attributed to promotion

We can conclude that this is big problem and should find a way cutting off the promotion

```
[682] Percent_Sales_Promo = (Total_Sales_Promo_L6M/ Total_Sales_L6M).round(2)
      Percent_Sales_Promo
```

```
0.61
```

```
# Total sales Life time sales
Total_Sales_LTM = (dfp_eda['AVG_TXN_AMT_LTM']*dfp_eda['TXN_CNT_LTM']).sum()
Total_Sales_LTM

# Total Lifetime Sales
Total_Sales_Promo_LTM = (dfp_eda['AVG_PROMO_TXN_AMT_LTM']*dfp_eda['PROMO_TXN_CNT_LTM']).sum()
Total_Sales_Promo_LTM

# Divide the Total Sales in response to promotion and total sales during 6 months
Percent_Sales_Promo_LTM = (Total_Sales_Promo_LTM / Total_Sales_LTM).round(2)
Percent_Sales_Promo_LTM
```

```
0.59
```

https://colab.research.google.com/drive/1D502YZ5YRt7M_SxhS4XQ1bR71R300bAS?usp=drive_link

Exploratory Data Analysis

Total Customer that promo-sensitive

Insights :

- Based on data above during account's last 6 months there are 71,790 promo-sensitive customers, and there are 74,667 promo-sensitive during lifetime tenure
- VS between Lifetime tenure and last 6 months account's that promo-sensitive customers only 3,85% differences

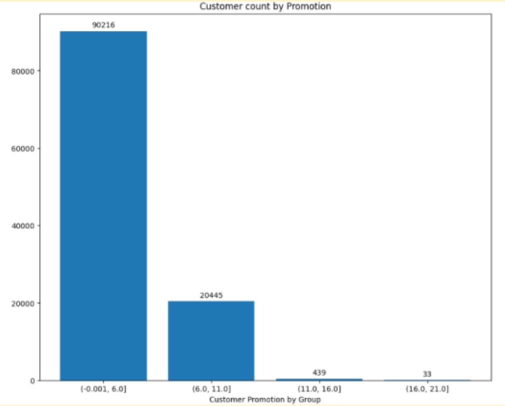
Exploratory Data Analysis

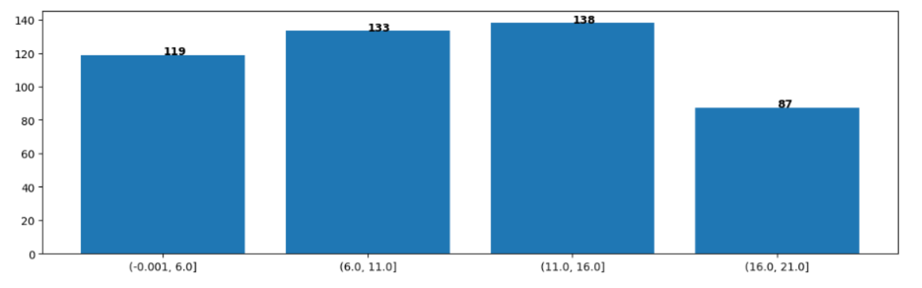
Correlation between count of customer receiving e-mail and SMS with sales Performance

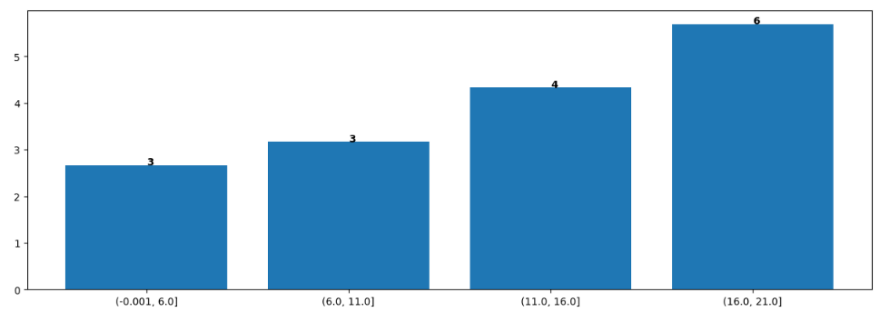
Insight :

- From **90,216 customer**s they only receive **1-6 promo by e-mail or SMS**
- Based data above we can conclude that **bigger number of promotion number average transaction was 11-16 transaction**
- Based on data above there is different relationship pattern between **number of e-mail/SMS with sales performance and also different pattern in number of promo received with average sales amount and number transaction**

Recommendation : since massive buyer promo was received by email or SMS we can cut the promo since it has no limitation of using promo or maybe we could make minimum requirement to using the promo so we can cut off the promotion spend







End of Milestone 1
Intermediate

Clustering Analysis

I rather choose K-means Clustering method because

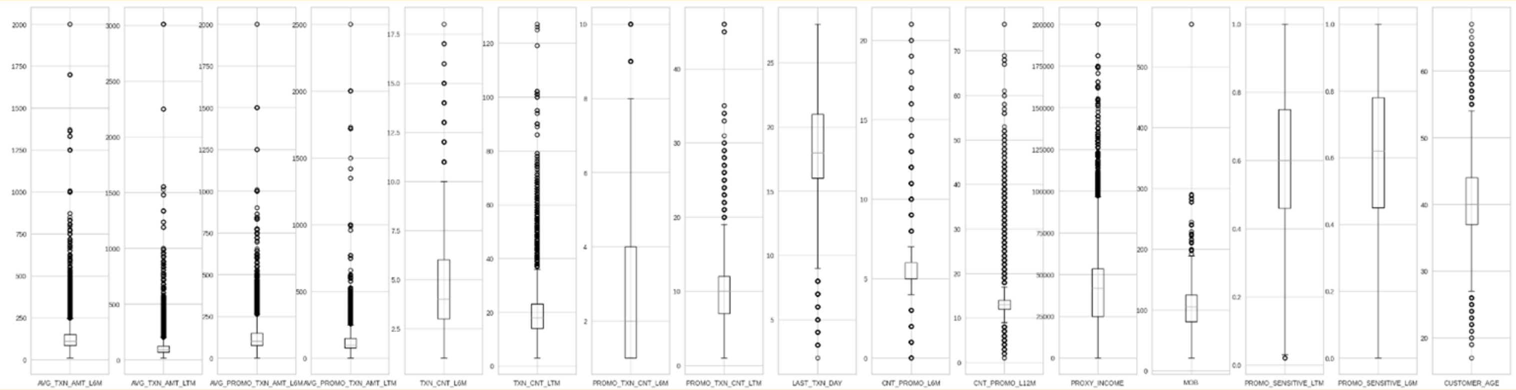
- RFM segmentation can’t use Age, MOB, and Gender
- RFM segmentation not that accurate because data is highly skewed, which lead to bias interpretation
- K-means help to segment customer on any various similar data type

Clustering Analysis

K-means clustering cleaning detect and cleaning outlier

Insights:

- From Plot most Outlier are on the upper bound
- From 16 Plot there is identical 9 plot pattern

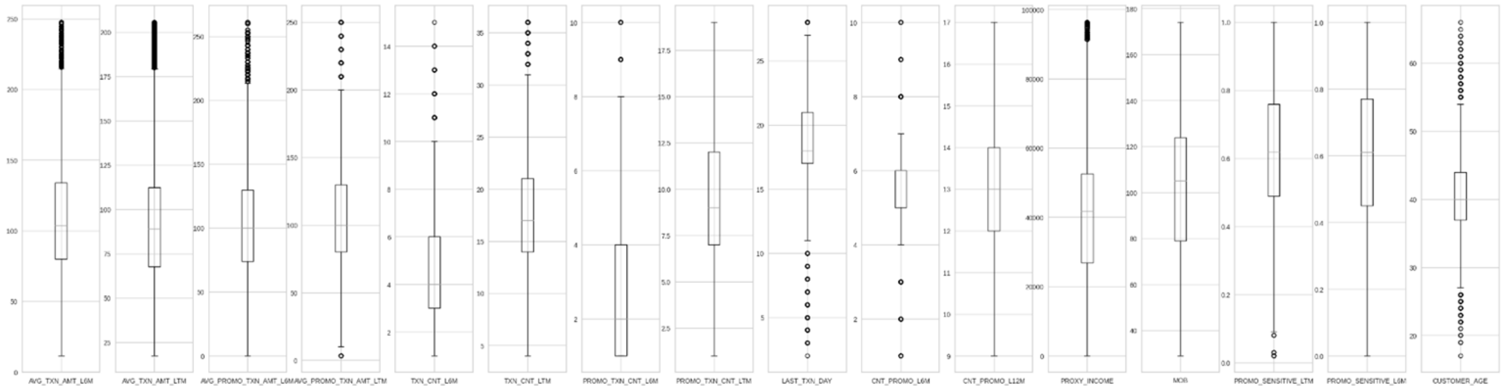


Clustering Analysis

K-means clustering cleaning detect and cleaning outlier Clean 1

Insights:

- Total Outlier : **111133 - 76098 = 35.035 rows**

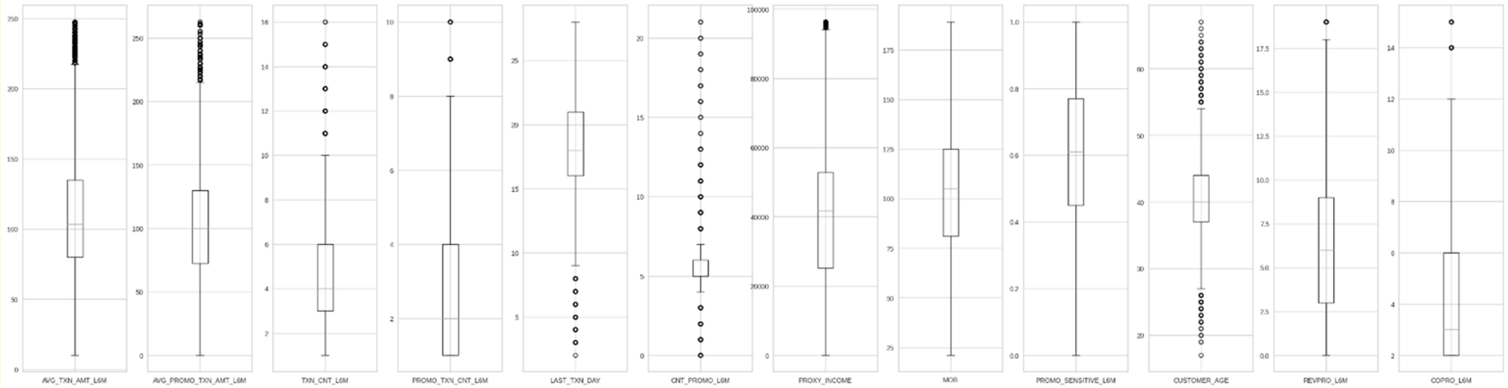


Clustering Analysis

K-means clustering cleaning detect and cleaning outlier Clean final

Insights:

- Total Outlier removed : **111333-99596 = 11.537 rows removed**

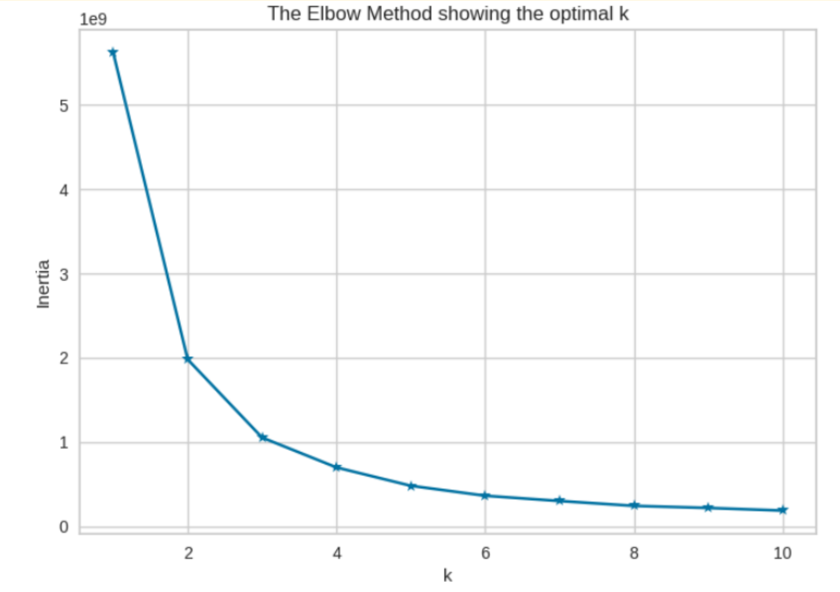


Cluster Analysis

Clustering Techniques : Elbow Method and silhouette

Insight :

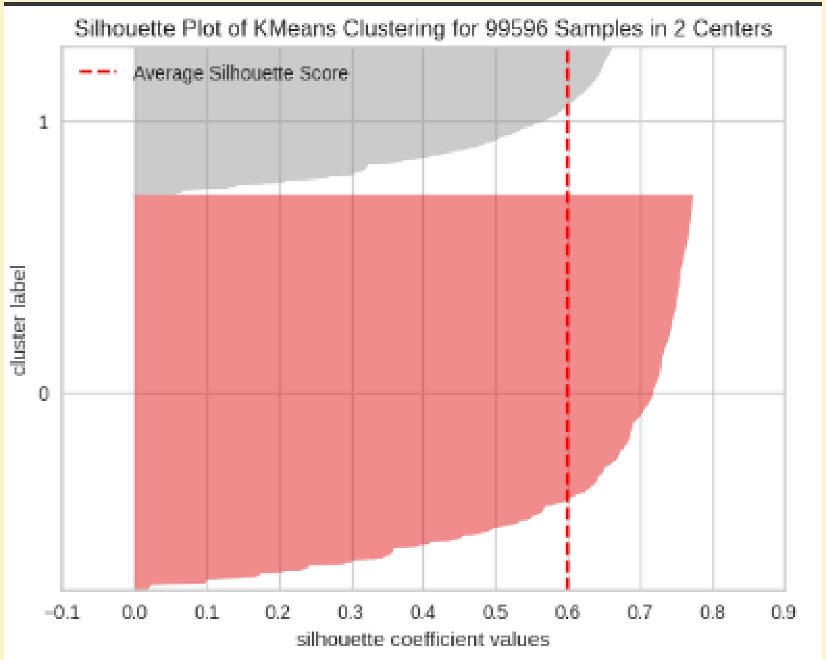
- Based on plot Elbow Method have potential on K=2 and K=3
- Highest silhouette is K=2
- Also need silhouette plot.

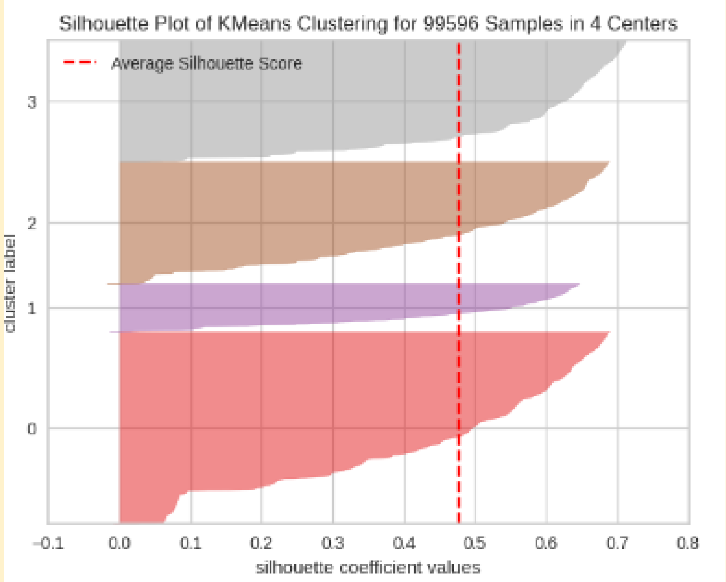


KMeans

```
KMeans(n_clusters=4, n_init='auto', random_state=1000)
```

Cluster Analysis





Insight silhouette diagram :

- K=2 has highest silhouette diagram score (0.6) but cannot interpreted as business-wise, because >50% customers are in the cluster 2
- K=4 has second highest score (0.477) and can be interpreted as business-wise

Interpreting Cluster Result

index	0	1	2
AVG_TXN_AMT_L6M	96.06216190238182	148.1865311102074	118.23797120344652
AVG_PROMO_TXN_AMT_L6M	90.45516445913994	144.6668483123221	113.8852502692591
TXN_CNT_L6M	3.4331109786808667	6.7111020740138265	5.208009750014171
PROMO_TXN_CNT_L6M	2.119071877583191	3.975193167954453	3.218439997732555
LAST_TXN_DAY	18.262788596474365	18.062627084180562	18.60880902443172
CNT_PROMO_L6M	5.307080105346124	6.311590077267182	6.040842355875517
PROXY_INCOME	37916.081854706936	35274.606832045545	37023.00606541579
MOB	106.02314538918472	92.58365189101261	98.12876254180603
PROMO_SENSITIVE_L6M	0.6228476133720371	0.5790711671411143	0.6010594637492206
CUSTOMER_AGE	40.52467367692574	40.54957299715331	40.503231109347546
SALE_L6M	280.71398911936024	923.5053273688492	544.2171929028966
REVPRO_L6M	4.063745938983833	12.696624644164295	7.87123745819398
COPRO_L6M	3.3249581883542554	5.917446116307442	4.807097103338813

- Cluster 0 : **Lowest** sales amount , **Lowest** Frequency transaction, **Oldest** tenure, **Highest** promo sensitive, **Lowest** Promo revenue, **Lowest** cost promo users
- Cluster 1 : **Highest** sales amount, **Highest** Frequency transaction, **Youngest** tenure, **Lowest** Promo sensitive, **Highest** promo revenue, **Highest** cost promo users
- Cluster 2 : **Middle** sales amount, **Middle** Frequency transaction, **Middle** tenure, **Middle** promo sensitive, **Middle** promo revenue, **Highest** cost promo users

Interpreting Cluster Result and Business Recommendations

Conclusion :

- Cluster 0 : This cluster where actually “older” account which only buy our product when we make campaign promo. They probably got our campaign from mobile app notification.
- Cluster 1 : This cluster type was good one because, they are type who always shop at REVOSHOP and they are new comer and has lowest promo sensitive from other cluster but still highest promo revenue and highest cost promo.
- Cluster 2 : This cluster has really moderate level of our cluster they not really much have transaction at Revoshop and not biggest promo sensitive and revenue, but they they buy always with promo.

Recommendations :

As we know from Interpreting cluster Result we know that our most account Holder from every cluster was **Female**, and we should focus on **Customer Cluster 1** as we know they are new comer but from the clustering they are probably ***actual*** Loyal Customer for us. For further action **we will make promotion transaction more limited** than before or we make **promotion only some feminine product like skin care, cosmetic, etc.** so we could <u>reduce</u> upcoming cost of promotion.

End of Milestone 2
Intermediate

Advanced Assignment Python

(Insert your name here)

Data Preparation for Propensity Model

Data Cleaning Steps and Considerations.

Propensity Model Training & Evaluation

Build a model to predict the target variable using the training dataset, and evaluating the model.