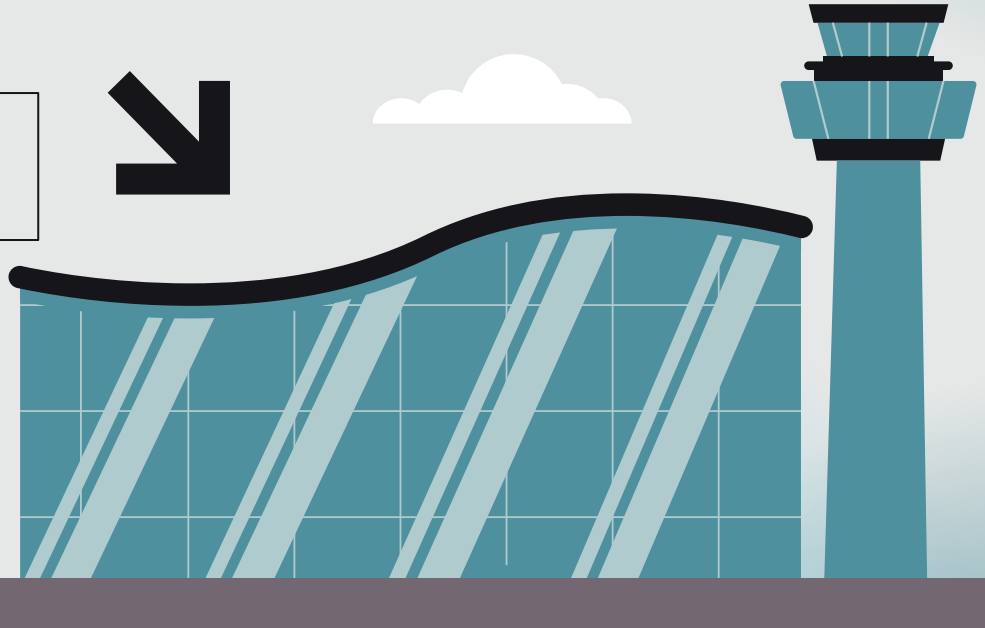
Aviation Customers Clustering

Unsupervised Learning Report

Gabrielle Maureen



LIBRARY AND DATASET

```
[ ] import warnings
    warnings.filterwarnings('ignore')

    import numpy as np
    import pandas as pd
    import seaborn as sns
    from scipy import stats
    from scipy.stats import uniform
    import matplotlib.pyplot as plt
    import matplotlib.patches as patches
    from matplotlib.patches import Ellipse
    from matplotlib import rcParams
    from yellowbrick.cluster import SilhouetteVisualizer
    %matplotlib inline
    from sklearn.preprocessing import MinMaxScaler, StandardScaler
    from sklearn.metrics import silhouette_score
    from sklearn.decomposition import PCA

    from sklearn.cluster import KMeans
    import gdown

    print('numpy version : ',np.__version__)
    print('pandas version : ',pd.__version__)
    print('seaborn version : ',sns.__version__)
```

```
numpy version :  1.23.5
pandas version :  1.5.3
seaborn version :  0.12.2
```

IMPORT PYTHON LIBRARIES

1

```
[ ] !gdown --id 14G4xOWK5e-QQ9S7GmBwULChNdeJZXs2U
```

```
/usr/local/lib/python3.10/dist-packages/gdown/cli.py:121: FutureWarning: Option `--id` was de
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=14G4xOWK5e-QQ9S7GmBwULChNdeJZXs2U
To: /content/flight.csv
100% 8.94M/8.94M [00:00<00:00, 30.3MB/s]
```

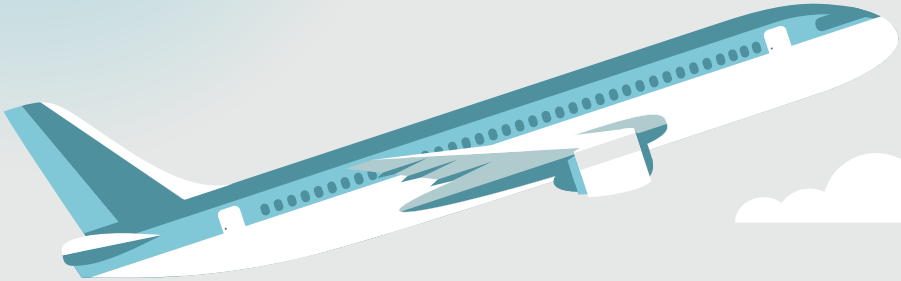
```
[ ] # import csv
    df = pd.read_csv("flight.csv")
    df.sample(5)
```

	MEMBER_NO	FFP_DATE	FIRST_FLIGHT_DATE	GENDER	FFP_TIER	WORK_CITY	WORK_PROVINCE	WOR
44453	20899	11/7/2007	2/22/2009	Male	4	beijing	beijing	
2552	57329	6/5/2008	12/31/2008	Male	6	zhengzhou	henan	
14808	11586	6/19/2007	6/28/2007	Male	4	guangzhou	guangdong	
44135	14498	6/14/2011	6/15/2011	Male	4	luochuan	shanxi	
42906	15217	1/26/2013	1/29/2013	Male	4	shenyang	liaoningsheng	

5 rows × 23 columns

DOWNLOAD DATASET AND LOAD IT

2



EXPLORATORY DATA ANALYSIS

MISSING VALUES & DUPLICATE VALUES

```
[5] # Checking the data incase there are NaN/Null values
    print(df.isnull().values.any())
    print(df.isna().sum())
```

```
True
MEMBER_NO               0
FFP_DATE                0
FIRST_FLIGHT_DATE       0
GENDER                  3
FFP_TIER                0
WORK_CITY            2269
WORK_PROVINCE        3248
WORK_COUNTRY           26
AGE                   420
LOAD_TIME               0
FLIGHT_COUNT            0
BP_SUM                  0
SUM_YR_1              551
SUM_YR_2              138
SEG_KM_SUM              0
LAST_FLIGHT_DATE        0
LAST_TO_END             0
AVG_INTERVAL            0
MAX_INTERVAL            0
EXCHANGE_COUNT          0
avg_discount            0
Points_Sum              0
Point_NotFlight         0
dtype: int64
```

There are 7 columns with missing values/ NaN/ NULL, i.e.,

- GENDER, 3 rows of missing values.
- WORK_CITY, 2269 rows of missing values.
- WORK_PROVINCE, 3248 rows of missing values.
- WORK_COUNTRY, 26 rows of missing values.
- AGE, 420 rows of missing values.
- SUM_YR_1, 551 rows of missing values.
- SUM_YR_2, 138 rows of missing values.

```
[ ] # Checking the data incase there are duplicates values
    print(df.duplicated().any())
    print(df.duplicated().sum())
```

```
False
0
```

```
[ ] df.duplicated(subset=['MEMBER_NO']).sum()
```

```
0
```

There's no duplicates values found.

DATASET & DATA TYPES

```
[ ] # Checking the datatype of every columns
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62988 entries, 0 to 62987
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   MEMBER_NO          62988 non-null  int64
 1   FFP_DATE           62988 non-null  object
 2   FIRST_FLIGHT_DATE  62988 non-null  object
 3   GENDER             62985 non-null  object
 4   FFP_TIER           62988 non-null  int64
 5   WORK_CITY          60719 non-null  object
 6   WORK_PROVINCE      59740 non-null  object
 7   WORK_COUNTRY       62962 non-null  object
 8   AGE                62568 non-null  float64
 9   LOAD_TIME          62988 non-null  object
 10  FLIGHT_COUNT       62988 non-null  int64
 11  BP_SUM             62988 non-null  int64
 12  SUM_YR_1           62437 non-null  float64
 13  SUM_YR_2           62850 non-null  float64
 14  SEG_KM_SUM         62988 non-null  int64
 15  LAST_FLIGHT_DATE   62988 non-null  object
 16  LAST_TO_END        62988 non-null  int64
 17  AVG_INTERVAL       62988 non-null  float64
 18  MAX_INTERVAL       62988 non-null  int64
 19  EXCHANGE_COUNT     62988 non-null  int64
 20  avg_discount       62988 non-null  float64
 21  Points_Sum         62988 non-null  int64
 22  Point_NotFlight    62988 non-null  int64
dtypes: float64(5), int64(10), object(8)
memory usage: 11.1+ MB
```

Column	Description	Datatype
MEMBER_NO	Unique Number of ID Member	well-suited
FFP_DATE	Frequent Flyer Program Join Date	timestamp (datetimes) is more suitable
FIRST_FLIGHT_DATE	Date of the first flight	timestamp (datetimes) is more suitable
GENDER	The gender of member holder	well-suited
FFP_TIER	Tier of Frequent Flyer Program	well-suited
WORK_CITY	Origin City	well-suited
WORK_PROVINCE	Origin Province	well-suited
WORK_COUNTRY	Origin Country	well-suited
AGE	The age of the member holder	well-suited, but since age usually in absolute form, integer is more suitable
LOAD_TIME	The date of this data being taken	timestamp (datetimes) is more suitable
FLIGHT_COUNT	The customer's number of flight	well-suited
BP_SUM	Planned Trip	well-suited
SUM_YR_1	Fare Revenue	well-suited
SUM_YR_2	Votes Prices	well-suited
SEG_KM_SUM	The number of flight distance (km) that has been traveled	well-suited
LAST_FLIGHT_DATE	Date of the last flight	timestamp (datetimes) is more suitable
LAST_TO_END	The duration from last flight to the next flight that has been reserved	well-suited
AVG_INTERVAL	Average the time distance	well-suited
MAX_INTERVAL	Maximum the time distance	well-suited
EXCHANGE_COUNT	The count of the exchange	well-suited
avg_discount	Average discount gained by the customers	well-suited
Points_Sum	Total points gained by the customers	well-suited
Point_NotFlight	Total points that aren't being used by the customers	well-suited

- Flight data frame is composed of 23 columns and 62988 rows.
- This dataframe has 8 categorical/object columns and 15 numerical columns.

STATISTICAL DESCRIPTIVE

```
[ ] # Grouping the columns based on the datatypes
    nums = ['int64', 'int32', 'int16', 'float64', 'float32', 'float16']
    nums = df.select_dtypes(include=nums)
    nums = nums.columns

    cats = ['object']
    cats = df.select_dtypes(include=cats)
    cats = cats.columns
```

```
[ ] # Checking the statistical descritive summary of the numeric columns
    df[nums].describe()
```

	MEMBER_NO	FFP_TIER	AGE	FLIGHT_COUNT	BP_SUM	SUM_YR_1	SUM_YR_2	SEG_KM_SUM	LAST_TO_END	AVG_INTERVAL	MAX_INTERVAL	EXCHANGE_COUNT	avg_discount	Points_Sum	Point_NotFlight
count	62988.000000	62988.000000	62568.000000	62988.000000	62988.000000	62437.000000	62850.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.000000	62988.0000	62988.000000
mean	31494.500000	4.102162	42.476346	11.839414	10925.081254	5355.376064	5604.026014	17123.878691	176.120102	67.749788	166.033895	0.319775	0.721558	12545.7771	2.728155
std	18183.213715	0.373856	9.885915	14.049471	16339.486151	8109.450147	8703.364247	20960.844623	183.822223	77.517866	123.397180	1.136004	0.185427	20507.8167	7.364164
min	1.000000	4.000000	6.000000	2.000000	0.000000	0.000000	0.000000	368.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000
25%	15747.750000	4.000000	35.000000	3.000000	2518.000000	1003.000000	780.000000	4747.000000	29.000000	23.370370	79.000000	0.000000	0.611997	2775.0000	0.000000
50%	31494.500000	4.000000	41.000000	7.000000	5700.000000	2800.000000	2773.000000	9994.000000	108.000000	44.666667	143.000000	0.000000	0.711856	6328.5000	0.000000
75%	47241.250000	4.000000	48.000000	15.000000	12831.000000	6574.000000	6845.750000	21271.250000	268.000000	82.000000	228.000000	0.000000	0.809476	14302.5000	1.000000
max	62988.000000	6.000000	110.000000	213.000000	505308.000000	239560.000000	234188.000000	580717.000000	731.000000	728.000000	728.000000	46.000000	1.500000	985572.0000	140.000000

```
[ ] # Checking the statistical descritive summary of the categorical columns
    df[cats].describe()
```

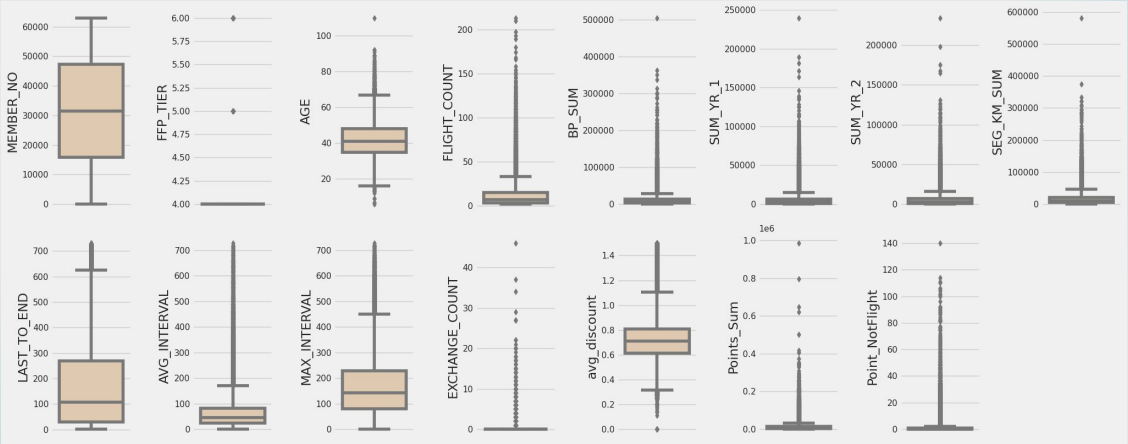
	FFP_DATE	FIRST_FLIGHT_DATE	GENDER	WORK_CITY	WORK_PROVINCE	WORK_COUNTRY	LOAD_TIME	LAST_FLIGHT_DATE
count	62988	62988	62985	60719	59740	62962	62988	62988
unique	3068	3406	2	3234	1165	118	1	731
top	1/13/2011	2/16/2013	Male	guangzhou	guangdong	CN	3/31/2014	3/31/2014
freq	184	96	48134	9386	17509	57748	62988	959

Numerical Columns

- MEMBER_NO, between mean and median don't have much difference, which indicates to normal distribution.
- FFP_TIER, between mean and median don't have much difference, which indicates to normal distribution.
- AGE, between mean and median don't have much difference, which indicates to normal distribution. The max value of Age reaching 110 years, the fact that it's might be outliers needed to be analyzed.
- FLIGHT_COUNT, mean > median, indicates to positive skewed.
- BP_SUM,mean > median, indicates to positive skewed.
- SUM_YR_1, mean > median, indicates to positive skewed.
- SUM_YR_2, mean > median, indicates to positive skewed.
- SEG_KM_SUM, mean > median, indicates to positive skewed.
- LAST_TO_END, mean > median, indicates to positive skewed.
- AVG_INTERVAL, mean > median, indicates to positive skewed.
- MAX_INTERVAL, mean > median, indicates to positive skewed.
- EXCHANGE_COUNT, between mean and median don't have much difference, which indicates to normal distribution.
- avg_discount, between mean and median don't have much difference, which indicates to normal distribution. The max value of average discount need to be analyzed, normally the range of discount is 0 to 100 (0.0 to 0.1) but in this dataframe the maximum discount is 150%, this could be errors.
- Points_Sum, mean > median, indicates to positive skewed.
- Point_NotFlight, mean > median, indicates to positive skewed.

Categorical Columns

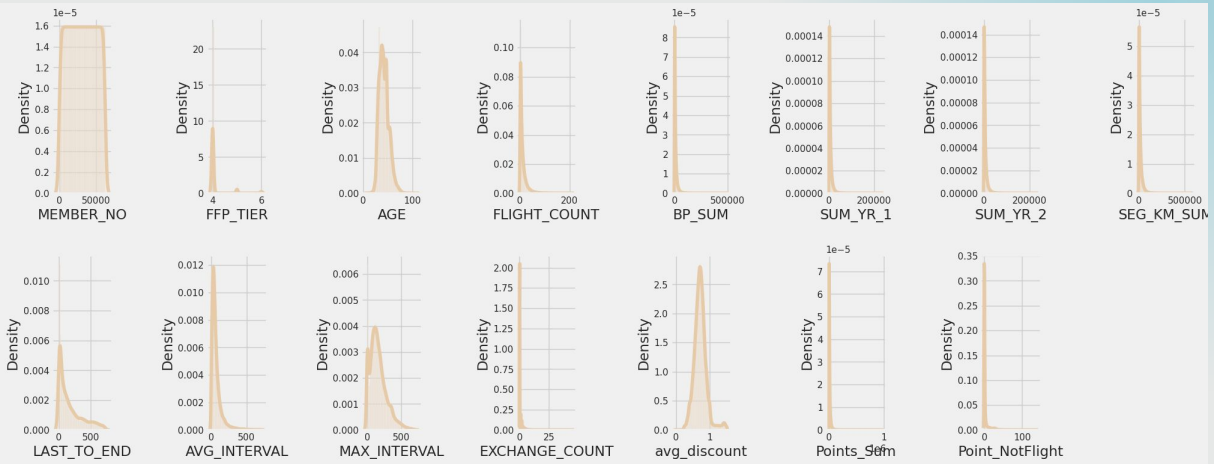
- FFP_DATE, consist 3068 unique values and the mode is 1/13/2011.
- FIRST_FLIGHT_DATE, consist 3406 unique values and the mode is 2/16/2013.
- GENDER, consist 2 unique values and the mode is Male.
- WORK_CITY, consist 3234 unique values and the mode is guangzhou.
- WORK_PROVINCE, consist 1165 unique values and the mode is guangdong.
- WORK_COUNTRY, consist 118 unique values and the mode is CN.
- LOAD_TIME, consist 1 unique values and the mode is 3/31/2014.
- LAST_FLIGHT_DATE, consist 731 unique values and the mode is 3/31/2014.



UNIVARIATE ANALYSIS

BOXPLOT

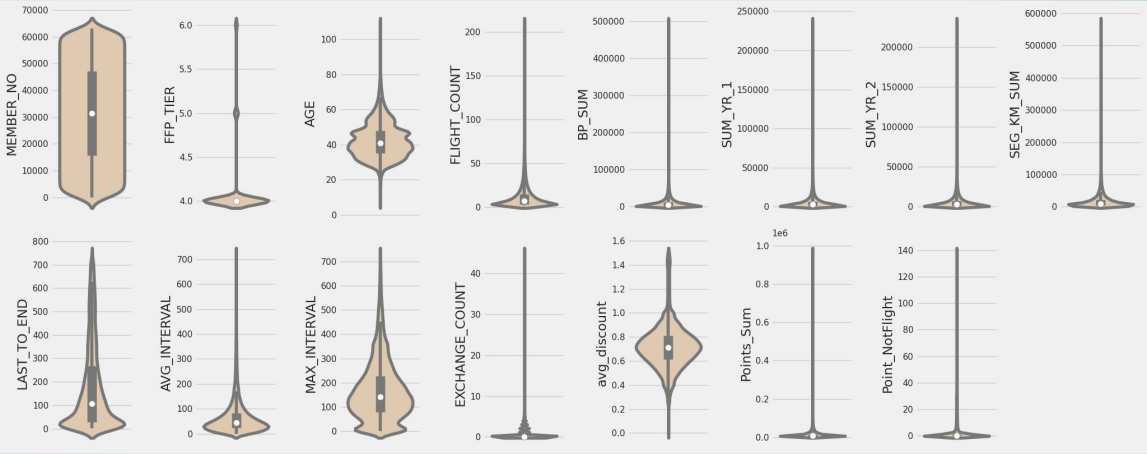
- MEMBER_NO, normal distribution.
- FFP_TIER, normal distribution, this column is a categorical column with 3 unique values (3 tiers) and the majority value is 4 hence this boxplot interpret value 5 and 6 as outliers.
- AGE, normal distribution with outliers.
- FLIGHT_COUNT, positive skewed with outliers.
- BP_SUM, positive skewed with outliers.
- SUM_YR_1, positive skewed with outliers.
- SUM_YR_2, positive skewed with outliers.
- SEG_KM_SUM, positive skewed with outliers.
- LAST_TO_END, positive skewed with outliers.
- AVG_INTERVAL, positive skewed with outliers.
- MAX_INTERVAL, positive skewed with outliers.
- EXCHANGE_COUNT, majority value is 0 which is almost +-80% of the data hence the other values are being assumed as outliers.
- avg_discount, normal distribution with outliers.
- Points_Sum, positive skewed with outliers.
- Point_NotFlight, There are 99 unique values but the 0 value is +-60% of the data dari data hence the other values are being assumed as outliers.



- MEMBER_NO, normal distribution.
- FFP_TIER, categorical column, trimodal distribution.
- AGE, normal distribution.
- FLIGHT_COUNT, positive skewed, outliers on the upper limit.
- BP_SUM, positive skewed, outliers on the upper limit.
- SUM_YR_1, positive skewed, outliers on the upper limit.
- SUM_YR_2, positive skewed, outliers on the upper limit.
- SEG_KM_SUM, positive skewed, outliers on the upper limit.
- LAST_TO_END, positive skewed, outliers on the upper limit.
- AVG_INTERVAL, positive skewed, outliers on the upper limit.
- MAX_INTERVAL, positive skewed, outliers on the upper limit.
- EXCHANGE_COUNT, outliers on the upper limit.
- avg_discount, normal distribution, outliers on the upper limit.
- Points_Sum, positive skewed, outliers on the upper limit.
- Point_NotFlight, positive skewed, outliers on the upper limit.

UNIVARIATE ANALYSIS

DISTRIBUTION PLOT

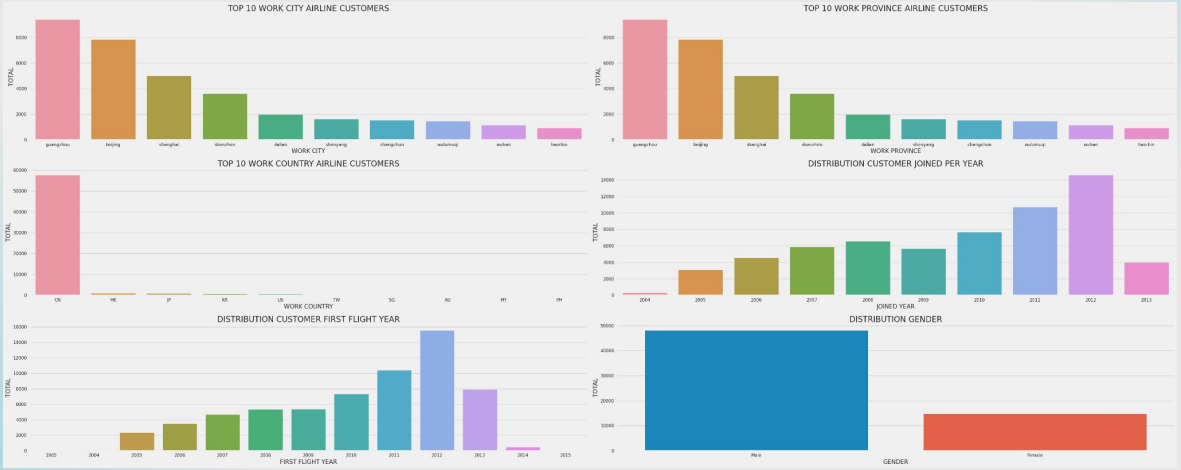


UNIVARIATE ANALYSIS

VIOLIN PLOT

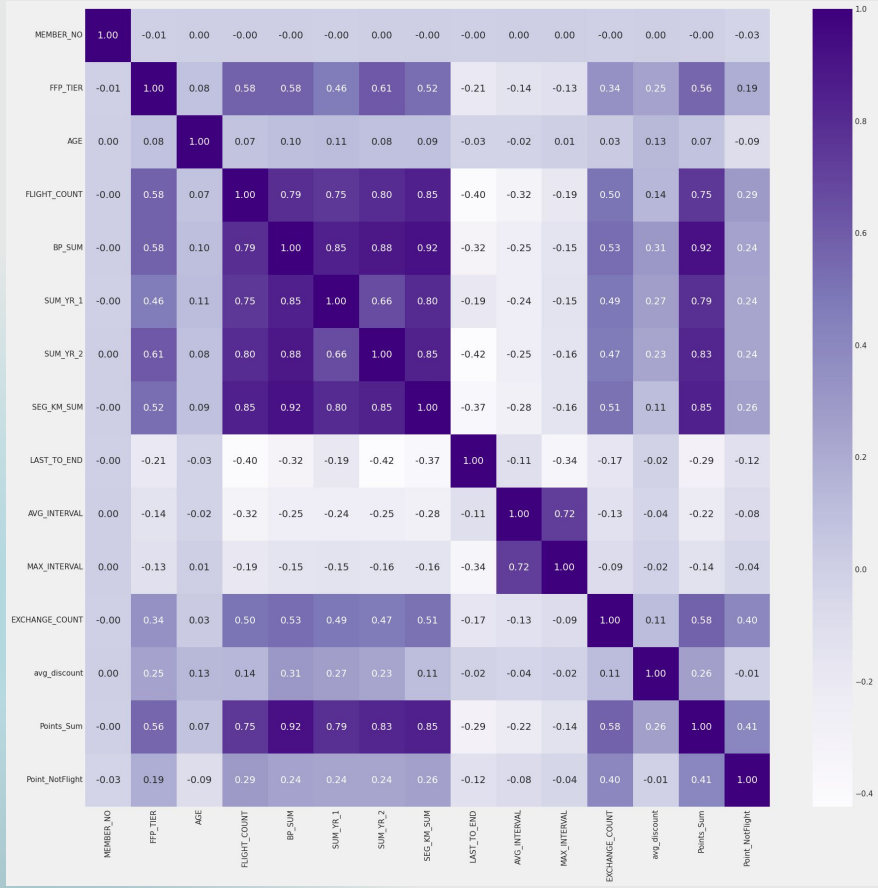
- MEMBER_NO, the distribution is spread from 0 to close to 70000.
- FFP_TIER, The distribution is spread from 4 - 6 with the majority of data being 4.
- AGE, the majority of data is spread from ages 25 - 70.
- FLIGHT_COUNT, the majority distribution is spread from 0 - 30.
- BP_SUM, the majority data is 0.
- SUM_YR_1, the majority data is 0.
- SUM_YR_2, the majority data is 0.
- SEG_KM_SUM, the majority of the data is spread from 0 - 3000.
- LAST_TO_END, the majority of the data is spread from 0 - 300.
- AVG_INTERVAL, the majority of the data is spread from 0 - 100.
- MAX_INTERVAL, the majority of the data is spread from 50 - 200.
- EXCHANGE_COUNT, the majority data is 0.
- avg_discount, the majority of data is spread from 0.4 - 1 and the discount data actually doesn't make sense if it is above 1 (100%) so it is necessary to clean outliers above 1.
- Points_Sum, the majority data is 0.
- Point_NotFlight, the majority data is 0.

UNIVARIATE ANALYSIS CATEGORICAL



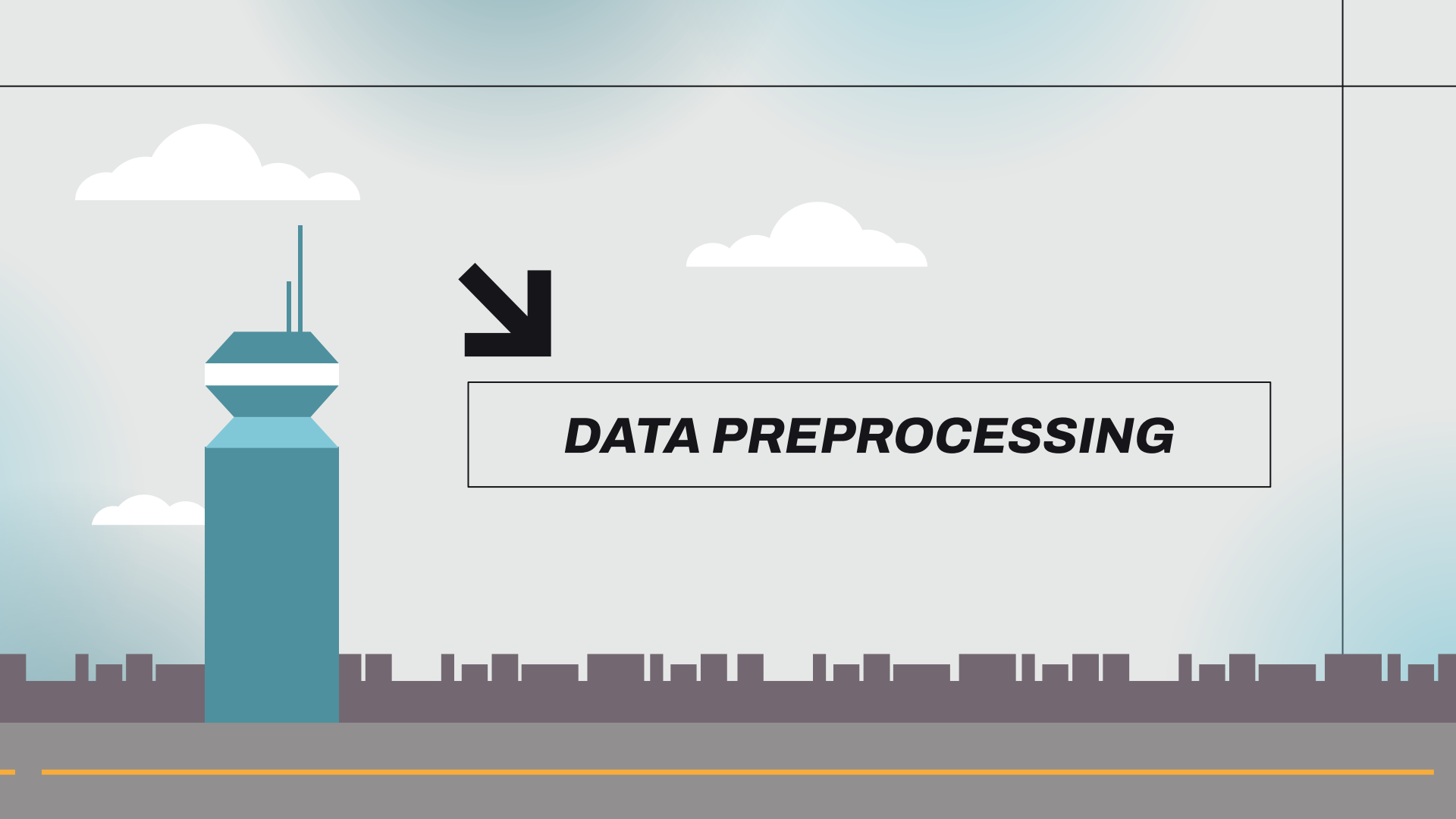
	Features	Unique
1	FIRST_FLIGHT_DATE	3406
3	WORK_CITY	3232
0	FFP_DATE	3068
4	WORK_PROVINCE	1165
7	LAST_FLIGHT_DATE	731
5	WORK_COUNTRY	118
2	GENDER	2
6	LOAD_TIME	1

- Column FFP_DATE, FIRST_FLIGHT_DATE, WORK_CITY, WORK_PROVINCE, WORK_COUNTRY, and LAST_FLIGHT_DATE are a categorical columns but has very diverse and complex unique values so it is better to drop it or do features extraction.
- Column GENDER need label encoding to facilitate the process of model learning.
- The LOAD_TIME column only has 1 value where the data is taken on the same day it is better not to use it for features.



MULTIVARIATE ANALYSIS HEATMAP

- AGE has a very weak correlation with almost every other columns.
- FLIGHT_COUNT, BP_SUM, SUM_YR_1, SUM_YR_2, SEG_KM_SUM and Points_Sum have multicollinearity (redundancy).
- AVG_INTERVAL and MAX_INTERVAL have multicollinearity (redundancy).


DATA PREPROCESSING

Handling Missing Values

Double-check for missing values along with special special characters such as "", "", "-", or "." .

```
{'. di WORK_PROVINCE', '. di WORK_CITY', '- di WORK_PROVINCE', '- di WORK_CITY'}
There are 1.06 % characters . in WORK_CITY
There are 1.48  % characters . in WORK_PROVINCE
There are 0.13 % characters - in WORK_CITY
There are 0.42  % characters - in WORK_PROVINCE
```

	Column	Total	Percentage
6	WORK_PROVINCE	3248	5.16
5	WORK_CITY	2269	3.60
12	SUM_YR_1	551	0.87
8	AGE	420	0.67
13	SUM_YR_2	138	0.22
7	WORK_COUNTRY	26	0.04
3	GENDER	3	0.00

Handling missing values process,

```
[12] # Drop columns for categorical columns and have very varied unique values (can cause the model to be too complex)
     df.drop(columns=['WORK_COUNTRY','WORK_CITY','WORK_PROVINCE'], axis=1, inplace=True)
```

```
[14] # Filling AGE column using fillna( pada kolom AGE
     df['AGE'].fillna(df['AGE'].median(), inplace=True)
```

```
[15] # Drop NaN in GENDER column because the percentage is unsignificant
     df.dropna(subset=['GENDER'], inplace=True)
```

```
[16] # Using the MICE (Multiple Imputation by Chained Equiation) method where missing values are filled in using a regression model
     from fancyimpute import IterativeImputer
     mice = IterativeImputer()
     x = ['SUM_YR_1','SUM_YR_2']
     df[x] = mice.fit_transform(df[x])
```

	Column	Total	Percentage
0	MEMBER_NO	0	0.0
1	FFP_DATE	0	0.0
18	Points_Sum	0	0.0
17	avg_discount	0	0.0
16	EXCHANGE_COUNT	0	0.0
15	MAX_INTERVAL	0	0.0
14	AVG_INTERVAL	0	0.0

Missing values have been handled.

Handling Outliers

Starting with the removal of discounts above 100% (invalid).

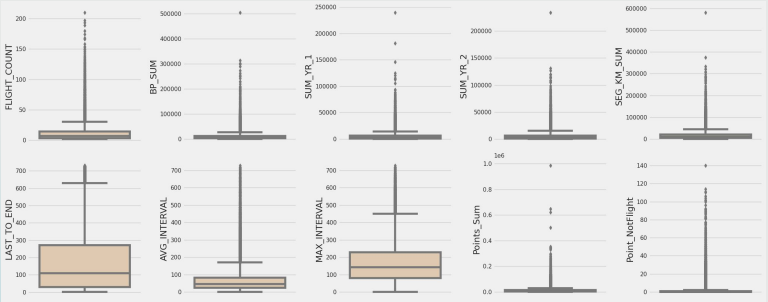
```
[19] # Remove avg_discount above 1 / 100%
     df = df[df['avg_discount']<=1]
     df[['avg_discount']].describe()
```

	avg_discount
count	60040.000000
mean	0.695878
std	0.144029
min	0.000000
25%	0.605630
50%	0.703391
75%	0.794527
max	1.000000

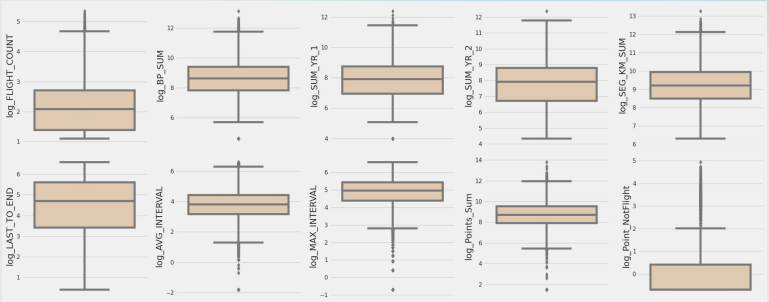
Log transformation is carried out for right skewed columns and followed by z-score.

```
[ ]  for i in right_skewed:
         df["log_"+i] = np.log(df[i] + (df[df[i] > 0][i].min() / 2))
```

Before



After



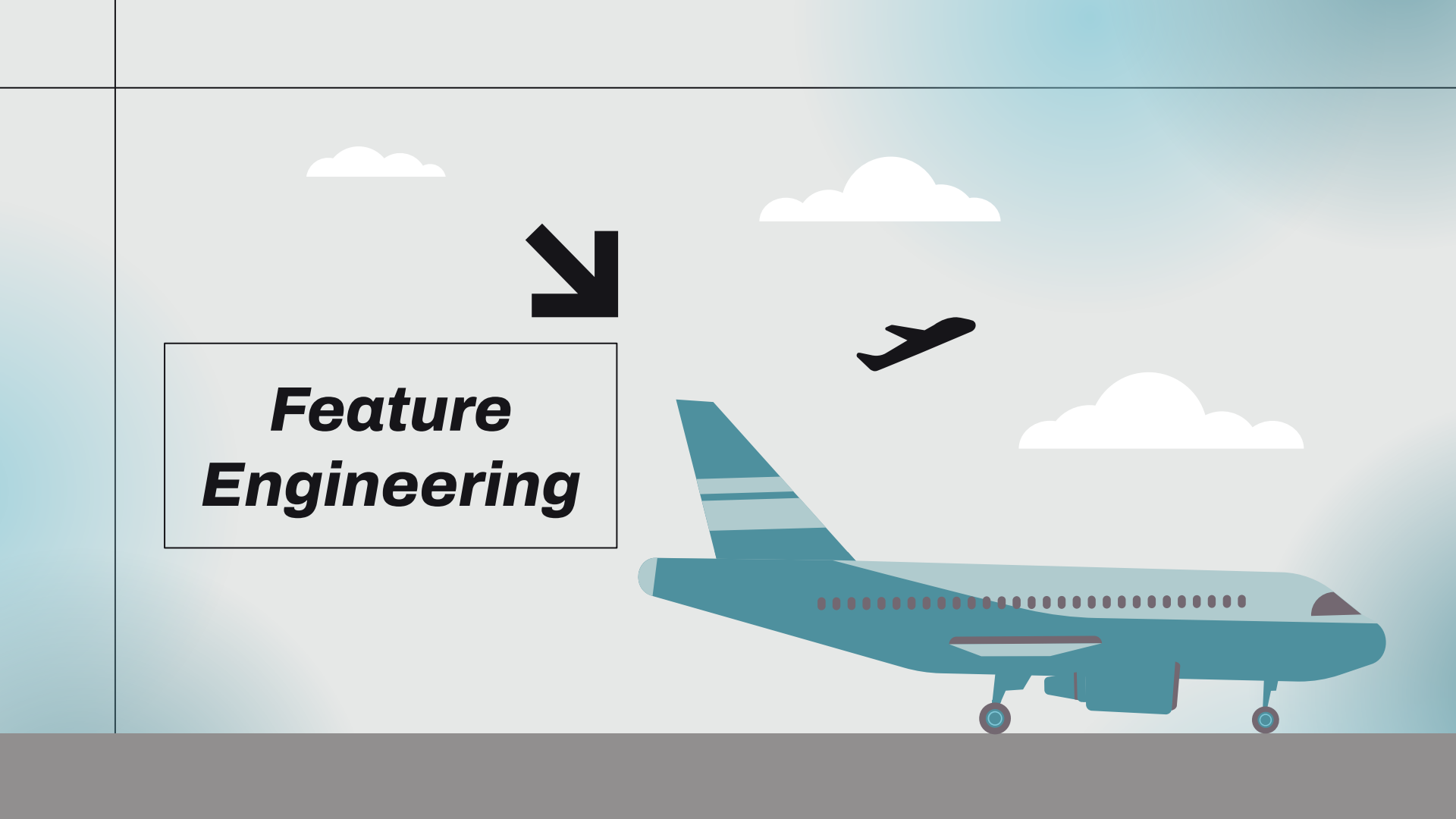

```
[23] # Handling Outliers using z-score
     zscores = ['MEMBER_NO', 'FFP_TIER', 'AGE', 'log_FLIGHT_COUNT', 'log_BP_SUM', 'log_SUM_YR_1', 'log_SUM_YR_2', 'log_SEG_KM_SUM', 'log_LAST_TO_END',
                'log_AVG_INTERVAL', 'log_MAX_INTERVAL', 'EXCHANGE_COUNT', 'avg_discount', 'log_Points_Sum', 'log_Point_NotFlight']

     print(f'Total Rows BEFORE Outlier Handling = {len(df)}')

     filtered_entries = np.array([True] * len(df))

     for i in zscores:
         zscore = abs(stats.zscore(df[i]))
         filtered_entries = (zscore < 3) & filtered_entries

     df = df[filtered_entries]

     print(f'Total Rows AFTER Outlier Handling = {len(df)}')

     Total Rows BEFORE Outlier Handling = 60040
     Total Rows AFTER Outlier Handling = 56028
```

Removing total 4012 datas, equal to 6.68 % datas.

FEATURE ENCODING

Perform One Hot Encoding for GENDER column using get_dummies() from pandas library.

```
[25] # Perform one hot encoding for the GENDER column
     for i in ['GENDER']:
         onehots = pd.get_dummies(df['GENDER'], prefix='Gen')
         df = df.join(onehots)

     df[['Gen_Female','Gen_Male']] = df[['Gen_Female','Gen_Male']].astype(int)
```


Feature
Engineering

Feature Extraction

Changing the feature should be datetimes.
Creating a new feature which is a feature for how long (duration) members have joined the FFP Membership.

```
[ ]  df['FFP_DATE'] = pd.to_datetime(df['FFP_DATE'])
     df['FIRST_FLIGHT_DATE'] = pd.to_datetime(df['FIRST_FLIGHT_DATE'])
     df['LOAD_TIME'] = pd.to_datetime(df['LOAD_TIME'])
     df['LAST_FLIGHT_DATE'] = pd.to_datetime(df['LAST_FLIGHT_DATE'], errors='coerce')
```

```
[28] # Creating the customers FFP Membership joined duration
     df['JOIN_DURATION'] = df['LOAD_TIME'] - df['FFP_DATE']
     df['JOIN_DURATION'] = df['JOIN_DURATION'].dt.days
```

Feature Selection

```
[ ]  Features = ['JOIN_DURATION', 'LAST_TO_END', 'FLIGHT_COUNT', 'SEG_KM_SUM', 'avg_discount']
     df_new = df[Features]
     df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 56028 entries, 114 to 62975
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   JOIN_DURATION  56028 non-null  int64
 1   LAST_TO_END    56028 non-null  int64
 2   FLIGHT_COUNT   56028 non-null  int64
 3   SEG_KM_SUM     56028 non-null  int64
 4   avg_discount   56028 non-null  float64
dtypes: float64(1), int64(4)
memory usage: 4.6 MB
```

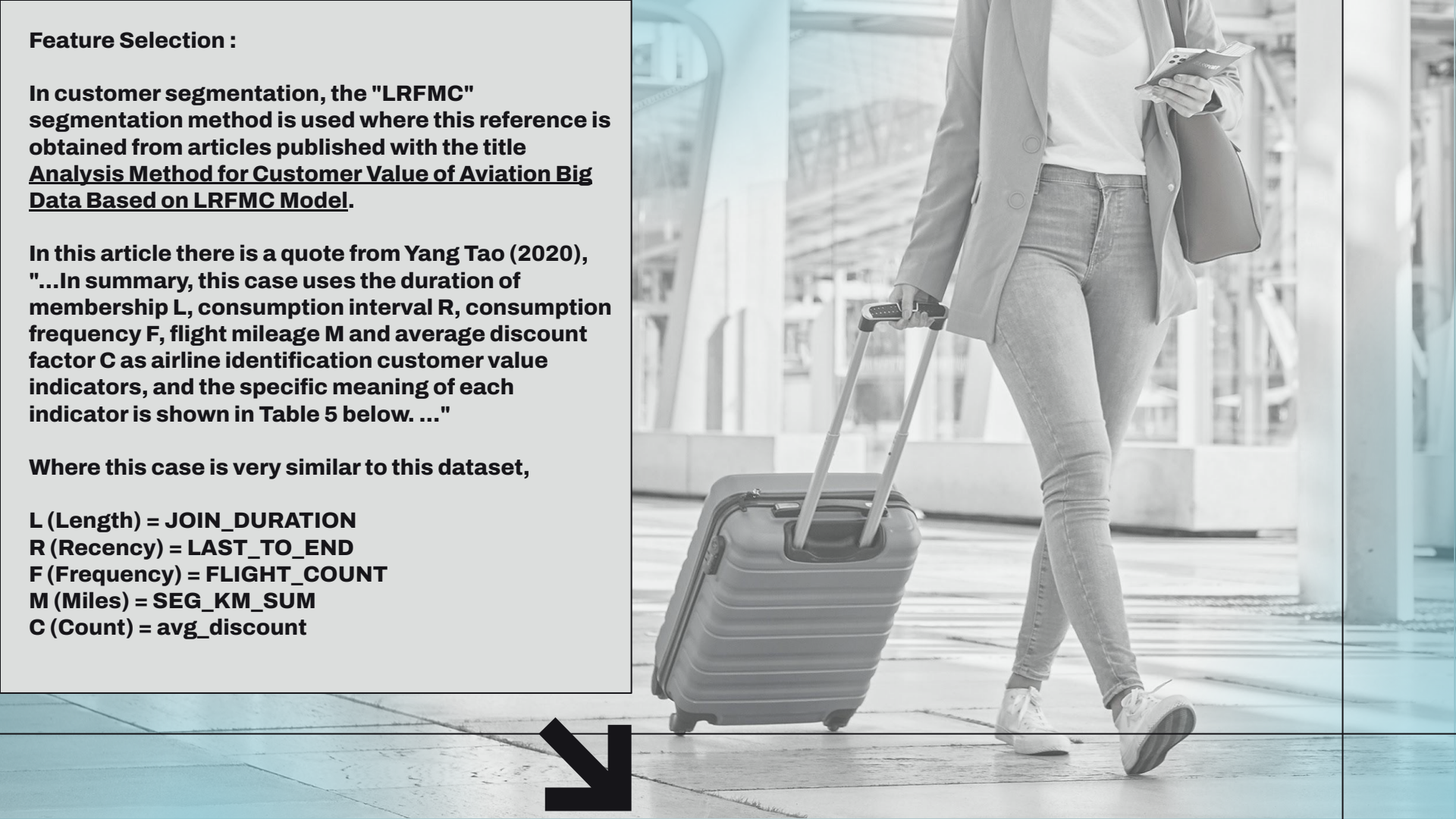
Feature Selection :

In customer segmentation, the "LRFMC" segmentation method is used where this reference is obtained from articles published with the title Analysis Method for Customer Value of Aviation Big Data Based on LRFMC Model.

In this article there is a quote from Yang Tao (2020), "...In summary, this case uses the duration of membership L, consumption interval R, consumption frequency F, flight mileage M and average discount factor C as airline identification customer value indicators, and the specific meaning of each indicator is shown in Table 5 below. ..."

Where this case is very similar to this dataset,

L (Length) = JOIN_DURATION
R (Recency) = LAST_TO_END
F (Frequency) = FLIGHT_COUNT
M (Miles) = SEG_KM_SUM
C (Count) = avg_discount



In addition to dealing with outliers, it is important to standardize before clustering. This is because if the range is not equal, clustering will assume that the larger scale is the most important feature.

```
[30] # Standardize using StandardScaler
     scaler = StandardScaler()
     scaler.fit(df_new)
     x_std = scaler.transform(df_new)
     df_std = pd.DataFrame(x_std, columns = df_new.columns)
```

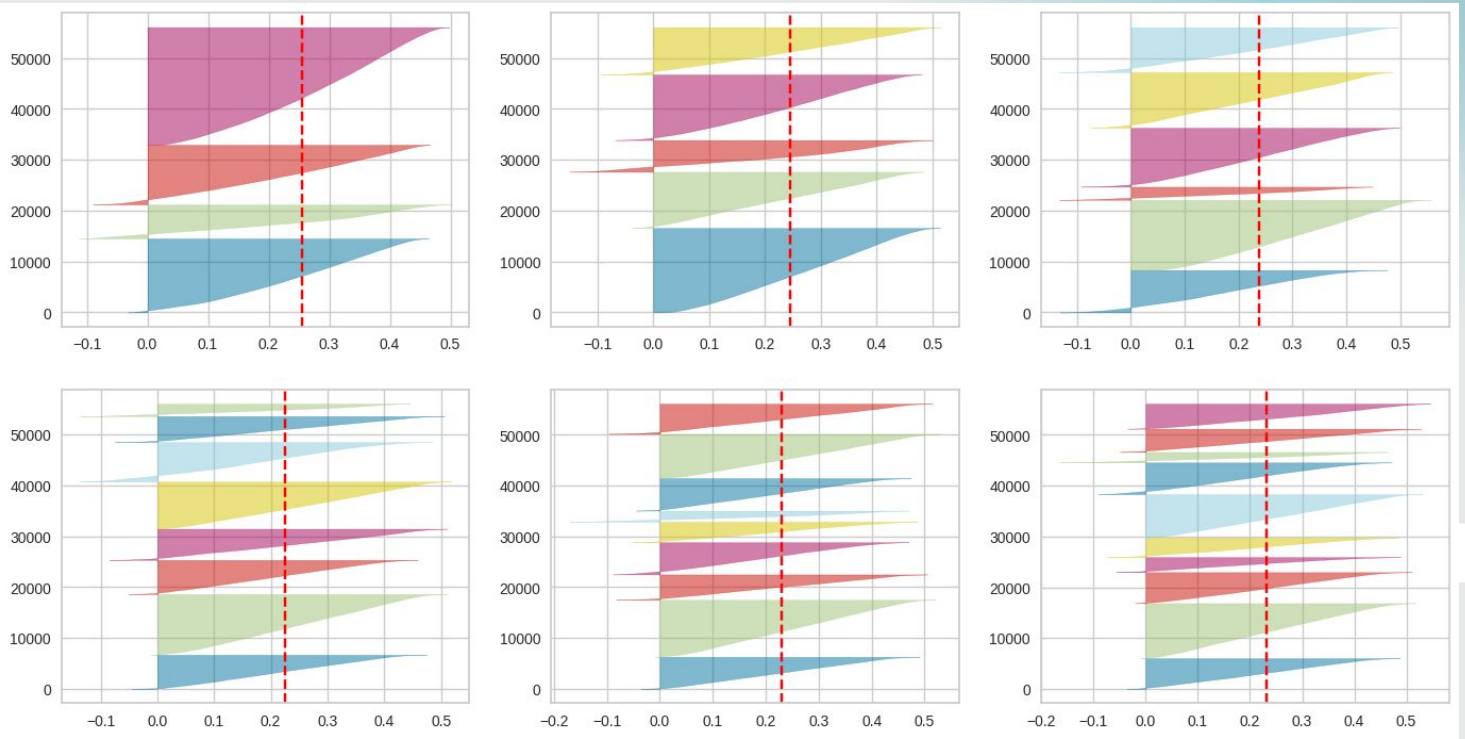
	JOIN_DURATION	LAST_TO_END	FLIGHT_COUNT	SEG_KM_SUM	avg_discount
1346	0.824639	-0.947959	3.015181	2.511587	0.690480
28088	1.355590	-0.522942	-0.517728	-0.187159	-1.161471
10475	-0.049093	-0.219358	0.297559	0.416616	0.556409
21414	0.668829	0.360211	-0.336553	-0.302647	1.381556
29754	-1.000730	0.183580	-0.427141	-0.323728	-0.679130

Modelling

↘ *K-Means*

Conduct Elbow Methods evaluation using inertia and Silhouette Score to find optimal clusters from the dataset.

```
[ ]  inertia = []

     for i in range(1,11):
         kmeans = KMeans(n_clusters = i, random_state = 42)
         kmeans.fit(df_std)
         inertia.append(kmeans.inertia_)
```



It can be seen from the visualization of the silhouette score and inertia of the cluster with the highest average and each cluster achieving an average and significant change in inertia is 5.

Implementation of K-Means Clustering Algorithm.

```
[ ]  kmeans = KMeans(n_clusters=5, random_state=42)
     kmeans.fit(df_std)
     df_std['Labels'] = kmeans.labels_
     df_std.sample(5)
```

	JOIN_DURATION	LAST_TO_END	FLIGHT_COUNT	SEG_KM_SUM	avg_discount	Labels
22290	-1.069047	1.353757	-0.245966	-0.042524	-0.669961	4
52929	-0.894060	2.733683	-0.789490	-0.849166	1.582294	4
23355	1.228545	-0.887242	0.478733	-0.207812	0.007156	3
22809	1.197383	-0.252476	-0.427141	-0.322628	1.124814	3
43703	-0.166550	-0.429107	-0.698903	-0.629740	-0.938234	1

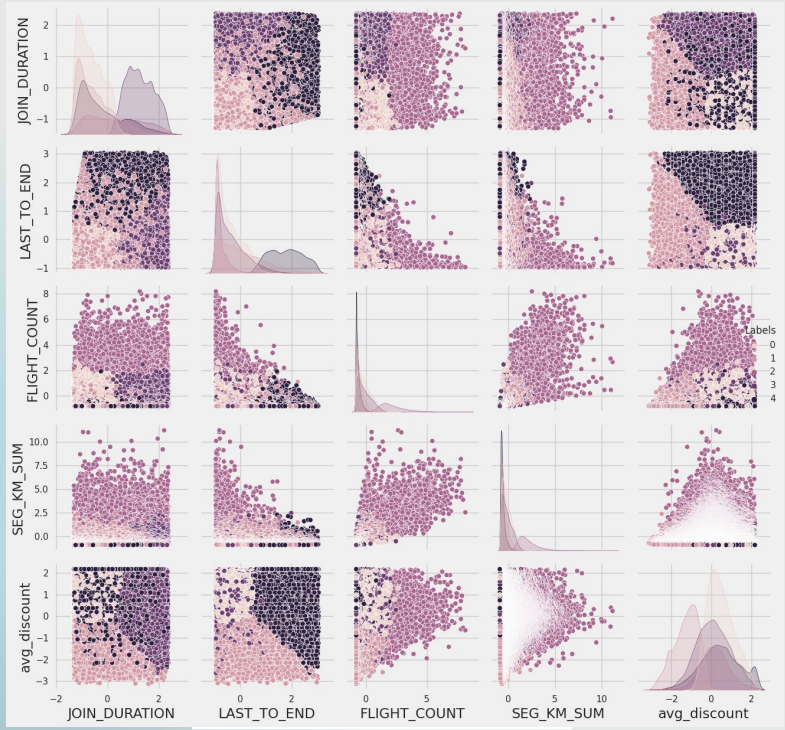
Principal Component Analysis

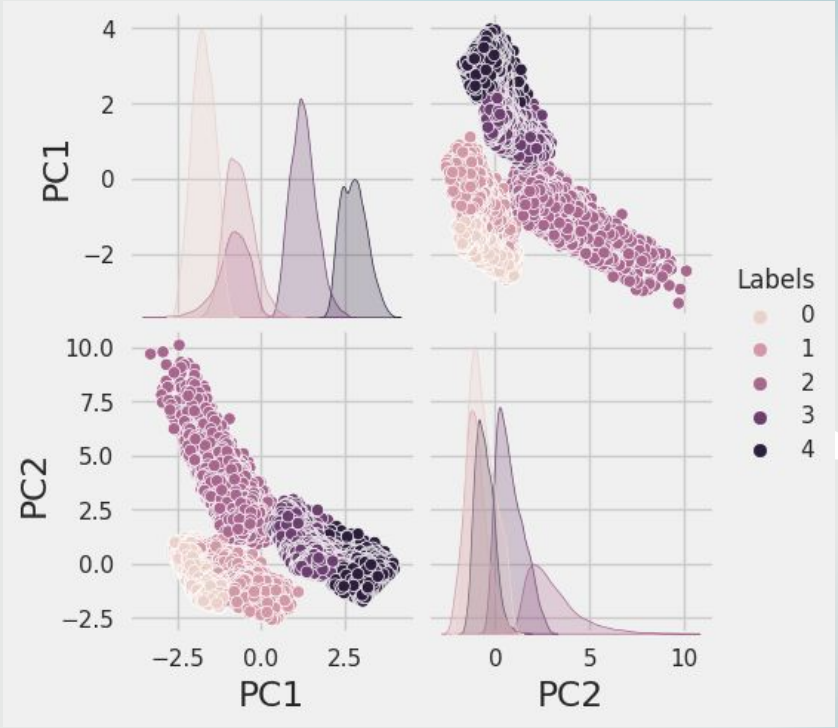
Perform PCA into 2 components x (PC1) and y (PC2).

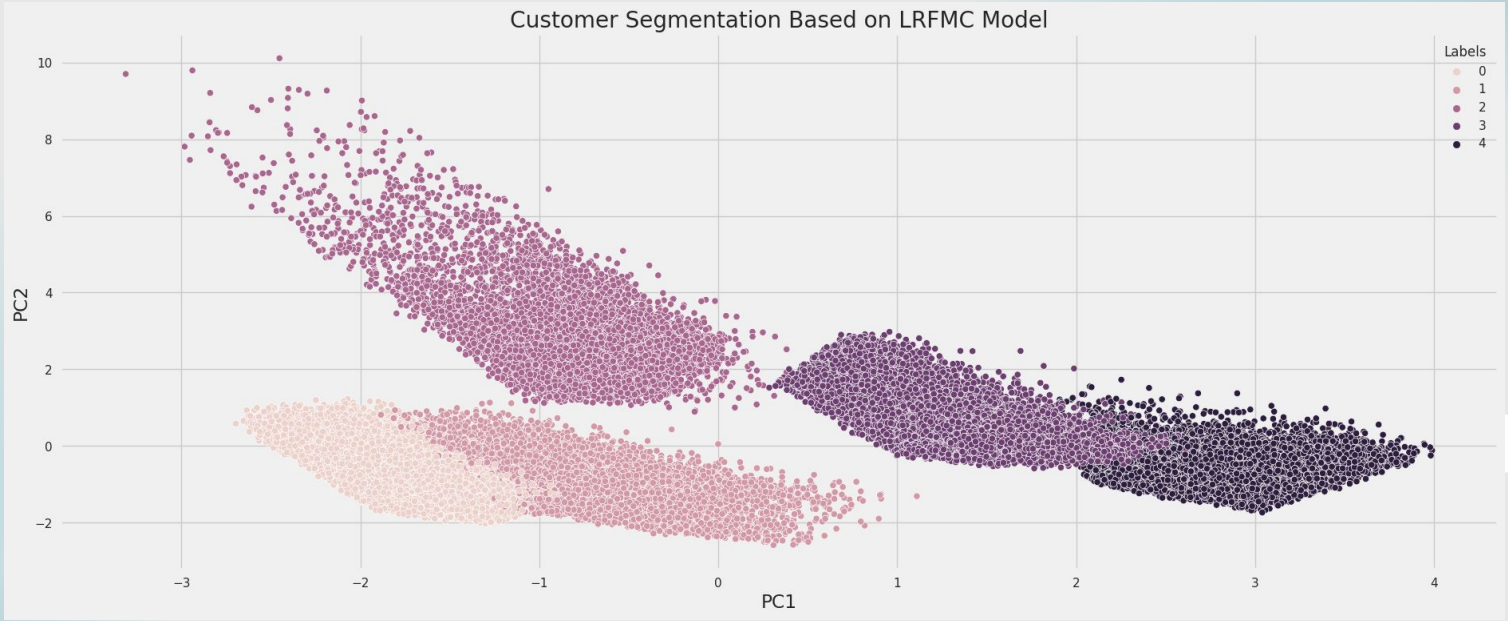
```
[ ]  # Performing PCA
     pca = PCA(n_components=2)

     pca.fit(df_std)
     pcas = pca.transform(df_std)

     # Changing pca result to dataframe
     df_pca = pd.DataFrame(data = pcas, columns = ['PC1', 'PC2'])
     df_pca.describe()
```

	PC1	PC2
count	56028.000000	5.602800e+04
mean	0.000000	-4.869858e-17
std	1.691503	1.447849e+00
min	-3.310981	-2.587070e+00
25%	-1.455932	-1.026449e+00
50%	-0.554077	-3.427475e-01
75%	1.357405	5.875501e-01
max	3.989519	1.011250e+01

Distribution (Scatterplot) from before and after PCA.



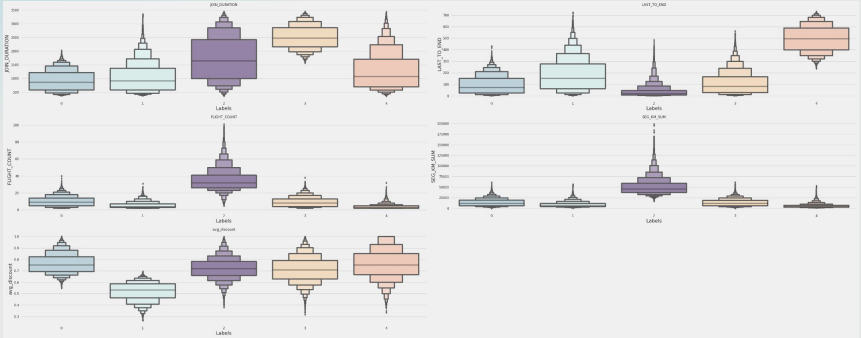



Customer Segmentation Based on LRFMC Model
Labels
0
1
2
3
4
PC2
PC1
10
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3
4

Model Analysis

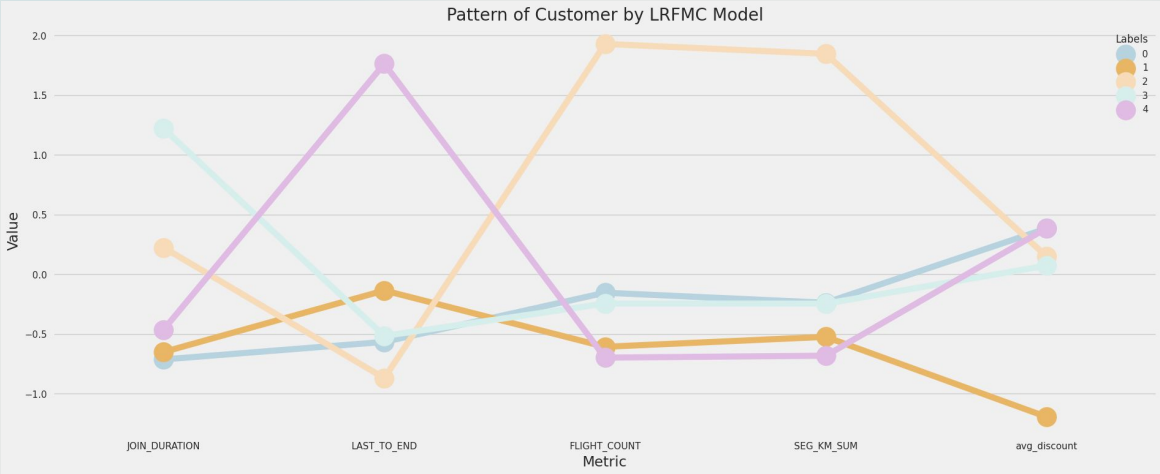
Divided into 5 Clusters

FFP Membership



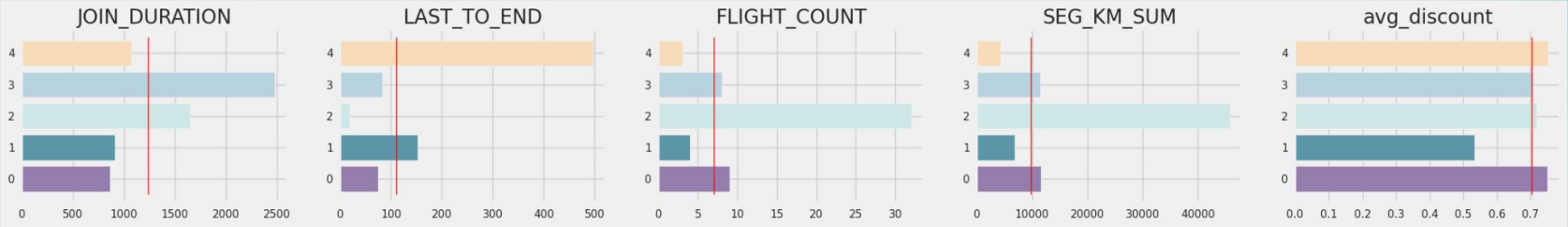
	JOIN_DURATION			LAST_TO_END			FLIGHT_COUNT			SEG_KM_SUM			avg_discount		
	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std
Labels															
0	919.427203	860.0	388.335289	96.996988	74.0	82.459680	9.820252	9.0	6.156315	13537.043793	11527.0	9068.812005	0.762782	0.750688	0.089997
1	1036.879750	912.0	551.950655	182.552916	152.0	144.205770	5.687766	4.0	3.931967	8874.401560	6855.0	6749.657777	0.520865	0.532233	0.084088
2	1731.156280	1642.0	829.626805	38.894796	19.0	53.936899	34.868294	32.0	13.279921	50891.339383	45642.5	20479.507286	0.720805	0.718324	0.097090
3	2508.803666	2474.0	441.458462	111.353992	83.0	100.804205	9.448043	8.0	6.129428	13380.363234	11405.5	8925.934436	0.710389	0.707650	0.117186
4	1268.656270	1068.0	700.447157	495.077436	496.0	116.567751	3.857281	3.0	2.632600	5773.601205	4268.0	4840.916665	0.758284	0.750980	0.135901

	JOIN_DURATION			LAST_TO_END			FLIGHT_COUNT			SEG_KM_SUM			avg_discount		
	max	min	mode	max	min	mode	max	min	mode	max	min	mode	max	min	mode
Labels															
0	2023	365	454	431	1	4	40	2	2	61184	368	1298	1.000000	0.548252	0.920000
1	3347	365	454	723	1	4	31	2	2	56889	716	3934	0.693133	0.264214	0.400000
2	3437	365	746	484	1	1	101	5	28	198627	16834	73392	0.998753	0.381285	0.663333
3	3437	1569	3125	561	1	1	38	2	2	61160	368	3934	1.000000	0.317766	0.920000
4	3429	365	699	729	238	402	32	2	2	52850	368	3934	1.000000	0.335723	1.000000



Insights for each feature:

1. L, JOIN_DURATION, the higher the more loyal.
2. R, LAST_TO_END, the higher the date the ticket booked is after the member's last flight.
3. F, FLIGHT_COUNT, the higher the member flies more often.
4. M, SEG_KM_SUM, the higher the flight distance the member has traveled. It should be positively correlated with FLIGHT_COUNT.
5. C, avg_discount, the higher the discount the member gets.



Label 0 = has a high C pattern and F and M above the median. L and R are low below the median.
Label 1 = has an R pattern above the median while LFMC is below the median.
Label 2 = has a very high F and M pattern and L and C above the median. R is low below the median.
Label 3 = has a high L pattern and FMC above the median. R is below the median.
Label 4 = has a very high R and C pattern but LFM is below the median.

New Potential Customers **0**

The majority of this group of members are the newest members among the other groups, get discounts that are on average high and fly quite often, and have fairly close flight schedules (2-3 months in the future).

This group of members has flight plans that are not very close (5 - 6 months in the future), are quite new members after New Potential Customers but do not have a high flight history and the average discount they get is also small.

1 **New Customers**

Impactful Customers **2**

This group of members flies very often and can be said to be the highest among other groups, this group gets quite high discounts and has been joining the FFP membership for quite a long time. The travel plans that this group is planning are also very close (2 weeks - 1 month in the future).

This group of members has the highest loyalty, with a history of flights and discounts that are quite high but not as high as **Impactful Customers**, planning their next flight quite close (2 - 4 months in the future).

3 **Most Loyal Customers**

Passive Customers **4**

They are loyal members for quite a long time but have very little flight history and the next flight plans are very long (1 - 2 years in the future) even though this group gets very high discounts.

From the dataset it can be seen that there are many outliers which tend to be less logical, therefore it is a good idea to update the data.

The Action Plan carried out is implemented via the FFP application and email for those who do not have the FFP application. Customers who do not have the FFP application will be directed via email so that customers immediately re-register in the FFP application to update their data and customers who have registered and updated their data in the FFP application will receive a "Welcome Gift" in the form of a discount voucher. If the customer already has the FFP application, he will be asked to update his data and will be given a discount voucher "Any New Destinations?".

Treatment for email customers and users of the FFP application will also be differentiated to increase customer interest in the FFP application.

Product *overview*

Email

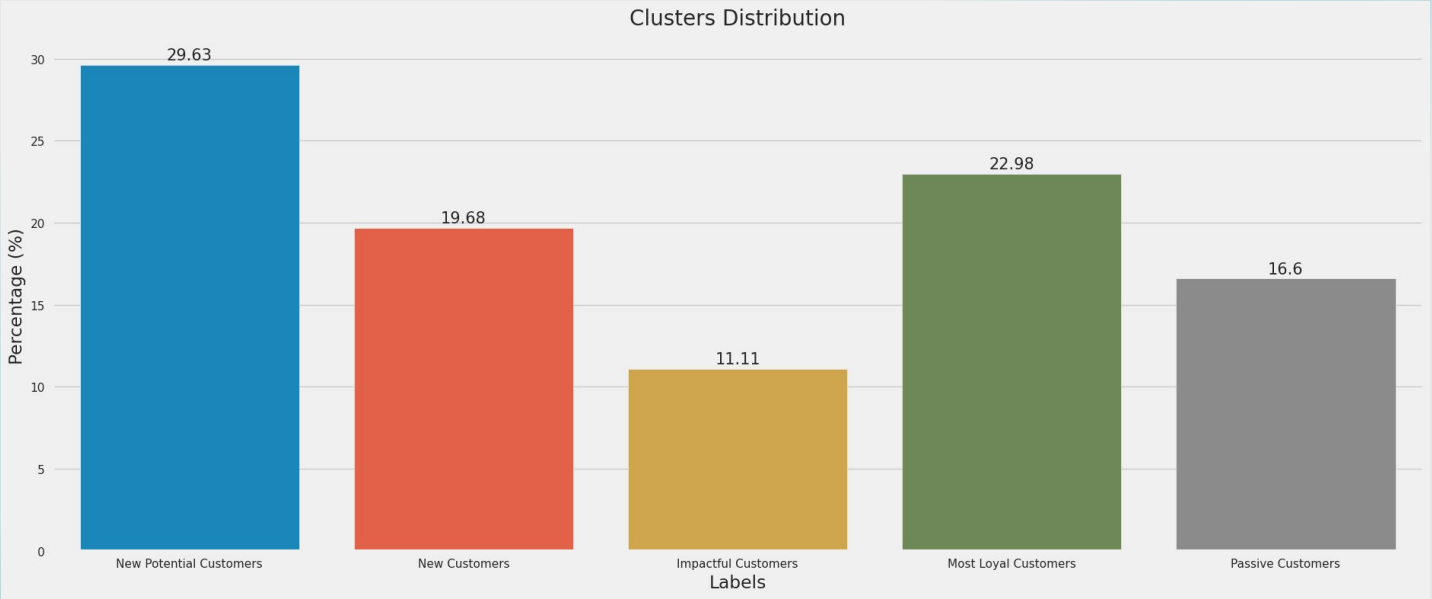
Promotions Purpose Only.

FFP App

Discounts and Promotions.

If you sort them from the most, you get the order,

New Potential Customers (29.63%) - New Customers (19.68%) - Most Loyal Customers (22.98%) - Passive Customers (16.6%) - Impactful Customers (11.11 %)



Airline

Action Plan:

- Providing recommendations for interesting destinations and the cheapest prices from these destinations, which along with the recommendations includes claiming additional discount vouchers which lead to the FFP application.
- Gives points if you fly 6 times in 1 year (Silver Member) and rewards in the form of products/souvenirs/discount vouchers that can be exchanged.
- Carrying out the "Fly with Us" campaign with the aim of keeping customers active so they can become Impactful Customers / Most Loyal Customers.

NOTE:

This is the cluster with the largest number.

New Potential Customers

- 0

This group is a new and quite active member and is very connected with the discounts they get.

Airline

Action Plan:

- Send an email redeem code discount which has a validity period via email and a recommendation to continue it to the FFP application or can be exchanged directly when buying tickets at the airport.
- Gives points if you fly 6 times in 1 year (Silver Member) and rewards in the form of products/souvenirs/discount vouchers that can be exchanged.
- Carry out an alternative campaign "Fly with Us" to make customers more active New Potential Customers.

New Customers

- 1

This group does not get many discounts which is thought to be the cause of the small running history this group has.

Airline

Action Plan:

- Sending "Thank you for Trusting Us." and provide discount vouchers for each customer as well as souvenirs that can be exchanged directly.
- Gives points for every 3 flights in 1 year (Gold Member) and rewards in the form of products/souvenirs/discount vouchers that can be exchanged.
- Carrying out the "Share to Explore the World" campaign where these customers can be considered affiliates of FFP and get a referral code which can be exchanged for points. Apart from that, this campaign also focuses on building good relationships with customers.

NOTE:

The cluster with the fewest numbers but highest consumption.

Impactful Customers

- 2

This group is the most active group carrying out flights so it is called Impactful.

Airline

Action Plan:

- Sending greetings in the form of a short letter "After together, we've shared trips together, I'm so thankful for always choosing us. I still remember when we went to together, it was so amazing. Why don't we plan our next trip?" By writing this down it is hoped that customers can reminisce and increase the possibility of flying.
- Gives points for every 3 flights in 1 year (Gold Member) and rewards in the form of products/souvenirs/discount vouchers that can be exchanged.
- Carrying out the "Share to Explore the World" campaign where these customers can be considered as affiliates of FFP and get a referral code which can be exchanged for points. Apart from that, this campaign also focuses on building good relationships with customers.

Most Loyal Customers

- 3

The cluster with the fewest but highest consumption.

Airline

Action Plan:

- This group requires special attention, perhaps by providing regular destination recommendations along with interesting descriptions of the destination.
- Providing attractive personalized discount/promo vouchers.
- Gives points for every 9 flights in 1 year (Bronze Member) and rewards in the form of products/souvenirs/discount vouchers that can be exchanged.
- Carrying out the "Escape Trip" campaign, this campaign was carried out to increase the interest of this group and survey the causes of the low interest of this group.

NOTE:

It needs to be monitored so that this cluster does not increase in number.

Most Loyal Customers

- 4

This is a group that tends to be inactive even though they have been joining for a long time and there are high discounts, their flight patterns are very low.

Thanks

Do you have any questions?

gbrllmrn2005@gmail.com