

Flight Dataset

Homework Supervised
Learning



Dataset

Airline Customer

Deskripsi

Dataset ini berisi data pelanggan sebuah perusahaan penerbangan dan beberapa fitur yang dapat memberikan gambaran mendalam tentang informasi penerbangan dari pelanggan tersebut.

Data

Setiap baris dalam dataset ini mewakili seorang pelanggan, dan setiap kolom berisi atribut-atribut yang merinci informasi tentang pelanggan tersebut.

Task

K-Means Clustering

Tujuan

Melihat karakteristik yang khas dari tiap segmen customer untuk selanjutnya memberikan perlakuan berbeda sesuai konteks bisnis



Dataset

Sekilas tentang Dataset

```
[ ] df = pd.read_csv(path)
df.sample(5)
```

MEMBER_NO	FFP_DATE	FIRST_FLIGHT_DATE	GENDER	FFP_TIER	WORK_CITY	WORK_PROVINCE	WORK_COUNTRY	AGE	LOAD_TIME	...	SUM_YR_2	SEG_KM_SUM	LAST_FLIGHT_DATE	LAST_TO_END	A	
13774	22596	6/19/2010		6/20/2010	Male	4	shenyang	liaoning	CN	47.0	3/31/2014	...	1926.0	27424	10/6/2013	178
17835	45338	11/8/2011		6/6/2012	Male	4	zhengzhou	henan	CN	42.0	3/31/2014	...	5569.0	15870	8/12/2013	233
30222	38138	4/28/2005		5/22/2012	Male	4	Nan	NaN	CN	66.0	3/31/2014	...	1824.0	5722	3/28/2014	4
17808	24844	7/5/2012		7/15/2012	Male	4	beijing	beijing	CN	42.0	3/31/2014	...	0.0	19142	7/23/2012	618
27879	60133	12/31/2011		9/13/2012	Male	4	shanghai	shanghai	CN	35.0	3/31/2014	...	0.0	9980	9/19/2013	195

5 rows × 23 columns

Lanjutan

AVG_INTERVAL	MAX_INTERVAL	EXCHANGE_COUNT	avg_discount	Points_Sum	Point_NotFlight
45.666667	230	2	0.624363	12295	2
19.636364	101	0	0.863913	15352	7
96.571429	355	0	1.293721	7355	0
2.666667	8	0	0.717119	12079	0
53.000000	233	0	0.827525	6509	0

1. EXPLORATORY DATA ANALYSIS

Descriptive Statistic

Before

```
[ ] # Informasi umum dataframe
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
```

Untuk tiga kolom, yaitu 'FFP_DATE', 'FIRST_FLIGHT_DATE', dan 'LAST_FLIGHT_DATE', seharusnya memiliki tipe data 'Date', sementara sisanya tetap mengikuti tipe data yang sesuai dengan konteksnya.

```
[ ] # Define the list of date columns
var_date = ['FFP_DATE', 'FIRST_FLIGHT_DATE', 'LOAD_TIME']

# Transform the data types to datetime
df[var_date] = df[var_date].applymap(lambda x: pd.to_datetime(x, format = '%m/%d/%Y'))

# Now the specified columns are of datetime data type
```

Perubahan tipe data, dan penggantian format tanggal menjadi bulan/hari/tahun

After

```
[ ] #dataframe after transformation data type to date
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   datetime64[ns]
 2   FIRST_FLIGHT_DATE 62988 non-null   datetime64[ns]
 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   datetime64[ns]
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: datetime64[ns](3), float64(5), int64(10), object(5)
memory usage: 11.1+ MB
```

Descriptive Statistic (Numeric)

```
# Statistik dari setiap kolom numeric
df[nums].describe().apply(lambda x: x.apply('{0:.5f}'.format))
```

	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	62988.00000	62988.00000	62568.00000	62988.00000	62988.00000	62437.00000	62850.00000	62988.00000	62988.00000	62988.00000	62988.00000	62988.00000	62988.00000	62988.00000	62988.00000
mean	31494.50000	4.10216	42.47635	11.83941	10925.08125	5355.37606	5604.02601	17123.87869	176.12010	67.74979	166.03390	0.31978	0.72156	12545.77710	2.72815
std	18183.21371	0.37386	9.88591	14.04947	16339.48615	8109.45015	8703.36425	20960.84462	183.82222	77.51787	123.39718	1.13600	0.18543	20507.81670	7.36416
min	1.00000	4.00000	6.00000	2.00000	0.00000	0.00000	0.00000	368.00000	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
25%	15747.75000	4.00000	35.00000	3.00000	2518.00000	1003.00000	780.00000	4747.00000	29.00000	23.37037	79.00000	0.00000	0.61200	2775.00000	0.00000
50%	31494.50000	4.00000	41.00000	7.00000	5700.00000	2800.00000	2773.00000	9994.00000	108.00000	44.66667	143.00000	0.00000	0.71186	6328.50000	0.00000
75%	47241.25000	4.00000	48.00000	15.00000	12831.00000	6574.00000	6845.75000	21271.25000	268.00000	82.00000	228.00000	0.00000	0.80948	14302.50000	1.00000
max	62988.00000	6.00000	110.00000	213.00000	505308.00000	239560.00000	234188.00000	580717.00000	731.00000	728.00000	728.00000	46.00000	1.50000	985572.00000	140.00000

Kolom MEMBER_NO, AGE, FLIGHT_COUNT, BP_SUM, SUM_YR_1, SUM_YR_2, SEG_KM_SUM, LAST_TO_END, AVG_INTERVAL, dan MAX_INTERVAL memiliki standar deviasi yang besar, yang mengindikasikan variasi yang besar menyebar luas dari nilai rata-ratanya.

Selain itu, ada perbedaan yang cukup signifikan antara nilai rata-rata (mean) dan median pada beberapa kolom, yaitu FLIGHT_COUNT, BP_SUM, SUM_YR_1, SUM_YR_2, SEG_KM_SUM, LAST_TO_END, AVG_INTERVAL, MAX_INTERVAL, dan Points_Sum. Perbedaan ini menunjukkan adanya skewness (kemiringan) dalam distribusi data.

Descriptive Statistic (Categorical)

```
[ ] # Statistik dari setiap kolom categorical  
df[cats].describe()
```

	GENDER	WORK_CITY	WORK_PROVINCE	WORK_COUNTRY	LAST_FLIGHT_DATE
count	62985	60719	59740	62962	62988
unique	2	3234	1165	118	731
top	Male	guangzhou	guangdong	CN	3/31/2014
freq	48134	9386	17509	57748	959

Variabel WORK_CITY dan WORK_PROVINCE memiliki unique value yang sangat besar, sedangkan variabel WORK_COUNTRY value “CN” memiliki nilai yang mendominasi yaitu 57748 yang terindikasi imbalance. Sehingga ke-tiga variable diatas kemungkinan akan dilakukan drop.

Descriptive Statistic (Date)

	FFP_DATE	FIRST_FLIGHT_DATE	LOAD_TIME
count	62988	62988	62988
unique	3068	3406	1
top	2011-01-13 00:00:00	2013-02-16 00:00:00	2014-03-31 00:00:00
freq	184	96	62988
first	2004-11-01 00:00:00	1905-12-31 00:00:00	2014-03-31 00:00:00
last	2013-03-31 00:00:00	2015-05-30 00:00:00	2014-03-31 00:00:00

Variabel LOAD_TIME memiliki data yang homogen, sehingga variabel ini hanya akan dipakai untuk membuat feature extraction, tidak untuk dimasukkan dalam analisis

Missing Value

```
[ ] # Missing values
missing_values = df_clean.isna().sum()
total_rows = len(df_clean)

missing_percentage = (missing_values / total_rows) * 100

missing_info = pd.DataFrame({
    'Missing Values': missing_values,
    'Percentage': missing_percentage
})

print(missing_info)
```

Pengecekan terhadap missing value pada dataset

	Missing Values	Percentage
MEMBER_NO	0	0.000000
FFP_DATE	0	0.000000
FIRST_FLIGHT_DATE	0	0.000000
GENDER	3	0.004763
FFP_TIER	0	0.000000
WORK_CITY	2269	3.602273
WORK_PROVINCE	3248	5.156538
WORK_COUNTRY	26	0.041278
AGE	420	0.666794
LOAD_TIME	0	0.000000
FLIGHT_COUNT	0	0.000000
BP_SUM	0	0.000000
SUM_YR_1	551	0.874770
SUM_YR_2	138	0.219089
SEG_KM_SUM	0	0.000000
LAST_FLIGHT_DATE	0	0.000000
LAST_TO_END	0	0.000000
AVG_INTERVAL	0	0.000000
MAX_INTERVAL	0	0.000000
EXCHANGE_COUNT	0	0.000000
avg_discount	0	0.000000
Points_Sum	0	0.000000
Point_NotFlight	0	0.000000

Data Imputation

```
[ ] # Menghitung rata-rata kolom 'AGE'
median_AGE = df_clean['AGE'].median()

# Mengisi missing value dalam kolom 'AGE' dengan rata-rata
df_clean['AGE'].fillna(median_AGE, inplace=True)

[ ] # Menghitung modus dari kolom 'gender'
mode_gender = df_clean['GENDER'].mode()[0]

# Mengisi missing value dalam kolom 'gender' dengan modus
df_clean['GENDER'].fillna(mode_gender, inplace=True)

[ ] # Menghitung rata-rata kolom 'SUM_YR'
median_Y1 = df_clean['SUM_YR_1'].median()
median_Y2 = df_clean['SUM_YR_2'].median()

# Mengisi missing value dalam kolom 'sum_yr_1' dan 'sum_yr_2' dengan median
df_clean['SUM_YR_1'].fillna(median_Y1, inplace=True)
df_clean['SUM_YR_2'].fillna(median_Y2, inplace=True)
```

Dilakukan imputasi data pada setiap missing value pada dataset

```
[ ] #cek setelah imputasi
df_clean.isna().sum()
```

MEMBER_NO	0
FFP_DATE	0
FIRST_FLIGHT_DATE	0
GENDER	0
FFP_TIER	0
WORK_CITY	2269
WORK_PROVINCE	3248
WORK_COUNTRY	26
AGE	0
LOAD_TIME	0
FLIGHT_COUNT	0
BP_SUM	0
SUM_YR_1	0
SUM_YR_2	0
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	

Duplicated Data

```
[ ] # Menghitung jumlah data yang duplikat
duplicate_count = df_clean.duplicated().sum()
print("Jumlah data yang duplikat:", duplicate_count)
```

```
Jumlah data yang duplikat: 0
```

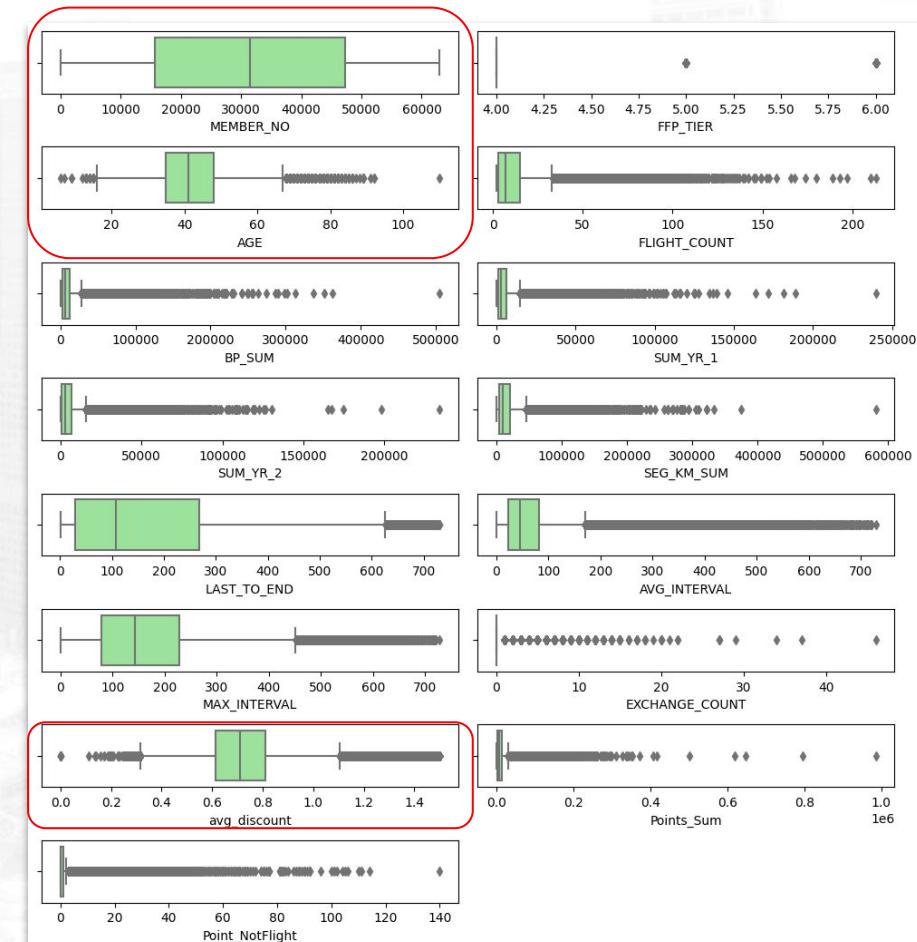
Tidak ada data yang duplikat pada dataset

Univariate Statistic

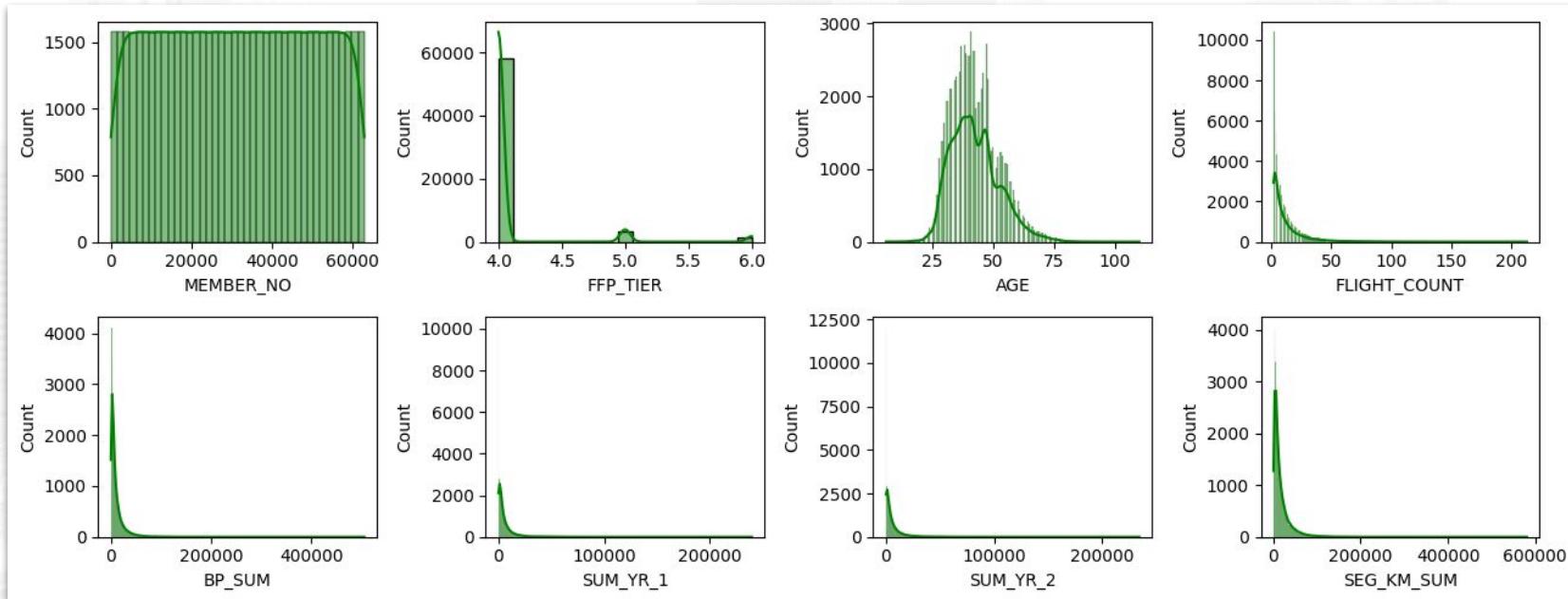
```
#Boxplot fitur numerical
plt.figure(figsize=(10,15))
for i in range (0, len(nums)):
    plt.subplot(12, 2, i+1)
    sns.boxplot (x=df_clean[nums [i]], color= 'lightgreen')
    plt.tight_layout()

plt.show()
```

MEMBER_NO, **AGE** dan **avg_discount**, memiliki distribusi yang cenderung normal atau skew sedikit, sisanya sangat skew dan mengandung banyak outliers extreme sehingga beberapa variabel distribusi 50% datanya sulit terlihat.

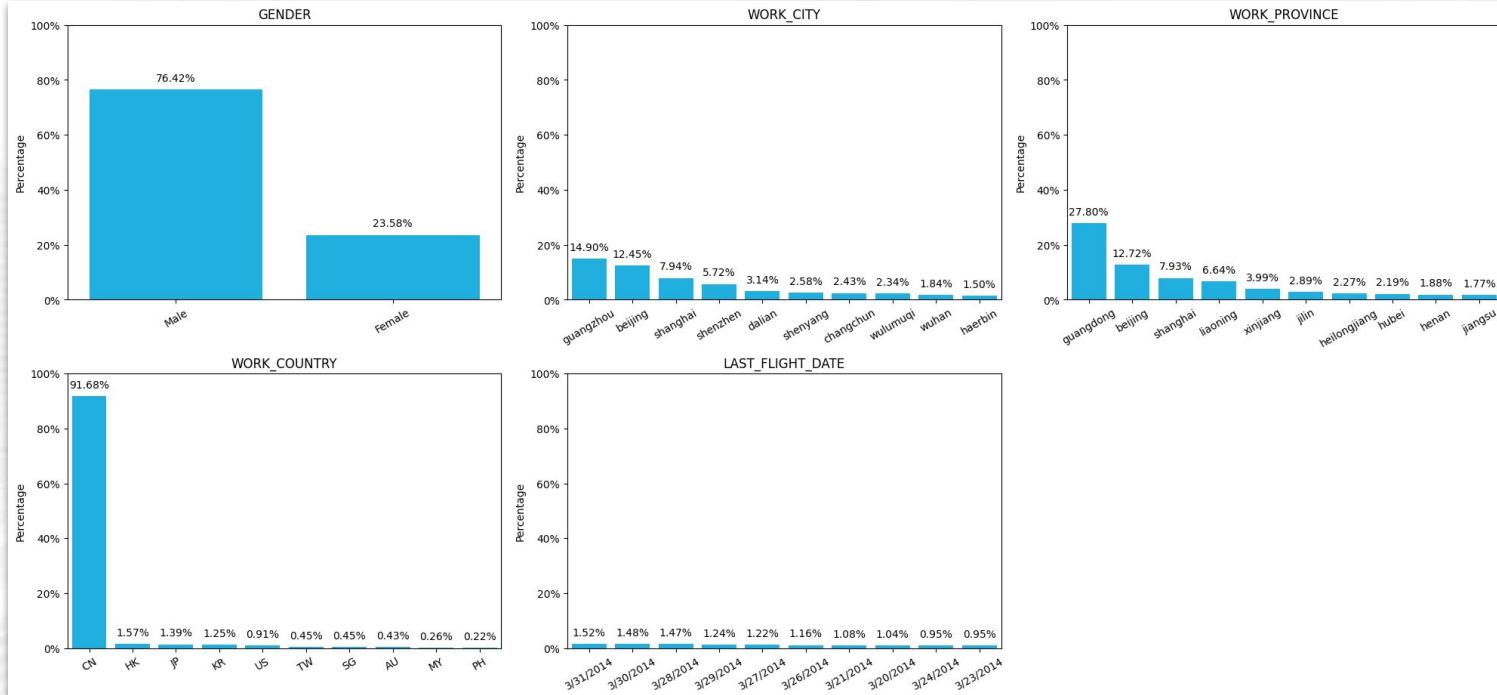


Univariate Statistic



MEMBER_NO berbentuk seperti itu karena semua data memiliki nilai unique, **AGE** cenderung berdistribusi cukup normal, sisanya skewed. Untuk **MEMBER_NO** akan dipertimbangkan untuk **di drop** karena berisi informasi yang kurang dibutuhkan dalam pembuatan model.

Univariate Statistic

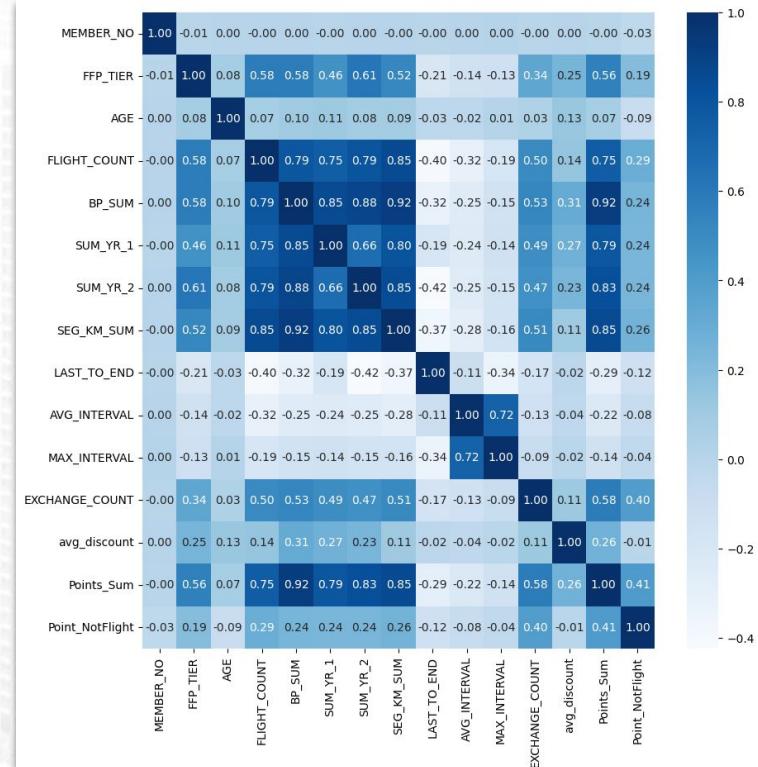


Untuk variabel **WORK_COUNTRY** sangat **imbalanced class** (majoritas penerbangan ada di CHINA), sisanya distribusinya cukup merata. Untuk **LAST_FLIGHT_DATE** nantinya akan dipakai untuk feature extraction bersama fitur 'Date' lainnya. Semua fitur 'WORK' kemungkinan akan di drop karena terlalu **imbalanced** dan juga banyak **unique** value.

Multivariate Analysis (heatmap-numerical column)

```
plt.figure(figsize=(10,10))
sns.heatmap(df_clean.corr(), cmap='Blues', annot=True, fmt='.2f')
#perlu make sure multi kolinearitas (hub antar var x)
```

Variabel **BP_SUM** memiliki **korelasi yang sangat tinggi** dengan 3 variabel lainnya yaitu **SUM_YR_1**, **SUM_YR_2** dan **SEG_KM_SUM** sehingga dipertimbangkan untuk di drop. Namun, sisanya akan coba kami masukan dalam **pilihan** untuk analisis clustering.



2. PREPROCESSING

DROP IRRELEVANT FEATURES

```
df_clean.drop(columns=['BP_SUM', 'MEMBER_NO','WORK_CITY','WORK_PROVINCE', 'WORK_COUNTRY'], inplace=True)
df_clean.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62988 entries, 0 to 62987
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   FFP_DATE        62988 non-null   datetime64[ns]
 1   FIRST_FLIGHT_DATE 62988 non-null   datetime64[ns]
 2   GENDER          62988 non-null   object  
 3   FFP_TIER        62988 non-null   int64  
 4   AGE             62988 non-null   float64 
 5   LOAD_TIME       62988 non-null   datetime64[ns]
 6   FLIGHT_COUNT    62988 non-null   int64  
 7   SUM_YR_1        62988 non-null   float64 
 8   SUM_YR_2        62988 non-null   float64 
 9   SEG_KM_SUM      62988 non-null   int64  
 10  LAST_FLIGHT_DATE 62988 non-null   object  
 11  LAST_TO_END     62988 non-null   int64  
 12  AVG_INTERVAL    62988 non-null   float64 
 13  MAX_INTERVAL    62988 non-null   int64  
 14  EXCHANGE_COUNT  62988 non-null   int64  
 15  avg_discount    62988 non-null   float64 
 16  Points_Sum      62988 non-null   int64  
 17  Point_NotFlight 62988 non-null   int64  
dtypes: datetime64[ns](3), float64(5), int64(8), object(2)
memory usage: 8.7+ MB
```

- **BP_SUM** dibuang karena memiliki korelasi yang sangat tinggi dengan 3-4 variabel lain.
- **MEMBER_NO** dibuang karena berisi data yang tidak informatif dikarenakan banyaknya unique value.
- Data '**WORK**' dibuang karena imbalance class dan juga terlalu banyak unique value.

2. PREPROCESSING

HANDLE OUTLIERS

```
| # Handle Outliers
| print(f'Number of rows before removing outlier: {len(df)}')

| filtered = np.array([True] * len(df))
| for f in ['LAST_TO_END', 'SEG_KM_SUM', 'FLIGHT_COUNT', 'SUM_YR_1', 'FFP_TIER', 'Point_NotFlight', 'AVG_INTERVAL']:
|     Q1 = df_clean[f].quantile(0.25)
|     Q3 = df_clean[f].quantile(0.75)
|     iqr = Q3 - Q1
|     b_thresh = Q1 - (1.5 * iqr)
|     u_thresh = Q3 + (1.5 * iqr)

|     filtered = ((df_clean[f] >= b_thresh) & (df_clean[f] <= u_thresh))
| df = df_clean[filtered]

| print(f'Number of rows after removing outlier: {len(df)}')

| Number of rows before removing outlier: 62988
| Number of rows after removing outlier: 58148
```

Handling outlier menggunakan metode **IQR**. Setelah menghapus beberapa outliers, terdapat sekitar **4 ribu data yang hilang (sekitar 8%)**. Pemilihan variabel berdasarkan yang memiliki **skew extreme dan berpeluang paling tinggi** untuk dipakai analisis clustering.

2. PREPROCESSING

ONE HOT ENCODING: GENDER

```
[ ] #OHE GENDER
encodee = ['GENDER']

for e in encodee:
    ohe= pd.get_dummies(df[e], prefix=e)
    df = df.join(ohe)

df = df.drop(encodee, axis=1)
```

	GENDER_Female	GENDER_Male
0	1	
0	1	
0	1	
0	1	

```
[ ] #delete hasil OHE
df.drop(columns=['GENDER_Female'], inplace=True)
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 60907 entries, 0 to 62987
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   FFP_DATE         60907 non-null   datetime64[ns]
 1   FIRST_FLIGHT_DATE 60907 non-null   datetime64[ns]
 2   FFP_TIER          60907 non-null   int64  
 3   AGE               60907 non-null   float64 
 4   LOAD_TIME         60907 non-null   datetime64[ns]
 5   FLIGHT_COUNT      60907 non-null   int64  
 6   SUM_YR_1           60907 non-null   float64 
 7   SUM_YR_2           60907 non-null   float64 
 8   SEG_KM_SUM         60907 non-null   int64  
 9   LAST_FLIGHT_DATE   60907 non-null   object  
 10  LAST_TO_END        60907 non-null   int64  
 11  AVG_INTERVAL       60907 non-null   float64 
 12  MAX_INTERVAL       60907 non-null   int64  
 13  EXCHANGE_COUNT     60907 non-null   int64  
 14  avg_discount        60907 non-null   float64 
 15  Points_Sum          60907 non-null   int64  
 16  Point_Notflight     60907 non-null   int64  
 17  GENDER_Male          60907 non-null   uint8  
dtypes: datetime64[ns](3), float64(5), int64(8), object(1), uint8(1)
memory usage: 10.4+ MB
```

Gender **female** di **drop** karena telah **diwakili** oleh **male**, male dipakai karena merupakan data yang lebih **dominan** secara jumlah ketimbang female

2. PREPROCESSING

FEATURE ENGINEERING

TOTAL_FLIGHT_YEARS

```
[ ] # Mengekstraksi tahun pada FIRST_FLIGHT_DATE
df['FIRST_FLIGHT_DATE'] = pd.to_datetime(df['FIRST_FLIGHT_DATE'], errors='coerce')
df['FIRST_FLIGHT:YEAR'] = df['FIRST_FLIGHT_DATE'].dt.year

[ ] # Mengekstraksi tahun pada LAST_FLIGHT_DATE
df['LAST_FLIGHT_DATE'] = pd.to_datetime(df['LAST_FLIGHT_DATE'], errors='coerce')
df['LAST_FLIGHT:YEAR'] = df['LAST_FLIGHT_DATE'].dt.year

[ ] # menambahkan feature TOTAL FLIGHT YEARS untuk melihat hubungan dengan FLIGHT COUNT, EXCHANGE COUNT, AVG DISCOUNT, POINTS_SUM, dan POINT_NOTFLIGHT
df['TOTAL_FLIGHT_YEARS'] = df['LAST_FLIGHT:YEAR'] - df['FIRST_FLIGHT:YEAR']

[ ] df.info()
```

FFP_JOIN_YEARS & RECENT_FLIGHT_YEARS

```
[ ] # Mengekstraksi tahun pada FFP_DATE
df['FFP_DATE'] = pd.to_datetime(df['FFP_DATE'], errors='coerce')
df['FFP:YEAR'] = df['FFP_DATE'].dt.year

[ ] # Mengekstraksi tahun pada LOAD_TIME
df['LOAD_TIME'] = pd.to_datetime(df['LOAD_TIME'], errors='coerce')
df['LOAD_TIME:YEAR'] = df['LOAD_TIME'].dt.year

[ ] # menambahkan feature durasi lamanya join FFP sampai saat data diambil
df['FFP_JOIN_YEARS'] = df['LOAD_TIME:YEAR'] - df['FFP:YEAR']

[ ] # menambahkan feature durasi lamanya jarak terakhir kali terbang sampai saat data diambil
df['RECENT_FLIGHT_YEARS'] = df['LOAD_TIME:YEAR'] - df['LAST_FLIGHT:YEAR']

[ ] df.info()
```

- **TOTAL_FLIGHT_YEARS** merupakan jarak waktu tahunan antara penerbangan pertama ke penerbangan terakhir.
- **FFP_JOIN_YEARS** adalah durasi/lamanya pelanggan (dalam tahun) bergabung menjadi member sampai data terakhir diambil.
- **RECENT_FLIGHT_YEARS** adalah jarak tahunan antara penerbangan terakhir dengan waktu terakhir data diambil.

2. PREPROCESSING

DROP FEATURES AFTER EXTRACTION

```
#delete fitur yang tidak diperlukan setelah extraction
df.drop(columns=['FFP_DATE', 'FIRST_FLIGHT_DATE','LOAD_TIME','LAST_FLIGHT_DATE', 'FIRST_FLIGHT:YEAR', 'LAST_FLIGHT:YEAR','FFP:YEAR', 'LOAD_TIME:YEAR'], inplace=True)
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 60907 entries, 0 to 62987
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   FFP_TIER        60907 non-null   int64  
 1   AGE              60907 non-null   float64 
 2   FLIGHT_COUNT    60907 non-null   int64  
 3   SUM_YR_1         60907 non-null   float64 
 4   SUM_YR_2         60907 non-null   float64 
 5   SEG_KM_SUM       60907 non-null   int64  
 6   LAST_TO_END      60907 non-null   int64  
 7   AVG_INTERVAL     60907 non-null   float64 
 8   MAX_INTERVAL     60907 non-null   int64  
 9   EXCHANGE_COUNT   60907 non-null   int64  
 10  avg_discount     60907 non-null   float64 
 11  Points_Sum       60907 non-null   int64  
 12  Point_NotFlight 60907 non-null   int64  
 13  GENDER_Male      60907 non-null   uint8  
 14  TOTAL_FLIGHT_YEARS 60505 non-null   float64 
 15  FFP_JOIN_YEARS   60907 non-null   int64  
 16  RECENT_FLIGHT_YEARS 60505 non-null   float64 
dtypes: float64(7), int64(9), uint8(1)
memory usage: 10.0 MB
```

17 Fitur di atas bisa dipakai untuk clustering

Fitur-fitur yang didrop merupakan fitur awal yang telah diekstraksi menjadi fitur baru yang dianggap lebih informatif dalam model, sehingga lebih baik didrop untuk mengurangi variance.

2. PREPROCESSING

DROP MISSING VALUE (AFTER EXTRACTION)

```
df.isna().sum()  
  
FFP_TIER          0  
AGE              0  
FLIGHT_COUNT     0  
SUM_YR_1          0  
SUM_YR_2          0  
SEG_KM_SUM        0  
LAST_TO_END       0  
AVG_INTERVAL      0  
MAX_INTERVAL      0  
EXCHANGE_COUNT    0  
avg_discount      0  
Points_Sum         0  
Point_NotFlight   0  
GENDER_Male        0  
TOTAL_FLIGHT_YEARS 402  
FFP_JOIN_YEARS    0  
RECENT_FLIGHT_YEARS 402  
dtype: int64
```



```
[ ] #hapus data yang missing  
df= df.dropna(subset=['TOTAL_FLIGHT_YEARS', 'RECENT_FLIGHT_YEARS'])  
  
df.isna().sum()  
  
FFP_TIER          0  
AGE              0  
FLIGHT_COUNT     0  
SUM_YR_1          0  
SUM_YR_2          0  
SEG_KM_SUM        0  
LAST_TO_END       0  
AVG_INTERVAL      0  
MAX_INTERVAL      0  
EXCHANGE_COUNT    0  
avg_discount      0  
Points_Sum         0  
Point_NotFlight   0  
GENDER_Male        0  
TOTAL_FLIGHT_YEARS 0  
FFP_JOIN_YEARS    0  
RECENT_FLIGHT_YEARS 0  
dtype: int64
```

Setelah proses ekstraksi, terdapat **missing value** dalam fitur **TOTAL_FLIGHT_YEARS** dan **RECENT_FLIGHT_YEARS**. Data yang missing langsung **di drop**.

2. PREPROCESSING

PERCOBAAN 6 VARIABEL PERTAMA

FEATURE SELECTION #1

Menggunakan konsep RFM, fitur yang termasuk di dalamnya yaitu :

Recency : FFP_JOIN_YEARS dan LAST_TO_END

Frequency : SEG_KM_SUM, AVG_INTERVAL dan FLIGHT_COUNT

Monetary : SUM_YR_1

Metode Feature Selection yang kami gunakan adalah metode RFM (Recency, Frequency, Monetary).

Untuk percobaan 1, features yang kami pilih adalah:

Recency : FFP_JOIN_YEARS, LAST_TO_END

Hal ini karena FFP_JOIN_YEARS menggambarkan berapa lama durasi customer join FFP sampai data diambil dan LAST_TO_END dipilih untuk mengetahui seberapa lama customer tidak terbang dengan membandingkan penerbangan seorang customer terakhir kali dengan book penerbangan terakhir oleh kebanyakan customer lain.

Frequency : SEG_KM_SUM, AVG_INTERVAL, FLIGHT_COUNT

SEG_KM_SUM menggambarkan total jarak penerbangan yang telah ditempuh, AVG_INTERVAL berapa jarak waktu yang ditempuh setiap penerbangan dan FLIGHT_COUNT menggambarkan berapa kali penerbangan dilakukan.

Monetary : SUM_YR_1

SUM_YR_1 dipilih karena menggambarkan jumlah pendapatan total yang diterima dari customer atau berapa besar total nominal uang yang telah dibelanjakan oleh tiap customer

2. PREPROCESSING

STANDARDIZATION

PERCOBAAN 6 VARIABEL PERTAMA

STANDARISASI

```
[ ] from sklearn.preprocessing import RobustScaler  
  
X = df_first[columns6].values  
scaler = RobustScaler()  
X_robust = scaler.fit_transform(X)  
df_6 = pd.DataFrame(data=X_robust, columns=columns6)  
  
[ ] df_6.describe()
```

	FFP_JOIN_YEARS	LAST_TO_END	SEG_KM_SUM	AVG_INTERVAL	FLIGHT_COUNT	SUM_YR_1
count	60505.000000	60505.000000	60505.000000	60505.000000	60505.000000	60505.000000
mean	0.078646	0.274952	0.423471	0.329475	0.422407	0.459884
std	0.586510	0.746420	1.253099	1.136694	1.184345	1.449882
min	-0.750000	-0.445783	-0.585140	-0.792295	-0.416667	-0.503986
25%	-0.500000	-0.329317	-0.321599	-0.373534	-0.333333	-0.320638
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.500000	0.670683	0.678401	0.626466	0.666667	0.679362
max	1.500000	2.485944	33.692162	7.517588	17.166667	41.933570

3. MODELLING DAN EVALUASI

CLUSTERING

CLUSTERING

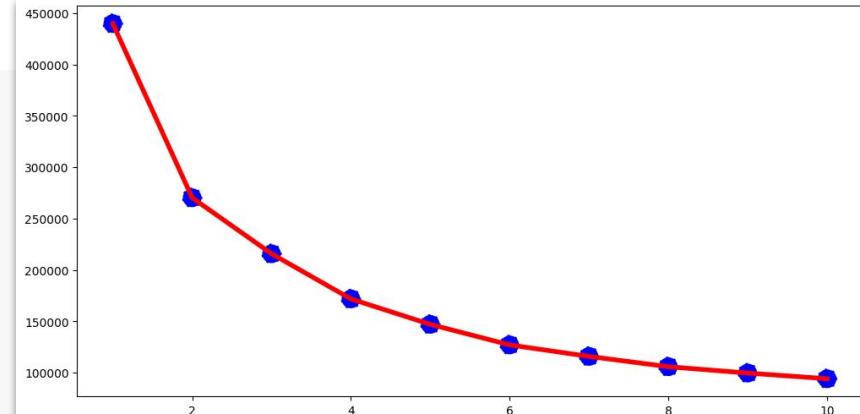
```
[ ] # K-means
from sklearn.cluster import KMeans
inertia = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(df_6)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(12, 6))

sns.lineplot(x=range(1, 11), y=inertia, color='red', linewidth = 4)
sns.scatterplot(x=range(1, 11), y=inertia, s=300, color='blue', linestyle='--')
```

PERCOBAAN 6 VARIABEL PERTAMA



Berdasarkan Elbow Method, **2 cluster** merupakan jumlah yang paling **direkomendasikan**.

3. MODELLING DAN EVALUASI

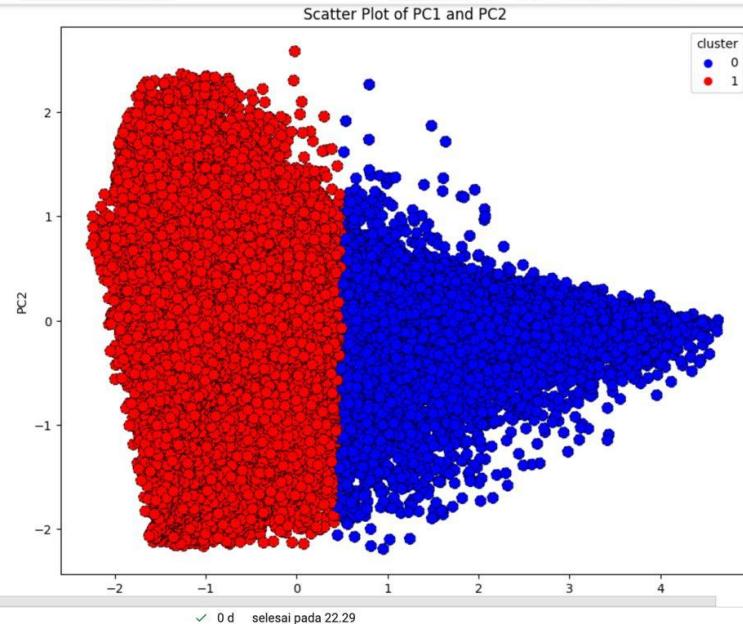
CLUSTERING

- Average of other columns within each cluster:

	index	FFP_JOIN_YEARS	LAST_TO_END	SEG_KM_SUM	\
cluster	0	15322.208296	4.675354	80.266505	23959.537573
cluster	1	40362.701483	4.029784	218.309112	7673.484843
	AVG_INTERVAL	FLIGHT_COUNT	SUM_YR_1		
cluster	0	37.881372	17.332872	6932.733840	
cluster	1	63.726391	5.394755	2101.431758	

Dari hasil, terlihat ada kesamaan segmen untuk variabel **FFP_JOIN_YEARS** dan sedikit kesamaan pada **AVG_INTERVAL**, sisanya terlihat perbedaan nilai antar segmen yang cukup jelas di sekitar **3x lipat** nilai lainnya.

PERCOBAAN 6 VARIABEL PERTAMA



Dari hasil gambar, terlihat pemisahan secara vertikal di sisi kiri dan kanan dari PCA. Cluster dipisahkan pada nilai antara 0-1 di PC1.

2B. PREPROCESSING

PERCOBAAN 6 VARIABEL KEDUA

FEATURE SELECTION

#	Column	Non-Null Count	Dtype
0	Point_NotFlight	51367	non-null
1	LAST_TO_END	51367	non-null
2	SEG_KM_SUM	51367	non-null
3	FFP_TIER	51367	non-null
4	FLIGHT_COUNT	51367	non-null
5	SUM_YR_1	51367	non-null

dtypes: float64(1), int64(5)
memory usage: 2.7 MB

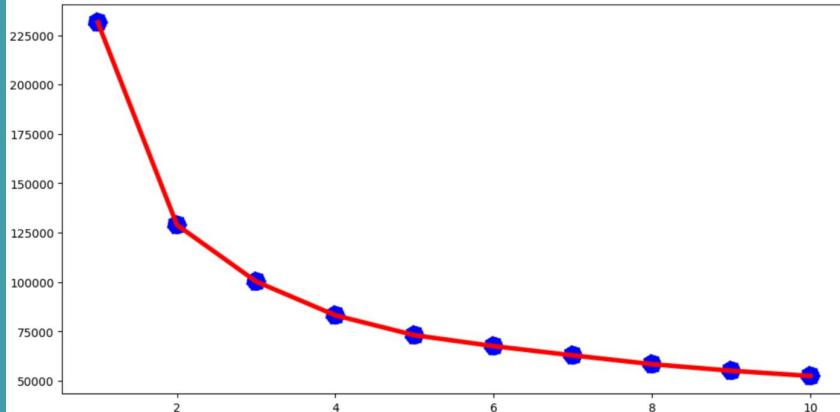
Kami memutuskan untuk mengganti variabel **FFP_JOIN_YEARS** dan **AVG_INTERVAL** dengan variabel **Point_NotFlight** dan **FFP_TIER**.

Karena selain **FFP_TIER** cenderung **segmented** sejak awal (bisa membantu memperbaiki model),

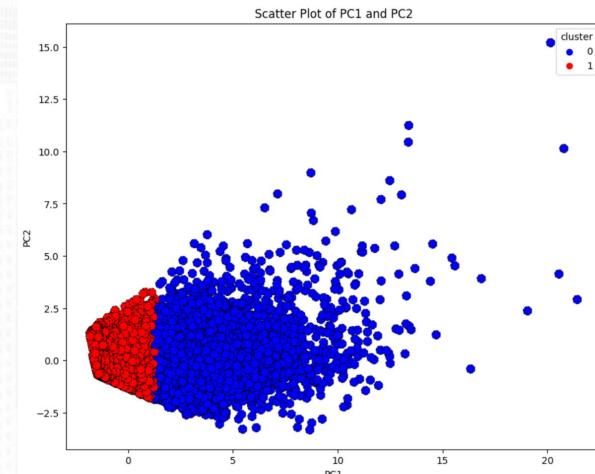
Point_NotFlight juga dianggap bisa membedakan **kecenderungan** customer yang aktif menggunakan point dengan yang pasif yang menunjukkan intensitas customer memesan tiket atau tidak, serta melengkapi aspek **monetary**, karena semakin poin tidak digunakan, **biaya** perusahaan yang keluar untuk memotong harga tiket juga lebih sedikit.

3B. MODELLING DAN EVALUASI

PERCOBAAN 6 VARIABEL KEDUA



Berdasarkan Elbow Method, sama seperti sebelumnya, **2 cluster** merupakan jumlah yang paling **direkomendasikan** ketimbang 3 cluster



Dari gambar **PCA**, terlihat **cluster 1** **tersegmentasi** dengan baik, lebih rinci dan **solid** meliputi sebagian kecil distribusi saja (**eksklusif**). Sedangkan **cluster 0** berisi data-data yang cenderung lebih menyebar dan **distribusinya lebih luas**, data-data yang terlihat seperti **outliers** semua masuk ke **cluster 0**.

4. INTERPRETASI

Average of other columns within each cluster:						
cluster	Point_NotFlight	LAST_TO_END	SEG_KM_SUM	FFP_TIER	FLIGHT_COUNT	SUM_YR_1
0	0.621226	67.670489	38037.394805	4.225369	25.498713	12361.105897
1	0.155251	215.718806	8995.594218	4.019663	6.255482	2554.002809

CLUSTER 0 :

- Memiliki banyak poin yang tidak digunakan, memberikan cukup spend money bagi perusahaan di sisi penghematan potongan harga
- Jarak antara penerbangan terakhir ke book tiket kebanyakan customer cukup kecil, bisa dibilang tipe ini adalah customer yang cukup aktif terbang dengan pesawat akhir-akhir ini.
- Secara total perjalanan dan jarak yang ditempuh, tipe ini aktif dalam penerbangan baik secara kuantitas maupun kualitas.
- Memberikan 6x lipat lebih banyak pendapatan kepada perusahaan ketimbang cluster lainnya.

CLUSTER 1 :

- Hampir semua aspek menunjukkan bahwa tipe ini sangat pasif dalam penerbangan
- Jarak terakhir penerbangan dengan rata-rata orang juga sangat jauh
- Poin yang tidak digunakan juga sedikit, bisa jadi karena pasif, poin juga jarang didapatkan

4. REKOMENDASI BISNIS

Rekomendasi Bisnis:

1. Meningkatkan Keterlibatan Pelanggan dalam Cluster 0:

- Perusahaan dapat meluncurkan program loyalitas tambahan yang dirancang khusus untuk pelanggan dalam Cluster 0. Program ini dapat mencakup keuntungan eksklusif, penghargaan khusus, atau penawaran diskon yang lebih menarik untuk mendorong mereka untuk tetap aktif dan meningkatkan tingkat keanggotaan.
- Meluncurkan undian seperti doorprize, bekerjasama dengan perusahaan lain agar promo yang didapatkan bisa sangat menguntungkan customer loyal.
- Personalisasi Pengalaman Pelanggan: Perusahaan dapat menggunakan data yang ada untuk memahami preferensi dan kebutuhan pelanggan dalam Cluster 0. Dengan mempersonalisasi pengalaman mereka, seperti penawaran spesial yang disesuaikan, rekomendasi penerbangan yang relevan, atau layanan yang disesuaikan, perusahaan dapat meningkatkan kepuasan pelanggan dan memperkuat keterikatan mereka.

2. Meningkatkan Keterlibatan Pelanggan dalam Cluster 1:

- Perusahaan dapat meluncurkan program stimulus penerbangan yang ditujukan khusus untuk pelanggan dalam Cluster 1. Program ini dapat mencakup penawaran diskon yang cukup besar agar customer terus mencoba memesan tiket, diskon yang ditawarkan jauh lebih besar ketimbang yang ditawarkan di cluster 0 demi keseimbangan keuangan.
- Komunikasi yang Terarah: Perusahaan perlu meningkatkan komunikasi dengan pelanggan dalam Cluster 1 untuk meningkatkan kesadaran tentang penawaran dan manfaat yang ditawarkan. Komunikasi yang terarah melalui saluran yang relevan, seperti email, pesan teks, atau media sosial, dapat membantu membangun keterlibatan yang lebih baik dan mendorong pelanggan untuk melakukan penerbangan lebih sering.