Rakamin Academy

UNSUPERVISED LEARNING

homework

Kelompok 4

pd.give_insight(✨)

Deskripsi Dataset

Code	Description
MEMBER_NO-b	: ID Member
FFP_DATE	: Frequent Flyer Program Join Date
FIRST_FLIGHT_DATE	: Tanggal Penerbangan pertama
GENDER	: Jenis Kelamin
FFP_TIER	: Tier dari Frequent Flyer Program
WORK_CITY	: Kota Asal
WORK_PROVINCE	: Provinsi Asal
WORK_COUNTRY	: Negara Asal
AGE	: Umur Customer
LOAD_TIME	: Tanggal data diambil
FLIGHT_COUNT	: Jumlah penerbangan Customer
BP_SUM	: Rencana Perjalanan
SUM_YR_1	: Fare Revenue
SUM_YR_2	: Votes Prices
SEG_KM_SUM	: Total jarak(km) penerbangan yg sudah dilakukan
LAST_FLIGHT_DATE	: Tanggal penerbangan terakhir
LAST_TO_END	: Jarak waktu penerbangan terakhir ke pesanan penerbangan paling akhir
AVG_INTERVAL	: Rata-rata jarak waktu
MAX_INTERVAL	: Maksimal jarak waktu
EXCHANGE_COUNT	: Jumlah penukaran
avg_discount	: Rata rata discount yang didapat customer
Points_Sum	: Jumlah poin yang didapat customer
Point_NotFlight	: point yang tidak digunakan oleh members

EDA
Pemahaman Data

Dari df.info() diperoleh kesimpulan bahwa :

1. Ada 62988 baris data
2. Terdapat 23 kolom
3. Ada 3 tipe data : integer, float dan object
4. Terlihat beberapa kolom memiliki Missing Value yang akan diperiksa kemudian.
5. Style penamaan kolom tidak rapih, akan diubah ke lowercase untuk memudahkan coding.

```
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
```

EDA

Statistik

1. Untuk memudahkan coding, maka semua kolom diganti menjadi lowercase sebagaimana terlihat pada df.describe() dibawah.
2. Dalam statistik sudah terlihat nilai-nilai outlier setidaknya pada kolom bp_sum, sum_yr_1, sum_yr_2, seg_km_sum. Akan dilakukan Handling Outlier.

```
df.columns = df.columns.str.lower()  # agar memudahkan coding
```

```
df.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	62988.00	62988.00	62568.00	62988.00	62988.00	62437.00	62850.00	62988.00	62988.00	62988.00	62988.00	62988.00	62988.00	62988.00	62988.00
mean	31494.50	4.10	42.48	11.84	10925.08	5355.38	5604.03	17123.88	176.12	67.75	166.03	0.32	0.72	12545.78	2.73
std	18183.21	0.37	9.89	14.05	16339.49	8109.45	8703.36	20960.84	183.82	77.52	123.40	1.14	0.19	20507.82	7.36
min	1.00	4.00	6.00	2.00	0.00	0.00	0.00	368.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	15747.75	4.00	35.00	3.00	2518.00	1003.00	780.00	4747.00	29.00	23.37	79.00	0.00	0.61	2775.00	0.00
50%	31494.50	4.00	41.00	7.00	5700.00	2800.00	2773.00	9994.00	108.00	44.67	143.00	0.00	0.71	6328.50	0.00
75%	47241.25	4.00	48.00	15.00	12831.00	6574.00	6845.75	21271.25	268.00	82.00	228.00	0.00	0.81	14302.50	1.00
max	62988.00	6.00	110.00	213.00	505308.00	239560.00	234188.00	580717.00	731.00	728.00	728.00	46.00	1.50	985572.00	140.00

```
df.describe(include=object)
```

	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

DATA CLEANING

Handling Missing Value

1. Tidak ditemukan duplicated data
2. Ditemukan Missing Value pada :
   a. Gender
   b. Work_city
   c. Work_province
   d. Work_country
   e. Age
   f. Sum_yr_1
   g. Sum_yr_2

Duplicated ¶

```
df.duplicated().sum()
```

```
0
```

Missing Values

```
df.isna().sum()
```

```
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
```

DATA CLEANING
Handling Missing Value

1. Pada data numerikal diimputasi dengan median untuk mengurangi beban skewness
2. Pada ‘gender’ (3 MV) dan ‘work_country’ (26 MV) di drop karena jumlahnya tidak signifikan.
3. ‘Work_city’ juga ternyata memiliki data yang berisi value ‘.’ (titik), di drop
4. Selanjutnya ‘work_city’ dan ‘work_province’ diisi dengan guangdong & guangzhou yang merupakan mayoritas value pada kolom.

```
df['sum_yr_1'].fillna(df['sum_yr_1'].median(), inplace=True)
df['sum_yr_2'].fillna(df['sum_yr_2'].median(), inplace=True)
df['age'].fillna(df['age'].median(), inplace=True)
```

```
df.dropna(subset=['gender'], inplace=True)
```

```
df.dropna(subset=['work_country'], inplace=True)
```

```
df = df[df['work_city'] != '.']
```

```
df['work_city'].fillna('guangzhou', inplace=True)
df['work_province'].fillna('guangdong', inplace=True)
```

DATA CLEANING
Handling Missing Value

1. Missing Value sudah selesai
2. Terdapat 696 data hilang dari awal 62988 menjadi 62292.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 62292 entries, 1 to 62987
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   member_no          62292 non-null  int64
 1   ffp_date           62292 non-null  object
 2   first_flight_date  62292 non-null  object
 3   gender             62292 non-null  object
 4   ffp_tier           62292 non-null  int64
 5   work_city          62292 non-null  object
 6   work_province      62292 non-null  object
 7   work_country       62292 non-null  object
 8   age                62292 non-null  float64
 9   load_time          62292 non-null  object
 10  flight_count       62292 non-null  int64
 11  bp_sum             62292 non-null  int64
 12  sum_yr_1           62292 non-null  float64
 13  sum_yr_2           62292 non-null  float64
 14  seg_km_sum         62292 non-null  int64
 15  last_flight_date   62292 non-null  object
 16  last_to_end        62292 non-null  int64
 17  avg_interval       62292 non-null  float64
 18  max_interval       62292 non-null  int64
 19  exchange_count     62292 non-null  int64
 20  avg_discount       62292 non-null  float64
 21  points_sum         62292 non-null  int64
 22  point_notflight    62292 non-null  int64
dtypes: float64(5), int64(10), object(8)
memory usage: 11.4+ MB
```

```
df.isna().sum()
```

```
member_no          0
ffp_date           0
first_flight_date  0
gender             0
ffp_tier           0
work_city          0
work_province      0
work_country       0
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
```

DATA CLEANING
Handling Tipe Data

Terdapat 4 kolom yang seharusnya bertipe data datetime, yaitu : ffp_date, first_flight_date, load_time & last_flight_date.

Untuk ffp_date, first_flight_date & load_time bisa dilakukan fungsi *pd.to_datetime(df['kolom'], format=date_format).*

Namun 'last_flight_date' karena ada perbedaan format penulisan jadi harus di-handling berbeda. Sebagaimana syntax di samping ini.

```
date_format = '%m/%d/%Y'
df['ffp_date'] = pd.to_datetime(df['ffp_date'], format=date_format)
df['first_flight_date'] = pd.to_datetime(df['first_flight_date'], format=date_format)
df['load_time'] = pd.to_datetime(df['load_time'], format=date_format)
```

```
df['last_flight_date'] = df['last_flight_date'].replace('2014/2/29  0:00:00', '2014-2-29')
```

```
# Membuat kolom baru untuk tanggal yang diubah formatnya
df['last_flight_date_2'] = df['last_flight_date'].apply(lambda x: pd.to_datetime(x, errors='coerce'))

# Menggunakan if else dan strftime untuk mengubah format tanggal
df['last_flight_date_2'] = df['last_flight_date_2'].apply(
    lambda x: x.strftime('%Y-%m-%d') if pd.notna(x) else '')
```

Merubah Kolom last_flight_date_2 Menjadi las_flight_date dengan Tipe Datetime

```
date_format = '%Y/%m/%d'
df['last_flight_date_2'] = pd.to_datetime(df['last_flight_date_2'], format=date_format)
```

```
median_date = df['last_flight_date_2'].median()
df['last_flight_date_2'].fillna(median_date, inplace=True)
```

```
df.drop(columns=['last_flight_date'], inplace=True)
df.rename(columns={'last_flight_date_2':'last_flight_date'}, inplace=True)
```

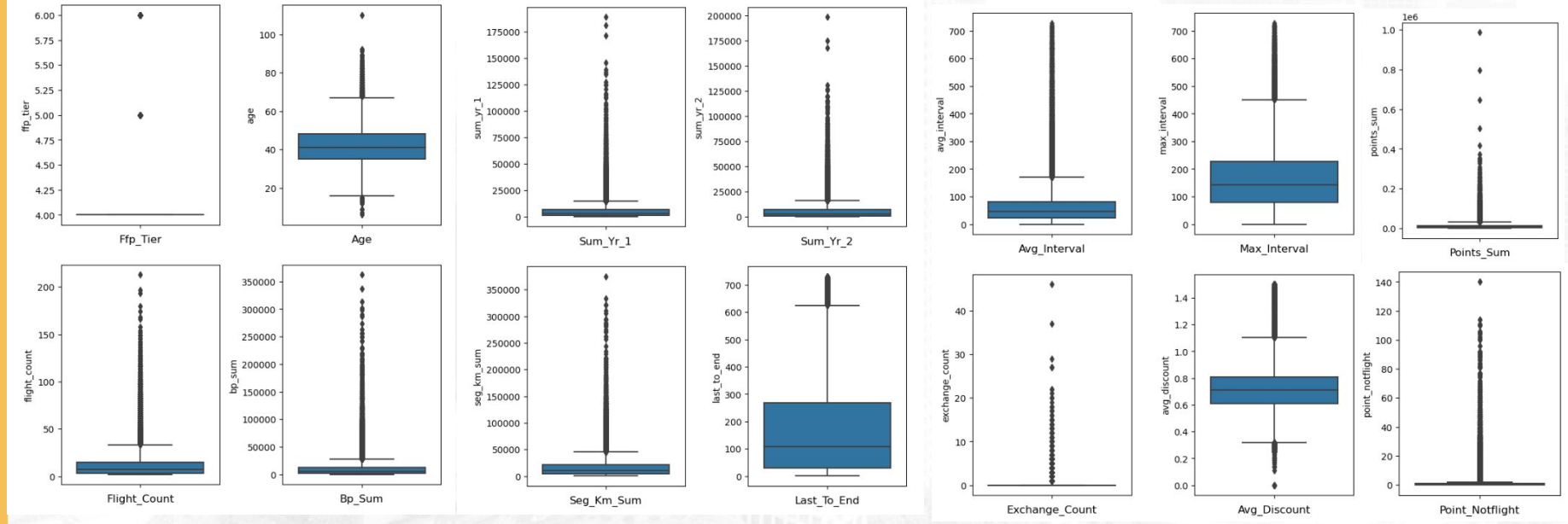
DATA CLEANING
Handling Datatype

Perubahan kolom dari tipe data object menjadi datetime64 sudah selesai dilakukan

```
df.info()
```

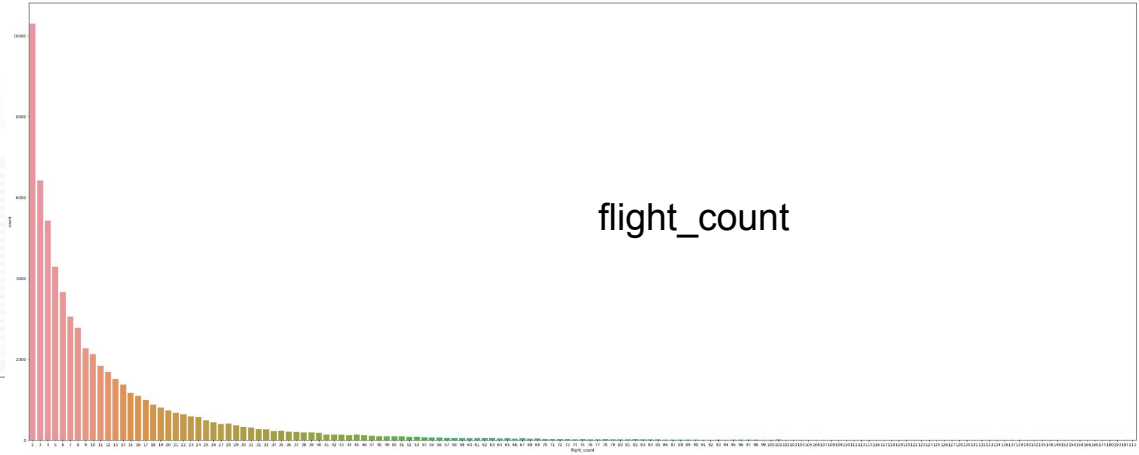
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 62292 entries, 1 to 62987
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   member_no          62292 non-null  int64
 1   ffp_date           62292 non-null  datetime64[ns]
 2   first_flight_date  62292 non-null  datetime64[ns]
 3   gender             62292 non-null  object
 4   ffp_tier           62292 non-null  int64
 5   work_city          62292 non-null  object
 6   work_province      62292 non-null  object
 7   work_country       62292 non-null  object
 8   age                62292 non-null  float64
 9   load_time          62292 non-null  datetime64[ns]
 10  flight_count       62292 non-null  int64
 11  bp_sum             62292 non-null  int64
 12  sum_yr_1           62292 non-null  float64
 13  sum_yr_2           62292 non-null  float64
 14  seg_km_sum         62292 non-null  int64
 15  last_to_end        62292 non-null  int64
 16  avg_interval       62292 non-null  float64
 17  max_interval       62292 non-null  int64
 18  exchange_count     62292 non-null  int64
 19  avg_discount       62292 non-null  float64
 20  points_sum         62292 non-null  int64
 21  point_notflight    62292 non-null  int64
 22  last_flight_date   62292 non-null  datetime64[ns]
dtypes: datetime64[ns](4), float64(5), int64(10), object(4)
memory usage: 11.4+ MB
```

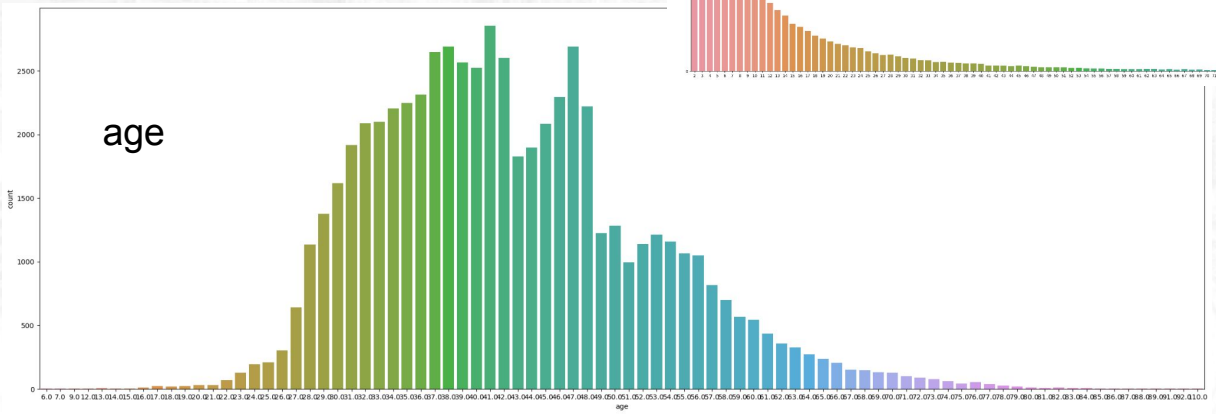
Univariate Analysis
Outliers



Univariate Analysis
Countplot

Dari *count plot* terlihat sebaran distribusi ekstrim dari outliers pada sample kolom ‘flight_count’ & ‘age’





Univariate Analysis
Data Unique Categorical

Melihat jumlah data unique pada Categorical nampaknya akan sangat rumit untuk di-explore karena dimensinya yang terlalu banyak.

```
hasil = []
for col in df.select_dtypes(include='object'):
    hasil.append([col, df[col].dtype, df[col].isna().sum(), (df[col].isna().sum()/len(df[col]))*100, df[col].nunique(), df[col].unique()[:4]])

output = pd.DataFrame(data=hasil, columns='kolom tipedata jumlah_null persen_null jumlah_unik contoh_unik'.split())
output
```

	kolom	tipedata	jumlah_null	persen_null	jumlah_unik	contoh_unik
0	gender	object	0	0.00	2	[Male, Female]
1	work_city	object	0	0.00	3231	[guangzhou, Los Angeles, guiyang, wulumuqishi]
2	work_province	object	0	0.00	1158	[beijing, CA, guizhou, guangdong]
3	work_country	object	0	0.00	117	[CN, US, FR, AN]

Multivariate Analysis

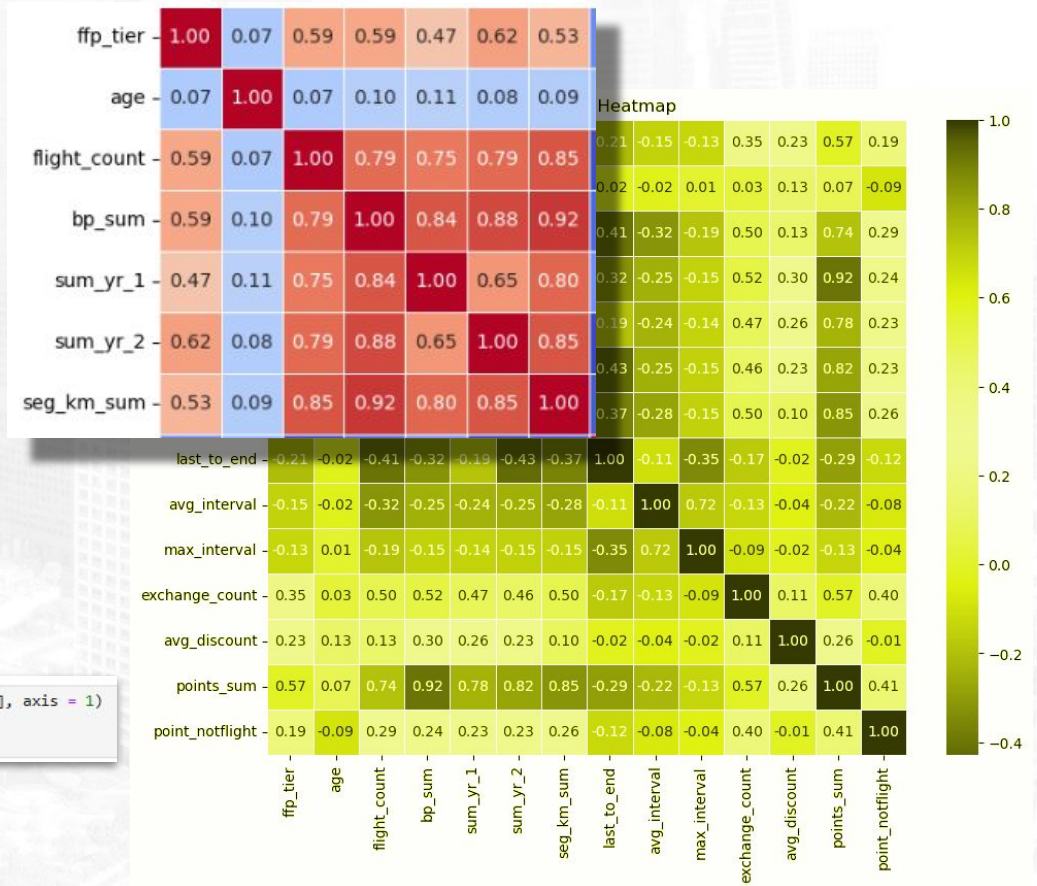
Correlation

Dari Correlation Heatmap terlihat selain ‘age’ antar fitur memiliki nilai koefisien diatas 0,5.

Fitur ‘seg_km_sum’ memiliki nilai koefisien tertinggi, yaitu 0.92

Oleh karena itu, fitur yang tidak memiliki nilai korelasi yang penting, di-drop. Yaitu : age, exchange_account, point_notflight, avg_interval, max_interval.

```
df = df.drop(['age','exchange_count','point_notflight','avg_interval','max_interval'], axis = 1)
```

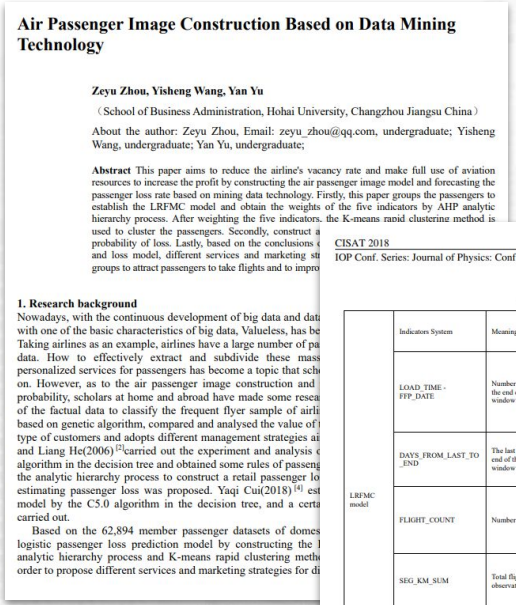


Feature

Feature Importance

Berdasarkan referensi yang kami dapatkan, bahwa fitur yang cocok untuk clustering adalah menggunakan **model LRFMC** yang merupakan pattern dalam industri penerbangan.

'First_flight_date' & 'last_flight_date' di-keep untuk keperluan feature engineering.



Air Passenger Image Construction Based on Data Mining Technology

Zeyu Zhou, Yisheng Wang, Yan Yu

（School of Business Administration, Hohai University, Changzhou Jiangsu China）

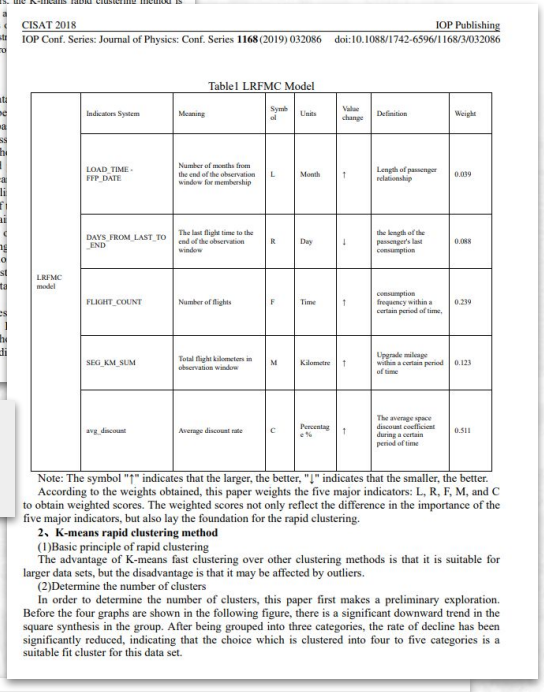
About the author: Zeyu Zhou, Email: zeyu_zhou@qq.com, undergraduate; Yisheng Wang, undergraduate; Yan Yu, undergraduate;

Abstract This paper aims to reduce the airline's vacancy rate and make full use of aviation resources to increase the profit by constructing the air passenger image model and forecasting the passenger loss rate based on mining data technology. Firstly, this paper groups the passengers to establish the LRFMC model and obtain the weights of the five indicators by AHP analytic hierarchy process. After weighting the five indicators, the K-means rapid clusterine method is used to cluster the passengers. Secondly, construct a
probability of loss. Lastly, based on the conclusions
and loss model, different services and marketing st
groups to attract passengers to take flights and to impro

1. Research background

Nowadays, with the continuous development of big data and dat
with one of the basic characteristics of big data, Valueless, has be
Taking airlines as an example, airlines have a large number of pa
data. How to effectively extract and subdivide these mass
personalized services for passengers has become a topic that sch
on. However, as to the air passenger image construction and
probability, scholars at home and abroad have made some resea
of the factual data to classify the frequent flyer sample of airli
based on genetic algorithm, compared and analysed the value of
type of customers and adopts different management strategies ai
and Liang He(2006) [2]carried out the experiment and analysis
algorithm in the decision tree and obtained some rules of passeng
the analytic hierarchy process to construct a retail passenger lo
estimating passenger loss was proposed. Yaqi Cui(2018) [4] est
model by the C5.0 algorithm in the decision tree, and a certa
carried out.

Based on the 62,894 member passenger datasets of domes
logistic passenger loss prediction model by constructing the
analytic hierarchy process and K-means rapid clustering meth
order to propose different services and marketing strategies for di



CISAT 2018 IOP Publishing

IOP Conf. Series: Journal of Physics: Conf. Series **1168** (2019) 032086 doi:10.1088/1742-6596/1168/3/032086

Table1 LRFMC Model

	Indicators System	Meaning	Symbol	Units	Value change	Definition	Weight
LRFMC model	LOAD_TIME - FFP_DATE	Number of months from the end of the observation window for membership	L	Month	↑	Length of passenger relationship	0.039
	DAYS_FROM_LAST_TO_END	The last flight time to the end of the observation window	R	Day	↓	the length of the passenger's last consumption	0.088
	FLIGHT_COUNT	Number of flights	F	Time	↑	consumption frequency within a certain period of time,	0.239
	SEG_KM_SUM	Total flight kilometers in observation window	M	Kilometre	↑	Upgrade mileage within a certain period of time	0.123
	avg_discount	Average discount rate	C	Percentage %	↑	The average space discount coefficient during a certain period of time	0.511

Note: The symbol "↑" indicates that the larger, the better, "↓" indicates that the smaller, the better.

According to the weights obtained, this paper weights the five major indicators: L, R, F, M, and C to obtain weighted scores. The weighted scores not only reflect the difference in the importance of the five major indicators, but also lay the foundation for the rapid clustering.

2、K-means rapid clustering method

(1)Basic principle of rapid clustering

The advantage of K-means fast clustering over other clustering methods is that it is suitable for larger data sets, but the disadvantage is that it may be affected by outliers.

(2)Determine the number of clusters

In order to determine the number of clusters, this paper first makes a preliminary exploration. Before the four graphs are shown in the following figure, there is a significant downward trend in the square synthesis in the group. After being grouped into three categories, the rate of decline has been significantly reduced, indicating that the choice which is clustered into four to five categories is a suitable fit cluster for this data set.

https://iopscience.iop.org/article/10.1088/1742-6596/1168/3/032086/pdf

https://towardsdatascience.com/data-driven-growth-with-python-part-2-customer-segmentation-5c019d150444

2.a Pemilihan Feature dari yang sudah ada

1. L = load_time - ffp_date : jumlah bulan dari membership - akhir observasi (bulan)
2. R = last_to_end : jumlah bulan dari penerbangan terakhir - akhir observasi (bulan)
3. F = flight_count : jumlah penerbangan dari pertama sampai observasi terakhir (bulan)
4. M = seg_km_sum : jumlah km yang dicapai (km)
5. C = avg_discount : discount rate

Load_time, ffp_date, last_to_end, flight_count, seg_km_sum, avg_discount

```
df = df[['member_no','load_time', 'ffp_date', 'last_to_end', 'flight_count', 'seg_km_sum', 'avg_discount','first_flight_date','last_flight_date']]
```

Feature
Feature Engineering

1. Masih dibutuhkan fitur rata-rata terbang dalam setahun, dan
2. Fitur membership dalam unit bulan (ffp_duration_month) untuk menemukan hitungan terbang per tahun.
3. Dibuat fitur baru bernama ‘fly_yearly’
4. Komputasinya menggunakan fitur ‘flight_count’ & ‘membership_year’ (durasi membership dalam tahun)

```
df['ffp_duration_days'] = (df['load_time'] - df['ffp_date'])
df['ffp_duration_month'] = round((df['load_time'] - df['ffp_date'])/np.timedelta64(1,'M'),0)
```

```
# Freq / years

df['membership_year'] = df['ffp_duration_month'] / 12
```

```
df['fly_yearly'] = df['flight_count'] / df['membership_year']
```

```
df.head()
```

	member_no	load_time	ffp_date	last_to_end	flight_count	seg_km_sum	avg_discount	first_flight_date	last_flight_date	ffp_duration_days	ffp_duration_month	membership_year	fly_yearly
1	28065	2014-03-31	2007-02-19	7	140	293678	1.25	2007-08-03	2014-03-25	2597 days	85.00	7.08	19.76
3	21189	2014-03-31	2008-08-22	97	23	281336	1.09	2008-08-23	2013-12-26	2047 days	67.00	5.58	4.12
4	39546	2014-03-31	2009-04-10	5	152	309928	0.97	2009-04-15	2014-03-27	1816 days	60.00	5.00	30.40
5	56972	2014-03-31	2008-02-10	79	92	294585	0.97	2009-09-29	2014-01-13	2241 days	74.00	6.17	14.92
6	44924	2014-03-31	2006-03-22	1	101	287042	0.97	2006-03-29	2014-03-31	2931 days	96.00	8.00	12.62

Feature
Feature Engineering

Ternyata masih ditemukan pada fitur, value yang tidak sesuai dan tidak logis. Dimana ada value ‘first_flight_date’ > ‘last_flight_date’, sehingga menghasilkan nilai NaT pada fitur baru ‘fly_yearly’.

Oleh karena itu perlu dibersihkan dari dataset dengan menggunakan syntax di bawah.

```
df = df[df['last_flight_date'] > df['first_flight_date']]
```

	FIRST_FLIGHT_DATE	NEW_LAST_FLIGHT_DATE
3293	2015-03-09	2014-01-08
3733	2015-02-15	2014-03-27
16393	2015-05-30	2014-02-17
25240	2014-07-14	2013-09-15
28231	2015-04-03	2014-03-24
33198	2014-09-11	2014-02-09

K-Means
PreProcessing

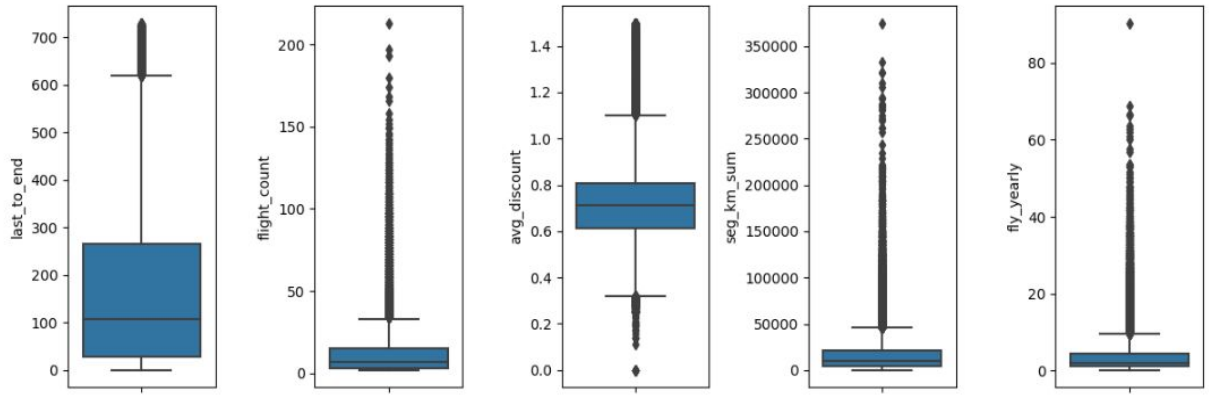
1. Handling Outliers

   Kelima fitur terpilih untuk proses selanjutnya adalah:

   a. Last_to_end
   b. Flight_count
   c. Avg_discount
   d. Seg_km_sum
   e. fly_yearly

   Fitur lain yang tidak diikutkan karena hanya diperlukan untuk membentuk fitur baru ‘fly_yearly’.

```
nums =['last_to_end','flight_count','avg_discount','seg_km_sum','fly_yearly']

plt.figure(figsize = (12,4))
for i in range(0, len(nums)):
    plt.subplot(1, 5, i+1)
    sns.boxplot(y = df[nums[i]], orient='v')
    plt.tight_layout()
```



K-Means
PreProcessing

1. Handling Outliers

Z-score dilakukan karena avg_discount relatif distribusi normal.

4 fitur lainnya dilakukan **log-transformation** dikarenakan kecenderungan mereka yang right-skewed.

IQR digunakan untuk mengatasi outlier yang tidak ter-cover oleh z-score dan log-transformation.

```
from scipy import stats

z_scores = np.abs(stats.zscore(df['avg_discount']))
filt_ent = (z_scores < 3)
df = df[filt_ent]
```

```
df['last_to_end'] = np.log(df['last_to_end'])
df['flight_count'] = np.log(df['flight_count'])
df['seg_km_sum'] = np.log(df['seg_km_sum'])
df['fly_yearly'] = np.log(df['fly_yearly'])
```

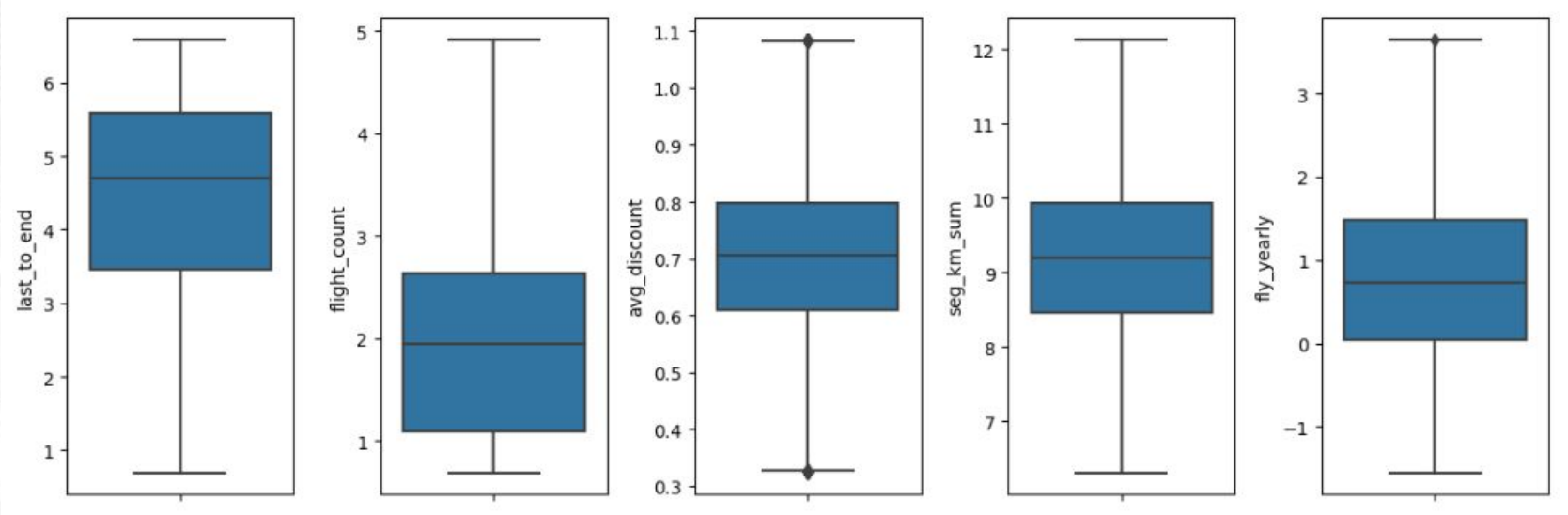
```
# pembuangan outlier

print(f'Jumlah baris sebelum memfilter outlier: {len(df)}')

filtered_entries = np.array([True] * len(df))
for col in ['last_to_end','flight_count','avg_discount','seg_km_sum','fly_yearly']:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    low_limit = Q1 - (IQR * 1.5)
    high_limit = Q3 + (IQR * 1.5)

    filtered_entries = ((df[col] >= low_limit) & (df[col] <= high_limit)) & filtered_entries

df = df[filtered_entries]

print(f'Jumlah baris setelah memfilter outlier: {len(df)}')
```

```
Jumlah baris sebelum memfilter outlier: 60767
Jumlah baris setelah memfilter outlier: 58403
```

K-Means
PreProcessing

1. Handling Outliers

   Sisa data setelah Handling Outliers adalah 58403 (dari 62988 pada dataset awal)



K-Means
PreProcessing

2. Standardization

Sisa data setelah Handling Outliers adalah 58403 (dari 62988 pada dataset awal)

```
new_df.head()
```

	seg_km_sum	last_to_end	avg_discount	fly_yearly	flight_count
0	2.92	-0.93	2.29	1.07	1.15
1	2.93	-1.83	1.65	2.29	2.64
2	2.79	-1.63	2.48	1.28	1.53
3	2.90	0.03	1.69	0.85	1.60
4	2.83	-1.03	2.17	1.42	1.67

Standardization

```
from sklearn.preprocessing import StandardScaler
feats = ['seg_km_sum','last_to_end','avg_discount','fly_yearly','flight_count']
X = df[feats].values
X_std = StandardScaler().fit_transform(X)
new_df = pd.DataFrame(data=X_std, columns=feats)
```

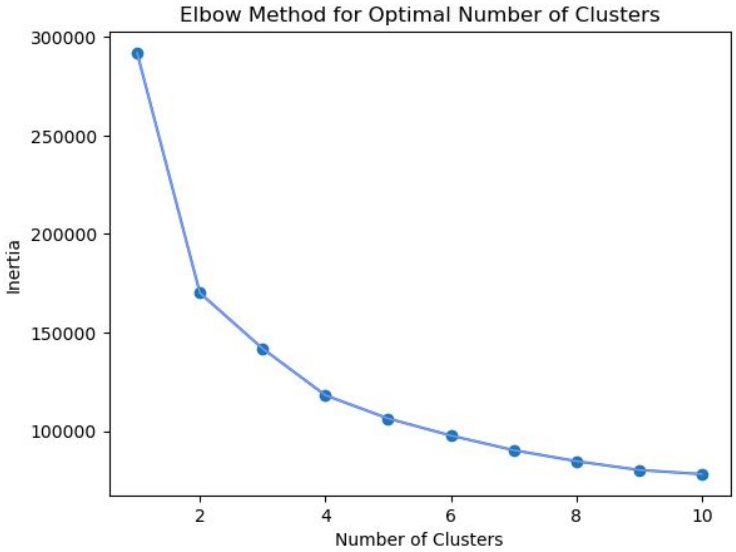
```
new_df.describe()
```

	seg_km_sum	last_to_end	avg_discount	fly_yearly	flight_count
count	58403.00	58403.00	58403.00	58403.00	58403.00
mean	-0.00	0.00	0.00	0.00	0.00
std	1.00	1.00	1.00	1.00	1.00
min	-2.91	-2.59	-2.66	-2.27	-1.38
25%	-0.75	-0.67	-0.65	-0.70	-0.95
50%	-0.02	0.19	0.02	-0.02	-0.03
75%	0.72	0.80	0.67	0.71	0.71
max	2.97	1.49	2.68	2.88	3.17

Standarisasi sukses karena :

1. Mean mendekati 0
2. std deviasi mendekati 1

K-Means
Elbow Methods



Berdasarkan Elbow Methods, nilai n_cluster adalah 4.

```
inertia
```

```
[292015.0000000001,
 170135.1756767065,
 141877.49628322435,
 118175.11342192024,
 106376.09612504492,
 97709.28622160411,
 90298.91165947267,
 85713.54181567705,
 80228.3332059756,
 76928.36019562783]
```

```
(pd.Series(inertia) - pd.Series(inertia).shift(-1))
```

```
0   121879.82
1    28257.68
2    23702.38
3    11799.02
4     8666.81
5     7410.37
6     4585.37
7     5485.21
8     3299.97
9         NaN
dtype: float64
```

K-Means
Statistic K-Means

```
new_df['cluster'] = kmeans.labels_
new_df.head()
```

	seg_km_sum	last_to_end	avg_discount	fly_yearly	flight_count	cluster
0	2.92	-0.93	2.29	1.07	1.15	0
1	2.93	-1.83	1.65	2.29	2.64	0
2	2.79	-1.63	2.48	1.28	1.53	0
3	2.90	0.03	1.69	0.85	1.60	0
4	2.83	-1.03	2.17	1.42	1.67	0

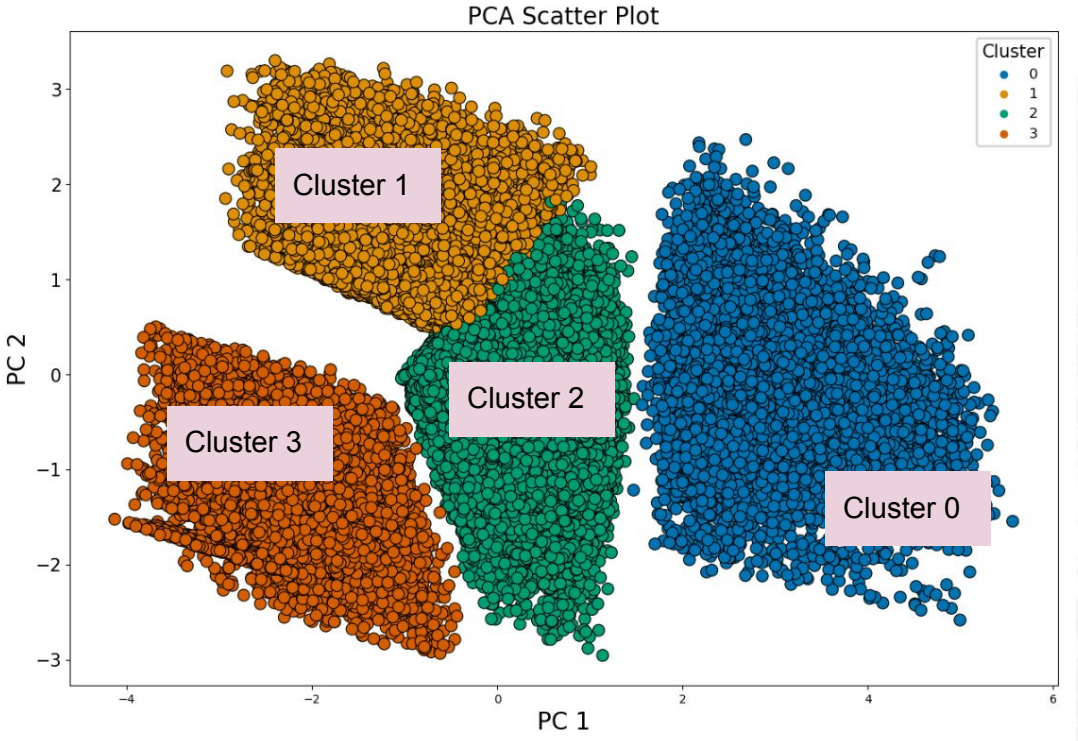
```
cluster_stats = display(new_df.groupby('cluster').agg(['mean','median']))
```

	seg_km_sum		last_to_end		avg_discount		fly_yearly		flight_count	
	mean	median	mean	median	mean	median	mean	median	mean	median
cluster										
0	1.25	1.23	-1.14	-1.15	0.18	0.16	1.17	1.16	1.36	1.34
1	-0.74	-0.72	0.49	0.66	-1.14	-1.07	-0.75	-0.74	-0.86	-0.95
2	0.29	0.28	-0.02	0.12	0.02	0.02	0.29	0.29	0.28	0.24
3	-0.95	-0.93	0.67	0.83	0.89	0.80	-0.86	-0.84	-0.92	-0.95

K-Means
Visualisasi PCA

Terbentuk 4 cluster, dimana cluster ke-4 (index 3) terpisah dari 3 cluster lainnya.

Sedangkan cluster 0 dihimpit oleh cluster 1 berada di kiri dan 2 berada di kanan.



K-Means

Cluster Statistic pada DF Awal

```
display(df_merged[nums].groupby('cluster').agg(['mean','median','min','max']))
```

	age				flight_count				bp_sum			
	mean	median	min	max	mean	median	min	max	mean	median	min	max
cluster												
0	42.99	42.00	13.00	89.00	29.18	25.00	5	137	25805.20	20676.00	413	219390
1	41.43	40.00	6.00	92.00	3.56	3.00	2	12	2101.30	1788.00	0	21607
2	42.21	41.00	15.00	110.00	10.21	9.00	2	38	8949.20	7543.00	765	140743
3	42.31	41.00	7.00	92.00	3.37	3.00	2	15	3609.33	2859.00	0	75605

	seg_km_sum				last_to_end				avg_discount				fly_yearly			
	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
0	40916.58	34272.00	3757	195712	30.48	16.00	2	484	0.73	0.73	0.34	1.09	1.95	1.94	-0.20	3.68
1	5681.29	4895.00	716	49671	263.69	220.00	2	730	0.54	0.55	0.32	0.71	-0.01	0.00	-1.55	2.00
2	15395.77	13395.50	1840	122320	130.90	100.00	2	694	0.71	0.71	0.33	1.08	1.05	1.05	-1.13	2.84
3	4750.04	3966.00	552	61160	311.90	281.00	2	730	0.83	0.82	0.65	1.08	-0.12	-0.09	-1.54	1.79

K-Means

Deskripsi Cluster Berdasarkan Statistic

	age				flight_count				bp_sum				seg_km_sum			
cluster	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
0	42.99	42.00	13.00	89.00	29.18	25.00	5.00	137.00	25,805.20	20,676.00	413.00	219,390.00	40,916.58	34,272.00	3,757.00	195,712.00
1	41.43	40.00	6.00	92.00	3.56	3.00	2.00	12.00	2,101.30	1,788.00	0.00	21,607.00	5,681.29	4,895.00	716.00	49,671.00
2	42.21	41.00	15.00	110.00	10.21	9.00	2.00	38.00	8,949.20	7,543.00	765.00	140,743.00	15,395.77	13,395.50	1,840.00	122,320.00
3	42.31	41.00	7.00	92.00	3.37	3.00	2.00	15.00	3,609.33	2,859.00	0.00	75,605.00	4,750.04	3,966.00	552.00	61,160.00

	last_to_end				avg_discount				membership_year				fly_yearly			
cluster	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
0	30.48	16.00	2.00	484.00	0.73	0.73	0.34	1.09	4.31	3.83	1.00	9.42	1.95	1.94	-0.20	3.68
1	263.69	220.00	2.00	730.00	0.54	0.55	0.32	0.71	3.94	3.33	1.00	9.42	-0.01	0.00	-1.55	2.00
2	130.90	100.00	2.00	694.00	0.71	0.71	0.33	1.08	3.93	3.33	1.00	9.42	1.05	1.05	-1.13	2.84
3	311.90	281.00	2.00	730.00	0.83	0.82	0.65	1.08	4.11	3.67	1.00	9.42	-0.12	-0.09	-1.54	1.79

K-Means

Deskripsi Cluster Berdasarkan Statistic

	age				flight_count				bp_sum				seg_km_sum				last_to_end				avg_discount				membership_year				fly_yearly			
	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max	mean	median	min	max
cluster																																
0	42.99	42.00	13.00	89.00	29.18	25.00	5	137	25805.20	20676.00	413	219390	40916.58	34272.00	3757	195712	30.48	16.00	2	484	0.73	0.73	0.34	1.09	4.31	3.83	1.00	9.42	1.95	1.94	-0.20	3.68
1	41.43	40.00	6.00	92.00	3.56	3.00	2	12	2101.30	1788.00	0	21607	5681.29	4895.00	716	49671	263.69	220.00	2	730	0.54	0.55	0.32	0.71	3.94	3.33	1.00	9.42	-0.01	0.00	-1.55	2.00
2	42.21	41.00	15.00	110.00	10.21	9.00	2	38	8949.20	7543.00	765	140743	15395.77	13395.50	1840	122320	130.90	100.00	2	694	0.71	0.71	0.33	1.08	3.93	3.33	1.00	9.42	1.05	1.05	-1.13	2.84
3	42.31	41.00	7.00	92.00	3.37	3.00	2	15	3609.33	2859.00	0	75605	4750.04	3966.00	552	61160	311.90	281.00	2	730	0.83	0.82	0.65	1.08	4.11	3.67	1.00	9.42	-0.12	-0.09	-1.54	1.79

Cluster 0:

1. Didominasi frequent flyer, terlihat dari total jarak penerbangan yang sudah dilakukan (**seg_km_sum**) dan angka (**fly_yearly)**.
2. Business traveller, (**bp_sum**) yang tinggi menandakan membuat perencanaan
3. Jika dilihat dari rentang usia kemungkinan besar terdapat business owner, director, manager yang traveling bersama anaknya.

Cluster 1:

1. Memiliki rentang umur 15 - 110, kemungkinan besar ini adalah Family Traveler. Beberapa keluarga suka berlibur bersama orang tua dan anak-anaknya pada musim libur.
2. mereka akan terbang walau dengan diskon yang terendah dibandingkan dengan cluster yang lain, ini artinya moment lebih penting daripada diskon.
3. biasanya memang pada musim liburan tiket pesawat relatif lebih mahal.

Cluster 2:

1. Rentang umur remaja - 110 tahun, kemungkinan besar ini adalah tipe orang yang adventurer atau travel enthusiast yang bisa individu maupun via travel agent.
2. Total jarak(km) penerbangan yg sudah mereka lakukan cukup tinggi (**seg_km_sum**)
3. Cukup sering bepergian (**flight_count** & **last_to_end**)
4. Cluster ini membuat perencanaan perjalanan yang sangat matang (**bp_sum**, max)

Cluster 3:

1. Kategori terakhir adalah occasional traveler, bisa dilihat dari nilai-nilai kolom seperti (**last_to_end**) yang menunjukan rata-rata jarak memesan tiket adalah 311 hari.
2. Cluster ini sensitif terhadap diskon.

K-Means

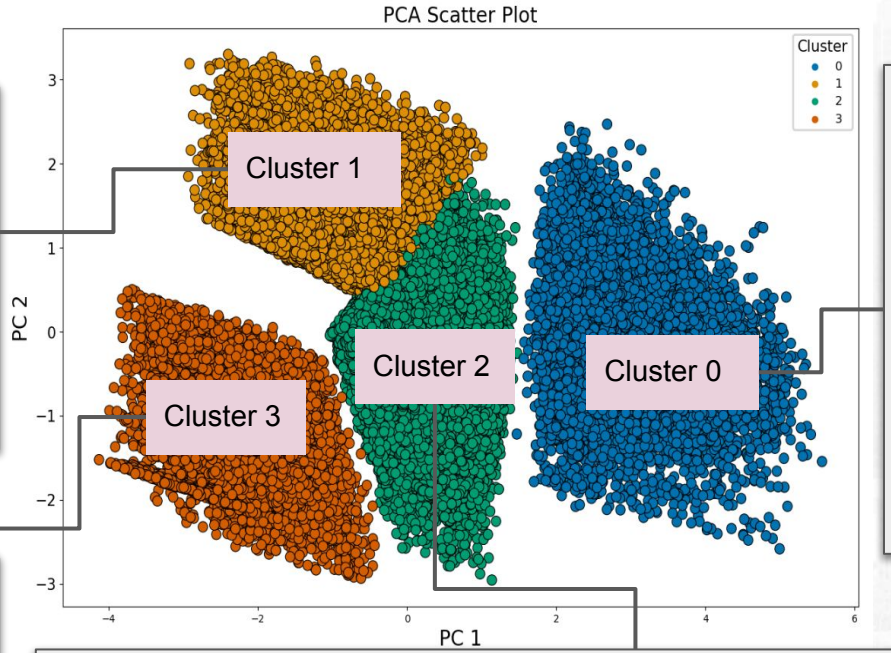
Deskripsi Cluster Berdasarkan Statistic

Cluster 1:

1. Memiliki rentang umur 15 - 110, kemungkinan besar ini adalah Family Traveler. Beberapa keluarga suka berlibur bersama orang tua dan anak-anaknya pada musim libur.
2. mereka akan terbang walau dengan diskon yang terendah dibandingkan dengan cluster yang lain, ini artinya moment lebih penting daripada diskon.
3. biasanya memang pada musim liburan tiket pesawat relatif lebih mahal.

Cluster 3:

1. Kategori terakhir adalah occasional traveler, bisa dilihat dari nilai-nilai kolom seperti (**last_to_end**) yang menunjukan rata-rata jarak memesan tiket adalah 311 hari.
2. Cluster ini sensitif terhadap diskon.



Cluster 0:

1. Didominasi frequent flyer, terlihat dari total jarak penerbangan yang sudah dilakukan (**seg_km_sum**) dan angka (**fly_yearly)**.
2. Business traveller, (**bp_sum**) yang tinggi menandakan membuat perencanaan
3. Jika dilihat dari rentang usia kemungkinan besar terdapat business owner, director, manager yang traveling bersama anaknya.

Cluster 2:

1. Rentang umur remaja - 110 tahun, kemungkinan besar ini adalah tipe orang yang adventurer atau travel enthusiast yang bisa individu maupun via travel agent.
2. Total jarak(km) penerbangan yg sudah mereka lakukan cukup tinggi (**seg_km_sum**)
3. Cukup sering bepergian (**flight_count** & **last_to_end**)
4. Cluster ini membuat perencanaan perjalanan yang sangat matang (**bp_sum**, max)

K-Means

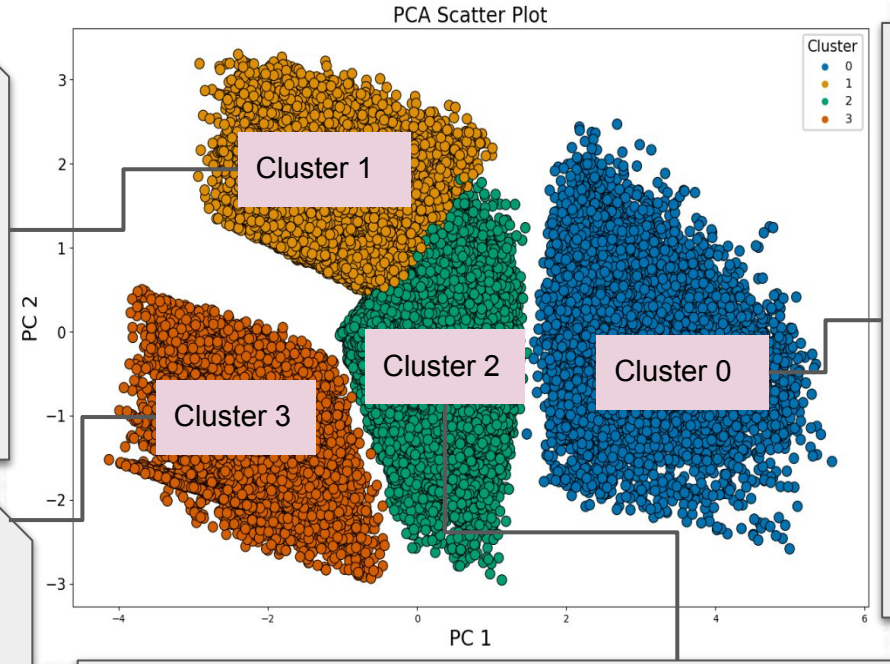
Rekomendasi Bisnis Berdasarkan Cluster

Cluster 1 - Family Travelers:

1. **Paket Liburan Keluarga All-Inclusive**: Tawarkan paket liburan yang mencakup semua aspek perjalanan, mulai dari tiket pesawat, akomodasi, hingga kegiatan keluarga selama liburan.
2. **Program Diskon Keluarga**: Buat program diskon khusus untuk keluarga yang melakukan perjalanan bersama, seperti diskon untuk anak-anak atau fasilitas khusus untuk keluarga besar.
3. **Liburan Tematik Keluarga**: Sediakan opsi liburan tematik yang cocok untuk berbagai usia, seperti liburan petualangan alam atau liburan budaya yang sesuai untuk seluruh keluarga.

Cluster 3 - Occasional Travelers:

1. **Program Diskon Tertarget**: Buat program diskon yang ditargetkan untuk menarik perhatian para pelancong gelegenheits yang sensitif terhadap harga.
2. **Paket Perjalanan Akhir Pekan**: Sediakan paket perjalanan singkat yang cocok untuk liburan akhir pekan atau cuti pendek dengan harga terjangkau.
3. **Layanan Pengingat Perjalanan**: Tawarkan layanan pengingat perjalanan melalui pesan teks atau email untuk membantu para pelancong yang sering melupakan perencanaan perjalanan.



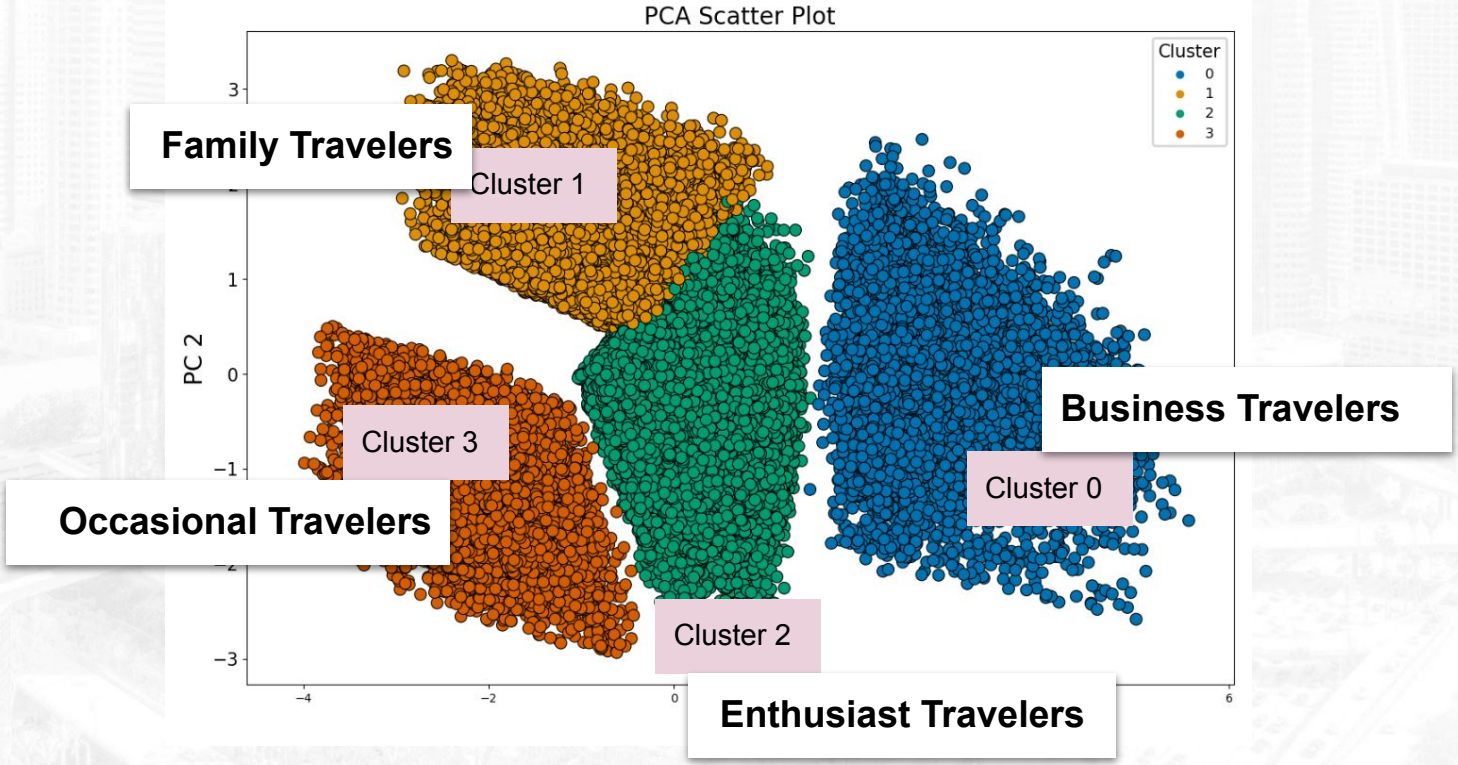
Cluster 0 - Frequent Flyers & Business Travelers:

1. **Layanan Keanggotaan VIP**: Tawarkan layanan keanggotaan khusus untuk frequent flyers yang memberikan keuntungan seperti akses prioritas, diskon tambahan, dan akses ke lounge bandara.
2. **Paket Perjalanan Bisnis**: Sediakan paket perjalanan khusus untuk pelaku bisnis dengan fasilitas seperti pemesanan tiket, akomodasi, dan layanan transportasi yang terintegrasi.
3. **Perjalanan Keluarga dengan Fasilitas Tambahan**: Untuk pelaku bisnis yang sering bepergian dengan keluarga, tawarkan paket perjalanan khusus yang mencakup fasilitas untuk anak-anak, seperti permainan di pesawat atau hiburan di bandara.

Cluster 2 - Travel Enthusiasts & Adventurers:

1. **Paket Petualangan Khusus**: Tawarkan paket perjalanan petualangan yang mengeksplor destinasi eksotis, olahraga ekstrem, atau kegiatan khusus lainnya yang menarik bagi para petualang.
2. **Panduan Wisata Lokal**: Sediakan panduan wisata lokal yang mendalam dan unik untuk membantu para petualang menjelajahi sisi tersembunyi suatu tempat
3. **Paket Perjalanan Khusus Minat**: Buat paket perjalanan berdasarkan minat khusus, seperti fotografi, kuliner, atau budaya lokal yang mengundang para petualang.

K-Means
Garis Besar Cluster



Penutup

Demikianlah laporan ini disajikan sebagai bagian dari
Homework Unsupervised Learning Week - 16

Kelompok 4
pd.give_insight(✨)

Disusun oleh :
Amarindra Ardinova | Kenneth Wahyudi | Anisa Millah T. | Elkania Samanta | M. Haniff