Clustering – Flight Customer Segmentation

Team:

1. Akhmad Yuzfa Salvian Idris
2. Arief Rahman Hakim
3. Bernardus Valentino
4. Milenia Nadia Afifah Puspitasari
5. Sean Frederic Wijaya

Exploratory Data Analysis

- Dataframe memiliki total 62.988 baris dan 23 kolom
- Tidak ada data yang duplikat
- Terdapat null values di kolom GENDER, WORK_CITY, WORK_PROVINCE, WORK_COUNTRY, AGE, SUM_YR_1, SUM_YR_2
- Distribusi nilai hampir normal: FLIGHT_COUNT, AVG_INTERVAL, MAX_INTERVAL, EXCHANGE_COUNT, Point_NotFlight
- Distribusi nilai tidak normal: BP_SUM, SUM_YR_1, SUM_YR_2, SEG_KM_SUM, LAST_TO_END, Points_Sum
- Semua feature memiliki outlier dan kebanyakan berdistribusi right skewed
- Insight:
  - Rata-rata diskon dari keseluruhan flight yang dilakukan: 0.72 %
  - Customer flight rata-rata berumur 42 tahun
  - Customer flight didominasi oleh gender male

Data Preprocessing

Data awal sebelum preprocessing: (62.988, 23)

1. *Dropping Wrong Data*

   Kami memulai membersihkan data dengan men-drop baris di kolom 'LAST_FLIGHT_DATE' dengan value '2014/2/29 0:00:00' karena:

   i. Tahun 2014 bukanlah tahun kabisat, tahun kabisat adalah tahun genap yang bisa dibagi oleh 4 tanpa mempunyai sisa pembagian seperti 2012, 2016, 2020, dst
   ii. Format data mempunyai waktu dimana data lainnya pada kolom ini tidak mempunyai waktu
   iii. Urutan tanggal adalah YYYY/MM/DD dimana data lainnya adalah MM/DD/YYYY

   Jadi kami menyimpulkan bahwa ini adalah kesalahan inputan admin. Kami juga perlu mengganti *data type* dari kolom ini untuk *feature engineering.*

   Data setelah *dropping wrong data*: (62.567, 23).

2. *Changing Datatype*
   a. Ada 5 kolom yang kami ubah tipe data nya, yaitu:
      i. 'FFP_TIER' = int64 → object / str
         karena merupakan data kategorikal dan untuk melihat statistika deskriptif nya
      ii. 'FFP_DATE' = object / str → datetime64
         untuk *feature engineering*
      iii. 'FIRST_FLIGHT_DATE' = object / str → datetime64
         untuk *feature engineering*
      iv. 'LOAD_TIME' = object / str → datetime64
         untuk *feature engineering*
      v. 'LAST_FLIGHT_DATE' = object / str → datetime64
         untuk *feature engineering*

3. *Missing Value*
   a. *Missing Value*
      i. 'GENDER' : 3
      ii. 'WORK_CITY' : 2.255
      iii. 'WORK_PROVINCE' : 3.226
      iv. 'WORK_COUNTRY' : 25
      v. 'AGE' : 419
      vi. 'SUM_YR_1' : 551
      vii. 'SUM_YR_2' : 138
   b. *Imputation*
      i. 'AGE': Mode
         diimputasi dengan umur mayoritas anggota *frequent flyer.*

ii. 'GENDER': Mode
diimputasi dengan gender mayoritas anggota *frequent flyer.*
iii. 'SUM_YR_1': Median
menggunakan median agar lebih *robust* dibandingkan jika menggunakan mean karena distribusi skewed.
iv. 'SUM_YR_2': Median
menggunakan median agar lebih *robust* dibandingkan jika menggunakan mean karena distribusi skewed.

c. *Drop*
Fitur yang didrop adalah 'WORK_CITY', 'WORK_PROVINCE', dan 'WORK_COUNTRY' karena tidak memungkinkan untuk mengisi *missing values* pada fitur-fitur tersebut.

Data setelah menghilangkan missing values: (58.843, 23)

4. *Duplicated Data*
Tidak ada data duplikat.

5. *Outliers*
*Dihandle m*enggunakan Z-score karena akan terlalu banyak baris yang akan didrop jika menggunakan IQR, yakni 20.000 baris.
Data setelah menghilangkan outliers: (51.316, 23)

6. *Label Encoding / One Hot Encoding*
Encoding tidak dilakukan karena fitur data kategorikal tidak kami gunakan untuk clustering.

Feature Engineering

1. *Length of Membership Time*
   Lama waktu keanggotaan customer hingga data diambil dapat dihitung berdasarkan tanggal customer memulai keanggotaan dan tanggal data diambil. Fitur ini dapat digunakan untuk melihat apakah customer tersebut adalah pengguna lama atau bukan.
   Langkah-langkah:
   a. Buat fitur baru Membership_Length dengan mengurangkan fitur FFP_DATE (Frequent Flyer Program Join Date) dari LOAD_TIME (Tanggal data diambil) untuk mendapatkan lama waktu keanggotaan customer hingga tanggal data diambil.
      Code:
      ```
      df['Membership_Length'] = df['LOAD_TIME'] - df['FFP_DATE']
      ```
   b. Untuk memudahkan komputasi nantinya, ambil bagian angka dari fitur into_trending dan ubah tipe manjadi integer (sebelumnya adalah timedelta64[ns]).
      Code:
      ```
      df['Membership_Length'] = df['Membership_Length'].astype('str')
      df['Membership_Length'] = df['Membership_Length'].str.split().str[0]
      df['Membership_Length'] = df['Membership_Length'].astype('int')
      ```

2. *The Average Fare per Kilometer*
   Tarif rata-rata per kilometer dapat dihitung berdasarkan total tarif pada tahun pertama, total tarif pada tahun kedua, dan total kilometer penerbangan. Fitur ini dapat digunakan untuk melihat customer mana yang dapat mendatangkan keuntungan yang lebih besar.
   Langkah-langkah:
   a. Buat fitur baru AvgFare_perKM dengan menjumlahkan fitur SUM_YR_1 (total tarif pada tahun pertama) dan SUM_YR_2 (total tarif pada tahun kedua). Kemudian dibagi dengan fitur SEG_KM_SUM (total jarak(km) penerbangan yg sudah dilakukan).
      Code:
      ```
      df['AvgFare_perKM']=(df['SUM_YR_1'] + df['SUM_YR_2'])/df['SEG_KM_SUM']
      ```

Data sebelum feature engineering: (51.316, 23)
Data setelah feature engineering: (51.316, 25)

Normalization/Standardization

1. StandardScaler
   Dilakukan normalisasi untuk fitur-fitur yang belum berdistribusi normal, yakni terdapat 13 fitur.
   Code:
   ```
   from sklearn.preprocessing import StandardScaler

   for i in
   ['FLIGHT_COUNT','BP_SUM','SUM_YR_1','SUM_YR_2','SEG_KM_SUM','LAST_TO_END','AVG_INTERVAL','MAX_INTERVAL','EXCHANGE_COUNT','Points_Sum','Point_NotFlight','Membership_Length','AvgFare_perKM']:

       df[i] = StandardScaler().fit_transform(df[i].values.reshape(len(df), 1))
   ```

2. MinMaxScaler
   Dilakukan scaling agar skala nilai dari setiap fitur sama, yakni minimum 0 dan maksimum 1 untuk memastikan agar interval setiap fitur sudah sama sebelum dilakukan clustering. Terdapat 15 fitur yang dilakukan scaling.
   Code:
   ```
   from sklearn.preprocessing import MinMaxScaler

   for i in
   ['AGE','FLIGHT_COUNT','BP_SUM','SUM_YR_1','SUM_YR_2','SEG_KM_SUM','LAST_TO_END','AVG_INTERVAL','MAX_INTERVAL','EXCHANGE_COUNT','Points_Sum','avg_discount','Point_NotFlight','Membership_Length','AvgFare_perKM']:

       df[i] = MinMaxScaler().fit_transform(df[i].values.reshape(len(df), 1))
   ```

Feature Selection

Karena terlalu banyak fitur dari data, dan tidak setiap fitur memberikan informasi yang berharga, jadi kami hanya memilih 3 fitur yang kami rasa dapat memberikan informasi yang berharga dalam penentuan segmentasi customer penerbangan nantinya. Berikut ada adalah tiga fitur yang kami ambil, dengan 2 fitur merupakan hasil dari feature engineering, yakni:

1. Length of Membership Time (Membership_Length)
   Fitur ini akan menunjukkan loyalitas dari customer (apakah customer merupakan member lama atau bukan).
2. The Average Fare per Kilometer (AvgFare_perKM)
   Fitur ini akan menunjukkan pengguna mana yang dapat mendatangkan profit yang lebih besar.
3. Average discount rate (avg_discount)
   Fitur ini akan menunjukkan value dari pelanggan.

Code:

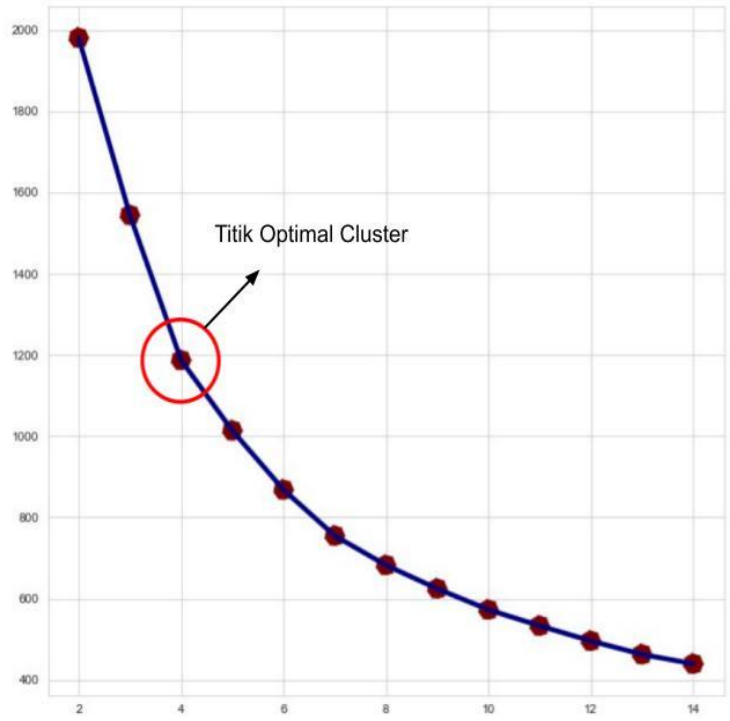
```
df2 = df[['Membership_Length','AvgFare_perKM','avg_discount']]
```

Total data untuk modeling: (51.316, 3)

Modeling

Pada tahap modeling ini, kami menggunakan algoritma K-means karena pada algoritma ini lebih efisien dan lebih cocok digunakan pada data yang besar dibandingkan menggunakan algoritma Agglomerative. Namun algoritma ini cukup sensitif terhadap outlier sehingga pada tahap data preprocessing, data-data yang memiliki outlier sudah dibuang. Pada algoritma ini juga memiliki sifat “non-deterministik” yang artinya inisialisasi centroidnya acak, sehingga ada kemungkinan menghasilkan cluster yang berbeda di setiap komputasinya. Disini kami mengalikannya dengan menambahkan parameter “random_state=0” pada algoritmanya sehingga inisialisasi centroidnya akan dilakukan dengan cara yang sama sehingga cluster yang dihasilkan akan tetap sama untuk seterusnya.

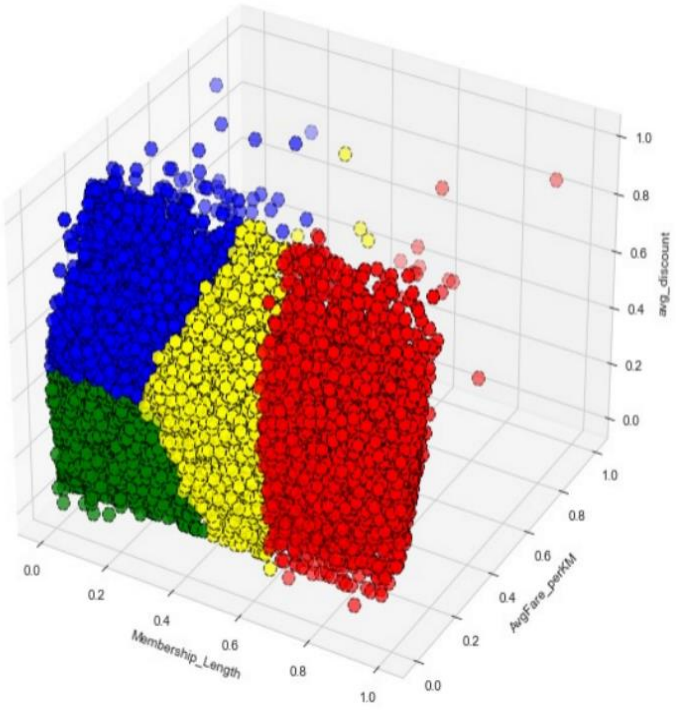
Langkah pertama, kami melakukan evaluasi K-means menggunakan Elbow Method untuk dapat menentukan berapa jumlah cluster yang akan digunakan. Setelah melakukan komputasinya, didapatkan grafik Elbow Method sebagai berikut:



Sumbu y menunjukkan besar inersianya dan sumbu x menunjukkan besar clusternya. Inertia adalah total jarak setiap data ke centroidnya yang mana semakin kecil inertia maka akan semakin bagus clusteringnya. Namun dampaknya itu sendiri adalah semakin banyaknya jumlah cluster. Maka dari itu perlu adanya penentuan jumlah cluster optimal yang dapat dilihat pada titik ketika menambahkan cluster tidak akan mengurangi inersia secara signifikan.

Dari grafik di atas dapat disimpulkan bahwa titik optimal jumlah cluster terletak pada cluster = 4 karena pada cluster selanjutnya total inertia mengalami penurunan yang sedikit (pada grafik tidak terlihat penurunan inersia yang sangat curam) dibandingkan dengan penurunan inertia pada cluster sebelumnya.

Setelah menentukan jumlah cluster yang akan digunakan, kita bisa lanjut ke tahap pengelompokan data berdasarkan clusternya. Pada feature selection, kami telah menetapkan 3 fitur yang kami rasa dapat memberikan informasi yang berharga dalam penentuan segmentasi customer penerbangan, yakni Membership_Length, AvgFare_perKM, dan avg_discount. Setelah clustering menggunakan algoritma K-Means, data tersebut divisualisasikan pada grafik 3 dimensi dengan sumbu x nya adalah Membership_Length, sumbu y adalah AvgFare_perKM, dan sumbu z adalah avg_discount. Diperoleh grafik seperti pada gambar dibawah ini:



Pada gambar grafik diatas terlihat 4 jenis cluster yang dapat dibedakan berdasarkan warnanya. Namun dari grafik tersebut masih belum jelas karakteristik setiap clusternya, sehingga kami melihat median dan rata-rata dari masing-masing cluster seperti pada tabel berikut:

Cluster	Membership_Length		AvgFare_perKM		avg_discount	
	mean	median	mean	median	mean	median
0	784.097665	746.0	0.763053	0.727623	0.817873	0.794670
1	2702.989010	2668.0	0.634205	0.624934	0.719469	0.716401
2	761.858020	713.0	0.486044	0.492998	0.559499	0.581882
3	1760.524932	1733.0	0.622492	0.612306	0.704104	0.703049

Dari tabel di atas, dapat ditarik kesimpulan bahwa terdapat 4 clusters customer penerbangan, yakni:

- **Cluster 0 (Warna Biru)**
  - *Membership_Length* rendah
    Maksud: Customer baru atau belum terlalu loyal
  - *AvgFare_perKM* paling tinggi
    Maksud: Customer potensial karena tidak terlalu bermasalah dengan harga penerbangan yang ditawarkan
  - *avg_discount* paling tinggi
    Maksud: Customer mendapatkan diskon yang tinggi karena memberikan revenue yang tinggi

  Conclusion: High Value Customer

- **Cluster 1 (Warna Merah)**
  - *Membership_Length* paling tinggi
    Maksud: Customer lama sehingga sudah loyal
  - *AvgFare_perKM* standar
    Maksud: Customer memberikan revenue yang cukup tinggi karena customer sering menggunakan penerbangan ini
  - *avg_discount* standar
    Maksud: Customer mendapatkan diskon yang cukup tinggi karena memberikan revenue yang cukup tinggi

  Conclusion: Loyal Customer

- **Cluster 2 (Warna Hijau)**
  - *Membership_Length* paling rendah
    Maksud: Customer baru sehingga belum loyal
  - *AvgFare_perKM* paling rendah
    Maksud: Customer cenderung memilih penerbangan jika harganya murah
  - *avg_discount* paling rendah
    Maksud: Customer mendapatkan diskon yang rendah karena memberikan revenue yang rendah

  Conclusion: Low Value Customer

- **Cluster 3 (Warna Kuning)**
  - *Membership_Length* rendah
    Maksud: Customer cukup loyal
  - *AvgFare_perKM* paling tinggi
    Maksud: Customer cenderung memilih penerbangan dengan harganya standar
  - *avg_discount* paling tinggi
    Maksud: Customer mendapatkan diskon yang cukup tinggi karena memberikan revenue yang cukup tinggi

  Conclusion: General Customer

Summary

Fitur yang kami gunakan pada Analisis Clustering kali ini ada 3 fitur yakni, **Length of Membership Time** (Membership_Length), **The Average Fare per Kilometer** (AvgFare_perKM), dan **Average discount rate** (avg_discount), karena ketiga fitur tersebut dapat memberikan informasi yang berharga dalam penentuan segmentasi customer penerbangan.

Model yang kami gunakan adalah algoritma **K-means** karena pada algoritma ini lebih cocok digunakan pada data yang besar serta lebih efisien dan juga memiliki sifat “Non-Deterministik” yang artinya inisialisasi centroidnya acak, sehingga ada kemungkinan kecil menghasilkan cluster yang berbeda di setiap komputasinya.

Berdasarkan evaluasi K-means dengan menggunakan **Elbow Method** dapat disimpulkan bahwa titik optimal jumlah cluster terletak pada **cluster = 4**, sehingga cluster dapat dibedakan menjadi:

1. **High Value Customer (Blue)**
   Customer ini pada umumnya pelanggan kelas atas yang biasanya memiliki kepentingan bisnis di penerbangan, sehingga customer tersebut perlu dipertahankan dan dikembangkan dengan diberikan strategi kebijakan preferensial (prioritas) yang relevan untuk meningkatkan jumlah perjalanan mereka (loyal).
2. **Loyal Customer (Red)**
   Customer yang sudah lama menjadi member dan ada kecenderungan akan hilang (tidak aktif), sehingga perlu untuk mengetahui informasi tentang customer ini, terus menjaga interaksi dengan pelanggan, dan memberikan strategi pemasaran tertentu seperti tindakan preferensial (prioritas) untuk pelanggan tersebut agar kembali aktif.
3. **Low Value Customer (Green)**
   Customer ini biasanya hanya akan menggunakan penerbangan sesekali saja, mungkin karena alasan promo yang menarik di penerbangan ini dan apabila ada promo yang lebih menarik di penerbangan yang lain maka customer ini cenderung akan menggunakan penerbangan yang lain. Customer seperti ini perlu dipertahankan dengan cara melakukan stimulasi untuk konsumsi sebanyak mungkin.
4. **General Customer (Yellow)**
   Customer dengan jumlah anggota terbanyak diantara cluster lainnya, tidak ada ciri-ciri khusus yang terlalu mencolok dari customer ini dengan cluster lainnya. Tetapi perlu dipertahankan dengan baik.