Customer Segmentation Behavior Using K Means and Principal Component Analysis

Sarah Safitri - 2440027511
Sharlene Regina - 2440032070
Tricia Estella - 2440003695

Table masukkan:

Bagian	Komentar	Perbaikan
Result, hasil visualisasi clustering	Centroid hasil clustering tidak ditunjukkan, sebaiknya ditunjukkan untuk dapat dianalisis visualisasinya	Centroid clustering ditunjukkan dengan lambang x, dengan mencari titik terdekat dengan centroid hasil PCA agar dapat divisualisasikan centroid pada titik asli sebelum dilakukan PCA
Result	Kurang dijelaskan analisis hasil berdasarkan visualisasinya	Dijelaskan analisis clustering berdasarkan hasil visualisasi pada result bagian perbandingan nilai K.

1. Introduction

1.1. Background of the study

Kartu kredit adalah bentuk pembayaran yang populer dan memungkinkan konsumen melakukan pembelian dan mengakses kredit dari lembaga keuangan. Kartu kredit dikeluarkan oleh bank dan lembaga keuangan lainnya, dan diterima oleh pedagang dan bisnis lain yang merupakan bagian dari jaringan kartu kredit. dengan adanya kartu kredit memungkinkan untuk melakukan transaksi di awal dibayarkan oleh bank dan diakhir membayar sesuai ketentuan yang telah diberikan oleh pihak bank.

Segmentasi pasar adalah strategi utama dalam pemasaran, karena memungkinkan bisnis menargetkan upaya pemasaran dan produk mereka secara lebih efektif ke kelompok konsumen tertentu. Ada berbagai pendekatan segmentasi pasar, seperti segmentasi demografis, yang membagi pasar berdasarkan karakteristik seperti usia, jenis kelamin, dan

pendapatan, dan segmentasi psikografis, yang membagi pasar berdasarkan gaya hidup, nilai, dan sikap. Segmentasi pasar membagi pasar yang lebih besar menjadi kelompok konsumen yang lebih kecil dan lebih homogen berdasarkan karakteristik atau perilaku yang sama. Hal ini memungkinkan pemasar untuk menargetkan upaya pemasaran mereka secara lebih efektif dan menyesuaikan produk dan layanan mereka dengan kebutuhan dan preferensi khusus dari setiap segmen [1].

Menurut data dari Bank Indonesia (2021) bahwa total volume transaksi kartu kredit pada bulan desember 2021 yang mencapai 27,857,966 dan merupakan angka tertinggi pada tahun 2021. Mencari segmentasi pasar dari pemakai kartu kredit bisa menjadi keuntungan dalam strategi marketing.

Data mining adalah suatu proses kegiatan menganalisa data guna untuk menemukan suatu pola dari sebuah kumpulan data [2]. Clustering adalah teknik yang digunakan dalam data mining untuk mengelompokkan objek data berdasarkan kemiripannya ke dalam cluster yang sama dan berbeda dengan objek pada cluster lain.

Menggunakan algoritma clustering untuk menganalisis data kartu kredit dapat membuat proses segmentasi pasar menjadi lebih efisien dan efektif. Pengelompokan memungkinkan identifikasi segmen pasar yang berbeda, dan dapat memberikan wawasan tentang perilaku dan preferensi konsumen yang mungkin tidak terlihat dari sumber data lain. Selain itu, pengelompokan dapat digunakan untuk melacak perubahan perilaku konsumen dari waktu ke waktu, memungkinkan bisnis menyesuaikan strategi pemasaran dan produk mereka untuk mengikuti perubahan kondisi pasar.

1.2. Problem definition

Meskipun penggunaan kartu kredit meluas dan potensi nilai data kartu kredit untuk segmentasi pasar, analisis manual terhadap data ini dapat memakan waktu dan sumber daya manusia yang banyak. Akibatnya, bisnis mungkin tidak dapat sepenuhnya memanfaatkan informasi yang terdapat dalam data kartu kredit untuk menargetkan upaya pemasaran mereka dan menyesuaikan produk mereka dengan segmen pasar tertentu.

1.3. Solution to the problem

analisis clustering merupakan salah satu metode data mining.
penggunaan algoritma clustering k-means untuk menganalisis data kartu kredit untuk segmentasi pasar, guna meningkatkan efisiensi dan efektivitas proses ini. Studi ini akan berusaha mengevaluasi keefektifan algoritma k-means untuk data kartu kredit, dan menilai potensi manfaat penggunaan k-means untuk segmentasi pasar.

2. Data

2.1. Dataset used

Dataset yang dipakai adalah dataset dari kaggle, dengan data sebanyak 9000 kartu kredit aktif selama 6 bulan terakhir. Data berfokus pada customer, dan terdapat 18 variabel:

- CUST_ID: Identifikasi pemegang kartu kredit (Kategorikal)
- BALANCE: Banyak saldo yang tersisa di akun untuk membuat transaksi
- BALANCE_FREQUENCY: Seberapa sering balance terupdate, score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- PURCHASES: Besar transaksi yang dilakukan suatu akun
- ONEOFF_PURCHASES: Maksimal pembelian dalam satu transaksi
- INSTALLMENTS_PURCHASES: Banyaknya pembelian dengan cicilan
- CASH_ADVANCE: Uang muka yang diberikan user
- PURCHASES_FREQUENCY: Seberapa sering pembelian dilakukan, score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- ONEOFF_PURCHASES_FREQUENCY: Seberapa sering pembelian dalam satu transaksi dilakukan, score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- PURCHASES_INSTALLMENTS_FREQUENCY: Seberapa sering pembelian dengan cicilan dilakukan. score terdiri dari 0 (not frequently updated) dan 1 (frequently updated)
- CASH_ADVANCE_FREQUENCY: Seberapa sering uang muka dibayarkan
- CASH_ADVANCE_TRX: Banyaknya transaksi dengan uang muka
- PURCHASES_TRX: Banyaknya transaksi pembelian dilakukan
- CREDIT_LIMIT: Limit dari credit card untuk user
- PAYMENTS: jumlah dari pembayaran yang telah dibayar user
- MINIMUM_PAYMENTS: Minimal pembayaran yang telah dibayar user
- PRC_FULL_PAYMENT: Persentase pembayaran penuh yang dibayarkan oleh user
- TENURE: Jangka waktu layanan kartu kredit untuk user

Pada project ini, kami menggunakan bahasa pemrograman python dan membutuhkan beberapa library seperti numpy, pandas, matplotlib, dan seaborn untuk memproses data tersebut dan mengembangkan model clustering. Adapun untuk file pendukung laporan ini (dataset, script, dan presentasi) dapat diakses di link drive berikut: https://drive.google.com/drive/folders/15en4lMsgaOdn5UitpXjAquZ_9oONN2Tt?usp=share_link

2.2 Preprocessing

a) Mengisi nilai NULL

Nilai NULL adalah penanda khusus yang digunakan dalam SQL untuk menunjukkan bahwa nilai data tidak ada dalam database. Dengan kata lain, ini hanyalah parameter untuk menunjukkan nilai yang hilang atau yang tidak kita

ketahui. Nilai NULL yang ditandakan dengan NaN, NA, Inf dapat saja dianggap ke dalam data string dan diperlakukan seperti bidang teks lainnya. Ini sangat bermasalah karena perintah tertentu seperti IS NULL atau LEN(0) tidak akan berfungsi lagi. Pada metode clustering, biasanya data yang memiliki nilai NULL akan diabaikan, sehingga perlu dilakukannya filling NULL value.

Ada beberapa cara untuk mengisi nilai NULL, antara lain:

b) Normalisasi data

Untuk meningkatkan akurasi pada model merupakan tujuan dari proses transformasi data. Proses transformasi data yang dilakukan adalah dengan normalisasi data sebelum dilakukannya proses clustering. Terdapat banyak metode yang dapat dilakukan untuk normalisasi data, misalnya standarisasi, min max, dan log transformation.

i. Standarisasi

Standarisasi merupakan proses yang mengubah distribution value pada nilai mean dan standard deviation menjadi 0 dan 1. Rumus dari Z-score normalization sebagai berikut:

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation } (x)}$$

Ada kalanya sebuah dataset memiliki data yang tidak berdistribusi normal (skew), sehingga pengambilan nilai mean dan standar deviasi tidak menjadi keputusan yang tepat. Hal ini dikarenakan karena mean tidak menggambarkan central tendency yang lebih baik dibandingkan dengan median/modus pada data skew.

ii. Min-Max Normalisasi

Min-Max Normalisasi atau dikenal dengan Min-Max Scaling. Rumus dari penggunaan Max-Min Scaling adalah sebagai berikut

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}.$$

Salah satu dari kelemahan penggunaan Min-Max Scaling adalah metode ini kurang cocok menangani outlier dengan baik. Jika terdapat outlier pada data yang digunakan, maka skala akan cenderung bernilai interval kecil, sehingga semua feature yang ada tidak akan mengalami perubahan skala yang berarti, sehingga tidak cocok untuk digunakan. Jika dibandingkan dengan standarisasi, standarisasi jauh lebih efektif dalam menangani outlier.

iii. Log Transformation

Log Transformation merupakan metode transformasi data yang mengubah setiap variabel x menjadi log(x). Interpretasi log transformation coefficient mempunyai rumus sebagai berikut:

$$Y = \log(X)$$

Adapun nilai X dapat diberi koefisien, misalnya seperti Y = log(1+X) untuk mengatasi nilai seperti log(0) jika nilai minimum pada variabel X adalah 0.

Disaat data continous yang digunakan tidak mengikuti bell curve, proses transformasi akan dilakukan untuk mengubah data menjadi senormal mungkin agar dapat meningkatkan keakuratan performa model yang terlihat pada hasil analisa. Syarat yang harus dipenuhi untuk melakukan log transformasi adalah data harus mengikuti/ mendekati distribusi log- normal. Bila tidak maka Log transformasi tidak dapat diberlakukan. Dilihat dari kondisi dataset yang ingin diolah, log transformation yang dirasa paling cocok di antara ketiga metode yang disebutkan sebelumnya.

3. Dataset Problem Solving: dataset exploratory

3.1 Informasi dataset

Kita dapat melihat informasi keseluruhan tiap variabel yang ada, yaitu jumlah data dan tipe datanya melalui function info().

Column	Non-Null Count	Dtype
CUST_ID	8950 non-null	object
BALANCE	8950 non-null	float64
BALANCE_FREQUENCY	8950 non-null	float64
PURCHASES	8950 non-null	float64
ONEOFF_PURCHASES	8950 non-null	float64
INSTALLMENTS_PURCHASES	8950 non-null	float64
CASH_ADVANCE	8950 non-null	float64
PURCHASE_FREQUENCY	8950 non-null	float64
ONEOFF_PURCHASES_FREQUENCY	8950 non-null	float64
PURCHASES_INSTALLMENT_F	8950 non-null	float64

REQUENCY		
CASH_ADVANCE_FREQUENCY	8950 non-null	float64
CASH_ADVANCE_TRX	8950 non-null	int64
PURCHASES_TRX	8950 non-null	int64
CREDIT_LIMIT	8949 non-null	float64
PAYMENTS	8950 non-null	float64
MINIMUM_PAYMENTS	8637 non-null	float64
PRC_FULL_PAYMENT	8950 non-null	float64
TENURE	8950 non-null	int64

Tipe data setiap feature mayoritas adalah float dan integer, dengan CUST_ID yang menjadi satu-satunya tipe data object. Menurut hasil tersebut, dapat kita lihat bahwa variabel CREDIT_LIMIT (8949) dan MINIMUM_PAYMENTS (8637) memiliki data dengan NaN, yang perlu kita isi nantinya. Hal tersebut dikarenakan count pada variabel tersebut lebih sedikit daripada baris dataset yang ada (8950).

Kita dapat melihat informasi seperti count, mean, standar deviasi, min, max, dan informasi lainnya menggunakan fungsi describe(), yang hanya berlaku untuk variabel numerik.

	count	mean	std	min	25%	50%	75%	max
BALANCE	8950.00	1564.47	2081.53	0.00	128.28	873.385	2054.14	19043.13
BALANCE_FREQUENCY	8950.00	0.87	0.24	0.00	0.89	1.00	1.00	1.0
PURCHASES	8950.00	1003.20	2136.63	0.00	39.64	361.28	1110.13	49039.57
ONEOFF_PURCHASES	8950.00	592.44	1659.88	0.00	0.00	38.00	577.40	40761025
INSTALLMENTS_PURCHASES	8950.00	411.06	904.34	0.00	0.00	89.00	468.64	22500.00
CASH_ADVANCE	8950.00	978.87	2097.16	0.00	0.00	0.00	1113.82	47137.21
PURCHASE_FREQUENCY	8950.00	0.49	0.40	0.00	0.083	0.50	0.92	1.00
ONEOFF_PURCHASES_FREQUENCY	8950.00	0.02	0.30	0.00	0.0	0.083	0.30	1.00

PURCHASES_INSTALLMENT_FREQUENCY	8950.00	0.364	0.40	0.00	0.0	0.167	0.75	1.00
CASH_ADVANCE_FREQUENCY	8950.00	0.13	0.20	0.00	0.0	0.00	0.22	1.50
CASH_ADVANCE_TRX	8950.00	3.25	6.82	0.00	0.0	0.00	4.00	123.00
PURCHASES_TRX	8950.00	14.71	24.86	0.00	1.0	7.00	17.00	358.00
CREDIT_LIMIT	8949.00	4494.45	3638.82	50.00	1600.0	3000.00	6500.00	30000.00
PAYMENTS	8950.00	1733.14	2895.06	0.00	383.27	856.901	1901.13	50721.48
MINIMUM_PAYMENTS	8637.00	864.21	2372.45	0.0192	169.12	312.34	825.48	76406.2001.00
PRC_FULL_PAYMENT	8950.00	0.15	0.29	0.00	0.0	0.00	0.14	1.00
TENURE	8950.00	11.52	1.34	6.00	12.0	12.00	12.00	12.00

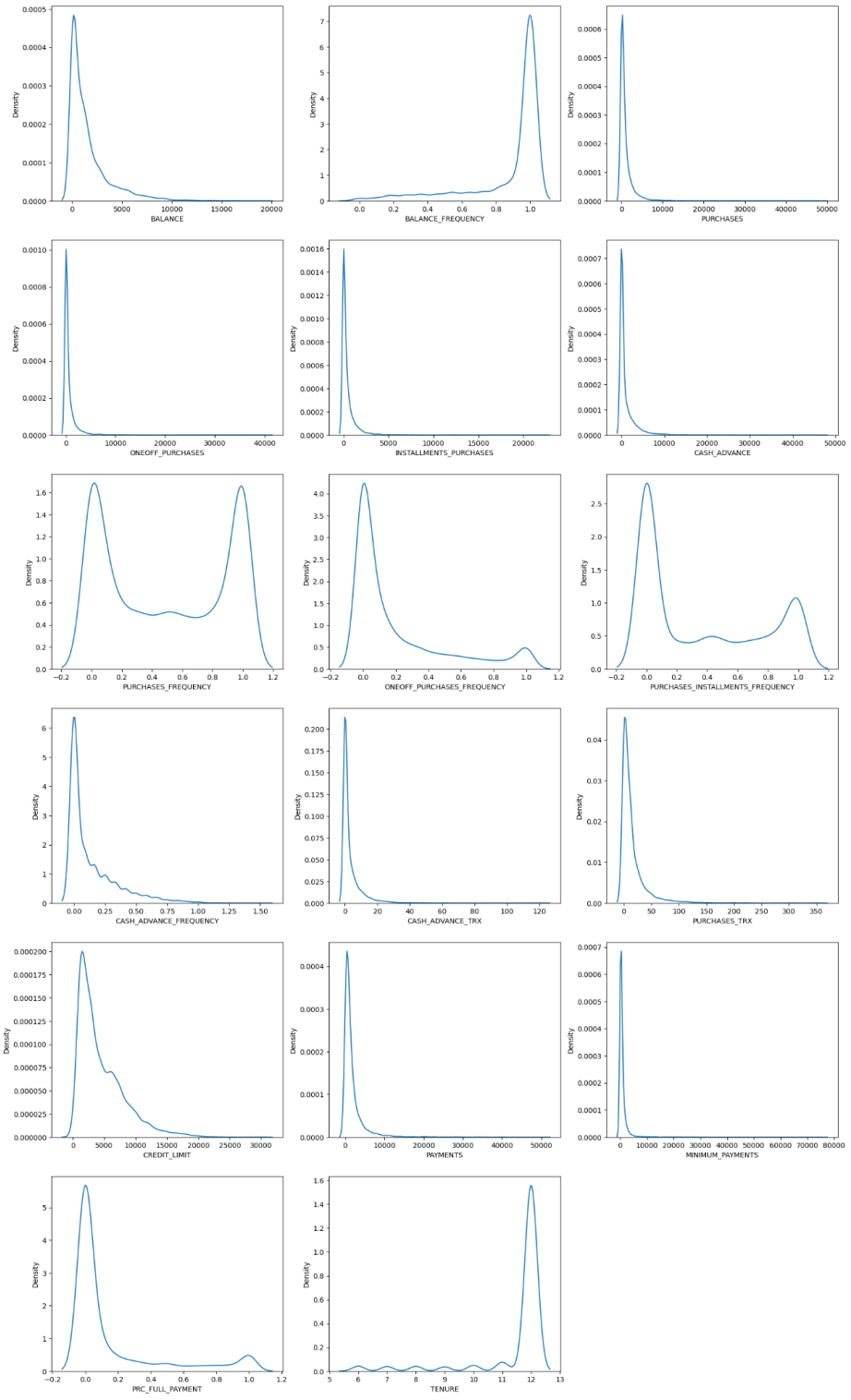
Hasil tersebut menunjukkan bahwa hampir semua feature memiliki mean yang lebih besar daripada nilai median, yang berarti terdapat _skewness_ pada dataset. Oleh karena itu, pada pre-processing data akan ada pengisian nilai null dan normalisasi pada dataset.

3.2. Distribution plot

Sebelum menentukan langkah normalisasi yang tepat, kita dapat memvisualisasikan distribusi setiap variabel terlebih dahulu.

```
plt.figure(figsize=(20,35))
for i, col in enumerate(df.columns):
    if df[col].dtype != 'object':
        ax = plt.subplot(6, 3, i+1)
        sb.kdeplot(df[col], ax=ax)
        plt.xlabel(col)

plt.show()
```

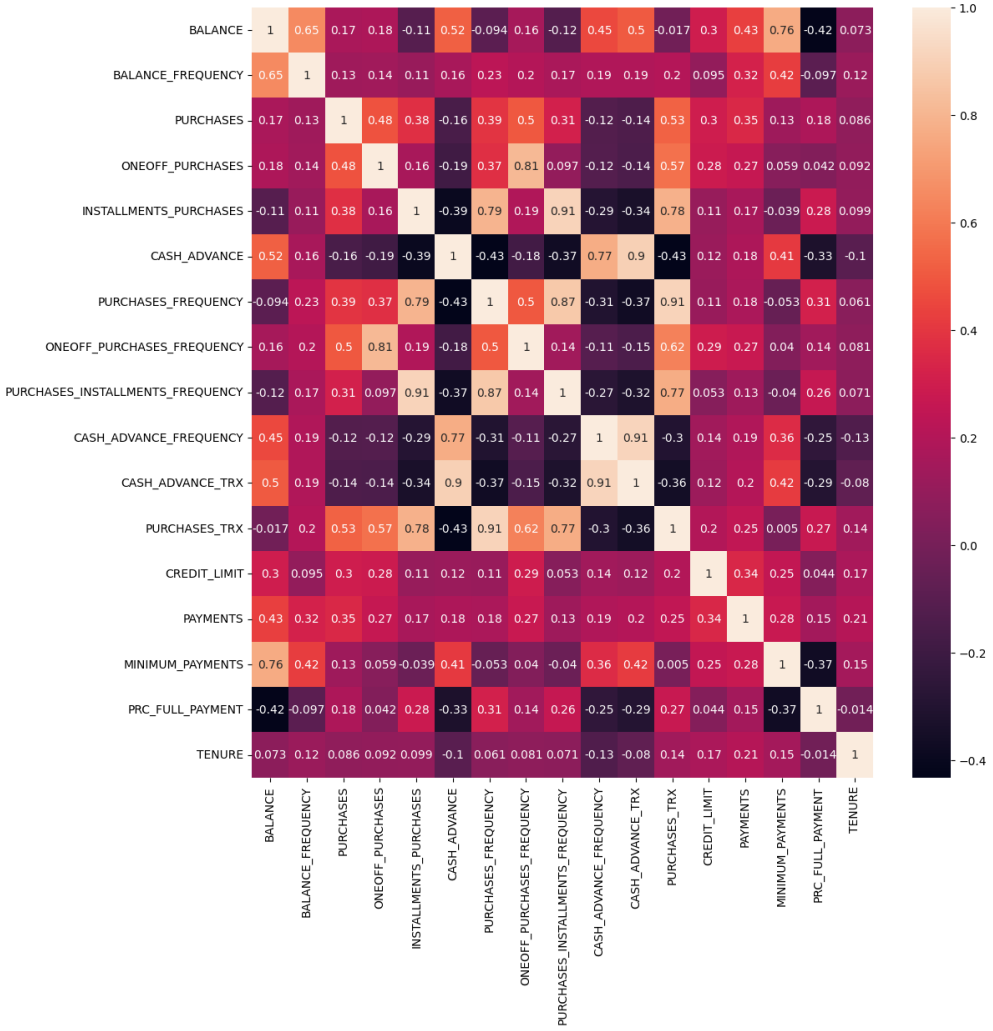


Dapat dilihat bahwa tiap variabel memiliki skewness yang bervariasi. Hal ini mungkin terjadi karena dalam dataset tersebut, behavior dari setiap customer pasti berbeda. Kemungkinan ada customer yang berbelanja dengan jumlah yang sangat besar, sehingga membuat data menjadi condong ke nilai yang besar (outliers). Untuk menghasilkan visualisasi cluster yang baik, kita dapat melakukan transformasi data menggunakan rumus logaritma (Logarithmic transformation).

3.3 Correlation Matrix

Terdapat 18 fitur pada dataset, yang bisa kita lakukan dimensionality reduction sehingga data akan memiliki dimensi yang lebih rendah. Sebelumnya, kita bisa melihat korelasi setiap feature menggunakan heatmap dan function corr().

```
plt.figure(figsize=(12,12))
sb.heatmap(df.corr(), annot=True)
plt.show()
```



Dapat dilihat bahwa cukup banyak feature yang berkorelasi, sehingga salah satu feature tersebut kurang lebih merepresentasikan informasi yang sama pada dataset. Oleh karena itu, dapat kita lakukan dimensionality reduction, dan kami akan menggunakan PCA.

4. Methods

4.1. K-Means

K-Means Merupakan salah satu algoritma clustering.
K-means adalah algoritma unsupervised learning yang digunakan untuk clustering. Ini mempartisi dataset menjadi sejumlah cluster (k) tertentu berdasarkan kesamaan titik data dalam sebuah cluster.

Algoritma k-means bekerja dengan terlebih dahulu menginisialisasi centroid cluster, yang merupakan titik di mana setiap cluster akan dibentuk. Kemudian, titik-titik data ditugaskan ke pusat cluster terdekat berdasarkan ukuran kesamaan, seperti jarak Euclidean. Setelah semua titik ditetapkan ke sebuah cluster, centroid diperbarui ke rata-rata titik dalam cluster mereka. Proses penetapan titik ke cluster dan pembaruan centroid ini diulang sampai cluster tidak lagi berubah atau jumlah iterasi maksimum tercapai. Algoritma k-means menggunakan jumlah cluster yang telah ditentukan sebelumnya, yang diwakili oleh variabel k. Angka ini perlu ditentukan oleh pengguna, dan dapat ditentukan melalui trial and error atau menggunakan heuristik seperti metode elbow [3].

4.2. Principal Component Analysis

Principal Component Analysis (PCA) adalah teknik pembelajaran unsupervised machine learning yang bekerja dengan mendapatkan fitur berdimensi rendah dari pengubahan fitur yang berdimensi lebih besar sambil tetap mempertahankan varians sebanyak mungkin. Metode PCA biasa digunakan untuk feature selection, menemukan hidden structure, mengurangi noise, dan mencegah curse of dimensionality dengan cara mereduksi dimensi data tinggi ke dimensi yang lebih rendah.

a. Pengurangan noise

   Standard measurement untuk menentukan noise adalah dengan signal-to-noise ratio (SNR) yang dihitung dengan rumus sebagai berikut

$$\mathrm{SNR} = \frac{\sigma^2_{\mathrm{signal}}}{\sigma^2_{\mathrm{noise}}}.$$

   Nilai SNR yang tinggi (bernilai lebih besar dari 1) menandakan bahwa data memiliki presisi yang tinggi. Semakin rendah nilai SNR yang didapatkan akan menandakan semakin banyaknya noise pada data.

b. Meminimalisir Redundancy

Principal Component Analysis bekerja dengan menggunakan *variance of feature*s dan *covariance of pairs of features* untuk mengelola covariance matrix, perhitungan dilakukan dengan rumus sebagai berikut

$$C_X = \frac{1}{m-1} X^T X.$$

C_x adalah matriks berukuran n × n yang menunjukan nilai fitur varians pada posisi diagonal dan nilai kovarians fitur pada posisi non-diagonal di matrix.

Algoritma PCA sebagai berikut:

- Standarisasi data original dari matrix X agar memiliki nilai zero mean,
- Menghitung covariance matrix C_x dari matrix X.
- Menghitung nilai eigen and the eigenvectors dari C_x.
- Mengurutkan nilai eigen dan eigenvectors dari C_x dengan urutan dari besar ke kecil (descending).
- Menentukan nilai k pertama dari eigenvectors, dan memasukan nilai tersebut sebagai nilai kolom matrix P.
- Menghitung reduced data matrix Y dengan rumus $Y = XP$.

5. Result and Analysis: the result of applied model

5.1. Preprocessing Result

a. Mengisi nilai NULL

Terdapat dua feature dengan nilai NULL, yaitu pada feature CREDIT_LIMIT dan MINIMUM_PAYMENTS.

Karena hanya ada satu data yang hilang pada CREDIT_LIMIT (8949 data terisi dari 8950 data yang ada, 0.01% data nilai NULL), maka kita bisa drop saja baris tersebut. Memungkinkan untuk mengisi data tersebut, tapi karena persentase nilai NULL sangat sedikit, maka pembuangan record ini tidak begitu berarti pada model yang akan dibuat.

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

Pada feature MINIMUM_PAYMENTS, karena persentase data NULL cukup besar, maka akan dilakukan pengisian data menggunakan nilai median. Hal ini dikarenakan tidak ada feature yang berelasi dengan feature ini untuk estimasi value yang hilang,

dan juga data tersebut memiliki _skewness_, sehingga lebih baik untuk mereplace NaN dengan median. Median akan memberikan estimasi central tendency yang lebih baik.

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

df.info()
```

Column	Non-Null Count	Dtype
CUST_ID	8949 non-null	object
BALANCE	8949 non-null	float64
BALANCE_FREQUENCY	8949 non-null	float64
PURCHASES	8949 non-null	float64
ONEOFF_PURCHASES	8949 non-null	float64
INSTALLMENTS_PURCHASES	8949 non-null	float64
CASH_ADVANCE	8949 non-null	float64
PURCHASE_FREQUENCY	8949 non-null	float64
ONEOFF_PURCHASES_FREQUENCY	8949 non-null	float64
PURCHASES_INSTALLMENT_FREQUENCY	8949 non-null	float64
CASH_ADVANCE_FREQUENCY	8949 non-null	float64
CASH_ADVANCE_TRX	8949 non-null	int64
PURCHASES_TRX	8949 non-null	int64
CREDIT_LIMIT	8949 non-null	float64
PAYMENTS	8949 non-null	float64
MINIMUM_PAYMENTS	8949 non-null	float64
PRC_FULL_PAYMENT	8949 non-null	float64
TENURE	8949 non-null	int64

Kita telah berhasil mengatasi missing values pada data, dimana record yang tersisa ada sebanyak 8949 dan setiap variabel memiliki jumlah data sebanyak record tersebut.

b. Normalisasi data

Sesuai dengan visualisasi plot distribusi data, dapat dilihat bahwa tiap variabel memiliki skewness yang bervariasi. Untuk menghasilkan visualisasi cluster yang baik, kita dapat melakukan transformasi data menggunakan rumus logaritma (Logarithmic transformation).

Ketika feature memiliki skewnes yang sangat besar, kurang baik jika kita menormalisasi dengan mean atau standar deviasi. Standarisasi menggunakan mean/std pada data tersebut akan mengubah value secara linear, dan tetap terjadi skewness pada data. Standarisasi tersebut lebih cocok untuk digunakan pada dataset dengan outliers yang sedikit, sehingga dapat mengikuti normal distribution dengan baik.

Begitu juga dengan min-max scaler, dimana kita hanya melakukan transformasi data sehingga data jatuh di angka 0 dan 1 untuk mempermudah perhitungan, dan tidak berpengaruh pada skewness.

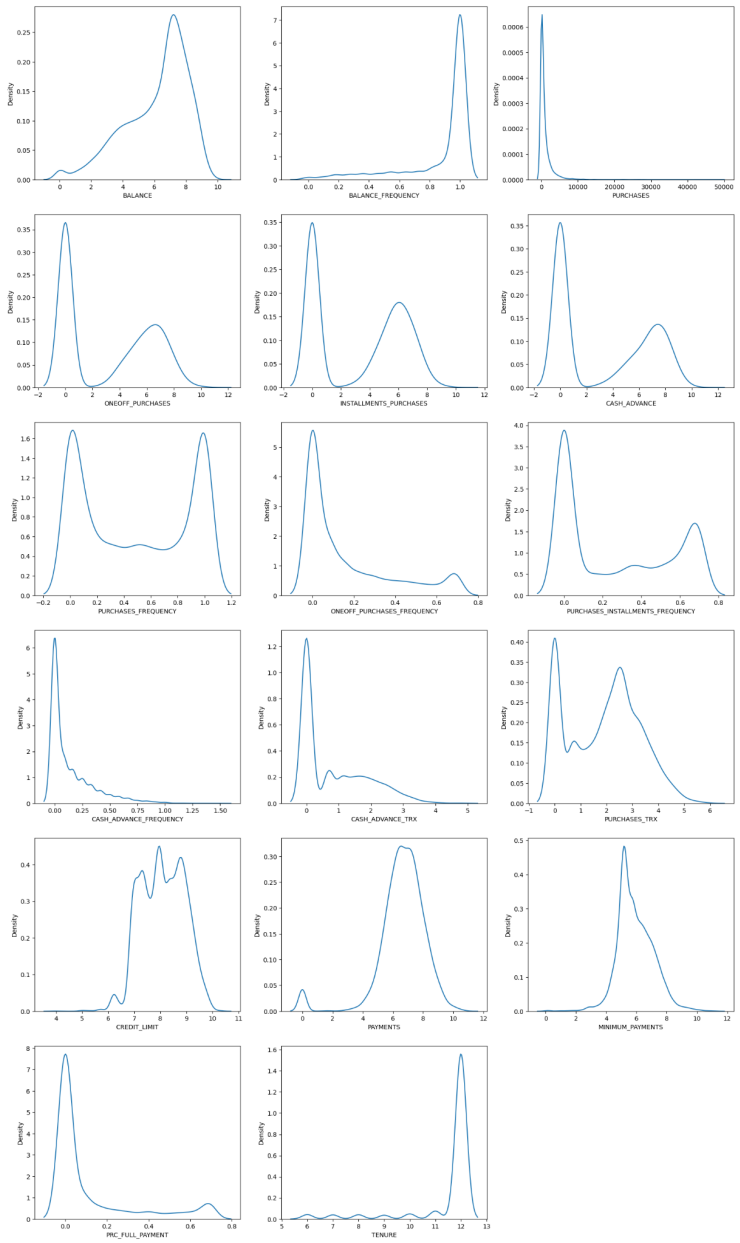
Oleh karena itu, untuk mengatasi skewness pada data, kami menggunakan logarithmic transformation. Transformasi ini cocok digunakan untuk data dengan heavily skewed feature dan diubah menjadi lebih dekat dengan normal distribution. Transformasi ini akan membuat semua baris (termasuk outliers) terlihat dengan jelas setelah dilakukan scaling. Hal ini akan membantu model untuk mengidentifikasi perbedaan satu data point dengan yang lainnya. Dengan kata lain, transformasi ini akan menyebar data sehingga model dapat membedakan cluster satu dengan cluster lain, dengan tetap mempertahankan value outliers. Analisis statistik pada data skew yang sudah mengikuti distribusi normal akan menjadi lebih akurat.

Adapun feature yang ditransformasi adalah BALANCE, ONEOFF_PURCHASES, INSTALLMENTS_PURCHASES, CASH_ADVANCE, ONE_OFF_PURCHASES_FREQUENCY, PURCHASES_INSTALLMENTS_FREQUENCY, CASH_ADVANCE_TRX, PURCHASES_TRX, CREDIT_LIMIT, PAYMENTS, MINIMUM_PAYMENTS, dan PRC_FULL_PAYMENT.

```
cols = ['BALANCE', 'ONEOFF_PURCHASES', 'INSTALLMENTS_PURCHASES',
'CASH_ADVANCE',
'ONEOFF_PURCHASES_FREQUENCY','PURCHASES_INSTALLMENTS_FREQUENCY',
'CASH_ADVANCE_TRX', 'PURCHASES_TRX', 'CREDIT_LIMIT', 'PAYMENTS',
'MINIMUM_PAYMENTS', 'PRC_FULL_PAYMENT']

for col in cols:
    df[col] = np.log(1 + df[col])
```

Kami menggunakan koefisien 1 dalam log untuk menghindari log(0), dikarenakan nilai minimum pada dataset adalah 0. Transformasi logaritma sudah selesai dan akan menghasilkan distribusi data sebagai berikut:



Walaupun hasil normalisasi terlihat tidak ideal, namun tetap lebih baik daripada data sebelum ditransformasi.

5.2 Dimensionality Reduction (PCA) Result

Secara singkat, PCA akan membuat axis baru untuk menjelaskan maximum variance pada dataset, dan menjadi principal component. Setelah itu, algoritma PCA akan memilih komponen lain yang tegak lurus dengan principal component dimensi pertama, untuk menjelaskan maximum variance nya.

Setelah kita mendapatkan principal components, maka kita dapat memilih berapa banyak komponen yang ingin kita ambil dan representasikan dalam data. Karena hanya mengambil principal component tersebut, maka dimensi data akan berkurang.

Pada kasus ini, kita akan mengambil n_components sebesar 95%, sehingga hasil data PCA akan merepresentasikan 95% varians dari data original sebelum direduksi menjadi satu fitur.

```
from sklearn.decomposition import PCA

pca = PCA(n_components=0.95)
reduced_df = pca.fit_transform(df)

pd.DataFrame(reduced_df)
```

	0
0	-907.918
1	-1003.322
2	-230.144
3	495.681
4	-987.318
…	…
8944	-712.198
8945	-703.318
8946	-858.919
8947	-1003.323
8948	89.933

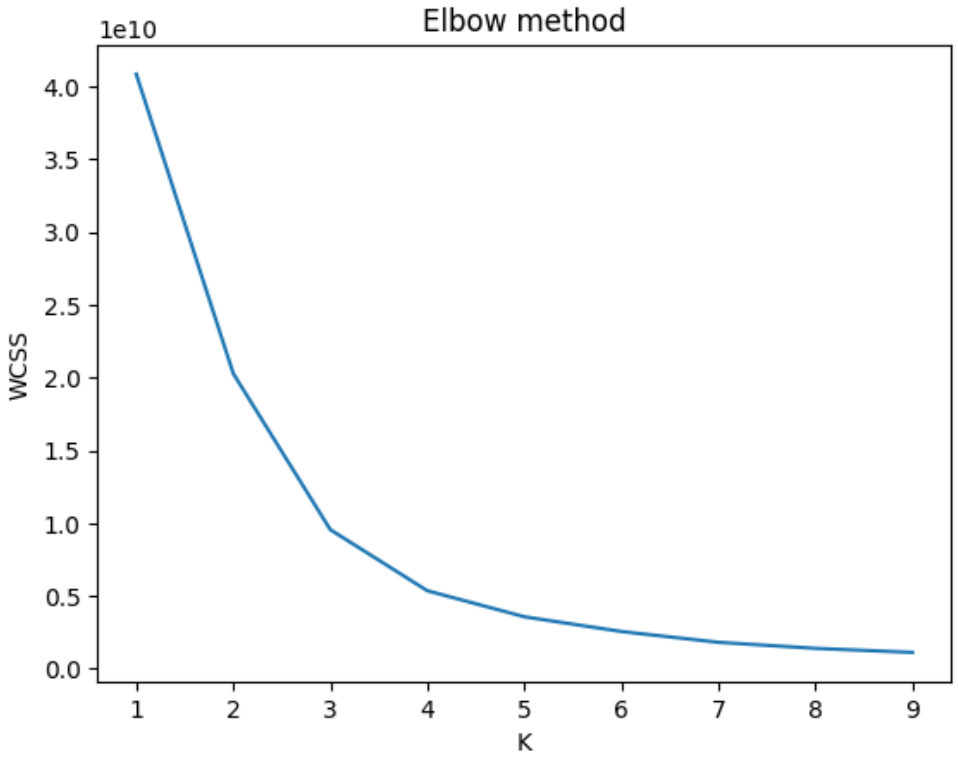
5.3 Clustering (elbow method) Result

K-Means Clustering adalah model unsupervised, kita memerlukan untuk menentukan nilai K terbaik, dengan menggunakan elbow method.

Clustering sendiri adalah meminimumkan jarak antara data point dan centroid cluster, serta memaksimumkan jarak antara centroid tiap cluster yang dihitung menggunakan WCSS (Within-Cluster Sum of Square). Elbow method akan memberi informasi nilai K mana yang

terbaik dengan menemukan WCSS, yaitu jumlah jarak kuadrat antara titik-titik dalam sebuah cluster dan centroid clusternya. Semakin tinggi nilai WCSS, maka cluster tersebut belum cukup baik, dimana ada dataset yang sangat jauh dengan nilai centroid clusternya. K terbaik akan dipilih, dimana nilai WCSS akan mulai turun dan membuat sebuah elbow.

Pada dataset ini, kami mendapatkan grafik elbow sebagai berikut:

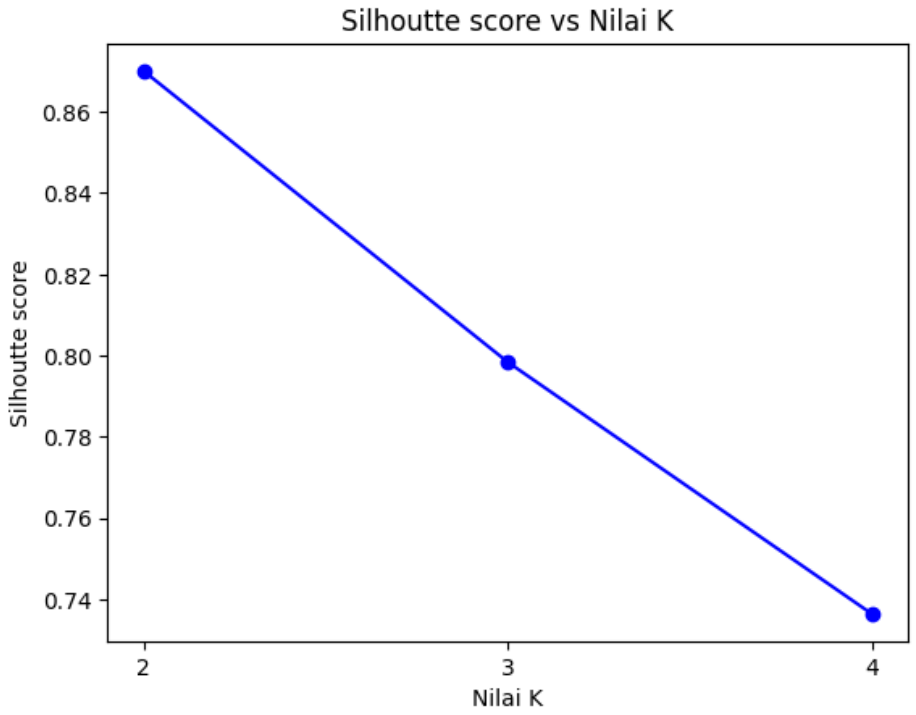


Elbow point yang dihasilkan tidak dapat ditentukan dengan jelas, antara K = 3 atau K = 4. Oleh karena itu, kita harus menguji kandidat nilai K tersebut menggunakan silhouette score. Rumus dari silhouette score adalah (b-a)/max(a,b); dimana a adalah rata-rata jarak diantara tiap poin dalam satu kluster, dan b adalah rata-rata jarak diantara semua cluster.

Silhouette score akan jatuh di nilai -1 sampai dengan 1, yang berarti:

- 1: Poin sudah berada pada cluster yang sesuai dan tiap cluster dapat dibedakan
- 0: Cluster overlapping
- -1: Poin berada pada cluster yang salah

Maka kita dapat melihat silhouette score diantara elbow, yaitu nilai 2-4. Untuk implementasinya, kita dapat menggunakan library silhoette_score dari sklearn.



Ternyata nilai silhouette score paling besar jatuh pada nilai K=2, dimana ketepatan poin berada pada cluster yang benar lebih tinggi dari yang lain. Hal ini menandakan bahwa hipotesis awal elbow point jatuh di titik 3 dan 4 salah, dan nilai K = 2 lebih baik untuk dataset ini.

Hasil dari elbow method ini menandakan bahwa kita akan membuat cluster pada dataset sebanyak 2 cluster.

6. Evaluation : evaluation of model performance

Hasil pembuatan model dari langkah-langkah yang diambil (preprocessing, dan elbow method) adalah clustering konsumen menjadi dua kelompok berdasarkan behavior mereka pada penggunaan kartu kreditnya. Pada bab ini kami akan membahas mengenai hasil clustering yang didapatkan, visualisasi clustering berdasarkan dua feature yang dipilih, dan juga perbandingan nilai K dengan yang lain.

6.1. Hasil model clustering dengan K = 2

Dengan nilai K = 2, kita dapat clustering dataset dan mendapatkan silhouette score sebesar 0.87. Kita dapat memvisualisasikan persebaran clustering dengan memilih dua feature untuk diwakilkan dalam sumbu x dan y pada plot.

Untuk kepentingan ini, kami memilih fitur ONEOFF_PURCHASES dan PURCHASES untuk memvisualisasikan hasil cluster dari dataset, dengan menggunakan scatter plot.

Adapun untuk menentukan centroid dengan nilai x dan y dari variabel ONEOFF_PURCHASES dengan PURCHASES berdasarkan hasil clustering PCA, maka kita dapat mencari indeks dari titik terdekat pada centroid reduced_df (dataframe PCA). Titik terdekat centroid ini untuk mewakilkan centroid sebenarnya yang didapatkan dari KMEANS dengan dataframe PCA, dikarenakan kita tidak dapat mengubah kembali nilai centroid PCA menjadi value ONEOFF_PURCHASES dan PURCHASES dengan tepat, sedangkan nilai centroid cluster tidak selalu ada pada cluster itu sendiri.

```
from sklearn.metrics import pairwise_distances_argmin_min

closest, _ = pairwise_distances_argmin_min(kmeans.cluster_centers_,
reduced_df)
centers = np.array([[df.iloc[i]['ONEOFF_PURCHASES'], df.iloc[i]["PURCHASES"]]
for i in closest])


plt.figure(figsize=(10,6))
sb.scatterplot(data=df, x='ONEOFF_PURCHASES', y='PURCHASES', hue='cluster_id')
plt.title('Distribution of clusters based on One off purchases and total
purchases')
plt.scatter(centers[:,0], centers[:,1], marker="x", color='black')
plt.show()
```

Dapat terlihat dua cluster, cluster 0 (biru) dan cluster 1 (oranye). Dapat kita analisa bahwa cluster 0 memiliki behavior pengeluaran lebih sedikit dari cluster 1. Melalui visualisasi tersebut, dapat dilihat kerapatan cluster 0 lebih padat dibandingkan dengan cluster 1. Hal itu juga berhubungan dengan jarak intracluster tiap data pada cluster 0 cenderung lebih kecil, sedangkan jarak intracluster 1 beragam dan lebih jauh. Selain itu, tidak terlihat adanya jarak signifikan pada tiap cluster yang berbeda (intercluster) pada skala ini. Centroid pada tiap cluster ditandai dengan x berwarna hitam. Dapat kita lihat jarak antar centroid sudah cukup jauh sehingga dapat menghasilkan cluster yang terpisah dengan cukup baik.

Perusahaan dapat mengambil informasi tersebut dan memberikan tindakan yang sesuai terhadap customer dalam satu cluster yang sama. Misalnya, perusahaan dapat menampilkan iklan promosi kepada cluster 0 sehingga mereka akan melakukan transaksi lebih _frequent_, sedangkan cluster 1 menjadi target pengiklanan utama saat adanya _launching_ produk baru dengan harga tinggi.

Contoh tersebut dapat berbeda, sesuai dengan kebutuhan perusahaan. Pada akhirnya, tugas kita untuk mengclustering customer berdasarkan behavior mereka dalam penggunaan kartu kredit sudah berhasil.

6.2 Perbandingan hasil clustering dengan nilai K yang lain

Pada pembuatan elbow method, sekilas cluster terbaik adalah 3. Oleh karena itu, kami ingin melihat hasil clustering jika K yang dipilih adalah 3.

```
kmeans = KMeans(n_clusters=3, random_state=23)
kmeans.fit(reduced_df)

df['cluster_id'] = kmeans.labels_
closest, _ = pairwise_distances_argmin_min(kmeans.cluster_centers_,
reduced_df)
centers = np.array([[df.iloc[i]['ONEOFF_PURCHASES'], df.iloc[i]["PURCHASES"]]
for i in closest])

plt.figure(figsize=(10,6))
sb.scatterplot(data=df, x='ONEOFF_PURCHASES', y='PURCHASES', hue='cluster_id')
plt.scatter(centers[:,0], centers[:,1], marker="x", color='r')
plt.title('Distribution of clusters based on One off purchases and total
purchases')
plt.show()
```

Hasil clustering bisa dibilang mirip, dimana cluster 0 tidak ada perubahan, namun cluster 1 pada K = 2 akan terbagi menjadi cluster 1 & 2 pada K = 3. Centroid setiap cluster ditandai dengan x merah. Sesuai hasil visualisasi, centroid cluster 0 dan cluster 1 cenderung berdekatan (lebih dekat dibandingkan hasil K = 2), sedangkan cluster 2 sangat jauh dibandingkan dengan kedua cluster lainnya. Jarak intracluster atau jarak tiap titik pada cluster 1 dan 2 juga lebih dekat dan rapat dibandingkan dengan cluster 3 yang saling berjauhan.

Berdasarkan hasil visualisasi scatter plot dan centroid kedua clustering (K = 2 dan K = 3), kami tetap lebih baik hasil clustering dengan K = 2 karena hasil analisis dan perbandingan sebagai berikut:

- Cluster 2 pada K = 3 sangat tersebar dan tidak seperti membentuk cluster. Pada dasarnya mereka memang lebih terpisah daripada cluster 1, namun setiap data cluster 2 akan berjauhan dengan centroidnya. Karena silhouette score dihitung berdasarkan rata-rata jarak semua titik dibanding rata-rata jarak pada satu cluster, clustering dengan nilai K = 2 akan lebih bagus karena kedua centroid dekat dengan *mayoritas* titik pada clusternya. Sedangkan jauhnya jarak setiap titik intercluster terhadap centroid (centroid diameter distance) pada cluster 2 dan K = 3 akan membuat rata-rata jarak pada satu cluster lebih tinggi, dan nilai silhouette score lebih rendah.
- Nilai WCSS pada elbow method menunjukkan K = 3 lebih baik karena centroid pada cluster 2 akan lebih dekat dengan titik terjauhnya, dibandingkan dengan K = 2 dimana centroid cluster 1 akan sangat jauh dengan titik terjauhnya. Namun kembali lagi pada poin sebelumnya, bahwa hasil centroid cluster 1 clustering K = 2 akan **lebih dekat dengan mayoritas** titik pada clusteringnya, sedangkan centroid cluster 3 clustering K

= 3 sangat **berjauhan dengan semua** titik pada clusternya, sehingga nilai evaluasi silhouette score lebih baik jika K = 2.

- Tidak terdapat pemisah signifikan/yang terlihat dengan jelas pada cluster 1 dan 2 pada K = 3.
- Jumlah data pada cluster 2 hasil KMEANS K = 3 sedikit, dan akan memakan biaya yang lebih dari perusahaan jika ingin membedakan keputusan terhadap cluster tersebut. Akan lebih baik jika cluster 2 dijadikan satu dengan yang lainnya seperti hasil clustering K = 3 sehingga perusahaan dapat mengefisiensi resourcenya.

Oleh karena itu, karena clustering adalah model *unsupervised learning* yang dapat dipilih nilai K-nya, kelompok kami beropini bahwa lebih baik dan efisien jika customer dikelompokkan menjadi dua saja.

7. Conclusion: conclusion, purpose of study

Berdasarkan dari penelitian yang dilakukan, berikut adalah point penting yang dapat disimpulkan:

1. Untuk mengatasi adanya multicollinearity pada feature dan variable, proses dimensionality reduction dengan PCA efektif meningkatkan akurasi dan presisi dari proses clustering. Berdasarkan dari hasil perbandingan proses K Mean clustering antara k-value bernilai 2 dan 3, model menunjukan hasil teroptimal pada k-value bernilai 2. Ini terlihat dari hasil silhouette score terbesar yang diperoleh yaitu sebesar 0.87, dan bukti visualisasi centroidnya.

2. Berdasarkan hasil customer behavior clustering dari jumlah pembayaran akun dan jumlah maksimal dalam satu transaksi, model berhasil mengelompokan customer kedalam 2 tipe, yaitu customer dengan total pembayaran kartu kredit sedikit (cluster_id = 0), dan customer dengan total pembayaran kartu kredit banyak (cluster_id = 1).

8. Implication: consequences, direct result, effect of finding in study to problems

1. Berdasarkan dari hasil K Means Clustering yang diberlakukan, jumlah titik data pada tiap kelompok cluster menandakan tingkat kepercayaan brand di mata publik. Semakin banyaknya jumlah customer dengan total pembayaran kartu kredit banyak (cluster_id = 1), maka semakin tinggi tingkat kepuasan customer terhadap produk yang dijual.

2. Hasil clustering juga dapat digunakan untuk membantu keputusan promosi yang tepat untuk setiap kelompok customer. Promosi yang ditargetkan untuk customer dengan total pembayaran kartu kredit sedikit bisa berfokuskan pada penawaran diskon, maupun product bundling. Sedangkan Promosi yang ditargetkan untuk customer dengan total pembayaran kartu kredit banyak dapat berupa penawaran additional service, produk varian terbaru, produk special edition, dan membership.

References

[1] TIRIS SUDRARTONO, "PENGARUH SEGMENTASI PASAR TERHADAP TINGKAT PENJUALAN PRODUK FASHION UMK," Coopetition : Jurnal Ilmiah Manajemen, vol. 10, no. 1. Institut Manajemen Koperasi Indonesia, pp. 53–64, Aug. 15, 2019. doi: 10.32670/coopetition.v10i1.40.

[2] Susy Rahmawati, Miftahul Nuril Silviyah, and Nur Syifa'ul Husna, "Implementation of Data Mining in Shopping Cart Analysis using the Apriori Algorithm," International Journal of Data Science, Engineering, and Analytics, vol. 1, no. 1. University of Pembangunan Nasional Veteran Jawa Timur, pp. 30–36, Jul. 18, 2021. doi: 10.33005/ijdasea.v1i1.5.

[3] T. Kansal, S. Bahuguna, V. Singh and T. Choudhury, "Customer Segmentation using K-means Clustering," 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), 2018, pp. 135-139, doi: 10.1109/CTEMS.2018.8769171.

[4] N. F. Jansson, R. L. Allen, G. Skogsmo, and S. Tavakoli, "Principal component analysis and K-means clustering as tools during exploration for Zn skarn deposits and industrial carbonates, Sala area, Sweden," *Journal of Geochemical Exploration*, vol. 233, p. 106909, 2022.

[5] E. Umargono, J. E. Suseno, and V. G. S. K., "K-means clustering optimization using the elbow method and early centroid determination based-on mean and Median," *Proceedings of the International Conferences on Information System and Technology*, 2019.

[6] V. Divya and K. N. Devi, "An Efficient Approach to Determine Number of Clusters Using Principal Component Analysis," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018, pp. 1-6, doi: 10.1109/ICCTCT.2018.8551182.

[7] K. Tabianan, S. Velu, and V. Ravi, "K-means clustering approach for intelligent customer segmentation using Customer Purchase Behavior Data," *Sustainability*, vol. 14, no. 12, p. 7243, 2022.

[8] J. Chen, L. Zhao, M. Zhou, Y. Liu, and S. Qin, "An approach to determine the optimal K-value of k-means clustering in adaptive random testing," *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, 2020.

[9] M. Zubair, M. D. A. Iqbal, A. Shil, M. J. Chowdhury, M. A. Moni, and I. H. Sarker, " K-means clustering algorithm towards an efficient data-driven modeling," *Annals of Data Science*, 2022.

[10] K. Rakhmanaliyeva, "Identifying customer buying patterns using market basket analysis," *Herald of the Kazakh-British technical university*, vol. 18, no. 3, pp. 95–101, 2021.