Nama: Mochamad Arief Dermawan

NIM: 110320128

SOAL ANALISA CLUSTERING

1. Jika algoritma K-Means menghasilkan nilai silhouette score rendah (0.3) meskipun elbow method menunjukkan K=5 sebagai optimal pada dataset ini, faktor apa yang menyebabkan inkonsistensi ini? Bagaimana strategi validasi alternatif (misal: analisis gap statistic atau validasi stabilitas cluster via bootstrapping) dapat mengatasi masalah ini, dan mengapa distribusi data non-spherical menjadi akar masalahnya?

2. Dalam dataset dengan campuran fitur numerik (Quantity, UnitPrice) dan kategorikal high-cardinality (Description), metode preprocessing apa yang efektif untuk menyelaraskan skala dan merepresentasikan fitur teks sebelum clustering? Jelaskan risiko menggunakan One-Hot Encoding untuk Description, dan mengapa teknik seperti TF-IDF atau embedding berdimensi rendah (UMAP) lebih robust untuk mempertahankan struktur cluster!

3. Hasil clustering dengan DBSCAN sangat sensitif terhadap parameter epsilon—bagaimana menentukan nilai optimal epsilon secara adaptif untuk memisahkan cluster padat dari noise pada data transaksi yang tidak seimbang (misal: 90% pelanggan dari UK)? Jelaskan peran k-distance graph dan kuartil ke-3 dalam automasi parameter, serta mengapa MinPts harus disesuaikan berdasarkan kerapatan regional!

4. Jika analisis post-clustering mengungkapkan overlap signifikan antara cluster "high-value customers" dan "bulk buyers" berdasarkan total pengeluaran, bagaimana teknik semi-supervised (contoh: constrained clustering) atau integrasi metric learning (Mahalanobis distance) dapat memperbaiki pemisahan cluster? Jelaskan tantangan dalam mempertahankan interpretabilitas bisnis saat menggunakan pendekatan non-Euclidean!

5. Bagaimana merancang temporal features dari InvoiceDate (misal: hari dalam seminggu, jam pembelian) untuk mengidentifikasi pola pembelian periodik (seperti transaksi pagi vs. malam)? Jelaskan risiko data leakage jika menggunakan agregasi temporal (misal: rata-rata pembelian bulanan) tanpa time-based cross-validation, dan mengapa lag features (pembelian 7 hari sebelumnya) dapat memperkenalkan noise pada cluster!

Jawab

1. **Penyebab:**

- **Distribusi data tidak sferikal**: K-Means mengasumsikan cluster berbentuk bulat dan berukuran sama. Jika data memiliki bentuk memanjang, tidak terpusat, atau noise tinggi, maka pemisahan jadi tidak optimal.
- **Overlapping cluster**: Elbow method hanya melihat penurunan WCSS (Within-Cluster Sum of Squares), bukan kualitas pemisahan.
- **Outlier dan noise** bisa mempengaruhi centroid dan mengaburkan pemisahan.

Solusi Alternatif:

- **Gap Statistic**: Bandingkan WCSS dari data asli dengan data acak, membantu mendeteksi *true K* meskipun cluster bentuknya kompleks.
- **Bootstrapping + stability validation**: Ulangi clustering pada subset data → ukur stabilitas label untuk tiap titik → validasi apakah cluster konsisten.

2. **Langkah-langkah efektif:**

- **Numerik**: Gunakan *StandardScaler* atau *MinMaxScaler* agar fitur seperti Quantity dan UnitPrice berada dalam skala setara.
- **Kategorikal**:
    - **Hindari One-Hot Encoding** untuk Description karena bisa menghasilkan ribuan kolom → menyebabkan curse of dimensionality.
    - Gunakan:
        - **TF-IDF Vectorization**: Menangkap pentingnya istilah dalam konteks global.
        - **UMAP/TruncatedSVD**: Untuk reduksi dimensi vektor teks ke representasi numerik padat.
        - **Word2Vec atau SentenceBERT**: Untuk representasi semantik yang lebih kaya.

3. **Masalah:**

- DBSCAN sensitif terhadap eps dan MinPts. Salah satu → hasil buruk atau semua dianggap noise.

Strategi adaptif:

- **K-distance graph**:
    - Hitung jarak ke-k tetangga terdekat untuk tiap titik.
    - Plot nilai jarak tersebut → titik "elbow" ≈ nilai optimal eps.
- **Kuartil ke-3 dari jarak** bisa digunakan sebagai threshold otomatis → menjaga outlier tetap dianggap noise.
- **MinPts disesuaikan** dengan densitas lokal:
    - Daerah padat → MinPts bisa lebih tinggi.
    - Daerah jarang → MinPts lebih rendah

4. Fitur Total Spending memicu overlap → sulit bedakan dua segmentasi penting secara bisnis.

Solusi:

- **Constrained Clustering**:
    - Tambahkan *must-link* dan *cannot-link constraints* → pandu clustering agar sesuai insight bisnis.
- **Metric Learning (e.g., Mahalanobis distance)**:
    - Belajar matriks jarak berbobot → fitur seperti frekuensi belanja bisa lebih ditekankan daripada total spending.

- Mahalanobis: memperhitungkan korelasi antar fitur → lebih sensitif pada outlier dan distribusi sejati.

5. **Desain fitur temporal:**

- Hari dalam minggu (weekday), jam pembelian (morning vs. night), musim → bantu deteksi pola periodik.
- **Cyclical encoding** untuk hari dan jam (misal: $\sin(2\pi \text{day}/7)$) → karena waktu bersifat siklus.
- Lag features (pembelian 7 hari sebelumnya, rolling mean) → tangkap *momentum* pembelian.

Risiko data leakage:

- Menggunakan **rata-rata bulanan atau rolling mean** bisa bocor ke masa depan jika tidak hati-hati.
- Harus **gunakan time-based cross-validation (TimeSeriesSplit)** → validasi tetap menghormati urutan waktu.