**Kinanti Aria Widaswara**

**1103213165**

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?**

   Ketidaksesuaian antara nilai silhouette score yang rendah (misalnya 0.3) dan hasil elbow method yang menunjukkan K=5 sebagai jumlah klaster optimal biasanya disebabkan oleh asumsi dasar KMeans yang tidak terpenuhi. Elbow method hanya melihat total penurunan inertia (SSE) dan tidak memperhitungkan seberapa baik klaster terpisah satu sama lain. Silhouette score, di sisi lain, mempertimbangkan kedekatan titik terhadap klasternya dibanding klaster lain, sehingga memberikan indikasi kualitas klaster secara struktural. Jika silhouette rendah, itu artinya klaster saling tumpang tindih atau tidak kompak, meskipun inertia terlihat menurun. Penyebab utamanya adalah distribusi data yang non-spherical, karena KMeans mengasumsikan bentuk klaster bulat dan seimbang. Untuk mengatasi ini, validasi alternatif seperti gap statistic dapat membandingkan struktur klaster terhadap data acak, dan validasi stabilitas via bootstrapping dapat memastikan apakah hasil klaster tetap konsisten pada data sampling ulang. Strategi ini lebih robust terhadap bentuk dan distribusi data yang kompleks.

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!**

   Untuk dataset yang berisi kombinasi fitur numerik seperti Quantity dan UnitPrice, serta fitur kategorikal dengan banyak nilai unik seperti Description, preprocessing harus dirancang secara hati-hati. Fitur numerik sebaiknya diskalakan menggunakan StandardScaler atau RobustScaler agar tidak mendominasi jarak antar titik. Fitur teks Description, jika diubah ke One-Hot Encoding, akan menghasilkan dimensi sangat besar dan sparse yang merusak struktur jarak Euclidean dalam clustering. Oleh karena itu, pendekatan yang lebih efektif adalah menggunakan representasi teks berbasis TF-IDF, yang memberi

bobot pada kata berdasarkan frekuensi relatifnya. Selanjutnya, dimensi dari TF-IDF dapat direduksi menggunakan teknik seperti UMAP atau TruncatedSVD untuk menghasilkan representasi padat berdimensi rendah. Metode ini lebih robust dalam mempertahankan struktur klaster dan jauh lebih efisien untuk algoritma clustering.

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!**

   DBSCAN sangat bergantung pada parameter epsilon yang mengontrol radius neighborhood untuk membentuk cluster. Jika epsilon terlalu kecil, sebagian besar titik akan dianggap noise, sedangkan jika terlalu besar, klaster menjadi kabur. Dalam data yang tidak seimbang seperti transaksi dengan 90% pelanggan dari satu negara, pemilihan epsilon secara adaptif menjadi penting. Pendekatan umum adalah menggunakan k-distance graph, yaitu grafik yang menunjukkan jarak ke tetangga ke-k terdekat untuk semua titik. Titik "elbow" dari grafik ini menunjukkan perubahan signifikan dalam jarak dan menjadi kandidat ideal untuk nilai epsilon. Secara statistik, nilai kuartil ke-3 dari distribusi jarak antar titik juga bisa dijadikan patokan awal. Selain itu, parameter MinPts harus disesuaikan dengan kerapatan lokal—area dengan konsentrasi tinggi membutuhkan MinPts lebih besar agar tidak overcluster, sementara area sepi perlu toleransi lebih rendah agar tidak dianggap noise secara keseluruhan.

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!**

   Ketika dua segmen seperti high-value customers dan bulk buyers menunjukkan overlap signifikan berdasarkan total pengeluaran, itu berarti satu fitur saja tidak cukup untuk memisahkan perilaku mereka. Teknik semi-supervised seperti constrained clustering dapat digunakan untuk memasukkan pengetahuan domain dalam bentuk must-link atau cannot-link constraint, yang

memaksa atau melarang dua data berada di klaster yang sama. Alternatif lain adalah metric learning, seperti penggunaan Mahalanobis distance, untuk mempelajari bobot antar fitur sehingga fitur pembeda menjadi lebih berpengaruh dalam perhitungan jarak. Namun, pendekatan ini memiliki tantangan dalam interpretabilitas. Model berbasis jarak kompleks dan transformasi ruang tidak selalu mudah dijelaskan ke stakeholder bisnis yang lebih menyukai insight sederhana seperti "klaster A adalah pelanggan UK dengan pembelian tinggi".

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!**

   Fitur temporal dari InvoiceDate dapat memberikan insight berharga tentang pola pembelian periodik. Misalnya, ekstraksi hari dalam minggu atau jam pembelian dapat mengungkap kebiasaan pelanggan seperti belanja di pagi hari atau saat akhir pekan. Namun, perlu hati-hati dalam menggunakan agregasi waktu seperti rata-rata pembelian bulanan. Jika dilakukan sebelum split data, agregasi ini bisa menyebabkan data leakage, di mana informasi masa depan bocor ke model pelatihan. Untuk menghindari ini, disarankan menggunakan time-based cross-validation seperti forward chaining. Selain itu, penggunaan lag features (contoh: total pembelian 7 hari sebelumnya) dapat menambah noise jika transaksi pelanggan tidak terjadi secara reguler. Fitur-fitur tersebut hanya berguna jika data memiliki kepadatan dan kontinuitas waktu yang cukup konsisten.