Nama : Chandra Aulia Haswangga

NIM : 1103223163

Kelas : TK-45-G05

Tugas : UTS – Analisis Clustering

1. Inkonsistensi Antara Silhouette Score dan Elbow Method pada K-Means
   a) Faktor Penyebab Inkonsistensi
      - Distribusi Data Non-Spherical
        K-Means mengasumsikan bahwa cluster berbentuk bulat (spherical) dengan kepadatan yang relatif merata. Jika data memiliki distribusi yang tidak berbentuk bulat, misalnya cluster berbentuk memanjang atau tidak teratur, K-Means tidak akan berhasil memisahkannya dengan baik, meskipun elbow method menunjukkan bahwa K=5 adalah jumlah cluster yang optimal. Silhouette score rendah (0.3) menunjukkan bahwa sebagian besar titik data tidak cocok dengan cluster yang mereka masuki, mengindikasikan bahwa distribusi data tidak sesuai dengan asumsi K-Means.

   b) Strategi Validasi Alternatif
      - Gap Statistic
        Gap statistic dapat membantu mengidentifikasi jumlah cluster yang optimal dengan membandingkan kinerja model clustering pada data aktual dengan data acak yang disimulasikan. Ini memberikan ukuran yang lebih objektif untuk jumlah cluster optimal dibandingkan elbow method.

      - Bootstrapping dan Validasi Stabilitas Cluster
        Bootstrapping dapat digunakan untuk mengevaluasi kestabilan cluster. Ini melibatkan pengambilan sampel ulang data dan melihat seberapa stabil hasil clustering dari satu sampel ke sampel lainnya.

   c) Mengatasi Distribusi Data Non-Spherical
      - Jika data memiliki distribusi non-spherical, menggunakan algoritma yang lebih fleksibel seperti DBSCAN (Density-Based Spatial Clustering of Applications with Noise) atau Gaussian Mixture Models (GMM) yang tidak mengasumsikan bentuk cluster yang bulat akan lebih efektif. DBSCAN, misalnya, dapat menangani bentuk cluster yang lebih kompleks dan juga dapat mengidentifikasi noise (outliers).

2. Metode Preprocessing untuk Fitur Numerik dan Kategorikal High-Cardinality dalam Clustering
   a) Metode Preprocessing
      - Numerik

Fitur numerik seperti Quantity dan UnitPrice harus dinormalisasi agar skala mereka tidak memengaruhi clustering. Metode StandardScaler atau Min-Max Scaler bisa digunakan untuk menyelaraskan skala antar fitur numerik.

- Kategorikal High-Cardinality
  Fitur kategorikal seperti Description dengan 1000 nilai unik dapat membuat model menjadi sangat besar jika menggunakan One-Hot Encoding. Ini bisa menyebabkan masalah dalam memori dan sparsitas yang tinggi.

b) Risiko Menggunakan One-Hot Encoding
   - Dimensi yang sangat besar
     One-Hot Encoding akan menghasilkan ribuan kolom untuk fitur kategorikal dengan banyak kategori. Ini dapat mengarah pada high dimensionality yang bisa mengurangi efisiensi model dan bahkan meningkatkan risiko overfitting.

   - Sparse Matrix
     Penggunaan one-hot encoding pada fitur dengan banyak kategori dapat menghasilkan matrix sparse yang membuat model lebih sulit untuk dikelola dan lebih memakan memori.

c) Alternatif Encoding yang Lebih Robust
   - TF-IDF (Term Frequency-Inverse Document Frequency)
     Untuk fitur teks, TF-IDF lebih efektif daripada One-Hot Encoding karena dapat menimbang kata-kata yang lebih relevan dalam Description dan mengurangi bobot kata-kata umum yang tidak memberikan banyak informasi untuk clustering.

   - Embedding Dimensionality Reduction (seperti UMAP)
     Teknik embedding atau dimensionality reduction seperti UMAP dapat mengubah fitur kategorikal menjadi vektor berdimensi rendah yang lebih mudah dikelola oleh model, sekaligus mempertahankan struktur penting dalam data.

3. Menentukan Nilai Optimal Epsilon dalam DBSCAN untuk Data Transaksi Tidak Seimbang
   a) Menggunakan k-Distance Graph
      - K-Distance Graph
        Untuk menentukan nilai epsilon secara adaptif, k-distance graph digunakan untuk mengevaluasi jarak antar titik dalam cluster. Dengan k=4 atau k=5, kita dapat melihat di mana terdapat "celah" yang signifikan dalam jarak antar titik. Epsilon dapat dipilih pada nilai jarak k-distance yang menunjukkan perbedaan tajam antara cluster padat dan noise.

- Kuartil ke-3
  Nilai epsilon seringkali dipilih berdasarkan kuartil ke-3 dari k-distance graph, di mana jarak antar titik meningkat secara signifikan, menunjukkan transisi antara cluster dan noise.

b) Penyesuaian MinPts Berdasarkan Kerapatan Regional
   - MinPts adalah jumlah minimum titik yang diperlukan untuk membentuk cluster. Untuk data transaksi yang tidak seimbang (misalnya 90% pelanggan dari UK), MinPts perlu disesuaikan untuk masing-masing area regional. Misalnya, jika satu region memiliki kepadatan lebih tinggi, MinPts di sana bisa lebih rendah dibandingkan dengan region yang lebih jarang.

4. Mengatasi Overlap Cluster dengan Semi-Supervised atau Metric Learning
   a) Semi-Supervised Clustering
      - Constrained Clustering
        Dengan menggunakan pendekatan semi-supervised, kita bisa menambahkan constrained clustering, seperti must-link (data harus ada dalam cluster yang sama) atau cannot-link (data tidak boleh berada dalam cluster yang sama), untuk membedakan antara "high-value customers" dan "bulk buyers". Ini memberikan model informasi tambahan yang membatasi kemungkinan overlap.

   b) Integrasi Metric Learning
      - Mahalanobis Distance
        Metric learning, seperti penggunaan Mahalanobis distance, dapat membantu dalam memperbaiki pemisahan cluster dengan memberikan model jarak yang lebih sensitif terhadap skala dan korelasi antar fitur. Ini membantu membedakan cluster yang sangat mirip dan memperbaiki pemisahan antar kelompok yang overlaping.

   c) Tantangan dalam Interpretabilitas Bisnis
      - Non-Euclidean Distance
        Menggunakan Mahalanobis distance atau pendekatan non-Euclidean lainnya dapat mengurangi interpretabilitas bisnis karena jarak antar titik tidak lagi dihitung dengan cara yang intuitif (misalnya, Euclidean distance). Ini bisa membuatnya sulit untuk menjelaskan keputusan model dalam bahasa bisnis yang sederhana.

5. Desain Temporal Features dan Risiko Data Leakage
   a) Merancang Temporal Features
      - Hari dalam Seminggu dan Jam Pembelian
        Dari InvoiceDate, kita bisa mengekstrak hari dalam seminggu (misalnya, Senin, Selasa, dll) dan jam pembelian (misalnya, pagi atau malam) untuk mengidentifikasi pola pembelian periodik. Ini dapat membantu memahami

kapan pelanggan lebih cenderung melakukan pembelian, seperti transaksi lebih banyak pada waktu tertentu.

b) Risiko Data Leakage
   - Jika agregasi temporal seperti rata-rata pembelian bulanan digunakan tanpa mempertimbangkan pembagian waktu yang tepat, ini dapat menyebabkan data leakage. Model akan belajar pola dari masa depan yang seharusnya tidak tersedia selama pelatihan. Contohnya, jika pembelian bulanan dihitung sebelum data dibagi menjadi training dan testing, data dari bulan yang lebih baru bisa "terlihat" oleh model dalam training.

c) Menghindari Data Leakage dengan Time-Based Cross-Validation
   - Gunakan time-based cross-validation untuk memastikan bahwa model tidak belajar dari data yang akan datang. Pastikan untuk membagi data dengan mempertimbangkan urutan waktu (misalnya, training pada bulan pertama hingga bulan ke-6, testing pada bulan ke-7).

d) Risiko Lag Features
   - Lag features (misalnya, pembelian 7 hari sebelumnya) dapat memperkenalkan noise dalam clustering, karena memberikan informasi yang terlalu mendalam yang mungkin tidak relevan dengan pola pembelian saat ini. Lag features juga bisa mengarah pada overfitting jika terlalu banyak menggunakan data masa lalu tanpa mempertimbangkan dinamika yang berubah.