Nama : Izzan Muhammad Fa'iz (1103210126)

Kelas : TK4505

UTS II Klasifikasi (KlasifikasiUTS)

1. Ketika model machine learning menunjukkan AUC-ROC tinggi (misalnya 0.92) tetapi memiliki presisi sangat rendah (misalnya 15%), hal ini menandakan bahwa model secara keseluruhan mampu memisahkan kelas positif dan negatif, tetapi threshold klasifikasi yang digunakan tidak optimal. AUC-ROC menghitung area di bawah kurva dari semua threshold, sehingga model dapat tampak "baik" secara agregat meskipun presisi atau recall di threshold tertentu buruk. Faktor penyebab utama dari ketidaksesuaian ini adalah threshold default (biasanya 0.5) yang tidak cocok untuk data dengan distribusi kelas tidak seimbang (imbalanced data). Dalam kasus seperti fraud detection atau diagnosis penyakit, jumlah kelas positif jauh lebih sedikit dibanding kelas negatif, sehingga presisi cenderung rendah secara default. Strategi tuning yang dapat diterapkan adalah sebagai berikut:
    - Mengatur threshold secara manual untuk meningkatkan presisi (misalnya menaikkan threshold dari 0.5 ke 0.7).
    - Tuning hyperparameter seperti class_weight, min_samples_leaf, atau max_depth (untuk tree-based models) agar lebih fokus pada prediksi positif yang benar.
    - Menggunakan precision-recall curve untuk menemukan trade-off terbaik.

    Recall menjadi pertimbangan kritis dalam konteks ini karena recall menunjukkan seberapa banyak kasus positif yang berhasil ditemukan oleh model. Dalam aplikasi seperti deteksi penipuan, diagnosis medis, atau keamanan, false negative lebih berbahaya dibanding false positive. Misalnya, gagal mendeteksi transaksi palsu lebih merugikan daripada memblokir transaksi normal.

2. Fitur kategorikal dengan nilai unik sangat banyak (high-cardinality) seperti 1000 kategori, jika diolah tanpa strategi yang tepat, dapat menyebabkan:
    - Overfitting, karena model mencoba menyesuaikan terhadap kategori yang jarang.
    - Ketidakstabilan estimasi koefisien pada model linier atau probabilistik.

- Presisi yang tidak stabil, karena beberapa kategori yang hanya muncul di data latih akan menghasilkan prediksi yang salah besar di data uji.

Target encoding, meskipun efektif, berisiko menyebabkan data leakage karena nilai rata-rata target dihitung dari seluruh data, termasuk data uji. Ini membuat model belajar dari informasi yang seharusnya tidak tersedia saat pelatihan. Adapun jalur alternatif yang lebih aman yaitu:

- Frequency encoding, yang menggantikan kategori dengan frekuensi kemunculannya.
- Leave-One-Out (LOO) encoding, yang menghitung rata-rata target tanpa melibatkan data itu sendiri.
- Embedding (jika pakai model neural networks), yang belajar representasi numerik dari kategori besar.

3. Peningkatan presisi dari 40% ke 60% setelah normalisasi Min-Max pada model SVM linear, disertai penurunan recall 20%, menunjukkan perubahan pada decision boundary model. SVM sangat sensitif terhadap skala fitur, dan normalisasi membuat semua fitur berada dalam rentang yang sama. Ini mengubah orientasi margin dan hyperplane, yang dapat menguntungkan kelas mayoritas (lebih banyak titik margin kecil), tetapi mengorbankan sensitivitas terhadap kelas minoritas (false negative meningkat → recall turun). Sebaliknya, jika normalisasi ini diterapkan pada model seperti Gradient Boosting, efeknya bisa berbeda. Gradient boosting tidak tergantung pada skala fitur karena model membuat split berdasarkan urutan nilai, bukan nilai absolut. Justru, normalisasi bisa mereduksi informasi dalam distribusi jika dilakukan berlebihan, menyebabkan model kehilangan "sinyal penting".
4. Ketika dua fitur dikalikan (misalnya X1 * X2) dan menyebabkan AUC-ROC meningkat dari 0.75 menjadi 0.82, hal ini menunjukkan bahwa terdapat interaksi non-linier antar fitur yang sebelumnya tidak tertangkap oleh model linier. Secara matematis, perkalian dua fitur menciptakan fitur kuadratik yang memperkenalkan kelengkungan pada decision boundary. Model seperti logistic regression dapat menangkap pola ini setelah interaksi ditambahkan, karena sebelumnya model hanya mengenal pemisah linier. Uji statistik seperti chi-square hanya mendeteksi hubungan individual antar fitur dan target, bukan interaksi antar fitur. Oleh karena itu, mereka gagal mengidentifikasi kontribusi gabungan dua fitur. Terdapat juga alternatif yang bisa digunakan untuk mendeteksi interaksi yaitu:

- Domain knowledge, misalnya memahami bahwa “umur * total pinjaman” mungkin lebih bermakna daripada masing-masing variabel secara terpisah.

- Model explainability tools seperti SHAP interaction values, yang secara eksplisit mengukur efek interaksi antar fitur terhadap output model.

5. Oversampling yang dilakukan sebelum pembagian train-test merupakan kesalahan umum yang menyebabkan data leakage. Dalam kasus ini, model mendapat "bocoran" dari distribusi kelas positif karena duplikasi dari data uji masuk ke data latih. Hal ini menjelaskan mengapa AUC-ROC validasi tinggi (0.95) tetapi testing drop drastis ke 0.65. Pada masalah seperti fraud detection yang bersifat time-sensitive, temporal split jauh lebih aman. Data dari masa depan seharusnya tidak muncul saat model dilatih, agar lebih sesuai dengan kondisi deployment nyata. Penggunaan stratified sampling juga bisa memperparah masalah jika dilakukan setelah oversampling, karena model akan melihat distribusi target yang tidak realistis di seluruh data. Desain preprocessing yang benar yaitu sebagai berikut:
    - Lakukan train-test split terlebih dahulu berdasarkan waktu atau secara acak.
    - Lakukan oversampling atau undersampling hanya di data latih.
    - Evaluasi metrik seperti presisi, recall, dan AUC-ROC hanya di data uji untuk memastikan hasil yang realistis.