1. Jika model Machine Learning menunjukkan AUC-ROC tinggi (0.92) tetapi Presisi sangat rendah (15%) pada dataset tersebut, jelaskan faktor penyebab utama ketidaksesuaian ini! Bagaimana strategi tuning hyperparameter dapat meningkatkan Presisi tanpa mengorbankan AUC-ROC secara signifikan? Mengapa Recall menjadi pertimbangan kritis dalam konteks ini, dan bagaimana hubungannya dengan cost false negative?

   Hal ini terjadi karena adanya class imbalance. Di dalam dataset terdapat banyak nilai 0, maka model dapat dengan mudah menebak nilai 0, namun karena sangat sedikit nilai 1 model akan sulit menebak nilai 1. Strategi yang dapat dilakukan adalah undersampling kelas mayoritas atau oversampling kelas minoritas.

   Pada praktek oversampling, presisi turun namun recall naik, atau hal sebaliknya bisa terjadi. Ketika kelas 1 dilakukan oversampling, karena jumlah aslinya sangat sedikit, maka data X pada kelas 1 mungkin menunjukkan hal yang mirip-mirip.

2. Sebuah fitur kategorikal dengan 1000 nilai unik (high-cardinality) digunakan dalam model machine learning. Jelaskan dampaknya terhadap estimasi koefisien dan stabilitas Presisi! Mengapa target encoding berisiko menyebabkan data leakage dalam kasus dataset tersebut, dan alternatif encoding apa yang lebih aman untuk mempertahankan AUC-ROC?

   Dengan banyaknya nilai kategorikal unik, model dapat overfitting ketika melihat fitur-fitur tersebut.
   Target encoding beresiko menyebabkan data leakage jika dilakukan sebelum split train/test, maka informasi dari target test ikut masuk.
   Encoding yang lebih aman salah satunya frequency encoding, yaitu mengganti kategori dengan frekuensi kemunculannya, dan tidak bergantung pada target.

3. Setelah normalisasi Min-Max, model SVM linear mengalami peningkatan Presisi dari 40% ke 60% tetapi Recall turun 20%. Analisis dampak normalisasi terhadap decision boundary dan margin kelas minoritas! Mengapa scaling yang sama mungkin memiliki efek berlawanan jika diterapkan pada model Gradient Boosting?

   Setelah saya coba, ternyata Min-Max dengan model SVM justru meningkatkan recall (notebook classification-SVM). Namun mungkin yang disebut presisi meningkan namun recall turun karena min-max adalah min-max menskala fitur ke jangka 0 hingga 1, jadi semua fitur berkontribusi secara relative dalam menghitung margin. Decision boundary menjadi lebih stabil karena skala fitur seragam. Margin lebih ketat terhadap kelas minoritas, sehingga distribusinya sangat berbeda. Presisi naik karena lebih sedikit false positive, dan recall turun karena lebih banyak false negative. Efeknya berlawanan

pada gradient boosting karena tree-based tidak menggunakan jarak, namun membandingkan nilai untuk membuat decision nya.

4. Eksperimen feature interaction dengan menggabungkan dua fitur melalui perkalian meningkatkan AUC-ROC dari 0.75 ke 0.82. Jelaskan mekanisme matematis di balik peningkatan ini dalam konteks decision boundary non-linear! Mengapa uji statistik seperti chi-square gagal mendeteksi interaksi semacam ini, dan metode domain knowledge apa yang dapat digunakan sebagai alternatif?

   Dalam model linear, decision boundary hanya bisa memisahkan data secara linear, dengan menggabungkan dua fitur, model dapat mewakili non-linear surface, dan hal ini membuat decision boundary menjadi non-linear.
   Chi-Square gagal mendeteksi interaksi tersebut karena ia hanya mengevaluasi hubungan antara dua variabel dan asumsinya univariat. Chi-Square tidak menguji kombinasi dua fitur terhadap target. Tree-based feature importance (seperti dari XGBoost) dapat digunakan sebagai alternatif.

5. Dalam pipeline preprocessing, penggunaan oversampling sebelum pembagian train-test menyebabkan data leakage dengan AUC-ROC validasi 0.95 tetapi AUC-ROC testing 0.65. Jelaskan mengapa temporal split lebih aman untuk fraud detection, dan bagaimana stratified sampling dapat memperparah masalah ini! Bagaimana desain preprocessing yang benar untuk memastikan evaluasi metrik Presisi/Recall yang realistis?

   Oversampling sebelum split dapat menyebabkan data leakage karena ia menciptakan sampel sintetis berdasarkan tetangga di data aslinya, oleh karena itu sampel sintetis dapat berasal dari data test jika belum displit.
   Temporal split lebih aman karena pola berubah seiring waktu, misalnya data train berada di awal, data test berada di akhir.
   Stratified sampling memperarah masalah karena memastikan distribusi seimbang, tetapi tidak memperhitungkan waktu. Dalam fraud detection, transaksi terbaru mungkin punya pola berbeda.
   Pipeline dengan urutan yang benar biasanya split data terlebih dahulu dan sampling data di train set.