Nama : Chandra Aulia Haswangga

NIM : 1103223163

Kelas : TK-45-G05

Tugas : UTS – Analisis Klasifikasi

1. AUC-ROC Tinggi tapi Precision Rendah
   a) Faktor Penyebab
      - Imbalance Class
        AUC-ROC yang tinggi menunjukkan bahwa model dapat memisahkan kelas positif dan negatif dengan baik, tetapi precision yang sangat rendah (15%) menunjukkan bahwa model sering mengklasifikasikan banyak kasus negatif sebagai positif (false positives). Hal ini sering terjadi pada dataset dengan imbalanced classes, di mana satu kelas jauh lebih banyak daripada kelas lainnya.

      - Threshold yang Tidak Optimal
        Meskipun AUC-ROC menunjukkan bahwa model dapat memisahkan dua kelas dengan baik, precision yang rendah dapat disebabkan oleh threshold probabilitas yang terlalu rendah untuk mendefinisikan positif. Jika threshold terlalu rendah, model akan mengklasifikasikan lebih banyak contoh sebagai positif, yang menyebabkan banyak false positives.

   b) Strategi Tuning Hyperparameter
      - Menyesuaikan Threshold:
        Mengatur threshold keputusan model untuk mendapatkan trade-off antara recall dan precision yang lebih baik. Misalnya, mengubah threshold untuk mengurangi jumlah positif yang diprediksi dapat meningkatkan precision.

      - Penggunaan Precision-Recall Curve
        Sebagai alternatif, AUC-ROC dapat diganti dengan Precision-Recall AUC untuk mendapatkan gambaran yang lebih jelas tentang kinerja model pada data yang tidak seimbang.

      - Hyperparameter Tuning
        Menggunakan teknik seperti grid search atau random search untuk mencari parameter yang lebih baik, misalnya pengaturan pada class weight atau penalty parameter untuk memperbaiki kelas minoritas tanpa merusak kinerja AUC-ROC.

   c) Pertimbangan Recall dan Cost False Negative
      - Recall menjadi penting dalam konteks ini, terutama jika biaya false negative (FN) sangat tinggi. Misalnya, dalam kasus deteksi penipuan (fraud detection), false negative berarti tidak mendeteksi penipuan yang

ada, yang dapat berakibat pada kerugian finansial besar. Recall mengukur seberapa banyak kasus positif yang berhasil diidentifikasi, jadi meskipun precision rendah, model yang memiliki recall tinggi dapat lebih berguna jika biaya FN lebih besar daripada FP.

2. Fitur Kategorikal dengan 1000 Nilai Unik
   a) Dampak Terhadap Estimasi Koefisien dan Stabilitas Precision
      - Overfitting
        Fitur dengan 1000 nilai unik dapat menyebabkan model menjadi sangat kompleks, dengan banyaknya parameter yang harus dipelajari. Hal ini meningkatkan risiko overfitting pada data training, karena model dapat "memahami" noise atau fluktuasi kecil pada data.

      - Estimasi Koefisien
        Pada model berbasis koefisien seperti regresi linear, adanya banyak nilai unik dalam fitur kategorikal dapat menyebabkan ketidakstabilan dalam estimasi koefisien, karena model akan memiliki banyak kategori yang sedikit kontribusinya.

   b) Risiko Data Leakage dengan Target Encoding
      - Target encoding menggantikan kategori dengan rata-rata target untuk masing-masing kategori. Namun, jika target encoding dilakukan sebelum pembagian data ke train-test, maka informasi dari data test akan bocor ke dalam model, yang mengarah pada data leakage. Model akan mendapatkan informasi yang seharusnya tidak tersedia selama pelatihan.

   c) Alternatif Encoding yang Lebih Aman
      - One-Hot Encoding
        Salah satu alternatif yang lebih aman adalah one-hot encoding. Ini akan mengubah fitur kategorikal menjadi serangkaian fitur biner, meskipun untuk fitur dengan banyak nilai unik, ini dapat menyebabkan masalah dimensi yang tinggi.

      - Hashing
        Teknik lain yang bisa dipertimbangkan adalah feature hashing, yang dapat mengurangi dimensi fitur kategorikal besar dengan menggunakan hash fungsi.

3. Dampak Normalisasi pada SVM dan Gradient Boosting
   a) Dampak Normalisasi pada SVM
      - Decision Boundary
        Normalisasi Min-Max merubah skala fitur sehingga setiap fitur memiliki rentang yang seragam. Ini sangat penting untuk SVM karena model ini sangat bergantung pada jarak antar titik data untuk menentukan decision boundary. Normalisasi meningkatkan marginal separation antara kelas dan memungkinkan SVM menemukan margin optimal yang lebih baik.

- Minority Class
  Dengan SVM, normalisasi mengurangi masalah bias terhadap fitur dengan rentang lebih besar dan dapat meningkatkan presisi dengan memaksimalkan margin pada kelas minoritas.

b) Dampak pada Gradient Boosting
   - Tidak Terpengaruh Secara Signifikan
     Gradient Boosting adalah model berbasis pohon keputusan, yang tidak terlalu sensitif terhadap skala fitur. Normalisasi dapat mempengaruhi performa, tetapi efeknya tidak sebesar pada SVM. Jika diterapkan pada Gradient Boosting, normalisasi mungkin tidak memperbaiki dan malah bisa memperburuk performa karena model ini lebih mengutamakan pemecahan fitur berdasarkan threshold (bukan jarak antar titik data).

4. Eksperimen Feature Interaction dan Peningkatan AUC-ROC
   a) Mekanisme Matematis
      - Interaksi Fitur
        Dengan menggabungkan dua fitur melalui perkalian, model sekarang dapat menangkap interaksi non-linear antara kedua fitur tersebut. Hal ini menciptakan fitur baru yang mengubah decision boundary dari linear menjadi lebih kompleks dan non-linear, memungkinkan model untuk lebih baik memisahkan kelas dengan meningkatkan AUC-ROC.

   b) Mengapa Chi-Square Gagal
      - Chi-square digunakan untuk mengukur ketergantungan antara fitur kategorikal, namun interaksi non-linear yang diciptakan melalui perkalian dua fitur numerik tidak dapat dideteksi oleh uji statistik ini. Chi-square tidak menangkap interaksi numerik yang tidak bersifat linear, yang merupakan alasan utama mengapa metode ini gagal mendeteksi interaksi semacam itu.

   c) Alternatif
      - Domain Knowledge
        Penggunaan pengetahuan domain untuk mengidentifikasi fitur yang mungkin memiliki hubungan interaktif, seperti fitur yang menggambarkan hubungan produk atau rasio antara dua variabel yang relevan.

5. Masalah Data Leakage pada Oversampling
   a) Mengapa Oversampling Sebelum Pembagian Train-Test Menyebabkan Data Leakage
      - Temporal Split lebih aman dalam fraud detection karena data penipuan biasanya mengikuti pola waktu (misalnya, penipuan yang terjadi selama periode tertentu). Dengan memisahkan data berdasarkan waktu, kita mencegah data dari masa depan digunakan untuk melatih model, menjaga integritas evaluasi.

b) Masalah dengan Stratified Sampling
    - Stratified sampling bertujuan untuk memastikan distribusi kelas tetap konsisten di seluruh subset data. Namun, pada kasus fraud detection, stratifikasi bisa memperburuk masalah jika data yang lebih relevan (misalnya, kejadian penipuan yang lebih baru) dibocorkan ke dalam data pelatihan.

c) Desain Preprocessing yang Benar
    - Untuk menghindari data leakage, lakukan oversampling setelah pembagian train-test. Selain itu, untuk memastikan metrik Precision/Recall yang realistis, gunakan cross-validation stratified pada data pelatihan, dan pastikan evaluasi dilakukan hanya pada data yang benar-benar tersedia selama pelatihan.