Nama : Michael Christopher
NIM : 1103210260
Analisis UTS ML Soal #2

Analisis Model Machine Learning dan Aspek Teknisnya

1. Ketidaksesuaian antara AUC-ROC Tinggi dan Presisi Rendah

Ketidaksesuaian antara AUC-ROC tinggi (0.92) dan Presisi rendah (15%) biasanya disebabkan oleh faktor utama: **ketidakseimbangan kelas (class imbalance)**. Model mungkin baik dalam membedakan kelas secara keseluruhan (AUC-ROC tinggi), tetapi gagal memberikan prediksi positif yang akurat (Presisi rendah).

Faktor penyebab utama:

- **Distribusi kelas yang tidak seimbang** (kelas positif sangat minoritas)
- **Threshold prediksi default (0.5)** yang tidak optimal
- **Optimasi fungsi objektif** yang tidak mempertimbangkan distribusi kelas

Strategi tuning hyperparameter untuk meningkatkan Presisi:

- **Kalibrasi threshold prediksi** - menggeser threshold dari 0.5 ke nilai yang lebih tinggi akan mengurangi false positives (meningkatkan Presisi)
- **Class weight adjustment** - memberikan bobot lebih tinggi pada kelas minoritas
- **Regularisasi yang tepat** (L1/L2) untuk mengurangi overfitting pada kelas mayoritas
- **Focal loss** atau teknik cost-sensitive learning untuk memfokuskan pada kasus sulit

Recall menjadi pertimbangan kritis karena memiliki hubungan trade-off dengan Presisi. Meningkatkan Presisi biasanya mengorbankan Recall. Dalam konteks false negative, Recall mengukur kemampuan model mengidentifikasi semua kasus positif yang sebenarnya. Cost false negative sering kali sangat tinggi pada domain seperti deteksi penipuan, diagnosis medis, atau sistem keamanan—di mana tidak mendeteksi kasus positif (false negative) dapat berakibat serius.

2. Dampak Fitur Kategorikal High-Cardinality

Fitur kategorikal dengan 1000 nilai unik (high-cardinality) berdampak signifikan pada:

- **Estimasi koefisien**: Menyebabkan sparse matrices yang menghasilkan estimasi koefisien tidak stabil dan bervariansi tinggi
- **Stabilitas Presisi**: Presisi menjadi tidak stabil karena model cenderung overfitting pada nilai kategori yang jarang muncul

- **Curse of dimensionality**: One-hot encoding akan menghasilkan 1000 fitur baru, meningkatkan dimensi secara drastis

Target encoding berisiko menyebabkan data leakage karena:

- Menghitung statistik target (mean, median) untuk setiap kategori menggunakan seluruh dataset
- Informasi dari data testing dapat "bocor" ke data training
- Overfitting pada kategori yang jarang muncul

Alternatif encoding yang lebih aman:

- **Leave-one-out encoding** - menghitung statistik tanpa melibatkan data yang sedang dilihat
- **Mean regularized target encoding** - gabungan global mean dan kategorikal mean
- **Bin-counting** - pengelompokan kategori jarang ke dalam bin
- **Feature hashing** - memetakan kategori ke dimensi yang lebih kecil
- **Embedding layers** - untuk model deep learning

3. Dampak Normalisasi Min-Max pada SVM vs Gradient Boosting

Normalisasi Min-Max meningkatkan Presisi SVM linear tetapi menurunkan Recall karena:

- Min-Max mengubah skala fitur ke rentang [0,1], menyamaratakan pengaruh setiap fitur
- Pada SVM linear, normalisasi mempengaruhi decision boundary secara langsung, menghasilkan hyperplane yang lebih optimal menurut margin maksimum
- Margin kelas minoritas menjadi lebih ketat, menghasilkan lebih sedikit false positives (Presisi naik) tetapi lebih banyak false negatives (Recall turun)
- Decision boundary bergeser karena perubahan bobot relatif fitur dalam fungsi keputusan linear

Gradient Boosting menunjukkan efek berlawanan karena:

- Gradient Boosting tidak sensitif terhadap penskalaan fitur—model berbasis tree membagi ruang fitur berdasarkan threshold relatif
- Algoritma tree secara inheren menangani berbagai skala data
- Normalisasi dapat menghilangkan informasi tentang outlier yang mungkin penting untuk model ensemble
- Split tree ditentukan oleh peringkat nilai, bukan nilai absolut, sehingga Min-Max normalisasi memiliki efek minimal atau bahkan merugikan

4. Mekanisme Peningkatan AUC-ROC melalui Feature Interaction

Feature interaction melalui perkalian dua fitur meningkatkan AUC-ROC dari 0.75 ke 0.82 karena:

Mekanisme matematis:

- Perkalian fitur ($F_1 \times F_2$) membuat decision boundary non-linear dalam ruang fitur asli
- Untuk model linear: $f(x) = w_1x_1 + w_2x_2 \rightarrow f(x) = w_1x_1 + w_2x_2 + w_3(x_1 \times x_2)$
- Transformasi ini memungkinkan model mengenali pola kompleks yang tidak terdeteksi oleh fitur individual
- Secara geometris, menciptakan dimensi baru yang memungkinkan separasi kelas yang lebih baik

Uji statistik seperti chi-square gagal mendeteksi interaksi tersebut karena:

- Chi-square melihat asosiasi antara variabel kategorik secara independen
- Tidak dirancang untuk mendeteksi interaksi non-linear kompleks
- Hanya mengevaluasi hubungan satu-ke-satu, bukan interaksi multi-dimensi

Metode domain knowledge alternatif:

- **Partial Dependence Plots (PDP)** untuk visualisasi interaksi
- **SHAP (SHapley Additive exPlanations)** interaction values
- **Feature crossing** berdasarkan pemahaman domain
- **Polynomial features** dengan koefisien domain-informed

5. Masalah Oversampling dan Strategi Preprocessing yang Benar

Oversampling sebelum pembagian train-test menyebabkan data leakage karena:

- Duplikasi/sintesis data minoritas sebelum split membuat data training dan testing tidak independen
- Informasi dari data testing "bocor" ke data training melalui sampel sintetis
- Model mengenali pola dalam data testing yang seharusnya tidak terlihat selama training

Temporal split lebih aman untuk fraud detection karena:

- Pola fraud berkembang seiring waktu (concept drift)
- Memungkinkan model divalidasi pada data masa depan, mempertahankan integritas temporal
- Mencegah data leakage dari masa depan ke masa lalu

Stratified sampling dapat memperparah masalah saat:

- Menjaga distribusi kelas yang sama di seluruh split tetapi mencampur karakteristik temporal
- Dalam fraud detection, stratifikasi dapat menyebabkan pola fraud masa depan bocor ke data training

Desain preprocessing yang benar:

1. **Split data terlebih dahulu** (train/validation/test)
2. **Terapkan preprocessing hanya pada data training**
3. **Simpan parameter transformasi** dari data training
4. **Terapkan transformasi yang sama** pada data validation dan test

5. **Lakukan oversampling hanya pada data training** setelah split
6. **Gunakan time-based validation** untuk kasus fraud detection
7. **Evaluasi metrik pada distribusi yang tidak dimodifikasi** untuk estimasi Presisi/Recall yang realistis