Kevin Alif Bagaskara

1103210075

UTS Machine Learning

1. Jika model linear regression atau decision tree mengalami underfitting pada dataset ini, strategi apa yang akan digunakan untuk meningkatkan performanya? Bandingkan setidaknya dua pendekatan berbeda (misal: transformasi fitur, penambahan features, atau perubahan model ke algoritma yang lebih kompleks), dan jelaskan bagaimana setiap solusi memengaruhi bias-variance tradeoff!

Jika model seperti Linear Regression atau Decision Tree mengalami underfitting, artinya model tersebut terlalu sederhana untuk menangkap pola yang kompleks dalam data. Dua pendekatan yang umum adalah meningkatkan kompleksitas model atau memperkaya fitur. Transformasi fitur seperti menambahkan interaksi antar variabel atau menggunakan teknik seperti PolynomialFeatures dapat membantu model linear menangkap non-linearitas. Ini mengurangi bias, tapi dapat meningkatkan variansi jika tidak diimbangi dengan regularisasi. Kedua adalah mengganti model ke algoritma yang lebih kompleks seperti Gradient Boosting atau Random Forest bisa mengatasi keterbatasan Decision Tree tunggal yang terlalu dangkal. Model sperti Gradient Boosting akan memperbaiki kesalahan dari model sebelumnya. Namun, ini juga meningkatkan risiko overfitting jika tidak dikontrol dengan hyperparameter tuning.

2. Selain MSE, jelaskan dua alternatif loss function untuk masalah regresi (misal: MAE, Huber loss) dan bandingkan keunggulan serta kelemahannya. Dalam skenario apa setiap loss function lebih cocok digunakan? (Contoh: data dengan outlier, distribusi target non-Gaussian, atau kebutuhan interpretasi model).

Selain MSE, dua loss function penting dalam regresi adalah MAE (Mean Absolute Error) dan Huber Loss. MAE menghitung rata-rata absolut dari selisih antara nilai prediksi dan aktual. Kelebihannya adalah lebih tahan terhadap outlier karena tidak menghukum kesalahan besar secara kuadrat, berbeda dengan MSE. Namun, MAE kurang stabil dalam proses optimasi karena gradiennya konstan, yang bisa memperlambat konvergensi. Di sisi lain, Huber Loss adalah kompromi antara MAE dan MSE: pada kesalahan kecil ia bersifat seperti MSE (kuadrat), sedangkan pada kesalahan besar ia bersifat seperti MAE (linear). Ini menjadikannya cocok untuk data yang mengandung outlier, tanpa sepenuhnya mengabaikannya seperti MAE atau memperbesarnya seperti MSE. Secara praktis, MAE cocok untuk skenario dengan distribusi target non-Gaussian dan banyak outlier, sementara Huber lebih stabil untuk pelatihan model ketika data memiliki kombinasi error kecil dan besar.

3. Tanpa mengetahui nama fitur, metode apa yang dapat digunakan untuk mengukur pentingnya setiap fitur dalam model? Jelaskan prinsip teknikal di balik metode tersebut (misal: koefisien regresi, feature importance berdasarkan impurity reduction) serta keterbatasannya!

Kita bisa menggunakan dua pendekatan utama: koefisien dalam model linier dan feature importance dari model pohon keputusan. Dalam Linear Regression, besar kecilnya koefisien menunjukkan kekuatan pengaruh fitur terhadap target, dengan asumsi semua fitur telah distandarisasi. Namun, pendekatan ini hanya mengukur korelasi linier dan bisa menyesatkan jika terjadi multikolinearitas. Dalam model Decision Tree atau ensemble-nya, feature importance dihitung berdasarkan seberapa banyak impurity yang berhasil dikurangi oleh fitur saat pemisahan data di tree. Kelebihan metode ini

adalah dapat menangkap interaksi non-linear antar fitur. Kekurangannya adalah bias terhadap fitur dengan banyak kategori, dan sulitnya interpretasi absolut jika fitur saling bergantung. Metode tambahan seperti permutation importance atau SHAP (SHapley Additive exPlanations) bisa digunakan untuk memberi pandangan yang lebih fair dan stabil.

4. Bagaimana mendesain eksperimen untuk memilih hyperparameter optimal (misal: learning rate untuk SGDRegressor, max_depth untuk Decision Tree) pada dataset ini? Sertakan analisis tradeoff antara komputasi, stabilitas pelatihan, dan generalisasi model!

Bisa menggunakan Grid Search atau Randomized Search dengan cross-validation. Misalnya kita ingin memilih max_depth untuk Decision Tree atau learning_rate untuk SGDRegressor. Dengan Grid Search, kita mencoba semua kombinasi nilai yang mungkin, sedangkan Randomized Search mencoba subset secara acak, yang jauh lebih efisien bila ruang parameter sangat besar. Menggunakan cross-validation penting agar hasil evaluasi stabil dan tidak overfit terhadap satu pembagian data. Namun, ada tradeoff: semakin banyak kombinasi dan fold, semakin besar juga biaya komputasinya. Untuk dataset besar atau model kompleks seperti Gradient Boosting, pendekatan seperti Bayesian Optimization atau Hyperband bisa digunakan untuk efisiensi. Dalam memilih hyperparameter, kita juga harus mempertimbangkan generalisasi. Misalnya, max_depth yang terlalu besar bisa menurunkan training error tapi meningkatkan overfitting.

5. Jika menggunakan model linear regression dan residual plot menunjukkan pola non-linear serta heteroskedastisitas, langkah-langkah apa yang akan diambil? (contohnya: Transformasi data/ubah model yang akan dipakai/etc)

Jika residual plot dari model Linear Regression menunjukkan pola non-linear dan heteroskedastisitas, itu adalah indikasi kuat bahwa model salah spesifikasi. Langkah pertama yang bisa diambil adalah mentransformasi fitur atau target. Misalnya, menggunakan log atau square root transformasi pada variabel yang sangat skewed bisa membantu menstabilkan varians dan mengurangi heteroskedastisitas. Kedua, menambahkan fitur polinomial atau interaksi antar fitur bisa membantu model linier menangkap pola non-linear. Namun, jika pola terlalu kompleks, pendekatan yang lebih cocok adalah mengganti model dengan algoritma non-linier seperti Decision Tree, Random Forest, atau SVR yang tidak mengasumsikan hubungan linier. Untuk masalah heteroskedastisitas secara spesifik, pendekatan seperti Weighted Least Squares (WLS) juga bisa digunakan, di mana observasi dengan varians lebih besar diberikan bobot lebih kecil. Residual plot juga sebaiknya tetap digunakan setelah transformasi atau pergantian model untuk memastikan masalah sudah teratasi.