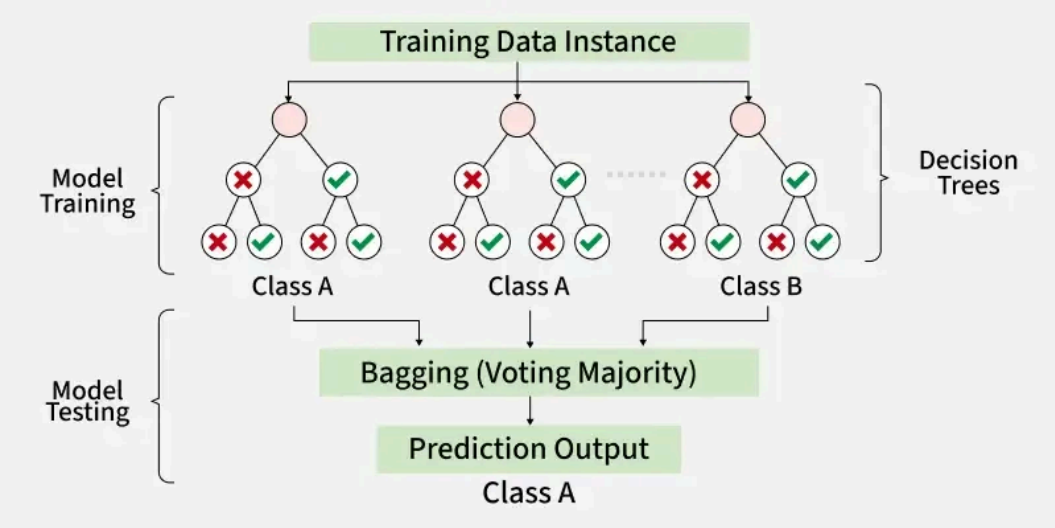
Random Forest

Salman Hanif - 13523056

Cara Kerja Algoritma

Random Forest adalah model ensemble yang menggabungkan banyak Decision Tree dan melakukan voting dalam pengambilan keputusan akhir. Konsep randomisasi di sini digunakan untuk pembuatan beberapa decision tree (pengambilan subset acak untuk pelatihan tree). Di setiap langkah pemisahan (split) dalam setiap pohon, algoritma hanya mempertimbangkan subset fitur yang dipilih secara acak. Ini mencegah satu fitur yang sangat kuat mendominasi semua pohon dan memastikan setiap pohon memiliki pandangan yang unik terhadap data.



Alur kerja fungsi fit dan transform :

Fit adalah proses di mana seluruh hutan (forest) dibangun :

1. Model akan mengulangi prosesnya sebanyak n_trees. Setiap iterasi akan membangun satu pohon keputusan baru. Di setiap iterasi, algoritma mengambil sampel acak dari data pelatihan (X, y) dengan pengembalian. Ini menciptakan subset data yang berbeda untuk setiap pohon, yang dikenal sebagai bootstrap sample.
2. Setelah mendapatkan sampel data, model membuat instance DecisionTree baru dan memanggil metode fit-nya. Di sinilah random sampling fitur terjadi. Selama proses growTree, di setiap pemisahan, model hanya memilih subset fitur acak (n_Features) untuk mencari pemisahan terbaik, bukan semua fitur.

3. Setiap tree yang sudah dibuat disimpan pada list tree class random forest

Predict:

1. untuk membuat prediksi. Metode tree.predict(X) dipanggil untuk setiap pohon, dan hasilnya disimpan dalam array tree_preds
2. Setelah semua pohon memberikan prediksinya, model akan menggabungkan prediksi-prediksi tersebut. Karena ini adalah masalah klasifikasi, metode getModusLabel digunakan. Modus yang ditemukan untuk setiap sampel data menjadi prediksi akhir dari model Random Forest.

Hasil Evaluasi model dari hasil scratch dan dari library. Jelaskan perbedaan

Dalam evaluasi, model saya dapat bekerja dengan baik, performa juga terukur lebih baik dari sklearn. Tetapi ada kemungkinan model saya lebih overfit. Hal ini dikarenakan saya belum melakukan tuning pada parameter model saya seperti jumlah tree yang dibangun, maxDepth tree, minDataSplit, n_features. Pada inisialisasi tanpa pengisian nilai parameter sklearn masih mengungguli model saya.

Dalam waktu pun masih belum seoptimal sklearn, hal ini karena saya belum melakukan multiprocessing, pembangunan model sklearn pun dikatakan pada bahasa C++ dan optimisasi nya di python lebih baik.

Improvement yang bisa dilakukan

Melakukan parameter tuning pada random forest agar performanya membaik. Penambahan Gini Impurity untuk penentuan pemisahan data bisa ditambahkan sebagai alternatif.

Perlakuan multithreading/multiprocessing selama pembangunan tree akan meningkatkan performa waktu.