Chapter 7: Ensemble Learning and Random Forests

Pendahuluan

Ensemble Learning adalah sebuah paradigma dalam Machine Learning di mana beberapa model individual, yang sering disebut *weak learners* (pembelajar lemah), digabungkan untuk menghasilkan satu model prediktor akhir yang lebih kuat, atau *strong learner*. Ide di baliknya mirip dengan pepatah "kebijaksanaan orang banyak" (*wisdom of the crowd*): keputusan yang dibuat oleh sekelompok besar orang yang beragam seringkali lebih baik daripada keputusan seorang ahli tunggal. Sebuah grup model yang digabungkan disebut sebagai *ensemble*.

1. Voting Classifiers

Cara paling sederhana untuk menggabungkan beberapa *classifier* yang berbeda adalah dengan mengagregasi prediksi dari setiap *classifier* dan memilih kelas yang mendapatkan suara terbanyak.

- **Hard Voting**: Metode ini mengambil prediksi dari setiap *classifier* dan memilih kelas yang paling sering muncul sebagai prediksi akhir. Ini adalah pendekatan mayoritas sederhana.
- **Soft Voting**: Metode ini menghitung rata-rata probabilitas prediksi dari setiap *classifier* untuk setiap kelas, kemudian memilih kelas dengan probabilitas rata-rata tertinggi sebagai prediksi akhir. Soft voting seringkali berkinerja lebih baik daripada hard voting karena memberikan bobot lebih pada suara yang sangat meyakinkan (memiliki probabilitas tinggi).

2. Bagging dan Pasting

Kedua metode ini menggunakan algoritma pelatihan yang sama untuk setiap prediktor, tetapi melatihnya pada **subset acak yang berbeda** dari data pelatihan.

- **Bagging (Bootstrap Aggregating)**: Setiap prediktor dilatih pada subset acak yang diambil dari data pelatihan **dengan pengembalian** (*with replacement*). Artinya, beberapa sampel mungkin digunakan beberapa kali untuk satu prediktor, sementara yang lain mungkin tidak digunakan sama sekali.
- **Pasting**: Setiap prediktor dilatih pada subset acak yang diambil **tanpa pengembalian** (*without replacement*). Artinya, setelah sebuah sampel dipilih, ia tidak dapat dipilih lagi untuk subset prediktor tersebut.

Setelah semua prediktor dilatih, *ensemble* membuat prediksi dengan mengagregasi prediksi dari semua prediktor individual (seperti *voting* untuk klasifikasi atau rata-rata untuk regresi). Karena setiap prediktor dilatih secara independen, proses pelatihan dan prediksi dapat dilakukan secara paralel.

Out-of-Bag (OOB) Evaluation

Dalam Bagging, karena pengambilan sampel dilakukan dengan pengembalian, rata-rata hanya sekitar 63% dari data pelatihan yang diambil untuk setiap prediktor. Sisa 37% yang tidak

terpakai disebut *out-of-bag* (OOB) instances. Sampel OOB ini dapat digunakan sebagai set validasi untuk mengevaluasi kinerja *ensemble* tanpa perlu menyisihkan set validasi terpisah secara manual.

3. Random Forests

Random Forest adalah sebuah *ensemble* dari Decision Trees, yang umumnya dilatih menggunakan metode Bagging. Namun, Random Forest menambahkan satu lapisan keacakan tambahan untuk meningkatkan keragaman pohon:

- Saat membelah sebuah *node* pada Decision Tree, alih-alih mencari fitur terbaik di antara *semua* fitur, algoritma hanya mencari fitur terbaik di antara **subset acak dari fitur**.

Pendekatan ini menghasilkan keragaman pohon yang lebih besar, yang mengurangi varians dan membuat model secara keseluruhan lebih baik.

Feature Importance

Random Forest menyediakan cara yang mudah untuk mengukur pentingnya setiap fitur. Caranya adalah dengan mengukur seberapa besar sebuah fitur berhasil mengurangi *impurity* (ketidakmurnian, seperti Gini impurity) secara rata-rata di semua pohon dalam *forest*. Fitur yang lebih penting cenderung berada lebih dekat ke *root* pohon dan menghasilkan pengurangan *impurity* yang lebih besar.

4. Boosting

Boosting adalah metode *ensemble* yang bekerja secara **sekuensial**. Ide dasarnya adalah melatih prediktor secara berurutan, di mana setiap prediktor baru mencoba untuk **memperbaiki kesalahan** dari pendahulunya.

- AdaBoost (Adaptive Boosting)

Metode ini berfokus pada instance pelatihan yang salah diklasifikasikan oleh prediktor sebelumnya. Caranya adalah dengan meningkatkan bobot relatif dari instance yang salah diklasifikasikan. Prediktor berikutnya kemudian dilatih dengan data yang telah diperbarui bobotnya, sehingga lebih fokus pada kasus-kasus yang sulit.

Prediksi akhir adalah jumlah terbobot dari semua prediksi prediktor, di mana bobot setiap prediktor (αj) ditentukan oleh tingkat kesalahannya. Bobot prediktor ke-j dihitung sebagai:

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

di mana rj adalah tingkat kesalahan terbobot (weighted error rate) dari prediktor ke-j dan$η$ adalah learning rate.

- Gradient Boosting

Metode ini juga bekerja secara sekuensial, tetapi pendekatannya berbeda. Alih-alih memperbarui bobot instance, Gradient Boosting mencoba untuk melatih prediktor baru pada residual errors (kesalahan sisa) yang dibuat oleh prediktor sebelumnya.

Misalnya, prediktor pertama dilatih pada data, lalu prediktor kedua dilatih untuk memprediksi kesalahan yang dibuat oleh prediktor pertama. Prediktor ketiga dilatih untuk memprediksi kesalahan yang dibuat oleh prediktor kedua, dan seterusnya. Prediksi akhir adalah jumlah dari prediksi awal ditambah semua koreksi kesalahan dari setiap prediktor dalam urutan.

Hyperparameter learning_rate pada Gradient Boosting mengontrol kontribusi setiap pohon. Nilai yang rendah membutuhkan lebih banyak pohon untuk melatih model, tetapi seringkali menghasilkan generalisasi yang lebih baik.

5. Stacking (Stacked Generalization)

Stacking adalah metode *ensemble* yang didasarkan pada ide sederhana: alih-alih menggunakan fungsi trivial seperti *voting* untuk mengagregasi prediksi, mengapa kita tidak **melatih sebuah model** untuk melakukan agregasi ini?

Prosesnya adalah sebagai berikut:

1. Dataset pelatihan dibagi menjadi beberapa bagian.
2. Beberapa model *base learner* (lapisan pertama) dilatih pada satu bagian data.
3. *Base learner* tersebut kemudian digunakan untuk membuat prediksi pada bagian data yang lain.
4. Prediksi-prediksi dari *base learner* ini kemudian digunakan sebagai **fitur input baru** untuk melatih sebuah model akhir yang disebut **blender** atau *meta-learner*. Blender inilah yang membuat prediksi akhir.