Chapter 7: Ensemble Learning and Random Forests

Tujuan Bab

Memahami ensemble learning, yaitu teknik menggabungkan beberapa model (misalnya beberapa decision tree) untuk mendapatkan performa yang lebih baik. Fokus utama di bab ini adalah Random Forest, salah satu metode ensemble paling populer.

Konsep Utama

1. Ensemble Learning

Adalah teknik menggabungkan beberapa prediksi dari model-model berbeda untuk menghasilkan prediksi yang lebih akurat.

Contoh:

- Voting classifier untuk klasifikasi
- Averaging untuk regresi

Intinya: “kebijaksanaan massa” (wisdom of the crowd)

2. Bagging (Bootstrap Aggregating)

- Melatih beberapa model pada subsample acak dari data training (dengan pengambilan kembali).
- Kemudian menggabungkan prediksinya (mayoritas suara untuk klasifikasi, rata-rata untuk regresi).

Keuntungan:

- Mengurangi varians tanpa meningkatkan bias
- Memperkuat model yang overfit seperti Decision Trees

3. Random Forest

Random Forest adalah sekumpulan decision tree yang dilatih menggunakan bagging + tambahan randomisasi pada pemilihan fitur.

Ciri khas:

- Setiap tree dilatih pada subset data yang berbeda
- Tiap node dalam pohon hanya mempertimbangkan subset acak dari fitur (untuk pemilihan split)

Hasil: model kuat, cepat, dan tahan terhadap overfitting

```
from sklearn.ensemble import RandomForestClassifier

rnd_clf = RandomForestClassifier(n_estimators=500, max_leaf_nodes=16, n_jobs=-1)

rnd_clf.fit(X_train, y_train)
```

4. Feature Importance

Random Forest dapat digunakan untuk mengukur seberapa penting setiap fitur dalam membuat prediksi.

```
rnd_clf.feature_importances_
```

Nilai ini sangat berguna dalam seleksi fitur dan interpretasi model.

5. Out-of-Bag Evaluation

Karena setiap tree dilatih pada subset data, sebagian data tidak ikut dilatih oleh tree tertentu. Data ini bisa digunakan untuk mengukur akurasi, tanpa memerlukan set validasi tambahan.

```
rnd_clf = RandomForestClassifier(oob_score=True)

print(rnd_clf.oob_score_)
```

6. Boosting

Selain bagging, ada juga boosting, yaitu melatih model secara berurutan di mana setiap model mencoba memperbaiki kesalahan model sebelumnya.

Jenis boosting populer:

- AdaBoost
- Gradient Boosting

Boosting bekerja sangat baik tapi lebih rentan terhadap overfitting dan lebih mahal secara komputasi.

Proyek / Notebook Praktik

Isi Praktik

1. Voting Classifier

Menggabungkan beberapa model:

```
from sklearn.ensemble import VotingClassifier
voting_clf = VotingClassifier(
   estimators=[('lr', log_clf), ('rf', rnd_clf), ('svc', svm_clf)],
   voting='hard'
)
```

Menampilkan bahwa gabungan model lebih akurat dibanding model tunggal.

2. Bagging dengan Decision Tree

```
from sklearn.ensemble import BaggingClassifier
bag_clf = BaggingClassifier(DecisionTreeClassifier(), n_estimators=500, bootstrap=True, n_jobs=-1)
```

Menunjukkan hasil akurasi lebih stabil dibandingkan decision tree tunggal.

3. Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rnd_clf = RandomForestClassifier(n_estimators=500, max_leaf_nodes=16)
```

Mengukur feature importance dan membandingkan performa terhadap model tunggal.

4. AdaBoost

```
from sklearn.ensemble import AdaBoostClassifier

ada_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=1), n_estimators=200)
```

Model boosting sederhana yang bekerja secara iteratif.

5. Gradient Boosting

```
from sklearn.ensemble import GradientBoostingRegressor

gbrt = GradientBoostingRegressor(max_depth=2, n_estimators=3, learning_rate=1.0)
```

Lebih fleksibel daripada AdaBoost dan sering digunakan di kompetisi seperti Kaggle.

Inti Pelajaran

Konsep	Penjelasan
Ensemble Learning	Gabungan beberapa model sederhana menjadi satu model kuat
Bagging	Melatih model secara paralel pada data acak
Random Forest	Sekumpulan decision tree dengan randomisasi fitur
Boosting	Melatih model secara berurutan untuk memperbaiki kesalahan
Feature Importance	Random Forest bisa menunjukkan fitur mana yang paling berpengaruh

Kelebihan Random Forest

- Akurasi tinggi
- Tidak mudah overfit
- Dapat digunakan untuk klasifikasi, regresi, dan estimasi fitur penting

Kekurangan

- Tidak terlalu interpretable
- Tidak secepat model linear pada data besar