Chapter 6: Decision Trees

Tujuan Bab

Bab ini menjelaskan algoritma Decision Tree, bagaimana cara kerjanya, bagaimana melatihnya menggunakan Scikit-Learn, serta kelemahan dan cara menghindarinya.

Konsep Utama

1. Intuisi Decision Tree

- Decision Tree adalah model yang memetakan keputusan dalam bentuk pohon bercabang.
- Setiap node mengajukan pertanyaan (misalnya $x_1 < 5$?), dan setiap cabang mewakili jawaban (ya atau tidak).
- Proses berlanjut sampai mencapai daun (leaf), yang memberikan prediksi akhir.

2. Gini Impurity dan Entropy

Decision Tree membagi data berdasarkan metrik ketidakmurnian seperti:

- Gini Impurity (default di Scikit-Learn)
- Entropy (mirip konsep informasi dalam teori informasi)

Tujuan pemilihan split adalah meminimalkan impurity.

3. Training Decision Tree

Proses pelatihan:

- Mulai dari root node
- Coba semua fitur dan threshold
- Pilih split terbaik berdasarkan Gini/Entropy
- Ulangi secara rekursif hingga kondisi berhenti

4. Overfitting dan Pruning

- Decision Trees bisa overfit jika dibiarkan tumbuh tanpa batas.
- Pre-pruning: membatasi kedalaman pohon saat pelatihan (max_depth, min_samples_split, dll).

- Post-pruning: pangkas pohon setelah pelatihan berdasarkan validasi.

5. Decision Tree Regression

Mirip seperti klasifikasi, tapi tiap daun menghasilkan nilai rata-rata target daripada label kategori.

Proyek / Notebook Praktik

Isi Praktik

1. Training Decision Tree Classifier

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
tree_clf = DecisionTreeClassifier(max_depth=2)
tree_clf.fit(iris.data, iris.target)
```

Model dilatih untuk mengklasifikasi dataset Iris berdasarkan sepal/petal.

2. Visualisasi Pohon

```
from sklearn.tree import export_graphviz
export_graphviz(tree_clf, out_file="iris_tree.dot", ...)
```

Menghasilkan file .dot untuk visualisasi struktur pohon.

3. Metrik Impurity

Menjelaskan bagaimana Gini dan Entropy dihitung secara manual dan bagaimana pengaruhnya terhadap split.

4. Decision Tree untuk Regression

```
from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(max_depth=2)
tree_reg.fit(X, y)
```

Model regresi yang memecah input dan memprediksi nilai rata-rata target di tiap segmen.

5. Overfitting dan Regularisasi

Menjelaskan penggunaan parameter seperti:

- max_depth
- min_samples_split
- min_samples_leaf
- max_leaf_nodes

Untuk mencegah model tumbuh terlalu kompleks dan overfit.

Inti Pelajaran

Konsep	Penjelasan
Decision Tree	Model interpretable yang bekerja dengan cara membagi fitur
Gini / Entropy	Digunakan untuk menentukan split terbaik
Overfitting	Decision Tree sangat rentan terhadap overfitting
Pruning	Cara mengurangi overfitting, baik sebelum atau setelah training
Tree Regression	Versi regresi dari Decision Tree, memecah input space dan mengoutput nilai rata-rata

Kelebihan Decision Tree

- Mudah dimengerti dan divisualisasi
- Tidak perlu scaling fitur
- Bisa menangani data kategorikal dan numerik

Kekurangan

- Rentan overfitting
- Sensitif terhadap perubahan data (non-robust)

- Tidak selalu menghasilkan model paling akurat