Rizqika Mulia Pratama
13522126

Decision Tree Classifier

1. Cara kerja algoritma

Decision tree classifier membagi data ke dalam cabang-cabang berdasarkan nilai fitur, dengan tujuan memaksimalkan informasi yang diperoleh dari setiap pembagian. Setiap cabang mewakili keputusan berdasarkan fitur tertentu, dan setiap daun (leaf) mewakili kelas akhir dari data.

1. Membangun pohon keputusan

   Decision tree membangun pohon dengan cara memilih fitur yang paling baik untuk membagi data pada setiap langkah. Pemilihan fitur ini dilakukan dengan menggunakan metrik seperti:
   - Gini Impurity: Mengukur seberapa sering suatu kelas akan dipilih secara acak jika kita memilih satu elemen dari dataset.
   - Entropy: Mengukur ketidakpastian dalam data. Semakin rendah entropi, semakin baik pemisahan kelas.
   - Information Gain: Selisih antara entropi sebelum dan sesudah pembagian. Fitur dengan information gain tertinggi dipilih untuk membagi data.

2. Membagi data

   Setelah memilih fitur terbaik, data dibagi menjadi subset berdasarkan nilai fitur tersebut. Proses ini diulang untuk setiap subset hingga salah satu dari kondisi berikut terpenuhi:
   - Semua data dalam subset termasuk dalam kelas yang sama.
   - Tidak ada fitur yang tersisa untuk dibagi.
   - Kedalaman maksimum pohon tercapai (untuk menghindari overfitting).

3. Membuat daun (leaf)

   Setiap daun pada pohon keputusan mewakili kelas akhir. Jika pohon mencapai kondisi berhenti, daun tersebut akan diberi label kelas berdasarkan mayoritas kelas dalam subset data yang mencapai daun tersebut.

4. Klasifikasi data baru

   Untuk mengklasifikasi data baru, decision tree mengikuti cabang pohon berdasarkan nilai fitur dari data tersebut. Proses ini berlanjut hingga mencapai daun, yang memberikan prediksi kelas untuk data baru.

5. Mengatasi overfitting

   Decision tree cenderung overfit pada data pelatihan, terutama jika pohon terlalu dalam. Untuk mengatasi ini, teknik seperti pruning (memangkas cabang yang tidak perlu) dan menetapkan kedalam maksimum pohon dapat digunakan.

6. Evaluasi

Rizqika Mulia Pratama
13522126

Setelah model dilatih, performanya dievaluasi menggunakan metrik seperti akurasi, precision, recall, dan F1-score pada data uji. Jika hasilnya tidak memuaskan, maka bisa dilakukan parameter tuning.

2. Perbandingan model buatan sendiri dan library

Berdasarkan hasil evaluasi, kedua model memiliki nilai Accuracy yang sama, yaitu 0.85. Ini menunjukkan bahwa kedua model memiliki performa yang setara dalam metrik yang diukur.

3. Improvement yang bisa dilakukan

- Optimasi Hyperparameter

  Gunakan teknik seperti Grid Search atau Random Search untuk menemukan kombinasi hyperparameter yang optimal, seperti max_depth, min_samples_split, min_samples_leaf, dan max_features

- Criterion for splitting

  Selain Gini impurity, dapat dipertimbangkan untuk menggunakan entropy sebagai kriteria pemisahan untuk melihat apakah itu memberikan hasil yang lebih baik.

- Pruning

  Implementasikan pruning (pre-pruning atau post-pruning) untuk menghindari overfitting. Pre-pruning dapat dilakukan dengan menetapkan batasan seperti max_depth, min_samples_split, dan min_samples_leaf. Post-pruning dapat dilakukan dengan memangkas cabang yang tidak memberikan kontribusi signifikan terhadap akurasi.