

Nicht-lineare Datenstrukturen

Der ID3-Algorithmus

Kopieren sie sich das Projekt 06-ID3-Algorithmus aus dem Tauschordner und öffnen sie es in BlueJ.

Aufgabe 1

Erstellen sie eine Instanz des `Entscheiders` und führen sie die Methode `ladeDatenUndZaehle()` aus. Vergleichen sie die angezeigten Daten mit der berechneten *Entropie* und dem *Informationsgewinn* der Attribute.

Aufgabe 2

Führen sie die Methode `ladeDatenUndErstelleBaum()` auf dem Objekt aus. Der Entscheidungsbaum wird nun rekursiv aus den Trainingsdaten aufgebaut.

Skizzieren sie den entstandenen Baum, indem sie mit dem Inspektor von BlueJ die Objekte erkunden.



Aufgabe 3

Die Trainingsdaten werden aus der Datei „`titanic50.csv`“ im Projektordner geladen. Der Dateiname wird im `Entscheider` in der Klassenvariablen `TRAININGSDATEN` festgelegt. Statt 50 Passagierdaten sind auch Datensätze mit 300 („`titanic300.csv`“) und 800 („`titanic800.csv`“) vorhanden.

Erstellen sie den Baum mit einem der anderen Trainingsdatensätze und vergleichen sie das Ergebnis mit dem ersten Baum.

① Zum Vergleich können sie die Methode `pruefeTestdaten()` verwenden, um einen Satz unbekannter Passagierdaten zu laden und durch den Entscheidungsbaum klassifizieren zu lassen.

Aufgabe 4

Mit größeren Datensätzen werden die Bäume feiner ausdifferenziert und lassen sich durch den Inspektor nur umständlich betrachten.

Implementieren sie die Methode `baumAusgeben(BinaryTree<Knoten> pBaum, int pTiefe)` wie folgt: Die Methode läuft rekursiv durch den Baum. Der Inhalt des Knotens wird ausgegeben, dann wird die Methode rekursiv auf dem linken, dann auf dem rechten Teilbaum aufgerufen. Die Rekursion bricht ab, wenn ein Blatt erreicht ist (linker und rechter Teilbaum gleich `null`).

Gestartet wird die Ausgabe mit der Methode `baumAusgeben()`.

Aufgabe 5

Die maximale Anzahl an Entscheidungen im Baum kann beim instanziiieren des `Entscheiders` über den Parameter `pMaximaleTiefe` festgelegt werden. Testen sie verschiedene Werte für den Parameter und lassen sie sich den entstehenden Baum mit `baumAusgeben()` anzeigen.

Aufgabe 6

Das rekursive Durchlaufen des Baumes, wie in der Methode `baumAusgeben()`, nennt man *Traversierung*.

Lesen sie im Buch den Abschnitt 5.2 (ab Seite 145) über Binärbäume - insbesondere über die verschiedenen Arten der Traversierung. Bearbeiten sie dann auf Seite 150 Aufgabe 3 a) und b). (b) in Pseudocode).