# 29. Warum sollte man einen Entscheidungsbaum kürzen (=„pruning“)? Gehen Sie auf Pre- und Post-Pruning Algorithmen ein.

Falsche Attributwerte oder Klassenzugehörigkeiten, die die Daten verfälschen, vergrößern den Entscheidungsbaum. Durch Pruning werden die unnötigen Sub-Bäume gekürzt und dadurch die Größe entscheidend verringert und die Klassifizierungsgenauigkeit ungesehener Objekte verbessert.

Beim Post Pruning werden Knoten und Teilbäume durch Blätter ersetzt.

# 30. Was ist die grundlegende Idee von Regression Trees? Was wird wie partitioniert? Warum heißt es "Recursive Partitioning"? Was haben Elemente innerhalb einer Partition gemeinsam? Was ist der Unterschied (im Prinzip, in der Vorgehensweise) zwischen Regression und Decision Tree?

# 31. Wie bildet man Klassen für k-NN Algorithmus und wie funktioniert er? Wie werden unklassifizierte Examples klassifiziert? Worauf ist zu achten wenn man k klein oder k groß ansetzt? Welche Ansätze gibt es, um den k-NN Algorithmus zu verbessern?

# 32. Was versteht man unter k-fold cross validation? Was versteht man unter data splitting?

# 33. Wozu gibt es AOP im Allgemeinen? Was wird dadurch gelöst, was in "herkömmlichem" OOP nur mit sehr viel Aufwand verbunden wäre.

Aspekt orientierte Programmierung (AOP) ist ein Programmierparadigma für die objektorientierte Programmierung, um generische Funktionalitäten über mehrere Klassen hinweg zu verwenden (**Cross-Cutting Concern**). Beispiele für solche Cross-Cutting Concerns sind: Security, logging, caching, etc.

Mit AOP wird die Trennung von **Core-Level-Concerns** (Businesslogic 🡪 Anforderungen an ein Programm 🡪 funktionalen Anforderungen) und **System-Level-Concerns** (betreffen das gesamte System/ technische Randbedingungen 🡪 Logging, Security, etc) ermöglicht, welches durch die herkömmliche OOP nur mit viel Aufwand gelöst werden kann.

Die beiden Teile Core-Level-Concerns und System-Level-Concerns sind miteinander „verwoben“. Die Core-Level-Concerns kann man als Komponenten bezeichnen und die System-Level-Concerns sind die Aspekte. Core-Level-Concerns werden überlicherweise als Module oder Objekte implementiert. Für die Aspekte gab es vor der aspektorientierten Programmierung keine elegante Lösung.

Das Problem der miteinander verwobenen Anforderungen wird auch als Cross-Cutting Concerns bezeichnet, denn sie „schneiden“ quer durch alle logischen Schichten des Systems. AOP ist das Werkzeug, um die logisch unabhängigen Belange auch physisch voneinander zu trennen. Dabei wird angestrebt, Code zu erzeugen, der besser wartbar und wiederverwendbar ist.