

Data-Science

Ziele

- Klasse ReadData: Einführung der Dateiformate HTML sowie XML und Anpassung der Datenspeicherung
- Dataflow: Datensätze beschreiben sowie interpretieren

Design

Klassen

- Klasse ReadData: Auslesen der Daten unter Verwendung von Mutexen und Zugriff auf den Cache
- Klasse Statistik: Statische Erfassung zu den Informationen aus dem Datensatz
- Klasse Logging: Überwachung und Monitoring vom System

Ordnerstruktur

- utils: Unterordner von inc bestehend aus den Header-Dateien für ReadData, Statistik und Logging
- tests: Unterordner von inc bestehend aus den Test-Dateien für die Klassen ReadData, Statistik und Logging
- src: Ordner mit den Source-Dateien für ReadData, Logging und Statistik
- build: Generierten Executables
- cpp_tests: Ordner zum Testen der Source-Dateien
- python_tests: Ordner zum Testen der Python-C++-Schnittstelle
- data: Datensätze zum Üben
- .github: Ordner zur Konfiguration vom CI
- bindings: Schnittstelle in C++ für die Integration in pybind11
- pybind11: Schnittstelle zur Integration von Code in C++ in Python

Weiterentwicklung der Klasse ReadData

- neue Dateiformate: HTML sowie XML
- Refactoring: Wiederkehrende Aufrufe in Funktionen, wie z.B. ReadCSV, ReadJSON und co. wurden in separate Funktionen untergliedert
- Speichermanagement: Daten werden nun tatsächlich im Cache abgelegt und der Cache kann wieder bereinigt werden
- Darstellung der Daten: Die Funktion DisplayData gibt eine Liste zurück, worin jedes Element einer Zeile vom Datensatz entspricht

Verbesserung vom CI/CD

- .pre-commit-config.yaml: vor jedem Commit erfolgt ein Style-Check