Software Engineering in Machine Learning und Data Science

Falco Winkler

June 24, 2019

Outline

Software Engineering

Definition

"The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software"—IEEE Standard Glossary of Software Engineering

Prinzipien

- SOLID
- Test Driven Development
- Design Patterns
- Continous Integration
- Refactoring

[3]

Mangelnde SE-Disziplin in Data Science

- Wissenschaftliche Methoden um Wissen aus Daten zu extrahieren
- These -> Überprüfung -> Neue Theorie
- Best Practices jedoch häufig vernachlässigt

[2]

Verschwimmende Systemgrenzen

- Modulares Design und Kapselung führen zu wartbarem Code
- Machine Learning wird angewendet wenn das gewünschte Verhalten schwer in Code ohne Abhängigkeiten auf externe Daten spezifiziert werden kann
- Systemverhalten hängt von Eigenschaften externer Daten ab

[4]

Entanglement / starke Kopplung

- Wenn ein Feature verändert wird, ändert sich das ganze Modell
- *CACE*: changing anything changes everything

[4]

Versteckte Feedback-Schleifen

- Modell verändert seine zukünftigen Trainingsdaten

Unbekannte Konsumenten

- Unvorhergesehene Änderungen des Systemverhaltens brechen Konsumenten
- Ein Konsument kann das Modell durch Feedbackschleife beeinflussen

[4]

Datenabhängigkeiten

- Datenabhängigkeiten sind schwer zu finden, und tragen viel zur Systemkomplexität bei
- Instabile Datenabhängigkeiten
- Unbenutzte Features
- sog. Korrektionskaskaden

[4]

Antipatterns auf Systemlevel

- "Glue code" - vor Allem bei general purpose frameworks
- "pipeline jungles"
- experimenteller Code
- Verteilte Konfigurationen

[4]

Features und Rohdaten

- Features auf erwartete Werte und Verteilungen testen
- Integration und Unit-Tests für ML Code
- Kosten und Korrelationen von Features testen
- Feature - Pipeline Code testen
- Featureextraktion und Training trennen

[1]

Model - Entwicklung

- Source code für das ML Artefakt nach bekannten Prinzipien handhaben
- Ältere Modelle auf neuen Daten testen
- Gegen einfaches Modell testen

[1]

Training

- CI
- Reproduzierbares Training
- Unit Tests für Code, Integration tests für Pipeline
- Rollback, Canary Deployments
- Nur Modelle mit guten Metriken deployen

[1]

Monitoring

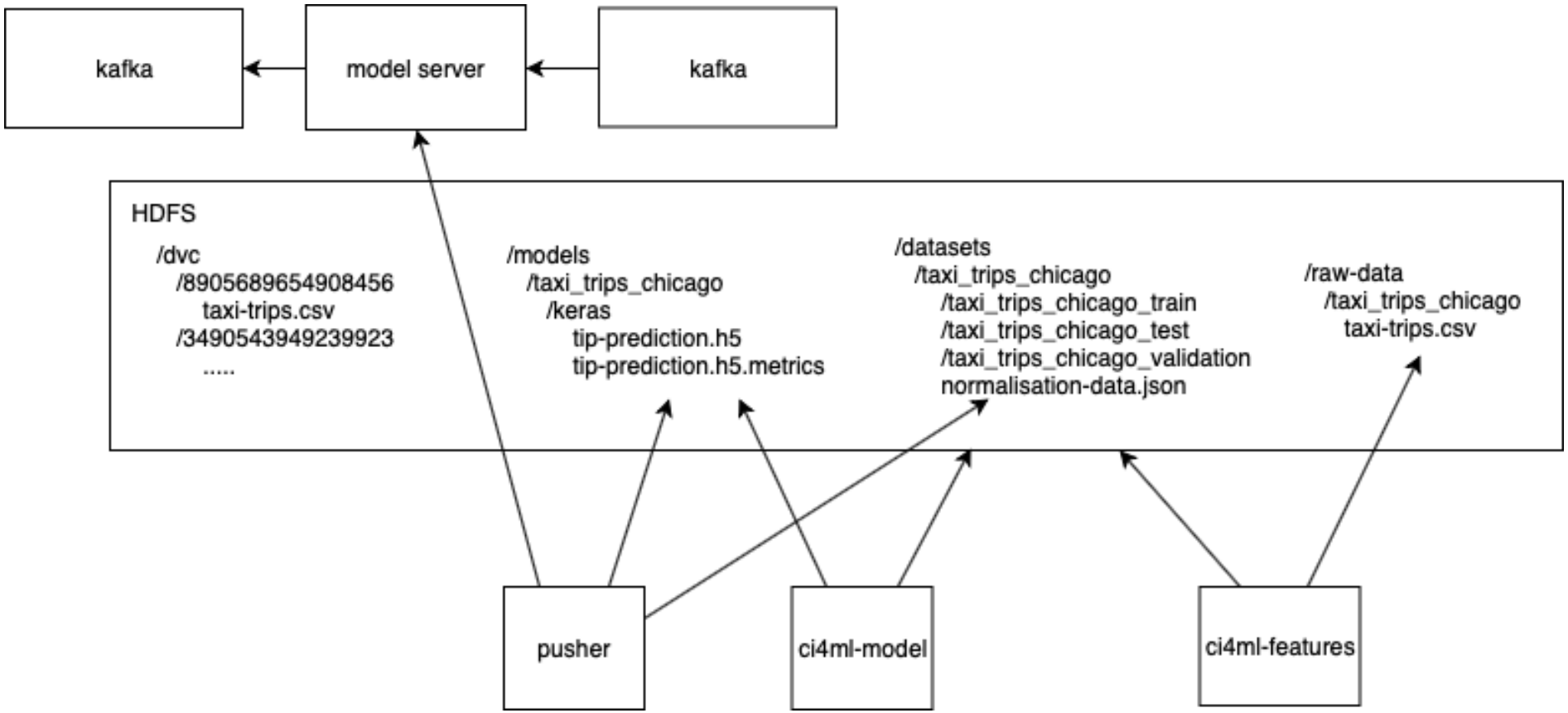
- Trainings mit live-Daten vergleichen
- Datenquellen auf Inhalte prüfen
- Modellausgaben und Kernmetriken der Eingabedaten tracken

[1]

Bezug auf Masterarbeit und Projekte

- Grundprojekt: Big Data Systemachitekturen
- Hauptprojekt: CI4ML - Beispielimplementierung eines continously integrated ML-Models
- Masterthesis: CI4ML - Platform ?

Hauptprojekt




Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley.
What's your ml test score? a rubric for ml production systems.
2016.


M. Kirk.
Thoughtful Machine Learning with Python: A Test-driven Approach.
O'Reilly, 2017.


Robert C. Martin.
Clean Code: A Handbook of Agile Software Craftsmanship.
Prentice Hall PTR, Upper Saddle River, NJ, USA, 1 edition, 2008.


D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young.
Machine learning: The high interest credit card of technical debt.
In *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*, 2014.