

Responsible AI in software development (RAIISD)

ID RAIISD Preis auf Anfrage Dauer 1 Tag

Zielgruppe

Alle Personen, die an der Nutzung von GenAl oder der Entwicklung von maschinellem Lernen beteiligt sind

Voraussetzungen

Allgemeine Entwicklung

Kursziele

- Verschiedene Aspekte der verantwortungsvollen Al verstehen
- Wie man generative KI verantwortungsvoll in der Softwareentwicklung einsetzt
- Schnelles Engineering für optimale Ergebnisse
- Wie man generative KI im gesamten SDLC einsetzt

Kursinhalt

Eine kurze Geschichte der Künstlichen Intelligenz

- Die Ursprünge der KI
- Neuronale Netze und "Wahrscheinlichkeitsmaschinen"
- Frühe ML-Codierungstools
- Die KI-Codierrevolution der 2020er Jahre
- Bedrohungen für ML-Systeme

Verantwortungsvolle AI

- Was ist verantwortungsvolle KI?
- Rechenschaftspflicht und Transparenz
- Verringerung schädlicher Verzerrungen
- Gültigkeit und Zuverlässigkeit
- Gültigkeit und Zuverlässigkeit Nicht-Determinismus des Codes
- Demonstration Experimentieren mit Gültigkeit und Zuverlässigkeit in Copilot
- Erklärbarkeit und Interpretierbarkeit
- Sicherheit, Schutz, Privatsphäre und Widerstandsfähigkeit
- Sicherheit und verantwortungsvolle KI in der Softwareentwicklung

GenAl verantwortungsvoll in der Softwareentwicklung einsetzen

- Grundlagen der LLM-Codeerzeugung
- Grundlegende Bausteine und Konzepte
- Eingabeaufforderung für Vorlagen
- Systemaufforderungen zur KI-gesteuerten Codierung
- GenAl-Werkzeuge f
 ür die Kodierung: Copilot, Codeium und andere
- Kann Kl... Ihre Produktivität steigern?
- Kann Kl... die "langweiligen Teile" übernehmen?
- Kann Al... gründlicher sein?
- Überprüfung des generierten Codes der Blackbox-Blues
- Die Gefahr von Halluzinationen
- Kann Kl... dir beibringen, wie man (besser) programmiert?
- Demonstration Experimentieren mit einer unbekannten API in Copilot
- Die Auswirkungen von GenAl auf die Programmierkenntnisse
- Einige weitere langfristige Auswirkungen der Nutzung von GenAl
- · Wo die KI-Codegenerierung nicht gut abschneidet
- Schnelles Engineering
 - Warum ist ein guter Souffleur so wichtig?
 - Schaffung des Kontexts für generative KI
 - Null-Schuss-, Ein-Schuss- und Wenig-Schuss-Eingabeaufforderung
 - Vernunftbasiertes Prompt-Engineering, Gedankenkette
 - Demonstration Experimentieren mit Eingabeaufforderungen in Copilot
 - o Durchsetzung und Einhaltung von Token-Limits
 - Aufforderungsmuster
 - Prompt-Muster und Prompt-Priming
 - Die 6 Kategorien von Aufforderungsmustern
 - Aufforderungsmuster: Meta-Sprache erstellen
 - Aufforderungsmuster: Persona
 - Aufforderungsmuster: Visualisierungs-Generator
 - Aufforderungsmuster: Faktencheck-Liste
 - Aufforderungsmuster: Alternative Lösungsansätze
 - Aufforderungsmuster: Verweigerungsbrecher
 - Aufforderungsmuster: Umgekehrte Interaktion

Responsible AI in software development (RAIISD)

- Aufforderungsmuster: Kontext-Manager
- o Einige weitere Souffleur-Ansätze
 - Least-to-Most und Self-Planning: Zerlegung komplexer Aufgaben
 - Demonstration Aufgabenzerlegung mit Copilot
 - Prompt-Engineering-Techniken für Verfeinerung und Iteration
 - Einheitstests, TDD und GenAl
 - Demonstration Testbasierte Codegenerierung mit Copilot

Integration von generativer KI in den SDLC

- Einsatz von GenAl über die Codegenerierung hinaus
- Einsatz von Al bei der Anforderungsspezifikation
- Aufforderungsmuster für die Erfassung von Anforderungen
- · Softwareentwicklung und KI
- Prompt-Muster für den Softwareentwurf
- Demonstration Anforderungserfassung und API-Design mit Copilot
- Einsatz von Al bei der Umsetzung
- Prompt-Muster für die Umsetzung
- Demonstration Auffinden versteckter Annahmen mit Copilot
- Einsatz von AI bei Tests und QA
- Einsatz von Al bei der Wartung
- Aufforderungsmuster für das Refactoring
- Demonstration Experimentieren mit Code-Refactoring in Copilot
- Aufforderungsmuster für die Simulation von Änderungsanträgen

Sicherheit von KI-generiertem Code

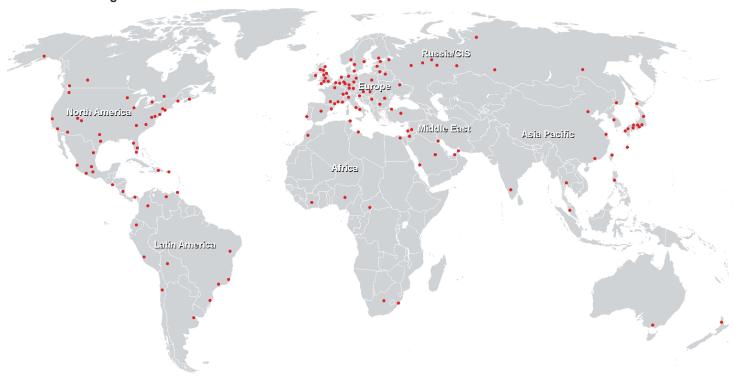
- Sicherheit von KI-generiertem Code
- Praktische Angriffe auf Tools zur Codegenerierung
- Abhängigkeits-Halluzination durch generative KI
- Fallstudie Eine Geschichte der Schwächen von GitHub Copilot (bis Mitte 2024)
- Ein Beispiel für eine Schwachstelle
 - Pfadüberquerung
 - Demonstration Pfadüberquerung
 - · Beispiele für die Pfadüberquerung
 - Bewährte Verfahren zur Pfadüberquerung
 - o Demonstration Kanonisierung von Pfaden
 - Demonstration Experimentieren mit der Pfadverfolgung in Copilot

Zusammenfassung und Schlussfolgerungen

- Verantwortungsvolle KI-Prinzipien in der Softwareentwicklung
- Generative AI Ressourcen und zusätzliche Anleitungen

Responsible AI in software development (RAIISD)

Weltweite Trainingscenter





Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3 CH-8304 Wallisellen Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch