




Informationen im Überblick

 Gute Kenntnisse der Python-Programmierung

 Forensiker*innen, Ermittler*innen, Data Scientists/Analysts, IT-Sicherheitsexpert*innen, Data Journalists, Social Media Manager, PR-Beratung, Financial Analysts

 4 Tage

 1500,-

 online

Veranstaltet durch




Referent*innen:

Lukas Graner,
wiss. Mitarbeiter,
Mediensicherheit
und IT-Forensik,
Fraunhofer SIT

Jeong-Eun Choi,
wiss. Mitarbeiterin,
Fraunhofer SIT

Inna Vogel,
wiss. Mitarbeiterin,
Mediensicherheit
und IT-Forensik,
Fraunhofer SIT

 Weitere Infos und aktuelle Termine buchen unter:

www.cybersicherheit.fraunhofer.de/textanalyse

Textdatenanalyse mit NLP und Maschinellem Lernen

»Schürfen« in Textdaten (Einführungskurs)

Die Herausforderung: Ein Großteil der weltweit verfügbaren Informationen liegt in Texten vor. Für viele Anwendungen ist die Suche nach relevanten Informationen in Textdaten wichtig, z.B. um nach Themen, Schreibstilen oder Sentiment zu klassifizieren oder den Verfasser eines Textes zu identifizieren. Unser Einführungskurs bietet Ihnen hierzu wichtige Vorkenntnisse des Natural Language Processing (NLP) und des maschinellen Lernens (ML) vermittelt, in der Theorie und als »Python«-Quellcode.

Inhalte des Seminars

Crawling von WWW-Seiten

Korpuserstellung

Datenbereinigung und Preprocessing

Grundlagen des ML

Interpretierbarkeit von ML-Modellen

**Statistische Evaluierung Ihrer Ergebnisse
Überblick über Anwendungen: Autor-**

schaftsanalyse, Topic Modeling, Sentiment-Analyse, Landessprachen-Erkennung

Livedemos zu den Kursthemen

Python-Programmierübungen als Jupyter-Notebooks

Intensive Online-Betreuung durch unsere Expertinnen und Experten

Musterlösungen zur Nachbereitung

Tipps zu Literatur und Software

Die Programmierübungen bearbeiten Sie online auf unserem Jupyter-Server (CoCalc-Umgebung) in Ihrem eigenen Browser. Keine Installation bei Ihnen benötigt!

Ihr Nutzen

- Nach dem Seminar verstehen Sie moderne Verfahren des NLP und ML und können diese in »Python« selbst implementieren.

