Privater Masterstudiengang

Künstliche Intelligenz in der Programmierung

tech technologische universität

tech technologische universität

Privater Masterstudiengang
Künstliche Intelligenz in der Programmierung

- Modalität: **online**
- Dauer: **12 Monate**
- Qualifizierung: **TECH Technologische Universität**
- Zeitplan: **in Ihrem eigenen Tempo**
- Prüfungen: **online**

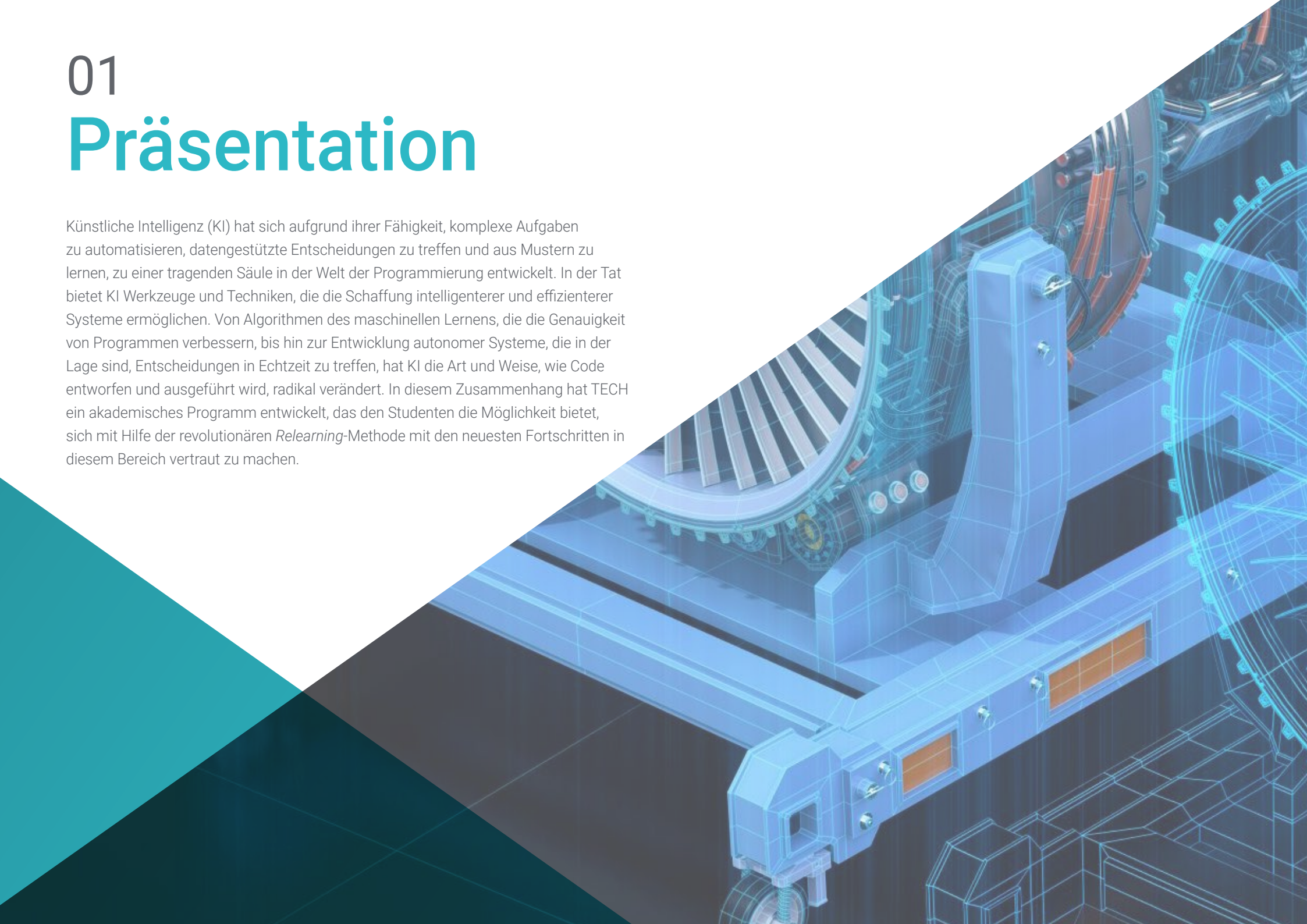
Internetzugang: www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-kunstliche-intelligenz-programmierung

Index

01

Präsentation

Künstliche Intelligenz (KI) hat sich aufgrund ihrer Fähigkeit, komplexe Aufgaben zu automatisieren, datengestützte Entscheidungen zu treffen und aus Mustern zu lernen, zu einer tragenden Säule in der Welt der Programmierung entwickelt. In der Tat bietet KI Werkzeuge und Techniken, die die Schaffung intelligenterer und effizienterer Systeme ermöglichen. Von Algorithmen des maschinellen Lernens, die die Genauigkeit von Programmen verbessern, bis hin zur Entwicklung autonomer Systeme, die in der Lage sind, Entscheidungen in Echtzeit zu treffen, hat KI die Art und Weise, wie Code entworfen und ausgeführt wird, radikal verändert. In diesem Zusammenhang hat TECH ein akademisches Programm entwickelt, das den Studenten die Möglichkeit bietet, sich mit Hilfe der revolutionären *Relearning*-Methode mit den neuesten Fortschritten in diesem Bereich vertraut zu machen.



Der Studiengang Künstliche Intelligenz in der Programmierung bietet Ihnen eine ganzheitliche Perspektive darauf, wie KI jede Phase der Softwareentwicklung beeinflusst und verbessert"

Die Bedeutung der künstlichen Intelligenz in der Programmierung liegt in ihrer Fähigkeit, Prozesse zu befähigen und zu automatisieren, die Softwareentwicklung zu optimieren und die Effizienz bei der Lösung komplexer Probleme zu verbessern. Ihre Fähigkeit, große Datenmengen zu analysieren und optimale Lösungen zu finden, hat zu bedeutenden Fortschritten in Bereichen wie der Optimierung von Algorithmen, der Schaffung von intuitiveren Schnittstellen und der Lösung komplexer Probleme in verschiedenen Bereichen geführt.

Aus diesem Grund hat TECH diesen privaten Masterstudiengang entwickelt, der sich als strategische Lösung zur Erweiterung der beruflichen Möglichkeiten und des Karrierewachstums von Informatikern erweist. Der Studiengang befasst sich mit der Verbesserung der Produktivität in der Softwareentwicklung durch KI und untersucht die Techniken und Werkzeuge, die Prozesse automatisieren, den Code optimieren und die Erstellung intelligenter Anwendungen beschleunigen.

Darüber hinaus wird sich das Programm auf die entscheidende Rolle der KI im Bereich *QA Testing* konzentrieren, wobei KI-Algorithmen und -Methoden zur Verbesserung der Qualität, Genauigkeit und Abdeckung von Tests sowie zur effizienteren Erkennung und Korrektur von Fehlern eingesetzt werden. Es wird auch die Integration von maschinellem Lernen und der Verarbeitung natürlicher Sprache in die Webentwicklung behandelt, um intelligente Websites zu erstellen, die sich anpassen und den Benutzern personalisierte Erfahrungen bieten.

Außerdem wird es sich mit KI-Techniken zur Verbesserung der Benutzerfreundlichkeit, der Interaktion und der Funktionalität mobiler Anwendungen befassen, um intelligente und vorausschauende Anwendungen zu schaffen, die sich an das Benutzerverhalten anpassen. Darüber hinaus wird ein eingehender Blick auf die KI-Softwarearchitektur geworfen, einschließlich der verschiedenen Modelle, die die Integration von KI-Algorithmen und deren Einsatz in Produktionsumgebungen erleichtern.

Mit dem Ziel, hochkompetente KI-Spezialisten fortzubilden, hat TECH ein umfassendes Programm konzipiert, das auf der exklusiven *Relearning*-Methode basiert.
Dieser Ansatz ermöglicht es den Studenten, ihr Verständnis durch Wiederholung grundlegender Konzepte zu festigen.

Dieser **Privater Masterstudiengang in Künstliche Intelligenz in der Programmierung** enthält das vollständigste und aktuellste Bildungsprogramm auf dem Markt.
Die hervorstechendsten Merkmale sind:

- Die Entwicklung von Fallstudien, die von Experten für künstliche Intelligenz in der Programmierung vorgestellt werden
- Der grafischen, schematischen und äußerst praktischen Inhalte bieten wissenschaftliche und praktische Informationen zu den Disziplinen, die für die berufliche Praxis unerlässlich sind
- Praktische Übungen, anhand derer der Selbstbewertungsprozess zur Verbesserung des Lernens verwendet werden kann
- Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- Theoretische Lektionen, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- Die Verfügbarkeit des Zugriffs auf die Inhalte von jedem festen oder tragbaren Gerät mit Internetanschluss

Sie werden innovative Projekte leiten, die den Anforderungen eines sich ständig weiterentwickelnden Technologiemarktes entsprechen. Worauf warten Sie, um sich einzuschreiben?"

Dank innovativer Multimedia-Ressourcen werden Sie in die Grundlagen der Software-Architektur eintauchen, einschließlich Leistung, Skalierbarkeit und Wartbarkeit“

Möchten Sie sich auf künstliche Intelligenz spezialisieren? Mit diesem Programm werden Sie die Optimierung des Bereitstellungsprozesses und die Integration von KI in Cloud Computing beherrschen.

Sie werden die Integration von KI-Elementen in Visual Studio Code und die Code-Optimierung mit ChatGPT in einem umfassenden akademischen Programm erlernen.

Das Dozententeam des Programms besteht aus Experten des Sektors, die ihre Berufserfahrung in diese Fortbildung einbringen, sowie aus renommierten Fachkräften von führenden Gesellschaften und angesehenen Universitäten.

Die multimedialen Inhalte, die mit der neuesten Bildungstechnologie entwickelt wurden, werden der Fachkraft ein situiertes und kontextbezogenes Lernen ermöglichen, d. h. eine simulierte Umgebung, die eine immersive Fortbildung bietet, die auf die Ausführung von realen Situationen ausgerichtet ist.

Das Konzept dieses Programms konzentriert sich auf problemorientiertes Lernen, bei dem die Fachkraft versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Zu diesem Zweck wird sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

02

Ziele

Das Hauptziel dieses Programms besteht darin, Fachleuten Zugang zu den neuesten Erkenntnissen in diesem Bereich zu verschaffen, wobei ein Ansatz verfolgt wird, der ihre umfassende Fortbildung fördert. So werden sie die Möglichkeit haben, an einem exklusiven und vollständig online durchgeführten akademischen Weg teilzunehmen. Die Teilnehmer werden mit nützlichen Spitzenkenntnissen ausgestattet, die von der KI-gestützten Softwareentwicklung bis hin zum Entwurf und der Ausführung von Webprojekten und mobilen Anwendungen mit Intelligenz und Anpassungsfähigkeit reichen. Mit diesem Programm wird der Informatiker die Grenzen der herkömmlichen Programmierung überschreiten und ein aktiver Akteur der technologischen Revolution werden.

Dank TECH werden Sie den gesamten Lebenszyklus des Testens von der Erstellung von Testfällen bis zur Entdeckung von Fehlern bewältigen"

Allgemeine Ziele

- Entwickeln von Fähigkeiten zur Einrichtung und Verwaltung effizienter Entwicklungsumgebungen, um eine solide Grundlage für die Umsetzung von KI-Projekten zu schaffen
- Erwerben von Kenntnissen über die Planung, Durchführung und Automatisierung von Qualitätstests unter Einbeziehung von KI-Tools zur Erkennung und Korrektur von Fehlern
- Verstehen und Anwenden von Grundsätzen der Leistung, Skalierbarkeit und Wartbarkeit bei der Entwicklung von Großrechnersystemen
- Kennenlernen der wichtigsten Entwurfsmuster und deren effektive Anwendung in der Softwarearchitektur

Spezifische Ziele

Modul 1. Grundlagen der künstlichen Intelligenz

- Analysieren der historischen Entwicklung der künstlichen Intelligenz, von ihren Anfängen bis zu ihrem heutigen Stand, Identifizierung der wichtigsten Meilensteine und Entwicklungen
- Verstehen der Funktionsweise von neuronalen Netzen und ihrer Anwendung in Lernmodellen der künstlichen Intelligenz
- Untersuchen der Prinzipien und Anwendungen von genetischen Algorithmen und analysieren ihren Nutzen bei der Lösung komplexer Probleme
- Analysieren der Bedeutung von Thesauri, Vokabularen und Taxonomien bei der Strukturierung und Verarbeitung von Daten für KI-Systeme
- Erforschen des Konzepts des semantischen Webs und seines Einflusses auf die Organisation und das Verständnis von Informationen in digitalen Umgebungen

Modul 2. Datentypen und Datenlebenszyklus

- Verstehen der grundlegenden Konzepte der Statistik und ihrer Anwendung in der Datenanalyse
- Identifizieren und Klassifizieren der verschiedenen Arten von statistischen Daten, von quantitativen bis zu qualitativen Daten
- Analysieren des Lebenszyklus von Daten, von der Erzeugung bis zur Entsorgung, und Identifizieren der wichtigsten Phasen
- Erkunden der ersten Phasen des Lebenszyklus von Daten, wobei die Bedeutung der Datenplanung und der Datenstruktur hervorgehoben wird
- Untersuchen der Prozesse der Datenerfassung, einschließlich Methodik, Tools und Erfassungskanäle
- Untersuchen des *Datawarehouse*-Konzepts mit Schwerpunkt auf den Elementen des *Datawarehouse* und seinem Design
- Analysieren der rechtlichen Aspekte im Zusammenhang mit der Datenverwaltung, der Einhaltung von Datenschutz- und Sicherheitsvorschriften sowie von *Best Practices*

Modul 3. Daten in der künstlichen Intelligenz

- Beherrschen der Grundlagen der Datenwissenschaft, einschließlich der Werkzeuge, Typen und Quellen für die Informationsanalyse
- Erforschen des Prozesses der Umwandlung von Daten in Informationen mithilfe von *Data Mining* und Datenvisualisierungstechniken
- Studieren der Struktur und der Eigenschaften von *Datasets* und verstehen ihrer Bedeutung für die Aufbereitung und Nutzung von Daten für KI-Modelle
- Analysieren von überwachten und unüberwachten Modellen, einschließlich Methoden und Klassifizierung
- Verwenden spezifischer Tools und bewährter Verfahren für die Datenverarbeitung, um Effizienz und Qualität bei der Implementierung von künstlicher Intelligenz zu gewährleisten

Modul 4. *Data Mining*. Auswahl, Vorverarbeitung und Transformation

- Beherrschen statistischer Inferenztechniken, um statistische Methoden im *Data Mining* zu verstehen und anzuwenden
- Durchführen detaillierter explorativer Analysen von Datensätzen, um relevante Muster, Anomalien und Trends zu erkennen
- Entwickeln von Fähigkeiten zur Datenaufbereitung, einschließlich Datenbereinigung, -integration und -formatierung für die Verwendung im *Data Mining*
- Implementieren effektiver Strategien für den Umgang mit fehlenden Werten in Datensätzen, indem je nach Kontext Imputations- oder Eliminierungsmethoden angewendet werden
- Identifizieren und Entschärfen von Datenrauschen, durch Anwendung von Filter- und Glättungsverfahren, um die Qualität des Datensatzes zu verbessern
- Eingehen auf die Datenvorverarbeitung in *Big-Data*-Umgebungen

Modul 5. Algorithmik und Komplexität in der künstlichen Intelligenz

- Einführen von Algorithmenentwurfsstrategien, die ein solides Verständnis der grundlegenden Ansätze zur Problemlösung vermitteln
- Analysieren der Effizienz und Komplexität von Algorithmen unter Anwendung von Analysetechniken zur Bewertung der Leistung in Bezug auf Zeit und Raum
- Untersuchen und Anwenden von Sortieralgorithmen, Verstehen ihrer Leistung und Vergleichen ihrer Effizienz in verschiedenen Kontexten
- Erforschen von baumbasierten Algorithmen, Verstehen ihrer Struktur und Anwendungen
- Untersuchen von Algorithmen mit *Heaps*, Analysieren ihrer Implementierung und ihrer Nützlichkeit bei der effizienten Datenmanipulation
- Analysieren graphenbasierter Algorithmen, wobei ihre Anwendung bei der Darstellung und Lösung von Problemen mit komplexen Beziehungen untersucht wird
- Untersuchen von *Greedy*-Algorithmen, Verständnis ihrer Logik und Anwendungen bei der Lösung von Optimierungsproblemen
- Untersuchen und Anwenden der *Backtracking*-Technik für die systematische Problemlösung und Analysieren ihrer Effektivität in verschiedenen Szenarien

Modul 6. Intelligente Systeme

- Erforschen der Agententheorie, Verstehen der grundlegenden Konzepte ihrer Funktionsweise und ihrer Anwendung in der künstlichen Intelligenz und im *Software Engineering*
- Studieren der Darstellung von Wissen, einschließlich der Analyse von Ontologien und deren Anwendung bei der Organisation von strukturierten Informationen
- Analysieren des Konzepts des semantischen Webs und seiner Auswirkungen auf die Organisation und den Abruf von Informationen in digitalen Umgebungen
- Evaluieren und Vergleichen verschiedener Wissensrepräsentationen und deren Integration zur Verbesserung der Effizienz und Genauigkeit von intelligenten Systemen
- Studieren semantischer *Reasoner*, wissensbasierter Systeme und Expertensysteme und Verstehen ihrer Funktionalität und Anwendungen in der intelligenten Entscheidungsfindung

Modul 7. Maschinelles Lernen und *Data Mining*

- Einführen in die Prozesse der Wissensentdeckung und in die grundlegenden Konzepte des maschinellen Lernens
- Untersuchen von Entscheidungsbäumen als überwachte Lernmodelle, Verstehen ihrer Struktur und Anwendungen
- Bewerten von Klassifikatoren anhand spezifischer Techniken, um ihre Leistung und Genauigkeit bei der Datenklassifizierung zu messen
- Studieren neuronaler Netze und Verstehen ihrer Funktionsweise und Architektur, um komplexe Probleme des maschinellen Lernens zu lösen
- Erforschen von Bayes'schen Methoden und deren Anwendung im maschinellen Lernen, einschließlich Bayes'scher Netzwerke und Bayes'scher Klassifikatoren
- Analysieren von Regressions- und kontinuierlichen Antwortmodellen zur Vorhersage von numerischen Werten aus Daten
- Untersuchen von Techniken zum *Clustering*, um Muster und Strukturen in unmarkierten Datensätzen zu erkennen
- Erforschen von *Text Mining* und natürlicher Sprachverarbeitung (NLP), um zu verstehen, wie maschinelle Lerntechniken zur Analyse und zum Verständnis von Texten eingesetzt werden

Modul 8. Neuronale Netze, die Grundlage von *Deep Learning*

- Beherrschen der Grundlagen des tiefen Lernens und Verstehen seiner wesentlichen Rolle beim *Deep Learning*
- Erkunden der grundlegenden Operationen in neuronalen Netzen und Verstehen ihrer Anwendung bei der Konstruktion von Modellen
- Analysieren der verschiedenen Schichten, die in neuronalen Netzen verwendet werden, und lernen, wie man sie richtig auswählt
- Verstehen der effektiven Verknüpfung von Schichten und Operationen, um komplexe und effiziente neuronale Netzarchitekturen zu entwerfen
- Verwenden von Trainern und Optimierern, um die Leistung von neuronalen Netzen abzustimmen und zu verbessern
- Erforschen der Verbindung zwischen biologischen und künstlichen Neuronen für ein tieferes Verständnis des Modelldesigns
- Feinabstimmen von Hyperparametern für das *Fine Tuning* neuronaler Netze, um ihre Leistung bei bestimmten Aufgaben zu optimieren

Modul 9. Training Tiefer Neuronaler Netze

- Lösen von Problemen im Zusammenhang mit Gradienten beim Training von tiefen neuronalen Netzen
- Erforschen und Anwenden verschiedener Optimierer, um die Effizienz und Konvergenz von Modellen zu verbessern
- Programmieren der Lernrate zur dynamischen Anpassung der Konvergenzrate des Modells
- Verstehen und Bewältigen von *Overfitting* durch spezifische Strategien beim Training
- Anwenden praktischer Richtlinien, um ein effizientes und effektives Training von tiefen neuronalen Netzen zu gewährleisten
- Implementieren von *Transfer Learning* als fortgeschrittene Technik zur Verbesserung der Modellleistung bei bestimmten Aufgaben
- Erforschen und Anwenden von Techniken der *Data Augmentation* zur Anreicherung von Datensätzen und Verbesserung der Modellgeneralisierung
- Entwickeln praktischer Anwendungen mit *Transfer Learning* zur Lösung realer Probleme
- Verstehen und Anwenden von Regularisierungstechniken zur Verbesserung der Generalisierung und zur Vermeidung von *Overfitting* in tiefen neuronalen Netzen

Modul 10. Anpassung von Modellen und Training mit *TensorFlow*

- Beherrschen der Grundlagen von *TensorFlow* und seiner Integration mit NumPy für effiziente Datenverwaltung und Berechnungen
- Anpassen von Modellen und Trainingsalgorithmen mit den fortgeschrittenen Fähigkeiten von *TensorFlow*
- Erforschen der tfdata-API zur effektiven Verwaltung und Manipulation von Datensätzen
- Implementieren des Formats TFRecord, um große Datensätze in *TensorFlow* zu speichern und darauf zuzugreifen
- Verwenden von Keras-Vorverarbeitungsschichten zur Erleichterung der Konstruktion eigener Modelle
- Erforschen des *TensorFlow Datasets*-Projekts, um auf vordefinierte Datensätze zuzugreifen und die Entwicklungseffizienz zu verbessern
- Entwickeln einer *Deep Learning*-Anwendung mit *TensorFlow* unter Einbeziehung der im Modul erworbenen Kenntnisse
- Anwenden aller Konzepte, die bei der Erstellung und dem Training von benutzerdefinierten Modellen mit *TensorFlow* erlernt wurden, auf praktische Art und Weise in realen Situationen

Modul 11. *Deep Computer Vision* mit *Convolutional Neural Networks*

- Verstehen der Architektur des visuellen Kortex und ihrer Bedeutung für *Deep Computer Vision*
- Erforschen und Anwenden von Faltungsschichten, um wichtige Merkmale aus Bildern zu extrahieren
- Implementieren von Clustering-Schichten und ihre Verwendung in *Deep Computer Vision*-Modellen mit Keras
- Analysieren verschiedener Architekturen von *Convolutional Neural Networks* (CNN) und deren Anwendbarkeit in verschiedenen Kontexten
- Entwickeln und Implementieren eines CNN ResNet unter Verwendung der Keras-Bibliothek, um die Effizienz und Leistung des Modells zu verbessern
- Verwenden von vorab trainierten Keras-Modellen, um das Transfer-Lernen für bestimmte Aufgaben zu nutzen
- Anwenden von Klassifizierungs- und Lokalisierungstechniken in *Deep Computer Vision*-Umgebungen
- Erforschen von Strategien zur Objekterkennung und -verfolgung mit *Convolutional Neural Networks*
- Implementieren von semantischen Segmentierungstechniken, um Objekte in Bildern im Detail zu verstehen und zu klassifizieren

Modul 12. Natürliche Sprachverarbeitung (NLP) mit rekurrenten neuronalen Netzen (RNN) und Aufmerksamkeit

- Entwickeln von Fähigkeiten zur Texterstellung mit rekurrenten neuronalen Netzen (RNN)
- Anwenden von RNNs bei der Meinungsklassifizierung zur Stimmungsanalyse in Texten
- Verstehen und Anwenden von Aufmerksamkeitsmechanismen in Modellen zur Verarbeitung natürlicher Sprache
- Analysieren und Verwenden von *Transformers*-Modellen in spezifischen NLP-Aufgaben
- Erkunden der Anwendung von *Transformers*-Modellen im Kontext von Bildverarbeitung und Computer Vision
- Kennenlernen der *Hugging Face Transformers*-Bibliothek für die effiziente Implementierung fortgeschrittener Modelle
- Vergleichen der verschiedenen *Transformers*-Bibliotheken, um ihre Eignung für bestimmte Aufgaben zu bewerten
- Entwickeln einer praktischen Anwendung von NLP, die RNN- und Aufmerksamkeitsmechanismen integriert, um reale Probleme zu lösen

Modul 13. *Autoencoder, GANs* und Diffusionsmodelle

- Entwickeln effizienter Datenrepräsentationen mit *Autoencodern, GANs* und Diffusionsmodellen
- Durchführen einer PCA unter Verwendung eines unvollständigen linearen *Autoencoders* zur Optimierung der Datendarstellung
- Implementieren und Verstehen der Funktionsweise von gestapelten *Autoencodern*
- Erforschen und Anwenden von *Convolutional Autoencoders* für effiziente visuelle Datendarstellungen
- Analysieren und Anwenden der Effektivität von *Sparse-Autoencodern* bei der Datendarstellung
- Generieren von Modebildern aus dem MNIST-Datensatz mit Hilfe von *Autoencoders*
- Verstehen des Konzepts der *Generative Adversarial Networks* (GANs) und Diffusionsmodelle
- Implementieren und Vergleichen der Leistung von Diffusionsmodellen und *GANs* bei der Datengenerierung

Modul 14. Bio-inspiriertes Computing

- Einführen in die grundlegenden Konzepte des bio-inspirierten Computings
- Erforschen sozialer Anpassungsalgorithmen als wichtiger Ansatz im bio-inspirierten Computing
- Analysieren von Strategien zur Erforschung und Ausnutzung des Raums in genetischen Algorithmen
- Untersuchen von Modellen des evolutionären Rechnens im Kontext der Optimierung
- Fortsetzen der detaillierten Analyse von Modellen des evolutionären Rechnens
- Anwenden der evolutionären Programmierung auf spezifische Lernprobleme
- Bewältigen der Komplexität von Multi-Objektiv-Problemen im Rahmen des bio-inspirierten Computings
- Erforschen der Anwendung von neuronalen Netzen im Bereich des bio-inspirierten Computings
- Vertiefen der Implementierung und des Nutzens von neuronalen Netzen im Bereich des bio-inspirierten Computings

Modul 15. Künstliche Intelligenz: Strategien und Anwendungen

- Entwickeln von Strategien für die Implementierung von künstlicher Intelligenz in Finanzdienstleistungen
- Analysieren der Auswirkungen von künstlicher Intelligenz auf die Erbringung von Dienstleistungen im Gesundheitswesen
- Identifizieren und Bewerten der Risiken im Zusammenhang mit dem Einsatz von KI im Gesundheitssektor
- Bewerten der potenziellen Risiken im Zusammenhang mit dem Einsatz von KI in der Industrie
- Anwenden von Techniken der künstlichen Intelligenz in der Industrie zur Verbesserung der Produktivität
- Entwerfen von Lösungen der künstlichen Intelligenz zur Optimierung von Prozessen in der öffentlichen Verwaltung
- Bewerten des Einsatzes von KI-Technologien im Bildungssektor
- Anwenden von Techniken der künstlichen Intelligenz in der Forst- und Landwirtschaft zur Verbesserung der Produktivität
- Optimieren von Personalprozessen durch den strategischen Einsatz von künstlicher Intelligenz

Modul 16. Produktivitätssteigerung in der Softwareentwicklung mit KI

- Vertiefen der Implementierung der wichtigsten KI-Erweiterungen in Visual Studio Code, um die Produktivität zu steigern und die Softwareentwicklung zu erleichtern
- Gewinnen eines soliden Verständnisses grundlegender KI-Konzepte und ihrer Anwendung in der Softwareentwicklung, einschließlich Algorithmen für maschinelles Lernen, Verarbeitung natürlicher Sprache, neuronale Netze usw.
- Beherrschen der Konfiguration optimierter Entwicklungsumgebungen, um sicherzustellen, dass die Studenten Umgebungen schaffen können, die für KI-Projekte förderlich sind
- Anwenden spezifischer Techniken unter Verwendung von ChatGPT für die automatische Identifizierung und Korrektur potenzieller Code-Verbesserungen, wodurch effizientere Programmiermethoden gefördert werden
- Fördern der Zusammenarbeit zwischen verschiedenen Programmierern (von Programmierern über Dateningenieure bis hin zu Designern für Benutzererfahrungen), um effektive und ethische KI-Softwarelösungen zu entwickeln

Modul 17. Softwarearchitektur mit KI

- Entwickeln von Fähigkeiten zur Erstellung robuster Testpläne, die verschiedene Testarten abdecken und die Softwarequalität sicherstellen
- Erkennen und Analysieren verschiedener Arten von Softwarearchitekturen, wie monolithisch, Microservices oder serviceorientiert
- Gewinnen eines umfassenden Überblicks über die Prinzipien und Techniken zur Entwicklung von Computersystemen, die skalierbar sind und große Datenmengen verarbeiten können
- Anwenden fortgeschrittener Fähigkeiten bei der Implementierung von KI-gestützten Datenstrukturen, um die Leistung und Effizienz von Software zu optimieren
- Entwickeln sicherer Entwicklungspraktiken, wobei der Schwerpunkt auf der Vermeidung von Schwachstellen liegt, um die Software-Sicherheit auf Architekturebene zu gewährleisten

Modul 18. Webprojekte mit KI

- Entwickeln umfassender Fähigkeiten für die Umsetzung von Webprojekten, vom *Frontend*-Design bis zur *Backend*-Optimierung, unter Einbeziehung von KI-Elementen
- Optimieren des Prozesses der Bereitstellung von Websites unter Einbeziehung von Techniken und Tools zur Verbesserung von Geschwindigkeit und Effizienz
- Integrieren von KI in das *Cloud Computing*, so dass die Studenten hoch skalierbare und effiziente Webprojekte erstellen können
- Erwerben von Fähigkeiten, um spezifische Probleme und Möglichkeiten in Webprojekten zu erkennen, bei denen KI effektiv eingesetzt werden kann, wie z. B. bei der Textverarbeitung, Personalisierung, Inhaltsempfehlungen usw.
- Ermutigen der Studenten, sich über die neuesten Trends und Fortschritte im Bereich der KI auf dem Laufenden zu halten, um sie in Webprojekten richtig einzusetzen

Modul 19. Mobile Anwendungen mit KI

- Anwenden fortgeschrittener Konzepte für eine *Clean Architecture, Datasources und Repositories*, um eine robuste und modulare Struktur in KI-gestützten mobilen Anwendungen zu gewährleisten
- Entwickeln von Fähigkeiten zur Gestaltung interaktiver Bildschirme, Icons und grafischer Ressourcen mit KI, um das Benutzererlebnis in mobilen Anwendungen zu verbessern
- Eingehen auf die Konfiguration des Frameworks für mobile Anwendungen und Nutzen von *Github Copilot* zur Rationalisierung des Entwicklungsprozesses
- Optimieren von KI-fähigen mobilen Anwendungen für eine effiziente Leistung unter Berücksichtigung von Ressourcenmanagement und Datennutzung
- Durchführen von Qualitätstests für mobile KI-Anwendungen, die es den Studenten ermöglichen, Probleme zu identifizieren und Fehler zu beheben

Modul 20. KI für QA-*Testing*

- Beherrschen von Prinzipien und Techniken zur Entwicklung von Computersystemen, die skalierbar sind und große Datenmengen verarbeiten können
- Anwenden fortgeschrittener Fähigkeiten bei der Implementierung von KI-gestützten Datenstrukturen, um die Leistung und Effizienz von Software zu optimieren
- Verstehen und Anwenden von sicheren Entwicklungspraktiken mit Schwerpunkt auf der Vermeidung von Schwachstellen wie Injektion, um die Software-Sicherheit auf architektonischer Ebene zu gewährleisten
- Erstellen automatisierter Tests, insbesondere in Web- und Mobilumgebungen, unter Einbeziehung von KI-Tools zur Verbesserung der Effizienz des Prozesses
- Einsetzen fortschrittlicher KI-gestützter QA-Tools für eine effizientere Erkennung von *Bugs* und kontinuierliche Softwareverbesserung

Mit diesem einzigartigen 100%igen Online-Studiengang werden Sie die Technologien der Zukunft beherrschen. Nur bei TECH!“

03

Kompetenzen

Dieser Studiengang verschafft den Absolventen einen bedeutenden Vorteil in der Computerentwicklungsbranche, indem er sie mit spezifischen und aktuellen Fähigkeiten im Bereich der künstlichen Intelligenz (KI) ausstattet. Fachleute werden nicht nur in der Lage sein, fortschrittliche Software zu entwerfen und zu entwickeln, sondern auch KI-Lösungen in verschiedenen Anwendungen effektiv zu implementieren, von Web- und Mobilprojekten bis hin zu groß angelegten Softwarearchitekturen. Darüber hinaus wird durch die Behandlung von Entwicklungsproduktivität und *Best Practices* für *QA Testing* sichergestellt, dass Informatiker auf reale Herausforderungen vorbereitet sind und sich in einem sich ständig weiterentwickelnden Bereich auszeichnen.

Dank dieses Universitätsprogramms werden Sie in der Lage sein, KI-Algorithmen in Webprojekten und mobilen Anwendungen zu implementieren"

Allgemeine Kompetenzen

- Anwenden von KI-Erweiterungen in Visual Studio Code und *No-Code*-Designtechniken zur Steigerung der Effizienz bei der Softwareentwicklung
- Verwenden von ChatGPT zur Optimierung und Verbesserung der Codequalität unter Anwendung fortschrittlicher Programmierpraktiken
- Implementieren von Webprojekten, von der Erstellung von *Workspaces* bis zur Bereitstellung, unter Einbeziehung von KI sowohl im *Frontend* als auch im *Backend*
- Entwickeln KI-gestützter mobiler Anwendungen, von der Umgebungskonfiguration bis zur Erstellung fortschrittlicher Funktionen und der Verwaltung grafischer Ressourcen
- Anwenden fortschrittlicher Speicherkonzepte und KI-gestützter Datenstrukturen zur Verbesserung der Effizienz und Skalierbarkeit von Systemen
- Einbeziehen sicherer Entwicklungspraktiken und Vermeiden von Schwachstellen wie Injection, um die Integrität und Sicherheit der entwickelten Software zu gewährleisten

Sie werden in der Lage sein, mit Hilfe von künstlicher Intelligenz personalisierte und intuitive Benutzererfahrungen zu gestalten. Schreiben Sie sich jetzt ein!"

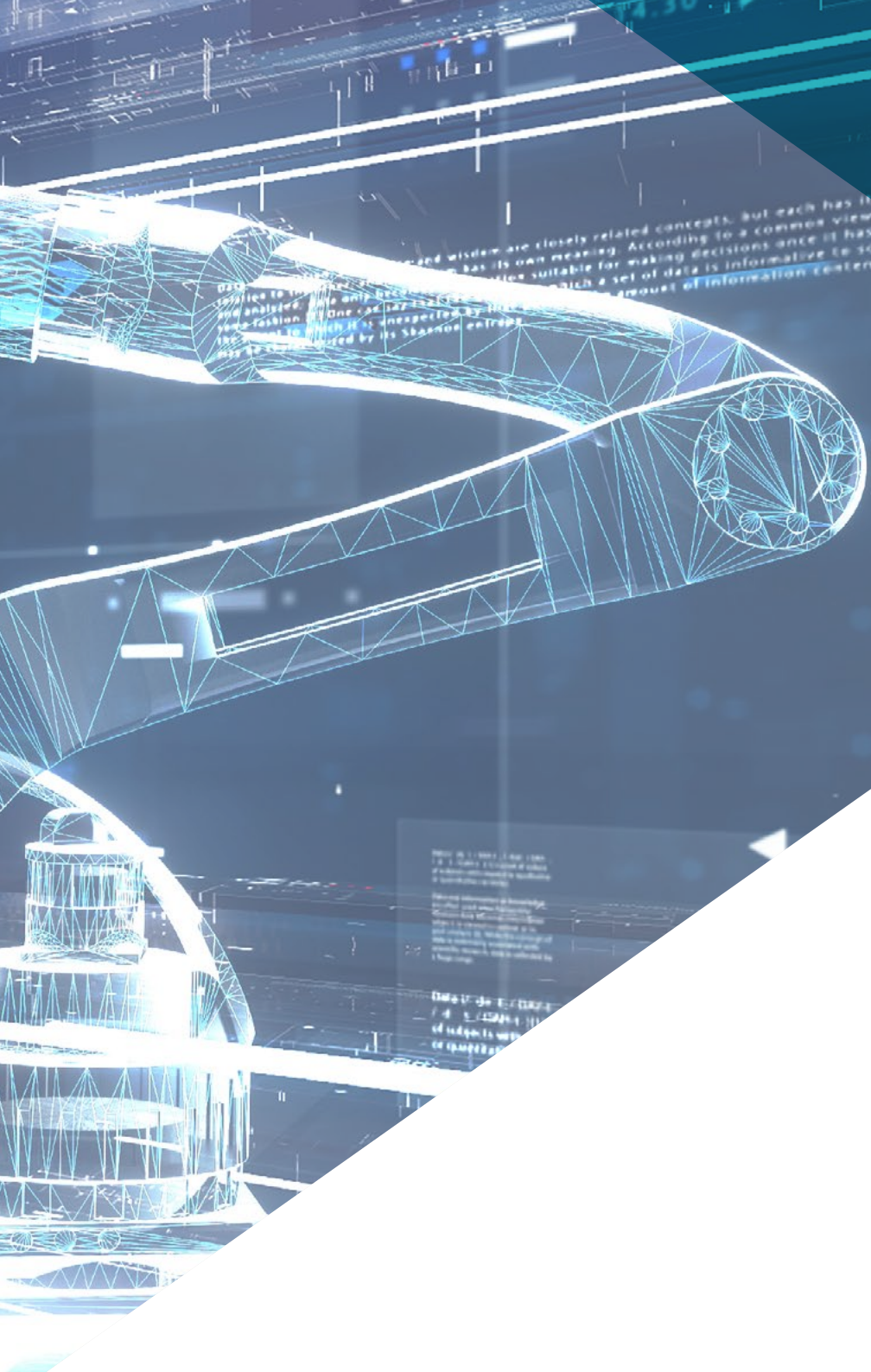
Spezifische Kompetenzen

- Anwenden von KI-Techniken und -Strategien zur Verbesserung der Effizienz im *Retail*
- Anwenden von Entrauschungstechniken unter Verwendung von automatischen Kodierern
- Effektives Erstellen von Trainingsdatensätzen für Aufgaben der natürlichen Sprachverarbeitung (NLP)
- Ausführen von *Clustering*-Schichten und deren Verwendung in *Deep Computer Vision*-Modellen mit Keras
- Verwenden von *TensorFlow*-Funktionen und Graphen, um die Leistung von benutzerdefinierten Modellen zu optimieren
- Optimieren der Entwicklung und Anwendung von *Chatbots* und virtuellen Assistenten, indem man versteht, wie sie funktionieren und welche Anwendungsmöglichkeiten sie bieten
- Beherrschen der Wiederverwendung von vortrainierten Schichten, um den Trainingsprozess zu optimieren und zu beschleunigen
- Erstellen eines ersten neuronalen Netzes, indem die erlernten Konzepte in der Praxis angewendet werden
- Aktivieren eines mehrschichtigen Perzeptrons (MLP) mit der Keras-Bibliothek
- Anwenden von Datenexplorations- und Vorverarbeitungstechniken zur Identifizierung und Vorbereiten von Daten für die effektive Verwendung in maschinellen Lernmodellen
- Untersuchen von Sprachen und Software für die Erstellung von Ontologien unter Verwendung spezifischer Tools für die Entwicklung semantischer Modelle
- Entwickeln von Techniken zur Datenbereinigung, um die Qualität und Genauigkeit der in der nachfolgenden Analyse verwendeten Informationen zu gewährleisten
- Beherrschen der Konfiguration optimierter Entwicklungsumgebungen, um sicherzustellen, dass die Studenten Umgebungen schaffen können, die für KI-Projekte förderlich sind
- Anwenden spezifischer Techniken unter Verwendung von ChatGPT für die automatische Identifizierung und Korrektur potenzieller Code-Verbesserungen, wodurch effizientere Programmierpraktiken gefördert werden
- Erstellen automatisierter Tests, insbesondere in Web- und Mobilumgebungen, unter Einbindung von KI-Tools zur Verbesserung der Effizienz des Prozesses
- Verwenden fortschrittlicher KI-gestützter QA-Tools für eine effizientere Fehlererkennung und kontinuierliche Softwareverbesserung
- Integrieren von KI in das *Cloud Computing*, um Studenten in die Lage zu versetzen, hochskalierbare und effiziente Webprojekte zu erstellen
- Konfigurieren des Frameworks für mobile Apps und Nutzung von Github Copilot zur Rationalisierung des Entwicklungsprozesses

04

Kursleitung

Im Rahmen ihres Engagements für eine exzellente Lehre hat die TECH die für die Entwicklung des Lehrplans dieses Studiengangs verantwortlichen Dozenten sorgfältig ausgewählt. Aus diesem Grund verfügt dieser Studiengang über einen erfahrenen Lehrkörper mit einem hervorragenden Hintergrund in der Anwendung von künstlicher Intelligenz bei Programmieraufgaben. Auf diese Weise haben die Studenten dieses privaten Masterstudiengangs Zugang zu einer erstklassigen Bildungserfahrung mit einer einzigartigen Kombination von Wissen, das in verschiedenen audiovisuellen Medien präsentiert wird, um eine effektivere und dynamischere Integration von Wissen zu ermöglichen.



Informieren Sie sich bei den besten Experten auf diesem Gebiet über die neuesten Trends im Bereich der auf die Programmierung angewandten künstlichen Intelligenz"

Leitung

Dr. Peralta Martín-Palomino, Arturo

- CEO und CTO bei Prometeus Global Solutions
- CTO bei Korporate Technologies
- CTO bei AI Shepherds GmbH
- Berater und strategischer Unternehmensberater bei Alliance Medical
- Direktor für Design und Entwicklung bei DocPath
- Promotion in Computertechnik an der Universität von Castilla La Mancha
- Promotion in Wirtschaftswissenschaften, Unternehmen und Finanzen an der Universität Camilo José Cela
- Promotion in Psychologie an der Universität von Castilla La Mancha
- Masterstudiengang Executive MBA von der Universität Isabel I
- Masterstudiengang in Business und Marketing Management von der Universität Isabel I
- Masterstudiengang in Big Data bei Formación Hadoop
- Masterstudiengang in Fortgeschrittene Informationstechnologie von der Universität von Castilla La Mancha
- Mitglied von: Forschungsgruppe SMILE

Hr. Castellanos Herreros, Ricardo

- *Chief Technology Officer* bei OWQLO
- Spezialist für Computersystemtechnik und *Machine Learning Engineer*
- *Freelance* Technischer Berater
- Entwickler von mobilen Anwendungen für eDreams, Fnac, Air Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón und Grupo Planeta
- Webentwickler für Openbank und Banco Santander
- Technischer Ingenieur für Computersysteme von der Universität von Castilla La Mancha

05

Struktur und Inhalt

Der Studiengang Künstliche Intelligenz in der Programmierung zeichnet sich durch seinen umfassenden Ansatz aus, der sich nicht nur mit der Implementierung intelligenter Algorithmen befasst, sondern auch mit der Verbesserung der Produktivität in der Softwareentwicklung und der Anwendung von KI in Schlüsselbereichen wie *QA Testing*, Webprojekten, mobilen Anwendungen und Softwarearchitektur. Die Kombination aus technischen Fähigkeiten, fortschrittlichen Tools und der praktischen Anwendung von KI in verschiedenen Entwicklungsphasen macht dieses Programm zu einem führenden Programm, das Fachleuten ein umfassendes und tiefes Verständnis der Anwendung von KI in der Programmierung vermittelt.

Sie werden sich mit der praktischen Anwendung von KI in Web-Projekten befassen, einschließlich der Frontend- und Backend-Entwicklung"

Modul 1. Grundlagen der künstlichen Intelligenz

1.1. Geschichte der künstlichen Intelligenz
- 1.1.1. Ab wann spricht man von künstlicher Intelligenz?
- 1.1.2. Referenzen im Kino
- 1.1.3. Bedeutung der künstlichen Intelligenz
- 1.1.4. Technologien, die künstliche Intelligenz ermöglichen und unterstützen

1.2. Künstliche Intelligenz in Spielen
- 1.2.1. Spieltheorie
- 1.2.2. *Minimax* und Alpha-Beta-Beschneidung
- 1.2.3. Simulation: Monte Carlo

1.3. Neuronale Netzwerke
- 1.3.1. Biologische Grundlagen
- 1.3.2. Berechnungsmodell
- 1.3.3. Überwachte und nicht überwachte neuronale Netzwerke
- 1.3.4. Einfaches Perzeptron
- 1.3.5. Mehrschichtiges Perzeptron

1.4. Genetische Algorithmen
- 1.4.1. Geschichte
- 1.4.2. Biologische Grundlage
- 1.4.3. Problem-Kodierung
- 1.4.4. Erzeugung der Ausgangspopulation
- 1.4.5. Hauptalgorithmus und genetische Operatoren
- 1.4.6. Bewertung von Personen: Fitness

1.5. Thesauri, Vokabularien, Taxonomien
- 1.5.1. Wortschatz
- 1.5.2. Taxonomie
- 1.5.3. Thesauri
- 1.5.4. Ontologien
- 1.5.5. Wissensrepräsentation: Semantisches Web

1.6. Semantisches Web
- 1.6.1. Spezifizierungen: RDF, RDFS und OWL
- 1.6.2. Schlussfolgerung/Begründung
- 1.6.3. *Linked Data*

1.7. Expertensysteme und DSS
- 1.7.1. Expertensysteme
- 1.7.2. Systeme zur Entscheidungshilfe

1.8. *Chatbots* und virtuelle Assistenten
- 1.8.1. Arten von Assistenten: sprach- und textbasierte Assistenten
- 1.8.2. Grundlegende Bestandteile für die Entwicklung eines Assistenten: *Intents*, Entitäten und Dialogablauf
- 1.8.3. Integrationen: Web, *Slack*, Whatsapp, Facebook
- 1.8.4. Tools für die Entwicklung von Assistenten: *Dialog Flow*, *Watson Assistant*

1.9. KI-Implementierungsstrategie

1.10. Die Zukunft der künstlichen Intelligenz
- 1.10.1. Wir wissen, wie man mit Algorithmen Emotionen erkennt
- 1.10.2. Eine Persönlichkeit schaffen: Sprache, Ausdrücke und Inhalt
- 1.10.3. Tendenzen der künstlichen Intelligenz
- 1.10.4. Reflexionen

Modul 2. Datentypen und Datenlebenszyklus

2.1. Die Statistik
- 2.1.1. Statistik: Deskriptive Statistik, statistische Schlussfolgerungen
- 2.1.2. Population, Stichprobe, Individuum
- 2.1.3. Variablen: Definition und Mess-Skalen

2.2. Arten von statistischen Daten
- 2.2.1. Je nach Typ
  - 2.2.1.1. Quantitativ: kontinuierliche Daten und diskrete Daten
  - 2.2.1.2. Qualitativ: Binomialdaten, nominale Daten und ordinale Daten
- 2.2.2. Je nach Form
  - 2.2.2.1. Numerisch
  - 2.2.2.2. Text
  - 2.2.2.3. Logisch
- 2.2.3. Je nach Quelle
  - 2.2.3.1. Primär
  - 2.2.3.2. Sekundär

2.3. Lebenszyklus der Daten
- 2.3.1. Etappen des Zyklus
- 2.3.2. Meilensteine des Zyklus
- 2.3.3. FAIR-Prinzipien

2.4. Die ersten Phasen des Zyklus
- 2.4.1. Definition von Zielen
- 2.4.2. Ermittlung des Ressourcenbedarfs
- 2.4.3. Gantt-Diagramm
- 2.4.4. Struktur der Daten

2.5. Datenerhebung
- 2.5.1. Methodik der Erhebung
- 2.5.2. Erhebungsinstrumente
- 2.5.3. Kanäle für die Erhebung

2.6. Datenbereinigung
- 2.6.1. Phasen der Datenbereinigung
- 2.6.2. Qualität der Daten
- 2.6.3. Datenmanipulation (mit R)

2.7. Datenanalyse, Interpretation und Bewertung der Ergebnisse
- 2.7.1. Statistische Maßnahmen
- 2.7.2. Beziehungsindizes
- 2.7.3. *Data Mining*

2.8. Datenlager (*Datawarehouse*)
- 2.8.1. Elemente, aus denen sie bestehen
- 2.8.2. Design
- 2.8.3. Zu berücksichtigende Aspekte

2.9. Verfügbarkeit von Daten
- 2.9.1. Zugang
- 2.9.2. Nützlichkeit
- 2.9.3. Sicherheit

2.10. Regulatorische Aspekte
- 2.10.1. Datenschutzgesetz
- 2.10.2. Bewährte Verfahren
- 2.10.3. Andere regulatorische Aspekte

Modul 3. Daten in der künstlichen Intelligenz

3.1. Datenwissenschaft
- 3.1.1. Datenwissenschaft
- 3.1.2. Fortgeschrittene Tools für den Datenwissenschaftler

3.2. Daten, Informationen und Wissen
- 3.2.1. Daten, Informationen und Wissen
- 3.2.2. Datentypen
- 3.2.3. Datenquellen

3.3. Von Daten zu Informationen
- 3.3.1. Datenanalyse
- 3.3.2. Arten der Analyse
- 3.3.3. Extraktion von Informationen aus einem *Dataset*

3.4. Extraktion von Informationen durch Visualisierung
- 3.4.1. Visualisierung als Analyseinstrument
- 3.4.2. Visualisierungsmethoden
- 3.4.3. Visualisierung eines Datensatzes

3.5. Qualität der Daten
- 3.5.1. Datenqualität
- 3.5.2. Datenbereinigung
- 3.5.3. Grundlegende Datenvorverarbeitung

3.6. *Dataset*
- 3.6.1. *Dataset*-Anreicherung
- 3.6.2. Der Fluch der Dimensionalität
- 3.6.3. Ändern unseres Datensatzes

3.7. Ungleichgewicht
- 3.7.1. Ungleichgewicht der Klassen
- 3.7.2. Techniken zur Begrenzung von Ungleichgewichten
- 3.7.3. *Dataset*-Abgleich

3.8. Unüberwachte Modelle
- 3.8.1. Unüberwachtes Modell
- 3.8.2. Methoden
- 3.8.3. Klassifizierung mit unüberwachten Modellen

3.9. Überwachte Modelle
- 3.9.1. Überwachtes Modell
- 3.9.2. Methoden
- 3.9.3. Klassifizierung mit überwachten Modellen

3.10. Tools und bewährte Verfahren
- 3.10.1. Bewährte Praktiken für einen Datenwissenschaftler
- 3.10.2. Das beste Modell
- 3.10.3. Nützliche Tools

Modul 4. Data Mining. Auswahl, Vorverarbeitung und Transformation

4.1. Statistische Inferenz
- 4.1.1. Deskriptive Statistik vs. Statistische Inferenz
- 4.1.2. Parametrische Verfahren
- 4.1.3. Nichtparametrische Verfahren

4.2. Explorative Analyse
- 4.2.1. Deskriptive Analyse
- 4.2.2. Visualisierung
- 4.2.3. Vorbereitung der Daten

4.3. Vorbereitung der Daten
- 4.3.1. Datenintegration und -bereinigung
- 4.3.2. Normalisierung der Daten
- 4.3.3. Attribute umwandeln

4.4. Verlorene Werte
- 4.4.1. Umgang mit verlorenen Werten
- 4.4.2. Maximum-Likelihood-Imputationsmethoden
- 4.4.3. Imputation verlorener Werte durch maschinelles Lernen

4.5. Datenrauschen
- 4.5.1. Lärmklassen und Attribute
- 4.5.2. Rauschfilterung
- 4.5.3. Rauscheffekt

4.6. Der Fluch der Dimensionalität
- 4.6.1. *Oversampling*
- 4.6.2. *Undersampling*
- 4.6.3. Multidimensionale Datenreduktion

4.7. Kontinuierliche zu diskreten Attributen
- 4.7.1. Kontinuierliche versus diskrete Daten
- 4.7.2. Prozess der Diskretisierung

4.8. Daten
- 4.8.1. Datenauswahl
- 4.8.2. Perspektiven und Auswahlkriterien
- 4.8.3. Methoden der Auswahl

4.9. Auswahl der Instanzen
- 4.9.1. Methoden für die Instanzauswahl
- 4.9.2. Auswahl der Prototypen
- 4.9.3. Erweiterte Methoden für die Instanzauswahl

4.10. Vorverarbeitung von Daten in *Big-Data*-Umgebungen

Modul 5. Algorithmik und Komplexität in der künstlichen Intelligenz

5.1. Einführung in Algorithmus-Design-Strategien
- 5.1.1. Rekursion
- 5.1.2. Aufteilen und erobern
- 5.1.3. Andere Strategien

5.2. Effizienz und Analyse von Algorithmen
- 5.2.1. Maßnahmen zur Steigerung der Effizienz
- 5.2.2. Messung der Eingabegröße
- 5.2.3. Messung der Ausführungszeit
- 5.2.4. Schlimmster, bester und durchschnittlicher Fall
- 5.2.5. Asymptotische Notation
- 5.2.6. Kriterien für die mathematische Analyse von nicht-rekursiven Algorithmen
- 5.2.7. Mathematische Analyse von rekursiven Algorithmen
- 5.2.8. Empirische Analyse von Algorithmen

5.3. Sortieralgorithmen
- 5.3.1. Konzept der Sortierung
- 5.3.2. Blase sortieren
- 5.3.3. Sortieren nach Auswahl
- 5.3.4. Reihenfolge der Insertion
- 5.3.5. Sortierung zusammenführen (*Merge_Sort*)
- 5.3.6. Schnelle Sortierung (*Quick_Sort*)

5.4. Algorithmen mit Bäumen
- 5.4.1. Konzept des Baumes
- 5.4.2. Binäre Bäume
- 5.4.3. Baumpfade
- 5.4.4. Ausdrücke darstellen
- 5.4.5. Geordnete binäre Bäume
- 5.4.6. Ausgeglichene binäre Bäume

5.5. Algorithmen mit *Heaps*
- 5.5.1. *Heaps*
- 5.5.2. Der *Heapsort*-Algorithmus
- 5.5.3. Prioritätswarteschlangen

5.6. Graph-Algorithmen
- 5.6.1. Vertretung
- 5.6.2. Lauf in Breite
- 5.6.3. Lauf in Tiefe
- 5.6.4. Topologische Anordnung

5.7. *Greedy*-Algorithmen
- 5.7.1. Die *Greedy*-Strategie
- 5.7.2. Elemente der *Greedy*-Strategie
- 5.7.3. Währungsumtausch
- 5.7.4. Das Problem des Reisenden
- 5.7.5. Problem mit dem Rucksack

5.8. Minimale Pfadsuche
- 5.8.1. Das Problem des minimalen Pfades
- 5.8.2. Negative Bögen und Zyklen
- 5.8.3. Dijkstra-Algorithmus

5.9. *Greedy*-Algorithmen auf Graphen
- 5.9.1. Der minimal aufspannende Baum
- 5.9.2. Algorithmus von Prim
- 5.9.3. Algorithmus von Kruskal
- 5.9.4. Komplexitätsanalyse

5.10. *Backtracking*
- 5.10.1. Das *Backtracking*
- 5.10.2. Alternative Techniken

Modul 6. Intelligente Systeme

6.1. Agententheorie
- 6.1.1. Geschichte des Konzepts
- 6.1.2. Definition von Agent
- 6.1.3. Agenten in der künstlichen Intelligenz
- 6.1.4. Agenten in der Softwareentwicklung

6.2. Agent-Architekturen
- 6.2.1. Der Denkprozess eines Agenten
- 6.2.2. Reaktive Agenten
- 6.2.3. Deduktive Agenten
- 6.2.4. Hybride Agenten
- 6.2.5. Vergleich

6.3. Informationen und Wissen
- 6.3.1. Unterscheidung zwischen Daten, Informationen und Wissen
- 6.3.2. Bewertung der Datenqualität
- 6.3.3. Methoden der Datenerfassung
- 6.3.4. Methoden der Informationsbeschaffung
- 6.3.5. Methoden zum Wissenserwerb

6.4. Wissensrepräsentation
- 6.4.1. Die Bedeutung der Wissensrepräsentation
- 6.4.2. Definition der Wissensrepräsentation durch ihre Rollen
- 6.4.3. Merkmale einer Wissensrepräsentation

6.5. Ontologien
- 6.5.1. Einführung in Metadaten
- 6.5.2. Philosophisches Konzept der Ontologie
- 6.5.3. Computergestütztes Konzept der Ontologie
- 6.5.4. Bereichsontologien und Ontologien auf höherer Ebene
- 6.5.5. Wie erstellt man eine Ontologie?

6.6. Ontologiesprachen und Software für die Erstellung von Ontologien
- 1.10.1. RDF-Tripel, *Turtle* und N
- 2.10.2. RDF-Schema
- 2.10.2. OWL
- 6.6.4. SPARQL
- 6.6.5. Einführung in die verschiedenen Tools für die Erstellung von Ontologien
- 6.6.6. Installation und Verwendung von *Protégé*

6.7. Das semantische Web
- 6.7.1. Der aktuelle Stand und die Zukunft des semantischen Webs
- 6.7.2. Anwendungen des semantischen Webs

6.8. Andere Modelle der Wissensdarstellung
- 6.8.1. Wortschatz
- 6.8.2. Globale Sicht
- 6.8.3. Taxonomie
- 6.8.4. Thesauri
- 6.8.5. Folksonomien
- 6.8.6. Vergleich
- 6.8.7. Mind Map

6.9. Bewertung und Integration von Wissensrepräsentationen
- 6.9.1. Logik nullter Ordnung
- 6.9.2. Logik erster Ordnung
- 6.9.3. Beschreibende Logik
- 6.9.4. Beziehung zwischen verschiedenen Arten von Logik
- 6.9.5. *Prolog*: Programmierung auf Basis der Logik erster Ordnung

6.10. Semantische *Reasoner*, wissensbasierte Systeme und Expertensysteme
- 6.10.1. Konzept des *Reasoners*
- 6.10.2. Anwendungen eines *Reasoners*
- 6.10.3. Wissensbasierte Systeme
- 6.10.4. MYCIN, Geschichte der Expertensysteme
- 6.10.5. Elemente und Architektur von Expertensystemen
- 6.10.6. Erstellung von Expertensystemen

Modul 7. Maschinelles Lernen und Data Mining

7.1. Einführung in die Prozesse der Wissensentdeckung und in die grundlegenden Konzepte des maschinellen Lernens
- 7.1.1. Schlüsselkonzepte von Prozessen der Wissensentdeckung
- 7.1.2. Historische Perspektive der Wissensentdeckungsprozesse
- 7.1.3. Phasen des Wissensentdeckungsprozesses
- 7.1.4. Techniken, die bei der Wissensentdeckung eingesetzt werden
- 7.1.5. Merkmale guter Modelle für maschinelles Lernen
- 7.1.6. Arten von Informationen zum maschinellen Lernen
- 7.1.7. Grundlegende Lernkonzepte
- 7.1.8. Grundlegende Konzepte des unüberwachten Lernens

7.2. Datenexploration und Vorverarbeitung
- 7.2.1. Datenverarbeitung
- 7.2.2. Datenverarbeitung im Datenanalysefluss
- 7.2.3. Datentypen
- 7.2.4. Datenumwandlung
- 7.2.5. Anzeige und Untersuchung von kontinuierlichen Variablen
- 7.2.6. Anzeige und Erkundung kategorialer Variablen
- 7.2.7. Korrelationsmaßnahmen
- 7.2.8. Die häufigsten grafischen Darstellungen
- 7.2.9. Einführung in die multivariate Analyse und Dimensionsreduktion

7.3. Entscheidungsbaum
- 7.3.1. ID-Algorithmus
- 7.3.2. Algorithmus C
- 7.3.3. Übertraining und Beschneidung
- 7.3.4. Analyse der Ergebnisse

7.4. Bewertung von Klassifikatoren
- 7.4.1. Konfusionsmatrizen
- 7.4.2. Numerische Bewertungsmatrizen
- 7.4.3. Kappa-Statistik
- 7.4.4. Die ROC-Kurve

7.5. Klassifizierungsregeln
- 7.5.1. Maßnahmen zur Bewertung von Regeln
- 7.5.2. Einführung in die grafische Darstellung
- 7.5.3. Sequentieller Überlagerungsalgorithmus

7.6. Neuronale Netze
- 7.6.1. Grundlegende Konzepte
- 7.6.2. Einfache neuronale Netze
- 7.6.3. *Backpropagation*-Algorithmus
- 7.6.4. Einführung in rekurrente neuronale Netze

7.7. Bayessche Methoden
- 7.7.1. Grundlegende Konzepte der Wahrscheinlichkeit
- 7.7.2. Bayes-Theorem
- 7.7.3. Naive Bayes
- 7.7.4. Einführung in Bayessche Netzwerke

7.8. Regressions- und kontinuierliche Antwortmodelle
- 7.8.1. Einfache lineare Regression
- 7.8.2. Multiple lineare Regression
- 7.8.3. Logistische Regression
- 7.8.4. Regressionsbäume
- 7.8.5. Einführung in *Support Vector Machines* (SVM)
- 7.8.6. Maße für die Anpassungsgüte

7.9. *Clustering*
- 7.9.1. Grundlegende Konzepte
- 7.9.2. Hierarchisches *Clustering*
- 7.9.3. Probabilistische Methoden
- 7.9.4. EM-Algorithmus
- 7.9.5. *B-Cubed*-Methode
- 7.9.6. Implizite Methoden

7.10. *Text Mining* und natürliche Sprachverarbeitung (NLP)
- 7.10.1. Grundlegende Konzepte
- 7.10.2. Erstellung eines Korpus
- 7.10.3. Deskriptive Analyse
- 7.10.4. Einführung in die Stimmungsanalyse

Modul 8. Neuronale Netze, die Grundlage von *Deep Learning*

8.1. Tiefes Lernen
- 8.1.1. Arten von tiefem Lernen
- 8.1.2. Anwendungen von tiefem Lernen
- 8.1.3. Vor- und Nachteile von tiefem Lernen

8.2. Operationen
- 8.2.1. Addition
- 8.2.2. Produkt
- 8.2.3. Transfer

8.3. Ebenen
- 8.3.1. Eingangsebene
- 8.3.2. Ausgeblendete Ebene
- 8.3.3. Ausgangsebene

8.4. Schichtenverbund und Operationen
- 8.4.1. Design-Architekturen
- 8.4.2. Verbindung zwischen Ebenen
- 8.4.3. Vorwärtsausbreitung

8.5. Aufbau des ersten neuronalen Netzes
- 8.5.1. Entwurf des Netzes
- 8.5.2. Festlegen der Gewichte
- 8.5.3. Training des Netzes

8.6. Trainer und Optimierer
- 8.6.1. Auswahl des Optimierers
- 8.6.2. Festlegen einer Verlustfunktion
- 8.6.3. Festlegung einer Metrik

8.7. Anwendung der Prinzipien des neuronalen Netzes
- 8.7.1. Aktivierungsfunktionen
- 8.7.2. Rückwärtsausbreitung
- 8.7.3. Einstellung der Parameter

8.8. Von biologischen zu künstlichen Neuronen
- 8.8.1. Funktionsweise eines biologischen Neurons
- 8.8.2. Wissensübertragung auf künstliche Neuronen
- 8.8.3. Herstellung von Beziehungen zwischen den beiden

8.9. Implementierung von MLP (Multilayer Perceptron) mit Keras
- 8.9.1. Definition der Netzstruktur
- 8.9.2. Modell-Kompilierung
- 8.9.3. Modell-Training

8.10. Feinabstimmung der Hyperparameter von neuronalen Netzen
- 8.10.1. Auswahl der Aktivierungsfunktion
- 8.10.2. Einstellung der *Learning Rate*
- 8.10.3. Einstellung der Gewichte

Modul 9. Training Tiefer Neuronaler Netze

9.1. Gradienten-Probleme
- 9.1.1. Techniken der Gradientenoptimierung
- 9.1.2. Stochastische Gradienten
- 9.1.3. Techniken zur Initialisierung der Gewichte

9.2. Wiederverwendung von vortrainierten Schichten
- 9.2.1. *Transfer Learning Training*
- 9.2.2. Merkmalsextraktion
- 9.2.3. Tiefes Lernen

9.3. Optimierer
- 9.3.1. Stochastische Gradientenabstiegs-Optimierer
- 9.3.2. Adam- und *RMSprop*-Optimierer
- 9.3.3. Moment-Optimierer

9.4. Planen der Lernrate
- 9.4.1. Automatische Steuerung der Lernrate
- 9.4.2. Lernzyklen
- 9.4.3. Bedingungen für die Glättung

9.5. Überanpassung
- 9.5.1. Kreuzvalidierung
- 9.5.2. Regulierung
- 9.5.3. Bewertungsmetriken

9.6. Praktische Leitlinien
- 9.6.1. Entwurf des Modells
- 9.6.2. Auswahl der Metriken und Bewertungsparameter
- 9.6.3. Testen von Hypothesen

9.7. *Transfer Learning*
- 9.7.1. *Transfer Learning Training*
- 9.7.2. Merkmalsextraktion
- 9.7.3. Tiefes Lernen

9.8. *Data Augmentation*
- 9.8.1. Bildtransformationen
- 9.8.2. Generierung synthetischer Daten
- 9.8.3. Textumwandlung

9.9. Praktische Anwendung von *Transfer Learning*
- 9.9.1. *Transfer Learning Training*
- 9.9.2. Merkmalsextraktion
- 9.9.3. Tiefes Lernen

9.10. Regulierung
- 9.10.1. L und L
- 9.10.2. Maximale Entropie-Regularisierung
- 9.10.3. *Dropout*

Modul 10. Anpassung von Modellen und Training mit *TensorFlow*

10.1. *TensorFlow*
- 10.1.1. Verwendung der *TensorFlow*-Bibliothek
- 10.1.2. Training von Modellen mit *TensorFlow*
- 10.1.3. Operationen mit Graphen in *TensorFlow*

10.2. *TensorFlow* und NumPy
- 10.2.1. NumPy-Berechnungsumgebung für *TensorFlow*
- 10.2.2. Verwendung von NumPy-Arrays mit *TensorFlow*
- 10.2.3. NumPy-Operationen für *TensorFlow*-Graphen

10.3. Anpassung von Modellen und Trainingsalgorithmen
- 10.3.1. Erstellen von benutzerdefinierten Modellen mit *TensorFlow*
- 10.3.2. Verwaltung von Trainingsparametern
- 10.3.3. Verwendung von Optimierungstechniken für das Training

10.4. *TensorFlow*-Funktionen und -Graphen
- 10.4.1. Funktionen mit *TensorFlow*
- 10.4.2. Verwendung von Graphen für das Modelltraining
- 10.4.3. Optimieren von Graphen mit *TensorFlow*-Operationen

10.5. Laden und Vorverarbeiten von Daten mit *TensorFlow*
- 10.5.1. Laden von Datensätzen mit *TensorFlow*
- 10.5.2. Vorverarbeiten von Daten mit *TensorFlow*
- 10.5.3. Verwendung von *TensorFlow*-Tools zur Datenmanipulation

10.6. Die *tfdata*-API
- 10.6.1. Verwendung der *tfdata*-API für die Datenverarbeitung
- 10.6.2. Konstruktion von Datenströmen mit *tfdata*
- 10.6.3. Verwendung der *tfdata*-API für das Modelltraining

10.7. Das *TFRecord*-Format
- 10.7.1. Verwendung der *TFRecord*-API für die Datenserialisierung
- 10.7.2. Laden von *TFRecord*-Dateien mit *TensorFlow*
- 10.7.3. Verwendung von *TFRecord*-Dateien für das Modelltraining

10.8. Keras Vorverarbeitungsschichten
- 10.8.1. Verwendung der Keras-API für die Vorverarbeitung
- 10.8.2. Aufbau von Keras-Vorverarbeitungs-*Pipelines*
- 10.8.3. Verwendung der Keras Vorverarbeitungs-API für das Modelltraining

10.9. Das Projekt *TensorFlow Datasets*
- 10.9.1. Verwendung von *TensorFlow Datasets* zum Laden von Daten
- 10.9.2. Vorverarbeitung von Daten mit *TensorFlow Datasets*
- 10.9.3. Verwendung von *TensorFlow Datasets* für das Modelltraining

10.10. Erstellen einer *Deep Learning*-Anwendung mit *TensorFlow*
- 10.10.1. Praktische Anwendung
- 10.10.2. Aufbau einer *Deep Learning*-Anwendung mit *TensorFlow*
- 10.10.3. Trainieren eines Modells mit *TensorFlow*
- 10.10.4. Verwendung der Anwendung für die Vorhersage von Ergebnissen

Modul 11. *Deep Computer Vision* mit *Convolutional Neural Networks*

11.1. Die *Visual-Cortex*-Architektur
- 11.1.1. Funktionen des visuellen Kortex
- 11.1.2. Theorien des rechnergestützten Sehens
- 11.1.3. Modelle der Bildverarbeitung

11.2. Faltungsschichten
- 11.2.1. Wiederverwendung von Gewichten bei der Faltung
- 11.2.2. Faltung D
- 11.2.3. Aktivierungsfunktionen

11.3. Gruppierungsschichten und Implementierung von Gruppierungsschichten mit Keras
- 11.3.1. *Pooling* und *Striding*
- 11.3.2. *Flattening*
- 11.3.3. Arten des *Pooling*

11.4. CNN-Architektur
- 11.4.1. VGG-Architektur
- 11.4.2. *AlexNet*-Architektur
- 11.4.3. *ResNet*-Architektur

11.5. Implementierung eines *ResNet*- CNN mit Keras
- 11.5.1. Initialisierung der Gewichte
- 11.5.2. Definition der Eingabeschicht
- 11.5.3. Definition der Ausgabe

11.6. Verwendung von vortrainierten Keras-Modellen
- 11.6.1. Merkmale der vortrainierten Modelle
- 11.6.2. Verwendung von vortrainierten Modellen
- 11.6.3. Vorteile von vortrainierten Modellen

11.7. Vortrainierte Modelle für das Transferlernen
- 11.7.1. Transferlernen
- 11.7.2. Prozess des Transferlernens
- 11.7.3. Vorteile des Transferlernens

11.8. Klassifizierung und Lokalisierung in *Deep Computer Vision*
- 11.8.1. Klassifizierung von Bildern
- 11.8.2. Objekte in Bildern lokalisieren
- 11.8.3. Objekterkennung

11.9. Objekterkennung und Objektverfolgung
- 11.9.1. Methoden zur Objekterkennung
- 11.9.2. Algorithmen zur Objektverfolgung
- 11.9.3. Verfolgungs- und Lokalisierungstechniken

11.10. Semantische Segmentierung
- 11.10.1. *Deep Learning* für semantische Segmentierung
- 11.10.2. Kantenerkennung
- 11.10.3. Regelbasierte Segmentierungsmethoden

Modul 12. Natürliche Sprachverarbeitung (NLP) mit rekurrenten neuronalen Netzen (RNN) und Aufmerksamkeit

12.1. Textgenerierung mit RNN
- 12.1.1. Training eines RNN für die Texterzeugung
- 12.1.2. Generierung natürlicher Sprache mit RNN
- 12.1.3. Anwendungen zur Texterzeugung mit RNN

12.2. Erstellung von Trainingsdatensätzen
- 12.2.1. Vorbereitung der Daten für das RNN-Training
- 12.2.2. Speicherung des Trainingsdatensatzes
- 12.2.3. Bereinigung und Transformation der Daten
- 12.2.4. Sentiment-Analyse

12.3. Ranking von Meinungen mit RNN
- 12.3.1. Erkennung von Themen in Kommentaren
- 12.3.2. Stimmungsanalyse mit *Deep-Learning*-Algorithmen

12.4. *Encoder-Decoder*-Netz für neuronale maschinelle Übersetzung
- 12.4.1. Training eines RNN für maschinelle Übersetzung
- 12.4.2. Verwendung eines *Encoder-Decoder*-Netzes für die maschinelle Übersetzung
- 12.4.3. Verbesserung der Genauigkeit der maschinellen Übersetzung mit RNNs

12.5. Aufmerksamkeitsmechanismen
- 12.5.1. Implementierung von Aufmerksamkeitsmechanismen in RNN
- 12.5.2. Verwendung von Betreuungsmechanismen zur Verbesserung der Modellgenauigkeit
- 12.5.3. Vorteile von Betreuungsmechanismen in neuronalen Netzen

12.6. *Transformer*-Modelle
- 12.6.1. Verwendung von *Transformer*-Modellen für die Verarbeitung natürlicher Sprache
- 12.6.2. Anwendung von *Transformer*-Modellen für die Sicht
- 12.6.3. Vorteile von *Transformer*-Modellen

12.7. *Transformers* für die Sicht
- 12.7.1. Verwendung von *Transformer* für die Sicht
- 12.7.2. Vorverarbeitung von Bilddaten
- 12.7.3. Training eines *Transformer*-Modells für die Sicht

12.8. *Hugging Face Transformers*-Bibliothek
- 12.8.1. Verwendung der *Hugging Face Transformers*-Bibliothek
- 12.8.2. Anwendung der *Hugging Face Transformers*-Bibliothek
- 12.8.3. Vorteile der *Hugging Face Transformers*-Bibliothek

12.9. Andere *Transformer*-Bibliotheken. Vergleich
- 12.9.1. Vergleich zwischen den verschiedenen *Transformer*-Bibliotheken
- 12.9.2. Verwendung der anderen *Transformer*-Bibliotheken
- 12.9.3. Vorteile der anderen *Transformer*-Bibliotheken

12.10. Entwicklung einer NLP-Anwendung mit RNN und Aufmerksamkeit. Praktische Anwendung
- 12.10.1. Entwicklung einer Anwendung zur Verarbeitung natürlicher Sprache mit RNN und Aufmerksamkeit
- 12.10.2. Verwendung von RNN, Aufmerksamkeitsmechanismen und *Transformers*-Modellen in der Anwendung
- 12.10.3. Bewertung der praktischen Umsetzung

Modul 13. *Autoencoder*, GANs und Diffusionsmodelle

13.1. Effiziente Datendarstellungen
- 13.1.1. Reduzierung der Dimensionalität
- 13.1.2. Tiefes Lernen
- 13.1.3. Kompakte Repräsentationen

13.2. Realisierung von PCA mit einem unvollständigen linearen automatischen Kodierer
- 13.2.1. Trainingsprozess
- 13.2.2. Python-Implementierung
- 13.2.3. Verwendung von Testdaten

13.3. Gestapelte automatische Kodierer
- 13.3.1. Tiefe neuronale Netze
- 13.3.2. Konstruktion von Kodierungsarchitekturen
- 13.3.3. Verwendung der Regularisierung

13.4. Faltungs-Autokodierer
- 13.4.1. Entwurf eines Faltungsmodells
- 13.4.2. Training von Faltungsmodellen
- 13.4.3. Auswertung der Ergebnisse

13.5. Automatische Entrauschung des Encoders
- 13.5.1. Anwendung von Filtern
- 13.5.2. Entwurf von Kodierungsmodellen
- 13.5.3. Anwendung von Regularisierungstechniken

13.6. Automatische Verteilkodierer
- 13.6.1. Steigerung der Kodierungseffizienz
- 13.6.2. Minimierung der Anzahl von Parametern
- 13.6.3. Verwendung von Regularisierungstechniken

13.7. Automatische Variationskodierer
- 13.7.1. Verwendung der Variationsoptimierung
- 13.7.2. Unüberwachtes tiefes Lernen
- 13.7.3. Tiefe latente Repräsentationen

13.8. Modische MNIST-Bilderzeugung
- 13.8.1. Mustererkennung
- 13.8.2. Bilderzeugung
- 13.8.3. Training Tiefer Neuronaler Netze

13.9. *Generative Adversarial Networks* und Diffusionsmodelle
- 13.9.1. Bildbasierte Inhaltsgenerierung
- 13.9.2. Modellierung von Datenverteilungen
- 13.9.3. Verwendung von *Adversarial Networks*

13.10. Implementierung der Modelle
- 13.10.1. Praktische Anwendung
- 13.10.2. Implementierung der Modelle
- 13.10.3. Verwendung von realen Daten
- 13.10.4. Auswertung der Ergebnisse

Modul 14. Bio-inspiriertes Computing

14.1. Einführung in das bio-inspirierte Computing
- 14.1.1. Einführung in das bio-inspirierte Computing

14.2. Algorithmen zur sozialen Anpassung
- 14.2.1. Bioinspiriertes Computing auf der Grundlage von Ameisenkolonien
- 14.2.2. Varianten von Ameisenkolonie-Algorithmen
- 14.2.3. Cloud-basiertes Computing auf Partikelebene

14.3. Genetische Algorithmen
- 14.3.1. Allgemeine Struktur
- 14.3.2. Implementierungen der wichtigsten Operatoren

14.4. Explorations-Ausbeutungsraum-Strategien für genetische Algorithmen
- 14.4.1. CHC-Algorithmus
- 14.4.2. Multimodale Probleme

14.5. Evolutionäre Berechnungsmodelle (I)
- 14.5.1. Evolutionäre Strategien
- 14.5.2. Evolutionäre Programmierung
- 14.5.3. Algorithmen auf der Grundlage der differentiellen Evolution

14.6. Evolutionäre Berechnungsmodelle (II)
- 14.6.1. Evolutionäre Modelle auf der Grundlage der Schätzung von Verteilungen (EDA)
- 14.6.2. Genetische Programmierung

14.7. Evolutionäre Programmierung angewandt auf Lernprobleme
- 14.7.1. Regelbasiertes Lernen
- 14.7.2. Evolutionäre Methoden bei Instanzauswahlproblemen

14.8. Multi-Objektive Probleme
- 14.8.1. Konzept der Dominanz
- 14.8.2. Anwendung evolutionärer Algorithmen auf multikriterielle Probleme

14.9. Neuronale Netze (I)
- 14.9.1. Einführung in neuronale Netzwerke
- 14.9.2. Praktisches Beispiel mit neuronalen Netzwerken

14.10. Neuronale Netze
- 14.10.1. Anwendungsbeispiele für neuronale Netze in der medizinischen Forschung
- 14.10.2. Anwendungsbeispiele für neuronale Netze in der Wirtschaft
- 14.10.3. Anwendungsfälle für neuronale Netze in der industriellen Bildverarbeitung

Modul 15. Künstliche Intelligenz: Strategien und Anwendungen

15.1. Finanzdienstleistungen
- 15.1.1. Die Auswirkungen von künstlicher Intelligenz (KI) auf Finanzdienstleistungen. Chancen und Herausforderungen
- 15.1.2. Anwendungsbeispiele
- 15.1.3. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.1.4. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.2. Auswirkungen von künstlicher Intelligenz im Gesundheitswesen
- 15.2.1. Auswirkungen von KI im Gesundheitswesen. Chancen und Herausforderungen
- 15.2.2. Anwendungsbeispiele

15.3. Risiken im Zusammenhang mit dem Einsatz von KI im Gesundheitswesen
- 15.3.1. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.3.2. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.4. *Retail*
- 15.4.1. Auswirkungen von KI im *Retail*. Chancen und Herausforderungen
- 15.4.2. Anwendungsbeispiele
- 15.4.3. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.4.4. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.5. Industrie
- 15.5.1. Auswirkungen von KI in der Industrie. Chancen und Herausforderungen
- 15.5.2. Anwendungsbeispiele

15.6. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI in der Industrie
- 15.6.1. Anwendungsbeispiele
- 15.6.2. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.6.3. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.7. Öffentliche Verwaltung
- 15.7.1. Auswirkungen von KI in der Öffentlichen Verwaltung. Chancen und Herausforderungen
- 15.7.2. Anwendungsbeispiele
- 15.7.3. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.7.4. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.8. Bildung
- 15.8.1. Auswirkungen von KI in der Bildung. Chancen und Herausforderungen
- 15.8.2. Anwendungsbeispiele
- 15.8.3. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.8.4. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.9. Forst- und Landwirtschaft
- 15.9.1. Auswirkungen von KI in der Forst- und Landwirtschaft. Chancen und Herausforderungen
- 15.9.2. Anwendungsbeispiele
- 15.9.3. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.9.4. Mögliche zukünftige Entwicklungen/Nutzungen von KI

15.10. Das Personalwesen
- 15.10.1. Auswirkungen von KI im Personalwesen. Chancen und Herausforderungen
- 15.10.2. Anwendungsbeispiele
- 15.10.3. Potenzielle Risiken im Zusammenhang mit dem Einsatz von KI
- 15.10.4. Mögliche zukünftige Entwicklungen/Nutzungen von KI

Modul 16. Produktivitätssteigerung in der Softwareentwicklung mit KI

16.1. Vorbereiten einer geeigneten Entwicklungsumgebung
- 16.1.1. Auswahl der wichtigsten Tools für die KI-Entwicklung
- 16.1.2. Konfiguration der ausgewählten Tools
- 16.1.3. Implementierung von CI/CD-Pipelines, die für KI-Projekte geeignet sind
- 16.1.4. Effiziente Verwaltung von Abhängigkeiten und Versionen in Entwicklungsumgebungen

16.2. Wesentliche KI-Erweiterungen für Visual Studio Code
- 16.2.1. Erkundung und Auswahl von KI-Erweiterungen für Visual Studio Code
- 16.2.2. Integration von statischen und dynamischen Analysewerkzeugen in die IDE
- 16.2.3. Automatisieren sich wiederholender Aufgaben mit spezifischen Erweiterungen
- 16.2.4. Anpassung der Entwicklungsumgebung zur Verbesserung der Effizienz

16.3. *No-Code*-Design von Benutzeroberflächen mit KI-Elementen
- 16.3.1. *No-Code*-Designprinzipien und ihre Anwendung auf Benutzeroberflächen
- 16.3.2. Einbindung von KI-Elementen in das Design visueller Schnittstellen
- 16.3.3. Tools und Plattformen für die *No-Code*-Erstellung von intelligenten Schnittstellen
- 16.3.4. Bewertung und kontinuierliche Verbesserung von KI-gestützten *No-Code*-Schnittstellen

16.4. Code-Optimierung mit ChatGPT
- 16.4.1. Identifizieren von doppeltem Code
- 16.4.2. *Refactoring*
- 16.4.3. Lesbaren Code erstellen
- 16.4.4. Verstehen, was ein Code macht
- 16.4.5. Verbesserung der Benennung von Variablen und Funktionen
- 16.4.6. Automatische Dokumentation erstellen

16.5. Verwaltung von Repositorien mit KI durch ChatGPT
- 16.5.1. Automatisierung von Versionskontrollprozessen mit KI-Techniken
- 16.5.2. Konflikterkennung und automatische Lösung in kollaborativen Umgebungen
- 16.5.3. Prädiktive Analyse von Änderungen und Trends in Code-Repositorien
- 16.5.4. Verbesserte Organisation und Kategorisierung von Repositorien mithilfe von KI

16.6. KI-Integration in die Datenbankverwaltung mit AskYourDatabase
- 16.6.1. Abfrage- und Leistungsoptimierung durch KI-Techniken
- 16.6.2. Prädiktive Analyse von Datenbankzugriffsmustern
- 16.6.3. Implementierung von Empfehlungssystemen zur Optimierung der Datenbankstruktur
- 16.6.4. Proaktive Überwachung und Erkennung von potenziellen Datenbankproblemen

16.7. KI-basierte Fehlersuche und Erstellung von Unit-Tests mit ChatGPT
- 16.7.1. Automatische Testfallerstellung mit KI-Techniken
- 16.7.2. Frühzeitige Erkennung von Schwachstellen und *Bugs* durch statische Analyse mit KI
- 16.7.3. Verbesserung der Testabdeckung durch Identifizierung kritischer Bereiche mittels KI

16.8. *Pair Programming* mit GitHub Copilot
- 16.8.1. Integration und effektive Nutzung von GitHub Copilot in *Pair-Programming*-Sitzungen
- 16.8.2. Integration und verbesserte Kommunikation und Zusammenarbeit zwischen Entwicklern mit GitHub Copilot
- 16.8.3. Integration von Strategien zur optimalen Nutzung der von GitHub Copilot generierten Code-Vorschläge
- 16.8.4. Integration von Fallstudien und *Best Practices* in KI-unterstütztem *Pair Programming*

16.9. Automatische Übersetzung zwischen Programmiersprachen mit ChatGPT
- 16.9.1. Programmiersprachenspezifische maschinelle Übersetzungstools und -dienste
- 16.9.2. Anpassung von maschinellen Übersetzungsalgorithmen an den Entwicklungskontext
- 16.9.3. Verbesserung der Interoperabilität zwischen verschiedenen Sprachen durch maschinelle Übersetzung
- 16.9.4. Bewertung und Abschwächung potenzieller Herausforderungen und Einschränkungen bei der maschinellen Übersetzung

16.10. Empfohlene KI-Tools zur Verbesserung der Produktivität
- 16.10.1. Vergleichende Analyse von KI-Tools für die Softwareentwicklung
- 16.10.2. Integration von KI-Tools in Arbeitsabläufe
- 16.10.3. Automatisierung von Routineaufgaben mit KI-Tools
- 16.10.4. Bewertung und Auswahl von Tools auf der Grundlage von Projektkontext und Anforderungen

Modul 17. Softwarearchitektur mit KI

17.1. Optimierung und Leistungsmanagement in KI-Tools mit Hilfe von ChatGPT
- 17.1.1. Leistungsanalyse und Profiling in KI-Tools
- 17.1.2. Optimierungsstrategien für KI-Algorithmen und -Modelle
- 17.1.3. Implementierung von *Caching*- und Parallelisierungstechniken zur Verbesserung der Leistung
- 17.1.4. Tools und Methoden für die kontinuierliche Leistungsüberwachung in Echtzeit

17.2. Skalierbarkeit in KI-Anwendungen mit ChatGPT
- 17.2.1. Entwurf skalierbarer Architekturen für KI-Anwendungen
- 17.2.2. Implementierung von Partitionierungs- und Lastverteilungstechniken
- 17.2.3. *Workflow*- und *Workload*-Management in skalierbaren Systemen
- 17.2.4. Strategien für horizontale und vertikale Expansion in Umgebungen mit unterschiedlicher Nachfrage

17.3. Wartbarkeit von Anwendungen mit KI unter Verwendung von ChatGPT
- 17.3.1. Designprinzipien zur Erleichterung der Wartbarkeit in KI-Projekten
- 17.3.2. Dokumentationsstrategien speziell für KI-Modelle und -Algorithmen
- 17.3.3. Implementierung von Unit- und Integrationstests zur Vereinfachung der Wartung
- 17.3.4. Methoden für *Refactoring* und kontinuierliche Verbesserung in Systemen mit KI-Komponenten

17.4. Entwurf von Großsystemen
- 17.4.1. Architektonische Prinzipien für den Entwurf von Großsystemen
- 17.4.2. Dekomposition komplexer Systeme in Mikrodienste
- 17.4.3. Implementierung spezifischer Entwurfsmuster für verteilte Systeme
- 17.4.4. Strategien zur Beherrschung der Komplexität in groß angelegten Architekturen mit KI-Komponenten

17.5. Großes *Data Warehousing* für KI-Tools
- 17.5.1. Auswahl von skalierbaren Datenspeichertechnologien
- 17.5.2. Entwurf von Datenbankschemata für die effiziente Handhabung großer Datenmengen
- 17.5.3. Partitionierungs- und Replikationsstrategien in massiven Datenspeicherumgebungen
- 17.5.4. Implementierung von Datenverwaltungssystemen zur Gewährleistung von Integrität und Verfügbarkeit in KI-Projekten

17.6. Datenstrukturen mit KI unter Verwendung von ChatGPT
- 17.6.1. Anpassung klassischer Datenstrukturen für die Verwendung in KI-Algorithmen
- 17.6.2. Entwurf und Optimierung spezifischer Datenstrukturen mit ChatGPT
- 17.6.3. Integration von effizienten Datenstrukturen in datenintensive Systeme
- 17.6.4. Strategien für Echtzeit-Datenmanipulation und Speicherung in KI-Datenstrukturen

17.7. Programmieralgorithmen für KI-Produkte
- 17.7.1. Entwicklung und Implementierung von anwendungsspezifischen Algorithmen für KI-Anwendungen
- 17.7.2. Strategien zur Auswahl von Algorithmen je nach Problemtyp und Produktanforderungen
- 17.7.3. Anpassung von klassischen Algorithmen für die Integration in KI-Systeme
- 17.7.4. Bewertung und Vergleich der Leistung verschiedener Algorithmen in KI-Entwicklungskontexten

17.8. Entwurfsmuster für die KI-Entwicklung
- 17.8.1. Identifizierung und Anwendung gemeinsamer Entwurfsmuster in Projekten mit KI-Komponenten
- 17.8.2. Entwicklung von spezifischen Mustern für die Integration von Modellen und Algorithmen in bestehende Systeme
- 17.8.3. *Pattern*-Implementierungsstrategien zur Verbesserung der Wiederverwendbarkeit und Wartbarkeit in KI-Projekten
- 17.8.4. Fallstudien und *Best Practices* bei der Anwendung von Entwurfsmustern in KI-Architekturen

17.9. Implementierung einer *Clean Architecture* mit ChatGPT
- 17.9.1. Grundlegende Prinzipien und Konzepte der *Clean Architecture*
- 17.9.2. Anpassung der *Clean Architecture* an Projekte mit KI-Komponenten
- 17.9.3. Implementierung von Schichten und Abhängigkeiten in *Clean-Architecture*-Systemen
- 17.9.4. Vorteile und Herausforderungen der Implementierung von *Clean Architecture* in der KI-Softwareentwicklung

17.10. Sichere Softwareentwicklung in Webanwendungen mit DeepCode
- 17.10.1. Sicherheitsprinzipien bei der Softwareentwicklung mit KI-Komponenten
- 17.10.2. Identifizierung und Entschärfung potenzieller Schwachstellen in KI-Modellen und -Algorithmen
- 17.10.3. Implementierung von sicheren Entwicklungspraktiken in Webanwendungen mit KI-Funktionalitäten
- 17.10.4. Strategien zum Schutz sensibler Daten und zur Verhinderung von Angriffen in KI-Projekten

Modul 18. Webprojekte mit KI

18.1. Vorbereitung der Arbeitsumgebung für die KI-Webentwicklung
- 18.1.1. Konfiguration von Web-Entwicklungsumgebungen für Projekte mit künstlicher Intelligenz
- 18.1.2. Auswahl und Vorbereitung der wichtigsten Tools für die KI-Webentwicklung
- 18.1.3. Integration von spezifischen Bibliotheken und *Frameworks* für KI-Webprojekte
- 18.1.4. Implementierung von *Best Practices* bei der Konfiguration von kollaborativen Entwicklungsumgebungen

18.2. Erstellung von *Workspaces* für KI-Projekte mit GitHub Copilot
- 18.2.1. Effektive Gestaltung und Organisation von *Workspaces* für Webprojekte mit Komponenten der künstlichen Intelligenz
- 18.2.2. Verwendung von Projektmanagement- und Versionskontroll-Tools im *Workspace*
- 18.2.3. Strategien für eine effiziente Zusammenarbeit und Kommunikation im Entwicklungsteam
- 18.2.4. Anpassung des *Workspace* an die spezifischen Bedürfnisse von Webprojekten mit KI

18.3. Entwurfsmuster für Produkte mit GitHub Copilot
- 18.3.1. Identifizierung und Anwendung gängiger Entwurfsmuster in Benutzeroberflächen mit Elementen der künstlichen Intelligenz
- 18.3.2. Entwicklung spezifischer Muster zur Verbesserung der Benutzererfahrung in Webprojekten mit KI
- 18.3.3. Integration von Entwurfsmustern in die allgemeine Architektur von Webprojekten mit KI
- 18.3.4. Bewertung und Auswahl geeigneter Entwurfsmuster je nach Projektkontext

18.4. *Frontend*-Entwicklung mit GitHub Copilot
- 18.4.1. Integration von KI-Modellen in die Präsentationsschicht von Webprojekten
- 18.4.2. Entwicklung von adaptiven Benutzeroberflächen mit KI-Elementen
- 18.4.3. Implementierung von Funktionalitäten zur Verarbeitung natürlicher Sprache (NLP) im *Frontend*
- 18.4.4. Strategien zur Leistungsoptimierung bei der KI-gestützten *Frontend*-Entwicklung

18.5. Erstellung von Datenbanken mit GitHub Copilot
- 18.5.1. Auswahl von Datenbanktechnologien für Webprojekte mit KI
- 18.5.2. Entwerfen von Datenbankschemata für die Speicherung und Verwaltung von KI-bezogenen Daten
- 18.5.3. Implementierung effizienter Speichersysteme für große Datenmengen, die aus KI-Modellen generiert werden
- 18.5.4. Strategien für die Sicherheit und den Schutz sensibler Daten in Datenbanken bei Webprojekten mit KI

18.6. *Backend*-Entwicklung mit GitHub Copilot
- 18.6.1. Integration von KI-Diensten und -Modellen in die Backend-Geschäftslogik
- 18.6.2. Entwicklung von spezifischen APIs und Endpunkten für die Kommunikation zwischen *Frontend*und KI-Komponenten
- 18.6.3. Implementierung von Datenverarbeitungslogik und Entscheidungsfindung im *Backend*mit KI
- 18.6.4. Strategien für Skalierbarkeit und Leistung bei der *Backend*-Entwicklung von Webprojekten mit KI

18.7. Optimierung Ihres Web-Implementierungsprozesses
- 18.7.1. Automatisierung des Prozesses der Erstellung und Bereitstellung von Webprojekten mit ChatGPT
- 18.7.2. Implementierung von CI/CD-Pipelines, angepasst an Webanwendungen mit GitHub Copilot
- 18.7.3. Strategien für effizientes *Release*- und *Update*-Management in kontinuierlichen Deployments
- 18.7.4. Überwachung und Analyse nach der Bereitstellung zur kontinuierlichen Prozessverbesserung

18.8. KI im *Cloud Computing*
- 18.8.1. Integration von Diensten der Künstlichen Intelligenz in *Cloud-Computing*-Plattformen
- 18.8.2. Entwicklung skalierbarer und verteilter Lösungen mit KI-fähigen *Cloud*-Diensten
- 18.8.3. Strategien für ein effizientes Ressourcen- und Kostenmanagement in *Cloud*-Umgebungen mit KI-gestützten Webanwendungen
- 18.8.4. Bewertung und Vergleich von *Cloud-Service*-Anbietern für KI-Webprojekte

18.9. Erstellen eines Projekts mit KI für LAMP-Umgebungen mit Hilfe von ChatGPT
- 18.9.1. Anpassung von Webprojekten auf der Basis des LAMP-Stacks zur Aufnahme von KI-Komponenten
- 18.9.2. Integration von KI-spezifischen Bibliotheken und *Frameworks* in LAMP-Umgebungen
- 18.9.3. Entwicklung von KI-Funktionen zur Ergänzung der traditionellen LAMP-Architektur
- 18.9.4. Strategien für die Optimierung und Wartung von KI-gestützten Webprojekten in LAMP-Umgebungen

18.10. Erstellen eines Projekts mit KI für MEVN-Umgebungen mit Hilfe von ChatGPT
- 18.10.1. Integration von Technologien und Tools aus dem MEVN-Stack mit KI-Komponenten
- 18.10.2. Entwicklung moderner, skalierbarer Webanwendungen in MEVN-Umgebungen mit KI-Funktionen
- 18.10.3. Implementierung von Datenverarbeitungs- und maschinellen Lernfunktionalitäten in MEVN-Projekten
- 18.10.4. Strategien zur Leistungs- und Sicherheitsverbesserung bei Webanwendungen mit KI in MEVN-Umgebungen

Modul 19. Mobile Anwendungen mit KI

19.1. Vorbereitung einer Arbeitsumgebung für die mobile KI-Entwicklung
- 19.1.1. Konfiguration von mobilen Entwicklungsumgebungen für KI-Projekte
- 19.1.2. Auswahl und Vorbereitung spezifischer Tools für die Entwicklung von KI-Mobilanwendungen
- 19.1.3. Integration von KI-Bibliotheken und -*Frameworks* in mobile Entwicklungsumgebungen
- 19.1.4. Konfiguration von Emulatoren und realen Geräten zum Testen mobiler Anwendungen mit Komponenten der künstlichen Intelligenz

19.2. Erstellen eines *Workspace* mit GitHub Copilot
- 19.2.1. Integration von GitHub Copilot in mobile Entwicklungsumgebungen
- 19.2.2. Effektive Nutzung von GitHub Copilot für die Codegenerierung in KI-Projekten
- 19.2.3. Strategien für die Zusammenarbeit von Entwicklern bei der Verwendung von GitHub Copilot im *Workspace*
- 19.2.4. Bewährte Verfahren und Einschränkungen bei der Verwendung von GitHub Copilot in der Entwicklung mobiler Anwendungen mit KI

19.3. Firebase-Konfiguration
- 19.3.1. Ersteinrichtung eines Firebase-Projekts für die mobile Entwicklung
- 19.3.2. Firebase-Integration in mobile Anwendungen mit KI-Funktionen
- 19.3.3. Nutzung von Firebase-Diensten wie Datenbank, Authentifizierung und Benachrichtigungen in KI-Projekten
- 19.3.4. Strategien für die Verwaltung von Echtzeitdaten und Ereignissen in mobilen Anwendungen mit Firebase

19.4. Konzepte der sauberen Architektur, Datenquellen, Repositories
- 19.4.1. Grundlegende Prinzipien der *Clean Architecture* in der mobilen Entwicklung mit KI
- 19.4.2. Implementierung von *DataSources*- und *Repositories*-Schichten mit GitHub Copilot
- 19.4.3. Design und Strukturierung von Komponenten in mobilen Projekten mit Github Copilot
- 19.4.4. Vorteile und Herausforderungen bei der Implementierung von *Clean Architecture* in mobilen Anwendungen mit KI

19.5. Erstellung von Authentifizierungsbildschirmen mit GitHub Copilot
- 19.5.1. Entwerfen und Entwickeln von Benutzeroberflächen für Authentifizierungsbildschirme in mobilen KI-Anwendungen
- 19.5.2. Integration von Authentifizierungsdiensten mit Firebase in den Anmeldebildschirm
- 19.5.3. Verwendung von Sicherheits- und Datenschutztechniken auf dem Authentifizierungsbildschirm
- 19.5.4. Personalisierung und Anpassung des Benutzererlebnisses im Authentifizierungsbildschirm

19.6. Erstellung von *Dashboards* und *Navigation* mit GitHub Copilot
- 19.6.1. *Dashboard*-Design und -Entwicklung mit Elementen der künstlichen Intelligenz
- 19.6.2. Implementierung von effizienten Navigationssystemen in mobilen Anwendungen mit KI
- 19.6.3. Integration von KI-Funktionalitäten in das *Dashboard* zur Verbesserung der Benutzererfahrung

19.7. Erstellung von *Listing*-Bildschirmen mit GitHub Copilot
- 19.7.1. Entwicklung von Benutzeroberflächen für *Listing*-Bildschirme in KI-fähigen mobilen Anwendungen
- 19.7.2. Integration von Empfehlungs- und Filteralgorithmen in den *Listing*-Bildschirm
- 19.7.3. Verwendung von Entwurfsmustern für die effektive Präsentation von Listendaten
- 19.7.4. Strategien für das effiziente Laden von Daten in Echtzeit in den *Listing*-Bildschirm

19.8. Erstellung von Detailbildschirmen mit GitHub Copilot
- 19.8.1. Entwurf und Entwicklung von detaillierten Benutzeroberflächen für die Präsentation bestimmter Informationen
- 19.8.2. Integration von KI-Funktionalitäten zur Bereicherung des Detailbildschirms
- 19.8.3. Implementierung von Interaktionen und Animationen auf dem Detailbildschirm
- 19.8.4. Strategien zur Leistungsoptimierung bei der Detailanzeige und dem Laden von KI-gestützten mobilen Anwendungen

19.9. Erstellung von Konfigurationsbildschirmen mit GitHub Copilot
- 19.9.1. Entwicklung von Benutzeroberflächen für Konfiguration und Einstellungen in KI-fähigen mobilen Anwendungen
- 19.9.2. Integration von benutzerdefinierten Einstellungen im Zusammenhang mit Komponenten der künstlichen Intelligenz
- 19.9.3. Implementierung von Anpassungsoptionen und Einstellungen im Konfigurationsbildschirm
- 19.9.4. Strategien für Benutzerfreundlichkeit und Klarheit bei der Darstellung der Optionen im *Settings*-Bildschirm

19.10. Erstellen von Icons, *Splashes* und grafischen Ressourcen für Ihre App mit KI
- 19.10.1. Entwerfen und Erstellen attraktiver Symbole zur Darstellung der KI-Mobilanwendung
- 19.10.2. Entwicklung von Startbildschirmen (*Splash*) mit eindrucksvollen Grafiken
- 19.10.3. Auswahl und Anpassung von grafischen Ressourcen zur Verbesserung der Ästhetik der mobilen Anwendung
- 19.10.4. Strategien für Konsistenz und visuelles *Branding* in den grafischen Elementen der Anwendung mit KI

Modul 20. KI für QA-*Testing*

20.1. *Testing*-Lebenszyklus
- 20.1.1. Beschreibung und Verständnis des *Testing*-Lebenszyklus in der Softwareentwicklung
- 20.1.2. Phasen des *Testing*-Lebenszyklus und ihre Bedeutung für die Qualitätssicherung
- 20.1.3. Integration von künstlicher Intelligenz in verschiedenen Phasen des *Testing*-Lebenszyklus
- 20.1.4. Strategien zur kontinuierlichen Verbesserung des *Testing*-Lebenszyklus durch den Einsatz von KI

20.2. *Test Cases* und *Bug*-Erkennung mit Hilfe von ChatGPT
- 20.2.1. Effektives Entwerfen und Schreiben von Testfällen im Kontext von QA *Testing*
- 20.2.2. Identifizierung von Bugs und Fehlern während der Ausführung von Testfällen
- 20.2.3. Anwendung von Techniken zur *Bugs*-Früherkennung durch statische Analyse
- 20.2.4. Einsatz von Tools der künstlichen Intelligenz zur automatischen Identifizierung von *Bugs* in *Test Cases*

20.3. Arten von *Testing*
- 20.3.1. Erkundung der verschiedenen *Testing*-Arten im Bereich der QS
- 20.3.2. Unit-, Integrations-, Funktions- und Akzeptanztests: Merkmale und Anwendungen
- 20.3.3. Strategien für die Auswahl und geeignete Kombination von *Testing*-Arten in Projekten mit ChatGPT

20.3.4. Anpassung konventioneller *Testing*-Arten an Projekte mit ChatGPT

20.4. Erstellen eines Testplans mit ChatGPT
- 20.4.1. Entwerfen und Strukturieren eines umfassenden Testplans
- 20.4.2. Identifizierung von Anforderungen und Testszenarien in KI-Projekten
- 20.4.3. Strategien für die manuelle und automatisierte Testplanung
- 20.4.4. Bewertung und kontinuierliche Anpassung des Testplans entsprechend der Projektentwicklung

20.5. Erkennung und Meldung von KI-*Bugs*
- 20.5.1. Implementierung automatischer Fehlererkennungstechniken unter Verwendung von Algorithmen des maschinellen Lernens
- 20.5.2. Einsatz von ChatGPT für die dynamische Codeanalyse zur Suche nach potenziellen Fehlern
- 20.5.3. Strategien für die automatische Erstellung von detaillierten Berichten über die von ChatGPT entdeckten *Bugs*
- 20.5.4. Effektive Zusammenarbeit zwischen Entwicklungs- und QA-Teams bei der Verwaltung von KI-identifizierten Fehlern

20.6. Erstellung von automatisierten Tests mit KI
- 20.6.1. Entwicklung von automatisierten Testskripten für Projekte mit ChatGPT
- 20.6.2. Integration von KI-basierten Testautomatisierungstools
- 20.6.3. Verwendung von ChatGPT für die dynamische Generierung von automatisierten Testfällen
- 20.6.4. Strategien für die effiziente Ausführung und Wartung von automatisierten Testfällen in KI-Projekten

20.7. API *Testing*
- 20.7.1. Grundlegende Konzepte des API-*Testing* und seine Bedeutung in der QA
- 20.7.2. Entwicklung von Tests zur Überprüfung von APIs in Umgebungen mit ChatGPT
- 20.7.3. Strategien zur Daten- und Ergebnisvalidierung bei API-*Testing* mit ChatGPT
- 20.7.4. Verwendung spezifischer Tools für API-*Testing* in KI-Projekten

20.8. KI-Tools für Web-*Testing*
- 20.8.1. Erkundung von Tools der künstlichen Intelligenz für die Testautomatisierung in Webumgebungen
- 20.8.2. Integration von Technologien zur Aufgabenerkennung und visuellen Analyse in Web-*Testing*
- 20.8.3. Strategien für die automatische Erkennung von Änderungen und Leistungsproblemen in Webanwendungen mit ChatGPT
- 20.8.4. Bewertung spezifischer Tools zur Verbesserung der Effizienz von Web-*Testing* mit KI

20.9. *Mobile Testing* mit KI
- 20.9.1. Entwicklung von *Testing*-Strategien für mobile Anwendungen mit Komponenten der künstlichen Intelligenz
- 20.9.2. Integration spezifischer *Testing*-Werkzeuge für mobile Plattformen auf der Grundlage von KI
- 20.9.3. Einsatz von ChatGPT zur Erkennung von Problemen bei der Leistung mobiler Anwendungen
- 20.9.4. Strategien für die Validierung spezifischer Schnittstellen und Funktionen mobiler Anwendungen mithilfe von KI

20.10. QA-Tools mit KI
- 20.10.1. Erkundung von QA-Tools und Plattformen mit KI-Funktionalität
- 20.10.2. Bewertung von Tools für effizientes Testmanagement und Testdurchführung in KI-Projekten
- 20.10.3. Verwendung von ChatGPT für die Optimierung und Generierung von Testfällen
- 20.10.4. Strategien für die effektive Auswahl und Einführung von KI-gestützten QA-Tools

Positionieren Sie sich auf dem Arbeitsmarkt mit einem 100%igen Online-Programm, das sich an Ihre Bedürfnisse anpasst und ein intensives und solides Studium ermöglicht"

06

Methodik

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: ***das Relearning***.
Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem ***New England Journal of Medicine*** als eines der effektivsten angesehen.

Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen hinter sich lässt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"

Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

Mit TECH werden Sie eine Art des Lernens erleben, die an den Grundlagen der traditionellen Universitäten auf der ganzen Welt rüttelt"

Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.

Der Student wird durch gemeinschaftliche Aktivitäten und reale Fälle lernen, wie man komplexe Situationen in realen Geschäftsumgebungen löst.

Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

> *Unser Programm bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein"*

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit Jurastudenten das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernen. Sie bestand darin, ihnen reale komplexe Situationen zu präsentieren, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen konnten, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode. Während des gesamten Kurses werden die Studenten mit mehreren realen Fällen konfrontiert. Sie müssen ihr gesamtes Wissen integrieren, recherchieren, argumentieren und ihre Ideen und Entscheidungen verteidigen.

Relearning Methodology

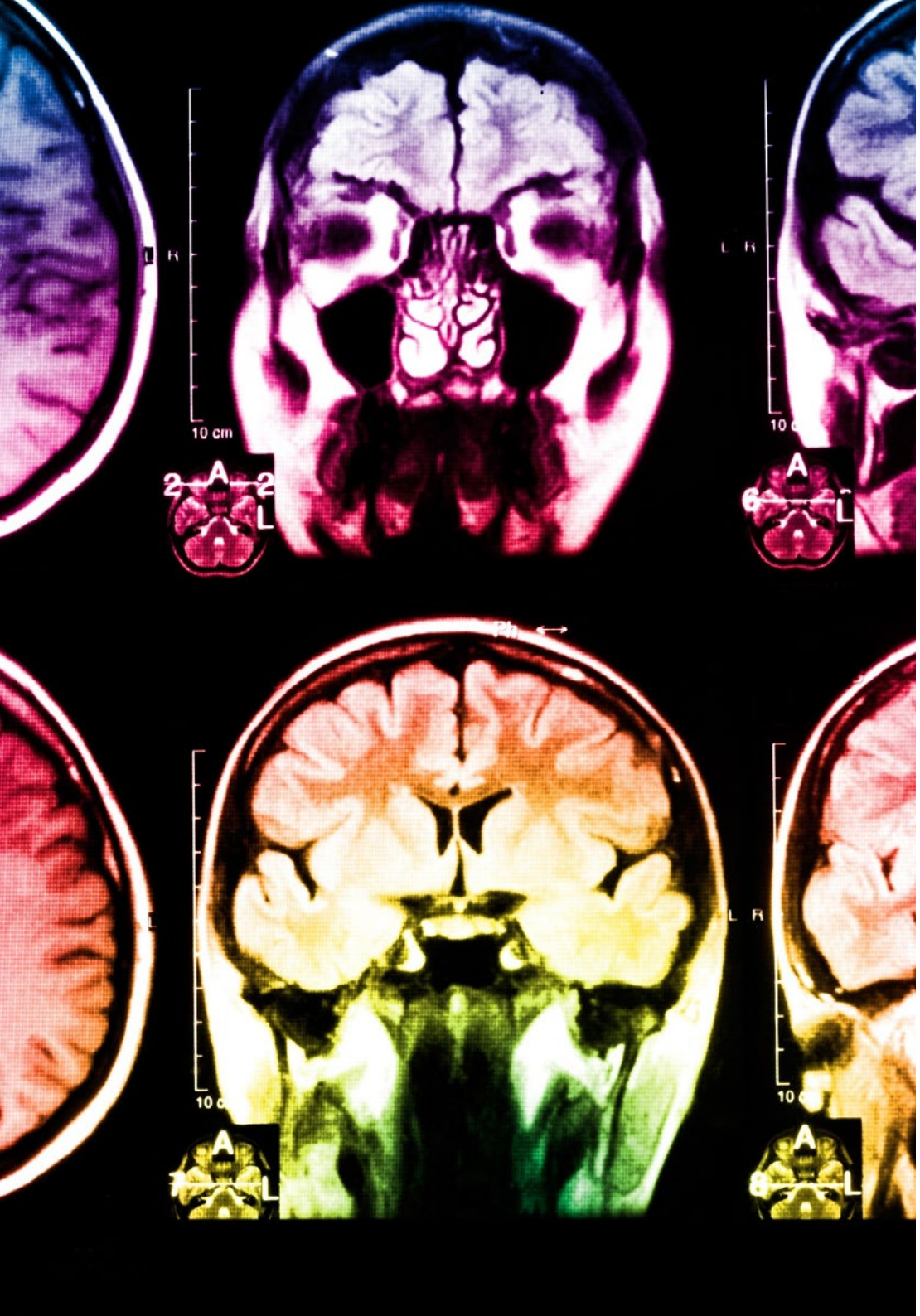
TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

Im Jahr 2019 erzielten wir die besten Lernergebnisse aller spanischsprachigen Online-Universitäten der Welt.

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft zu spezialisieren. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten spanischsprachigen Online-Universität zu verbessern.



In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -instrumente fortgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten Neurocognitive Context-Dependent E-Learning mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.

Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:

Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf das audiovisuelle Format angewendet, um die Online-Arbeitsmethode von TECH zu schaffen. All dies mit den neuesten Techniken, die in jedem einzelnen der Materialien, die dem Studenten zur Verfügung gestellt werden, qualitativ hochwertige Elemente bieten.

Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

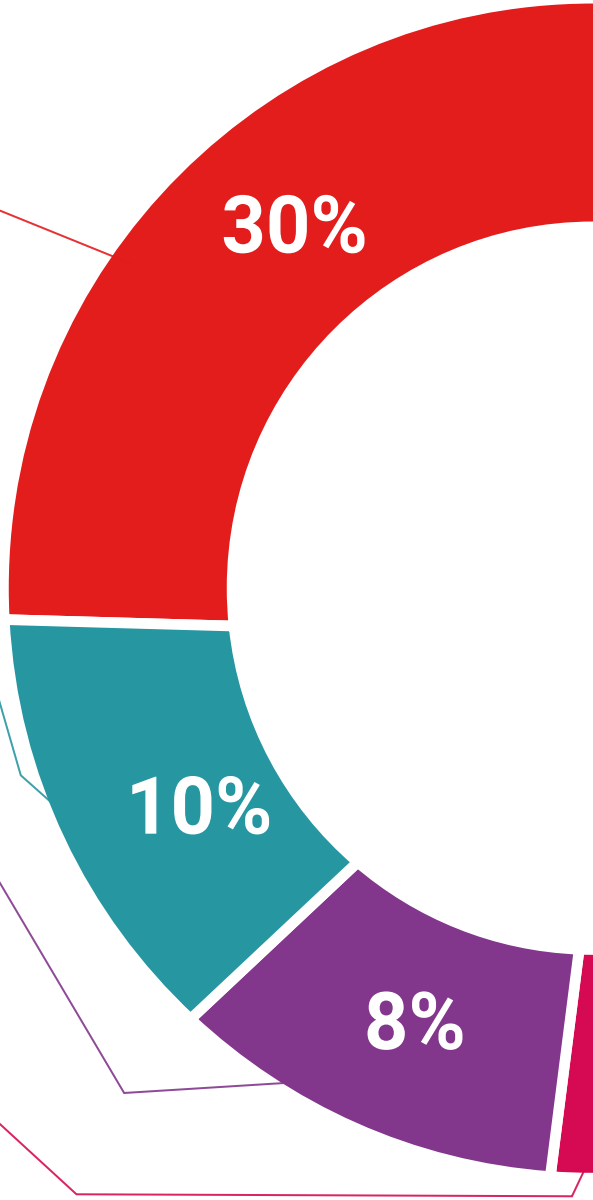
Das sogenannte Learning from an Expert festigt das Wissen und das Gedächtnis und schafft Vertrauen für zukünftige schwierige Entscheidungen.

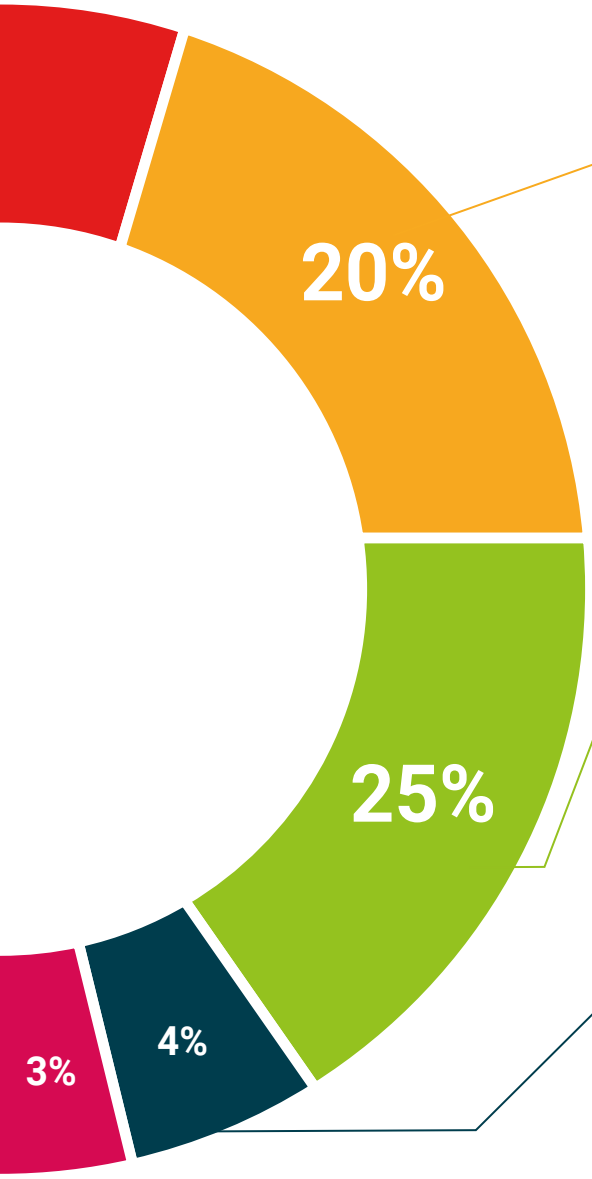
Übungen für Fertigkeiten und Kompetenzen

Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.

Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u. a. In der virtuellen Bibliothek von TECH hat der Student Zugang zu allem, was er für seine Fortbildung benötigt.





Case Studies

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.

Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

Dieses einzigartige Bildungssystem für die Präsentation multimedialer Inhalte wurde von Microsoft als "Europäische Erfolgsgeschichte" ausgezeichnet.

Testing & Retesting

Die Kenntnisse des Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass der Student überprüfen kann, wie er seine Ziele erreicht.

07

Qualifizierung

Der Privater Masterstudiengang in Künstliche Intelligenz in der Programmierung garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Technologischen Universität ausgestellten Diplom.

Schließen Sie dieses Programm erfolgreich ab und erhalten Sie Ihren Universitätsabschluss ohne lästige Reisen oder Formalitäten”

Dieser **Künstliche Intelligenz in der Programmierung** enthält das vollständigste und aktuellste Programm auf dem Markt.

Sobald der Student die Prüfungen bestanden hat, erhält er/sie per Post* mit Empfangsbestätigung das entsprechende Diplom, ausgestellt von der **TECH Technologischen Universität.**

Das von **TECH Technologische Universität** ausgestellte Diplom drückt die erworbene Qualifikation aus und entspricht den Anforderungen, die in der Regel von Stellenbörsen, Auswahlprüfungen und Berufsbildungsausschüssen verlangt werden.

Titel: **Privater Masterstudiengang in Künstliche Intelligenz in der Programmierung**

Modalität: **online**

Dauer: **12 Monate**


tech technologische universität

Verleiht dieses

DIPLOM

an

Herr/Frau _____, mit Ausweis-Nr. _____

Für den erfolgreichen Abschluss und die Akkreditierung des Programms

PRIVATER MASTERSTUDIENGANG

in

Künstliche Intelligenz in der Programmierung

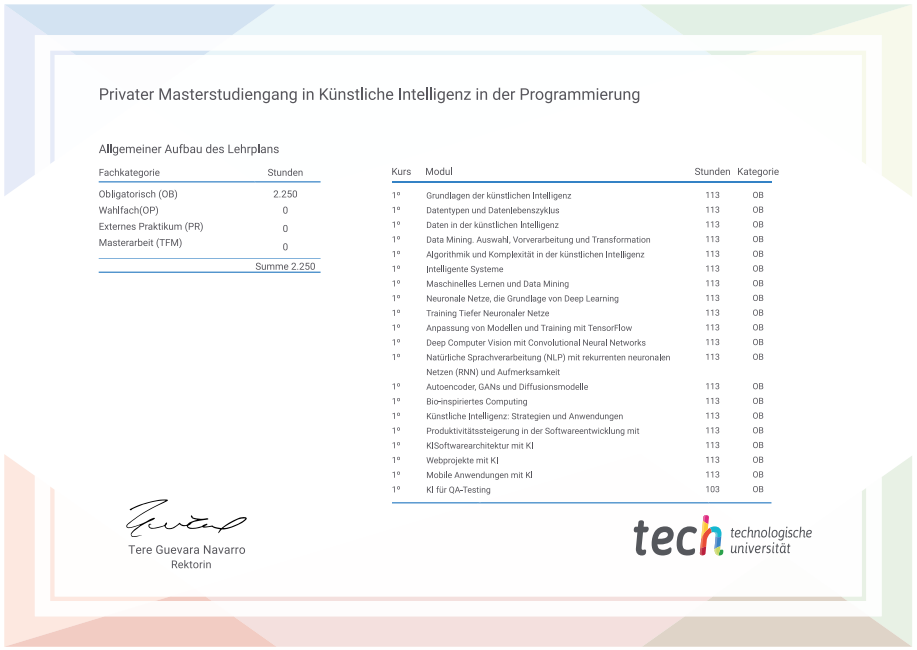
Es handelt sich um einen von dieser Universität verliehenen Abschluss, mit einer Dauer von 2.250 Stunden, mit Anfangsdatum tt/mm/jjjj und Enddatum tt/mm/jjjj.

TECH ist eine private Hochschuleinrichtung, die seit dem 28. Juni 2018 vom Ministerium für öffentliche Bildung anerkannt ist.

Zum 17. Juni 2020

Tere Guevara Navarro
Rektorin

einzigartiger Code TECH: AFWOR23S techtitute.com/titel


Privater Masterstudiengang in Künstliche Intelligenz in der Programmierung

Allgemeiner Aufbau des Lehrplans

Fachkategorie	Stunden
Obligatorisch (OB)	2.250
Wahlfach(OP)	0
Externes Praktikum (PR)	0
Masterarbeit (TFM)	0
	Summe 2.250

Kurs	Modul	Stunden	Kategorie
1°	Grundlagen der künstlichen Intelligenz	113	OB
1°	Datentypen und Datenlebenszyklus	113	OB
1°	Daten in der künstlichen Intelligenz	113	OB
1°	Data Mining. Auswahl, Vorverarbeitung und Transformation	113	OB
1°	Algorithmik und Komplexität in der künstlichen Intelligenz	113	OB
1°	Intelligente Systeme	113	OB
1°	Maschinelles Lernen und Data Mining	113	OB
1°	Neuronale Netze, die Grundlage von Deep Learning	113	OB
1°	Training Tiefer Neuronaler Netze	113	OB
1°	Anpassung von Modellen und Training mit TensorFlow	113	OB
1°	Deep Computer Vision mit Convolutional Neural Networks	113	OB
1°	Natürliche Sprachverarbeitung (NLP) mit rekurrenten neuronalen Netzen (RNN) und Aufmerksamkeit	113	OB
1°	Autoencoder, GANs und Diffusionsmodelle	113	OB
1°	Bio-inspiriertes Computing	113	OB
1°	Künstliche Intelligenz: Strategien und Anwendungen	113	OB
1°	Produktivitätssteigerung in der Softwareentwicklung mit KISoftwarearchitektur mit KI	113	OB
1°	Webprojekte mit KI	113	OB
1°	Mobile Anwendungen mit KI	113	OB
1°	KI für QA-Testing	103	OB

Tere Guevara Navarro
Rektorin

tech technologische universität

» *Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH EDUCATION die notwendigen Vorkehrungen treffen, um diese gegen eine zusätzliche Gebühr zu beschaffen.

tech technologische universität

Privater Masterstudiengang
Künstliche Intelligenz in der Programmierung

- Modalität: **online**
- Dauer: **12 Monate**
- Qualifizierung: **TECH Technologische Universität**
- Zeitplan: **in Ihrem eigenen Tempo**
- Prüfungen: **online**

Privater Masterstudiengang

Künstliche Intelligenz in der Programmierung

tech technologische universität