校级 硕士
编程中的人工智能

tech 科学技术大学

tech 科学技术大学

校级硕士
编程中的人工智能

- 模式:**在线**
- 时长: **7个月**
- 学位: **TECH 科技大学**
- 课程表:**自由安排时间**
- 考试模式:**在线**

网页链接: www.techtitute.com/cn/information-technology/professional-master-degree/master-artificial-intelligence-programming

目录

01

介绍

人工智能(IA) 能够自动执行复杂的任务，根据数据做出决策，并从模式中学习，因此已成为编程界的主流。事实上，人工智能提供的工具和技术可以创建更智能,更高效的系统。从提高程序准确性的机器学习算法，到开发能够实时决策的自主系统，人工智能从根本上改变了代码的设计和执行方式。在此背景下，TECH 设计了一门学术课程，通过革命性的Relearning方法，让毕业生有机会沉浸在这一领域的最新进展中。

人工智能编程课程将为你提供一个全面的视角让你了解人工智能如何影响和改进软件开发的每一个阶段"

人工智能在编程中的重要性在于它能够增强流程并使其自动化，优化软件开发并提高解决复杂问题的效率。能够分析大量数据并找到最佳解决方案，在优化算法，创建更直观的界面和解决不同领域的复杂问题等领域取得了重大进展。

因此，TECH设计了这个校级硕士，作为扩大计算机科学家专业机会和职业发展的战略解决方案。将通过人工智能提高软件开发的生产力，探索自动化流程,优化代码和加速创建智能应用程序的技术和工具。

此外，课程还将重点关注人工智能在 质量保证测试领域的关键作用，实施人工智能算法和方法来提高测试质量,准确性和覆盖率，更高效地检测和纠正错误还将深化机器学习和自然语言处理功能与网络开发的整合，创建能够适应和提供个性化用户体验的智能网站。

还将深入研究人工智能技术，以提高移动应用程序的可用性,交互性和功能性，创建适应用户行为的智能和预测性应用程序。此外，还将深入分析人工智能软件的架构，包括有助于集成人工智能算法并将其部署到生产环境中的各种模型。

为了培养高度胜任的人工智能专家，TECH 根据独特的 Relearning方法设计了一门综合课程。这种方法可以让学生通过重复关键概念来巩固对知识的理解。

这个**编程中的人工智能校级硕士**包含市场上最完整和最新的课程。主要特点是：

- 由人工智能编程专家介绍案例研究的发展情况
- 内容图文并茂，示意性强，实用性强为那些视专业实践至关重要的学科提供了科学和实用的信息
- 包括自我评估的实践过程以推进学习并特别强调创新的方法论
- 特别强调创新的方法论
- 提供理论课程，专家解答问题争议话题的讨论论坛以及个人思考作业等
- 可以从任何联网的固定或移动设备上观看内容

> 你将领导适应不断发展的技术市场需求的创新项目。你还在等什么呢?现在就报名吧"

借助最具创新性的多媒体资源，你将沉浸在软件架构的基础知识中包括性能，可扩展性和可维护性"

想专门从事人工智能工作?通过本课程的学习，你将掌握Cloud计算中部署流程的优化和人工智能的整合。

你将通过全面的学术课程，深入学习如何在 Visual Studio Code 中集成人工智能元素，以及如何使用ChatGPT优化代码。

这门课程的教学人员包括来自这个行业的专业人士，他们将自己的工作经验带到了这一培训中还有来自领先公司和著名大学的公认专家。

通过采用最新的教育技术制作的多媒体内容，专业人士将能够进行情境化学习即通过模拟环境进行沉浸式培训以应对真实情况。

这门课程的设计集中于基于问题的学习，通过这种方式专业人士需要在整个学年中解决所遇到的各种实践问题。为此，你将得到由知名专家制作的新型交互式视频系统的帮助。

02

目标

这门课程的主要目标是向专业人员提供该领域最前沿的知识，促进他们的全面培训。因此，他们将有机会参加一个独家的完全在线的学术途径。毕业生将掌握有用的前沿技能，从人工智能驱动的软件开发，到具有智能性和适应性的网络项目和移动应用程序的设计与实施。通过这门课程，计算机科学家将超越传统编程的限制，成为技术革命的积极参与者。

通过TECH，你将了解从创建测试用例到检测错误的整个测试生命周期"

总体目标

- 培养配置和管理高效开发环境的技能，为人工智能项目的实施提供坚实的基础
- 掌握质量测试的规划，执行和自动化技能，并结合人工智能工具来检测和纠正错误
- 在设计大规模计算系统时了解并应用性能可扩展性和可维护性原则
- 熟悉最重要的设计模式并将其有效地应用于软件架构

具体目标

模块 1. 人工智能基础

- 分析人工智能从开始到现在的历史演变，确定关键的里程碑和发展
- 了解神经网络的功能及其在人工智能学习模型中的应用
- 研究遗传算法的原理和应用，分析其在解决复杂问题中的作用
- 分析词库，词汇表和分类法在构建和处理人工智能系统数据方面的重要性
- 探索语义网的概念及其对数字环境中信息组织和理解的影响

模块 2. 数据类型和周期

- 了解统计学的基本概念及其在数据分析中的应用
- 从定量数据到定性数据，识别和分类不同类型的统计数据
- 分析数据从生成到处置的生命周期，识别关键阶段
- 探索数据生命周期的初始阶段，强调数据规划和数据结构的重要性
- 研究数据收集过程，包括收集方法，工具和渠道
- 探索 Datawarehouse 概念，重点是其构成要素和设计
- 分析与数据管理，遵守隐私和安全法规以及最佳实践相关的监管问题

模块 3. 人工智能中的数据

- 掌握数据科学的基础知识，包括信息分析的工具，类型和来源
- 探索利用数据挖掘和可视化技术将数据转化为信息的过程
- 学习datasets的结构和特征，理解其在准备和利用数据用于人工智能模型时的重要性
- 分析监督和非监督模型，包括方法和分类
- 在数据处理和加工中使用特定工具和最佳实践，确保人工智能实施的效率和质量

模块 4. 数据挖掘。选择，预处理和转换

- 掌握统计推理技术理解并在数据挖掘中应用统计方法
- 对数据集进行详细的探索性分析以确定相关模式，异常现象和趋势
- 培养数据准备技能，包括数据清理，整合和格式化以便用于数据挖掘
- 实施有效策略处理数据集中的缺失值，根据具体情况应用估算或消除方法
- 利用过滤和平滑技术识别并减少数据中的噪音以提高数据集的质量
- 解决Big Data环境中的数据预处理问题

模块 5. 人工智能中的算法与复杂性

- 介绍算法设计策略，让学生扎实了解解决问题的基本方法
- 分析算法的效率和复杂性，应用分析技术评估时间和空间方面的性能
- 研究和应用排序算法，了解工作原理并比较它们在不同情况下的效率
- 探索基于树的算法，了解其结构和应用
- 研究具有堆Heaps的算法，分析其实现以及在高效处理数据方面的实用性
- 分析基于图形的算法，探索其在表示和解决涉及复杂关系的问题中的应用
- 学习Greedy算法，了解其逻辑和在解决优化问题中的应用
- 研究并应用backtracking 技术系统地解决问题分析其在各种情况下的有效性

模块 6. 智能系统

- 探索代理理论，了解其工作原理的基本概念及其在人工智能和软件工程中的应用
- 研究知识表示法，包括分析本体及其在组织结构化信息中的应用
- 分析语义网的概念及其对数字环境中信息组织和检索的影响
- 评估和比较不同的知识表示法，整合它们以提高智能系统的效率和准确性
- 研究语义推理器，基于知识的系统和专家系统，了解在智能决策中的功能和应用

模块 7. 机器学习和数据挖掘

- 介绍知识发现过程和机器学习的基本概念
- 研究作为监督学习模型的决策树，了解其结构和应用
- 使用特定技术评估分类器衡量其在数据分类方面的性能和准确性
- 研究神经网络，了解其运行和架构以解决复杂的机器学习问题
- 探索贝叶斯方法及其在机器学习中的应用，包括贝叶斯网络和贝叶斯分类器
- 分析从数据中预测数值的回归和连续反应模型
- 研究clustering技术以识别无标签数据集的模式和结构
- 探索文本挖掘和自然语言处理（NLP），了解如何应用机器学习技术来分析和理解文本

模块 8.神经网络，Deep Learning的基础

- 掌握深度学习的基本原理，了解其在 Deep Learning中的重要作用
- 探索神经网络的基本操作了解其在模型构建中的应用
- 分析神经网络中使用的不同层，学习如何适当选择这些层
- 了解如何有效连接各层和操作以设计复杂而高效的神经网络架构
- 使用训练器和优化器来调整和提高神经网络的性能
- 探索生物神经元与人工神经元之间的联系加深对模型设计的理解
- Fine Tuning神经网络的超参数，优化其在特定任务中的表现

模块 9. 深度神经网络训练

- 解决深度神经网络训练中的梯度相关问题
- 探索和应用不同的优化器以提高模型的效率和收敛性
- 设置学习率动态调整模型的收敛速度
- 在培训期间通过具体策略了解和解决过度调整问题
- 应用实用指南确保高效和有效地训练深度神经网络
- 将Transfer Learning作为一种先进技术来提高模型在特定任务中的性能
- 探索和应用数据增强技术丰富数据集提高模型的泛化能力
- 利用Transfer Learning开发实际应用解决现实世界中的问题
- 了解并应用正则化技术以提高深度神经网络的泛化能力并避免过度拟合

模块 10. 使用TensorFlow进行模型定制和训练

- 掌握TensorFlow的基础知识及其与NumPy的集成以实现高效的数据处理和计算
- 利用TensorFlow的高级功能定制训练模型和算法
- 探索API tfdata应用程序接口高效管理和操作数据集
- 在TensorFlow中实现用于存储和访问大型数据集的TFRecord格式
- 使用Keras预处理层方便构建自定义模型
- 探索TensorFlow 数据集项目访问预定义数据集提高开发效率
- 利用TensorFlow开发Deep Learning应用程序，将本模块所学知识进行整合
- 在现实世界中实际应用所学的所有概念使用TensorFlow建立和训练自定义模型

模块 11. 使用卷积神经网络的Deep Computer Vision

- 了解视觉皮层的结构及其与Deep Computer Vision的相关性
- 探索和应用卷积层从图像中提取关键特征
- 使用Keras在Computer Vision 模型中实施聚类层及其应用
- 分析各种卷积神经网络（CNN）架构及其在不同情况下的适用性
- 使用Keras库开发并实施CNN ResNet以提高模型的效率和性能
- 使用预训练的Keras模型利用迁移学习完成特定任务
- 在Deep Computer Vision环境中应用分类和定位技术
- 利用卷积神经网络探索物体检测和物体跟踪策略
- 采用语义分割技术详细了解图像中的物体并对其进行分类

模块 12.用自然递归网络(RNN)和注意力进行自然语言处理(NLP)

- 培养使用递归神经网络（RNN）生成文本的技能
- 在文本情感分析中应用RNN进行观点分类
- 理解并在自然语言处理模型中应用注意力机制
- 在特定NLP任务中分析和使用Transformer模型
- 探索Transformers 模型在图像处理和计算机视觉中的应用
- 熟悉Hugging的Transformers库以便高效地实施高级模型
- 比较不同的Transformers 库评估它们对特定任务的适用性
- 开发NLP的实际应用整合RNN和注意力机制以解决现实世界中的问题

模块 13. 自动编码器，GAN和扩散模型

- 使用自动编码器,GAN和扩散模型开发高效的数据表示
- 使用不完全线性自动编码器执行 PCA优化数据表示
- 执行并理解自动堆叠编码器的操作
- 探索和应用卷积自动编码器实现视觉数据的高效表达
- 分析和应用稀疏自动编码器在数据表示中的有效性
- 使用自动编码器从MNIST数据集生成时尚图像
- 了解生成对抗网络（GAN）和扩散模型的概念
- 在数据生成中实施并比较扩散模型和 GAN的性能

模块 14. 生物启发式计算

- 介绍生物启发计算的基本概念
- 探索社会自适应算法作为生物启发计算的关键方法
- 分析遗传算法中的空间探索-开发策略
- 研究优化背景下的进化计算模型
- 继续详细分析进化计算模型
- 将进化编程应用于特定的学习问题
- 在生物启发计算框架内解决多目标问题的复杂性
- 探索神经网络在生物启发计算领域的应用
- 深化神经网络在生物启发计算中的实施和应用

模块 15. 人工智能：策略和应用

- 制定在金融服务中实施人工智能的策略
- 分析人工智能对提供医疗服务的影响
- 识别和评估在卫生领域使用人工智能的相关风险
- 评估工业领域使用人工智能的潜在风险
- 在工业中应用人工智能技术提高生产力
- 设计人工智能解决方案优化公共管理流程
- 评估人工智能技术在教育领域的实施情况
- 在林业和农业中应用人工智能技术提高生产力
- 通过策略性使用人工智能优化人力资源流程

模块 16. 利用人工智能提高软件开发效率

- 深入研究在Visual Studio Code中实施必备的人工智能扩展以提高工作效率并促进软件开发
- 扎实了解人工智能的基本概念及其在 软件开发中的应用，包括机器学习算法，自然语言处理，神经网络等
- 掌握优化开发环境的设置，确保学生能够创建有利于人工智能项目的环境
- 使用ChatGPT自动识别和纠正可能的代码改进，鼓励更高效的编程实践
- 促进不同编程专业人员（从程序员到数据工程师再到用户体验设计师）之间的合作，以开发有效且符合道德规范的人工智能 软件 解决方案

模块 17. 质量保证 测试的 软件架构

- 培养设计稳健测试计划的技能，涵盖不同类型的测试，确保软件质量
- 识别并分析不同类型的 软件框架，如单体框架，微服务框架或面向服务框架
- 全面了解设计可扩展并能处理大量数据的计算机系统的原理和技术
- 运用高级知识实施人工智能驱动的数据结构，优化软件性能和效率
- 开发安全的开发实践，重点是避免漏洞以确保架构层面的软件安全

模块 18. 人工智能网络项目

- 培养实施网络项目的综合技能，从 前端 设计到 后台 优化并加入人工智能元素
- 优化网站部署流程，采用技术和工具来提高速度和效率
- 将人工智能融入Cloud计算，使学生能够创建高度可扩展和高效的网络项目
- 掌握在网络项目中发现可有效应用人工智能的具体问题和机会的能力，如文本处理，定制，内容推荐等
- 鼓励学生了解人工智能的最新趋势和发展，以便在网络项目中正确应用

模块 19. 人工智能移动应用

- 应用先进的简洁架构、数据源和存储库 概念，确保人工智能移动应用程序具有稳健的模块化结构
- 培养使用人工智能设计交互式屏幕，图标和图形资产的技能以增强移动应用程序的用户体验
- 深化移动应用框架的配置，使用 Github Copilot 加快开发进程
- 利用人工智能优化移动应用程序，提高性能，同时考虑到资源管理和数据使用情况
- 利用人工智能对移动应用程序进行质量测试，让学生能够发现问题并调试错误

模块 20. 用于QA测试的人工智能

- 掌握设计可扩展并能处理大量数据的计算机系统的原则和技术
- 运用高级知识实施人工智能驱动的数据结构，优化软件 性能和效率
- 了解并应用安全开发实践，重点是避免注入等漏洞以确保 软件 在架构层面的安全性
- 生成自动测试，特别是在网络和移动环境中集成人工智能工具以提高流程效率
- 使用先进的人工智能驱动的质量保证工具更高效地检测错误并持续改进软件

> 通过这个独一无二的100%在线大学学位你将掌握未来的技术。只有TECH！"

03

能力

这门课程将使毕业生掌握人工智能（IA）方面的最新具体技能，从而在计算机开发行业中占据重要优势。专业人员不仅能够设计和开发先进的软件，还能在从网络和移动项目到大型软件架构等各种应用中有效实施人工智能解决方案。此外，提高开发效率和 质量保证测试最佳实践将确保计算机科学家做好准备，迎接现实世界的挑战，并在不断发展的领域中脱颖而出。

感谢这门大学的课程，你将能够在网络项目和移动应用程序中实施人工智能算法"

总体能力

- 在Visual Studio Code中应用人工智能扩展和 非代码设计技术，提高软件开发效率使用ChatGPT来优化和提高代码质量，应用先进的编程实践
- 实施网络项目，从 工作区创建到部署，将人工智能与 前端和 后端相结合
- 开发人工智能驱动的移动应用程序，从环境配置到高级功能创建和图形资产管理
- 应用先进的存储概念和人工智能驱动的数据结构提高系统的效率和可扩展性
- 包括安全开发实践，避免注入等漏洞，以确保所开发软件的完整性和安全性

> 你将能够通过人工智能设计个性化和直观的用户体验。现在报名吧！”

具体能力

- 应用人工智能技术和策略提高零售业的效率
- 使用自动编码器实施去噪技术
- 为自然语言处理(NLP)任务有效创建训练数据集
- 使用Keras运行聚类层及其在 深度计算机视觉 模型中的应用
- 使用 TensorFlow 功能和图形优化自定义模型的性能
- 优化 聊天机器人和虚拟助手的开发和应用，了解它们的工作原理和潜在应用
- 掌握预训练层的重复使用优化并加速训练过程
- 应用实践中学到的概念，构建第一个神经网络
- 使用Keras库激活多层感知器(MLP)
- 应用数据探索和预处理技术，识别和准备数据以便在机器学习模型中有效使用
- 利用开发语义模型的特定工具，研究创建本体的语言和软件
- 开发数据清理技术确保后续分析所用信息的质量和准确性
- 掌握优化开发环境的设置，确保学生能够创建有利于人工智能项目的环境
- 应用特定技术，使用ChatGPT自动识别和纠正可能的代码改进，鼓励更高效的编程实践
- 创建自动化测试，特别是在网络和移动环境中，集成人工智能工具以提高流程效率使用先进的人工智能驱动的质量保证工具，更高效地检测错误并持续改进软件
- 将人工智能融入Cloud计算，使学生能够创建高度可扩展和高效的网络项目建立移动应用程序框架，使用 Github Copilot加快开发进程

04

课程管理

TECH 致力于精英教学，精心挑选了负责开发该学位课程的教师。因此，这门学术课程拥有一支经验丰富的教师队伍，他们在人工智能编程应用方面有着杰出的背景。这样，攻读该硕士学位的学生就能获得一流的教育体验，通过各种视听媒体获得独特的知识组合，从而更有效，更动态地整合知识。

从该领域最优秀的专家那里了解人工智能应用于编程的最新趋势"

管理人员

Peralta Martín-Palomino, Arturo 博士

- Prometeus Global Solutions的首席执行官和首席技术官
- Korporate Technologies的首席技术官
- IA Shepherds GmbH 首席技术官
- 联盟医疗顾问兼业务策略顾问
- DocPath设计与开发总监
- -卡斯蒂利亚拉曼恰大学计算机工程博士
- 卡米洛-何塞-塞拉大学的经济学，商业和金融学博士
- -卡斯蒂利亚拉曼恰大学心理学博士
- 伊莎贝尔一世大学行政工商管理硕士
- 伊莎贝尔一世大学商业管理与营销硕士
- Hadoop培训大数据专家硕士
- -卡斯蒂利亚拉曼恰大学高级信息技术硕士
- 成员：SMILE研究组

Castellanos Herreros, Ricardo 先生

- OWQLO首席技术官
- 计算机系统工程专家和机器 学习工程师
- 自由职业技术顾问
- 为eDreams, Fnac, IAr Europa, Bankia, Cetelem, Banco Santander, Santillana, Groupón 和Grupo Planeta 开发移动应用程序
- 开放银行和桑坦德银行网站开发人员
- 卡斯蒂利亚拉曼恰大学计算机系统技术工程师

05

结构和内容

人工智能编程课程以其全面的方法脱颖而出，不仅涉及智能算法的实施，还涉及软件开发中生产力的提高以及人工智能在质量保证测试，网络项目，移动应用和软件架构等关键领域的应用。该课程将技术技能，先进工具和人工智能在各个开发阶段的实际应用相结合，使其成为一个领先的课程，让专业人员全面深入地了解人工智能在编程中的应用。

你将研究人工智能在网络项目中的实际应用，包括前端和后端开发"

模块 1. 人工智能基础

1.1. 人工智能的历史
- 1.1.1. 我们是从什么时候开始谈论人工智能的?
- 1.1.2. 电影参考资料
- 1.1.3. 人工智能的重要性
- 1.1.4. 支持人工智能的技术

1.2. 游戏中的人工智能
- 1.2.1. 博弈论
- 1.2.2. Minimax和Alpha-Beta修剪
- 1.2.3. 仿真: Monte Carlo

1.3. 神经网络
- 1.3.1. 生物学基础
- 1.3.2. 计算模型
- 1.3.3. 有监督和无监督的神经元网络
- 1.3.4. 简单的感知器
- 1.3.5. 多层感知器

1.4. 遗传算法
- 1.4.1. 历史
- 1.4.2. 生物学基础
- 1.4.3. 问题编码
- 1.4.4. 最初的人口生成
- 1.4.5. 主要算法和遗传算子
- 1.4.6. 对个人的评价:健身

1.5. 术语表, 词汇表, 分类法
- 1.5.1. 词汇
- 1.5.2. 分类法
- 1.5.3. 叙词表
- 1.5.4. 体论
- 1.5.5. 知识表示:语义网

1.6. 语义网
- 1.6.1. 规格: RDF, RDFS和OWL
- 1.6.2. 推论/推理
- 1.6.3. 关联数据

1.7. 专家系统和DSS
- 1.7.1. 专家系统
- 1.7.2. 摄影的支持系统

1.8. 聊天机器人 和虚拟助理
- 1.8.1. 助手的类型:语音和文字助手
- 1.8.2. 发展助理的基础部分:意图, 实体和对话流
- 1.8.3. 集成:网络, Slack, Whatsapp, Facebook
- 1.8.4. 培养助手的工具:Dialog Flow, Watson Assistant

1.9. 人工智能实施策略

1.10. 人工智能的未来
- 1.10.1. 我们了解如何通过算法检测情绪
- 1.10.2. 创造个性:语言, 表达和内容
- 1.10.3. 人工智能的发展趋势
- 1.10.4. 反思

模块 2. 数据类型和周期

2.1. 统计数据
- 2.1.1. 统计:描述性统计, 统计推断
- 2.1.2. 总体, 样本, 个体
- 2.1.3. 变量:定义, 测量尺度

2.2. 统计数据类型
- 2.2.1. 根据类型
  - 2.2.1.1. 定量:连续数据和离散数据
  - 2.2.1.2. 定性:二项式数据, 名义数据和有序数据
- 2.2.2. 根据形式
  - 2.2.2.1. 数字
  - 2.2.2.2. 文本
  - 2.2.2.3. 逻辑
- 2.2.3. 根据来源
  - 2.2.3.1. 一级
  - 2.2.3.2. 二级

2.3. 数据生命周期
- 2.3.1. 周期的段
- 2.3.2. 周期里程碑
- 2.3.3. FIAR原则

2.4. 周期的初始阶段
- 2.4.1. 定义目标
- 2.4.2. 确定必要的资源
- 2.4.3. 甘特图
- 2.4.4. 数据结构

2.5. 数据收集
- 2.5.1. 收集方法
- 2.5.2. 收集工具
- 2.5.3. 收集渠道

2.6. 数据清理
- 2.6.1. 数据清理阶段
- 2.6.2. 数据质量
- 2.6.3. 数据操作(使用 R)

2.7. 数据分析,解释和结果评估
- 2.7.1. 统计措施
- 2.7.2. 关系指数
- 2.7.3. 数据挖掘

2.8. 数据仓库 (Datawarehouse)
- 2.8.1. 整合的元素
- 2.8.2. 设计
- 2.8.3. 需要考虑的问题

2.9. 可用性数据
- 2.9.1. 访问
- 2.9.2. 实用性
- 2.9.3. 安全

2.10. 监管方面
- 2.10.1. 数据保护法
- 2.10.2. 最佳实践
- 2.10.3. 其他规范方面

模块 3. 人工智能中的数据

3.1. 数据科学
- 3.1.1. 数据科学
- 3.1.2. 数据科学的高级工具

3.2. 数据,信息和知识
- 3.2.1. 数据,信息和知识
- 3.2.2. 数据类型
- 3.2.3. 数据源

3.3. 从数据到信息
- 3.3.1. 数据分析
- 3.3.2. 分析类型
- 3.3.3. 从数据集中提取信息

3.4. 通过可视化提取信息
- 3.4.1. 可视化作为分析工具
- 3.4.2. 可视化方法
- 3.4.3. 查看数据集

3.5. 数据质量
- 3.5.1. 质量数据
- 3.5.2. 数据清理
- 3.5.3. 基本数据预处理

3.6. 数据集
- 3.6.1. 丰富数据集
- 3.6.2. 维度的祸害
- 3.6.3. 修改我们的数据集

3.7. 不平衡
- 3.7.1. 阶级不平衡
- 3.7.2. 不平衡缓解技术
- 3.7.3. 平衡数据集

3.8. 无监督模型
- 3.8.1. 无监督模型
- 3.8.2. 方法
- 3.8.3. 使用无监督模型进行分类

3.9. 监督模型
- 3.9.1. 监督模型
- 3.9.2. 方法
- 3.9.3. 使用监督模型进行分类

3.10. 工具和好的做法
- 3.10.1. 数据科学的正确实践
- 3.10.2. 最佳模型
- 3.10.3. 有用的工具

模块 4. 数据挖掘选择，预处理和转换

4.1. 统计推断
- 4.1.1. 描述性统计对统计推断
- 4.1.2. 参数化程序
- 4.1.3. 非参数过程

4.2. 探索性分析
- 4.2.1. 描述性分析
- 4.2.2. 视觉化
- 4.2.3. 数据准备

4.3. 数据准备
- 4.3.1. 数据整合和清理
- 4.3.2. 数据标准化
- 4.3.3. 转换属性

4.4. 缺失值
- 4.4.1. 缺失值的处理
- 4.4.2. 最大似然插补方法
- 4.4.3. 使用机械学习估算缺失值

4.5. 数据中的噪音
- 4.5.1. 噪声类别和属性
- 4.5.2. 噪声过滤
- 4.5.3. 噪音的影响

4.6. 维度的祸害
- 4.6.1. 过度采样
- 4.6.2. 采样不足
- 4.6.3. 多维数据缩减

4.7. 从连续属性到离散属性
- 4.7.1. 连续数据与离散数据
- 4.7.2. 离散化过程

4.8. 数据
- 4.8.1. 数据选择
- 4.8.2. 前景与选择标准
- 4.8.3. 挑选方法

4.9. 选择阶段
- 4.9.1. 选择阶段的方法
- 4.9.2. 原型的选择
- 4.9.3. 选择阶段的高级方法

4.10. 大数据环境的数据预处理

模块 5. 人工智能中的算法与复杂性

5.1. 算法设计策略简介
- 5.1.1. 递归
- 5.1.2. 分而治之
- 5.1.3. 其他策略

5.2. 算法的效率与分析
- 5.2.1. 效率措施
- 5.2.2. 测量输入的大小
- 5.2.3. 测量执行时间
- 5.2.4. 最坏情况，最好情况和中间情况
- 5.2.5. 渐近符号
- 5.2.6. 非递归算法的数学分析准则
- 5.2.7. 递归算法的数学分析
- 5.2.8. 算法的实证分析

5.3. 排序算法
- 5.3.1. 协调概念
- 5.3.2. 冒泡排序
- 5.3.3. 选择排序
- 5.3.4. 插入排序
- 5.3.5. 合并排序 (Merge_Sort)
- 5.3.6. 快速排序 (Quicksort)

5.4. 带树的算法
- 5.4.1. 树的概念
- 5.4.2. 二叉树
- 5.4.3. 树游览
- 5.4.4. 表示表达式
- 5.4.5. 有序二叉树
- 5.4.6. 平衡二叉树

5.5. 带Heaps的算法
- 5.5.1. Heaps
- 5.5.2. 堆排序算法
- 5.5.3. 优先队列

5.6. 带图的算法
- 5.6.1. 代表
- 5.6.2. 行程宽度
- 5.6.3. 深度游览
- 5.6.4. 拓扑排序

5.7. Greedy的算法
- 5.7.1. Greedy的策略
- 5.7.2. Greedy策略元素
- 5.7.3. 货币兑换
- 5.7.4. 旅人的问题
- 5.7.5. 背包问题

5.8. 搜索最小路径
- 5.8.1. 最短路径的问题
- 5.8.2. 负弧和循环
- 5.8.3. Dijkstra的算法

5.9. 图上的Greedy算法
- 5.9.1. 最小生成树
- 5.9.2. Prim算法
- 5.9.3. Kruskal算法
- 5.9.4. 复杂性分析

5.10. 溯源
- 5.10.1. Backtracking
- 5.10.2. 替代技术

模块 6. 智能系统

6.1. 代理理论
- 6.1.1. 概念的历史
- 6.1.2. 代理定义
- 6.1.3. 人工智能中的代理
- 6.1.4. 软件工程中的代理

6.2. 代理架构
- 6.2.1. 代理的推理过程
- 6.2.2. 反应性
- 6.2.3. 演绎
- 6.2.4. 混合代理
- 6.2.5. 比较

6.3. 信息和知识
- 6.3.1. 数据，信息和知识之间的区别
- 6.3.2. 数据质量评估
- 6.3.3. 数据采集方法
- 6.3.4. 信息获取方式
- 6.3.5. 知识获取方式

6.4. 知识表示
- 6.4.1. 知识表示的重要性
- 6.4.2. 通过其角色定义知识表示
- 6.4.3. 知识表示的特征

6.5. 体论
- 6.5.1. 元数据介绍
- 6.5.2. 体论的哲学概念
- 6.5.3. 体论的计算概念
- 6.5.4. 领域本体和更高层本体
- 6.5.5. 如何建立一个体论?

6.6. 本体语言和本体编写软件
- 6.6.1. 三胞胎 RDF, Turtle和 N
- 6.6.2. RDF模式
- 6.6.3. OWL
- 6.6.4. SPARQL
- 6.6.5. 简介用于创建本体论的不同工具
- 6.6.6. Protégé安装和使用

6.7. 语义网
- 6.7.1. 语义网的现状和未来
- 6.7.2. 语义网应用

6.8. 其他知识表示模型
- 6.8.1. 词汇
- 6.8.2. 全球视野
- 6.8.3. 分类法
- 6.8.4. 叙词表
- 6.8.5. 大众分类法
- 6.8.6. 比较
- 6.8.7. 心理地图

6.9. 知识表示的评估和整合
- 6.9.1. 零阶逻辑
- 6.9.2. 一阶逻辑
- 6.9.3. 描述性逻辑
- 6.9.4. 不同类型逻辑之间的关系
- 6.9.5. Prolog：基于一阶逻辑的程序设计

6.10. 语义推理器，基于知识的系统和专家系统
- 6.10.1. 推理概念
- 6.10.2. 推理机的应用
- 6.10.3. 基于知识的系统
- 6.10.4. MYCIN，专家系统的历史
- 6.10.5. 专家系统的元素和架构
- 6.10.6. 专家系统的创建

模块 7. 机器学习和数据挖掘

7.1. 简介知识发现过程和机器学习的基础概念
- 7.1.1. 知识发现过程的关键概念
- 7.1.2. 知识发现过程的历史视角
- 7.1.3. 知识发现过程的各个阶段
- 7.1.4. 知识发现过程中使用的技术
- 7.1.5. 佳的机器学习模型的特点
- 7.1.6. 机器学习信息的类型
- 7.1.7. 学习的基础概念
- 7.1.8. 无监督学习的基础概念

7.2. 数据探索和预处理
- 7.2.1. 数据处理
- 7.2.2. 数据分析流程中的数据处理
- 7.2.3. 数据类型
- 7.2.4. 数据转换
- 7.2.5. 连续变量的可视化和探索
- 7.2.6. 分类变量的显示和探索
- 7.2.7. 相关性措施
- 7.2.8. 最常见的图形表示法
- 7.2.9. 多变量分析和降维简介

7.3. 决策树
- 7.3.1. ID算法
- 7.3.2. C 算法
- 7.3.3. 过度训练和修剪
- 7.3.4. 结果分析

7.4. 对分类器的评估
- 7.4.1. 混淆矩阵
- 7.4.2. 数值评价矩阵
- 7.4.3. Kappa统计学
- 7.4.4. ROC曲线

7.5. 分类规则
- 7.5.1. 规则评价措施
- 7.5.2. 图形表示法简介
- 7.5.3. 顺序叠加算法

7.6. 神经网络
- 7.6.1. 基础概念
- 7.6.2. 简单的神经网络
- 7.6.3. 反向传播算法
- 7.6.4. 递归神经网络简介

7.7. 贝叶斯方法
- 7.7.1. 概率的基础概念
- 7.7.2. 贝叶斯定理
- 7.7.3. 奈何贝叶斯
- 7.7.4. 贝叶斯网络简介

7.8. 回归和连续反应模型
- 7.8.1. 简单线性回归
- 7.8.2. 多重线性回归
- 7.8.3. 逻辑回归
- 7.8.4. 回归树
- 7.8.5. 支持向量机(SVM)简介
- 7.8.6. 拟合度测量

7.9. 聚类
- 7.9.1. 基础概念
- 7.9.2. 分层Clustering
- 7.9.3. 概率论的方法
- 7.9.4. EM算法
- 7.9.5. B-立方体法
- 7.9.6. 隐式方法

7.10. 文本挖掘和自然语言处理(NLP)
- 7.10.1. 基础概念
- 7.10.2. 语料库的创建
- 7.10.3. 描述性分析
- 7.10.4. 情感分析简介

模块 8. 神经网络, Deep Learning的基础

8.1. Deep Learning
- 8.1.1. 深度学习的类型
- 8.1.2. 深入学习应用
- 8.1.3. 深入学习优点和缺点

8.2. 业务
- 8.2.1. 加
- 8.2.2. 产品
- 8.2.3. 转移

8.3. 图层
- 8.3.1. 输入层
- 8.3.2. 隐藏层
- 8.3.3. 输出层

8.4. 联合层和操作
- 8.4.1. 架构设计
- 8.4.2. 层与层之间的连接
- 8.4.3. 前向传播

8.5. 第一个神经网络的构建
- 8.5.1. 网络设计
- 8.5.2. 设置权重
- 8.5.3. 网络培训

8.6. 训练器和优化器
- 8.6.1. 优化器选择
- 8.6.2. 损失函数的建立
- 8.6.3. 建立指标

8.7. 神经网络原理的应用
- 8.7.1. 激活函数
- 8.7.2. 反向传播
- 8.7.3. 参数设定

8.8. 从生物神经元到人工神经元
- 8.8.1. 生物神经元的功能
- 8.8.2. 知识转移到人工神经元
- 8.8.3. 建立它们俩之间的关系

8.9. 使用Keras实现MLP(多层感知器)
- 8.9.1. 网络结构的定义
- 8.9.2. 模型编译
- 8.9.3. 模型训练

8.10. 微调神经网络的超参数
- 8.10.1. 激活函数选择
- 8.10.2. 设置 学习率
- 8.10.3. 权重的调整

模块 9. 深度神经网络训练

9.1. 梯度问题
- 9.1.1. 梯度优化技术
- 9.1.2. 随机梯度
- 9.1.3. 权重初始化技术

9.2. 预训练层的重用
- 9.2.1. 学习迁移培训
- 9.2.2. 特征提取
- 9.2.3. 深度学习

9.3. 优化
- 9.3.1. 随机梯度下降优化器
- 9.3.2. Adam和RMSprop优化器
- 9.3.3. 矩优化器

9.4. 学习率编程
- 9.4.1. 机器学习速率控制
- 9.4.2. 学习周期
- 9.4.3. 平滑项

9.5. 过拟合
- 9.5.1. 交叉验证
- 9.5.2. 正规化
- 9.5.3. 评估指标

9.6. 实用指南
- 9.6.1. 模型设计
- 9.6.2. 指标和评估参数的选择
- 9.6.3. 假设检验

9.7. Transfer Learning
- 9.7.1. 学习迁移培训
- 9.7.2. 特征提取
- 9.7.3. 深度学习

9.8. 数据扩充
- 9.8.1. 图像变换
- 9.8.2. 综合数据生成
- 9.8.3. 文本转换

9.9. Transfer Learning的实际应用
- 9.9.1. 学习迁移培训
- 9.9.2. 特征提取
- 9.9.3. 深度学习

9.10. 正规化
- 9.10.1. L和L
- 9.10.2. 通过最大熵正则化
- 9.10.3. Dropout

模块 10. 使用TensorFlow进行模型定制和训练

10.1. TensorFlow
- 10.1.1. 使用TensorFlow库
- 10.1.2. 使用TensorFlow进行模型训练
- 10.1.3. TensorFlow中的图操作

10.2. TensorFlow和NumPy
- 10.2.1. 用于TensorFlow的NumPy计算环境
- 10.2.2. 在TensorFlow中使用NumPy数组
- 10.2.3. 用于TensorFlow图形的NumPy运算

10.3. 训练模型和算法定制
- 10.3.1. 使用TensorFlow构建自定义模型
- 10.3.2. 训练参数管理
- 10.3.3. 使用优化技术进行训练

10.4. TensorFlow函数和图形
- 10.4.1. 使用TensorFlow的功能
- 10.4.2. 使用图表来训练模型
- 10.4.3. 利用TensorFlow操作优化图形

10.5. 使用TensorFlow加载和预处理数据
- 10.5.1. 使用TensorFlow加载数据集
- 10.5.2. 使用TensorFlow进行数据预处理
- 10.5.3. 使用TensorFlow工具进行数据操作

10.6. tfdata应用程序接口
- 10.6.1. 使用tfdataAPI进行数据处理
- 10.6.2. 使用tfdata构建数据流
- 10.6.3. 使用tfdataAPI训练模型

10.7. TFRecord格式
- 10.7.1. 使用TFRecordAPI进行数据序列化
- 10.7.2. 使用TensorFlow加载TFRecord文件
- 10.7.3. 使用TFRecord文件进行模型训练

10.8. Keras预处理层
- 10.8.1. 使用Keras预处理API
- 10.8.2. 使用Keras构建预pipelined管道
- 10.8.3. 使用Keras预处理API进行模型训练

10.9. TensorFlow数据集项目
- 10.9.1. 使用TensorFlow数数集加载数据
- 10.9.2. 使用TensorFlow Datasets进行数据预处理
- 10.9.3. 使用TensorFlow数据集训练模型

10.10. 使用TensorFlow构建深度学习应用程序
- 10.10.1. 实际应用
- 10.10.2. 使用 TensorFlow构建深度 学习 应用程序
- 10.10.3. 使用TensorFlow进行模型训练
- 10.10.4. 使用应用程序预测结果

模块 11. 利用卷积神经网络实现深度计算机视觉

11.1. 视觉皮层架构
- 11.1.1. 视觉皮层的功能
- 11.1.2. 计算机视觉理论
- 11.1.3. 图像处理模型

11.2. 卷积层
- 11.2.1. 卷积中权重的重用
- 11.2.2. D 卷积
- 11.2.3. 激活函数

11.3. 池化层以及使用Keras实现池化层
- 11.3.1. Pooling和Striding
- 11.3.2. Flattening
- 11.3.3. Pooling类型

11.4. CNN 架构
- 11.4.1. VGG-架构
- 11.4.2. AlexNet架构
- 11.4.3. ResNet架构

11.5. 使用Keras实现CNNResNet
- 11.5.1. 权重初始化
- 11.5.2. 输入层定义
- 11.5.3. 输出定义

11.6. 使用预训练的Keras模型
- 11.6.1. 预训练模型的特点
- 11.6.2. 预训练模型的用途
- 11.6.3. 预训练模型的优点

11.7. 用于迁移学习的预训练模型
- 11.7.1. 迁移学习
- 11.7.2. 迁移学习过程
- 11.7.3. 迁移学习的优点

11.8. 深度计算机视觉中的分类和定位
- 11.8.1. 图像分类
- 11.8.2. 定位图像中的对象
- 11.8.3. 物体检测

11.9. 物体检测和物体跟踪
- 11.9.1. 物体检测方法
- 11.9.2. 对象跟踪算法
- 11.9.3. 追踪技术

11.10. 语义分割
- 11.10.1. 语义分割的深度学习
- 11.10.2. 边缘检测
- 11.10.3. 基于规则的分割方法

模块 12. 用自然递归网络(RNN)和注意力进行自然语言处理(NLP)

12.1. 使用RNN生成文本
- 12.1.1. 训练RNN进行文本生成
- 12.1.2. 使用RNN生成自然语言
- 12.1.3. RNN的文本生成应用

12.2. 创建训练数据集
- 12.2.1. 训练RNN的数据准备
- 12.2.2. 存储训练数据集
- 12.2.3. 数据清理和转换
- 12.2.4. 情绪分析

12.3. 使用RNN对意见进行分类
- 12.3.1. 检测评论中的主题
- 12.3.2. 使用Deep Learning算法进行情感分析

12.4. 用于神经机器翻译的编码器-解码器网络
- 12.4.1. 训练用于机器翻译的RNN
- 12.4.2. 使用encoder-decoder器网络进行机器翻译
- 12.4.3. 使用RNN提高机器翻译准确性

12.5. 注意力机制
- 12.5.1. 关怀机制在RNN中的应用
- 12.5.2. 使用注意力机制提高模型准确性
- 12.5.3. 神经网络中注意力机制的优点

12.6. Transformer模型
- 12.6.1. 使用Transformers模型进行自然语言处理
- 12.6.2. Transformers 模型在视觉中的应用
- 12.6.3. Transformers模型的优点

12.7. Transformers视觉
- 12.7.1. 使用Transformers模型实现视觉
- 12.7.2. 图像数据预处理
- 12.7.3. 为视觉训练 变形金刚 模型

12.8. 拥抱脸 变形金刚 书架
- 12.8.1. 使用Hugging FaceTransformer库
- 12.8.2. 抱抱脸的 变形金刚图书馆应用程序
- 12.8.3. 抱抱脸 变形金刚图书馆的优势

12.9. 其他Transformer库比较
- 12.9.1. 不同Transformers库之间的比较
- 12.9.2. 使用其他Transformers库
- 12.9.3. 其他Transformers库的优点

12.10. 使用RNN和Attention开发NLP应用程序。实际应用
- 12.10.1. 使用RNN和注意力机制开发自然语言处理应用程序
- 12.10.2. 在实施过程中使用RNN，护理机制和 Transformers模型
- 12.10.3. 实际应用评价

模块 13. 自动编码器，GAN和扩散模型

13.1. 高效的数据表示
- 13.1.1. 降维
- 13.1.2. 深度学习
- 13.1.3. 紧凑的表示

13.2. 使用不完全线性自动编码器执行PCA
- 13.2.1. 训练过程
- 13.2.2. Python中的实现
- 13.2.3. 测试数据的使用

13.3. 堆叠式自动编码器
- 13.3.1. 深度神经网络
- 13.3.2. 编码架构的构建
- 13.3.3. 使用正则化

13.4. 卷积自动编码器
- 13.4.1. 卷积模型设计
- 13.4.2. 训练卷积模型
- 13.4.3. 评估结果

13.5. 去噪自动编码器
- 13.5.1. 过滤器应用
- 13.5.2. 编码模型设计
- 13.5.3. 使用正则化技术

13.6. 分散自动编码器
- 13.6.1. 提高编码效率
- 13.6.2. 最小化参数数量
- 13.6.3. 使用正则化技术

13.7. 变分自动编码器
- 13.7.1. 使用变分优化
- 13.7.2. 无监督深度学习
- 13.7.3. 深层潜在表征

13.8. 时尚MNIST图像的生成
- 13.8.1. 模式识别
- 13.8.2. 影像学
- 13.8.3. 深度神经网络训练

13.9. 生成对抗网络和扩散模型
- 13.9.1. 从图像生成内容
- 13.9.2. 数据分布建模
- 13.9.3. 使用对抗性网络

13.10. 模型的实施
- 13.10.1. 实际应用
- 13.10.2. 模型的实施
- 13.10.3. 使用真实数据
- 13.10.4. 评估结果

模块 14. 生物启发式计算

14.1. 仿生计算简介
- 14.1.1. 仿生计算简介

14.2. 社会适应算法
- 14.2.1. 基于蚁群的仿生计算
- 14.2.2. 蚁群算法的变体
- 14.2.3. 粒子云计算

14.3. 遗传算法
- 14.3.1. 总体结构
- 14.3.2. 主要算子的实现

14.4. 遗传算法的空间探索-开发策略
- 14.4.1. CHC算法
- 14.4.2. 多模式问题

14.5. 进化计算模型(一)
- 14.5.1. 进化策略
- 14.5.2. 进化编程
- 14.5.3. 基于差分进化的算法

14.6. 进化计算模型(二)
- 14.6.1. 基于分布估计(EDA)的演化模型
- 14.6.2. 遗传编程

14.7. 进化规划应用于学习问题
- 14.7.1. 基于规则的学习
- 14.7.2. 实例选择问题中的进化方法

14.8. 多目标问题
- 14.8.1. 支配的概念
- 14.8.2. 进化算法在多目标问题中的应用

14.9. 神经网络 (一)
- 14.9.1. 神经网络简介
- 14.9.2. 神经网络的实际例子

14.10. 神经网络(二)
- 14.10.1. 神经网络在医学研究中的用例
- 14.10.2. 神经网络在经济学中的使用案例
- 14.10.3. 神经网络在计算机视觉中的使用案例

模块 15. 人工智能:策略和应用

15.1. 金融服务
- 15.1.1. 人工智能 (IA) 对金融服务的影响。机遇与挑战
- 15.1.2. 使用案例
- 15.1.3. 使用人工智的相关潜在风险
- 15.1.4. 人工智能未来的潜在发展/用途

15.2. 人工智能对医疗保健服务的影响
- 15.2.1. 人工智能对医疗保健领域的影响机遇与挑战
- 15.2.2. 使用案例

15.3. 与在医疗服务中使用人工智能相关的风险
- 15.3.1. 使用人工智的相关潜在风险
- 15.3.2. 人工智能未来的潜在发展/用途

15.4. 零售
- 15.4.1. 人工智能对RetIAl业的影响机遇与挑战
- 15.4.2. 使用案例
- 15.4.3. 使用人工智的相关潜在风险
- 15.4.4. 人工智能未来的潜在发展/用途

15.5. 行业
- 15.5.1. 人工智能对工业的影响。机遇与挑战
- 15.5.2. 使用案例

15.6. 在工业中使用人工智能的潜在风险
- 15.6.1. 使用案例
- 15.6.2. 使用人工智的相关潜在风险
- 15.6.3. 人工智能未来的潜在发展/用途

15.7. 公共行政
- 15.7.1. 人工智能对公共行政的影响。机遇与挑战
- 15.7.2. 使用案例
- 15.7.3. 使用人工智的相关潜在风险
- 15.7.4. 人工智能未来的潜在发展/用途

15.8. 教育
- 15.8.1. 人工智能对教育的影响。机遇与挑战
- 15.8.2. 使用案例
- 15.8.3. 使用人工智的相关潜在风险
- 15.8.4. 人工智能未来的潜在发展/用途

15.9. 林业和农业
- 15.9.1. 人工智能对林业和农业的影响机遇与挑战
- 15.9.2. 使用案例
- 15.9.3. 使用人工智的相关潜在风险
- 15.9.4. 人工智能未来的潜在发展/用途

15.10人力资源
- 15.10.1. 人工智能人力资源的影响。机遇与挑战
- 15.10.2. 使用案例
- 15.10.3. 使用人工智的相关潜在风险
- 15.10.4. 人工智能未来的潜在发展/用途

模块 16. 利用人工智能提高软件开发效率

16.1. 准备合适的开发环境
- 16.1.1. 选择开发人工智能的基本工具
- 16.1.2. 配置所选工具
- 16.1.3. 实施适应人工智能项目的CI/CD流程
- 16.1.4. 开发环境中高效的依赖关系和版本管理

16.2. Visual Studio Code必备的IA扩展工具
- 16.2.1. 探索和选择Visual Studio Code的人工智能扩展
- 16.2.2. 在IDE中集成静态和动态分析工具
- 16.2.3. 利用特定扩展功能自动执行重复性任务
- 16.2.4. 定制开发环境提高效率

16.3. 使用Flutterflow进行无代码用户界面设计
- 16.3.1. 无代码设计原则及其在用户界面中的应用
- 16.3.2. 在视觉界面设计中融入人工智能元素
- 16.3.3. 无代码 创建智能界面的工具和平台
- 16.3.4. 利用人工智能评估和持续改进 无代码界面

16.4. 使用ChatGPT优化代码
- 16.4.1. 识别重复代码
- 16.4.2. 重构
- 16.4.3. 建构可读代码
- 16.4.4. 了解代码的作用
- 16.4.5. 改进变量和函数名称
- 16.4.6. 自动创建文档

16.5. 使用ChagGPT进行人工智能资源库管理
- 16.5.1. 利用人工智能技术实现版本控制流程自动化
- 16.5.2. 协作环境中的冲突检测和自动解决
- 16.5.3. 对代码库中的变化和趋势进行预测分析
- 16.5.4. 利用人工智能改进资料库的组织和分类工作

16.6. 将人工智能与AskYourDatabase集成到数据库管理中
- 16.6.1. 利用人工智能技术优化查询和性能
- 16.6.2. 数据库访问模式的预测分析
- 16.6.3. 实施推荐系统优化数据库结构
- 16.6.4. 主动监控和检测潜在的数据库问题

16.7. 使用ChatGPT进行基于人工智能的故障查找和单元测试创建
- 16.7.1. 利用人工智能技术自动生成测试用例
- 16.7.2. 利用人工智能静态分析及早发现漏洞和错误
- 16.7.3. 通过人工智能识别关键领域提高测试覆盖率

16.8. 使用GitHub Copilot进行结对编程
- 16.8.1. 在配对编程中有效地集成和使用GitHub Copilot
- 16.8.2. 集成GitHub Copilot改进开发人员的沟通与协作
- 16.8.3. 充分利用GitHub Copilot生成的代码建议的集成策略
- 16.8.4. 人工智能辅助 结对编程的集成案例研究和最佳实践

16.9. 使用ChatGPT实现编程语言之间的自动翻译
- 16.9.1. 针对编程语言的特定语言机器翻译工具和服务
- 16.9.2. 使机器翻译算法适应发展环境
- 16.9.3. 通过机器翻译提高不同语言之间的互操作性
- 16.9.4. 评估和缓解机器翻译的潜在挑战和限制

16.10. 提高生产力的人工智能工具推荐
- 16.10.1. 用于软件开发的人工智能工具比较分析
- 16.10.2. 在工作流程中集成人工智能工具
- 16.10.3. 利用人工智能工具实现日常工作自动化
- 16.10.4. 根据项目背景和要求评估和选择工具

模块 17. 基于人工智能软件架构

17.1. 借助ChatGPT进行AI工具的优化和性能管理
- 17.1.1. 人工智能工具的性能剖析和分析
- 17.1.2. 人工智能算法和模型的优化策略
- 17.1.3. 采用缓存并行化技术提高性能
- 17.1.4. 持续实时性能监测的工具和方法

17.2. 使用ChatGPT的AI应用程序的可扩展性
- 17.2.1. 为人工智能应用设计可扩展架构
- 17.2.2. 实施分区和负载分担技术
- 17.2.3. 可扩展系统中的工作流程和工作量管理
- 17.2.4. 多变需求环境下的横向和纵向扩展策略

17.3. 使用ChatGPT维护人工智能应用程序
- 17.3.1. 促进人工智能项目可维护性的设计原则
- 17.3.2. 人工智能模型和算法的具体记录策略
- 17.3.3. 实施单元和集成测试以方便维护
- 17.3.4. 人工智能组件系统的重构和持续改进方法

17.4. 大型系统设计
- 17.4.1. 设计大型系统的架构原则
- 17.4.2. 将复杂系统分解为微服务
- 17.4.3. 实施分布式系统的特定设计模式
- 17.4.4. 采用人工智能组件的大规模架构的复杂性管理策略

17.5. 用于人工智能工具的大规模数据仓库
- 17.5.1. 选择可扩展的数据存储技术
- 17.5.2. 有效处理海量数据的数据库模式设计
- 17.5.3. 海量存储环境中的分区和复制策略
- 17.5.4. 实施数据管理系统确保人工智能项目的完整性和可用性

17.6. 使用ChatGPT的AI数据结构
- 17.6.1. 将经典数据结构应用于人工智能算法
- 17.6.2. 使用ChatGPT设计和优化特定数据结构
- 17.6.3. 在数据密集型系统中整合高效数据结构
- 17.6.4. 利用人工智能在数据结构中进行实时数据操作和存储的策略

17.7. 人工智能产品的编程算法
- 17.7.1. 开发和实施针对人工智能应用的特定算法
- 17.7.2. 根据问题类型和产品要求选择算法策略
- 17.7.3. 改造经典算法将其融入人工智能系统
- 17.7.4. 人工智能开发环境中不同算法性能的评估和比较

17.8. 人工智能开发的设计模式
- 17.8.1. 包含人工智能组件的项目中识别和应用常见的设计模式
- 17.8.2. 开发将模型和算法集成到现有系统中的具体模式
- 17.8.3. 提高人工智能项目可重用性和可维护性的模式实施策略
- 17.8.4. 在人工智能架构中应用设计模式的案例研究和最佳实践

17.9. 使用ChatGPT实现清洁的架构
- 17.9.1. 清洁架构的基本原则和概念
- 17.9.2. 使简洁架构适用于包含人工智能组件的项目
- 17.9.3. 以简洁的架构在系统中实施层级和依赖关系
- 17.9.4. 在人工智能软件开发中实施清洁架构的优势和挑战

17.10. 使用DeepCode在网络应用程序中开发安全软件
- 17.10.1. 使用人工智能组件进行软件开发的安全原则
- 17.10.2. 识别并减少人工智能模型和算法中的潜在漏洞
- 17.10.3. 在具有人工智能功能的网络应用程序中实施安全开发实践
- 17.10.4. 在人工智能项目中保护敏感数据和防止攻击的策略

模块 18. 人工智能网络项目

18.1. 为人工智能网络开发准备工作环境
- 18.1.1. 为人工智能项目配置网络开发环境
- 18.1.2. 选择和准备人工智能网络开发的基本工具
- 18.1.3. 为人工智能网络项目整合特定的库和 框架
- 18.1.4. 在配置协作开发环境方面实施最佳做法

18.2. 使用GitHub Copilot为人工智能项目创建工作区
- 18.2.1. 有效设计和组织包含人工智能组件的网络项目工作区
- 18.2.2. 在 工作区使用项目管理和版本控制工具
- 18.2.3. 开发团队高效协作和沟通的策略
- 18.2.4. 用人工智能调整 工作区以 适应网络项目的特殊需求

18.3. 使用GitHub Copilot的产品设计模式
- 18.3.1. 具有人工智能元素的用户界面中常见设计模式的识别和应用
- 18.3.2. 开发特定模式利用人工智能改善网络项目的用户体验
- 18.3.3. 将设计模式与人工智能整合到网络项目的整体架构中
- 18.3.4. 根据项目背景评估和选择合适的设计模式

18.4. 使用GitHub Copilot进行前端开发
- 18.4.1. 将人工智能模型集成到网络项目的表现层中
- 18.4.2. 开发具有人工智能元素的自适应用户界面
- 18.4.3. 在前端实现自然语言处理(PLN)功能
- 18.4.4. 利用人工智能优化前端开发性能的策略

18.5. 使用GitHub Copilot创建数据库
- 18.5.1. 为人工智能网络项目选择数据库技术
- 18.5.2. 用于存储和管理人工智能相关数据的数据库模式设计
- 18.5.3. 为人工智能模型生成的大量数据实施高效的存储系统
- 18.5.4. 人工智能网络项目中数据库敏感数据的安全和保护策略

18.6. 使用GitHub Copilot进行后端开发
- 18.6.1. 将人工智能服务和模型集成到后台业务逻辑中
- 18.6.2. 为前端和人工智能组件之间的通信开发特定的应用程序接口和端点
- 18.6.3. 在后台利用人工智能实现数据处理逻辑和决策制定
- 18.6.4. 人工智能网络项目后台开发的可扩展性和性能策略

18.7. 优化网络部署流程
- 18.7.1. 使用ChatGPT自动完成网络项目的构建和部署过程
- 18.7.2. 利用GitHub Copilot为网络应用程序量身定制CI/CD管道
- 18.7.3. 持续部署中的高效发布和升级管理策略
- 18.7.4. 部署后监测和分析以持续改进流程

18.8. 云计算中的人工智能
- 18.8.1. 将人工智能服务整合到云计算平台中
- 18.8.2. 利用具有人工智能功能的云服务开发可扩展的分布式解决方案
- 18.8.3. Cloud环境中人工智能网络应用程序的高效资源和成本管理策略
- 18.8.4. 评估和比较人工智能网络项目的云服务提供商

18.9. 借助ChatGPT为LAMP环境创建带人工智能的项目
- 18.9.1. 调整基于LAMP堆栈的网络项目使其包含人工智能组件
- 18.9.2 在LAMP环境中集成人工智能专用库和框架
- 18.9.3 开发人工智能功能为了补充传统的 LAMP架构
- 18.9.4 LAMP环境中人工智能网络项目的优化和维护策略

18.10. 使用ChatGPT为MEVN环境创建人工智能项目
- 18.10.1. 将MEVN堆栈中的技术和工具与人工智能组件整合在一起
- 18.10.2. 在具有人工智能功能的MEVN环境中开发可扩展的现代网络应用程序
- 18.10.3. 在MEVN项目中实施数据处理和机器学习功能
- 18.10.4. 在MEVN环境中提高人工智能网络应用程序性能和安全性的策略

模块 19. 人工智能移动应用

19.1. 为人工智能移动开发准备工作环境
- 19.1.1. 为人工智能项目建立移动开发环境
- 19.1.2. 选择并准备开发人工智能移动应用程序的特定工具
- 19.1.3. 在移动开发环境中集成人工智能库和 **框架**
- 19.1.4. 设置模拟器和真实设备以测试带有人工智能组件的移动应用程序

19.2. 使用GitHub Copilot创建**工作区**
- 19.2.1. 将GitHub Copilot集成到移动开发环境中
- 19.2.2. 在人工智能项目中有效使用GitHub Copilot生成代码
- 19.2.3. 在**工作区**使用GitHub Copilot时的开发人员协作策略
- 19.2.4. 在利用人工智能开发移动应用时使用 GitHub Copilot的最佳实践和局限性

19.3. Firebase配置
- 19.3.1. 为移动开发初步设置Firebase项目
- 19.3.2. 将Firebase集成到具有人工智能功能的移动应用程序中
- 19.3.3. 在人工智能项目中使用Firebase服务作为数据库，身份验证和通知功能
- 19.3.4. 使用Firebase在移动应用程序中进行实时数据和事件管理的策略

19.4. 简洁架构概念，数据源，存储库
- 19.4.1. 利用人工智能进行移动开发的简洁架构基本原则
- 19.4.2. 使用GitHub Copilot实施数据源和资源库层
- 19.4.3. 使用GitHub Copilot设计和构建移动项目中的组件
- 19.4.4. 在人工智能移动应用中实施**简洁架构**的优势和挑战

19.5. 使用GitHub Copilot创建身份验证屏幕
- 19.5.1. 设计和开发人工智能移动应用中验证屏幕的用户界面
- 19.5.2. 在登录屏幕中整合Firebase认证服务
- 19.5.3. 在身份验证屏幕中使用安全和数据保护技术
- 19.5.4. 个性化和定制化用户在身份验证屏幕上的体验

19.6. 使用GitHub Copilot创建仪表板并进行导航
- 19.6.1. 设计和开发包含人工智能元素的 **仪表盘**
- 19.6.2. 在人工智能移动应用中实施高效导航系统
- 19.6.3. 在**仪表板**中集成人工智能功能提升用户体验

19.7. 使用GitHub Copilot创建列表界面
- 19.7.1. 为人工智能移动应用程序中的列表屏幕开发用户界面
- 19.7.2. 在列表屏幕中集成推荐和过滤算法
- 19.7.3. 使用设计模式在列表中有效展示数据
- 19.7.4. 在显示屏上高效加载实时数据和列表的策略

19.8. 使用GitHub Copilot创建详细界面
- 19.8.1. 设计和开发用于展示特定信息的详细用户界面
- 19.8.2. 集成人工智能功能为了丰富细节界面
- 19.8.3. 在详细屏幕上实现交互和动画
- 19.8.4. 优化人工智能移动应用程序加载和显示性能的策略

19.9. 使用GitHub Copilot创建设置界面
- 19.9.1. 基于人工智能开发移动应用程序配置和设置的用户界面
- 19.9.2. 集成与人工智能组件相关的自定义设置
- 19.9.3. 在配置屏幕中执行自定义选项和首选项
- 19.9.4. 在设置屏幕上显示选项时的可用性和清晰度策略

19.10. 使用人工智能为你的应用程序创建图标，喷溅和图形资源
- 19.10.1. 设计和创建有吸引力的图标用人工智能表现移动应用程序
- 19.10.2. 开发具有冲击力视觉效果的(闪屏)
- 19.10.3. 选择和调整图形资源提高移动应用程序的美感
- 19.10.4. 人工智能应用图形元素的一致性和视觉品牌策略

模块 20. 用于QA测试的人工智能

20.1. 测**试**生命周期
- 20.1.1. 描述并了解软件开发中的 测**试** 生命周期
- 20.1.2. 测**试** 生命周期的各个阶段及其在质量保证中的重要性
- 20.1.3. 在 测**试**生命周期的不同阶段整合人工智能
- 20.1.4. 通过使用人工智能持续改进 测**试** 生命周期的策略

20.2. 在ChatGPT的帮助下进行测试用例和错误检测
- 20.2.1. 在QA测**试**的背景下设计和编写有效的测试用例
- 20.2.2. 在测试用例执行过程中识别错误和误差
- 20.2.3. 通过静态分析应用早期错误检测技术
- 20.2.4. 使用人工智能工具自动识别测试用例中的错误

20.3. 测试类型
- 20.3.1. QA领域不同类型测试的探索
- 20.3.2. 单元测试，集成测试，功能测试和验收测试：功能和应用程序
- 20.3.3. ChatGPT项目中测试类型的选择和适当组合的策略
- 20.3.4. 使用ChatGPT将传统类型的测试调整到项目中

20.4. 使用ChatGPT创建测试计划
- 20.4.1. 设计和构建全面的测试计划
- 20.4.2. 识别人工智能项目中的需求和测试场景
- 20.4.3. 手动和自动测试规划策略
- 20.4.4. 根据项目的发展不断评估和调整测试计划

20.5. 人工智能 错误 检测和报告
- 20.5.1. 利用机器学习算法实现自动错误检测技术
- 20.5.2. 使用ChatGPT进行动态代码分析以发现潜在错误
- 20.5.3. 自动生成人工智能检测到的错误的详细报告的策略
- 20.5.4. 开发和QA团队在管理人工智能识别的错误方面进行有效协作

20.6. 利用人工智能创建自动测试
- 20.6.1. 使用ChatGPT为项目开发自动化测试脚本
- 20.6.2. 集成基于人工智能的测试自动化工具
- 20.6.3. 使用ChatGPT动态生成自动化测试用例
- 20.6.4. 在人工智能项目中高效执行和维护自动化测试的策略

20.7. API测试
- 20.7.1. API 测试 的基本概念及其在质量保证中的重要性
- 20.7.2. 使用ChatGPT开发用于验证环境中 API的测试
- 20.7.3. 使用ChatGPT验证API测试中的数据和结果的策略
- 20.7.4. 使用特定工具在人工智能项目中 测试 API

20.8. 用于网络测试的人工智能工具
- 20.8.1. 探索网络环境中测试自动化的人工智能工具
- 20.8.2. 在网络 测试中整合元素识别和视觉分析技术
- 20.8.3. 利用ChatGPT自动检测网络应用程序变化和性能问题的策略
- 20.8.4. 评估利用人工智能提高网络测试效率的特定工具

20.9. 通过人工智能进行移动测试
- 20.9.1. 为带有人工智能组件的移动应用程序制定测试策略
- 20.9.2. 为移动平台整合特定的人工智能 测试工具
- 20.9.3. 使用ChatGPT检测性能问题移动应用
- 20.9.4. 利用人工智能验证特定移动应用程序界面和功能的策略

20.10. 人工智能的QA工具
- 20.10.1. 探索包含人工智能功能的QA工具和平台
- 20.10.2. 评估人工智能项目中高效测试管理和执行工具
- 20.10.3. 使用ChatGPT进行测试用例生成和优化
- 20.10.4. 选择和有效采用人工智能质量保证工具的策略

> 100%在线课程可满足你的需求，让你能够身临其境地进行扎实学习，从而在就业市场上为自己定位"

06

方法

这个培训计划提供了一种不同的学习方式。我们的方法是通过循环的学习模式发展起来的: **Re-learning**。

这个教学系统被世界上一些最著名的医学院所采用，并被**新英格兰医学杂志**等权威出版物认为是最有效的教学系统之一。

发现 Re-learning，这个系统放弃了传统的线性学习，带你体验循环教学系统：这种学习方式已经证明了其巨大的有效性，尤其是在需要记忆的科目中”

案例研究,了解所有内容的背景

我们的方案提供了一种革命性的技能和知识发展方法。我们的目标是在一个不断变化,竞争激烈和高要求的环境中加强能力建设。

> 和TECH,你可以体验到一种正在动摇世界各地传统大学基础的学习方式”

你将进入一个以重复为基础的学习系统,在整个教学大纲中采用自然和渐进式教学。

学生将通过合作活动和真实案例，学习如何解决真实商业环境中的复杂情况。

一种创新并不同的学习方法

该技术课程是一个密集的教学计划，从零开始，提出了该领域在国内和国际上最苛刻的挑战和决定。由于这种方法，个人和职业成长得到了促进，向成功迈出了决定性的一步。案例法是构成这一内容的技术基础，确保遵循当前经济，社会和职业现实。

> 我们的课程使你准备好在不确定的环境中面对新的挑战，并取得事业上的成功”

在世界顶级计算机科学学校存在的时间里，案例法一直是最广泛使用的学习系统。1912年开发的案例法是为了让法律学生不仅在理论内容的基础上学习法律，案例法向他们展示真实的复杂情况，让他们就如何解决这些问题作出明智的决定和价值判断。1924年，它被确立为哈佛大学的一种标准教学方法。

在特定情况下，专业人士应该怎么做？这就是我们在案例法中面对的问题，这是一种以行动为导向的学习方法。在整个课程中，学生将面对多个真实的案例。他们必须整合所有的知识，研究，论证和捍卫他们的想法和决定。

Re-learning 方法

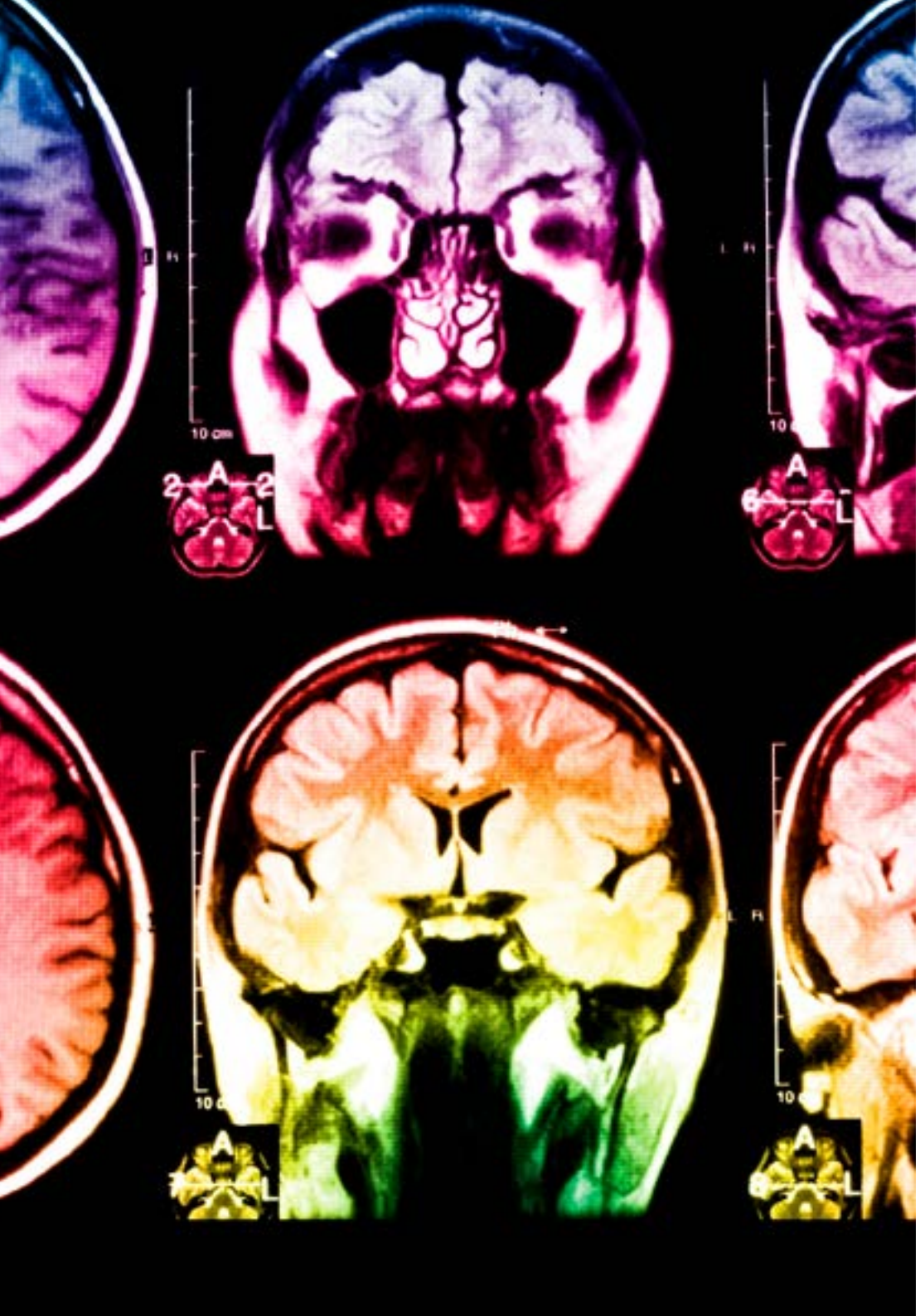
TECH有效地将案例研究方法与基于循环的100%在线学习系统相结合，在每节课中结合了个不同的教学元素。

我们用最好的100%在线教学方法加强案例研究：Re-learning。

> 在2019年，我们取得了世界上所有西班牙语在线大学中最好的学习成绩。

在TECH，你将用一种旨在培训未来管理人员的尖端方法进行学习。这种处于世界教育学前沿的方法被称为 Re-learning。

我校是唯一获准使用这一成功方法的西班牙语大学。2019年，我们成功地提高了学生的整体满意度（教学质量,材料质量,课程结构,目标……），与西班牙语最佳在线大学的指标相匹配。



在我们的方案中，学习不是一个线性的过程，而是以螺旋式的方式发生（学习，解除学习，忘记和重新学习）。因此，我们将这些元素中的每一个都结合起来。这种方法已经培养了超过65万名大学毕业生，在生物化学，遗传学，外科，国际法，管理技能，体育科学，哲学，法律，工程，新闻，历史，金融市场和工具等不同领域取得了前所未有的成功。所有这些都是在一个高要求的环境中进行的，大学学生的社会经济状况很好，平均年龄为43.5岁。

Re-learning 将使你的学习事半功倍，表现更出色，使你更多地参与到训练中，培养批判精神，捍卫论点和对比意见：直接等同于成功。

从神经科学领域的最新科学证据来看，我们不仅知道如何组织信息，想法，图像y记忆，而且知道我们学到东西的地方和背景，这是我们记住它并将其储存在海马体的根本原因，并能将其保留在长期记忆中。

通过这种方式，在所谓的神经认知背景依赖的电子学习中，我们课程的不同元素与学员发展其专业实践的背景相联系。

该方案提供了最好的教育材料，为专业人士做了充分准备：

学习材料

所有的教学内容都是由教授该课程的专家专门为该课程创作的，因此，教学的发展是具体的。

然后，这些内容被应用于视听格式，创造了TECH在线工作方法。所有这些，都是用最新的技术，提供最高质量的材料，供学生使用。

大师课程

有科学证据表明第三方专家观察的有用性。

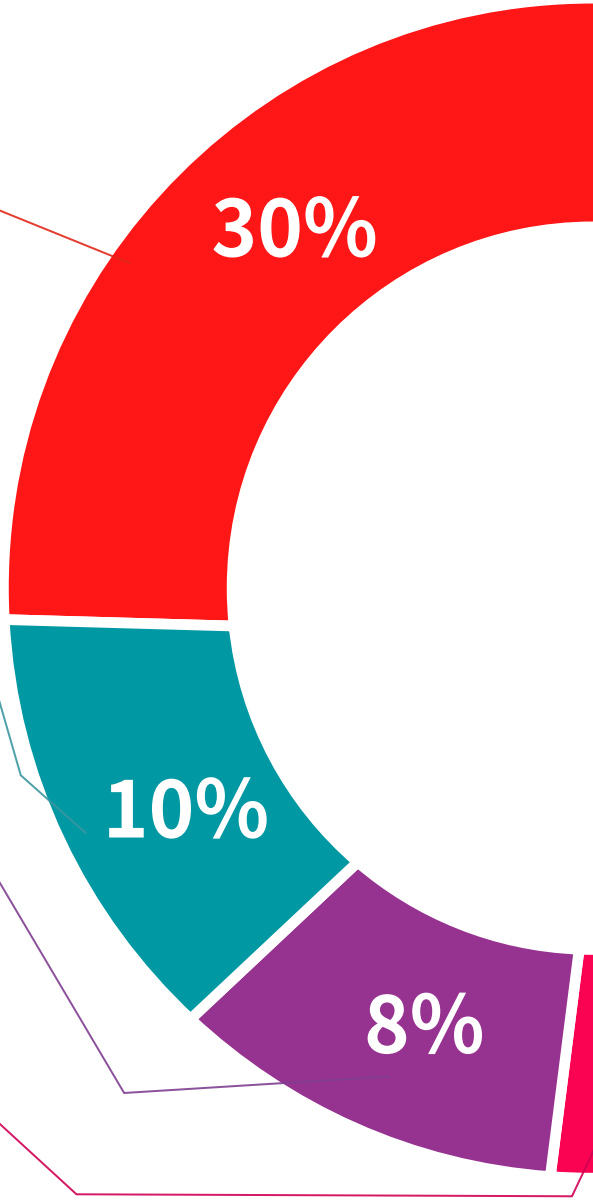
向专家学习可以加强知识和记忆，并为未来的困难决策建立信心。

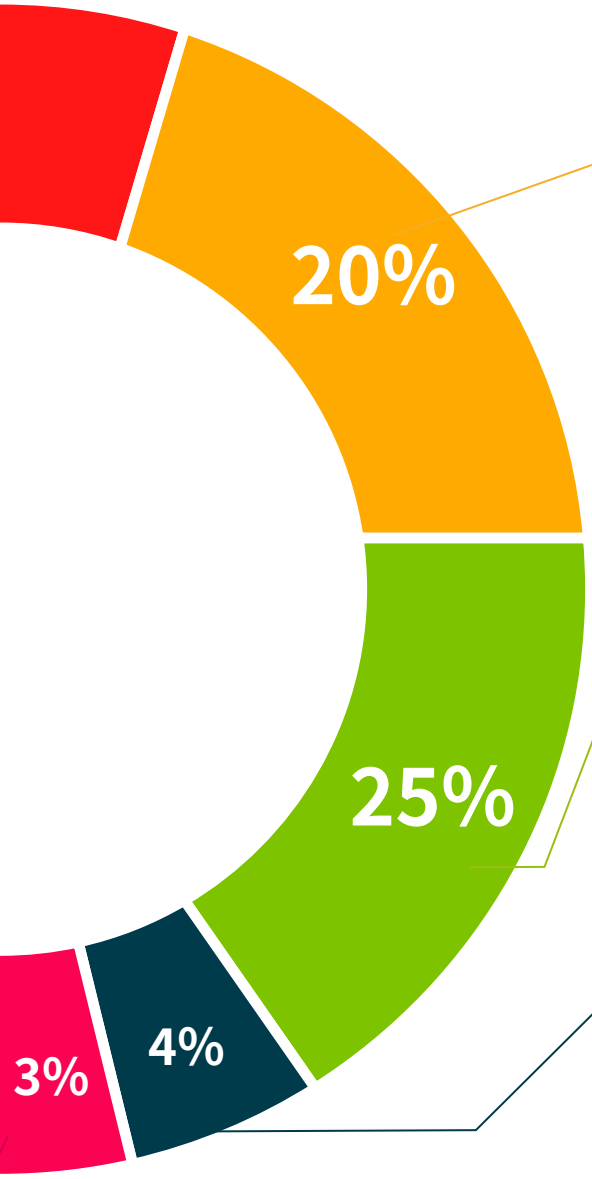
技能和能力的实践

你将开展活动以发展每个学科领域的具体能力和技能。在我们所处的全球化框架内，我们提供实践和氛围帮你取得成为专家所需的技能和能力。

延伸阅读

最近的文章，共识文件和国际准则等。在TECH的虚拟图书馆里，学生可以获得他们完成培训所需的一切。





案例研究

他们将完成专门为这个学位选择的最佳案例研究。由国际上最好的专家介绍,分析和辅导案例。

互动式总结

TECH团队以有吸引力和动态的方式将内容呈现在多媒体丸中,其中包括音频,视频,图像,图表和概念图,以强化知识。

这个用于展示多媒体内容的独特教育系统被微软授予“欧洲成功案例”称号。

测试和循环测试

在整个课程中,通过评估和自我评估活动和练习,定期评估和重新评估学习者的知识:通过这种方式,学习者可以看到他/她是如何实现其目标的。

07

学位

编程中的人工智能校级硕士除了保证最严格和最新的培训外，还可以获得由 TECH 科技大学 颁发的校级硕士学位证书。

顺利完成该课程后你将获得大学学位证书无需出门或办理其他手续"

这个**编程中的人工智能校级硕士**包含了市场上最完整和最新的课程。

评估通过后，学生将通过邮寄收到**TECH科技大学**颁发的相应的**校级硕士**学位。

学位由**TECH科技大学**颁发，证明在校级硕士学位中所获得的资质，并满足工作交流，竞争性考试和职业评估委员会的要求。

学位：**编程中的人工智能校级硕士**

模式:**在线**

时长: **7个月**


tech 科学技术大学

授予

文凭

_____先生/女士, 身份证号码_____

顺利通过并获得认证的课程是

校级硕士

编程中的人工智能

这是一个由本大学授予的学位，相当于2,250个小时，
开始日期是 dd/mm/aaaa,结束日期是dd/mm/aaaa。

截至2018年6月28日，TECH是一所被公共教育部认可的私立高等教育机构。

2020年六月17日z

Tere Guevara Navarro女士
校长

Unique TECH Code: AFWORD23S techtitute.com/certificates


设计中的人工智能校级硕士

科目类型	小时
必修(OB)	2,250
选修 (OP)	0
外部实习 (PR)	0
硕士论文 (TFM)	0
	总计2,250

教学大纲概况

学期	学科	小时	性质
1°	人工智能基础	112	必修
1°	数据类型和周期	112	必修
1°	人工智能中的数据	112	必修
1°	数据挖掘选择, 预处理和转换	112	必修
1°	人工智能中的算法与复杂性	112	必修
1°	智能系统	112	必修
1°	机器学习和数据挖掘	112	必修
1°	神经网络, Deep Learning的基础	112	必修
1°	深度神经网络训练	112	必修
1°	使用TensorFlow进行模型定制和训练	112	必修
1°	利用卷积神经网络实现深度计算机视觉	113	必修
1°	用自然递归网络(RNN)和注意力进行自然语言处理(NLP)	113	必修
1°	自动编码器, GAN和扩散模型	113	必修
1°	生物启发式计算	113	必修
1°	人工智能：策略和应用	113	必修
1°	利用人工智能提高软件开发效率	113	必修
1°	基于人工智能软件架构	113	必修
1°	人工智能网络项目	113	必修
1°	人工智能移动应用	113	必修
1°	用于QA测试的人工智能	113	必修

Tere Guevara Navarro女士
校长

tech 科学技术大学

*海牙加注。如果学生要求为他们的纸质资格证书提供海牙加注，TECH EDUCATION将采取必要的措施来获得，但需要额外的费用。

tech 科学技术大学

校级硕士
编程中的人工智能

- 模式:**在线**
- 时长: **7个月**
- 学位: **TECH 科技大学**
- 课程表:**自由安排时间**
- 考试模式:**在线**

校级 硕士
编程中的人工智能

tech 科学技术大学