

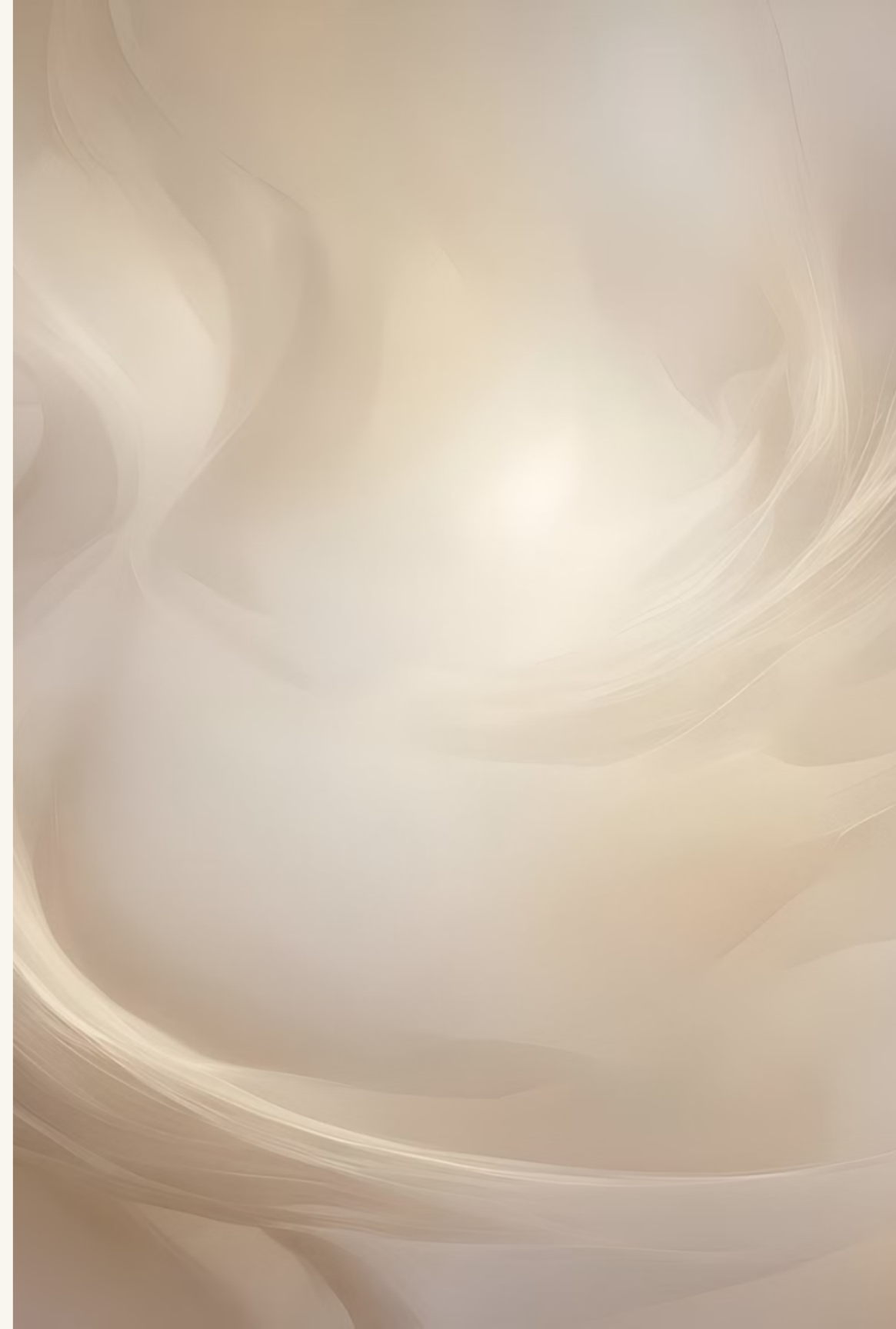
课程介绍

在这个课程中，我们将全面介绍机器学习的预备知识。从基础的数学、概率论和编程语言开始，逐步深入到数据预处理、特征工程和常见机器学习算法的应用。通过实战案例演示，让学员对机器学习有更深入的理解和实

践。

sa

by



课程目标

掌握基础知识

通过学习数学、概率论、统计学以及编程语言的基础知识,为后续的机器学习打下坚实的基础。

熟悉工具使用

学习使用Python以及相关的数据分析库,如NumPy、Pandas和Matplotlib,为实践和项目开发做好准备。

理解数据处理

学习数据清洗、特征工程等数据预处理技能,为后续的模型训练和优化做好铺垫。

掌握机器学习

了解机器学习的基本算法及其应用,为独立完成机器学习项目奠定基础。

学习内容概览



基础知识

包括数学、概率论、统计学和编程语言基础,为后续学习打下坚实基础。



工具使用

学习Python及其常用数据分析库,如NumPy、Pandas和Matplotlib,为实践项目做好准备。



数据处理

掌握数据清洗、特征工程等预处理技能,为模型训练和优化做好铺垫。



机器学习

学习机器学习的基本算法及其应用,为独立完成项目奠定基础。

预备知识重要性



奠定基础

掌握数学、概率统计和编程语言等基础知识,为后续机器学习学习奠定坚实的基础。



数据处理能力

学习数据预处理技能,包括数据清洗和特征工程,为模型训练优化做好准备。



实践能力培养

通过学习工具使用和机器学习算法应用,培养独立完成项目的实践能力。

数学基础知识

微积分

了解微分和积分的基本概念,掌握常见函数的求导和积分方法。这是机器学习中许多优化算法的基础。

线性代数

学习矩阵、向量、特征值等概念。这些在机器学习中用于建模和优化问题的表示。

概率统计

掌握随机变量、概率分布、统计推断等知识。这对于理解机器学习模型的工作原理很重要。

概率论基础

1 概率概念

理解随机事件的概率定义及其计算方法, 为学习概率分布和统计推断奠定基础。

3 条件概率

理解条件概率的概念及其在机器学习中的应用, 如贝叶斯理论等。

2 概率分布

学习常见的概率分布模型, 如正态分布、二项分布等, 并掌握它们的性质和应用场景。

4 随机变量

掌握随机变量的定义和性质, 为后续的统计推断奠定基础。

统计学基础

1 描述性统计

学习收集、整理和分析数据的方法,包括均值、中位数、方差等常用统计量的计算。为进一步的数据分析奠定基础。

3 统计推断

学习参数估计、假设检验等统计推断的基本方法,为后续分析数据、评估模型提供统计学支撑。

2 概率分布

掌握常见的离散概率分布如二项分布和泊松分布,以及连续概率分布如正态分布的性质和应用。这对于理解机器学习中的概率模型很重要。

4 回归分析

理解线性回归、逻辑回归等常用回归模型,能够解释变量之间的关系并进行预测。这是机器学习中广泛使用的基础技术。

编程语言基础

语法结构

掌握常见编程语言如 Python、Java、C++ 的基本语法结构, 包括变量、数据类型、运算符、控制流等。这些是编程的基础。

数据结构

学习列表、字典、集合等常见数据结构的定义和使用方法。这些为数据处理和算法实现提供支撑。

函数编程

理解函数的定义、参数传递和返回值等概念。函数是实现代码复用和抽象的重要手段。

面向对象

学习类、对象、继承等面向对象编程的基本原理。这是构建复杂程序的重要思想。

Python简介

Python是一种通用的、高级的编程语言,以其简单、优雅和可读性著称。它被广泛应用于数据分析、机器学习、自动化等领域,是数据科学家和工程师的首选语言之一。

Python拥有丰富的标准库和第三方库,为开发人员提供了强大的工具和功能,大大提高了开发效率。它能够与其他编程语言无缝集成,是一种灵活且可扩展的编程语言。



Python环境搭建



高效的Python开发环境搭建是掌握Python编程的关键一步。从安装Python、设置环境变量到选择合适的IDE,一步一步打造您的Python编程基础。

Python基本语法



基本语法结构

Python采用简洁明了的语法结构,如缩进表示代码块、变量命名规范、基本数据类型定义等,易于初学者掌握。



控制流程

Python提供if-else、for、while等常见的控制流结构,帮助开发者编写复杂的条件判断和循环逻辑。



函数定义

Python支持函数定义,可通过参数传递和返回值实现代码复用和模块化设计,提高开发效率。

Python数据类型

12

数字类型

Python支持整数、浮点数和复数等常见的数字类型。可进行算术运算并支持数学函数。



文本类型

Python的字符串类型可表示文本数据,支持索引、切片和格式化等操作,广泛应用于数据处理。



布尔类型

Python的布尔类型只有True和False两个值,用于条件判断和逻辑运算,是控制流的基础。

Python控制流

If-Else 语句

Python的条件判断语句if-else可用于实现基本的逻辑分支,灵活控制程序的执行流程。

For 循环

Python的for循环可轻松遍历各种可迭代对象,如列表、元组和字符串,实现重复性操作。

While 循环

While循环可根据特定条件持续执行代码块,适用于不确定循环次数的场景。

Try-Except 异常处理

Python的异常处理机制可捕获和处理运行时错误,提高程序的健壮性和可靠性。

Python函数

$f(x)$

函数定义

Python允许用户自定义函数,通过def关键字声明函数名称、参数列表和函数体,实现代码的模块化和复用。



参数传递

函数可接受各种数据类型的参数,包括必选参数、默认参数和可变参数等,灵活满足不同的调用需求。



返回值

使用return语句可以让函数返回计算结果,支持单返回值和多返回值,增强函数的表达能力。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/856054020044010143>