项目1

走进机器学习的世界

项目导读

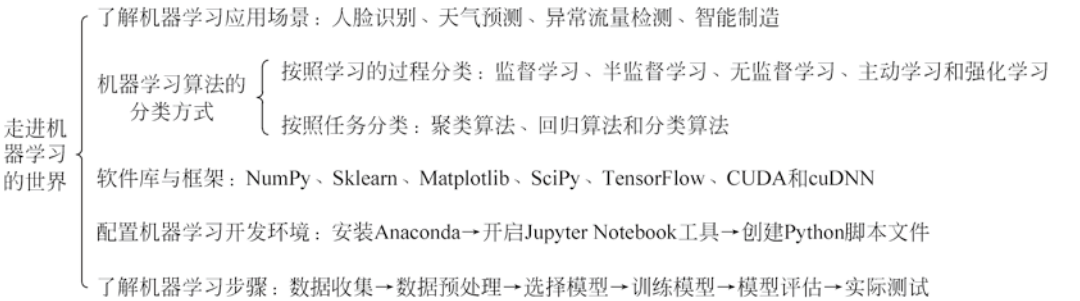
在过去的10年里，机器学习技术实现了自动驾驶、实用的语音识别、有效的网络搜索等。机器学习在今天是如此普及，以至于人们可能一天要使用几十次而不自知。

本项目将从机器学习的应用场景开始，依次给读者介绍生活工作中哪些地方应用了机器学习，这些机器学习算法有哪些分类方式，应用开发涉及的框架和软件库，以及机器学习开发环境如何配置，等等。带读者走进机器学习的大门，为后续项目的学习打下基础。

学习目标

- 了解机器学习的应用场景。
- 掌握机器学习算法的分类方式。
- 了解机器学习常用的软件库。
- 掌握机器学习开发环境配置。

知识导图



任务 1-1　了解机器学习应用场景

任务描述

通过已经在日常生活中出现的人工智能应用，了解机器学习目前主流的应用场景。

任务目标

通过了解机器学习的应用场景，掌握不同应用场景所需要的机器学习算法。

任务实施

目前机器学习算法在很多场景下得到了应用，如人脸识别、天气预测、异常流量检测和智能制造等。

步骤 1　了解人脸识别

人脸识别是一种通过人的面部特征信息进行身份识别的技术，可以用来识别照片、视频等。人脸识别是生物特征安全的一个范畴，其他形式的生物识别软件包括语音识别、指纹识别和视网膜（或虹膜）识别。这项技术主要用于智能安防和执法场景中。

步骤 2　了解天气预测

现在的天气预报系统虽然在数值预报模式方面取得了很好的效果，但依靠人们对大气物理的理解建立的物理模式往往受到各种随机因素的干扰，不能满足复杂多变气候地区的预报需要。随着智能时代的到来，人们开始应用先进技术建立各种天气预报系统，其中机器学习算法在天气预报领域日益活跃。

步骤 3　了解异常流量检测

传统的基于静态规划匹配的网络异常检测方法难以在动态复杂的网络环境中检测出未知的异常和攻击类型，不能满足网络安全检测的要求。机器学习具有自学习和自进化的特点，它能适应复杂多变的网络环境，检测未知异常，满足实时准确检测的需要。利用机器学习方法及其自学习特性，可以对异常流量进行学习。使用合适的机器学习算法，可以发现未知的异常流量。

步骤 4 了解智能制造

机器学习在智能制造中推动整个业务运营的效率，成为智能制造的重要组成部分。它通过消除浪费和创建更精简的价值链带来更高水平的预测能力和整体洞察力，基于多种方式增加价值并最终提高企业收入和客户满意度：预测有助于生产计划更符合实际需求，并对生产进度进行实时监控和调整；预测工厂车间机器的磨损，能够提前安排维护停机时间，既可避免故障，又提高了供应链的稳定性，确保货物准时交付。

任务 1-2 机器学习算法的分类方式

任务描述

了解机器学习算法有哪些，按照不同的划分方法对机器学习算法进行分类。

任务目标

根据要求对给出的机器学习算法进行分类，并掌握每个机器学习算法的分类过程。

知识准备

人工智能是计算机学科中的一个重要分支，近些年来获得了快速的发展，在很多领域得到了广泛的应用。关于人工智能，尼尔逊（Nilsson）教授曾对它下了这样一个定义：“人工智能是关于知识的学科——怎样表示知识以及怎样获得知识并使用知识的科学。”

机器学习是人工智能领域中的一个子集，是人工智能的核心，专门研究计算机如何模拟和实现人类的学习行为。深度学习是机器学习领域中的一个重要研究方向。

任务实施

目前机器学习包含多种算法，如监督学习、无监督学习、聚类算法和分类算法等。按照不同的划分方法，可以得到不同的分类。

步骤 1 按照学习的过程分类

按照学习的过程，机器学习算法分为监督学习、半监督学习、无监督学习、主动学

习、强化学习，如图 1-1 所示。

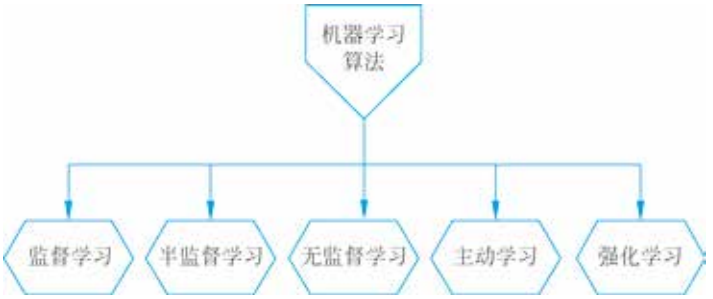


图 1-1　机器学习算法按学习的过程分类

监督学习又称有监督机器学习，它使用已经标记的数据集来训练算法，以便对数据进行分类或准确预测结果。当数据输入模型中，模型会调整其权重，直到模型得到适当的拟合。这个过程是交叉验证过程的一部分。监督学习有助于大规模地解决各种现实问题，例如，将垃圾邮件分类在收件箱的单独文件夹中。

半监督学习使用大量未标记的样本进行模型训练。半监督学习可以减少大量人为标记工作，例如，用于生物学中蛋白质的特征鉴定。

无监督学习又称无监督机器学习，其用来分析和聚类未标记的数据集。这类算法不需要人工干预就可以发现隐藏的模式或数据分组。它能够发现信息的相似性和差异性，是探索性数据分析、交叉销售策略制定、客户细分和图像识别的理想解决方案。

当数据量太大而无法标记时，可以使用主动学习。为了更准确有效地标记数据，需要设置一些数据的优先级。主动学习与半监督学习类似，都适用于标注成本较高的场景中。

强化学习是一种训练机器学习模型做出一系列决策，在不确定的、潜在的复杂环境中实现目标的机器学习算法。在强化学习中，计算机采用反复试验的方法来解决这个问题，例如，解决无人驾驶、棋牌类博弈等相关应用场景问题。

步骤 2　按照任务分类

从任务的角度，机器学习可以分为聚类算法、回归算法和分类算法这三类，如图 1-2 所示。

聚类算法是指按照一定的标准（如距离），将一个数据集划分为不同的类或簇的机器学习算法。同一簇中数据对象的相似性要尽可能大，不同簇中数据对象的差异性要尽可能大。常见的应用场景如处理目标用户群体分类等问题。

图 1-2　机器学习算法按任务分类

回归算法是一种用于数值连续的随机变量预测和建模的有监督学习算法。用例一般包括持续变化的情况，如房价预测、股票趋势或测试结果。常见的回归方法包括线性回归、逻辑回归和岭回归。常见的应用场景如处理机场客流量分布预测等问题。

分类算法根据样本的特点，将样本划分为适当的类别。具体而言，利用训练样本进行训练，得到样本特征到样本标签的映射，再利用映射得到新样本的标签，最后将样本划分为不同的类别。常用的应用场景如处理市民出行选乘公交线路预测等问题。

任务 1-3 软件库与框架

任务描述

认识了机器学习的应用场景和算法分类后，让我们继续了解机器学习需要什么软件库来支持算法的运行。

任务目标

了解目前主流的机器学习 Python 开发框架和软件库。

任务实施

基于 Java、C、Python 等编程语言开发的机器学习常用库有很多，本任务主要介绍基于 Python 的软件库。Python 软件库具有以下三个特点：其一，Python 语言的实现过程相对简单，可以减少在工程实现过程上的时间，增加在算法设计上的时间，提高工作效率；其二，Python 语言有非常丰富的库可以调用，如 NumPy、Sklearn、Pandas 等；其三，Python 代码调试比较容易。以下是机器学习中常用的 Python 库及开发框架。

步骤 1 认识 NumPy

NumPy 是 Python 语言的一个扩展库，支持大量的维数数组和矩阵运算。此外，它还为数组操作提供了大量的数学函数库，运算速度非常快，拥有在线性代数、傅立叶变换和矩阵领域工作的函数。NumPy 由特拉维斯·奥利芬特（Travis Oliphant）于 2005 年创建。它是一个开源项目，可以供开发者自由使用。

步骤 2 认识 Sklearn

Sklearn 是 Python 中最健壮的机器学习库。它通过 Python 的一致性接口为机器学习和统计建模提供了一系列有效的工具，如图 1-3 所示，包括分类、回归、聚类和降维四

类算法，可以直接调用里面的机器学习方法。Sklearn 主要是用 Python 编写的，是基于 NumPy、SciPy 和 Matplotlib 构建的。

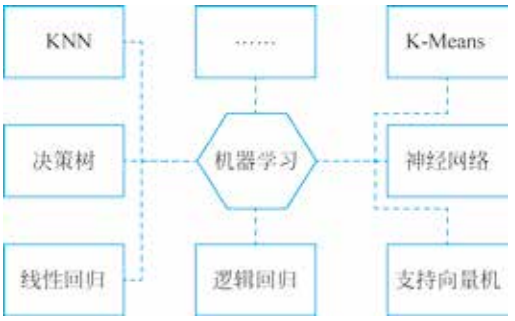


图 1-3　Sklearn 包含的机器学习方法

步骤 3　认识 Matplotlib

Matplotlib 是 Python 中的一个低级图形打印库，可用于可视化。Matplotlib 是开源的，开发者可以免费使用。Matplotlib 大部分是采用 Python 编写的，为了与平台兼容，有一些片段是采用 C、Objective-C 和 JavaScript 语言编写的。

步骤 4　认识 SciPy

SciPy 库依赖于 NumPy，它提供了方便快捷的 N 维数组操作。SciPy 库是为使用 NumPy 数组而构建的，它提供了很多对用户友好且高效的数值计算功能，如用于数值积分和优化的例程。

步骤 5　认识 TensorFlow

TensorFlow 是一个用于机器学习端到端的开源平台。它有一个由工具、库和社区资源组成的全面、灵活的生态系统，使研究人员能够推动深度学习技术发展，开发人员可以轻松地构建和部署基于深度学习的应用程序。

TensorFlow 最初是由谷歌大脑团队的研究人员和工程师开发的，用于进行机器学习和神经网络研究，提供稳定的 Python API 和 C++ API，以及其他语言的非保证向后兼容 API。TensorFlow 具有足够的通用性，可以应用于许多领域。

步骤 6　认识 CUDA

CUDA 是 NVIDIA 推出的通用并行计算框架，为图形处理器（Graphics Processing Unit，GPU）上的通用计算而开发。有了 CUDA，开发人员可以利用 GPU 的强大功能极

大地加快计算应用程序的速度。在 GPU 加速的应用程序中，CPU 针对单线程进行了优化，工作负载的顺序指令执行的部分在 CPU 上运行。

步骤 7 认识 cuDNN

CUDA 深度神经网络（CUDA Deep Neural Network，cuDNN）库是 GPU 加速的深度神经网络原始库。cuDNN 可以极大地优化标准例程的实现，如卷积层、池化层、规范化层和激活层，用于前向和后向传播。

大多数的深度学习研究人员和框架开发人员都依赖 cuDNN 实现高性能 GPU 加速。有了 cuDNN，研究人员和开发人员可以专注于训练神经网络和开发软件应用程序，而无须花费时间进行低水平的 GPU 性能调整。cuDNN 可以加速诸多被广泛使用的深度学习框架，包括 Caffe2、Chainer、Keras、MATLAB、Mxnet、PyTorch 和 TensorFlow。

任务 1-4 配置机器学习开发环境

任务描述

机器学习算法的运行需要一定的软件环境支撑，本任务主要学习配置机器学习的开发环境。

任务目标

熟练掌握机器学习开发环境配置的每个步骤。

任务实施

Anaconda 是一个围绕编程语言 Python 构建的数据科学平台，作为一个一体化的数据管理工具，它创建了一个方便访问大量数据的环境。在默认情况下，Anaconda 已经包含了 Jupyter Notebook（Jupyter Notebook 是基于网页交互式的开发工具，便于初学者调试程序）。Anaconda 可通过官网下载，读者可根据个人计算机的操作系统来选择版本。本书选择 64-bit Graphical Installer（477 MB），里面已经包含了 Python 3.8。

步骤 1 安装 Anaconda

Anaconda 下载完毕后，打开 .exe 文件，进入安装界面，如图 1-4 所示。根据安装提示单击 Next 按钮，完成 Anaconda 的安装。安装成功界面如图 1-5 所示。

Anaconda 的安装

图 1-4　安装初始界面

图 1-5　安装成功界面

步骤 2　开启 Jupyter Notebook 工具

安装完成后，在 Windows 系统下的“开始”菜单栏中打开 Anaconda Navigator。启动后的界面如图 1-6 所示。然后单击 Jupyter Notebook 图标，打开 Jupyter Notebook 工具操作界面，如图 1-7 所示。

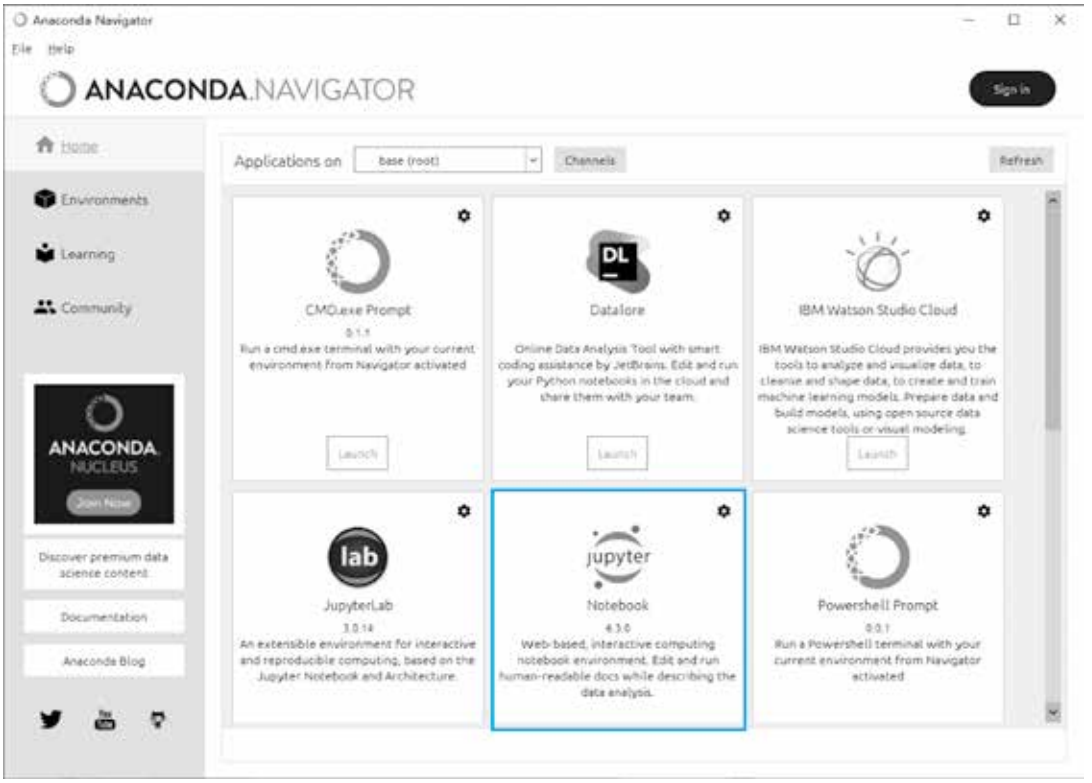


图 1-6　Anaconda Navigator 启动界面

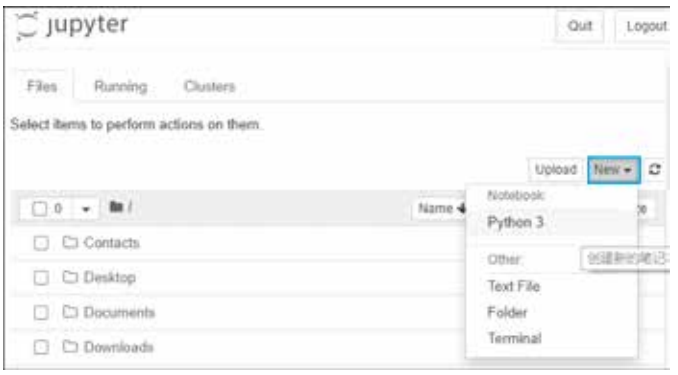


图 1-7　Jupyter Notebook 工具操作界面

步骤 3　创建 Python 脚本文件

在图 1-7 中单击 New 下拉按钮，在弹出的下拉框中选择“Python 3”标签，即可创建一个 Python 脚本文件，如图 1-8 所示。

如图 1-9 所示，从“开始”菜单中选择“Anaconda3（64-bit）”文件夹，单击“Jupyter Notebook（Anaconda3）”也可以直接进入图 1-8 所示的界面。

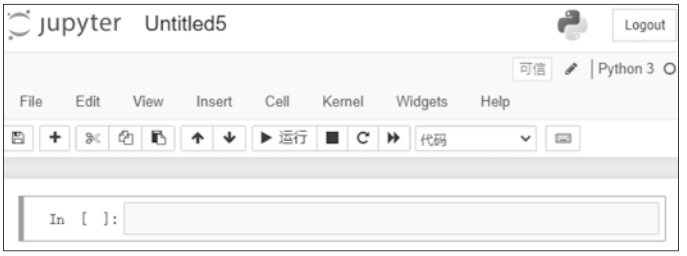


图 1-8　Python 3 脚本编程界面

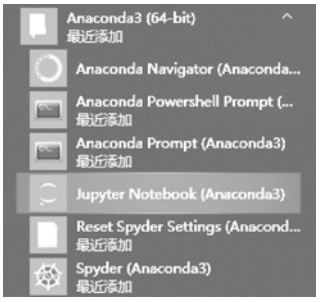


图 1-9　从文件夹 Anaconda3 下直接进入 Jupyter Notebook 编辑界面

任务 1-5　了解机器学习步骤

任务描述

根据所学习到的机器学习知识，了解机器学习算法的实现步骤，以及每个步骤的作用与意义。

任务目标

掌握机器学习算法每个步骤的含义。

任务实施

机器学习中的项目开发步骤基本类似，如图 1-10 所示。第一步是收集数据，第二步是对数据进行预处理，第三步是根据数据的特征选择合适的模型，第四步是使用数据对选择的模型进行训练，第五步是对模型评估，第六步是模型的实际测试。

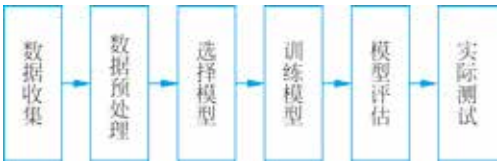


图 1-10　机器学习的步骤

机器学习的过程

步骤 1　数据收集

俗话说“巧妇难为无米之炊”，如果没有数据，那么无论多么优秀的算法都没有意义。机器学习中的数据可以通过传感器（如温湿度传感器、图像传感器、红外传感器）获取，也可以通过网络爬虫进行收集。

步骤 2　数据预处理

因为获取的数据会出现多种问题，如数据缺失、度量标准不一致、数据特征冗余较大等，所以在进行机器学习之前，需要进行数据处理，如通过主成分分析来消除数据中的冗余信息。

步骤 3　选择模型

数据处理完后，选择可能适用于此数据的模型。有时候可能同时选择多个模型，在模型的选择中，一般通过观察数据的特点，根据经验选择多个可能适合的模型，如温度的预测可以选择神经网络、支持向量回归、线性回归等模型。

步骤 4　训练模型

通过所提供的数据来求解模型中的参数，此过程称为模型训练。这个过程可能比较漫长。例如，在深度神经网络中，需要反复不断地迭代来求解网络模型中的参数。当迭代次数达到上限或者损失小于设定的标准时停止迭代，得到训练好的模型。

步骤 5　模型评估

模型训练好之后，需要对选择的模型进行评估。评估的方式有主观评价和客观评价，