HZ BOOKS
华章IT

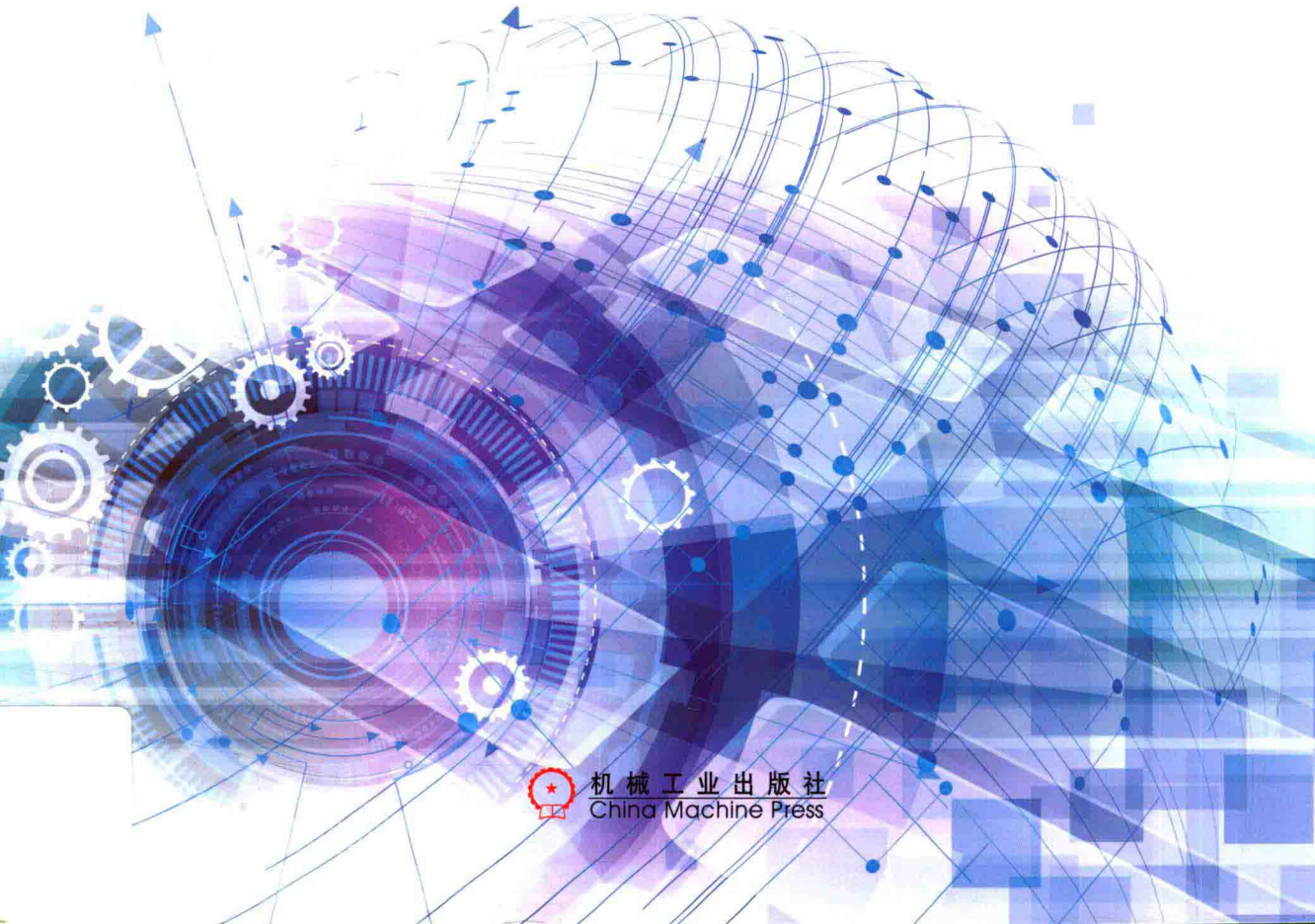
智能系统与技术丛书

作者拥有10余年Python大数据挖掘与分析经验，对AI技术驱动的智能数据分析有深入研究
为零Python基础和零AI技术基础的读者量身打造，系统讲解Python3智能数据分析必备知识，
配有大量示例代码、数据和教学资源

Get Start with the Intelligent Data Analysis by Python3

Python3智能数据分析快速入门

李明江 张良均 周东平 张尚佳 著



机械工业出版社
China Machine Press

智能系统与技术丛书

Get Start with the Intelligent Data Analysis by Python3

Python3智能数据分析快速入门

李明江 张良均 周东平 张尚佳 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Python3 智能数据分析快速入门 / 李明江等著 . —北京：机械工业出版社，2019.6
（智能系统与技术丛书）

ISBN 978-7-111-62805-7

I. P… II. 李… III. 软件工具 – 程序设计 IV. TP311.561

中国版本图书馆 CIP 数据核字（2019）第 098862 号

Python3 智能数据分析快速入门

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：李 艺 责任校对：殷 虹
印 刷：北京文昌阁彩色印刷有限责任公司 版 次：2019 年 6 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：31.25
书 号：ISBN 978-7-111-62805-7 定 价：119.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 读者信箱：hzit@hzbook.com

PREFACE

前　言

为什么要写这本书

2017 年 7 月，国务院下达了关于印发《新一代人工智能发展规划》的通知。通知明确指出要加快培养聚集人工智能高端人才，把高端人才队伍建设作为人工智能发展的重中之重。而后，2018 年 12 月，普华永道发布了《人工智能和相关技术对中国就业的净影响》，报告预测，人工智能及相关技术在未来 20 年将取代中国现有约 26% 的工作岗位，高于对英国 20% 的预估，但也能通过提升生产率和实际收入水平在中国创造出大量的新的工作机会。根据普华永道估计，人工智能对中国就业的净影响可能将创造约 12% 的净增岗位，相当于未来 20 年内增加约 9000 万个就业岗位。那么如何才能够赶上人工智能的浪潮呢？

人工智能是一门综合了计算机科学、生理学、哲学的交叉学科。凡是使用机器替代人类实现认知、识别、分析、决策等功能，均可认为使用了人工智能技术。按照技术分支，可以将人工智能分为问题求解，知识、推理与规划，学习，通信、感知与行动四个大方向。其中学习即机器学习，与智能数据分析紧密相关。

跟国外相比，我国智能数据分析在零售、银行、保险、证券等行业中的应用并不太理想。但随着市场竞争的加剧，各行业对智能数据分析技术的意愿越来越强烈，可以预计，未来几年各行业的数据分析应用一定会从传统的统计分析发展到智能数据分析应用。在大数据时代，数据过剩、人才短缺，智能数据分析专业人才的培养又需要专业知识和职业经验积累。所以，本书在编程知识的基础之上，大篇幅地描写了智能分析常备知识，希望能为智能数据分析人才的培养提供参考。

总的来说，随着云时代的来临，智能数据分析技术将具有越来越重要的战略意义。大数据已经渗透到每一个行业和业务职能领域，逐渐成为重要的生产要素，人们对于海量数据的运用将预示着新一轮生产率增长和消费者盈余浪潮的到来。智能数据分析技术

将帮助企业用户在合理时间内攫取、管理、处理、整理海量数据，为企业经营决策提供积极的帮助。智能数据分析作为数据存储和挖掘分析的前沿技术，已广泛应用于物联网、云计算、移动互联网等战略性新兴产业。虽然智能数据分析目前在国内还处于初级阶段，但是其商业价值已经显现出来，特别是有实践经验的智能数据分析人才更是成为各企业争夺的热门。为了满足日益增长的智能数据分析人才需求，很多大学开始尝试开设不同程度的智能数据分析课程。“智能数据分析”作为大数据时代的核心技术，必将成为高校数学与统计学专业的重要课程之一。

本书特色

本书作者从实践出发，总结了智能数据分析常用的方法，深入浅出地介绍了智能数据分析编程过程中的相关知识。书中涵盖 Python 环境搭建、Python 基础语法、控制语句、函数、面向对象编程、数值计算、数据处理、绘图、模型构建等内容，还配套提供了程序代码及数据。此外，每章的最后均提供课后习题，帮助读者快速掌握 Python 的使用方法。

为了帮助读者更好地使用本书，泰迪云课堂（https://edu.tipdm.org）提供了配套的教学视频。对于本书配套的原始数据文件、Python 程序代码，均可以通过关注泰迪学社微信公众号（TipDataMining），回复“图书资源”进行获取。为方便教师授课，本书还提供了 PPT 课件、教学大纲、教学进度表和教案等教学资源，教师可在泰迪学社微信公众号回复“教学资源”进行获取。

本书适用对象

❑ 开设有数据分析课程的高校的教师和学生。

目前国内不少高校将数据分析引入本科教学中，在数学、计算机、自动化、电子信息、金融等专业开设了数据分析技术相关的课程，但目前这一课程使用的教学工具仍然为 SPSS、SAS 等传统统计工具，并没有使用 Python 作为教学工具。本书提供了 Python 语言相关的从安装到使用的一系列知识，将有效指导高校教师和学生使用 Python 作为数据分析的工具之一。

❑ 数据分析开发人员。

数据分析开发人员的主要工作是将数据分析相关的算法应用于实际业务系统。本书提供了详细的机器学习与数据分析算法接口的用法与说明，能够帮助此类人员快速且有效地建立起数据分析应用的算法框架，帮助其迅速完成开发。

❑ 进行数据分析应用研究的科研人员。

许多科研院所为了更好地对科研工作进行管理，纷纷开发了适应自身特点的科研业务管理系统，并在使用过程中积累了大量的科研信息数据。但是，这些科研业务管理系统一般没有对这些数据进行深入分析，对数据所隐藏的价值并没有充分分析利用。科研人员需要数据分析工具及有关方法论来深挖科研信息的价值，从而提高科研水平。

❑ 关注高级数据分析的人员。

Python 作为广泛应用于数据分析领域的编程语言，能为数据分析人员提供快速的、可靠的分析依据。本书提供全面的 Python 智能数据分析知识，能够指导这类人员快速入门数据分析，完成指定的数据分析任务。

如何阅读本书

本书从逻辑上可分为两大部分。

第一部分是 Python 编程基础（第 1~4 章），介绍了 Python 环境搭建、Python 基础语法、控制语句、函数、面向对象编程等。第 1 章旨在让读者从全局把握 Python，了解利用 Python 进行智能数据分析的优势，并详细介绍了 Python 环境搭建与配置，同时还对两个常用集成开发环境做了详细介绍。第 2 章先对 Python 固定语法做了介绍，包括编码声明、注释、缩进等；而后介绍了 Python 常见的数据类型，包括 str、list、tuple、dict、set 等；还介绍了 Python 常用运算符，包括算术运算符、逻辑运算符、成员运算符、位运算符等。第 3 章主要对控制语句做了详细介绍，包括条件语句和循环语句，同时还介绍了和条件语句类似的异常处理 try-except-else 语句。第 4 章主要介绍了 Python 的内置函数、自定义函数、面向对象编程以及第三方库的安装与使用方法。

第二部分是数据分析编程（第 5~9 章），主要对数据分析中常用的第三方库做了详细介绍，强调在 Python 中对应函数的使用方法及其结果的解释说明。内容涵盖数值分析库 NumPy，数据处理库 pandas，绘图库 Matplotlib、Seaborn、Bokeh，机器学习与数据分析建模库 scikit-learn。这一部分涉及数据读取、数据预处理、模型构建、模型评价、结果可视化，几乎涵盖了整个数据分析过程，充分而又详细地说明了 Python 数据分析的常用操作，相信在本书的指导下，读者能够从零开始快速数据入门分析。

勘误和支持

我们已经尽最大努力避免在文本和代码中出现错误，但是由于水平有限，编写时间

仓促，书中难免出现一些疏漏和不足的地方。如果你有更多宝贵意见，欢迎在泰迪学社微信公众号回复“图书反馈”进行反馈。更多本系列图书的信息可以在“泰迪杯”数据挖掘挑战赛网站（http://www.tipdm.org/tj/index.jhtml）查阅。

张良均

2019 年于广州

CONTENTS

目　录

CHAPTER 1

第 1 章

Python 概述

人工智能已成为当今世界上最受人瞩目的领域之一。各大公司纷纷在人工智能领域展开角逐，Google、Facebook、Amazon 都已经在这个领域里取得了令人瞩目的成果。同时，机器学习作为人工智能的一个分支，已经在不知不觉中深入人们的生活中，如电商平台上的商品推荐和街头监控的图像识别等。由于 Python 第三方库中集成了诸多算法，所以本书选择 Python 作为机器学习语言进行介绍。

1.1 Python 语言介绍

Python 是一门集解释性、编译性、互动性和面向对象为一体的高层次计算机程序语言，也是一门功能强大而完善的通用型语言，已有 20 多年的发展历史，技术成熟且稳定。相比于 C++ 或 Java，Python 让开发者能够用更少的代码实现更多的想法。

1.1.1 Python 的发展史

荷兰人 Guido van Rossum 是 Python 的创始人。1989 年圣诞节期间，Guido 决心开发一个新的脚本解释程序用来打发圣诞节的无趣。Python 这个名字并不是来源于蟒蛇，而是因为 Guido 是一个名为 Monty Python 的飞行马戏团的爱好者。他希望这个新的叫作 Python 的语言能实现他的理念，成为一种位于 C 和 shell 之间、功能全面、易学易用、可拓展的语言。

Python 从 ABC 语言上继承了一部分特性，如使用冒号“:”和缩进来表示程序块，而在 C 语言中使用 {} 来表示程序块，行尾没有分号，for 和 if 结构中也没有括号 ()。Guido 认为，ABC 语言非常优美和强大，是专门为非专业程序员设计的。Guido 认为 ABC 语言没有成功的原因在于其非开放性。于是，Guido 决心在 Python 中避免复现这一错误，这收获了非常好的效果，Python 能够完美结合 C 语言和其他的一些语言。

Python 就这样从 Guido 手中诞生。1991 年，第一个 Python 编译器（解释器）诞生。

它使用C语言实现，并能够调用C库（.so）文件。1994年1月，Python 1.0版本正式发布。Python 2.0于2000年10月16日发布，增加实现完整的垃圾回收的功能，并且支持Unicode。同时，整个开发过程更加透明，社群对开发进度的影响逐渐扩大。2008年12月，Python 3.0正式发布，此时Python 3又被称为“Python 3000”或者“Py3K”，此版不完全兼容之前的Python源代码。出于兼容性的考虑，很多新特性后来也被移植到旧的Python 2.6/2.7版本上。截至2018年6月27日，Python已经更新至最新版的Python 3.7.0版本。

Python的语法很多来自C语言，但又受到ABC语言的强烈影响。Python从ABC语言中发展起来，结合了UNIX shell和C语言的习惯，最终成为一门为众多UNIX系统和Linux系统开发者所青睐的开发语言。截至2018年8月的TIOBE语言排行，Python的名次已上升至第4名，成为全球范围内仅次于Java、C语言及C++的开发语言，如图1-1所示。

Aug 2018	Aug 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.881%	+3.92%
2	2		C	14.966%	+8.49%
3	3		C++	7.471%	+1.92%
4	5	^	Python	6.992%	+3.30%
5	6	^	Visual Basic .NET	4.762%	+2.19%
6	4	v	C#	3.541%	-0.65%
7	7		PHP	2.925%	+0.63%
8	8		JavaScript	2.411%	+0.31%
9	-	^^	SQL	2.316%	+2.32%
10	14	^^	Assembly language	1.409%	-0.40%

图1-1　TIOBE语言排行榜

1.1.2　Python特性

Python语言是一门解释型、动态、强类型的面向对象编程语言。Python的解释型特性与计算机执行程序的步骤有关。由于计算机无法理解除机器语言以外的任何语言，所以必须将程序员编写的程序语言翻译成机器语言，计算机才能执行程序。按翻译的时机的不同，翻译的过程可分为编译和解释两种。在程序编写完成后进行翻译的过程称为编译，而在编写后不进行翻译在执行时才进行翻译的过程称为解释。解释型语言与编译型语言不同，解释性语言编写的程序不需要编译，节省了一道工序，在运行程序的时候才翻译。而编译型语言编写的程序在执行之前，需要一个专门的编译过程，将程序编译成为机器语言的文件。这使得Python比起其他编译型语言发布时更为便捷，但因为在执行时有一个翻译的过程，Python在执行效率上比编译型语言低一些。

动态类型语言是指在程序运行期间才去做数据类型检查的语言。在用动态类型的语言编程时，永远也不用给任何变量指定数据类型，该语言会在第一次赋值给变量时，在内部将数据类型记录下来。Python就是一种典型的动态类型语言。而静态语言与动态语言相反，静态语言的数据类型是在编译期间检查的，在编写程序时需要声明所有变量的数据类型。

C/C++ 是静态类型语言的典型代表，其他的静态类型语言还有 C#、Java 等。

强类型定义语言（Explicit Type Conversion）是指一门计算机语言是否为强制数据类型定义的语言。强类型语言中的变量被指定为某个数据类型后，在被强制转换前永远是该数据类型。而弱类型语言与强类型定义语言相反，一个变量的数据类型是不确定的。Python 是一门强类型的语言，Python 不会对数据类型作隐式转换，而是调用方法对数据类型进行强制转换。除 Python 外，Java、C、C++ 和 C# 也是典型的强类型语言。

1.1.3　Python 应用领域

Python 作为一个设计优秀的程序语言，现在已广泛应用于各种领域，依靠其强大的第三方类库，Python 在各个领域都能发挥巨大的作用。

1. 数值计算

数值计算是数据挖掘、机器学习的基础。Python 提供多种强大的扩展库用于数值计算，常用的数值计算库如表 1-1 所示。

表 1-1　常用数值计算库及其简介

数值计算库	简　介
NumPy	支持多维数组与矩阵运算，也针对数组运算提供大量的数学函数库。通常与 SciPy 和 Matplotlib 一起使用，支持比 Python 更多种类的数值类型，其中定义的最重要的对象是称为 ndarray 的 n 维数组类型，用于描述相同类型的元素集合，可以使用基于 0 的索引访问集合中元素
SciPy	在 NumPy 库的基础上增加了众多的数学、科学及工程计算中常用的库函数，如线性代数、常微分方程数值求解、信号处理、图像处理、稀疏矩阵等，可进行插值处理、信号滤波，以及使用 C 语言加速计算
Pandas	基于 NumPy 的一种工具，为解决数据分析任务而生。纳入大量库和一些标准的数据模型，提供高效地操作大型数据集所需的工具及大量的能快速便捷处理数据的函数和方法，为时间序列分析提供很好的支持，提供多种数据结构，如 Series、Time-Series、DataFrame 和 Panel

2. 数据可视化

数据可视化是展示数据、理解数据的有效手段，常用的 Python 数据可视化库如表 1-2 所示。

表 1-2　Python 数据可视化库

数据可视化库	简　介
Matplotlib	第一个 Python 可视化库，有许多别的程序库都是建立在其基础上或者直接调用该库，可以很方便地得到数据的大致信息，功能非常强大，但也非常复杂
Seaborn	利用了 Matplotlib，用简洁的代码来制作好看的图表。与 Matplotlib 最大的区别为默认绘图风格和色彩搭配都具有现代美感
ggplot	基于 R 的一个作图库 ggplot2，同时利用了源于《图像语法》（The Grammar of Graphics）中的概念，允许叠加不同的图层来完成一幅图，并不适用于制作非常个性化的图像，为操作的简洁度而牺牲了图像的复杂度

（续）

数据可视化库	简　介
Bokeh	跟 ggplot 一样，Bokeh 也基于《图形语法》的概念。与 ggplot 不同之处为它完全基于 Python 而不是从 R 处引用。长处在于能用于制作可交互、可直接用于网络的图表。图表可以输出为 JSON 对象、HTML 文档或者可交互的网络应用。Bokeh 也支持数据流和实时数据，为不同的用户提供了 3 种控制水平，最高的控制水平用于快速制图，主要用于制作常用图像；中等控制水平与 Matplotlib 一样允许开发人员控制图像的基本元素（例如分布图中的点）；最低的控制水平主要面向开发人员和软件工程师。没有默认值，需要定义图表的每一个元素
Plotly	可以通过 Python notebook 使用，与 Bokeh 一样致力于交互图表的制作，但提供在别的库中几乎没有的几种图表类型，如等值线图、树形图和三维图表
pygal	与 Bokeh 和 Plotly 一样，提供可直接嵌入网络浏览器的可交互图像。与其他两者的主要区别在于可将图表输出为 SVG 格式，所有的图表都被封装成方法，且默认的风格也很漂亮，用几行代码就可以很容易地制作出漂亮的图表
geoplotlib	用于制作地图和地理相关数据的工具箱。可用来制作多种地图，比如等值区域图、热度图、点密度图。必须安装 Pyglet（一个面向对象编程接口）方可使用
missingno	用图像的方式快速评估数据缺失的情况，可根据数据的完整度对数据进行排序或过滤，或者根据热度图或树状图对数据进行修正

3. Web 开发

Web 应用开发可以说是目前软件开发中最重要的部分。Python 提供各种 Web 开发框架，帮助使用者快速实现功能开发。常用的 Python 网络开发类库如表 1-3 所示。

表 1-3　Python 常用网络开发类库及其简介

网络编程库	简　介
Socket	一个套接字通讯底层库，用于在服务器和客户端间建立 TCP 或 UDP 连接，通过连接发送请求与响应
Django	一个高级的 Python Web 框架，支持快速开发，提供从模板引擎到 ORM 所需的一切东西，使用该库构建 App 时，必须遵循 Django 的方式
Flask	一个基于 Werkzeug、Jinja 2 的 Python 轻量级框架（microframework），默认配备 Jinja 模板引擎，也包含其他模板引擎或 ORM 供选择，适合用来编写 API 服务（RESTful rervices）
Twisted	一个使用 Python 实现的基于事件驱动的网络引擎框架，建立在 deferred object 之上，一个通过异步架构实现的高性能的引擎，不适用于编写常规的 Web Apps，更适用于底层网络
Tornado	一个由 FriendFeed 开发的 Python Web 框架和异步网络库，采用非阻塞网络 I/O 模型，可以处理数以千计的网络连接。对于 long polling、WebSockets 和其他需要长时间实时连接的 Apps，Tornado 是一个理想的 Web 框架，它介于 Django 和 Flask 之间，能很好地处理 C10K 问题

4. 数据库管理

数据库是企业用于存放数据的主要工具，数据库管理包括了数据定义、数据操作、数据库运行管理、数据组织、数据库库保护、数据库维护等。Python 提供了所有主流关系数据库管理接口，常用的 Python MySQL 连接库及其简介如表 1-4 所示。

表 1-4　MySQL 连接库

MySQL 连接库	简　介
MySQL-python	又称 MySQLdb，是 Python 连接 MySQL 最流行的一个驱动，很多框架也基于此库进行开发。只支持 Python 2.x，且安装时有许多前置条件。由于该库基于 C 语言开发，在 Windows 平台上的安装非常不友好，经常出现失败的情况，现在基本不推荐使用，取代品为衍生版本
mysqlclient	完全兼容 MySQLdb，同时支持 Python 3.x，是 Django ORM 的依赖工具，可使用原生 SQL 来操作数据库，安装方式与 MySQLdb 一致
PyMySQL	纯 Python 实现的驱动，速度比 MySQLdb 慢，最大的特点为安装方式简洁，同时也兼容 MySQL-python
SQLAlchemy	一种既支持原生 SQL，又支持 ORM 的工具。ORM 是 Python 对象与数据库关系表的一种映射关系，可有效提高写代码的速度，同时兼容多种数据库系统，如 SQLite、MySQL、PostgreSQL，代价为性能上的一些损失

5. 自动化运维

运维的主要内容包括保障业务长期稳定运行、保障数据安全可靠、自动化完成部署任务。Python 能够满足绝大部分自动化运维的需求，目前在 Linux 运维中已用 Python 实现的应用如表 1-5 所示。

表 1-5　Python 自动化运维应用及相关功能

自动化运维应用	功　能
jumpsever 跳板机	一种由 Python 编写的开源跳板机（堡垒机）系统，实现了跳板机的基本功能，包含认证、授权和审计，集成了 Ansible、批量命令等。支持 WebTerminal Bootstrap 编写，界面美观，自动收集硬件信息，支持录像回放、命令搜索、实时监控、批量上传下载等功能，基于 SSH 协议进行管理，客户端无须安装 agent。主要用于解决可视化安全管理，因完全开源，容易再次开发
Magedu 分布式监控系统	一种用 Python 开发的自动化监控系统，可监控常用系统服务、应用、网络设备，可在一台主机上监控多个不同服务，不同服务的监控间隔可以不同，同一个服务在不同主机上的监控间隔、报警阈值可以不同，并提供数据可视化界面
Magedu 的 CMDB	一种用 Python 开发的硬件管理系统，包含采集硬件数据、API、页面管理 3 部分功能，主要用于自动化管理笔记本、路由器等常见设备的日常使用。由服务器的客户端采集硬件数据，将硬件信息发送至 API，API 负责将获取的数据保存至数据库中，后台管理程序负责对服务器信息进行配置和展示
任务调度系统	一种由 Python 开发的任务调度系统，主要用于自动化地将一个服务进程分布到其他多个机器的多个进程中，一个服务进程可作为调度者依靠网络通信完成这一工作
Python 运维流程系统	一种使用 Python 语言编写的调度和监控工作流的平台，内部用于创建、监控和调整数据管道。允许工作流开发人员轻松创建、维护和周期性地调度运行工作流，包括了如数据存储、增长分析、Email 发送、A/B 测试等诸多跨多部门的用例

6. GUI 编程

GUI（Graphical User Interface，图形用户界面）是指采用图形方式显示的计算机操作用户界面。Python 提供多个图形开发界面的库用于 GUI 编程，常用 Python GUI 库如表 1-6 所示。

表 1-6 Python 常用 GUI 库及其简介

GUI 库	简 介
Tkinter	一个 Python 的标准 GUI 库，可以快速地创建 GUI 应用程序，可以在大多数的 UNIX 平台下使用，同样可以应用在 Windows 和 Macintosh 系统中，Tkinter 8.0 的后续版本可以实现本地窗口风格，并良好地运行在绝大多数平台中
wxPython	一款开源软件跨平台 GUI 库 wxWidgets 的 Python 封装和 Python 模块，是 Python 语言的一套优秀的 GUI 图形库，允许程序员很方便地创建完整的、功能健全的 GUI 用户界面
PyQt	一个创建 GUI 应用程序的工具库，是 Python 编程语言和 Qt 的成功融合，可以运行在所有主要操作系统上，包括 UNIX、Windows 和 Mac。PyQt 采用双许可证，开发人员可以选择 GPL 和商业许可，从 PyQt 的版本 4 开始，GPL 许可证可用于所有支持的平台
PySide	一个跨平台的应用程式框架 Qt 的 Python 绑定版本，提供与 PyQt 类似的功能，并相容 API，但与 PyQt 不同处为其使用 LGPL 授权

1.1.4 Python 机器学习优势

Python 是机器学习领域最优秀的编程语言之一，与同样支持机器学习且自带机器学习应用的 MATLAB 不同的是，由于 Python 是开源项目，所以几乎所有必要的组件都是完全免费的。当前机器学习领域的主流编程语言为 Python、R、MATLAB、Java 和 C/C++，如表 1-7 所示。

表 1-7 主流机器学习编程语言

编程语言	特 点	是否开源	运行速度	机器学习支持
Python	简洁、灵活、可读性高，易于理解，可连接其他语言	是	较快	NumPy、SciPy 和 Pandas 扩展库提供良好的数学处理与科学计算支持，并拥有海量开源框架、专业机器学习库，如 Scikit-learn、Theano 和 TensorFlow
R	开发速度快，社区强大	是	一般	CRAN 资源库中有海量的工具包，囊括几乎所有的机器学习算法、数据测试和分析过程
MATLAB	精于数学计算，图表功能强大，提供良好的调试器用于测试与支持	否	较快	拥有丰富的工具箱，基于矩阵，经常用于机器学习算法的原型设计
Java	功能强大，简单易用，使用人数最多	是	快	拥有开发大规模分布式学习系统的多种工具，如 Spark+MLlib、Mahout、H2O 和 Deeplearning4j
C/C++	计算速度极快，内存效率极高，对初学者极不友好，是编写底层的理想语言	是	极快	常用于嵌入式智能系统，也拥有机器学习库，如 LibSVM、Shark 和 mlpack

相比起 Java 与 C/C++，Python 代码非常容易阅读和学习，使得大多数从事机器学习和人工智能的研究（工作）人员能以最方便的方式来实现自己的想法。Python 具有严格且一致的语法风格，这使得每个 Python 使用者都可以更好地理解其他人的 Python 代码，而其他语言的语法有可能会导致混淆和不一致的编程范例。