浙江省普通高校“十三五”新形态教材　　大数据与人工智能技术丛书

Python 数据分析与实践

◎ 柳毅 主编　毛峰 李艺 副主编

400分钟 视频讲解

教学课件　教学大纲　程序源码

电子教案　习题答案

清華大學出版社

浙江省普通高校“十三五”新形态教材　　大数据与人工智能技术丛书

Python 数据分析与实践

◎ 柳毅 主编　毛峰 李艺 副主编

清華大學出版社
北京

内 容 简 介

Python是信息管理与信息系统、电子商务等信息管理类本科学生进行数据分析所需要掌握的基础性语言和分析工具，是未来学生掌握大数据分析技术的学习基础。本书共分12章，着重讲述Python语言和数据分析工具包的应用。第1章主要介绍Python的发展历史、特点、集成开发环境、内置模块、帮助的使用等内容；第2章主要介绍Python语言的基础知识；第3章主要介绍Python中的常用数据结构，包括序列、字典、集合等，以及函数的定义和调用等；第4章主要介绍Python中类、对象和方法的相关内容；第5章主要介绍Python进行数据分析常用的NumPy、Pandas、Matplotlib、SciPy和Scikit-learn等基础库内容；第6章主要介绍网络数据获取的HTML和XML两种网页组织形式，以及urllib和BeautifulSoup4两个模块内容；第7章主要介绍文件的操作；第8章主要介绍数据可视化，以及使用Python绘制图表的知识；第9章主要介绍利用Python进行数据库应用开发；第10、11章主要介绍Python机器学习的基本概念以及有监督、无监督学习算法的原理；第12章主要介绍Python在地理空间分析上的应用。本书中的代码均在Python 3.5中测试通过。

本书一方面侧重对Python数据分析基础知识的讲解，另一方面注重Python数据处理方法的应用。本书适合作为计算机科学与技术专业学生学习数据分析的入门教材，也适合作为Python爱好者的参考书。

图书在版编目(CIP)数据

Python数据分析与实践/柳毅主编. —北京：清华大学出版社，2019
(大数据与人工智能技术丛书)
ISBN 978-7-302-51579-1

Ⅰ. ①P…　Ⅱ. ①柳…　Ⅲ. ①软件工具—程序设计　Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2018)第257419号

策划编辑：魏江江
责任编辑：王冰飞
封面设计：刘　键
责任校对：时翠兰
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：21.5　　**字　　数**：370千字
版　　次：2019年6月第1版　　**印　　次**：2019年6月第1次印刷
印　　数：1～1500
定　　价：59.00元

产品编号：080337-01

序

人类社会已经进入数字经济时代，大数据、云计算、机器学习、人工智能等技术纷至沓来，数据的管理和应用已经渗透到每一个行业的业务领域，成为当今乃至将来企业运作的基础资产。只有掌握数据并善于运用数据的人，才会在未来社会日益激烈的竞争环境中保持领先地位。

Python 语言很好地融合了大数据分析、机器学习以及人工智能技术，是目前大数据和机器学习领域热门的语言之一。本书为学习者深入浅出地介绍 Python 数据分析的原理、建模过程、统计应用方法，具有极强的实践性。

本书基于 Python 3.5 工具环境，通过实践案例讲解 Python 控制、处理、分析数据的算法和工具，让学生了解如何利用 Python 编程和数据处理库（包括 NumPy、SciPy、Matplotlib、Pandas 及 Scikit-learn 等）高效地解决各种数据分析问题，发挥 Python 在数据分析、可视化、机器学习、地理空间信息分析方面的优势，引导读者成为数据分析的高手。

本书内容严谨，逻辑清晰，可供计算机科学与技术专业以及信息管理与信息系统、电子商务等信息管理类的本科与研究生学习使用，为大数据时代的企业管理、市场营销、金融等行业从事大量数据分析的从业人员提供科学的学习资源。

浙江工业大学计算机科学与技术学院院长

国家级教学名师、教授、博士生导师

2019 年 3 月

前言

Python是大数据时代非常受欢迎的数据分析编程语言，近年来随着机器学习、云计算、人工智能等技术的发展，Python的流行趋势扶摇直上，已经成为数据分析和数据科学事实上的标准语言和标准平台之一。

本书针对数据分析人员和Python编程学习者进行内容编排和章节讲述，Python数据分析整个学习路线图计划分成16周，120天左右。本书主要内容包括以下五大部分。

(1) Python工作环境及基础语法知识：认识Python程序运行方式，使用Python 3.5开发集成环境与工具；学习Python程序基本结构，理解Python的面向对象定义和类、对象的操作方法，以及Python异常处理机制。本部分为基础内容，建议学习时间为4周。

(2) Python数据分析相关知识：Python生态系统为数据分析师和数据科学家提供了各种程序库，例如NumPy、SciPy、Pandas和Matplotlib，使其在数据分析领域也有广泛的应用。Python数据分析的学习主要是对相关库的使用，例如数据整理需要用到NumPy库，数据描述与分析则主要用到Pandas库。由于有前面的学习基础，本部分学习时间建议为3周。

(3) Python数据可视化：Python数据可视化的过程就是学习Matplotlib库的过程，Matplotlib库包含丰富的数据可视化资源，地图、3D等都有涉及，基于前面两部分的学习经验，这部分内容在两周内基本可以完成。

(4) Python机器学习：Scikit-learn是本书所使用的核心程序库，依托于上述几种工具包，封装了大量的经典以及最新的机器学习模型。通过介绍有监督和无监督机器学习原理，学习有监督学习的线性回归、Logistic回归、朴素贝叶斯、SVM、KNN和决策树等几个常用算法，以及无监督学习的K-Means聚类算法。在前面三部分学习的基础上，本部分内容建议学习时间在4周左右。

(5) Python地理空间数据分析应用：向读者介绍地理空间分析的基本概念和常

用的地理空间数据，然后介绍 Python 中与地理数据处理和地理分析相关的工具，最后以 Python 处理和分析矢量数据与栅格数据的方法，对浙江省实施的“五水共治”行动中劣五类水体在地理空间模型上分布的卫星图像数据进行可视化数据分析的综合应用，本部分内容建议学习时间为 3 周。

本书编写人员具有丰富的 Python 数据分析实践经验和多年的信息管理教学能力，第 1～3 章由沈阳工业大学李艺老师编写；第 4～7、10 章由杭州电子科技大学柳毅老师编写；第 8、9、11、12 章由杭州电子科技大学毛峰老师编写；王健、陆佳涣等硕士研究生参与了本书相关章节内容和程序代码的完善工作；浙江工业大学计算机科学与技术学院院长、国家教学名师王万良教授对本书进行了认真的审阅，并提出许多宝贵的建设性意见，使本书内容日臻完善，在此对他们所付出的辛勤劳动表示诚挚的感谢。

本书结合大数据管理与应用的最新发展，针对计算机科学与技术、信息管理与信息系统、电子商务等经管类本科教学特点进行撰写。本书提供教学课件、教学大纲、电子教案、习题答案和程序源码，读者可以扫描封底的课件二维码下载。本书还提供 400 分钟的视频讲解，扫描书中的二维码，可以在线观看。

由于编者水平所限，书中难免有疏漏之处，敬请读者批评指正。

编　者

2019 年 3 月

目 录

源码下载

第1章

Python简介

本章学习目标：

- 了解 Python 语言的发展历史及特点
- 深刻了解 Python 2 和 Python 3 的区别
- 熟练掌握 Python 中 IDLE 的编程特点
- 熟练掌握 Python 的开发环境

本章首先向读者介绍 Python 发展的历史及其特点；然后分析 Python 2 与 Python 3 不同的编程特点；接着以 Eclipse+PyDev 为例对 Python 的环境搭建进行介绍，并对 Python 自带的 IDLE 界面进行讲解；最后对 Python 的各种开发环境进行讲解。

1.1 Python 语言的发展史

Python 的作者是荷兰人 Guido von Rossum，尽管拥有阿姆斯特丹大学数学和计算机双硕士学位，Guido 总趋向于做计算机相关的工作，并热衷于做任何与编程相关的活儿。Guido 接触并使用过 Pascal、C、Fortran 等语言，这些语言的基本设计原则是让计算机能更快地运行。在 20 世纪 80 年代，虽然 IBM 和苹果公司已经掀起了个

人计算机浪潮，但这些个人计算机的配置很低。所有的编译器的核心是做优化，以便让程序能够运行。为了提高效率，语言也迫使程序员像计算机一样思考，以便能写出更适合计算机的程序。在那个时代，程序员恨不得拥有使用计算机每一点空间的能力，有人甚至认为C语言的指针是在浪费内存。至于动态类型、内存自动管理、面向对象等，那就不用想了，否则会让计算机陷入瘫痪。

这种编程方式让Guido感到苦恼。Guido知道如何用C语言写出一个功能，但整个编写过程需要耗费大量的时间。他的另一个选择是Shell。Bourne Shell作为UNIX系统的解释器已经长期存在，UNIX的管理员们经常用Shell去写一些简单的脚本，以进行一些系统维护的工作，比如定期备份、文件系统管理等。许多用C语言编写上百行的程序，在Shell下只用几行就可以完成。然而，Shell的本质是调用命令，它并不是一个真正的语言。比如说，Shell没有数值型的数据类型，即使加法运算也很复杂。总之，Shell不能全面地调用计算机的功能。

Guido希望有一种语言能够像C语言那样可以全面调用计算机的功能接口，又能够像Shell那样可以轻松地编程，ABC语言让Guido看到希望。ABC是由荷兰的数学和计算机研究所开发的。Guido在该研究所工作，并参与到ABC语言的开发中。与当时的大部分语言不同，ABC语言的目标是“让用户感觉更好”。ABC语言希望让程序变得容易阅读、使用、记忆和学习，并以此来激发人们学习编程的兴趣。尽管已经具备了良好的可读性和易用性，但ABC语言最终没有流行起来。在当时ABC语言的设计也存在一些致命的问题。

(1) ABC语言不是模块化语言。如果想在ABC语言中增加功能，比如对图形化的支持，就必须改动很多地方。

(2) ABC语言不能直接操作文件系统。尽管用户可以通过诸如文本流的方式导入数据，但ABC语言无法直接读写文件，输入输出的困难对于计算机语言来说是致命的。

(3) ABC语言用自然语言的方式来表达程序的意义，然而对于程序员来说，他们更习惯用function或者define来定义一个函数，用等号来分配变量。尽管ABC语言很特别，但学习难度也很大。

(4) ABC语言编译器很大，必须被保存在磁带上，这样ABC语言很难快速传播。

1989年，为了打发圣诞节假期，Guido开始写Python语言的编译器。“Python”这个名字来自Guido所挚爱的电视剧 *Monty Python's Flying Circus*。他希望

Python语言能符合他的理想——创造一种C和Shell之间的功能全面、易学易用、可拓展的语言。1991年，第一个Python编译器诞生。它是用C语言实现的，并能够调用C语言的库文件。从一诞生，Python已经具有了类、函数、异常处理，包含表和词典在内的核心数据类型，以及以模块为基础的拓展系统。

Python语法很多来自C语言，但又受到ABC语言的很大影响。来自ABC语言的一些规定直到今天还有争议，比如强制缩进，但这些语法规定让Python容易阅读。另外，Python聪明地选择服从一些惯例，特别是C语言的惯例，比如回归等号赋值。

Python从一开始就特别注重可拓展性，Python可以在多个层次上拓展。用户可以直接引入.py文件，也可以引用C语言的库。Python程序员可以快速地使用Python写.py文件作为拓展模块，但当性能是考虑的重要因素时，Python程序员可以深入底层写C程序，编译为.so文件引入到Python中使用。Python就好像是使用钢构建房一样，先规定好大的框架，而程序员可以在此框架下自由地拓展或更改。

最初的Python完全由Guido本人开发。Python受到Guido同事的欢迎，他们迅速地反馈使用意见，并参与到Python的改进中。Guido和一些同事构成Python的核心团队，他们将自己的大部分业余时间用于hack Python。随后，Python被拓展到研究所之外。Python将许多机器层面上的细节隐藏，交给编译器处理，并凸显出逻辑层面的编程思考。Python程序员可以花更多时间用于思考程序的逻辑，而不是具体的实现细节，这一特征吸引了广大的程序员，Python开始流行。

Guido维护了一个邮件列表，Python用户能通过邮件进行交流。Python用户来自许多领域，不同的背景对Python也有不同的需求。Python相当开放，又容易拓展，所以当用户不满足现有功能时很容易对Python进行拓展或改造。随后，这些用户将改动发给Guido，并由Guido决定是否将新的特征加入到Python或者标准库中。如果代码能被纳入Python自身或者标准库，这将是极大的荣誉。由于Guido有着至高无上的决定权，所以他被称为“终身的仁慈独裁者”。

Python被称为“Battery Included”，是说它的标准库功能强大。这是整个社区的贡献，Python的开发者来自不同领域，他们将不同领域的优点带给Python，比如Python标准库中的正则表达式参考Perl，而lambda匿名函数以及map()、filter()、reduce()等函数参考了Lisp。在Python的开发过程中，社区起到了重要的作用。