

将机器学习、统计、通过Web应用程序进行原型设计三种不同的知识组合在一起，并与云服务提供商合作；提供了一个简单的、云相关的、技术无限的指南，帮助你尽可能快地将Python建模工作推广到世界各地；讨论了在Web上快速进行统计和建模工作原型设计的艺术。

Monetizing Machine Learning

Quickly Turn Python ML Ideas into Web Applications on the Serverless Cloud

机器学习即服务

将Python机器学习创意
快速转变为云端Web应用程序

[美] 曼纽尔·阿米纳特吉 (Manuel Amunategui) 著
迈赫迪·洛佩伊 (Mehdi Roopaei)
刘世民 山金孝 史天 肖力 译



机械工业出版社
China Machine Press

Monetizing Machine Learning

Quickly Turn Python ML Ideas into Web Applications on the Serverless Cloud

机器学习即服务

将Python机器学习创意
快速转变为云端Web应用程序

[美] 曼纽尔·阿米纳特吉 (Manuel Amunategui) 著
迈赫迪·洛佩伊 (Mehdi Roopaei)
刘世民 山金孝 史天 肖力 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

机器学习即服务: 将 Python 机器学习创意快速转变为云端 Web 应用程序 / (美) 曼纽尔·阿米纳特吉 (Manuel Amunategui), (美) 迈赫迪·洛佩伊 (Mehdi Roopaei) 著; 刘世民等译. —北京: 机械工业出版社, 2019.6

(智能系统与技术丛书)

书名原文: Monetizing Machine Learning: Quickly Turn Python ML Ideas into Web Applications on the Serverless Cloud

ISBN 978-7-111-62703-6

I. 机… II. ①曼… ②迈… ③刘… 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 086816 号

本书版权登记号: 图字 01-2019-0942

First published in English under the title

Monetizing Machine Learning: Quickly Turn Python ML Ideas into Web Applications on the Serverless Cloud

by Manuel Amunategui, Mehdi Roopaei

Copyright © 2018 by Manuel Amunategui, Mehdi Roopaei

This edition has been translated and published under licence from

Apress Media, LLC, part of Springer Nature.

Chinese simplified language edition published by China Machine Press, Copyright © 2019.

This edition is licensed for distribution and sale in the People's Republic of China only, excluding Hong Kong, Taiwan and Macao and may not be distributed and sold elsewhere.

本书原版由 Apress 出版社出版。

本书简体字中文版由 Apress 出版社授权机械工业出版社独家出版。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

此版本仅限在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 销售发行, 未经授权的本书出口将被视为违反版权法的行为。

机器学习即服务

将 Python 机器学习创意快速转变为云端 Web 应用程序

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: 朱秀英

责任校对: 殷虹

印刷: 北京瑞德印刷有限公司

版次: 2019 年 6 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 23.25

书号: ISBN 978-7-111-62703-6

定价: 99.00 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

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

直强调的，把机器学习和算法理论应用到实际场景中，进而解决特定领域问题，才是我们学习的最终目的，我相信这也是机器学习和人工智能的最终目的。

在此，我们把这本书翻译并推荐给各位读者，希望本书能为大家带来一个轻松愉快的机器学习之旅，能让大家发现 THE TRANSLATOR'S WORDS 1 译者，事竟成，祝各位朋友心想事成！

译者序 1

山金李

近几年来，人工智能（AI）一直在飞速发展着，它正在开启一个全新的创新时代。从智能音箱到各种机器人，从人脸识别到语言翻译，从智能推荐到自动驾驶，人工智能在各行各业应用得越来越广泛。

人工智能技术是引领未来的创新性技术，它将拥有非常光明的前景。译者对这一点深信不疑。于是，大概半年前，为了学习 AI，译者买来了大学时期曾经学过的《概率论》《微积分》《线性代数》等数学书籍，以及几本讲述各种算法和模型的大部头。尽管很努力地学习，但是自我感觉效果不太理想。一来要学习的内容非常多，二来各种算法非常枯燥，三来没有太多使用机会，因此很难在短时间内就理解得比较透彻，有时甚至有了打退堂鼓的念头。

直至看到 Apress 出版的《Monetizing Machine Learning》，我才知道学习 AI 还有如此轻松有趣的方法。书中实现了多个包含 AI 能力的完整 Web 应用程序，包括利用 Python、Flask、Google Analytics、Google Maps、PythonAnywhere、Memberful 等编程工具和云服务，在 GCP 上实现一个交互式画布并利用 TensorFlow 来对手写数字进行预测，在 GCP 上使用奇异值分解算法来向用户推荐电影，在 AWS 上使用朴素贝叶斯算法来为高尔夫球手实时预测天气，在 Azure 上利用云端回归系数来预测共享单车租赁需求等。从这些例子可以看出，本书另辟蹊径，不从各种算法和模型推理入手，而是一步一步从产品设计开始，到代码实现，再到云上部署，让读者学习如何利用 AI 来实现这些 Web 应用程序中的智能能力。

本书的顺利出版是各位编辑和我们四人翻译团队通力合作的结果。感谢机械工业出版社对我们的信任和支持，以及各位编辑的专业和细心，感谢四人小组的互相支持和鼓励。还要感谢我的妻子和儿子，谢谢你们对我的默默支持。

最后，由于本人水平有限，书中难免会出现错误或者不妥之处，欢迎批评指正。希望本书能给大家带来一个轻松、有趣、有成就感，甚至还能赚到钱的 AI 学习奇妙之旅。

刘世民 (Sammy Liu)

图书在版编目(CIP)数据

机器学习图灵秀：将Python机器学习算法与数据集应用到Web应用程序 / (美) 曼纽尔·阿桑斯纳吉 (Manuel Asuncion), (美) 梅赫迪·雷查格 (Mehdi Rezaghi) 著; 刘世英等译. --北京: 机械工业出版社, 2019.
(智能系统与技术丛书)

书名原文: Monetizing THE TRANSLATOR'S WORDS 2 from ML Ideas into Web Applications of the 21st Century

ISBN 978-7-111-62703-2

I. ①M... ②O... ③O... ④O... ⑤O... ⑥O... ⑦O... ⑧O... ⑨O... ⑩O...

中国版本图书馆CIP数据核字(2019)第004321号

本书定价: 49.00元

本书封面贴有图灵社区会员专享标签

译者序 2

人工智能的崛起经历了半个多世纪,从20世纪五六十年代AI概念的提出,到近些年AI热潮的此起彼伏,都与基础设施的进步和技术的革新密不可分。回顾半个多世纪,与其说AI技术在兴起,不如说AI在复兴!在信息技术领域,硬件架构经历了大型机、小型机、X86和云计算,软件架构经历了单体、多层、分布式集群、微服务和无服务器计算。历史上,没有任何一个时代计算资源的获取如此便利,应用软件的实现如此简单,而这一切,都意味着AI复兴的时代已然到来!

从技术层面来看,AI的复兴取决于两大要素:一是计算力,二是数据量。计算力越强,数据量越大,AI的发展时机就越成熟。分布式云计算技术解决了AI对计算力的需求,移动互联网让数据不断自我繁衍,大数据技术解决了海量数据的处理问题,AI复兴的道路已经铺平。相比云计算和大数据技术,AI更多表现为算法的最终呈现,或者说是机器学习在不同领域的应用,而机器学习是AI的核心和基础。提到机器学习或者算法,多数人的第一反应就是深奥难懂的数学知识和高深莫测的函数模型。必须承认,在提出AI概念后的半个多世纪,其一直难以在实际场景中得以应用的原因,除了基础设施和场景数据的匮乏外,还有就是其脱离技术领域的理论难度。译者曾经从事过计算机视觉和模式识别领域的研究工作,深知在算法理论和应用场景之间存在着巨大差距。如何把机器学习和人工智能应用到实际场景中,从而解决特定领域的问题,也是译者一直在思索的问题,因为即使我们花费大量时间来学习并理解算法背后的实际意义,但如果没有相应的实验数据,那么我们对于机器学习的研究也仅仅是停留在理论阶段。

当再次决定迈向机器学习的旅途时,译者有幸接触了Apress出版的《Monetizing Machine Learning》一书,这是一本完全从应用场景出发,讲解如何将各种机器学习算法以一种简单明了的方式应用到不同领域以解决特定问题的书籍。书中不仅有实现各种机器学习算法的代码,还提供了相应的实验数据集以供读者进行算法验证,更难能可贵的是这些代码完全可复现作者在书中的结论!与机器学习和人工智能相关的理论书籍很多,但正如我们一

直强调的，把机器学习和算法理论应用到实际场景中，进而解决特定领域问题，才是我们学习的最终目的，我相信这也是机器学习和人工智能的最终目的。

在此，我们把这本书翻译并推荐给各位读者，希望本书能为大家带来一个轻松愉快的机器学习之旅，能让大家发现机器学习原来也是件很有趣的事情！有志者，事竟成，祝各位朋友心想事成！

孝金山

译者序 3

近年来人工智能的发展正处在黄金时期，随着这个行业的持续升温，也引发了一波又一波的学习和创业热潮。但是目前市面上存在的与人工智能和机器学习相关的书籍或者在线学习课程，其内容往往侧重理论知识，实践内容则相对偏少，或者缺少场景化示例。但实际上，在读过本书原著后，我发现作者写作时非常用心，尽可能地结合理论和实践，为读者呈现出一个“平易近人、和蔼可亲”的人工智能。

本书的核心是帮助读者快速实现人工智能和机器学习工程化、产品化，简单、高效地推向市场，每章都会实现一个机器学习的创意项目（基于经典的数据科学问题），解决一个实际问题，并在开始时先进行活泼生动的项目结果展示，然后再循序渐进地指导读者完成项目的每个步骤。

本书的另一个特色是充分利用 AWS 等公有云资源，几乎每个章节都会先实现一个本地版本的 Flask 应用程序，然后再将这个应用程序部署到云端，并且尽可能选用托管解决方案。这里补充说明下，本书中提到的 Serverless 概念和目前流行的诸如 AWS Lambda 等无服务器框架有些不同，其介绍的内容更贴近 PaaS（平台即服务），但作者的主要意图是建议大家充分利用公有云的资源优势，尽量释放诸如弹性伸缩、负载均衡、持续部署等的运维工作压力，将更多的精力投入到机器学习的目标上来，这一点希望大家能够理解。

本书翻译的完成是团队协作的成果，翻译过程中团队针对每个细节进行在线会议的讨论和研究，精益求精，希望能尽量为读者呈现出原著的核心和灵魂。这里，感谢世民兄、今孝兄和力哥的支持和鼓励，他们的态度和专业性持续引领着整个团队。感谢编辑们的悉心指导，他们对本书做了大量的校对和编辑工作，保证了本书的顺利出版。最后，感谢我的家人，是他们在背后给予我默默的支持，让我可以专心地投入到本书的翻译工作中。由于本人水平有限，书中难免会出现错误或者不妥之处，欢迎批评指正，谢谢。

关于作者



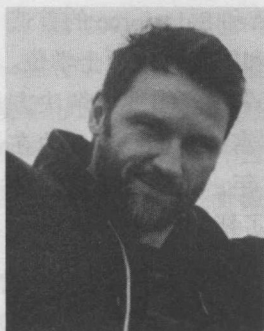
Manuel Amunategui 是 SpringML (谷歌云和 Salesforce 的首选合作伙伴) 的数据科学副总裁, 拥有预测分析和国际管理硕士学位。在过去 20 年中, 他在科技行业中实施了数百个端到端客户解决方案, 在机器学习、医疗健康建模等方面有着丰富的咨询经验。他在华尔街金融行业工作了 6 年, 在微软工作了 4 年, 这些经历使他意识到应用数据科学教育和培训材料的匮乏。为了帮助缓解这一问题, 他一直通过文字博客、视频博客和教育材料传播应用数据科学知识。他发展并策划了各种高度细分的社交媒体频道, 包括 YouTube 频道 (www.youtube.com/user/mamunate/videos) 和一个广受欢迎的应用数据科学博客 amunategui.github.io (<http://amunategui.github.io>)。



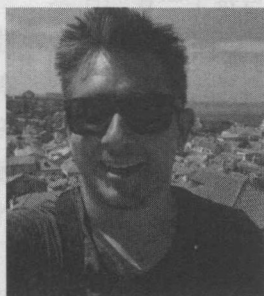
Mehdi Roopaei 是 IEEE、AIAA 和 ISA 的高级成员。2011 年获 Shiraz 大学计算机工程博士学位, 研究方向为动态系统的智能控制。2012 年至 2018 年夏季, 他在得克萨斯大学圣安东尼奥分校从事博士后研究工作; 2018 年秋季, 他在威斯康星大学普拉特维尔分校担任助理教授。他的研究兴趣包括人工智能驱动的控制系统、数据驱动决策、机器学习和物联网 (IoT), 以及沉浸式分析。他是 IEEE Access 的副主编, 也是 IoT Elsevier 杂志的编委会成员。他还是 IoT Elsevier 特刊《IoT Analytics for Data Streams》的兼职编辑, 出版了《Applied Cloud Deep Semantic Recognition: Advanced Anomaly Detection》(CRC Press, 2018) 一书。他于 2018 年 1 月至 7 月在圣安东尼奥担任 IEEE 联合通信和信号处理社区分会主席。他发表了 60 多篇备受同行好评的技术论文, 在多个会议上担任项目委员会成员, 并担任多个期刊的技术审稿人。

ABOUT THE TECHNICAL REVIEWERS

关于技术审校者



Rafal Buch 是一名生活和工作在纽约的金融系统架构师，也是一名技术专家。他有 20 余年的软件工程经历，大部分空闲时间都是在咖啡店里钻研新技术。其博客为 rafalbuch.com。



Matt Katz 自 2001 年以来一直在金融科技行业工作，他总是对新事物充满兴趣。他维护着 www.morelight.com 网站，在纽约生活，有两个可爱的孩子和一位迷人且有耐心的妻子。

PREFACE

前言

几十年前，当我还是个学编程的孩子时，我有一本用 ASCII 码编写的 Apple II 游戏书（我想不起它的名字了），每一章开头都有一幅最终的游戏成品截图。这幅截图是预告，可以引起读者兴趣，否则书就是由一页一页的计算机代码组成的了。虽然这是在 GitHub 和互联网出现之前的很多年发生的，但仿佛就在昨天。我还记得那种为了玩游戏而快速浏览代码、逐行复制它、修改拼写错误和擦眼泪的兴奋感。今天，尽管很多东西已经改变，代码也是可下载的了，但我们还是在每一章的开头放了一个最终产品的屏幕截图，给你解决问题的动力并使你感受到完成后的兴奋之情。

进入门槛降低，市场变化加快

本书将指导你完成各种各样的项目，这些项目探索不同的 Python 机器学习创意，以及将这些创意转化为 Web 应用程序的各种方法。每一章都以一个无服务器计算的 Web 应用程序结尾，全世界任何人都可以通过 Internet 连接并访问该应用程序。这些项目基于经典和流行的 Python 数据科学问题，难度逐步增加。本书研究和设计了一个建模方案，有趣的是，最后该方案会被实现为交互式 and 诱人的 Web 应用程序。

数据科学家是一个了不起的职业，但在人们试图成长为一名数据科学家时，他们的学习材料千差万别。数据科学不是关于统计和建模的，而是用于满足人类需求和解决实际问题的。

我们不会把“成为一名数据科学家”作为本书的最终目标，关于这个主题的书籍已经很多了。相反，我们将专注于把机器学习产品快速、简单地推向市场，并始终关注客户！这是职业教育大纲所缺少的。如果先构建，然后再与客户沟通，那么你的解决方案将无法满足他们的要求。我重新绘制了 Drew Conway 的数据科学维恩图，将倾听客户的声音作为首要任务（图 1）。

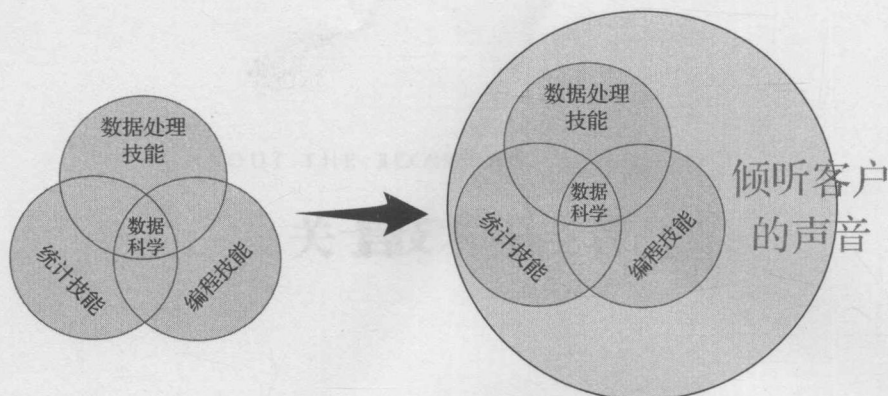


图1 经典的数据科学维恩图^①，右边是我更新的版本

我们为写作本书倾尽全力。我们花时间开发概念，确保它们对读者有实际价值（读者即我们的客户，应始终关注他们）。我撰写了初稿，然后 Mehdi 再修改它。这是一本涉猎广泛的书。工具在不断变化，当你阅读本书时，一些内容可能过时，但这永远不是问题（你可以去 GitHub 仓库^②查看更新）。毕竟，一切都在变化，但往往都是变得更好！因此，学习新技巧通常意味着学到更好、更快、更强大的方法来做事。本书不仅向你展示如何构建 Web 应用程序，还将为你指明正确的方向，以便你更深入地学习感兴趣领域的知识。

建议你通读本书，理解工具，然后复制它们，把它们变成你自己的。这些模板可以帮助你快速启动其他类似项目，为你的客户构建有效的工具。

什么是无服务器云计算

云提供商一直在努力改进 Web 托管解决方案并降低成本。最近出现的“无服务器计算”（serverless）服务抽象了大部分的配置过程，本书提到的四大云提供商中有三个提供这种服务。这意味着你可以在完全托管的平台上启动和运行项目，实现自动负载均衡、根据吞吐量弹性伸缩、快速部署等，而无须选择、配置和担心任何事情。你可自行决定架构上的定制程度和监控项。你可选择哪些自己控制，哪些由提供商控制。有一件事是可以保证的：网站将自动调整流量，并提供无与伦比的正常运行时间。

这使我们能专注于更重要的事情，而不会陷入实现目标时可能的陷阱之中。这些所谓的“陷阱”至关重要，需要认真对待。这就是我们找了这四家信誉良好的云提供商的原因，它们将让我们完全专注于 Web 应用程序，而不用担心网站崩溃或提供商倒闭。让我们专注于重要的事情，开始工作吧！

① <http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>

② <https://github.com/Apress/monetizing-machine-learning>

Web 应用开发中的关键路径

如此多的机器学习模型在其最初编码阶段就让人停滞不前：这些模型很难理解，没有简单方法让人从其洞察中受益。这些模型注定要被遗忘，即使是那些设法摆脱集成开发接口限制的开发人员，在简化静态图表或为建模评分时，也无法发挥其潜力。本书旨在将扩展机器学习模型到通用的 Web 应用程序这一过程进行分解来避免这种陷阱。每一章都遵循以下三个关键步骤：

1. 以正确的方式建模。我们从最后的成果开始，首先理解用户想要看到什么，并将时间和精力投入到最终目标和用户体验上。我们使用适当的建模方法，以便快速且毫不意外地实现 Web 应用程序的目标（图 2）。



图 2 在构建之前，始终检查你的想法是否有受众（来源：Lucas Amunategui）

2. 设计和开发本地 Web 应用程序。此步骤需利用各种 Web 前端技术来实现所需的交互性和动态性，以突出模型的洞察力，从而让用户感兴趣。此阶段的最终产品与下一阶段的产品是完全一样的，只是它托管在你的本地计算机上，而不是云上。
3. 部署到流行且可靠的无服务器计算云平台上。每个云提供商都有独特的要求、优点和缺点，需要充分理解这些问题。这是全世界享受和学习你的工作成果的最后一个环节。

我们从提供智能交互性的简单方法入手（比如调整模型的系数或利用保存好的训练过的模型），然后转向复杂方法（比如使用数据库跟踪参与性，或依赖开放源码的训练过的模型进行图像识别）。案例研究 1 是一个关于股票市场预测的虚构案例，它在后面的章节中会被反复提及，并添加新功能，直到它最终成为一个带有付费使用门槛的复杂仪表盘，为付费用

户提供可定制的智能化功能。

通过关注经典的数据科学问题，再加上流行的开源技术（如 Python、Flask、Ajax、Bootstrap、JavaScript 和 HTML），你会发现你已经了解了一些基础知识；即使你还不了解，你的学习曲线也会大大缩短。我们专注于使用简单的工具和简单的技术，去可靠、快速地将机器学习创意付诸实践。工具和方法在每一章都会被重新提及，所以如果你对某些东西一开始还不太清楚，也不用担心，只要继续下去，事情就会越来越清晰。

我们还在每章中轮流提到云提供商，因此你将接触到最受欢迎的云提供商。这将为“给未来项目选择提供商”提供丰富经验。建议仔细阅读所有章节，因为每个章节都会展示不同的项目及其实现方式，在强调提供商优势的同时，也展示快速完成工作的独特技巧。

致读者

本书是为那些有兴趣将统计模型、机器学习流水线、数据驱动项目或任何独立 Python 脚本扩展到任何人都可以通过 Web 浏览器访问的交互式仪表板的人编写的。互联网是最强大的媒体，进入门槛极低，任何人都可以访问它，本书是为那些想要利用它的人准备的。

本书假设你有 Python 编程经验，你需有一个代码编辑器和解释器来实现你的创意。你应该具有测试、排除故障、安装 Python 库以及熟悉流行的包（如 NumPy、Pandas 和 Scikit-learn）的能力。本书没有介绍这些基本知识。这里给出的代码仅适用于 Python 3.x，尚未对较低版本进行测试。此外，你还需要有关于 Web 脚本语言的基本知识。

本书的目标读者是：想把创意发布到网上的有创业精神的人；没有 IT 团队的小公司；想接触和接受真实世界培训的学生；数据科学专业人士。

如何使用本书

每章都以最终实现的 Web 应用程序的截图开始，并描述了它将做什么。这种方法有多个目的：

- 它可以是吸引你投入工作的动力。
- 它直观地解释了项目的内容。
- 更重要的是，它告诉我们，在处理项目时对最终结果有明确的以客户为中心的理解是多么重要。

书中只显示源代码的重点部分，完整版本放在相应的存储库中。当涉及数据探索和 Web 应用程序的压缩文件夹时会用到 Jupyter Notebook。

本书介绍的实际项目简单明了，可以用作模板来启动许多其他类型的应用程序。无论是围绕数字性或分类性预测创建 Web 应用程序，分析文本，创建强大的交互式演示文稿，提供对受限数据的访问，还是利用 Web 插件接受订阅付款和捐赠，本书都将帮助你快速实现

项目并将其交付给世界各地的人们。

提示 有关编辑 / 错误的信息, 请登录 www.apress.com/9781484238721 并提交报告。

交互工具

下面简要介绍将机器学习创意快速、简单和漂亮地转换为 Web 应用程序的工具。这并不是一个完整列表, 如果你想了解更多 (我希望你愿意), 可以尝试使用的不同技术, 并且需要遵循我们的提示。

Jupyter Notebook

本书只显示了代码片段, 因此每一章中你都需要下载并运行 Jupyter Notebook, 以便自己尝试各种特性和不同的参数。Jupyter Notebook 是基于 Web 的交互式 Python 解释器, 非常适合构建、调整和发布任何使用 Python 脚本的东西。它被附加到一个功能完整的 Python 内核 (将其设置为 Python 3.x), 并且可以像其他解释器一样加载和运行库及脚本。要安装 Jupyter Notebook, 请参考 <http://jupyter.readthedocs.io/en/latest/install.html> 上的官方文档。

安装方式多种多样, 包括使用 “pip3” 命令, 如果这种方法不适合你, 则请查看官方文档, 了解不同的方法 (代码清单 1)。

代码清单1 安装Jupyter

```
sudo pip3 install jupyter
```

Jupyter Notebook 使用起来既简单又强大。你只需将它下载到本地计算机 (它是带有 *.ipynb 扩展名的文件), 打开命令 / 终端 Shell 窗口, 导航到该文件夹, 然后运行 “notebook” 命令 (代码清单 2)。

代码清单2 运行Notebook (查看官方文档, 了解启动Notebook的其他方法)

```
jupyter notebook
```

此命令将打开一个网页, 显示它从中启动的文件夹的内容 (图 3)。你可以通过单击文件列表正上方的文件夹图标向下导航文件夹结构。

要打开 Jupyter Notebook, 只需单击任何带有 “*.ipynb” 扩展名的文件。如果要创建全新 Notebook, 请单击紧接刷新按钮的仪表板右侧的 “New” 按钮。

提示 有关其他信息、Jupyter Notebook 问题以及附加内核的信息, 请参阅 <http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html>。

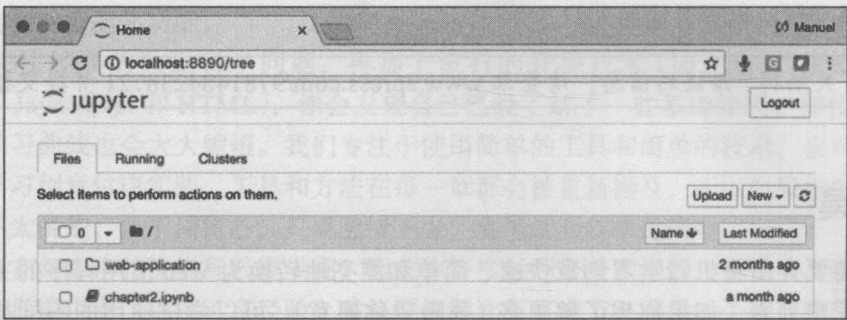


图 3 Jupyter Notebook 登录页面

Flask

Flask 是一个轻量级但非常强大的服务器端 Web 框架。它是本书中所有应用程序背后的“大脑”，也是 Python 数据生成函数和网页之间的黏合剂。我喜欢使用 Flask 的原因之一是，它允许我们在不离开 Python 语言的情况下将独立的 Python 脚本链接到服务器端 Web 框架，使得在对象之间传递数据更加容易！

Flask 附带了发布网页的最低要求。如果你需要其他支持，例如数据库、表单控件等，则必须安装其他库，这就是它被称为轻量级微框架的原因。这也是它易于使用的原因，因为你只需学习一些技巧，其他一切都可使用熟悉的经过验证的 Python 库。

遗憾的是，我们只能使用 Python 至此，最终你需要进入前端 Web 脚本。但是不要有困扰——互联网上有很多很棒的例子（Stackoverflow.com、w3schools.com）和令人难以置信的 GetBootstrap.com 模板，可以让你尽快到达目的地。

提示 有关 Flask 的更多信息，请参阅 <http://flask.pocoo.org/> 上的官方 Flask 文档。

HTML

HTML（超文本标记语言）是网络技术方面最基本的东西之一。它已存在多年，用于创建几乎所有的网页和 Web 应用程序。

对于那些想要了解这一主题的人来说，网上的免费资料浩如烟海。要了解 HTML，推荐 w3schools.com 网站，这个网站的学习材料组织良好、全面，而且通常是交互式的。

CSS

CSS（Cascading Style Sheet，层叠样式表）使大多数网站看起来很棒！我们在这里使用两种类型的 CSS 文件：大多数网页（最常见）的“<HEAD>”部分中加载的 CSS 链接和代码清单 3 中显示的自定义 CSS。

代码清单3 自定义CSS脚本块

```
<STYLE>
.btn-circle.btn-xl {
    width: 70px;
    height: 70px;
    padding: 10px 2px;
    border-radius: 35px;
    font-size: 17px;
    line-height: 1.33;
}
</STYLE>
```

托管在外部服务器上的 CSS 文件无法自定义，但通常是同类最佳的。有时你只需要在页面上自定义功能，即在 HTML 页面中直接创建本地 CSS 文件或样式标签，然后使用“class”参数将其应用于特定标签或区域（代码清单 4）。

代码清单4 将CSS标签应用于HTML标签

```
<button type="button" onclick="calculateBikeDemand(this)"
id="season_spring" class="btn btn-info btn-circle btn-xl


---



```

CSS 非常详细地定义了所有的尺寸、颜色、字体。它还允许你创建所见即所感的 Web 门户。只需创建一次，即可让所有页面调用它来继承该特定样式。

提示 有关 CSS 的其他信息，请访问 w3schools.com。

Jinja2

Jinja2 用于生成标记和 HTML 代码，并与 Flask 变量紧密配合。它由 Armin Ronacher 创建，广泛用于处理 Flask 生成的数据以及直接在 HTML 模板中的 if/then 逻辑。

在此 HTML 模板示例中，使用 Jinja2 将名为“previous_slider_value”的 Flask 生成的值注入幻灯片的“value”参数。注意使用双花括号（代码清单 5）。

代码清单5 Jinja2将数据传递给HTML输入控件

```
<input type="range" min="1" max="100" value="{{previous_slider_value}}"
id="my_slider">
```

提示 有关 Jinja2 的其他信息，请参阅 <http://jinja.pocoo.org/docs/2.10/>。

JavaScript

JavaScript 本身就是一种真正的编程语言，它可以为你的任何前端控件添加极其强大的

行为。JavaScript 为网页带来了很高的交互性，本书中将大量使用它。

这是一个有趣的示例，我们捕获 HTML 滑块控件的鼠标按钮松开（mouse-up）事件，以将表单提交到 Flask 服务器。这个想法是，每当用户更改滑块值时，Flask 需要使用新的滑块值进行一些服务器端处理并重新生成网页（代码清单 6）。

代码清单6 JavaScript捕获滑块onmouseup事件

```
slider1.onmouseup = function ()
{
    document.getElementById("submit_params").submit();
}
```

提示 有关 JavaScript 的其他信息，请访问 w3schools.com。

jQuery

jQuery 是一个定制的 JavaScript 库，可以帮助处理复杂的前端和行为事件，并确保不同浏览器版本之间的兼容性。

jQuery 帮助优化按钮、下拉动态行为，甚至 Ajax 交互（在本书的许多项目中大量使用的关键技术）。

提示 有关 jQuery 的更多信息，请查看 jquery.com 上的官方文档。

Ajax

Ajax 是一种出色的前端脚本技术，可以为网页添加动态服务器端行为。它允许发送和接收数据，而无须像表单提交那样重建或重新加载整个页面。一个常用领域是地图网页，例如 Google 地图，它允许拖动和滑动地图，而无须在每次移动后重新加载整个页面。

提示 有关 Ajax 的其他信息，请访问 w3schools.com。

Bootstrap

Bootstrap 是一个非常强大、近乎神奇的前端 Web 工具。根据 BuiltWith Trends 的说法^①，它几乎占据了 Web 的 13%。它包含大多数 Web 标签和控件的各种外观及行为。通过简单地将你的网页链接到最新的 Bootstrap，CSS 将为任何无聊的 HTML 页面提供即时和专业的改造！

如果你查看本书中的任何 HTML 文件，那么首先要注意的是页面顶部的 LINK 和 SCRIPT 标签中包含的链接。这是构建网页的最佳捷径（代码清单 7）。

① <https://trends.builtwith.com/docinfo/Twitter-Bootstrap>