Broadview
www.broadview.com.cn

简单的Python，可以完成复杂的机器学习算法，跟我来吧！

Python与机器学习实战

决策树、集成学习、支持向量机与神经网络算法详解及编程实现

何宇健 编著

很丰富： 7种算法，50段实现，55个实例，总代码量5295行，全面而不冗余。

很扎实： 对经典有效的机器学习算法的核心内容进行了详细推导。

很应用： 将理论实打实地用Python代码写出来，可以完成一定的任务。

很前沿： 叙述了Inception-v3 from Google、迁移学习等前沿技术。

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Python与机器学习实战

决策树、集成学习、支持向量机与神经网络算法详解及编程实现

何宇健 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Python 与机器学习这一话题是如此的宽广，仅靠一本书自然不可能涵盖到方方面面，甚至即使出一个系列的书也难能做到这点。单就机器学习而言，其领域就包括但不限于如下：有监督学习（Supervised Learning），无监督学习（Unsupervised Learning）和半监督学习（Semi-Supervised Learning）。而其具体的问题又大致可以分为两类：分类问题（Classification）和回归问题（Regression）。

Python 本身带有许多机器学习的第三方库，但本书在绝大多数情况下只会用到 Numpy 这个基础的科学计算库来进行算法代码的实现。这样做的目的是希望读者能够从实现的过程中更好地理解机器学习算法的细节，以及了解 Numpy 的各种应用。不过作为补充，本书会在适当的时候应用 scikit-learn 这个成熟的第三方库中的模型。

本书适用于想了解传统机器学习算法的学生和从业者，想知道如何高效实现机器学习算法的程序员，以及想了解机器学习算法能如何进行应用的职员、经理等。

图书在版编目（CIP）数据

Python 与机器学习实战：决策树、集成学习、支持向量机与神经网络算法详解及编程实现 / 何宇健编著.—北京：电子工业出版社，2017.7

ISBN 978-7-121-31720-0

Ⅰ. ①P… Ⅱ. ①何… Ⅲ. ①软件工具－程序设计②机器学习 Ⅳ. ①TP311.561②TP181

中国版本图书馆 CIP 数据核字（2017）第 121087 号

策划编辑：张月萍
责任编辑：徐津平
特约编辑：顾慧芳
印　　刷：北京京科印刷有限公司
装　　订：北京京科印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　　印张：20.5　　字数：381 千字
版　　次：2017 年 7 月第 1 版
印　　次：2017 年 7 月第 1 次印刷
印　　数：3000 册　　定价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819　faq@phei.com.cn。

前　言

自从 AlphaGo 在 2016 年 3 月战胜人类围棋顶尖高手李世石后，“人工智能”“深度学习”这一类词汇就进入了大众的视野；而作为更加宽泛的一个概念——“机器学习”则多少顺势成为了从学术界到工业界都相当火热的话题。不少人可能都想尝试和体验一下“机器学习”这个可以说是相当神奇的东西，不过可能又苦于不知如何下手。编著本书的目的，就是想介绍一种入门机器学习的方法。虽然市面上已经有许多机器学习的书籍，但它们大多要么过于偏重理论，要么过于偏重应用，要么过于“厚重”；本书致力于将理论与实践相结合，在讲述理论的同时，利用 Python 这一门简明有力的编程语言进行一系列的实践与应用。

当然，囿于作者水平，本书实现的一些模型从速度上来说会比成熟的第三方库中实现的模型要慢不少。一方面是因为比较好的第三方库背后往往会用底层语言来实现核心算法，另一方面则是本书通常会把数据预处理的过程涵盖在模型中。以决策树模型为例，scikit-learn 中的决策树模型会比本书的实现要快很多，但本书实现的模型能够用 scikit-learn 中决策树模型训练不了的训练集来训练。

同时，限于篇幅，本书无法将所有代码都悉数放出（事实上这样做的意义也不是很大），所以我们会略去一些相对枯燥且和相应算法的核心思想关系不大的实现。对于这些实现，我们会进行相应的算法说明，感兴趣的读者可以尝试自己一步一步地去实现，也可以直接在 GitHub 上面查看笔者自己实现的版本（GitHub 地址会在相应的地方贴出）。本书所涉及的所有代码都可以参见 https://github.com/carefree0910/MachineLearning，笔者也建议在阅读本书之前先把这个链接里面的内容都下载下来作为参照。毕竟即使在本书收官之后，笔者

仍然会不时地在上述链接中优化和更新相应的算法，而这些更新是无法反映在本书中的。

虽说确实可以完全罔顾理论来用机器学习解决许多问题，但是如果想要理解背后的道理并借此提高解决问题的效率，扎实的理论根基是必不可少的。本书会尽量避免罗列枯燥的数学公式，但是基本的公式常常不可或缺。虽然笔者想要尽量做到通俗易懂，但仍然还是需要读者拥有一定的数学知识。不过为了阅读体验良好，本书通常会将比较烦琐的数学理论及相关推导放在每一章的倒数第二节（最后一节是总结）作为某种意义上的“附加内容”。这样做有若干好处：

- 对于已经熟知相关理论的读者，可以不再重复地看同样的东西；
- 对于只想了解机器学习各种思想、算法和实现的读者，可以避免接受不必要的知识；
- 对于想了解机器学习背后道理和逻辑的读者，可以有一个集中的地方进行学习。

本书的特点

- 理论与实践结合，在较为详细、全面地讲解理论之后，会配上相应的代码实现以加深读者对相应算法的理解。
- 每一章都会有丰富的实例，让读者能够将本书所阐述的思想和模型应用到实际任务中。
- 在涵盖了诸多经典的机器学习算法的同时，也涵盖了许多最新的研究成果（比如最后一章所讲述的卷积神经网络（CNN）可以说就是许多“深度学习”的基础）。
- 所涉及的模型实现大多仅仅基于线性代数运算库（Numpy）而没有依赖更高级的第三方库，读者无须了解 Python 那浩如烟海的第三方库中的任何一个便能读懂本书的代码。

本书的内容安排

第 1 章　Python 与机器学习入门

本章介绍了机器学习的概念和一些基础术语，比如泛化能力、过拟合、经验风险（ERM）和结构风险（SRM）等，还介绍了如何安装并使用 Anaconda 这一 Python 的科学运算环境。同时在最后，我们解决了一个小型的机器学习问题。本章内容虽不算多，却可说是本书所有内容的根基。

第 2 章　贝叶斯分类器

作为和我们比较熟悉的频率学派相异的学派，贝叶斯学派的思想相当耐人寻味，值得进行研究与体会。本章将主要介绍的朴素贝叶斯正是贝叶斯决策的一个经典应用，虽然它加了很强的假设，但其在实际应用中的表现仍然相当优异（比如自然语言处理中的文本分类）。而为了克服朴素贝叶斯假设过强的缺点，本章将简要介绍的，诸如半朴素贝叶斯和贝叶斯网这些贝叶斯分类器会在某些领域拥有更好的性能。

第 3 章　决策树

决策树可以说是最直观的机器学习模型之一，它多多少少拥有信息论的一些理论背景作为支撑。决策树的训练思想简洁，模型本身可解读性强，本章将会在介绍其生成、剪枝等一系列实现的同时，通过一些可视化来对其有更好的理解。

第 4 章　集成学习

真所谓“三个臭皮匠，赛过诸葛亮”。集成学习的两大板块“Bootstrap”和“Boosting”所对应的主流模型——“随机森林（Random Forest）”和“AdaBoost”正是这句俗语的最佳解释。本章在介绍相关理论与实现的同时，将通过相当多的例子来剖析集成学习的一些性质。

第 5 章　支持向量机

支持向量机（SVM）有着非常辉煌的历史，它背后那套相当深刻而成熟的数学理论让它在现代的深度学习“异军突起”之前，占据着相当重要的地位。本章会尽量厘清支持向量机的思想与相关的比较简明的理论，同时会通过一些对比来体现支持向量机的优越之处。

第 6 章　神经网络

神经网络在近现代可以说已经成为“耳熟能详”的词汇了，它让不少初次听说其名号的人（包括笔者在内）对其充满着各种幻想。虽说神经网络算法的推导看上去繁复而“令人生畏”，但其实所用到的知识并不深奥。本章会相当详细地介绍神经网络中的两大算法——“前向传导算法”和“反向传播算法”，同时还会介绍诸多主流的“参数更新方法”。除此之外，本章还会提及如何在“大数据”下改进和优化我们的神经网络模型（这一套思想是可以推广到其他机器学习模型上的）。

第 7 章　卷积神经网络

卷积神经网络是许多深度学习的基础结构，它可以算是神经网络的一种拓展。卷积神经网络的思想具有很好的生物学直观性，适合处理结构性的数据。同时，利用成熟的卷积

神经网络模型，我们能够比较好地完成许多具有一定难度而相当有趣的任务；本章则会针对这些任务中的“图像分类”任务，提出一套比较详细的解决方案。

本书由浅入深，理论与实践并存，同时将理论也进行了合理的分级；无论读者在此前对机器学习有何种程度的认知，想必都能通过不同的阅读方式有所收获吧。

适合阅读本书的读者

- 想要了解某些传统机器学习算法细节的学生、老师、从业者等。
- 想要知道如何“从零开始”高效实现机器学习算法的程序员。
- 想要了解机器学习算法能如何进行应用的职员、经理等。
- 对机器学习抱有兴趣并想要入门的爱好者。

编者　何宇健

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误**：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：*http://www.broadview.com.cn/31720*

三步加入“人工智能交流群”，实时获取资源共享

1. 扫码添加小编为微信好友。
2. 申请验证时输入“AI”。
3. 小编带你加入“人工智能交流群”。

目　录

第 1 章

Python 与机器学习入门

“机器学习”在最近虽可能不至于到人尽皆知的程度，却也是非常火热的词汇。机器学习是英文单词“Machine Learning”（简称 ML）的直译，从字面上便说明了这门技术是让机器进行“学习”的技术。然而我们知道机器终究是死的，所谓的“学习”归根结底亦只是人类“赋予”机器的一系列运算。这个“赋予”的过程可以有很多种实现，而 Python 正是其中相对容易上手、同时性能又相当不错的一门语言。作为第 1 章，我打算先谈谈机器学习相关的一些比较宽泛的知识，再介绍并说明为何要使用 Python 来作为机器学习的工具。最后，我们会提供一个简短易懂的、具有实际意义的例子来给大家提供一个直观的感受。

具体而言，本章主要涉及的知识点有：

- 机器学习的定义及重要性；
- Python 在机器学习领域的优异性；
- 如何在电脑上配置 Python 机器学习的环境；
- 机器学习一般性的步骤。

1.1 机器学习绪论

正如前言所说，由于近期的各种最新成果，使得“机器学习”成为了非常热门的词汇。

机器学习在各种领域的优异表现（围棋界的 Master 是其中最具代表性的存在），使得各行各业的人们都或多或少地对机器学习产生了兴趣与敬畏。然而与此同时，对机器学习有所误解的群体也日益壮大；他们或将机器学习想得过于神秘，或将它想得过于万能。本节拟对机器学习进行一般性的介绍，同时会说明机器学习中一些常见的术语以方便之后章节的叙述。

1.1.1 什么是机器学习

清晨的一句“今天天气真好”、朋友之间的寒暄“你刚刚是去吃饭了吧”、考试过后的感叹“复习了那么久终有收获”……这些日常生活中随处可见的话语，其背后却已蕴含了“学习”的思想——它们都是利用以往的经验、对未知的新情况作出的有效的决策。而把这个决策的过程交给计算机来做，可以说就是“机器学习”的一个最浅白的定义。

我们或许可以先说说机器学习与以往的计算机工作样式有什么不同。传统的计算机如果想要得到某个结果，需要人类赋予它一串实打实的指令，然后计算机就根据这串指令一步步地执行下去。这个过程中的因果关系非常明确，只要人类的理解不出偏差，运行结果是可以准确预测的。但是在机器学习中，这一传统样式被打破了：计算机确实仍然需要人类赋予它一串指令，但这串指令往往不能直接得到结果；相反，它是一串赋予了机器“学习能力”的指令。在此基础上，计算机需要进一步地接受“数据”，并根据之前人类赋予它的“学习能力”，从中“学习”出最终的结果。这个结果往往是无法仅仅通过直接编程得出的。因此这里就导出了稍微深一点的机器学习的定义：它是一种让计算机利用数据而非指令来进行各种工作的方法。在这背后，最关键的就是“统计”的思想，它所推崇的“相关而非因果”的概念是机器学习的理论根基。在此基础上，机器学习可以说是计算机使用输入给它的数据，利用人类赋予它的算法得到某种模型的过程，其最终的目的则是使用该模型，预测未来未知数据的信息。

既然提到了统计，那么一定的数学理论就不可或缺。相关的、比较简短的定义会在第 4 章给出（PAC 框架），这里我们就先只叙述机器学习在统计理论下的、比较深刻的本质：它追求的是合理的假设空间（Hypothesis Space）的选取和模型的泛化（Generalization）能力。该句中出现了一些专用术语，详细的定义会在介绍术语时提及，这里我们提供一个直观的理解：

- 所谓的假设空间，就是我们的模型在数学上的“适用场合”。
- 所谓的泛化能力，就是我们的模型在未知数据上的表现。

注意：上述本质上严格来说，应该是 PAC Learning 的本质；在其余的理论框架下，机器学习是可以具有不同的内核的。

从上面的讨论可以看出，机器学习和人类思考的过程有或多或少的类似。事实上，我们在第 6、第 7 章讲的神经网络（Neural Network，NN）和卷积神经网络（Convolutional Neural Network，CNN）背后确实有着相应的神经科学的理论背景。然而与此同时需要知道的是，机器学习并非是一个“会学习的机器人”和“具有学习能力的人造人”之类的，这一点从上面诸多讨论也可以明晰（惭愧的是，笔者在第一次听到“机器学习”四个字时，脑海中浮现的正是一个“聪明的机器人”的图像，甚至还幻想过它和人类一起生活的场景）。相反的，它是被人类利用的、用于发掘数据背后信息的工具。

当然，现在也不乏“危险的人工智能”的说法，霍金大概是其中的“标杆”，这位伟大的英国理论物理学家甚至警告说“人工智能的发展可能意味着人类的灭亡”。孰好孰坏果然还是见仁见智，但可以肯定的是：本书所介绍的内容绝不至于导致世界的毁灭，大家大可轻松愉快地进行接下来的阅读！

1.1.2　机器学习常用术语

机器学习领域有着许多非常基本的术语，这些术语在外人听来可能相当高深莫测。它们事实上也可能拥有非常复杂的数学背景，但需要知道：它们往往也拥有着相对浅显平凡的直观理解（上一小节的假设空间和泛化能力就是两个例子）。本小节会对这些常用的基本术语进行说明与解释，它们背后的数学理论会有所阐述，但不会涉及过于本质的东西。

正如前文反复强调的，数据在机器学习中发挥着不可或缺的作用；而用于描述数据的术语有好几个，需要被牢牢记住的如下。

- “数据集”（Data Set），就是数据的集合的意思。其中，每一条单独的数据被称为“样本”（Sample）。若没有进行特殊说明，本书都会假设数据集中样本之间在各种意义下相互独立。事实上，除了某些特殊的模型（如隐马尔可夫模型和条件随机场），该假设在大多数场景下都是相当合理的。
- 对于每个样本，它通常具有一些“属性”（Attribute）或者说“特征”（Feature），特征所具体取的值就被称为“特征值”（Feature Value）。
- 特征和样本所张成的空间被称为“特征空间”（Feature Space）和“样本空间”（Sample Space），可以把它们简单地理解为特征和样本“可能存在的空间”。
- 相对应的，我们有“标签空间”（Label Space），它描述了模型的输出“可能存在的空间”；当模型是分类器时，我们通常会称之为“类别空间”。

其中、数据集又可以分为以下三类。

- 训练集（Training Set）：顾名思义，它是总的数据集中用来训练我们模型的部分。虽说将所有数据集都拿来当作训练集也无不可，不过为了提高及合理评估模型的泛化能力，我们通常只会取数据集中的一部分来当训练集。
- 测试集（Test Set）：顾名思义，它是用来测试、评估模型泛化能力的部分。测试集不会用在模型的训练部分，换句话说，测试集相对于模型而言是“未知”的，所以拿它来评估模型的泛化能力是相当合理的。
- 交叉验证集（Cross-Validation Set，CV Set）：这是比较特殊的一部分数据，它是用来调整模型具体参数的。

注意：需要指出的是，获取数据集这个过程是不平凡的；尤其是当今“大数据”如日中天的情景下，诸如“得数据者得天下”的说法也不算谚语。在此笔者推荐一个非常著名的含有大量真实数据集的网站：http://archive.ics.uci.edu/ml/datasets.html，本书常常会用到其中一些合适的数据集来评估我们自己实现的模型。

可以通过具体的例子来理解上述概念。比如，我们假设小明是一个在北京读了一年书的学生，某天他想通过宿舍窗外的风景（能见度、温度、湿度、路人戴口罩的情况等）来判断当天的雾霾情况并据此决定是否戴口罩。此时，他过去一年的经验就是他拥有的数据集，过去一年中每一天的情况就是一个样本。“能见度”、“温度”、“湿度”、“路人戴口罩的情况”就是四个特征，而（能见度）“低”、（温度）“低”、（湿度）“高”、（路人戴口罩的）“多”就是相对应的特征值。现在小明想了想，决定在脑中建立一个模型来帮自己做决策，该模型将利用过去一年的数据集来对如今的情况做出“是否戴口罩”的决策。此时小明可以用过去一年中 8 个月的数据量来做训练集、2 个月的量来做测试集、2 个月的量来做交叉验证集，那么小明就需要不断地思考（训练模型）下列问题：

- 用训练集训练出的模型是怎样的？
- 该模型在交叉验证集上的表现怎么样？
    - 如果足够好了，那么思考结束（得到最终模型）。
    - 如果不够好，那么根据模型在交叉验证集上的表现，重新思考（调整模型参数）。

最后，小明可能会在测试集上评估自己刚刚思考后得到的模型的性能，然后根据这个性能和模型做出的“是否戴口罩”的决策来综合考虑自己到底戴不戴口罩。