

doi:10.13582/j.cnki.1674-5884.2020.03.010

# 应用驱动的渐进式 Python 案例 教学探索与实践

文一凭<sup>a</sup>,张进良<sup>b</sup>,姜磊<sup>a</sup>,朱建军<sup>a</sup>,王湘群<sup>a</sup>,胡蓉<sup>a</sup>

(湖南科技大学 a.计算机科学与工程学院; b.教育学院, 湖南 湘潭 411201)

**摘要:**随着大数据与人工智能等技术的飞速发展,大学生学习信息技术的需求日益迫切。Python 语言为大学生解决实际问题提供了良好的工具。本研究立足大学 Python 教学,从应用与实践的视角,以大学生的专业特点与需求为导向,探索应用驱动的渐进式 Python 案例教学方法,并以向商学院学生讲授图书销售应用为例,给出了基础实例与进阶实例,以期对 Python 案例教学实践提供一定的借鉴。

**关键词:**Python;案例教学;程序设计;应用

**中图分类号:**G642.4

**文献标志码:**A

**文章编号:**1674-5884(2020)03-0051-06

为适应新时期信息技术的发展,教育部考试中心于2017年对全国计算机等级考试(National Computer Rank Examination, NCRE)体系进行调整,并决定从2018年3月起执行2018版NCRE考试大纲。这次调整取消了“Visual FoxPro 数据库程序设计”科目,新增了“Python 语言程序设计”科目。在此背景下,国内高校纷纷开设“Python 语言程序设计”课程。

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言,并具有开发效率高、足够灵活和易上手等优点,已广泛应用于人工智能、数据分析、机器学习等领域。Python 自身免费开源与跨平台执行的特性吸引着越来越多的人参与到 Python 程序库的开发与应用中,这些程序库大多都允许个人免费使用甚至商用,而且功能更为丰富完备的程序库正在不断增长,这为利用 Python 语言解决实际问题创建了良好的环境。目前,pypi.org 网站上已提供 18 万个以上的 Python 项目,从常规的各类软件、Web 应用与游戏开发,到高级的科学计算、云计算、操作系统管

理与服务器的自动化运维,均可以利用 Python 解决其相关应用问题。因此,在“Python 语言程序设计”课程教学中,可充分利用 Python 语言的优势与其应用进行教学案例设计。

但是,如何根据学生的专业背景、学习需求与计算机基础,设计合适的 Python 教学案例并不是一件容易的事情。文献[1]提供了一种以图形牵引兴趣的 Python 案例教学方法,并给出了入门案例以及配合各教学内容的开展所设置的基于 turtle 库的相关图形案例。文献[2]分析了人文社科类专业学生的学习特点与解决其专业领域问题的一般步骤,并利用电影数据与 Requests 和 BS4 等库设计了面向人文社科类专业的 Python 教学案例,该案例涉及电影数据爬取、电影信息处理、电影相关信息统计、电影主题分析和演员合作网络分析。文献[3]提出了“面向问题、引发兴趣、培养能力、引导创新”的 Python 语言程序设计教学案例新思维,指出了可围绕有特定用途的功能库(例如 random 库、jieba 库、turtle 库等)来设计入门案例,并给出了基于 turtle 库的图形绘制

收稿日期:20190603

基金项目:湖南科技大学教学改革研究项目(G31811、G31909);2018年湖南省社科基金项目(18YBA173);湖南省教育厅教学改革项目(湘教通[2018]436号文件,序号364;湘教通[2019]291号文件,序号474)

作者简介:文一凭(1981-),男,湖南祁阳人,副教授,博士,主要从事业务过程管理、数据挖掘、云计算与分布式处理方面的研究。

案例以及基于经典文学作品数据的人物出场统计案例。文献[4]针对工程教育专业认证标准中的要求,设计了一些与材料专业相关的综合性教学案例(例如X射线衍射图谱绘制),并尝试向学生介绍材料科学领域相关的功能库,引导学生开发工具包。文献[5]在Python案例教学中引入了机器学习方法,并围绕与机器学习相关的numpy、pandas和sklearn库设计了基于K-means与DBSCAN这两个聚类算法的应用案例。这些优秀的案例都能有效激发学生的学习兴趣,但是,从实际的Python教学内容与教学过程看,基于函数库设计的入门案例尽管并不复杂,但仍需要学生在掌握一定的Python基础后才能讲授,然后才能引导学生创新实践。

笔者近两年一直从事高校非计算机专业的Python教学与研究,为解决上述案例设计存在的问题,通过多轮思考和实践,摸索形成了一种应用驱动的渐进式Python案例教学方法。该方法针对不同专业学生需要设计一些难度合适、贯穿Python基础学习过程的应用案例,应用案例可在第一节Python课时便开始介绍一部分,而不要求学生掌握一定的Python基础,而且后续可配合该案例安排渐进式展开教学内容,以达到持续激发学习兴趣、培养学生应用能力并增强学习效果的目的。

## 1 应用驱动的渐进式Python案例设计方法

### 1.1 设计的目标与特点

Python语言功能强大,各种社区提供的Python函数模块(即第三方函数库)在不断增长,基于Python的各种应用与技术也在飞速发展。这些都是学习Python语言的优势,同时也给Python的教学工作提供了广阔的内容选择空间。文献[6]分析了Python教学时学生群体的不同特征与培养模式差异,并从计算机基础教学培养目标的角度,指出对人文社科类学生的Python学习需要注重培养其解决基础问题的能力,对理工类学生则可注重高级问题,而商学类学生可结合其学科专业的问题进行定制培养。这几种培养模式差异,直接体现在教学内容的难易程度上,以尽可能贴近不同学生群体应用Python解决应用问题的特点并培养其在各自领域解决问题的能力。例如在讲授“网络数据获取”这一教学内容时,对人

文社科类学生不要求其掌握开发者工具,而对理工类学生则要求掌握开发者工具及基础的网络爬虫框架。文献[7]分析了Fortran语言、C语言与Python语言各自的诞生背景与其擅长的应用领域差异,将Python教学分为计算机基础课(即作为非信息类专业学生的入门程序设计课)与专业应用课程(即学生已学习了某种程序设计语言)这两个层次,并指出如作为计算机基础课,Python的教学重点是Python基础,并适当介绍一些Python函数模块以引导学习方向、进行应用示范。这些研究成果为Python案例教学设计指明了方向。

基于上述Python语言教学研究成果,笔者在Python教学中,根据Python擅长解决大数据与人工智能应用与技术问题的优势,尝试以应用驱动的方式,根据学生的学科专业特点,渐进式设计教学案例,并配合这些案例安排Python基础教学内容。

### 1.2 Python教学内容设计

目前国内的Python学习教材或资料在介绍Python基础时大多遵循基本数据类型、程序流程控制、复合数据类型、函数与模块或者文件操作的章节安排顺序。而大数据与人工智能技术目前已广泛应用于商业、医疗、教育、交通、社交等领域,并将对学生未来的工作与生活产生越来越重要的影响,许多学生对这些技术也有着浓厚的兴趣。例如语音识别便是一种常见的人工智能应用。但是,从教学效果的角度,语音识别应用原理相对较为复杂,并不适合作为所有专业学生的Python应用案例。以商学院学生为例,笔者经过分析与探索,将广泛应用于电子商务的推荐技术作为Python应用基础案例,然后,渐进式引导学生自主思考商品销量预测等Python应用进阶案例。

配合这些案例,可按照基本数据类型、复合数据类型、程序流程控制、文件操作、函数与模块这一顺序安排教学内容。具体而言,通过介绍图书购买页面中图书名称、价格、作者等基本信息,引入基本数据类型、复合数据类型;通过介绍电子商务网站中的“同类图书排行榜”栏目,引入程序流程控制;通过介绍电子商务网站中的“浏览此商品的顾客也同时浏览”“经常一起购买的商品”“购买此商品的顾客还购买过”等栏目,引入文件操作、函数与模块,并启发学生思考如何解决相关问题。

## 2 渐进式 Python 案例教学设计的详细实例

由于应用背景与需求差异,基于 Python 的应用在实践中一般有着多种相关实现算法。因此,需要根据学生与教学内容的特点,进行详细的案例设计,并通过介绍合适的算法来激发学生的学习兴趣,培养学生解决实际应用问题的能力。下面以图书推荐与销量预测应用为例,进一步介绍在向商学院学生讲授“函数与模块”课程内容时的两个详细教学案例。

### 2.1 基础案例:图书推荐

在推荐应用案例介绍上,现有 Python 学习教材或资料中大多采用了基于协同过滤的推荐算法,并结合电影评分数据(例如 GroupLens 提供的 MovieLens 数据集)提供了相关代码与运行结果。但是,对商学院的一部分学生而言,难以在短时间内理解“如何根据用户对电影的评分确定用户或者电影之间的相似度”这一关键问题,而且相似度的计算公式与算法的实现过程相对较为复杂。

购物篮分析是实现物品推荐的另一种方法,在该方法中,只需使用“哪些物品被一起购买”的数据,通常这些数据比评分数据更容易获取,因为很多应用不会提供评分。而且,这种推荐方法并不仅仅适用于实际的购物篮,它适用于任何根据已有的多个对象推荐另一个对象的情况。例如当用户正在 MOOC 平台中学习课程时,可以给他推荐更多的课程资源。

因此,可选用购物篮分析方法来介绍图书推荐应用的实现算法与原理。不过,购物篮分析涉及关联规则挖掘算法(algorithm for mining association rules)、频繁项集(frequent itemsets)、最小支持度(minimum support)和最小可信度(minimum confidence)等概念,如果以数学公式的方式介绍这些,不仅需要占用大量课堂时间,而且需要学生有一定的数学基础才能理解。因此,可结合应用场景与算法原理,设计适合学生理解的实例,在教学方法层次上,用表格作为可视化手段辅助案例讲解。具体过程:首先,以表格的方式直观介绍图书推荐应用案例(如表 1~表 4 所示),并讲解购物篮分析的原理;其次,启发学生如何利用已学习的课程内容编程实现购物篮分析中的一些步骤;最后,介绍如何利用“函数与模块”实现这些步骤。

例如表 1 为示例的交易数据,每次交易对应

购物篮中的一组商品代码。表 2、表 3 分别为根据表 1 得到的商品、商品组合购买次数与支持度,其中支持度可由购买次数除以交易总数得到(例如  $3/5=0.6$ )。表 4 为根据表 2、表 3 得到的购物篮分析结果,其中,规则“ $\{31\} \rightarrow \{32\}$ ”表示如果用户购买了商品  $\{31\}$ ,那么他们还会购买商品  $\{32\}$ ,其可信度可由商品组合  $\{31,32\}$  的支持度除以商品  $\{31\}$  的支持度得到(例如  $0.4/0.6 \approx 0.67$ )。表 2、表 3 与表 4 中的数据均可在较短时间内引导学生使用已学的程序流程控制与复合数据类型等教学内容编程实现。

表 1 交易记录

OID	交易记录中的商品代码
1001	32,41
1002	32,47
1003	31,32
1004	31,32,47
1005	31,47

表 2 商品购买次数与支持度

商品代码	购买次数	支持度
31	3	0.6
32	4	0.8
41	1	0.2
47	3	0.6

表 3 商品组合的支持度

商品组合	购买次数	支持度
$\{31,32\}$	2	0.4
$\{31,47\}$	2	0.4
$\{32,41\}$	1	0.2
$\{32,47\}$	2	0.4
$\{31,32,47\}$	1	0.2

表 4 购物篮分析结果

规则	可信度
$\{31\} \rightarrow \{32\}$	0.67
$\{31\} \rightarrow \{47\}$	0.67
$\{32\} \rightarrow \{41\}$	0.25
$\{32\} \rightarrow \{47\}$	0.5
$\{31\} \rightarrow \{32,47\}$	0.33
$\{32\} \rightarrow \{31,47\}$	0.25
$\{47\} \rightarrow \{31,32\}$	0.67

在介绍如何利用“函数与模块”实现时,可采用规模合适的真实数据集,用更高效、简洁、实用的代码进行讲解,以强化学生解决实际问题

的能力。具体而言,可采用比利时哈瑟尔特大学(Hasselt University)Tom brijs 提供的匿名交易记录数据集<sup>[8]</sup>,该数据文件可以从网上(<http://fimi.uantwerpen.be/data/retail.dat>)获取。该文件每行分别记录一次交易中用户所购买的商品代码(用空格隔开),共有 88 162 条交易记录,数据规模合适。另外,Python 中 collections 模块中的 defaultdict 函数可快速构建一个字典对象,而 itertools 模块中的 chain 函数可快速构建一个大的迭代器,利用这两个函数可更高效获取各商品的购买次数,具体程序代码如下:

```
from collections import defaultdict
from itertools import chain
dataset = [[int(t) for t in line.strip().split()] for line in open('retail.dat')]
counts = defaultdict(int)
for elem in chain(*dataset):
    counts[elem] += 1
for i, j in counts.items():
    print('ID', i, 'count', j)
```

作为对比,若不采用 collections 与 itertools 模块,基于之前已学习的程序流程控制与字典实现的统计相应程序代码具体如下:

```
dataset = [[int(t) for t in line.strip().split()] for line in open('retail.dat')]
counts = dict()
listd = []
for data in dataset:
    listd += data
for elem in listd:
    counts[elem] = counts.setdefault(elem, 0)+1
for i in counts.keys():
    print('ID', i, 'count', counts[i])
```

从这两种代码容易看出,合适地利用已有的 Python 函数模块可明显提高编程效率。而且,还可进一步向学生介绍 time 模块中的计时函数

time,以对比这两种代码的运行时间差异。

## 2.2 进阶案例:销量预测

预测是人们工作与生活中的另一类常见问题,人们也通常基于信息做出决策。近年来,随着机器学习与人工智能技术的发展,新的预测方法不断涌现,现代预测科学也已在各行各业得到充分应用,例如社会预测、经济预测、军事预测等。但是,针对具体问题,利用合适的预测方法给出精确的预测结果并不容易。因此,预测方法相对物品推荐方法而言,学生理解与应用的难度更大。

目前,在预测应用案例介绍上,现有 Python 学习教材或资料中大多采用了股票价格、房价、球赛结果等数据。但是,这些数据与预测案例和我们之前介绍的基础案例关联并不紧密。而且,这些案例并没有详细介绍如何合适地对原始数据进行预处理、特征获取等操作,以提升预测效果,而这些操作对于解决具体应用问题非常重要。因此,对商学院的学生而言,在商品推荐基础案例的基础上,我们可进一步选用基础的预测方法,循序渐进地设计一个进阶案例,例如与商品销售相关的销量预测应用案例,并引导学生对比将原始的数据进行预处理、特征获取等操作后的预测效果,这样给学生的印象更为深刻,更能提升学生的学习兴趣。

具体而言,可选用真实的销售记录数据集。例如 CSDN 网站提供了一个包含 57 291 条销售交易记录的数据文件(<https://download.csdn.net/download/everestr/10902970>),其部分原始数据如表 5 所示,其中销售时间精确到秒,例如 20141101070457,20141101 表示年月日(2014 年 11 月 1 日),070457 表示具体时间(7 时 04 分 57 秒)。在预测方法选择上,可采用最经典的基于最小二乘法的回归预测方法,该方法容易计算,而且估计得到的预测参数具有较好的解释性。该方法的基本思想是要使各个预测值与观测值(即实际值)之间的距离(即误差)平方和达到最小。

表 5 销售交易记录的部分原始数据

单据号	销售时间	商品 ID	销售单价	销售数量	销售金额
20004053	20141101070457	995000886	4.3	1	4.3
20004054	20141101070933	1426-304	5.9	1	5.9
20004056	20141101071820	7320-320	5	1	5.0
20004058	20141101073654	995000637	3.2	1	3.2

对于表 5 中的数据,可首先给学生演示并讲解,如果直接将“销售金额”列为结果变量,其他列为预测变量,如何通过调用 sklearn 库中的 LinearRegression 等函数,以建立一个基于最小二乘法的线性回归预测模型。然后,通过指出该模型的不足(例如预测值与实际值相差较远),引入数据预处理等操作以及另一份经过处理后的数据集。最后,引导学生基于处理后的数据集更好地进行预测,具体代码如下:

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
data = np.loadtxt ( open ( 'data.txt', 'rb' ),
delimiter=',')
data_x = data[:, :2]
data_y = data[:, 2]
x_train, x_test, y_train, y_test = train_test_split(
data_x, data_y, train_size=20)
model = LinearRegression ( fit_intercept = True)
model.fit(x_train, y_train)
k_de = model.intercept_
x1_x2 = model.coef_
predict = model.predict(x_test)
plt.xlabel('序号')
plt.ylabel('总销售额')
plt.plot(y_test, 'red', label="实际值", marker='o')
plt.plot(predict, 'black', label="预测值", marker='D')
plt.legend(loc='upper right')
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['font.family'] = 'sans-serif'
plt.show()
```

### 3 应用驱动的渐进式 Python 案例教学的效果

在多年的高校程序设计语言教学实践中,笔者深深感受到,由于计算机的处理方式有时与人们的思维习惯不同,在向学生介绍程序的输入与

输出、程序是指令和数据的集合等原理时,很多毫无编程经验的学生,尤其是非计算机专业的学生,在短时间内很难理解变量、函数与程序的流程等概念。即使教师可以多花时间进行讲解,使这些学生能在上课时听懂一些对程序语句或概念的解释,如果课后学生不愿意继续学习或实践,学生真正遇到问题时依然束手无策,很难做到活学活用。因此,如何不断激发学生的程序设计兴趣,是促进学生掌握知识、提升技能并解决问题的关键所在。

根据近两年的实践,笔者认为应用驱动的渐进式 Python 案例教学方法有助于解决该问题。具体而言,笔者在本校人文学院、建筑与艺术设计学院、商学院与潇湘学院四个学院的 Python 语言程序设计必修课程中对该方法进行了实践验证。在实践中,我们以学生熟悉的应用问题为引子,循序渐进地解决应用问题,并在这个过程中向学生介绍相关概念。这不仅有助于学生在短时间内建立应用问题与 Python 语言程序设计相关概念间的逻辑对应或关联关系,而且有助于教师不断启发学生积极思考利用计算机程序设计语言解决应用问题相关的原理,引导学生发现与体会计算机程序设计的美感与乐趣,进而促进其课后主动学习并应用 Python 语言编程解决其感兴趣的问题。实践表明:采用应用驱动的渐进式 Python 案例教学可让学生从 Python 学习中逐渐提升学习兴趣与信心,激发学生自主学习与实践的动力与愿望,并有效提升学生思考与解决实际应用的能力。

### 4 结语

目前,全国已有超过 200 所高校开设了 Python 程序设计相关课程,未来将有更多的高校开设这些课程。笔者的实践表明,合适地采用应用驱动的渐进式 Python 案例教学,不仅有助于促进学生的 Python 学习,而且有助于学生理解或开展大数据处理、人工智能等前沿技术的应用。另外,在应用此方法进行教学的过程中,笔者体会到 Python 语言程序设计课程比 C 程序设计课程更适合非计算机专业学生学习,并更能促进学生主动思考与学习。希望本文能为大学 Python 课程的教学人员提供新的参考案例与方法,从而推动 Python 教学的研究和发展。

### 参考文献:

[1] 黄天羽,嵩天.以图形牵引兴趣的 Python 案例教学方

- 法与实践[J].计算机教育,2017(8):32-37.
- [2] 刘文飞,赵铭伟.面向人文社科类专业的Python教学案例实践[J].计算机教育,2017(12):15-19.
- [3] 嵩天,黄天羽.Python语言程序设计教学案例新思维[J].计算机教育,2017(12):11-14+19.
- [4] 赵广辉.面向新工科的Python程序设计交叉融合案例教学[J].计算机教育,2017(8):23-27.
- [5] 刘永芬,赖晓燕,李盼盼.人工智能时代背景下Python课程的案例教学研究[J].计算机时代,2019(4):89-92.
- [6] 张莉,金莹,张洁.多样化可扩展的Python教学体系探索与实践[J].计算机教育,2017(8):18-22.
- [7] 刘卫国,奎晓燕.Python语言程序设计教学体系建设[J].计算机教育,2017(8):13-17.
- [8] Tom Brijs, Gilbert Swinnen, Koen Vanhoof, et al. The use of association rules for product assortment decisions: a case study[C]// Chairman Fayyad, Usama, Chairman Chaudhuri, et al. Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining. New York: ACM 1999.

## Exploration and Practice of Application Driven Progressive Case-based Python Instruction

WEN Yiping<sup>a</sup>, ZHANG Jingliang<sup>b</sup>, JIANG Lei<sup>a</sup>, ZHU Jianjun<sup>a</sup>, WANG Xiangqun<sup>a</sup>, HU Rong<sup>a</sup>

(a. School of Computer Science and Engineering;

b. School of Education, Hunan University of Science and Technology, Xiangtan 411201, China)

**Abstract:** With the rapid development of big data and artificial intelligence, the college students' requirements of learning information technologies, are increasingly urgent. Python provides a useful tool for college students to solve actual application problems. Based on the practice and experience of teaching college Python courses, an application driven progressive case-based Python instruction method, is explored, which considers the professional characteristics and needs of different college students, combining appropriate application cases. Two detailed Python content teaching cases on book sale application for students from the School of Business, are provided, and it can also provide some references for Python case teaching practice.

**Key words:** Python; case-based instruction; programming; application

(责任校对 朱正余)