



上海大学

SHANGHAI UNIVERSITY

本科毕业论文（设计）

UNDERGRADUATE THESIS (PROJECT)

题 目： 测试驱动的数据科学教学平台的设计

学 院： 中欧工程技术学院

专 业： 信息工程

学 号： 19124694

学生姓名： 王宏乾

指导教师： 杜金鑫

起讫日期： 2023.1-2023.6



姓 名：王宏乾

学号：19124694

论文题目：测试驱动的数据科学教学平台的设计

原创性声明

本人声明：所呈交的论文是本人在指导教师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名：_____日 期：_____

本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

签 名：_____指导教师签名：_____日期：_____

摘要

随着数据科学的快速发展，教授数据科学技术并培养学生的数据科学能力已成为教育领域中一项重要的挑战。为了应对这一挑战，本论文提出了一种基于测试驱动的数据科学教学平台的设计，旨在通过实践和互动提高学生对数据科学的理解和应用能力。

该教学平台结合了 Jupyter Notebook、GitHub Classroom、Python 编程语言及其相关库。该平台的优势在于帮助学生更好地掌握 Python 编程语言及其相关库的应用，以及数据科学中常用的技术和工具。同时，通过集成 GitHub Classroom，学生可以方便地访问课程内容、提交作业，并与教师和其他同学进行交流和合作，促进学生与教师之间的互动和反馈。

本文详细介绍了该平台的理论背景，讨论了 Python 编程语言在数据科学中的角色，以及 Python、SQL、NumPy 库、Pandas 库、Matplotlib 库的特点和应用；详细阐述了测试驱动教学平台的设计、开发和使用，包括流程设计、任务选择以及本地项目环境的设置；展示了作业和 GitHub Classroom 的使用情况。未来的研究可以进一步探索如何优化教学平台的功能和用户体验，以满足不断发展的数据科学教育需求。

关键词：测试驱动开发； 数据科学； 教学平台； Jupyter Notebook； GitHub Classroom

ABSTRACT

Avec le développement rapide de la science des données, l'enseignement des techniques de science des données et le développement des capacités des étudiants en science des données sont devenus un défi important dans le domaine de l'éducation. Pour relever ce défi, cet article propose une conception basée sur une plate-forme d'enseignement de la science des données pilotée par des tests, qui vise à améliorer la compréhension et la capacité d'application de la science des données par les étudiants par la pratique et l'interaction.

La plateforme d'enseignement combine Jupyter Notebook, GitHub Classroom, le langage de programmation Python et ses bibliothèques associées. L'avantage de cette plateforme est d'aider les étudiants à mieux appréhender l'application du langage de programmation Python et de ses bibliothèques associées, ainsi que les techniques et outils couramment utilisés en science des données. Dans le même temps, en intégrant GitHub Classroom, les étudiants peuvent facilement accéder au contenu du cours, soumettre des devoirs, communiquer et coopérer avec les enseignants et les autres étudiants, et promouvoir l'interaction et les commentaires entre étudiants et enseignants.

Cet article présente en détail le contexte théorique de la plate-forme, discute du rôle du langage de programmation Python dans la science des données, ainsi que des caractéristiques et des applications de Python, SQL, de la bibliothèque NumPy, de la bibliothèque Pandas et de la bibliothèque Matplotlib ; développement et utilisation, y compris processus conception, sélection des tâches et configuration d'un environnement de projet local ; montre l'utilisation des devoirs et de GitHub Classroom. Les recherches futures peuvent explorer plus avant comment optimiser la fonctionnalité et l'expérience utilisateur de la plate-forme d'enseignement pour répondre aux besoins changeants de l'enseignement de la science des données.

Keywords: test-driven development; data science; teaching platform; Jupyter Notebook; GitHub Classroom

目录

摘 要.....I

ABSTRACT II

1 引言 1

1.1 研究背景与目的 1

1.2 国内外研究现状 3

1.2.1 国外发展现状 4

1.2.2 国内发展现状 5

1.2.3 未来发展趋势 5

1.3 论文结构 6

2 相关理论与背景..... 7

2.1 Python 编程语言 7

2.1.1 Python 的概述与特点 7

2.1.2 Python 在数据科学中的角色 7

2.2 SQL 8

2.2.1 SQL 的概述与特点 8

2.2.2 SQL 在数据科学中的角色 8

2.3 NumPy 库 8

2.3.1 NumPy 的概述与特点 8

2.3.2 NumPy 在数据科学中的应用 9

2.4 Pandas 库 9

2.4.1 Pandas 的概述与特点 9

2.4.2 Pandas 在数据科学中的应用 10

2.5 Matplotlib 库 10

2.5.1 Matplotlib 的概述与特点 10

2.5.2 Matplotlib 在数据科学中的应用 10

2.6 Jupyter Notebook 11

2.6.1 Jupyter Notebook 简介 11

2.6.2 Jupyter Notebook 在教学中的应用 12

2.7 Github Classroom	13
2.7.1 Github Classroom 简介	13
2.7.2 Github Classroom 在教学中的应用	14
3 测试驱动的教学平台的设计与开发	15
3.1 流程与任务	15
3.1.1 流程设计	15
3.1.2 任务设计	15
3.2 开发环境	16
3.2.1 GitHub	16
3.2.2 Pytest	18
3.3 本地化项目搭建	19
3.3.1 平台环境	19
3.3.2 搭建本地项目	21
3.3.3 Github Pull Request	22
3.3.4 常用 Git 操作	23
4 测试驱动的数据科学教学平台的使用	26
4.1 作业设计	26
4.2 Github Classroom 使用展示	33
5 总结与展望	36
5.1 总结	36
5.2 展望	37
参考文献	38
致 谢	40

1 引言

1.1 研究背景与目的

当今社会，数据已经成为一种宝贵的资源，无论是企业、政府还是学术界，都对数据的处理和分析有着日益增长的需求。数据科学作为一门跨学科的领域，旨在应对这一需求，并通过深入研究和探索数据背后的规律，为决策提供科学依据和指导。在当今信息时代，科技飞速发展，给人们的生活和工作带来了巨大的变化。特别是在数据科学领域，随着技术的飞跃，我们能够处理和分析前所未有的庞大数据集，从中提取有价值的信息和见解。这一科技飞跃的背后，是数据科学所依赖的各种技术和工具的快速发展与创新。

数据科学的研究背景可以追溯到信息技术和互联网的迅猛发展。互联网的普及和数字化的进程，使得大量的数据被记录、存储和传输。这些数据涵盖了各个领域和行业，包括社交媒体、金融交易、医疗健康、物联网等。然而，海量的数据中蕴含着许多隐藏的信息和知识，传统的数据处理和分析方法已经无法满足对数据的深入挖掘和理解的需求。因此，数据科学的兴起成为了一种必然趋势，旨在通过创新的技术和方法来处理和分析数据，从中发现新的见解和洞察。

从知识体系看，数据科学主要以统计学、机器学习、数据可视化以及（某一）领域知识为理论基础，其主要研究内容包括数据科学基础理论、数据加工、数据计算、数据管理、数据分析和数据产品开发^[1]，一方面数据科学作为新的研究方向，进一步拓展了统计学和计算机科学与技术的研究范畴^[2, 3]；另一方面，数据科学不仅继承了统计学和计算机科学等基础理论，而且对其进行了创新与发展，逐渐成为一门独立学科^[4]。

我们身处在一个数据爆炸的时代。大量的数据源不断涌现，包括社交媒体、传感器、日志文件等等。这些数据蕴含着巨大的价值和潜力，但是同时也带来了巨大的挑战。在过去，处理如此庞大和复杂的数据集几乎是不可想象的任务，然而，随着数据科学的发展，我们拥有了更加强大和高效的工具来应对这一挑战。

同时，数据科学的发展也为个人提供了更多的机会和可能性。人们现在可以通过学习数据科学技术，掌握数据处理和分析的能力，成为数据驱动决策的专家。

无论是在职场还是个人生活中，数据科学都为我们提供了更好的方式来解决实际问题、做出决策和创造价值。

当前，国内外的数据科学教育正处于快速发展的阶段。许多教育机构和学校都意识到培养学生的数据科学能力的重要性，并开始加强相关课程的设置和教学实践。然而，传统教学模式在满足学生需求和教学效果方面存在一定的不足。一方面，传统教学模式往往注重理论知识的传授，而在实践能力培养方面存在欠缺。传统的课堂讲授模式很难取得满意的教学效果^[5]。目前，国内数据科学教学主要使用 Jupyter Notebook，但大多数学习网站不支持动态的 Python 后端，学生往往缺乏实际操作的机会，难以将理论知识应用于实际场景。另一方面，数据科学的教学资源分散，学生需要从各个渠道获取学习材料和工具，缺乏统一的学习环境和支持^[6]。

为了应对传统教学模式的不足，我们需要设计和实施一个测试驱动的数据科学教学平台。该平台在网站开源，采用现有数据集^[7,8]。利用 Git 等工具更好的实行教学^[9]，主要采用 Python 语言教授数据科学的内容。这个平台将集成 Jupyter Notebook 和 GitHub Classroom 等技术，为学生提供一个全面的数据科学学习环境。首先，这个平台将整合各种数据科学所需的工具和资源，使学生能够在统一的环境中进行数据处理、分析和可视化等任务。这样一来，学生不再需要在不同的工具之间切换，能够更加专注地进行学习和实践。其次，该平台将注重实践能力的培养。通过项目实践和实际操作，学生将有机会将所学的理论知识应用到真实的数据问题中。这样的实践经验不仅能够加深对知识的理解，还能提升学生的问题解决能力和创新思维。另外，该平台将提供即时的反馈和评估机制。结合测试驱动的编程教学方法，学生可以及时了解自己的学习进展和成果。Glassey, Richard 在其论文 “Adopting Git/Github within teaching: A survey of tool support.” 中指出，由于 Github 在问题跟踪器和 pull 请求机制方面对开发人员之间的沟通和协调提供了良好的支持，因此学生反馈的机会非常丰富^[10]。他们可以通过与平台交互，获得关于代码正确性、解决问题的方法和改进建议等方面的反馈，从而不断提高自己的技能。平台将提供交流和讨论的功能，学生可以与同伴和教师进行互动，分享经验、解决问题，并一起合作完成项目任务。这种合作学习的模式能够激发学生的学习兴趣，培养团队合作和沟通能力。

本平台的设计和实施，可为数据科学教育提供一个创新的教学模式和平台支持。这将有助于提升学生的数据科学能力培养和创新思维的发展，帮助教师更好地进行教学，为数据科学教学的改进和实践提供有益的借鉴和启示。

1.2 国内外研究现状

数据科学一词最早出现在 1974 年 Peter Naur 出版的著作《Concise Survey of Computer Methods》的前言中，书中写到“数据科学是一门基于数据处理的科学”，并提及了数据科学与数据学（Datalogy）的不同——数据科学是解决数据的科学，而数据学偏向于数据处理的用于教育领域^[11]。

随着计算能力和数据存储能力的快速增长，数据科学开始成为人们探索和利用大规模数据的重要工具和方法。数据科学的研究和应用范围涉及数据获取、数据清洗、数据分析、机器学习、可视化等方面，其目标是从数据中发现模式、获得洞察并做出预测，以支持决策和创新。数据科学是一门新时代面临的新的挑战、新的机会、新的思维和新的方法为核心内容，包括新的理念、新的理论、新的方法、新的模型、新的技术、新的平台、新的工具、新的应用和实践在内的一整套完善的知识体系^[12]。Drew Conway 将数据科学定义为黑客技术、数学统计知识与实质性的专业知识的综合^[13]，如图 1.1 所示。

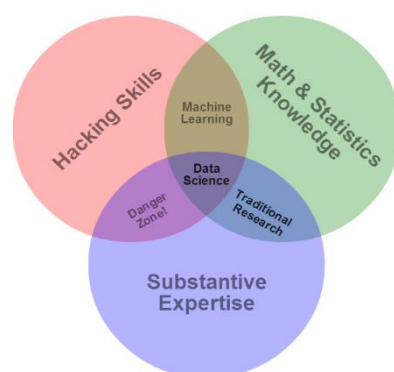


图 1.1 Drew Conway 的数据科学维恩图

然而，在数据科学领域，适合初学者自主学习的实践平台仍然相对匮乏。当前存在的一些平台或网站仅仅是进行了知识的普及，而在实际应用方面则显得不够充分。如：Coursera, California Institute of Technology Learning From Data Course 等。这些平台通常缺乏在线编程或在线实践的支持。

未来，数据科学领域将持续朝着多领域和在线学习等方向发展。数据科学的应用场景将涉及更广泛的领域，涵盖图像、文本、语音等多样化的数据类型。与此同时，为了提高数据科学教育的效果，开发更加个性化和智能化的教育平台也成为重要的方向。这样的平台能够根据学习者的兴趣、学习进度和学习风格，提供个性化的学习路径和建议，从而最大程度地促进学习效果的提升。

因此，建立一个以实践为核心的数据科学教学平台，具备重要意义。该平台将整合 Jupyter Notebook 等工具和技术，提供丰富的学习资源和实践项目，引导学习者通过实际动手操作来巩固所学知识，并培养解决实际问题的能力。同时，该平台还应提供反馈和评估机制，帮助学习者及时纠正错误并改进学习方法。通过这样的实践平台，初学者将能够更好地掌握数据科学的技术和方法，为未来在数据驱动的社会和职业中奠定坚实的基础。这样的平台的建立将填补当前数据科学教育中的空白，满足初学者自主学习的需求，并推动数据科学领域的进一步发展。

1.2.1 国外发展现状

在国际上，数据科学领域的研究和应用日益活跃。许多知名高校和研究机构建立了数据科学相关的研究中心和实验室，吸引了大量的科研人才，如哥伦比亚大学数据科学研究所、纽约大学数据科学中心等。数据科学的学术期刊和国际会议也纷纷涌现，促进了学术交流和成果共享，如 *Journal of Data Science*, *Big Data&Societ* 等。同时，一些大型科技公司和互联网企业（如 IBM、Google、Facebook 等）也成立了数据科学部门，将数据科学作为核心能力，通过数据驱动的商业模式取得了巨大成功。这些都表明数据科学在国际上的重要性和发展潜力。

由于技术原因，国外教学平台的构建已比较完善，为课程相关视频配上英、中、日、西文等字幕。在资源数量方面，国外开设的数据科学课程数量也较多^[14]。同时国外的数据集资源网站数量如 Kaggle 等，极大方便了数据集的获取。然而，为学习者提供实践操作的平台少之又少，大部分如 Coursera, KDnuggets 平台提供课程，小部分如 Topcoder 在线编程也仅是竞赛来锻炼学习者。这对于既想学课程又想立即付诸实践的学习者来说是不友好的。

1.2.2 国内发展现状

随着国内对数据科学的重视和投入增加，政府和教育机构也在积极推动数据科学教育的发展，中国国内的数据科学发展前景广阔，取得了极大的进步，总体而言正处于快速追赶的阶段。在中国，学习者会面临相较于外国更多的挑战和限制。如学生们普遍倾向于通过访问外网来获取最新的数据科学知识和资源。这主要是因为国内一些优质的学习资源和数据集在互联网上并不容易获取的缘故。而且，对于初学者而言，适合自主学习的数据科学学习平台相对较少。尽管一些在线教育平台如中国慕课 MOOC 提供了一些数据科学课程，但整体数量和质量仍有待提高。相对于大量的理论资料，国内缺乏高质量的数据科学视频资源。视频资源在学习过程中能够提供更直观和生动的教学方式，但目前可供选择的数据科学视频较为有限。与外国网站存在的问题一致，这些平台往往注重理论知识的传授，但在实践应用和项目实践方面相对较少。缺乏适合初学者进行实践的平台，导致学习者在掌握实际应用技能方面存在困难。

1.2.3 未来发展趋势

未来，数据科学将与其他学科领域深度融合，例如统计学、计算机科学等，产生跨学科的创新，产生跨学科的创新，更加深度解决实际问题。数据科学还将深度应用于多个行业领域，如金融、商业、政务、医疗、交通领域，以提高各个行业的运行效率^[15]。

为了促进数据科学技术的发展和应用，实践教学平台的设计尤为重要，这关系到学生实践能力、创新能力、思考能力等能力的培养。例如：数据科学教育平台将为不同类型的学习者提供个性化的学习路径。数据科学教育平台将提供多样化的学习内容和资源，涵盖数据处理、机器学习、数据可视化等多个领域。这些资源可以包括在线教程、教学视频、案例分析和实际数据集等，以满足学习者的不同需求。同时数据科学教学平台可以营造实践导向的学习环境：数据科学教育平台将注重实践导向，为学习者提供真实的数据科学项目和实践机会。通过参与实际项目和解决真实问题，学习者可以应用所学的数据科学技术和方法，提升实践能力和解决实际挑战的能力。这样个性化的教学平台将提高学习者的兴趣和学习效果，进一步推动数据科学技术的发展和应用。

1.3 论文结构

随着数据科学飞速发展，数据科学知识的教学越发重要。数据科学的教学不仅仅需要教学基础知识，更是需要实际性的项目操作教学来直观显现数据科学知识。在教授数据科学的过程中，既要考虑到学生的接受度问题，更要考虑到学生实践性的操作不足的问题。

在这样的理念之下，本文设计的测试驱动的数据科学教学平台旨在使用项目化方式，引导学生在实践操作中掌握数据科学知识。可以同时解决学生实践操作不足和基础知识难以理解的难题。

全文共分为五个部分，分别是：

第一章，引言。在数据科学教学的研究背景下，分多角度多层面介绍了国内外的相关研究现状以及趋势，同时对本课题的研究目标进行了阐述。

第二章，测试驱动的数据科学教学平台相关技术介绍。本章节主要介绍本课题需要使用到的相关技术，分为七个小节，分别为 Python 编程语言、Numpy 库、Pandas 库、Matplotlib 库、SQL、Jupyter Notebook、Github Classroom。对以上所需的技术进行了特点介绍与其在数据科学中的应用介绍。

第三章，测试驱动的教学平台的设计与开发。本章节说明了该教学平台的内容流程，介绍了开发环境，以及该教学平台的具体搭建。

第四章，测试驱动的数据科学教学平台的使用。本节从作业设计和 Github Classroom 使用展示两方面完整的展示了一个作业从设计、制作到上传、发给学生、再到查看学生完成情况的流程。

第五章，总结与展望。本章节总结了本课题的主要内容以及不足之处，同时点明进一步可开发方向。

2 相关理论与背景

数据科学进行数据的处理时，用到了许多积累的技术。以下将会展开说明所用到的技术功能特点以及其在数据科学中的应用。

2.1 Python 编程语言

2.1.1 Python 的概述与特点

Python 是一种高级编程语言，由 Guido van Rossum 于 1991 年创建。它具有简洁、易读和易学的语法，广泛应用于各领域。Python 使用空格和缩进来表示代码块，代码可读性强，适合初学者学习，并减少了错误可能性。它完全支持面向对象编程，使用继承、封装和多态等概念组织和管理代码。Python 拥有丰富的标准库，包括文件操作、网络通信和数据处理等功能模块。此外，它还有众多第三方库和框架，如 NumPy、Pandas、Matplotlib 和 Django，提供广泛的功能和工具。Python 跨平台，在 Windows、MacOS 和 Linux 等多个操作系统上运行。它拥有活跃的开源社区，提供丰富的资源和解决方案，不断贡献新的库和工具，使开发工作更高效便捷。

2.1.2 Python 在数据科学中的角色

Python 有比较完善的数据处理、分析和可视化的生态系统，在数据科学领域被广泛应用^[16]，包括：

1. 数据处理与清洗：Python 提供了强大的库和工具，如 Pandas 和 NumPy，用于数据的导入、清洗、转换和整理。这些工具使得数据科学家能够高效地处理和准备数据，为后续的分析 and 建模工作提供良好的数据基础。
2. 数据分析与建模：Python 拥有丰富的数据分析和建模库，如 Scikit-learn 和 StatsModels，用于统计分析、机器学习和预测建模等任务。这些库提供了各种算法和技术，帮助数据科学家从数据中发现模式、进行预测和做出决策^[17]。
3. 可视化和探索性数据分析：Python 的 Matplotlib 和 Seaborn 等库提供了丰富的可视化功能，可以创建各种类型的图表和图形，帮助数据科学家

更好地理解数据、发现趋势和传达分析结果。

4. 大数据处理和分布式计算：Python 的 Spark 和 Dask 等库提供了分布式计算和大数据处理的能力，使得数据科学家能够处理大规模数据集和进行并行计算，加快数据处理和分析的速度。同时 Python 的 sqlite3 库等可以进行数据库操作，为数据科学家提供全面的数据视图^[18]。

2.2 SQL

2.2.1 SQL 的概述与特点

SQL（Structured Query Language）是一种用于管理和操作关系型数据库的标准化语言，是数据库开始就建立的标准的操作协议。它简单易学，因其采用类似自然语言的语法结构，易于理解和使用。SQL 具有强大的数据操作能力，有丰富的操作命令和函数，可以对数据库进行数据的插入、更新、删除和查询等操作。它支持复杂的数据过滤、排序、聚合和连接等操作，满足各种数据分析和处理需求。同时，SQL 可以实现数据库管理功能，SQL 不仅用于数据的查询和操作，还提供了数据库的管理功能，包括表的创建、索引的定义、用户权限的管理等。

2.2.2 SQL 在数据科学中的角色

SQL 在数据科学中应用非常广泛，可用于数据库查询和数据提取，SQL 用于从关系型数据库中提取数据，进行数据的过滤、排序和聚合等操作。它可以帮助数据科学家快速获取所需的数据，并进行后续的分析 and 建模工作；SQL 可用于数据清洗和预处理：SQL 提供了强大的数据处理和转换功能，可以对原始数据进行清洗、去重、格式转换等操作。它可以帮助数据科学家准备高质量的数据集，以支持后续的分析 and 建模任务；SQL 可用于数据集成和数据仓库：SQL 用于数据的集成和整合，将多个数据源的数据进行统一的存储和管理。它可以实现数据的抽取、转换和加载（ETL）过程，构建数据仓库或数据湖，为数据科学家提供全面的数据操作支持。

2.3 NumPy 库

2.3.1 NumPy 的概述与特点

NumPy 是 Python 中的重要科学计算库，提供了高效的多维数组对象和丰富的数学函数。它的核心是多维数组（Nddarray），可用于存储和处理大量数据，适用于各种数值计算和数据分析任务。NumPy 的数组运算基于 C 语言编写，因此执行速度快。它提供了丰富的数学、逻辑和统计函数，可快速对数组进行元素级操作，加速数值计算。NumPy 支持广播操作，使得处理不同形状的数组变得简单，能处理不同维度和大小的数据。此外，NumPy 还提供了许多科学计算功能，如线性代数运算、傅里叶变换和随机数生成。它还与其他科学计算库（如 SciPy 和 Pandas）紧密集成，为数据科学家提供全面的计算和分析工具^[19]。

2.3.2 NumPy 在数据科学中的应用

NumPy 提供了丰富的数组操作和计算功能，如索引、切片、聚合、排序等。数据科学家可以使用 NumPy 对数据进行处理和转换，进行各种数值计算和统计分析。NumPy 的数组运算是针对整个数组进行的，而不是逐个元素进行的。这种向量化运算使得代码更简洁、高效，并提高了数值计算的速度。NumPy 提供了丰富的函数和方法，用于调整数组的形状和结构。数据科学家可以使用 NumPy 对数据进行重塑、转置、合并等操作，以满足特定的数据处理需求。NumPy 提供了各种数学和统计计算的函数，如加法、乘法、平均值、方差等。数据科学家可以利用 NumPy 进行数据的描述性统计分析、假设检验和概率计算等任务。

2.4 Pandas 库

2.4.1 Pandas 的概述与特点

Pandas 是基于 NumPy 的开源数据分析库，提供高效的数据结构和操作工具。它引入了 Series 和 DataFrame 两种核心数据结构。Series 是一维标记数组，存储不同类型的数据；DataFrame 是二维表格数据结构，类似于关系型数据库中的表，用于处理结构化数据。Pandas 具备丰富的数据操作功能，包括选择、过滤、排序、聚合等操作。它还支持处理缺失数据、合并连接数据、数据透视等高级操作，简化了数据分析和预处理流程。此外，Pandas 针对时间序列数据提供了特殊支持，包括时间索引、时间重采样、滚动窗口计算等功能。Pandas 结合了 Matplotlib 库，提供了简单而强大的数据可视化功能^[20]。

2.4.2 Pandas 在数据科学中的应用

Pandas 可以从多种数据源中导入数据，包括 CSV 文件、Excel 文件、数据库等。数据科学家可以使用 Pandas 读取数据，并将其转换为 DataFrame 的形式，以便进行后续的分析 and 处理。Pandas 提供了强大的数据清洗和预处理功能，如缺失数据的处理、重复数据的处理、异常值的处理等。数据科学家可以使用 Pandas 对原始数据进行清洗和转换，准备好用于分析的数据集。Pandas 提供了灵活的数据选择和过滤功能，可以根据条件筛选和提取数据。数据科学家可以使用 Pandas 的索引和条件查询功能，对数据进行快速的子集选择和过滤，以满足特定的分析需求。Pandas 支持数据的聚合和分组计算，可以根据某个或多个列对数据进行分组，然后进行统计计算，如求和、均值、计数等。这对于数据的汇总和摘要分析非常有用。Pandas 结合了 Matplotlib 库的功能，可以快速绘制各种类型的图表和图形。数据科学家可以使用 Pandas 的可视化功能，快速生成图表，帮助理解数据和发现数据中的模式和趋势。

2.5 Matplotlib 库

2.5.1 Matplotlib 的概述与特点

Matplotlib 是一款强大的数据可视化库，被广泛应用于数据科学。它提供了丰富的绘图工具和函数，用于创建高质量、美观、可定制的图形，用于数据的探索、分析和展示。

Matplotlib 的特点包括多种图形类型，如折线图、散点图、柱状图、饼图、等高线图等，还支持 3D 图形和动画。它具有灵活性和可定制性，用户可以根据需要自定义图形的各个方面，如线条样式、颜色映射、坐标轴标签和图例。同时，Matplotlib 支持高质量的输出，用户可以方便地将图形保存为不同格式的文件，以便于出版和展示。

2.5.2 Matplotlib 在数据科学中的应用

Matplotlib 在数据科学中具有广泛的应用。Matplotlib 提供了强大的工具和函数，使用户能够对数据进行可视化探索。用户可以绘制各种类型的图形，例如折

线图用于显示时间序列数据的趋势，散点图用于观察变量之间的关系，柱状图用于比较不同组别的数据等。这些图形有助于揭示数据中的模式、趋势和异常情况。**Matplotlib** 可以用于可视化数据分析的结果和模型评估的指标。用户可以绘制柱状图或箱线图来比较不同算法或模型的性能，绘制误差条图来显示预测结果的置信区间，以及绘制热力图或等高线图来展示数据的相关性或空间分布等。这有助于用户对数据分析和模型评估结果进行直观的理解和解释。**Matplotlib** 提供了丰富的绘图功能和定制选项，使用户能够创建高质量的图形用于报告和展示。用户可以设置图形的标题、坐标轴标签和图例，调整线条和颜色的样式，添加注释和文本等，以使图形更具吸引力和专业性。这有助于用户将数据科学的结果和发现以清晰、简洁和易于理解的方式呈现给他人。

2.6 Jupyter Notebook

2.6.1 Jupyter Notebook 简介

Jupyter Notebook 是一种流行的交互式计算环境，广泛应用于数据科学和机器学习领域。它的灵活性和可视化能力使得数据分析和实验变得更加直观和易于理解。本节将更详细地介绍 **Jupyter Notebook** 的特点和在数据科学中的应用。

Jupyter Notebook 最初是由 Fernando Pérez 于 2011 年创建的，其名称源自三个主要编程语言的缩写：**Julia**、**Python** 和 **R**。它是一个开源项目，旨在提供一个交互式的计算环境，使得代码、文本、图像和其他多媒体内容能够结合在一个单一的文档中。**Jupyter Notebook** 支持多种编程语言，包括 **Python**、**R**、**Julia**、**Scala** 等，因此在各个数据科学领域都有广泛的应用。

Jupyter Notebook 提供了一个交互式的计算环境，用户可以在浏览器中编写和执行代码，并即时获取结果。这种实时反馈的特性使得数据科学工作变得更加迭代和探索性，可以快速尝试不同的代码片段和数据处理方法。**Jupyter Notebook** 将代码和文档整合在一起，用户可以在同一个文档中编写代码和记录分析过程。这种文档式编程的方式使得数据科学工作更具可读性和可理解性，方便其他人理解代码和分析结果。**Jupyter Notebook** 支持在文档中插入图像、公式、**HTML**、**Markdown** 和其他多媒体内容，使得结果展示更加丰富和易于理解。用户可以通过

过 Markdown 语法编写文本说明、添加标题和分段，以及插入图片和超链接等，从而使文档更具可读性和可视化效果。Jupyter Notebook 可以保存为可分享的文件，并通过各种方式共享给他人。用户可以将 Notebook 文件发布到公共或私有的存储库中，例如 GitHub，方便团队成员之间的协作和交流。此外，还可以将 Notebook 转换为不同的格式，如 HTML、PDF 或 Slides，以便于在不同场景中展示和分享。Jupyter Notebook 支持通过插件和扩展来增强其功能。用户可以安装各种扩展，如自动完成、语法高亮和代码格式化等，以满足个人的编程习惯和需求。此外，还可以编写自定义的扩展和魔术命令，扩展 Notebook 的功能和能力。Jupyter Notebook 界面如图 2.1 所示。

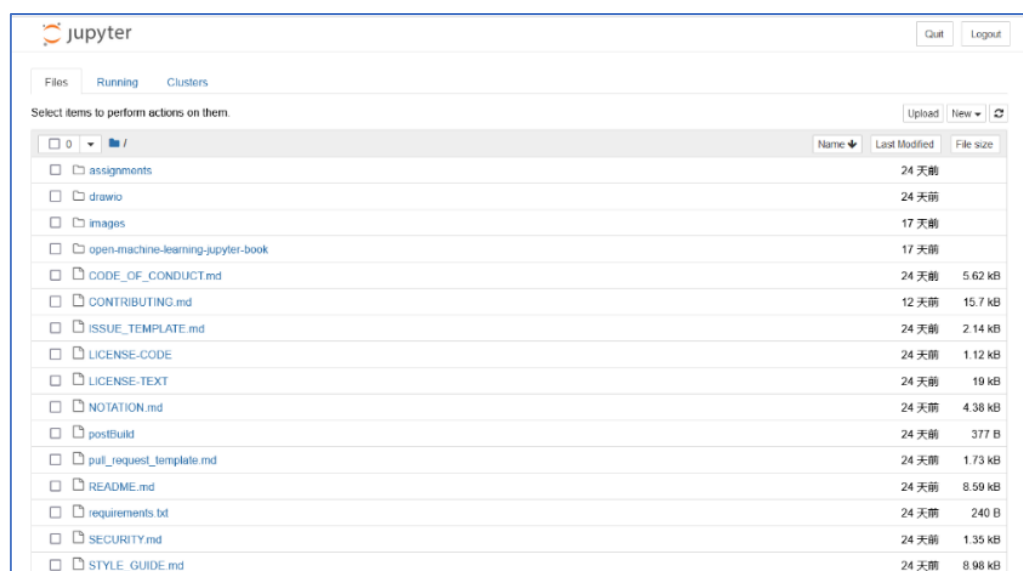


图 2.1 Jupyter Notebook 界面

2.6.2 Jupyter Notebook 在教学中的应用

教师期望设计学习条件，以此让学生更好的参与教学，展现自我思考，学习掌握知识等等，使用 notebook 能帮助教师更好地达到这些目标。而学生能通过 notebook 提高自己的知识水平，培养编程思维，自主学习能力等等。教师在教学可以使用填空模式，在布置任务时可以写好整体框架，留空给学生填写，比如，在学习数据科学时，教师在 notebook 中写好函数 `drop_columns` 的输入，加以注释，函数内部的逻辑空着，留给学生填，空白多少照这节课的难易度而定^[21]。教师和学生可以把 notebook 当作教科书来使用，同传统的纸质教科书不同，notebook 是动态的，可以直观显现每一步操作，让学生更好的接受那些新知识。

同时 Jupyter notebook 的每个单元格都能很方便嵌入视频或者音频，方便学生观看。Jupyter notebook 可以很好的胜任作业任务，使用 Jupyter notebook 编写作业并检验作业是个很好的选择。使用 JupyterBook 进行数据科学的项目式教学，将极大提升课程教学效果，加快课程效率，打开学生视野，有利于学生动手能力的提升。

Jupyter 平台可以支持代码与结果展示一体融合；支持当下最为流行的 Python 语言及其各种数值分析、数据处理和机器学习、深度学习开发包^[22, 23]，如：Numpy、Pandas、Scikit—learn、Tensorflow、Pytorch、Keras 等；支持分布式运行，教学过程中，可以不顾课堂对于各种复杂运行环境配置和安装的要求，进而实现一处安装轻松讲课和完美展示的效果^[24]。教师可以利用 Jupyter Notebook 设计电子学案，根据教学内容难度系数合理进行学习资料的配给，并通过不同资料讲解来生动教学^[25]。Jupyter Notebook 编程工具，可以很好地为项目教学发挥支撑作用，是教学中的利器，值得广泛应用^[26]。

2.7 Github Classroom

2.7.1 Github Classroom 简介

GitHub Classroom 是一个基于 GitHub 的教育工具，旨在简化教师和学生之间的协作和管理。它提供了一个集成的平台，使教师能够创建和分发作业，学生能够提交作业，并进行评估和反馈。下面将详细介绍 GitHub Classroom 的特点和在教学中的应用。GitHub Classroom 提供了教师管理作业的功能。教师可以轻松创建课程、作业和项目，并将其与 GitHub 存储库相关联。通过简单的界面和指导，教师可以管理学生的代码库，查看提交记录和评估学生的作业。学生可以使用 GitHub Classroom 提交作业。他们可以通过链接加入课程和作业，自动创建与作业相关的个人代码库，并将作业提交到相应的代码库中。这种方式使得学生作业的提交过程更加简便和集中化。GitHub Classroom 建立在 Git 版本控制系统之上，为学生提供了版本控制和协作的能力。学生可以通过分支、合并和提交等操作进行代码的管理和协作，同时可以随时回溯和恢复之前的代码状态。GitHub Classroom 支持与自动化评估工具的集成，如测试框架（如 Pytest）和代码检查工

具（如 Pylint）。教师可以设置自定义的测试用例和评分规则，自动对学生的代码进行评估，并提供相应的反馈和成绩。GitHub Classroom 允许学生之间进行社交学习和讨论。学生可以在作业和项目的评论区进行交流和互动，分享问题、解决方案和经验。这种交流和合作的方式促进了学生之间的互动和知识共享。GitHub Classroom 界面如图 2.2 所示。

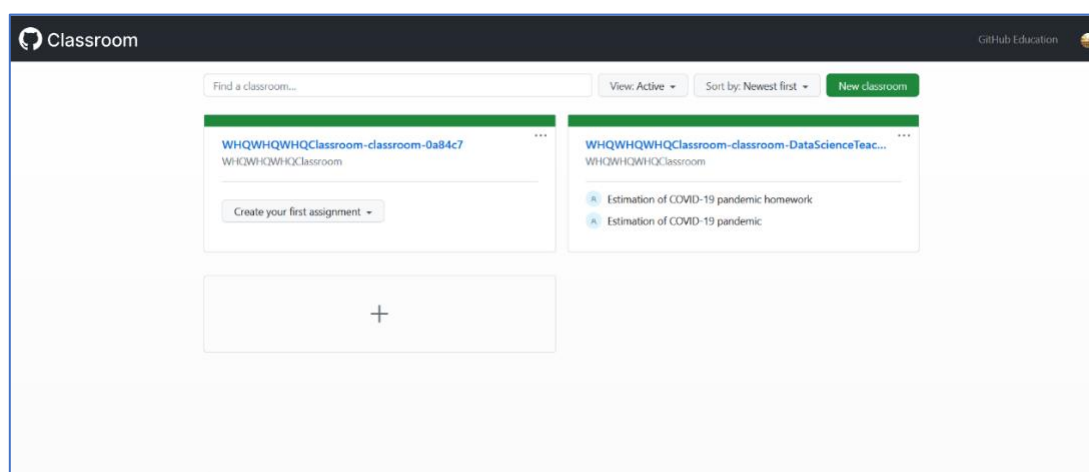


图 2.2 Github Classroom 界面

2.7.2 Github Classroom 在教学中的应用

教师可以使用 GitHub Classroom 创建编程作业并将其分发给学生。通过 GitHub 的版本控制功能，学生可以克隆作业存储库并将其作为基础进行开发和提交。教师可以轻松跟踪学生的提交历史、提供反馈和评分。学生可以创建自己的分支，独立地进行实验和改进，并通过合并请求将更改合并到主分支中。这种协作和版本控制的能力培养了学生的团队合作和代码管理技能。

GitHub Classroom 可以与自动化测试框架（如 Pytest）集成，对学生的作业进行自动化测试。这样，教师可以为作业定义测试用例，并自动评估学生的代码是否通过了这些测试。这种自动化测试和评估的方式减轻了教师的负担，并提供了及时反馈给学生。

GitHub Classroom 还可以鼓励学生参与开源项目的贡献。教师可以选择开源项目作为作业，并要求学生提交代码贡献。通过这种方式，学生可以学习开源社区的工作流程、协作和代码贡献的实践经验^[27]。

3 测试驱动的教学平台的设计与开发

3.1 流程与任务

3.1.1 流程设计

该平台的设计主旨是提供高效和在线的学习体验，因此教学内容着重于数据科学领域最关键且必须掌握的知识。学生通过学习并完成作业，并通过在线检测获得提升的机会。这种学习方式不仅帮助学生提升技能，还通过反馈机制不断改进和丰富教学内容，使其更加完善和高效。流程如图 3.1 所示。

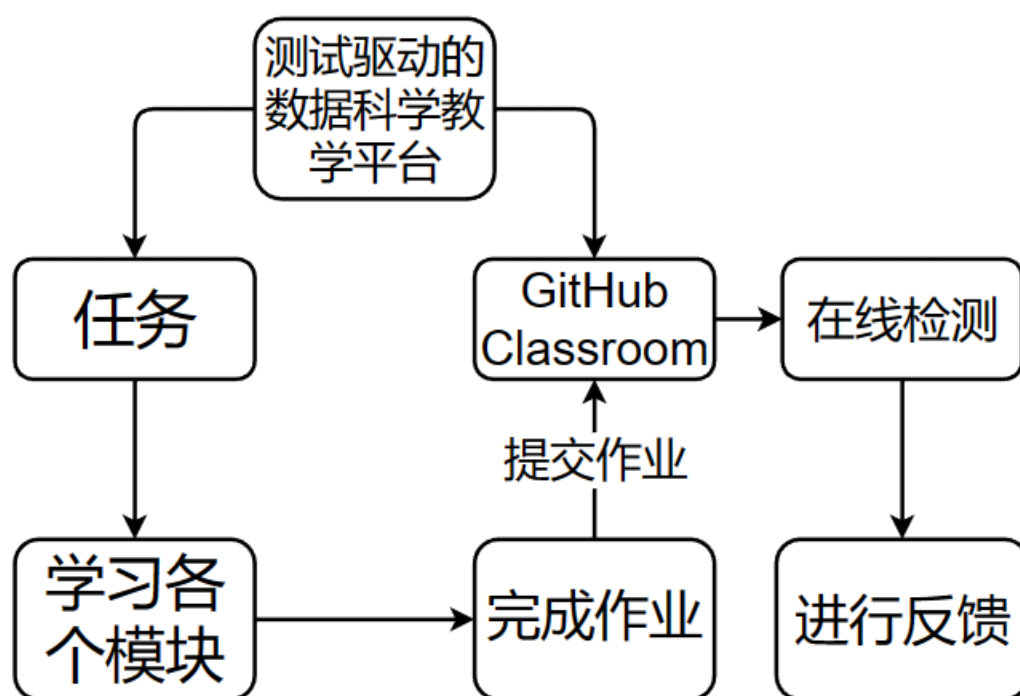


图 3.1 进行教学的整体流程

3.1.2 任务设计

在测试驱动的数据科学教学平台中，通过做作业的方式来进行在线检测正确性，采取了以下设计方式：

1. 题目设计：每个作业包含多个题目，每个题目对应一个要求学生实现的函数或任务。题目应该涵盖数据科学中的关键概念和技术，既能够测试学生对知识的理解，又能够锻炼他们的编程和数据处理能力。
2. TEST CASES 编写：为每个题目编写一组充分的测试用例，覆盖各种可

能的输入情况和预期输出。测试用例涵盖正常情况和边界情况，以确保学生的代码在各种情况下都能正确运行。

3. 挖空设计：在每个题目中，将函数的部分代码挖空，要求学生根据题目要求和给定的测试用例，补充完整函数的实现。挖空的部分应该涵盖核心的算法或数据处理步骤，以便学生能够理解和应用相关的数据科学技术。
4. 在线检测：学生在平台上提交他们的代码实现后，平台会自动运行测试用例，并根据实际输出与预期输出进行比对，判断学生代码的正确性。如果学生的代码通过了所有测试用例，表示实现正确；如果有测试用例失败，平台会提供相应的错误信息和反馈，帮助学生找出问题并改进代码。
5. 提供解析和讨论：对于每个题目，平台提供解析和讨论，解释正确的实现方法和相关的数据科学知识，帮助学生理解和巩固所学的内容。这有助于学生对测试结果进行分析和思考，进一步提升他们的学习效果。

通过以上设计，学生可以通过完成作业并在线检测正确性来进行自我评估和学习进度的掌握。同时，平台的反馈和解析部分也能够提供学生所需的指导和辅助，使他们能够更好地理解和应用数据科学技术。这种实践和交互的学习环境有助于学生深入理解数据科学的核心技术，并培养解决实际问题的能力。

3.2 开发环境

3.2.1 GitHub

GitHub 是一个基于互联网的代码托管平台，旨在为开发人员和团队提供集中存储、版本控制、协作和代码管理的解决方案。作为一个使用 Git 作为底层版本控制系统的平台，GitHub 为开发人员提供了强大的工具和功能，以便更好地管理和协作开发项目。

在 GitHub 上，用户可以轻松创建存储库（Repository），用于存储其代码和项目文件，并利用 Git 的版本控制功能来跟踪和管理代码的变更历史。这使得团队成员能够在不同的分支上并行开发，每个分支代表着一个独立的工作流程，而无

需直接影响主分支。每个开发者可以根据项目需求，在自己的分支上进行代码编写、修改和提交，并使用 Git 命令进行提交和同步操作。

GitHub 提供了丰富的协作功能，其中包括问题追踪、代码审查、讨论区和项目文档等。这些功能有助于团队成员之间更好地交流和合作，提高项目开发的效率和质量。通过问题追踪，团队成员可以共享和解决项目中的问题和任务，保持对项目进展的清晰认识。代码审查功能使团队能够对彼此的代码进行仔细检查，提出建议和改进意见，以确保代码的质量和一致性。讨论区则为团队成员提供了一个交流的平台，可以讨论项目中的想法、技术问题和解决方案。此外，GitHub 还允许用户编写和共享项目文档，使得项目的文档化和知识的传递更加方便。GitHub 界面如图 3.2 所示。

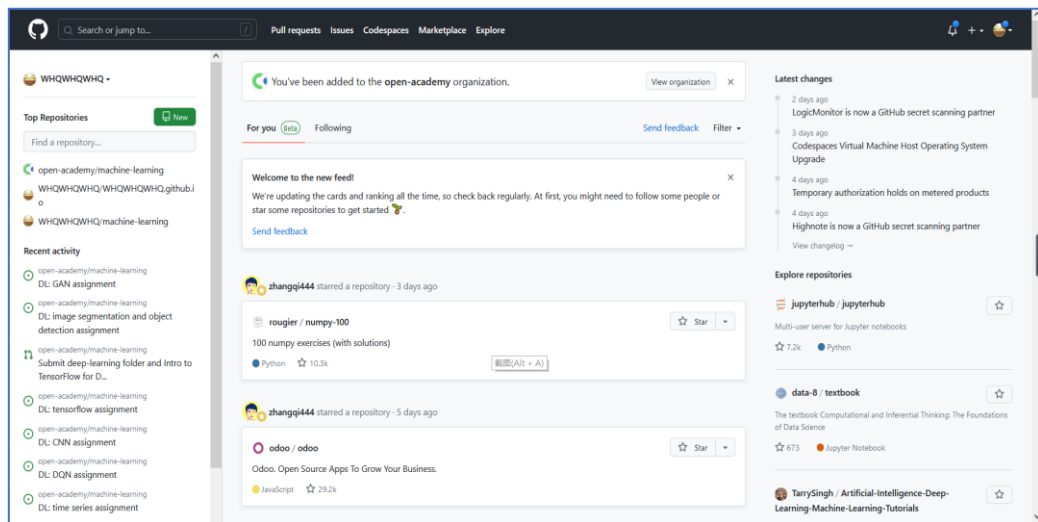


图 3.2 GitHub 页面

当多人协作开发一个项目时，Git 和 GitHub 提供了一种称为 Pull Request (PR) 的机制，用于管理代码变更和进行代码审查。

1. 使用命令 `git clone <repository_url>`。使每个开发者将远程存储库（GitHub 上的项目）克隆到本地计算机上。
2. 使用命令 `git checkout -b <branch_name>`。这样可以确保每个开发者在独立的分支上进行工作，而不会直接影响主分支。
3. 使用命令 `git add <file>` 和 `git commit -m "commit message"`。使得每个开发者在自己的分支上进行代码编写、修改和提交 `commit`。
4. 使用命令 `git push origin <branch_name>`。使得开发者完成某个功能或修

复之后将本地的分支推送到远程存储库。

在 GitHub 上，开发者可以创建一个 Pull Request（以下简称 PR），将他们的分支与主分支进行比较和合并。他们可以提供有关所做更改的描述和说明。其他团队成员或项目维护者可以查看 PR，并进行代码审查。他们可以提出建议、提问或对代码进行评论。这样可以确保代码的质量和一致性。在 PR 中，团队成员可以对代码进行讨论，并根据审查意见进行修改和改进。这可以促进团队合作和知识共享。当代码经过审查并达到一定质量标准后，可以选择合并（Merge）PR。这将把开发者的更改应用到主分支上。通过这种方式，多人可以并行开发不同的功能或修复不同的 bug，并通过 PR 进行代码审查和合并。这种工作流程能够提高代码质量、减少冲突，并促进团队合作和代码协作。

GitHub 通过提供强大的代码托管、版本控制和协作功能，为开发人员和团队提供了一个优秀的平台来管理和协作开发项目。它极大地提高了开发效率、代码质量和团队合作，并推动了知识共享和技术发展的进程。无论是个人项目、团队项目还是开源项目，GitHub 都是一个不可或缺的工具和社区。

3.2.2 Pytest

Pytest 是一个功能强大、易于使用且灵活的测试框架，被广大 Python 开发者所喜爱和采用。它提供了许多优秀的特性和功能，可以帮助开发者编写高质量的测试代码，确保软件的稳定性和可靠性。

1. Pytest 的语法简洁清晰，相比传统的 Unittest 模块，编写测试用例更加直观和简单。开发者只需按照约定的命名规则编写测试函数，Pytest 会自动识别并执行这些函数作为测试用例。这种简洁的语法使得编写和维护测试代码变得更加轻松和高效。
2. Pytest 具有强大的自动化测试发现功能。通过遵循一些命名规则和目录结构约定，Pytest 能够自动发现并执行项目中的所有测试用例，无需手动指定。这使得测试的管理和执行变得非常便捷，节省了开发者的时间和精力。
3. Pytest 提供了丰富的断言库，涵盖了常见的相等性、包含性和异常等断言。开发者可以编写详细而准确的测试断言，验证代码的预期行为。这

些断言库的丰富性和灵活性使得开发者能够更全面地测试代码，发现潜在的问题和错误。

4. **Pytest** 还支持参数化测试，开发者可以通过一组参数运行相同的测试用例，以覆盖不同的输入和边界条件。这样可以减少测试用例的重复编写，并提高测试的覆盖率。参数化测试的支持使得测试更加全面和可靠。
5. **Pytest** 具有丰富的插件生态系统，开发者可以根据需要选择并安装各种插件，以扩展 **Pytest** 的功能。这些插件提供了额外的报告生成、代码覆盖率分析、并发执行等功能，满足各种测试需求。开发者可以根据项目的具体需求，选择适合的插件来增强测试框架的能力。
6. **Pytest** 与其他常用工具（如持续集成工具、代码覆盖率工具、测试管理工具等）集成良好。它可以与这些工具无缝配合使用，提高测试的效率和质量。这种良好的集成性使得开发者能够在整个开发流程中，无缝地进行测试，并与其他工具进行交互和协作。

Pytest 通过其简洁的语法、自动化测试发现、丰富的断言库和插件扩展性，为开发者提供了优秀的测试支持，使其成为 **Python** 开发者喜爱的工具。它不仅适用于单元测试、集成测试和功能测试，还与其他工具的集成使得测试工作更加高效和可靠。无论是单元测试、集成测试还是功能测试，无论是小型项目还是大型项目，**Pytest** 都能提供优秀的测试支持，帮助开发者编写高质量的代码并保证软件的稳定性^[28]。

3.3 本地化项目搭建

3.3.1 平台环境

本平台需要搭建一个全新的虚拟环境，以下是学习数据科学，在该虚拟环境下需用的 **Python** 库：

1. **NumPy**: **NumPy** 是 **Python** 中科学计算的核心库，提供了高性能的多维数组对象和各种数学函数，用于处理大规模的数值数据。
2. **Pandas**: **Pandas** 是一个强大的数据处理和分析库，提供了高性能、灵活和易于使用的数据结构，如 **DataFrame** 和 **Series**，用于数据清洗、转换、

合并和统计分析等任务。

3. **Matplotlib:** Matplotlib 是一个广泛使用的数据可视化库，提供了各种绘图工具和函数，用于创建各种类型的图表、图像和可视化效果。
4. **Seaborn:** Seaborn 是基于 Matplotlib 的统计数据可视化库，提供了更高级别的接口和样式，用于绘制各种统计图表和热图等。
5. **Scikit-learn:** Scikit-learn 是一个流行的机器学习库，提供了各种机器学习算法和工具，包括分类、回归、聚类、降维和模型选择等，用于构建和训练机器学习模型。
7. **TensorFlow:** TensorFlow 是一个开源的机器学习框架，用于构建和训练各种机器学习模型，包括深度神经网络和自然语言处理模型等。
8. **Keras:** Keras 是一个高级神经网络库，提供了简单而强大的接口，用于构建和训练深度学习模型。
9. **PyTorch:** PyTorch 是另一个流行的深度学习框架，提供了动态计算图和强大的 GPU 加速，用于构建和训练深度神经网络模型。

该虚拟环境具体的安装方式如下：

1. 首先，确保已经安装了 Anaconda。这是 Conda 的发行版，提供了 Python 和 Conda 包管理工具。
2. 打开终端（对于 Windows 用户，打开 Anaconda Prompt 或命令提示符）。
3. 创建一个新的 Conda 虚拟环境，使用以下命令：

```
1. conda create --name myenv
```

这将创建一个名为“myenv”的新环境。在配置时，将“myenv”替换为“myconda_1”，这样就有了一个名为“myconda_1”的 conda 虚拟环境。

4. 激活虚拟环境，运行以下命令：

```
1. conda activate myconda_1
```

这样就激活了“myconda_1”的虚拟环境。

5. 安装所需的包和软件包，例如：

```
1. conda install numpy pandas matplotlib
```

这样虚拟环境中就有了 `numpy`, `pandas`, `matplotlib` 库, 以此类推, 把所需的库全部下载下来。

如此, 就成功安装和配置了一个 Conda 虚拟环境, 如图 3.3 所示。

```
加载个人及系统配置文件用了 2930 毫秒。  
(base) C:\Users\HP> conda activate myconda_1  
(myconda_1) C:\Users\HP>
```

图 3.3 激活新虚拟环境

3.3.2 搭建本地项目

本地项目环境的搭建需要在 Powershell 中的“终端”来完成, 且提前下载 Git 工具, 具体搭建方式如下:

1. 首先将远程仓库克隆到本地, 使用如下指令:

```
1. git clone https://github.com/<UserName>/machine-learning.git
```

2. 按照 Jupyter Notebook 官方说明文档下载并配置 Jupyter Notebook^[29]
3. 移动到工作目录

```
1. cd machine-learning/open-machine-learning-jupyter-book/
```

4. 我们必须先确认我们项目的源, 确保源是正常的, 输入指令:

```
1. git remote -v
```

来查看我们的远程配置情况。

5. 我们必须正确配置 SSH 协议确保网络连接稳定。这里我们必须先配置 SSH 公钥和私钥。我们按照 atlassian 提供的配置方法进行配置^[30]。
6. 我们将远程仓库的 http 协议替换为 ssh 协议, 具体操作为输入指令

```
1. git remote set-url origin git@github.com:<username>/<repository>.git
```

7. 增加 upstream 源

```
1. git remote set-url upstream git@github.com:open-academy/machine-learning.git
```

8. 本地项目搭建完毕。

3.3.3 Github Pull Request

GitHub Pull Request（简称 PR）是 GitHub 平台上用于管理代码变更和进行代码审查的机制。当多人协作开发一个项目时，PR 提供了一种结构化的方式来处理代码修改和团队合作。下面将详细介绍 GitHub Pull Request 的工作流程和主要功能：

1. 创建分支（Branch）：每个开发者在开始工作之前，需要将远程存储库（GitHub 上的项目）克隆到本地计算机，并创建一个新的分支来独立进行工作。这可以确保每个开发者在自己的分支上进行代码编写，而不会直接影响主分支。可以使用命令 `git clone <repository_url>` 克隆存储库，然后使用命令 `git checkout -b <branch_name>` 创建并切换到新的分支。
2. 编写和提交代码：每个开发者在自己的分支上进行代码编写、修改和提交。可以使用命令 `git add <file>` 将文件添加到暂存区，然后使用命令 `git commit -m "commit message"` 提交代码变更。开发者可以根据项目需求进行功能开发或修复 bug。
3. 推送分支：当开发者完成某个功能或修复时，他们将本地的分支推送到远程存储库，以便与其他团队成员共享和进行代码审查。可以使用命令 `git push origin <branch_name>` 将本地分支推送到远程存储库。
4. 创建 PR：在 GitHub 上，开发者可以创建一个 PR，将他们的分支与主分支进行比较和合并。通过在项目页面上点击"New Pull Request"按钮并选择相应的分支，可以创建一个新的 PR。在创建 PR 时，开发者可以提供有关所做更改的描述和说明。此描述通常包括解决的问题、新增功能的说明或修复的 bug 的描述等。
5. 代码审查和讨论：其他团队成员或项目维护者可以查看并审查 PR。他们可以查看代码变更的具体内容，提出建议、提问或对代码进行评论。这样可以确保代码的质量和一致性。GitHub 提供了丰富的协作功能，包括评论、线上讨论和代码比较等，以便团队成员之间进行交流和协作。在 PR 中，团队成员可以对代码进行讨论，并根据审查意见进行修改和改进。

6. 迭代和改进：根据代码审查的反馈，开发者可以进行相应的修改和改进。这可能包括修复 bug、优化代码或添加缺失的功能。开发者可以将修改后的代码推送到同一分支，并 PR 会自动更新，审查者可以查看更新后的代码。
7. 合并 PR：当 Pull Request 通过代码审查并满足项目要求时，可以选择将其合并到主分支。通过点击"Merge Pull Request"按钮，可以将更改合并到主分支中。这样，主分支就包含了最新的修改和改进。

GitHub Pull Request 的优势在于它提供了一个结构化的协作和审查流程，使多人协作开发更加方便和高效。它促进了团队成员之间的交流和讨论，帮助发现和修复问题，并确保代码质量和项目稳定性。此外，通过使用 PR，可以跟踪代码变更的历史记录，保留审查意见和讨论，并记录每个功能或修复的来源。

总而言之，GitHub Pull Request 是一种强大的机制，用于管理代码变更、促进代码审查和团队协作。它在开源项目和团队开发中被广泛采用，为代码质量和项目管理提供了重要支持。

3.3.4 常用 Git 操作

由于平台需要不断迭代，不断更新，所以必须有合适的指令进行操作。其中，Git Pull、Git Push，Git Rebase 是 GitHub 中常用的操作，用于将本地代码推送到远程仓库或从远程仓库拉取更新。下面详细介绍这三个操作的过程和作用。

3.3.4.1 Git Pull 操作

在 GitHub 中，Pull 操作用于从远程仓库拉取更新到本地仓库。它的主要作用是将其他开发者提交的代码更新同步到本地仓库，以便在本地进行开发工作。以下是 Pull 操作的详细步骤：

打开终端或命令行界面，进入本地仓库的目录。执行命令“git remote add origin 远程仓库的 URL”，将远程仓库与本地仓库关联起来（如果还未关联），这将把远程仓库的 URL 与本地仓库关联起来，方便进行 Pull 和 Push 操作。执行以下命令，拉取远程仓库的更新到本地仓库“git pull origin 分支名称”，这将从远程仓库的特定分支拉取最新的更新到本地仓库。如果有冲突（多个开发者在同一文件

的同一位置做了修改），Git 会提示解决冲突。这时需要手动解决冲突，编辑文件并选择要保留的更改。

通过使用 Pull 操作，开发者可以将其他人的代码更改同步到本地仓库，以便保持最新的代码状态，并避免与其他人的工作产生冲突。

3.3.4.2 Git Push 操作：

在 GitHub 中，Push 操作用于将本地仓库的更改推送到远程仓库，以便其他开发者可以获取到最新的代码。以下是 Push 操作的详细步骤：

打开终端或命令行界面，进入本地仓库的目录。对本地仓库进行修改、添加或删除文件，完成相关的开发工作。完整的 Git Push 操作包含三个步骤：

1. 执行命令 `git add 目录/文件`，可以将修改的文件添加到本地仓库的暂存区。
2. 执行命令 `git commit -m "提交信息"`，可以将暂存区的更改提交到本地仓库，并添加一条描述性的提交信息。
3. 执行命令 `git push origin 分支名称`，可以将本地仓库的更改推送到远程仓库。

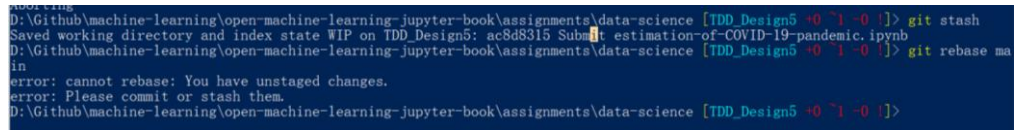
通过使用 Git Push 操作，开发者可以将本地仓库的更改推送到远程仓库，使其他开发者能够获取到最新的代码，并在协作开发中进行同步和合并操作。

3.3.4.3 Git Rebase 操作

当开发者执行 Pull 或 Push 操作时遇到冲突，这时考虑使用 Rebase 操作来解决冲突。Rebase 操作允许本地使用者将本地提交放置在远程提交之前，从而更好地整合代码，并解决冲突。以下是使用 Rebase 操作解决冲突的一般步骤：

1. 更新本地仓库：在执行任何操作之前，确保本地仓库是最新的，可以使用 `git pull` 命令将远程仓库的更改合并到本地仓库。使用 `git rebase` 命令来执行 Rebase 操作。例如，如果想要将当前分支与远程分支进行 Rebase 操作，可以使用命令 `git rebase origin/branch_name`，将分支名替换为要 Rebase 的远程分支名称。

2. 解决冲突：执行 Rebase 操作后，Git 提醒需要将文件冲突解决。打开这些文件，手动编辑冲突部分，并根据需求进行调整。确保最终的代码是符合预期的，并解决冲突。删除冲突标记，只保留需要的代码，并合并不同版本的代码。Rebase 失败界面如图 3.4 所示：



```
D:\Github\machine-learning\open-machine-learning-jupyter-book\assignments\data-science [TDD_Design5 +0 ~1 -0 ~]> git stash
Saved working directory and index state WIP on TDD_Design5: ac8d8315 Submission: estimation-of-COVID-19-pandemic.ipynb
D:\Github\machine-learning\open-machine-learning-jupyter-book\assignments\data-science [TDD_Design5 +0 ~1 -0 ~]> git rebase ma
in
error: cannot rebase: You have unstaged changes.
error: Please commit or stash them.
D:\Github\machine-learning\open-machine-learning-jupyter-book\assignments\data-science [TDD_Design5 +0 ~1 -0 ~]>
```

图 3.4 Git rebase 显示冲突

3. 保存更改：完成冲突的解决后，保存文件，并确保文件处于正确的状态。
4. 继续 Rebase 操作：执行 `git rebase --continue` 命令来继续进行 Rebase 操作。如果还有其他冲突需要解决，重复步骤 3 和步骤 4，直到所有冲突都得到解决。
5. Push 更改：在完成 Rebase 操作并解决所有冲突后，使用 `git push` 命令将更改推送到远程仓库。

通过使用 Rebase 操作来解决冲突，开发者可以更好地整合代码并保持代码库的整洁性。Rebase 操作允许将本地提交放置在远程提交之前，从而使代码历史更加线性和清晰。然而，需要注意的是，在执行 Rebase 操作时，要小心处理冲突并确保代码的正确性。合理使用 Rebase 操作可以提高团队协作的效率，确保代码变更的顺利合并和整合。通过了解和掌握 Rebase 操作的步骤，开发者可以更好地处理冲突，并保持代码库的可维护性和稳定性。

4 测试驱动的数据科学教学平台的使用

4.1 作业设计

在该小节中，本文将展示作业文档的制作和上传。以“Estimation of COVID-19 pandemic”作业为例。首先，为了制作一份作业，我们要使用 Jupyter Notebook 的后缀名为.ipynb 的文件，这类文件可以支持 Markdown 语言和 Python 代码块，我们要在 Markdown 语言部分中写文字部分，在 Python 代码块中写代码。在文件的开头，我们要先写入标题，写指示，然后加载必不可少的库，如图 4.1 所示。

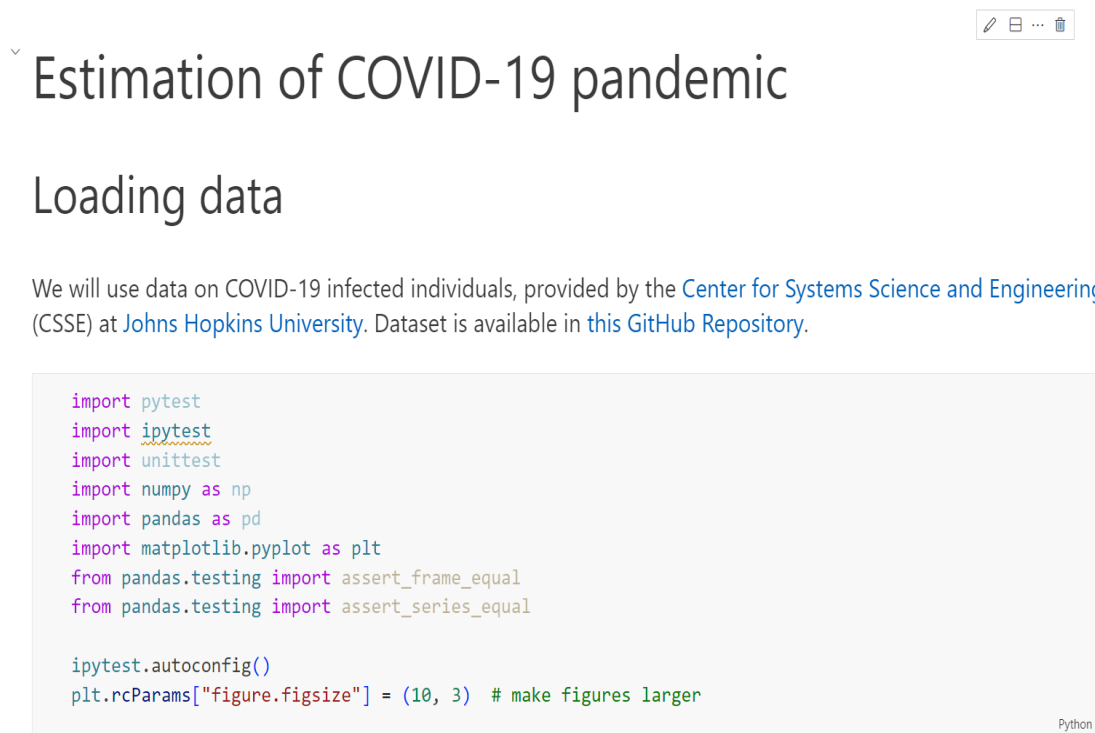


图 4.1 写入标题，指示和加载库

为了读取 csv 数据，我们可以选择从已下载好的数据集文件中读取。这里使用到了 Pandas 库的 `pd.read_csv(file)` 函数，用于读取数据集文件并自动构建 DataFrame。如图 4.2 所示。

We can load the most recent data directly from GitHub using `pd.read_csv`. If for some reason the data is not available, you can always use the copy available locally in the `data` folder - just uncomment the line below that defines `base_url`:

```
# base_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/"
base_url = "../assets/data/estimation-covid-19/" # loading from disk
infected_dataset_url = base_url + "time_series_covid19_confirmed_global.csv"
recovered_dataset_url = base_url + "time_series_covid19_recovered_global.csv"
deaths_dataset_url = base_url + "time_series_covid19_deaths_global.csv"
countries_dataset_url = base_url + "UID_ISO_FIPS_LookUp_Table.csv"
```

Let's now load the data for infected individuals and see how the data looks like:

```
infected = pd.read_csv(infected_dataset_url)
infected.head()
```

图 4.2 读取数据文件

接下来，我们要根据教学需求编写相应的函数代码。我们现在读取了相应的数据集，接下来想查看中国地区的感染情况，我们可以定义一个函数，用于返回选取指定国家情况的 `DataFrame`。如图 4.3 所示。

From the names we can deduce that countries like Australia and China have more detailed breakdown by provinces. Let's look for information on China to see the example:

```
def filter(df, column_name, column_value):
    """
    Filters a pandas DataFrame based on a column value.

    Returns:
        pandas.DataFrame: The filtered DataFrame.
    """
    if df is None or not isinstance(df, pd.DataFrame) or df.empty:
        raise Exception("df is not a valid DataFrame")
    if column_name not in df.columns:
        raise Exception(f"{column_name} does not exist in df")
    return df[df[column_name] == column_value]

filter(infected, "Country/Region", "China")
```

图 4.3 Filter 函数展示

在这段函数代码中，遵循了 `black` 格式，明明采取 `snake` 命名法，以小写字母书写，采用下划线作间隔。作业中，括号部分内为需要输入的参数，其中 `df` 为 `DataFrame`，`column_name` 为 `df` 中指定的那一列，`column_value` 为想寻找的列名。函数部分开头添加了注释，进一步解释了函数的作用。`if` 语句判断情况，`raise` 语

句抛出异常。最终返回一个包含所求列的信息的 `DataFrame`。

由于设置了 `.ipynb` 文件代码块最后自动输出结果，所以在函数定义完后，通过使用，会自动生成结果显示。通过不断的调试，确保函数无误。进行该操作读取国家为“中国”地区的感染人数情况。

随后要为函数编写测试用例代码。测试用例代码要放在单独的测试类中，测试类要放在代码块中，采用 `pytest`、`ipytest`、`unittest` 库测试。测试用例代码要包含所有的参数传空、无效、还有传入 `None` 的情形。同时还要给出一组有效的情况。下面以该函数的 `df` 参数为例，详细解释测试用例的编写。部分测试用例函数界面如图 4.4 所示。

```
%%ipytest -qq

def create_test_df():
    return pd.DataFrame(
        {"numbers": [1, 2, 3, 4, 5], "bools": [False, False, True, True, True]}
    )

class TestFilter(unittest.TestCase):
    def test_filter_happy_case(self):
        # assign
        test_df = create_test_df()
        expected_result = pd.DataFrame({"numbers": [3], "bools": [True]})

        # act
        result = filter(test_df, "numbers", 3)

        # assert
        assert result.reset_index(drop=True).equals(expected_result)

    def test_filter_with_none_df(self):
        # act & assert
        with pytest.raises(Exception):
            filter(None, "numbers", 3)
```

图 4.4 测试用例函数部分展示

在编写测试类之前，要创造一些可控的数据帧，当作初始测试的数据帧。所以我们编写了 `create_test_df()` 函数完成这一作用。如图 4.5 所示。

```
def create_test_df():  
    return pd.DataFrame(  
        {"numbers": [1, 2, 3, 4, 5], "bools": [False, False, True, True, True]}  
    )
```

图 4.5 create_test_df()函数展示

编写测试类时，要用驼峰命名法来命名类名。并使用 `unittest.TestCase` 来进行类参数的传入。随后要编写一个期望函数并给出期望结果，在传入初始化的 `test_df` 后，我们要使其经函数 `filter()` 操作后与期望结果一致。即函数正确。例如我们现在想挑出 `test_df` 中“numbers”为 3 的数据，则我们需要使用 `filter()` 函数，向其传入 `test_df`、“numbers”、3 这三个数据。如果函数正常，那么我们最后会得到“numbers”列值为 3 的全部与之相关的信息。如图 4.6 所示。

```
class TestFilter(unittest.TestCase):  
    def test_filter_happy_case(self):  
        # assign  
        test_df = create_test_df()  
        expected_result = pd.DataFrame({"numbers": [3], "bools": [True]})  
  
        # act  
        result = filter(test_df, "numbers", 3)  
  
        # assert  
        assert result.reset_index(drop=True).equals(expected_result)
```

图 4.6 类与期望函数

对于参数 `df`，我们进行测试函数编写。如果传值为空，无效和 `None`，则会报出相应的错误。如图 4.7 所示。

```
def test_filter_with_none_df(self):
    # act & assert
    with pytest.raises(Exception):
        filter(None, "numbers", 3)

def test_filter_with_empty_df(self):
    # act & assert
    with pytest.raises(Exception):
        filter(pd.DataFrame(), "numbers", 3)

def test_filter_with_invalid_df_type(self):
    # act & assert
    with pytest.raises(Exception):
        filter(1, "numbers", 3)
```

图 4.7 参数 df 测试用例

在编写完成测试用例后，运行该代码块，将会检验函数正确性。如果函数是正确的。那么测试用例会全部通过。通过的函数会在下方以绿色小点形式说明通过了，右侧会显示[100%]的字样。如果没有全部通过的话，则会以“x”号进行标注。如第一个，第四个测试用例通过了，第二个第三个没有通过，图形为“..xx.”。全部通过的情形如图 4.8 所示。

```

1 %%pytest -qq
2
3 def create_test_df():
4     return pd.DataFrame({'numbers': [1, 2, 3, 4, 5], 'bools': [False, False, True, True, True]})
5
6 class TestFilter(unittest.TestCase):
7
8     def test_filter_with_happy_case(self):
9         # assign
10        test_df = create_test_df()
11        expected_result = pd.DataFrame({'numbers': [3], 'bools': [True]})
12
13        #act
14        result = filter(test_df, 'numbers', 3)
15
16        #assert
17        assert result.reset_index(drop=True).equals(expected_result)
18
19        # Test 2 - Filter on non-existing value
20    def test_filter_with_invalid_number(self):
21        # assign
22        test_df = create_test_df()
23
24        #act
25        with pytest.raises(Exception):
26            filter(test_df, 'numbers', 6)

```

[100%]

图 4.8 测试用例全部通过

随后，就可以进行挖空操作，使学生填写该函数。如图 4.9 所示。

From the names we can deduce that countries like Australia and China have more detailed breakdown by provinces. Let's look for information on China to see the example:

```

def filter(df, column_name, column_value):
    """
    Filters a pandas DataFrame based on a column value.

    Returns:
    | pandas.DataFrame: The filtered DataFrame.
    """
    if df is None or not isinstance(df, pd.DataFrame) or df.empty:
        raise Exception("df is not a valid DataFrame")
    if column_name not in df.columns:
        raise Exception(f"{column_name} does not exist in df")
    return df[df[_____] == _____]

filter(infected, "Country/Region", "China")

```

图 4.9 挖空后的函数

为了使学生能够在做不出来时得到反馈，必要的提示也是必不可少的。所以我们将测试用例折叠起来，在其下方加入提示并折叠。这样学生在做不出来时就可以点开提示观看。如图 4.10 和图 4.11 所示。



图 4.10 未点开时提示界面

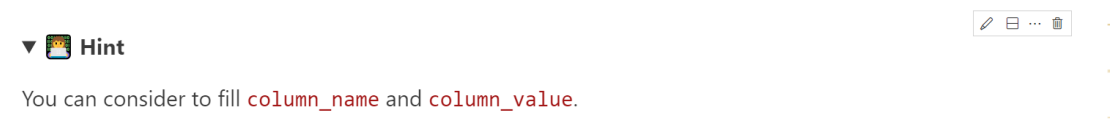


图 4.11 点开后提示界面

接下来我们继续进行引导。在直观看到了中国感染人员数据之后，我们设计直接呈现中国这一个整体的信息，而不是中国的各个地区诸如北京、上海、山东、江苏等省份的信息。即，我们想把所有省份的信息聚合在“中国”这一要素之中来整体展现。所以我们要引导学生掌握 `DataFrame.groupby()` 函数。该函数可以进行聚合操作。并且我们加以文字注释，确保学生了解所操作的对象。同时我们仍要编写测试用例和提示。完整的经设计后的待填函数如图 4.12 所示。

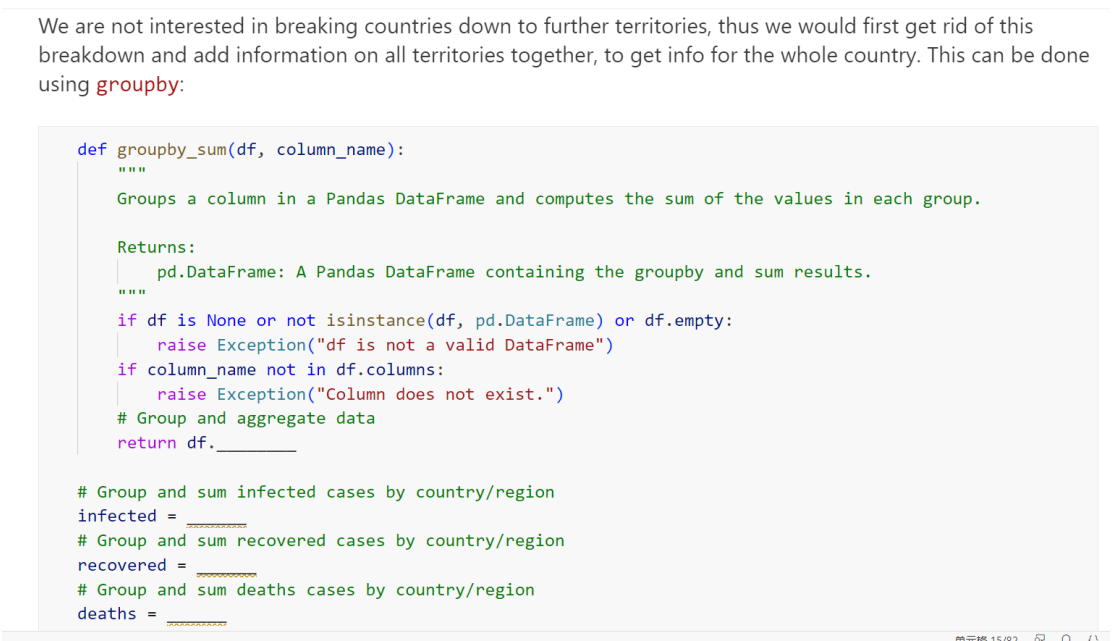


图 4.12 设计后的待填函数

当学生填完全部空后，如同 `filter()` 函数一样，能够从测试用例的通过情况看到自己正确与否。为了能够使学生有在线纠错的机会，作业每一个函数都配备了

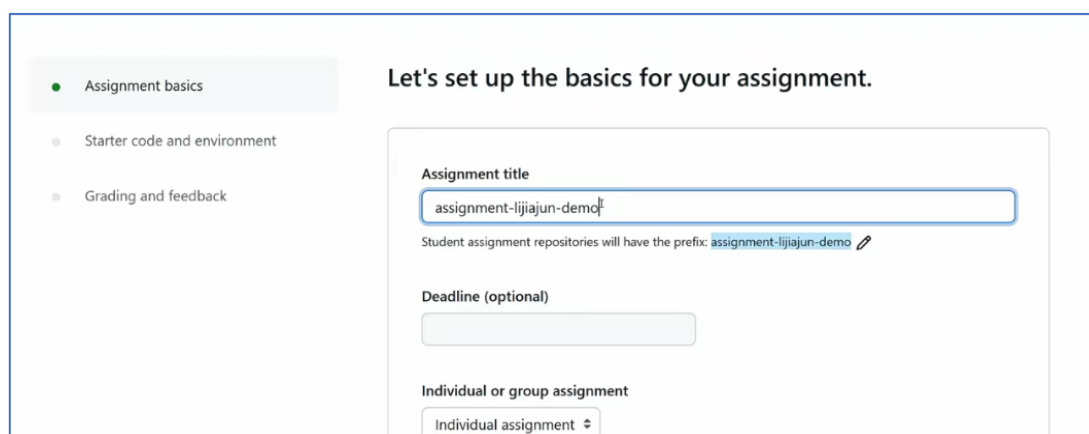
相关的测试代码和配套的测试用例与提示，确保这一个作业涉及的项目能够顺利推行。

最后，设计完该作业后将该作业通过 `git push` 操作推送到远程仓库中，就可以完成作业文档内容的制作与上传。

4.2 Github Classroom 使用展示

在该小节中，本文将展示 Github Classroom 的使用。目的是将该作业在 Github Classroom 上发布，并在学生完成后给予反馈。本小节展示 Github Classroom 的上传作业操作、查看作业操作，作业内容基于上一小节作业设计。

布置作业给学生，第一步是设置作业的名称等各种属性，如图 4.13 所示：



The screenshot shows the 'Assignment basics' setup page in Github Classroom. On the left, a sidebar lists three steps: 'Assignment basics' (highlighted with a green dot), 'Starter code and environment', and 'Grading and feedback'. The main content area is titled 'Let's set up the basics for your assignment.' It contains three form fields: 'Assignment title' with the value 'assignment-ljjajun-demo', 'Deadline (optional)' which is empty, and 'Individual or group assignment' with a dropdown menu set to 'Individual assignment'. Below the title field, a note states 'Student assignment repositories will have the prefix: assignment-ljjajun-demo' with an edit icon.

图 4.13 布置作业第一步

在布置作业的第二步，要设置一个远程仓库，如图 4.14 所示：

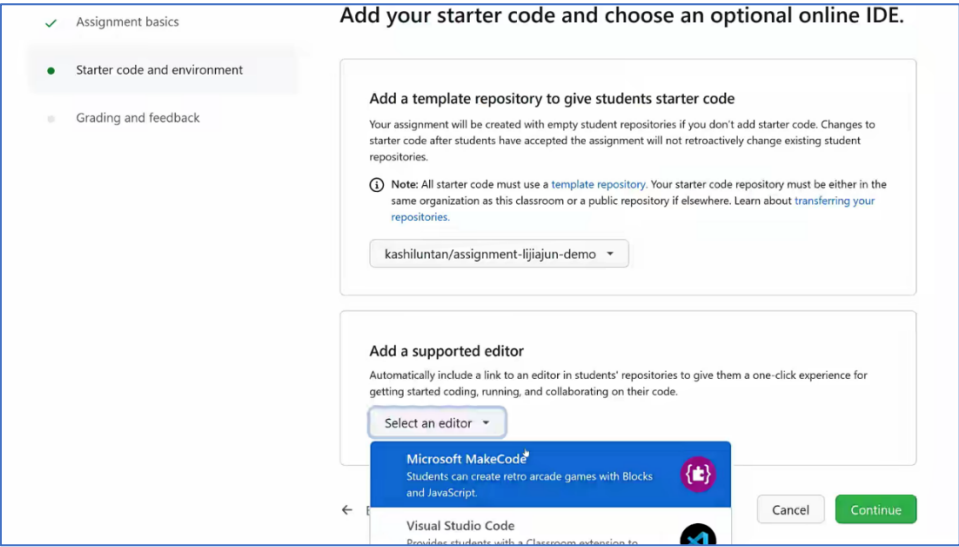


图 4.14 布置作业第二步

第三步，设置一些 test，如图 4.15 所示：

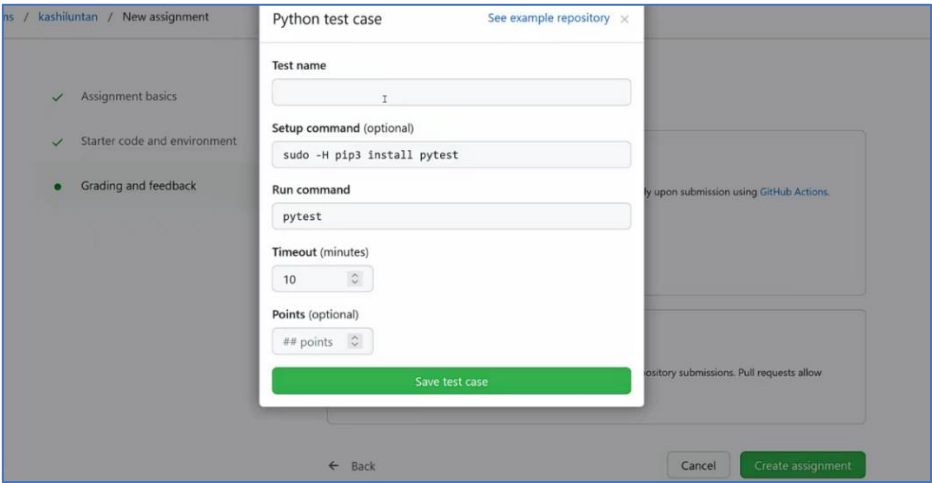


图 4.15 布置作业第三步

接下来查看学生提交的状态，如图 4.16 所示：

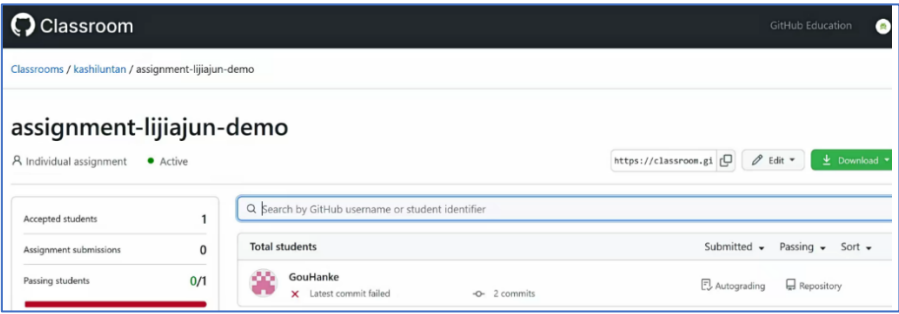


图 4.16 查看学生提交状态

该学生提交了作业，状态会变成 Submitted，但其作业运行失败了（有错误），利用 Github Classroom Feedback 可以随时对学生提出意见，建议。也可以随时进入学生提交作业后的的检验界面查看错误，如果学生的作业运行成功，则教师能在班级作业界面可以看到学生的作业成功提交，如图 4.17 所示：

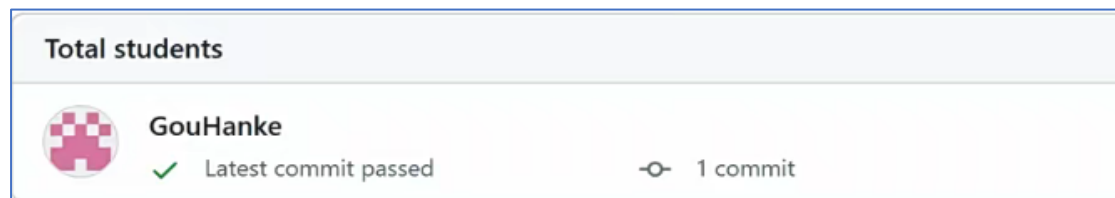


图 4.17 学生作业通过时班级作业界面

以上就是该平台利用 Github Classroom 的步骤及效果展示。

5 总结与展望

5.1 总结

本论文所做的测试驱动的数据科学教学平台的目标是通过以测试为导向的学习方法，帮助学生在数据科学领域建立坚实的技能 and 知识基础。该平台通过构建详细的测试用例、提供完整的代码模板和注释、搭建良好的在线测试环境、运用丰富的教学资源来扩充学生的学习体验，下面详细总结本毕设如何体现的以上四点：

1. 详细的测试用例：针对每个功能函数，编写了详尽的测试用例，涵盖不同的输入数据和期望的输出结果。这些测试用例为学生提供了明确的目标和参考，帮助他们理解所需实现的功能，并激发他们思考如何编写能够通过测试的代码。
2. 代码模板和注释：为了引导学生在正确的方向上进行编码，平台提供了包含函数框架和注释的代码模板。这些模板指导学生在适当的位置填写代码，帮助他们了解函数的输入和输出，以及实现功能所需的步骤和逻辑。通过模板的使用，学生可以更好地组织代码，并提高代码的可读性和可维护性。
3. 在线测试环境：平台搭建了一个在线的测试环境，可以自动运行学生提交的代码，并根据预先编写的测试用例检查函数的正确性。学生可以即时获取测试结果的反馈，了解代码是否通过了测试，并根据结果进行调试和修改。这种实时的测试反馈促使学生思考和解决问题，加深对数据科学技术的理解 and 应用。
4. 丰富的学习资源：为了支持学生的学习，平台提供了丰富的学习资源，包括教学视频、文档和实例代码。教学视频通过讲解概念和示范操作，帮助学生理解和掌握数据科学技术和工具。文档提供了进一步的解释和指导，帮助学生更好地理解测试驱动的学习方法和相关知识。实例代码展示了实际应用场景和解决方案，帮助学生将理论知识应用到实际问题中。

通过本论文所构建的测试驱动的数据科学教学平台，学生可以通过实践和反馈的循环，逐步提升他们在数据科学领域的技能和能力。这种以测试为导向的学习方法不仅能够培养学生的编码能力，还能够加深他们对数据科学概念和技术的理解，并培养解决问题的能力 and 思维方式。

5.2 展望

然而，目前所设计的测试驱动的数据科学教学平台仍有许多不足，比如测试驱动的教学平台的有效性高度依赖于编写全面而准确的测试用例，如果测试用例设计不合理或存在缺陷，可能会导致学生无法正确理解和应用数据科学技术；尽管测试驱动的教学平台可以帮助学生掌握基本的数据科学技术和编程概念，但在面对复杂的实际问题时可能存在局限性，某些数据科学任务可能需要更复杂的算法和工具，而平台可能无法提供全面的支持；缺乏灵活性和个性化定制：测试驱动的教学平台通常是按照固定的课程设计和教学进度进行开发。这可能限制了学生根据自身需求和兴趣进行个性化学习的能力。某些学生可能需要更深入或更广泛的学习内容，而平台可能无法满足他们的需求；测试驱动的教学平台通常需要学生具备一定的编程和数据科学基础，以便理解和应用所提供的代码框架和测试用例。对于初学者来说，可能需要一定的学习曲线来适应平台的使用方式和要求。

所以该测试驱动的数据科学教学平台有许多进一步发展和改进的可能性，可以优化测试用例的设计；可以着重提供实时引导和指导；可以增加更多的学习模块，扩展平台的内容，涵盖更多的数据科学领域和技术，如机器学习、深度学习、自然语言处理等，以满足不同层次和兴趣的学生需求；可以结合其他教学资源 and 社区，提供更全面的学习体验和支持；可以强化自动化评估：进一步改进在线测试环境，引入更全面的自动化评估机制，包括代码风格检查、性能评估等，提供更全面的反馈和指导。

通过不断改进和完善，测试驱动的数据科学教学平台有望成为一种有效的教学工具，帮助学生系统地学习和应用数据科学技术，培养数据科学的思维 and 实践能力，为他们在数据驱动的世界中取得成功打下坚实的基础。

参考文献

- [1] 朝乐门.数据科学理论与实践[M].北京:清华大学出版社, 2017:15.
- [2] Cleveland W S. Data science: an action plan for expanding the technical areas of the field of statistics[J]. International statistical review, 2001, 69(1): 21-26.
- [3] Mattmann C A. Computing: A vision for data science[J]. Nature, 2013, 493(7433): 473-475.
- [4] Schutt R, O'Neil C. Doing data science: Straight talk from the frontline[M]. "O'Reilly Media, Inc.", 2013.
- [5] 黄丽仟,玉洪荣,刘鹏.基于超星学习通平台线上教学模式与传统教学模式的比较研究[J].蛇志,2022,34(04):577-580.
- [6] 朱毅,李云,强继朋,等. 数据挖掘课程教学模式改革与探索[J]. 科教文汇,2021(18):108-109.
- [7] <https://open-academy.github.io/machine-learning>
- [8] https://github.com/Nyandwi/machine_learning_complete
- [9] Glassey, Richard. "Adopting Git/Github within teaching: A survey of tool support." Proceedings of the ACM Conference on Global Computing Education. 2019.
- [10] 刘征, 郭君红.探究 Jupyter Notebook 编程工具在项目式学习中的应用[J]. 中国现代教育装备, 2020, No.342(14):29-30+33.
- [11] Naur P.Concise survey of computer methods[M].Studentlitteratur AB:1974.
- [12] 朝乐门.数据科学[M].北京:清华大学出版社, 2016.
- [13] Conway D.The data science venn diagram[OL].[2016-9-30].<http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>.
- [14] 桂思思, 任珂, 王玉琦等.基于 MOOC 平台的国内外数据科学课程建设对比分析及经验启示[J].图书情报知识, 2018, No.183(03):119-128.
- [15] 朝乐门, 卢小宾.数据科学及其对信息科学的影响[J].情报学报, 2017, 36(08):761-771.
- [16] 杨溢. 基于分类算法的研究生数据分析系统的设计与实现[D].北京交通大

- 学, 2018.
- [17] 翟高粤.基于 Python 的数据分析概述[J].甘肃科技纵横, 2018, 47(11):5-7+26.
- [18] 朱辉生. 数据库原理及应用实验教程[M].南京大学出版社:, 202106.235.
- [19] 华振宇.两个 Python 第三方库: Pandas 和 NumPy 的比较[J].电脑知识与技术, 2023, 19(01):71-73+76.
- [20] 何春燕, 王超宇.基于 python+pandas 的数据分析处理应用[J].数码世界, 2018, No.153(07):386.
- [21] <https://github.com/jupyter4edu/jupyter-edu-book>
- [22] 杨军, 张岳, 刘燕峰. 基于 Python 语言的数据挖掘课程的建设与研究[J]. 科技风, 2021(14): 80—82.
- [23] 朱毅, 李云, 强继朋, 等. 数据挖掘课程教学模式改革与探索[J]. 科教文汇(下旬刊), 2021(06): 108—109.
- [24] 王茂发, 王子民, 汪华登, 等. 基于 Jupyter 的数据挖掘课程建设与研究[J]. 电脑与电信, 2021(7):12-16.
- [25] 马伟良. Jupyter Notebook 平台在 Python 教学中的应用[J]. 数字通信世界, 2022(5):82-84.
- [26] 刘征, 郭君红.探究 Jupyter Notebook 编程工具在项目式学习中的应用[J]. 中国现代教育装备, 2020, No.342(14):29-30+33.
- [27] <https://docs.github.com/en/education/manage-coursework-with-github-classroom/get-started-with-github-classroom>
- [28] 李一风.基于 pytest 和 JMeter 的自动化测试系统设计和实现[J].智能计算机与应用, 2019, 9(01):277-279+283.
- [29] <https://jupyterbook.org/en/stable/start/overview.html>
- [30] <https://www.atlassian.com/git/tutorials/git-ssh>

致 谢

一个人的成长，离不开无数人的支持和帮助。在我撰写这篇论文的过程中，我深深感受到了亲人、朋友和老师们的关爱和支持。在这里，我想借此机会向所有给予我帮助和鼓励的人们表达我最衷心的感谢之情。

首先，我要感谢我的导师杜金鑫先生，和我的恩师陈伦德先生，还有张旗先生，他们在我学术道路上给予了我无私的指导和悉心的培养。他们耐心地解答我的问题，激发了我对研究的热情，并引导我探索新的领域。他们的经验和专业知识为我的研究提供了宝贵指引，使我能够更好理解和应用所学的知识。

我还要感谢我的爸爸，妈妈，我的家人们，朋友们，他们一直以来的支持和鼓励是我前进的动力。他们在我遇到困难时给予我温暖和鼓励，无论是在学业上还是生活中，他们的支持都是我坚强前行的动力。

此外，我还要感谢与我共事的同学们，我的室友冯周，吴逸钊，还有刘宇昕。他们在研究中与我分享了自己的经验和见解，我们一起合作、讨论问题、解决难题。他们的智慧和团队合作精神使得我的研究更加丰富和深入。我们在一起生活了四年，在此离别之际向他们表示由衷的祝福。

还后，我要感谢所有曾经帮助过我的人们，无论是给予我建议、提供技术支持还是提供资源和文献，你们的帮助对我来说都是非常宝贵的。没有你们的支持，我将无法完成这篇论文的撰写。

最后，我要感谢我家的小狗，每天照在我身上的阳光，我读的文学，我看的剧等一切温暖的生命和事物。在我的大学四年中给予了我无限的动力。

再次向所有给予我帮助和支持的人表示衷心的感谢。你们的支持和鼓励让我更加坚定地追求知识，为人生的道路增添了色彩。我的本科结束了，但我的人生才刚刚开始。愿我们共同努力，共同成长，为更美好的未来而努力。感谢你们！

王宏乾

上海大学

2023 年 05 月 31 日

