Machine Learning with PySpark:
With Natural Language Processing and
Recommender Systems

PySpark

机器学习、自然语言处理与推荐系统

[印] 普拉莫德•辛格(Pramod Singh) 著
蒲 成 译

Apress®

清华大学出版社

Machine Learning with PySpark:
With Natural Language Processing and Recommender Systems

PySpark机器学习、自然语言处理与推荐系统

使用PySpark构建机器学习模型、自然语言处理应用程序以及推荐系统，从而应对各种业务挑战。本书首先介绍Spark的基础知识及其演进，然后讲解使用PySpark构建传统机器学习算法以及自然语言处理和推荐系统的全部知识点。

《PySpark机器学习、自然语言处理与推荐系统》阐释如何构建有监督机器学习模型，比如线性回归、逻辑回归、决策树和随机森林，还介绍了无监督机器学习模型，比如K均值和层次聚类。本书重点介绍特征工程，以便使用PySpark创建有用的特征，从而训练机器学习模型。自然语言处理的相关章节将介绍文本处理、文本挖掘以及用于分类的嵌入。

在阅读完本书后，读者将了解如何使用PySpark的机器学习库构建和训练各种机器学习模型。此外，还将熟练掌握相关的PySpark组件，比如数据获取、数据处理和数据分析，通过使用它们开发数据驱动的智能应用。

主要特色

- ◇ 构建一系列有监督和无监督机器学习算法
- ◇ 使用Spark MLlib库实现机器学习算法
- ◇ 使用Spark MLlib库开发推荐系统
- ◇ 处理与特征工程、分类平衡、偏差和方差以及交叉验证有关的问题，以便构建最优的拟合模型

读者对象

数据科学家、机器学习工程师。

Apress®
www.apress.com

ISBN 978-7-302-54090-
9 787302 540908

定价：59.00元

PySpark 机器学习、自然语言处理与推荐系统

[印] 普拉莫德·辛格(Pramod Singh) 著
蒲 成 译

清华大学出版社
北 京

Machine Learning with PySpark: With Natural Language Processing and Recommender Systems

Pramod Singh

EISBN：978-1-4842-4130-1

北京市版权局著作权合同登记号　图字：01-2019-5767

图书在版编目(CIP)数据

PySpark 机器学习、自然语言处理与推荐系统 / (印)普拉莫德 • 辛格 著；蒲成 译. —北京：清华大学出版社，2020

书名原文：Machine Learning with PySpark: With Natural Language Processing and Recommender Systems

ISBN 978-7-302-54090-8

Ⅰ. ①P… Ⅱ. ①普… ②蒲… Ⅲ. ①机器学习 ②自然语言处理 Ⅳ. ①TP181 ②TP391

中国版本图书馆 CIP 数据核字(2019)第 241996 号

责任编辑：王　军
装帧设计：孔祥峰
责任校对：成凤进
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
印 装 者：三河市吉祥印务有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　印　　张：10.75　　字　　数：235 千字
版　　次：2020 年 1 月第 1 版　　印　　次：2020 年 1 月第 1 次印刷
定　　价：59.00 元

产品编号：084081-01

译 者 序

随着人工智能的兴起，与之相关的知识和技术越来越受大众所关注，神经网络、机器学习、深度学习、自然语言处理等专业术语也开始为大家所广泛探讨。现在市面上可用的大数据处理分析甚或人工智能框架很多，所以对于刚入门或者想要入门的新手而言，选择一款合适的框架作为起步学习之用是非常重要的。

作为目前处理和使用大数据的使用最广泛的框架之一，Spark 已经被各大企业投入实际应用中。Spark 是在 Scala 中设计的，以强大的处理速度和缓存能力见长，不过对于程序员来说，考虑到语法和标准库，Python 相对来说更容易学习，而且 Python 是数据分析、机器学习等方面使用最广泛的编程语言之一。因此，为了支持 Spark 和 Python，Apache Spark 社区发布了 PySpark，也就是说，PySpark 是 Spark 的 Python Shell。

本书首先将介绍机器学习和 Spark，然后会结合大数据进一步详细讲解机器学习，进而通过示例展示如何使用 PySpark 构建推荐系统和 NLP。虽然是一本与机器学习有关的专业技术书籍，但本书内容浅显易懂，对于刚开始接触 PySpark 并且想要系统地理解 PySpark 基础知识结构以及相关算法的读者而言，本书将会是很好的入门指南。

本书不仅涵盖与 PySpark 组件相关的知识，比如数据获取、数据处理和数据分析等，还讲解如何使用 PySpark 构建基础的机器学习算法和模型。相信在阅读完本书后，读者将获悉如何将 PySpark 用于工作实践之中，并且可以用来构建专业的人工智能应用。

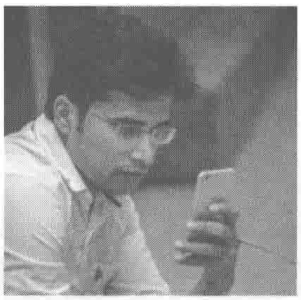
在此要特别感谢清华大学出版社的编辑们，在本书翻译过程中他们提供了颇有助益的帮助，没有他们的热情付出，本书将难以付梓。

由于译者水平有限，难免会出现一些错误或翻译不准确的地方，如果读者能够指出并勘正，译者将不胜感激。

译者

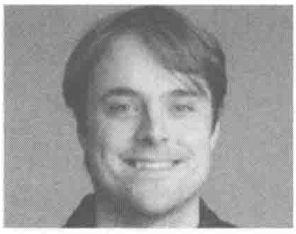
2019 年 6 月

作 者 简 介



Pramod Singh 是 Publicis.Sapient 公司数据科学部门的经理，目前正作为数据科学跟踪负责人与梅赛德斯奔驰的一个项目进行合作。他在机器学习、数据工程、编程，以及为各种业务需求设计算法方面拥有丰富的实践经验，领域涉及零售、电信、汽车以及日用消费品等行业。他在 Publicis.Sapient 主导了大量应对机器学习和 AI 的战略计划。他在孟买大学获得了电气与电子工程的学士学位，并且在印度共生国际大学获得了 MBA 学位(运营&财务)，还在 IIM – Calcutta(印度管理学院加尔各答分校)获得了数据分析认证。在过去八年中，他一直在跟进多个数据项目。在大量客户项目中，他使用 R、Python、Spark 和 TensorFlow 应用机器学习和深度学习技术。他一直是各重大会议和大学的演讲常客。他会在 Publicis.Sapient 举办数据科学聚合并且定期出席关于 ML 和 AI 的网络研讨会。他和妻子以及两岁的儿子居住在班加罗尔。闲暇的时候，他喜欢弹吉他、写代码、阅读以及观看足球比赛。

技术编辑简介



Leonardo De Marchi 拥有人工智能专业的硕士学位，并且曾经作为数据科学家服务于体育行业，客户包括纽约尼克斯队和曼联，也曾与 Justgiving 这样的大型社交网络进行过合作。

他如今是 Badoo 的首席数据科学家，Badoo 是拥有超过 3 亿 6 千万用户的全球最大交友网站。他也是 ideai.io 的首席执教官，ideai.io 是一家专门从事深度学习和机器学习训练的公司，并且是欧盟委员会的供应商。

致　谢

在编写本书的过程中，如果没有一些人的帮助，那么本书将无法顺利付梓。在我的人生当中，我多次听到过“说易行难”这句话，在本书编写期间我真切地体会到了其中的含义。坦率地说，一开始我对于编写本书是非常有信心的，但实际上在编写期间，这件事开始变得困难起来。这真的很讽刺，因为在我思考内容时，我的脑海中是非常清晰的，不过当我开始动手在纸上写下这些内容时，突然就会开始感到困惑。在此期间我内心十分纠结，不过这段时期对我个人而言不仅仅是一次革新。首先，我必须感谢我生命中最重要的人——我挚爱的妻子 Neha，在此期间她给予我无私的支持，并且做出很多牺牲以确保我能完成本书的编写。

我想要感谢 Suresh John Celestin，他给予我充分的信任，并且为我提供编写本书的机会。Aditee Mirashi 是可以与之协作的最佳编辑之一。她给予我极大的支持，并且总是能够及时回应我的所有请求。试想一下，对于一个正编写自己第一本书的人而言，我必定有大量的问题想要咨询。我要特别感谢 Matthew Moodie，他专门花时间阅读了每一章的内容，并且提出了许多有意义的建议。谢谢 Matthew，我真的很感激。我希望感谢的另一个人是 Leonardo De Marchi，他耐心地检查了本书中的每一行代码，并且检查了每个示例是否恰当。谢谢 Leo，感谢你的反馈和鼓励。你的帮助对于我和这本书而言非常关键。我还想要感谢导师们，你们不断地推动着我追寻梦想。Alan Wexler、Dr. Vijay Agneeswaran、Sreenivas Venkatraman、Shoaib Ahmed 和 Abhishek Kumar，谢谢你们为我花费的时间和精力。

最后，我无比感激我的儿子 Ziaan 以及我的父母，他们给予我无尽的爱，并且无论环境如何，都给予我毫无保留的支持。与你们在一起才让我感受到生命如此美好。

前　言

在开始编写本书之前，我曾经问过自己一个问题：是否有必要写一本关于机器学习的书？我的意思是，市面上已经有很多关于这一主题的书籍。为了找到答案，我花费了大量时间进行思考，不久之后，一些规律开始浮现在我的脑海中。目前关于机器学习的书籍都过于关注细节而缺乏一种顶层概览。这些书刚开始的内容真的很简单，不过几章之后，随着内容变得过于深入，就会让读者感到难以继续阅读下去。因而，读者就会由于放弃阅读而无法从书中汲取足够的知识。这就是我想要编写本书的原因，本书揭示使用机器学习的不同方式，虽然不会过于深入细节，不过也会让读者了解全新构建 ML 模型所需的完整方法论。另一个显而易见的问题就是：为何要使用 PySpark 进行机器学习？找到这个问题的答案并没有花费我太长时间，因为我是一位拥有实践经验的数据科学家并且非常清楚处理数据的人所面临的挑战。大多数的包或模块通常在使用方面都是受限的，因为它们在单台机器上处理数据。如果 ML 模型的目的不是处理大数据并且最终数据处理本身需要变得快速且可扩展，那么从开发环境迁移到生产环境会变成一场噩梦。出于所有这些原因，编写这本关于使用 PySpark 进行机器学习的书籍就是完全合理的，以便让读者能够理解从大数据角度使用机器学习的处理过程。

现在我来谈谈《PySpark 机器学习、自然语言处理与推荐系统》这本书的核心内容。这本书分为三大部分。第一部分将介绍机器学习和 Spark；第二部分会使用大数据详细讲解机器学习；第三部分会展示如何使用 PySpark 构建推荐系统和 NLP。这本书可能也与数据分析师和数据工程师有关，因为它还介绍了使用 PySpark 处理大数据的步骤。想要切入数据科学和机器学习领域的读者会发现本书更易于入门，并且后续能够逐步学习掌握更复杂的知识。书中的案例研究和示例会让本书内容以及基础概念的学习理解变得非常容易。此外，目前市面上关于 PySpark 的书籍非常少，而这本书必定会让读者汲取到一些新的知识。本书的优点在于，以浅显易懂的方式阐释机器学习算法，并且针对使用 PySpark 构建这些算法提供一种切实可行的方法。

我将自己的所有经验和所掌握的知识都融入本书之中，并且我认为它们确实与那些现在寻求应对实际挑战的企业紧密相关。我希望读者能从本书中汲取到一些有用的知识。

目　录

第 1 章　数据革命 …… 1
1.1　数据生成 …… 1
1.2　Spark …… 2
1.2.1　Spark Core …… 3
1.2.2　Spark 组件 …… 4
1.3　设置环境 …… 5
1.3.1　Windows …… 5
1.3.2　iOS …… 6
1.4　小结 …… 7

第 2 章　机器学习简介 …… 9
2.1　有监督机器学习 …… 10
2.2　无监督机器学习 …… 12
2.3　半监督机器学习 …… 14
2.4　强化学习 …… 14
2.5　小结 …… 15

第 3 章　数据处理 …… 17
3.1　加载和读取数据 …… 17
3.2　添加一个新列 …… 20
3.3　筛选数据 …… 21
3.3.1　条件 1 …… 21
3.3.2　条件 2 …… 22
3.4　列中的非重复值 …… 23
3.5　数据分组 …… 23
3.6　聚合 …… 25
3.7　用户自定义函数(UDF) …… 26
3.7.1　传统的 Python 函数 …… 26
3.7.2　使用 lambda 函数 …… 27
3.7.3　Pandas UDF(向量化的 UDF) …… 28
3.7.4　Pandas UDF(多列) …… 29
3.8　去掉重复值 …… 29
3.9　删除列 …… 30
3.10　写入数据 …… 30
3.10.1　csv …… 31
3.10.2　嵌套结构 …… 31
3.11　小结 …… 31

第 4 章　线性回归 …… 33
4.1　变量 …… 33
4.2　理论 …… 34
4.3　说明 …… 41
4.4　评估 …… 42
4.5　代码 …… 43
4.5.1　数据信息 …… 43
4.5.2　步骤1：创建 SparkSession对象 …… 44
4.5.3　步骤2：读取数据集 …… 44
4.5.4　步骤3：探究式数据分析 …… 44
4.5.5　步骤4：特征工程化 …… 45
4.5.6　步骤5：划分数据集 …… 47
4.5.7　步骤6：构建和训练线性回归模型 …… 47

4.5.8 步骤 7：在测试数据上评估线性回归模型 …… 48
4.6 小结 …… 48

第 5 章 逻辑回归 …… 49
5.1 概率 …… 49
5.1.1 使用线性回归 …… 50
5.1.2 使用 Logit …… 53
5.2 截距(回归系数) …… 54
5.3 虚变量 …… 55
5.4 模型评估 …… 56
5.4.1 正确的正面预测 …… 56
5.4.2 正确的负面预测 …… 57
5.4.3 错误的正面预测 …… 57
5.4.4 错误的负面预测 …… 57
5.4.5 准确率 …… 57
5.4.6 召回率 …… 57
5.4.7 精度 …… 58
5.4.8 F1 分数 …… 58
5.4.9 截断/阈值概率 …… 58
5.4.10 ROC 曲线 …… 58
5.5 逻辑回归代码 …… 59
5.5.1 数据信息 …… 59
5.5.2 步骤 1：创建 Spark 会话对象 …… 60
5.5.3 步骤 2：读取数据集 …… 60
5.5.4 步骤 3：探究式数据分析 …… 60
5.5.5 步骤 4：特征工程 …… 63
5.5.6 步骤 5：划分数据集 …… 68
5.5.7 步骤 6：构建和训练逻辑回归模型 …… 69
5.5.8 训练结果 …… 69
5.5.9 步骤 7：在测试数据上评估线性回归模型 …… 70
5.5.10 混淆矩阵 …… 71
5.6 小结 …… 72

第 6 章 随机森林 …… 73
6.1 决策树 …… 73
6.1.1 熵 …… 75
6.1.2 信息增益 …… 76
6.2 随机森林 …… 78
6.3 代码 …… 80
6.3.1 数据信息 …… 80
6.3.2 步骤 1：创建 SparkSession 对象 …… 81
6.3.3 步骤 2：读取数据集 …… 81
6.3.4 步骤 3：探究式数据分析 …… 81
6.3.5 步骤 4：特征工程 …… 85
6.3.6 步骤 5：划分数据集 …… 86
6.3.7 步骤 6：构建和训练随机森林模型 …… 87
6.3.8 步骤 7：基于测试数据进行评估 …… 87
6.3.9 准确率 …… 89
6.3.10 精度 …… 89
6.3.11 AUC 曲线下的面积 …… 89
6.3.12 步骤 8：保存模型 …… 90
6.4 小结 …… 90

第 7 章 推荐系统 …… 91
7.1 推荐 …… 91
7.1.1 基于流行度的 RS …… 92
7.1.2 基于内容的 RS …… 93
7.1.3 基于协同过滤的 RS …… 95
7.1.4 混合推荐系统 …… 103
7.2 代码 …… 104
7.2.1 数据信息 …… 105
7.2.2 步骤 1：创建 SparkSession 对象 …… 105
7.2.3 步骤 2：读取数据集 …… 105
7.2.4 步骤 3：探究式数据分析 …… 105

7.2.5 步骤 4：特征工程 108
7.2.6 步骤 5：划分数据集 109
7.2.7 步骤 6：构建和训练推荐系统模型 110
7.2.8 步骤 7：基于测试数据进行预测和评估 110
7.2.9 步骤 8：推荐活动用户可能会喜欢的排名靠前的电影 111
7.3 小结 114

第 8 章 聚类 115
8.1 初识聚类 115
8.2 用途 117
8.2.1 K-均值 117
8.2.2 层次聚类 127
8.3 代码 131
8.3.1 数据信息 131
8.3.2 步骤 1：创建 SparkSession 对象 131
8.3.3 步骤 2：读取数据集 131
8.3.4 步骤 3：探究式数据分析 131
8.3.5 步骤 4：特征工程 133
8.3.6 步骤 5：构建 K 均值聚类模型 133
8.3.7 步骤 6：聚类的可视化 136
8.4 小结 137

第 9 章 自然语言处理 139
9.1 引言 139
9.2 NLP 涉及的处理步骤 139
9.3 语料 140
9.4 标记化 140
9.5 移除停用词 141
9.6 词袋 142
9.7 计数向量器 143
9.8 TF-IDF 144
9.9 使用机器学习进行文本分类 145
9.10 序列嵌入 151
9.11 嵌入 151
9.12 小结 160

第 1 章

数 据 革 命

在理解 Spark 之前，当务之急是要理解当今我们正在见证的数据洪流背后的原因。早些年代，数据是由员工生成或累积下来的，因此只有公司职员才会将数据输入系统中，并且这些数据点的范围都很窄，仅涉及一些领域。之后，互联网时代来临，每一个使用互联网的人都能轻易获取信息。如今，用户已经有能力输入和生成自己的数据了。这是一次巨大的转变，因为互联网用户的数量呈指数增长，并且由这些用户创造的数据增长量甚至更高。例如：登录/注册表单允许用户填写他们自己的详细信息，将照片和视频上传到各种社交平台，这就会产生海量的数据以及处理大量数据所需的快速且可伸缩的框架。

1.1 数据生成

如今，这些数据的生成已经增长到一个新的水平，因为许多机器都在产生和累积数据，如图 1-1 所示。我们周围的每一台设备都在捕获数据，例如汽车、建筑物、手机、手表、飞机发动机。这些设备都内置了多个监控传感器并且每秒都会记录数据。这部分数据的量级甚至比用户生成的数据还要高。

图 1-1　数据革命

早些时候，当数据仍旧处于企业级应用时，关系型数据库就能够很好地应对系统需要了，但由于过去几十年中数据量呈指数增长，大数据的处理已经发生了一种结构性的变化，而这种变化正是由 Spark 的诞生而引发的。传统上讲，我们习惯于

获取数据并且将它们放入处理器中以进行处理，不过现在，由于数据量过大，处理器已经无法应对了。目前，我们应用了多处理器机制来处理数据。这被称为并行处理，因为同一时间会在多个位置处理数据。

我们来看一个示例，以便理解并行处理这一概念。假设在某条高速公路上，只有一个收费站，而每一辆车都必须排成一行以便通过该收费站，如图 1-2 所示。如果平均每辆车需要花费 1 分钟通过该收费站，那么八辆车总共就需要 8 分钟，100 辆车就需要花费 100 分钟。

图 1-2　单线程处理

但是想象一下，如果这条高速公路上有八个收费站而不是一个，并且车辆可以通过其中任意一个收费站进行收费，那么所有八辆车通过收费站的总时长只需要 1 分钟就够了，因为其中没有依赖关系了，如图 1-3 所示，收费操作已经并行化了。

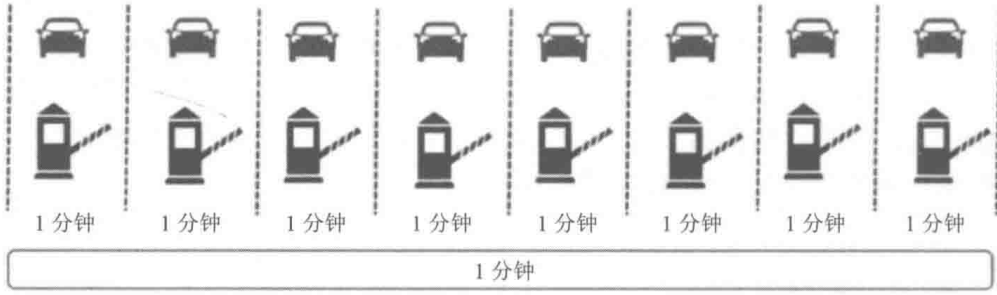


图 1-3　并行处理

并行或分布式计算遵循类似的原则，因为会并行处理任务并且在处理结束时聚积最终结果。Spark 是一个框架，它可以采用并行处理的方式高速应对海量数据，并且它是一种健壮的机制。

1.2　Spark

Apache Spark 源自 2009 年美国加州大学伯克利分校 AMPLab 的一个研究项目并且于 2010 年初开源，如图 1-4 所示。自那时以来，Spark 一直处于高速发展之中。2016 年，Spark 发布了用于深度学习的 TensorFrames。

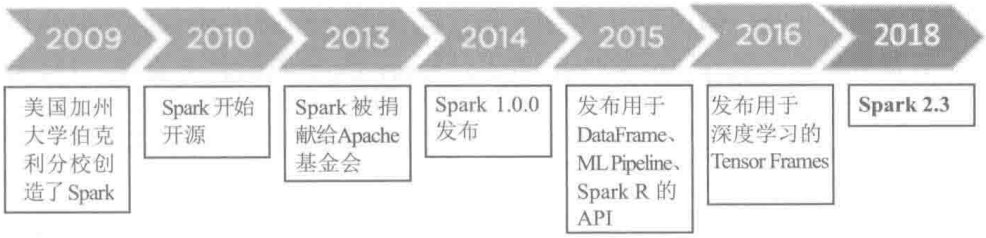


图 1-4 Spark 的发展历程

在底层，Spark 使用名为 RDD(Resilient Distributed Dataset，弹性分布式数据集)的一种独特的数据结构。弹性的含义在于，在执行处理期间，数据结构具有重建任意时点数据流的能力。因此，RDD 会使用最后一个时点的数据流创建一个新的 RDD，并且就算出现任何错误，也总是拥有重构的能力。它们是不可变的，因为原始的 RDD 会保持不变。Spark 由于是一种分布式框架，因此它是基于主节点和工作节点的设置来运行的，如图 1-5 所示。执行任意活动的代码一开始都是在 Spark 驱动程序上编写的，之后会共享到实际存留数据的各个工作节点。每个工作节点都包含一些执行器，它们将实际执行代码。集群管理器会持续检查各工作节点的可用性以便下一次分配任务。

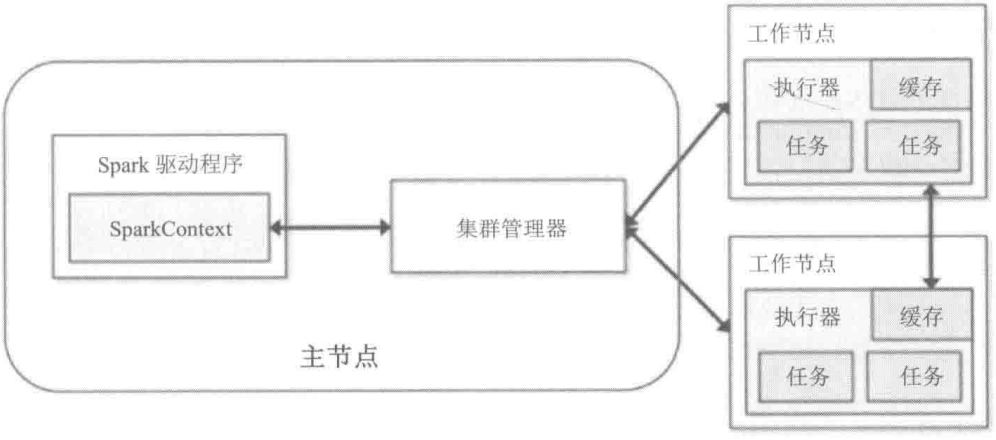


图 1-5 Spark 运行机制

Spark 大受欢迎的主要原因在于，它实际上非常适用于数据处理、机器学习以及流式数据；并且处理速度相对而言非常快，因为它所进行的都是内存中的计算。由于 Spark 是一种通用数据处理引擎，因此可以很容易地将其与各种数据源结合使用，例如 HBase、Cassandra、Amazon S3、HDFS 等。Spark 为用户提供了可在其上使用的四种语言选项：Java、Python、Scala 和 R。

1.2.1 Spark Core

Spark Core 是 Spark 最基础的组成部分，如图 1-6 所示，它是 Spark 高级功能特

性的支柱。Spark Core 使得驱动并行和分布式数据处理的内存中计算成为可能。Spark 的所有特性都构建在 Spark Core 之上。Spark Core 负责任务管理、I/O 操作、容错以及内存管理等。

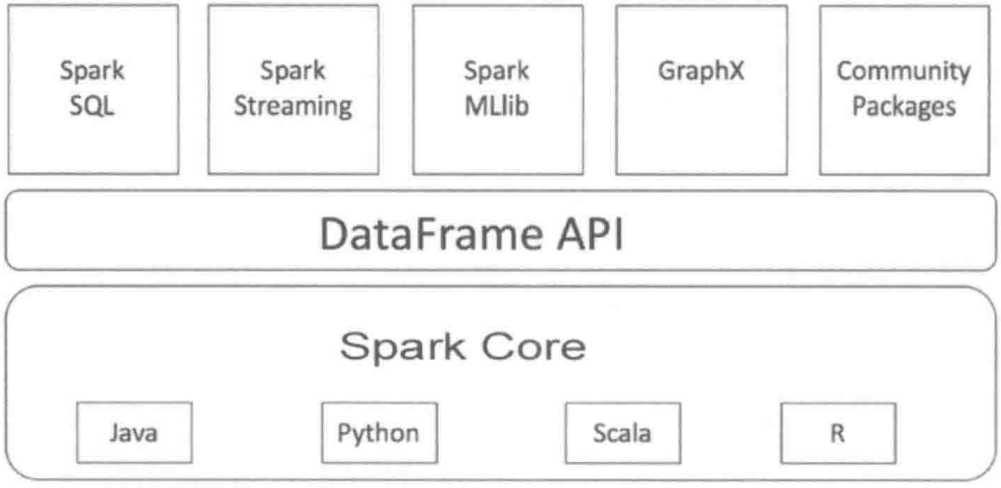


图 1-6 Spark 架构

1.2.2 Spark 组件

下面介绍 Spark 组件。

1. Spark SQL

这一组件主要应对的是结构化数据处理。其关键理念在于，获取与数据结构有关的更多信息以便执行额外的优化。它可以被视作一个分布式 SQL 查询引擎。

2. Spark Streaming

这一组件的任务是，以一种可伸缩且可容错的方式处理实时的流式数据。它使用小批量处理的方式读取和处理传入的数据流。它会创建小批量的流式数据、执行批处理，并且将之传递到一些文件存储或实时仪表盘。Spark Streaming 可以从多个源中摄取数据，例如 Kafka 和 Flume。

3. Spark MLlib

这一组件用于以分布式方式构建基于大数据的机器学习模型。当数据量很大时，使用 Python 的 scikit-learn 库构建 ML(机器学习，Machine Learning)模型的传统技术面临着极大挑战，而 Spark MLlib 旨在以大规模方式提供特征工程和机器学习。Spark Mllib 的大部分算法实现都是为了用于分类、回归分析、聚类分析、推荐系统和自然语言处理。