

# 数据工程师成长指南

---

## 数据工程师成长指南

### 一. 数据工程与数据科学的关系

### 二. 数据工程与云计算的关系

### 三. 数据工程成长路径

#### 3.1 操作系统

#### 3.2 编程语言

##### 3.2.1 Python语言

##### 3.2.2 Java语言

##### 3.2.3 Scala语言

##### 3.2.4 Go语言

##### 3.2.5 Rust语言

##### 3.2.6 C++语言

#### 3.3 云计算

#### 3.4 分布式存储

##### 3.4.1 HDFS

#### 3.5 分布式协调

##### 3.5.1 Apache Zookeeper

#### 3.6 消息队列

##### 3.6.1 Apache Kafka

##### 3.6.2 ActiveMQ

#### 3.7 SQL数据库

#### 3.8 交互式数据分析

##### 3.8.1 Apache Hive

##### 3.8.2 Apache Pig

##### 3.8.3 Apache Kylin

#### 3.9 分布式计算

##### 3.9.1 离线批处理

##### 3.9.1.1 MapReduce

##### 3.9.2 流式计算

##### 3.9.2.1 Apache Storm

##### 3.9.2.2 Amazon Kinesis

##### 3.9.2.3 Apache Samza

##### 3.9.2.4 Apache Flink

##### 3.9.3 内存计算

##### 3.9.3.1 Apache Spark

#### 3.10 NoSQL数据库

##### 3.10.1 KV数据库

##### 3.10.1.1 Redis

##### 3.10.2 列式存储数据库

##### 3.10.2.1 HBASE

##### 3.10.2.2 Cassandra

##### 3.10.3 文档数据库

##### 3.10.3.1 MongoDB

##### 3.10.3.2 Couchbase

- 3.11 资源调度
- 3.12 日志采集
- 3.13 RPC框架
- 3.14 综合资料
  - 3.14.1 Airbnb数据工程入门
  - 3.14.2 O'Reilly免费数据工程电子书套件
  - 3.14.3 数据工程综合课程
  - 3.14.4 24个终极数据科学项目

## 五. 数据工程培养计划

## 四. 数据工程师必备前置知识

- 4.1 分布式系统基础知识
- 4.2 机器学习基础知识

## 六. 数据工程相关认证考试

- 6.1 谷歌认证专家
- 6.2 IBM认证数据工程师
- 6.3 Cloudera的CCP数据工程师

## 七. 参考文献



# Data Engineering

在大数据领域发展的职业路径中，有一套二元对立的职业技能发展体系：数据科学和数据工程。二者都有各自的侧重点，但它们之间相互依赖，任何一方的发展都离不开另一方。数据工程为数据科学搭建各种架构的数据密集型应用；数据科学在数据工程的帮助下完成最后的建模——评估——发布，形成最终的数据产品。任何一个体系都有博大精深的技能需要掌握，它们所关注的点并不相同，要想精通两个体系，做到面面俱到是不现实的。因此我们应该把它们区分开，先弄清楚每个体系的侧重点，才能根据个人情况有的放矢的训练和培养相关技能。然后将相关人才有机组合起来，形成数据科学团队，发挥强大的战斗力。这篇文章首先想要理清的是数据工程和数据科学之间的区别与联系，内涵与外延，然后在这个基础上整理和汇编了国内外的各种资源，并重点回答一个问题：**如何系统的训练和提升数据工程技能，特别是以自学的方式。**

## 一. 数据工程与数据科学的关系

---

大数据是一个多学科交叉的领域，涉及到计算机科学，数学以及相关领域知识的方方面面，对相关从业者的素质要求比较高。其所需要的专业技能大致可以分为两个方面：**数据科学和数据工程**。

**数据科学**人才的初级形态为数据分析师，高级形态为数据科学家。数学与统计学是数据科学家的核心，他们需要在很强的**数学和统计背景**基础上建立高级分析的能力，建立各种机器学习和深度学习模型。对于数据科学家而言，他们除了数学基础扎实，还需要理解业务领域，在充分理解业务的基础上分析数据，这需要一定的商业敏锐度，最后要以业务方能够理解的方式发布他们得到的成果，帮助业务方进行决策。数据科学家需要具有一定的编程能力，因为他们需要编程训练各种机器学习模型，甚至根据业务领域的特殊性对标准模型源代码进行修改，以提高模型的性能。

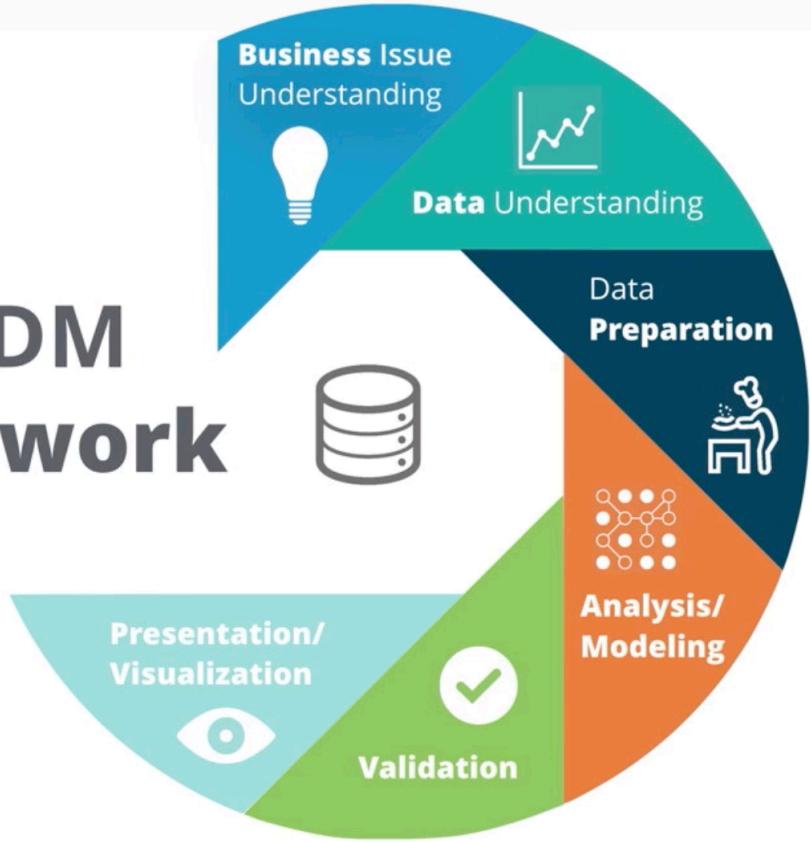
**数据工程**人才的初级形态为数据工程师，高级形态为数据架构师。编程能力是数据工程师的核心，这种能力和背景通常来自于计算机科学和软件工程。后端工程师转型成数据工程师具有先天的优势，因为知识都是相通的。他们的工作重点和专业能力主要放在**分布式系统**方面，数据架构师必须具备高级编程和系统架构能力。利用工程技能，数据工程师需要将10-30种不同的大数据技术整合到一起创建数据管道，构建和维护适用于数据收集，处理和部署数据密集型应用的数据结构和体系结构，将数据汇总给数据科学家，从而将模型投入生产。数据工程师必须深入理解各种技术和框架，以及如何选取合适的组件并将它们有机组合在一起创建解决方案。数据工程中最受欢迎的技能之一是设计和构建**数据仓库**的能力。数据仓库是收集，存储和检索所有原始数据的地方。

数据科学家和数据工程师之间的技能一定存在着重叠区域，这个重叠区域包含**数据分析技能**，**编程技能**和**大数据技能**。数据科学家的分析技能远远超过数据工程师的分析技能，数据工程师可以懂一些基础到中级的分析技术，但很难进行数据科学家熟悉的高级数据分析。数据科学家和数据工程师在编程能力上还有一些重叠，但数据工程师的编程能力远远超过数据科学家的编程能力。数据工程师为数据科学家创建数据管道，ETL（提取，转换和载入）是数据工程师构建数据管道所遵循的步骤。他们的角色是互补的，数据工程为数据科学工作提供支撑。数据科学家和数据工程师在大数据方面还存在一个重叠：数据工程师通过高级编程和系统架构创建大数据管道，数据科学家运用他们的高级数学技能，通过数据管道创建高级数据产品。

一个数据科学团队的常见配置是1个数据科学家配2-3名数据工程师。在另外一些更复杂的情况下，可能需要为1个数据科学家配4-5名数据工程师，因为创建数据管道要比创建ML/AI部分花费更多的时间和精力。

1997年欧盟起草了被称为"跨行业数据挖掘标准流程"的CRISP-DM模型(**CRoss-Industry Standard Process for Data Mining**)。

# CRISP-DM Framework



该模型通过以下步骤来系统地解决问题：

- 业务理解 Business Issue Understanding
- 数据理解 Data Understanding
- 数据准备 Data Preparation
- 分析/建模 Analysis/Modeling
- 模型评估 Validation
- 模型发布/可视化 Presentation/Visualization

数据工程师要为数据科学家服务，数据工程师搭建满足上述标准流程的基础设施和服务，确保数据科学家能够完成最终的工作。

## 二. 数据工程与云计算的关系

**AI+BigData+Cloud**可以说是当下信息产业的代名词。上面我们纵向比较了数据工程在大数据领域所处的位置以及它与数据科学之间的关系。这里我们还需要横向比较一下与数据工程有关的其他领域。其中我认为最重要的就是云计算技术。大数据是原料，云计算是容器。云计算的进步是大数据得以发展的基础条件。我们都知道，有一个概念叫**数据重力**，简单说就是所有的数据都要汇聚集中到一个地方然后才能进行分析。但传统的单机或者本地数据中心已经无法容纳海量的数据了。

云计算的出现就是要打破这种局限性，为海量的数据提供容器。云计算把众多的数据聚集起来，云端就成了数据分析最理想的发生点，要充分利用大数据的“大”，就要靠云计算技术实现。未来大数据的发展一定是和云计算共生的，即**数据往云端迁移**，数据的重心在云端。“云”作为一个海纳百川的数据中心，能激发大数据的真正潜力。

因此，对于数据工程师能力的培养，应当把重心放在分布式系统和云计算上。数据工程技能结合云计算，才能充分发挥数据工程的强大威力，有力地支撑数据密集型应用系统的架构和实施。

数据工程能力培养的具体策略可以归纳为八个字"先总后分，各个击破"，分三个阶段实施。第一阶段，先学习Udacity的数据工程师，云计算软件开发和云计算DevOps三个纳米学位，建立全局观；第二阶段，沿着下面要介绍的数据工程成长路径逐步熟悉各数据工程组件；第三阶段，结合云平台设计各种贴近实战的教学项目，进一步磨炼数据工程+云计算的综合技能，以臻融会贯通之境界。

### 技能，才是学习的终点。——《精进》

"做中学"是最有效的学习方式。数据工程的能力培养会设计成一系列循序渐进的关卡，每个关卡都有相应的任务和挑战，尽量以丰富多彩的形式调动多种即时反馈：

1. 文字材料阅读
2. 互动课程
3. 教学视频
4. 学习笔记
5. 实战项目
6. 启动会/分享会

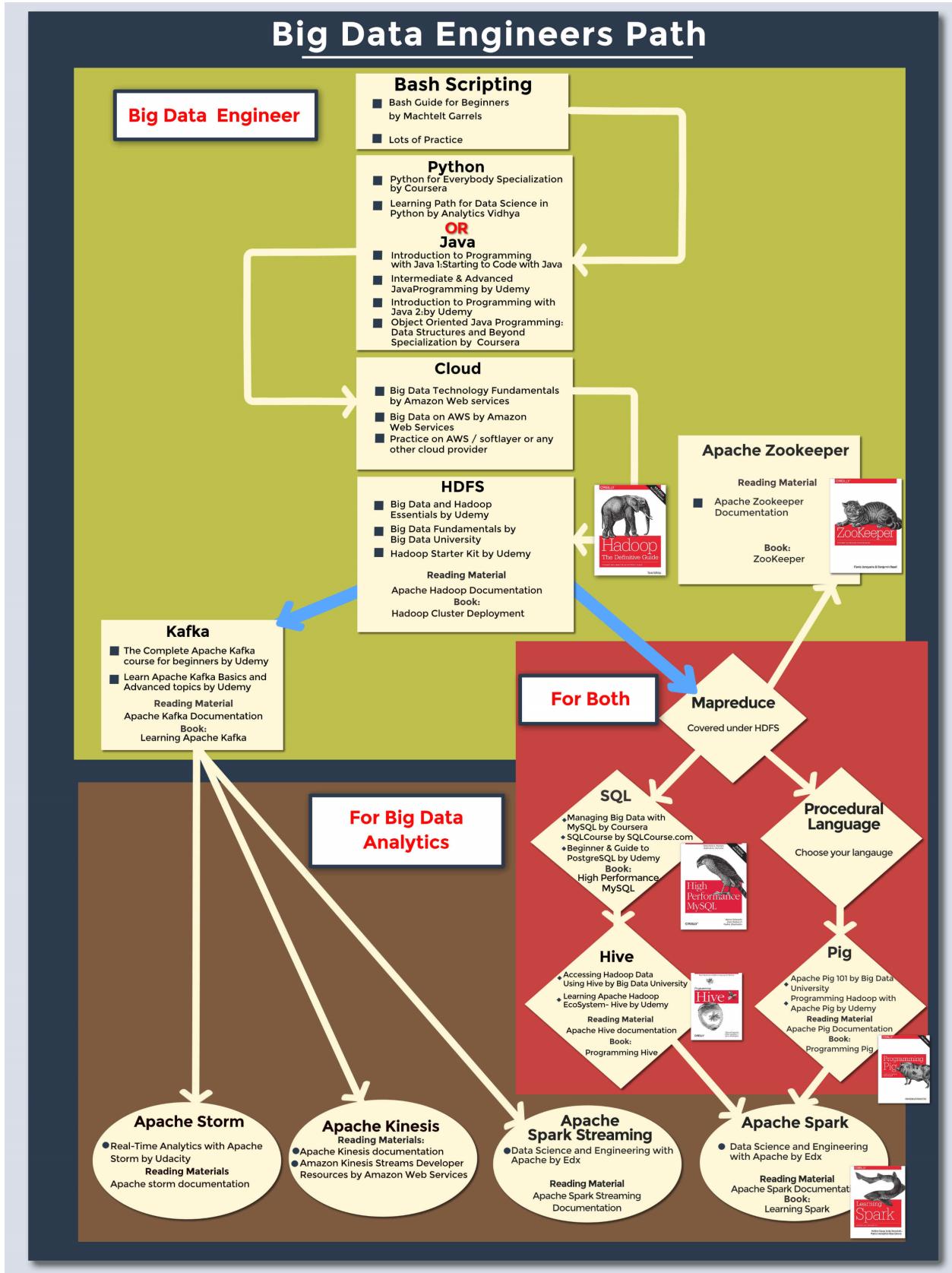
相关实战项目的设计，会先确定目标人群，学习目标和学习时长，然后根据这些目标合理拆分并制作内容。这么做的核心理念就是把知识当成一项技能来操练。我们掌握了多少知识，并不取决于记忆了多少知识以及知识的关联，而是取决于能调用多少知识以及知识的关联。在"知识技能化"的理念下，可以通过三种方式进行操练：

1. 写作式操练：阅读是一种知识输入，而写作是一种知识输出。前者是从浅表层理解知识，后者是从深层结构重构知识框架，观察和调用知识与知识之间的深层关联，需要缜密的思维，清晰的表达以及翔实的依据。
2. 游戏式操练：把学习当成一种游戏，这种操练就具有参与性，互动性和投入性。使学习者能够在更丰富多样的情景下去应用知识。
3. 设计式操练：调用已有的知识，设计某种解决方案来解决某个特定的问题。迫使学习者为了解决某个现实问题，综合性，创造性地调用和提取知识及其关联。在设计中，当学习者发现缺少某方面必须的知识时，这会驱动TA对这些知识进行学习。

数据工程能力的培养会融入以上三种知识技能化的操练方式。学习者既可以通过下面罗列的成长路径结合自身情况自由学习，又可以按部就班地根据路径逐步深入。知识的每一次提取都是一次重新写入，因此通过不断的提取调用强化知识之间的关联，做到磨炼技能，日日精进，而不是只单纯的接受信息和记忆知识。

## 三. 数据工程成长路径

---



## 3.1 操作系统

1. [Linux服务器管理和安全](#)
2. [CS401: Operating Systems](#)
3. [Bash Guide for Beginners](#)
4. 《现代操作系统》

5. 《Linux命令与shell脚本编程大全》

## 3.2 编程语言

---

### 3.2.1 Python语言

1. [Programming Languages Part A](#)
2. [Programming Languages Part B](#)
3. [Programming Languages Part C](#)
4. [Coursera零基础Python入门专项课程](#)
5. [Python 3专项课程](#)
6. [Python - 100天从新手到大师](#)

### 3.2.2 Java语言

1. 《Java编程思想》
2. 《Java并发编程实战》
3. [Object Oriented Java Programming: Data Structures and Beyond 专项课程](#)

### 3.2.3 Scala语言

1. [Functional Programming in Scala专项课程](#)

### 3.2.4 Go语言

1. 《The Go Programming Language》

### 3.2.5 Rust语言

1. 《深入浅出Rust》
2. 《Rust编程之道》
3. [Operating Systems Design and Implementation](#)

### 3.2.6 C++语言

1. [Udacity C++程序设计纳米学位](#)
2. 《C++ Primer》

## 3.3 云计算

---

1. [AWS 大数据技术基础知识](#)
2. [Big Data on AWS](#)
3. [Udacity云计算软件开发纳米学位](#)
4. [Udacity云计算DevOps](#)
5. [Data Engineering on Google Cloud Platform专项课程](#)
6. [Preparing for the Google Cloud Professional Data Engineer Exam](#)
7. [Google Cloud Platform Fundamentals: Core Infrastructure](#)

## 3.4 分布式存储

---

### 3.4.1 HDFS

1. [Google的GFS论文](#)
2. [Big Data and Hadoop Essentials](#)
3. [Big Data Fundamentals](#)
4. [Hadoop Starter Kit](#)
5. [Hadoop基础知识](#)
6. [HortonWorks教程](#)
7. [Hadoop详解](#)
8. [Hadoop：你应该了解的](#)
9. 《Hadoop: The Definitive Guide》

## 3.5 分布式协调

---

### 3.5.1 Apache Zookeeper

1. 《ZooKeeper》

## 3.6 消息队列

---

### 3.6.1 Apache Kafka

1. [The Complete Apache Kafka course for beginners](#)
2. [Learning Apache Kafka Basics and Advanced topics](#)
3. [使用Apache Kafka简化数据管道](#)
4. [用Kafka给数据科学家赋能](#)
5. 《Learning Apache Kafka》

### 3.6.2 ActiveMQ

1. JavaEE企业级架构师课程-ActiveMQ

## 3.7 SQL数据库

---

1. [使用MySQL管理大数据](#)
2. [Beginner's Guide to PostgreSQL](#)
3. [MySQL教程](#)
4. [PostgreSQL教程](#)
5. [Oracle Live SQL](#)
6. 《高性能MySQL》
7. 《SQL基础教程》
8. 《SQL进阶教程》

## 3.8 交互式数据分析

---

## 3.8.1 Apache Hive

1. [Accessing Hadoop Data Using Hive](#)
2. 《Programming Hive》

## 3.8.2 Apache Pig

1. [Apache Pig 101](#)
2. 《Programming Pig》

## 3.8.3 Apache Kylin

1. [Apache Kylin主页](#)

# 3.9 分布式计算

---

## 3.9.1 离线批处理

### 3.9.1.1 MapReduce

1. [Introduction to MapReduce](#)
2. [Hadoop beyond traditional MapReduce](#)
3. [使用MapReduce进行数据密集型文本处理](#)

## 3.9.2 流式计算

### 3.9.2.1 Apache Storm

1. [用 Apache Storm 进行实时分析](#)

### 3.9.2.2 Amazon Kinesis

1. [Amazon Kinesis 文档](#)
2. [Amazon Kinesis Data Streams 资源](#)

### 3.9.2.3 Apache Samza

1. [Apache Samza 文档](#)

### 3.9.2.4 Apache Flink

1. [Apache Flink 文档](#)

## 3.9.3 内存计算

### 3.9.3.1 Apache Spark

1. [Data Science and Engineering With Spark](#)
2. [Udacity数据工程师纳米学位](#)
3. [Comprehensive Introduction to Apache Spark, RDDs & Dataframes](#)

4. [初学者学习Spark R的详细指南](#)
5. [Spark的基础知识](#)
6. [Apache Spark和AWS简介](#)
7. [《Learning Spark》](#)

## 3.10 NoSQL数据库

---

### 3.10.1 KV数据库

#### 3.10.1.1 Redis

1. [Redis University](#)

### 3.10.2 列式存储数据库

#### 3.10.2.1 HBASE

1. [Google Big Table论文](#)

#### 3.10.2.2 Cassandra

1. [Cassandra Tutorial](#)

### 3.10.3 文档数据库

#### 3.10.3.1 MongoDB

1. [Introduction to MongoDB](#)
2. [MongoDB在线课程合集](#)

#### 3.10.3.2 Couchbase

1. [Couchbase Training](#)

## 3.11 资源调度

---

1. [IBM微服务专项课程](#)

## 3.12 日志采集

---

1. [Flume教程](#)

## 3.13 RPC框架

---

1. [网络编程实践](#)
2. [《Netty in Action》](#)
3. [《Linux多线程服务端编程》](#)

## 3.14 综合资料

---

### 3.14.1 Airbnb数据工程入门

1. [数据工程入门指南Part I](#)
2. [数据工程入门指南Part II](#)
3. [数据工程入门指南Part III](#)

### 3.14.2 O'Reilly免费数据工程电子书套件

1. [O'Reilly的免费数据工程电子书套件](#)

### 3.14.3 数据工程综合课程

1. [Big Data for Data Engineers](#)
2. [大数据专项课程](#)
3. [IBM数据科学专业证书](#)
4. [IBM高级数据科学专项课程](#)
5. [物联网程序设计专项课程](#)
6. 《数据密集型应用系统设计》

### 3.14.4 24个终极数据科学项目

1. [提升你知识和技能的24个终极数据科学项目](#)

## 五. 数据工程培养计划

---

## 四. 数据工程师必备前置知识

---

### 4.1 分布式系统基础知识

---

1. [一致性哈希算法](#)
2. [CAP理论](#)
3. [幂等性：很多分布式系统状态管理的基石](#)
4. 各种一致性模型：强一致性、弱一致性、最终一致性
5. 备份机制：主从的叫法已经不怎么流行了，当前更cool的叫法是Leader-Follower模式
6. 共识协议：国内通常翻译成一致性协议(consensus protocol)。学习常见的几种：Paxos和Raft

### 4.2 机器学习基础知识

---

1. [机器学习新手指南](#)
2. [机器学习算法基础知识](#)
3. [新手必读的机器学习和人工智能书籍](#)

# 六. 数据工程相关认证考试

## 6.1 谷歌认证专家

目前最重要的数据工程认证之一，要获得这个证书需要通过2小时的考试，题目是多项选择题，该网页提供了一些实际操作谷歌云技术的实践指南。

## 6.2 IBM认证数据工程师

考试包括54个问题，必须正确回答44个才能通过，这里提供了进一步的学习资料，可以参考这些资料进行准备。[这里是考试页面](#)。

## 6.3 Cloudera的CCP数据工程师

另一个全球公认的认证，相当于具有挑战性，需要有一些使用数据工程工具的实践经验。

### 1. Cloudera的CCP数据工程师

路漫漫其修远兮，吾将上下而求索。



## 七. 参考文献

---

1. [数据工程师vs数据科学家](#)
2. [终于可以弄明白大数据、云计算、商业智能到底什么关系了](#)
3. [Big Data Learning Path for all Engineers and Data Scientists out there](#)
4. [一份数据工程师必备的学习资源，干货满满（附链接）](#)