第 1 章
什么是数据科学

本章主要介绍数据科学的定义和关键技术，数据科学的关键技术包括数据存储计算、数据治理、结构化数据分析、语音分析、视觉分析、文本分析和知识图谱等，本章对每一项关键技术进行基本的介绍。

1.1 数据科学的定义

1.1.1 数据科学的背景

新一代信息技术进入生产成熟期。数字经济迎来战略机遇期，市场推崇务实和价值，企业和人才必须具备很强的综合实力。数字经济迎来战略机遇期。数字经济正在成为重组全球要素资源、重塑全球经济结构、改变全球竞争格局的关键力量，发展数字经济是把握新一轮科技革命和产业变革新机遇的战略选择。“十四五”的重点是产业数字化，即数字技术赋能传统行业，这是对数据科学的极大利好。

市场更加理性，推崇务实和价值。行业拥抱数字技术的过程，也是传统文化和创新文化融合的过程。传统行业关注场景、注重实效、追求性价比的风格，势必使得只有能解决最复杂场景问题的数字技术企业才有市场空间，这类企业只有将场景、技术和数据深度融合才能创造价值。

数字技术企业需要具备端到端的场景解决方案构建能力。当前，产业数字化处于孵化阶段，行业场景解决方案还不成熟，无法做到精细化分工，这就要求先行者必须打通数据集成、数据治理、数据分析、数据应用的全流程，形成端到端的解决方案。也势必要求先行者掌握数据科学的理论、方法和技术，具备业务分析、数据建模和应用、工程实现等全方位的数据价值实现能力。

数据科学人才需求旺盛，产、学、研协同势在必行。在产业数字化进程中，行业客户和数字技术企业都需要具备数据素养的复合型人才，这必须依靠高校和企业携手培养人才，更需要兼具理论、方法、工具和实践的平台支撑学科建设。

1.1.2 数据科学的定义

数据科学是为数字经济提供基础与技术支撑的学科，是有关数据价值链实现过程的基础理

论与方法学。它运用建模、分析、计算和学习融合的方法研究从数据到信息、从信息到知识、从知识到决策的转换，并实现对现实世界的认知与操控。

其中数据价值链是由“数据集成－数据治理－数据建模－数据分析－数据应用”组成的一个数据价值增值过程，如图 1-1 所示。

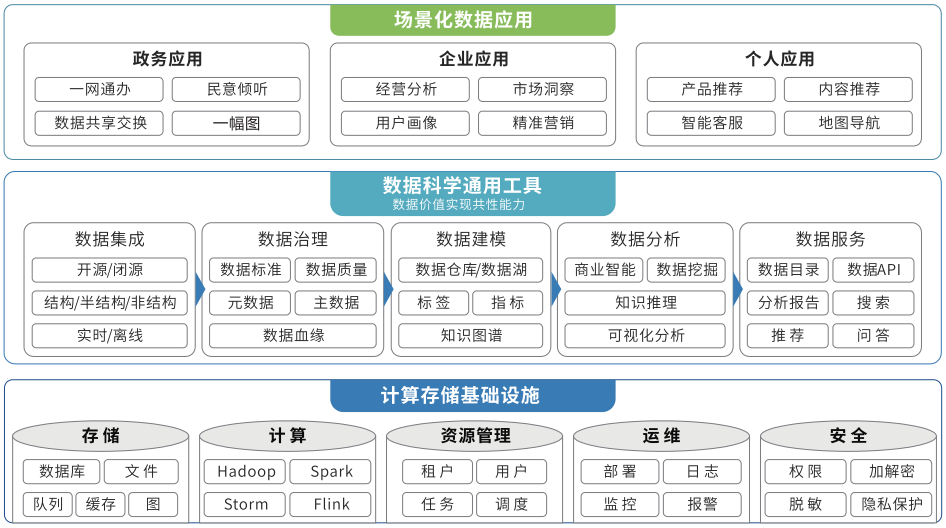

图 1-1　数据科学的组成

数据科学和大数据有着密切的关系，对大数据应用提供了强有力的支撑。随着大数据技术的不断发展和深度应用，大数据应用系统相比传统信息化系统表现出更高的技术难度和复杂度。通常来说，大数据应用的技术架构包含如图 1-1 所示的三个层次：

（1）场景化数据应用。面向业务用户的应用系统，按照用户群体大致可分为政务应用、企业应用和个人应用等类别。数据应用系统服务于特定的业务场景，为用户提供有价值的数据并利用数据驱动业务流程和决策，其中有价值的数据来自中间层数据科学通用工具。

（2）数据科学通用工具。面向数据工程师、数据科学家和数据分析师，帮助他们高效地开展数据集成、治理、建模、数据分析和服务等各项工作，快速实现数据的价值。数据科学通用工具一方面依赖底层计算存储基础设施，另一方面又必须对底层的基础能力进行提炼、封装和组合，让用户专注于核心的数据价值实现，免于陷入底层设施复杂的技术选型和环境配置。

（3）计算存储基础设施。面向系统工程师，帮助他们构建大数据系统必需的存储、计算和管理能力。计算存储基础设施层的技术复杂性体现在两个方面：①本层包含众多相对独立且专业程度较高的技术组件，例如仅数据存储组件便可分为关系数据库、文件系统、消息队列、缓存、图数据库、搜索引擎等多种类型，它们适用于不同的应用场景，一个应用往往需要组合使用多种存储组件，这就导致了大数据系统在底层基础设施的设计和管理上具有不可避免的复杂性；②存储计算基础设施层虽然有大量开源软件可供选择，但开源软件的使用门槛较高，往往需要系统工程师进行大量的选型、适配、配置和二次开发工作，以打造出更加集成、方便、安全和兼容性良好的产品。

大数据、数据科学、人工智能、数据智能这些领域有很多交集，容易发生混淆，下面给出它们的联系和区别：

- 大数据包含计算存储基础设施、数据科学通用工具、场景化数据应用三个细分领域，其中数据科学通用工具是大数据价值实现的关键，也是数据科学的研究重点。
- 数据科学研究数据价值链中的理论、技术和方法，侧重多模态数据融合、数据建模、知识发现、分析洞察、数据可视化、数据解释等方面。数据科学会应用统计学、信息学、人工智能、管理学、社会学等多领域的知识。
- 人工智能以实现模拟人的智能为目标，包括感知、认知、决策和行动，侧重智能的数学表示、构建和应用方面的理论和技术。
- 数据智能是指利用大数据和人工智能技术，用数据描述并分析现实和驱动业务智能化，更侧重场景化数据应用方面。

1.2 数据科学的关键技术

数据科学的关键技术包括数据存储计算、数据治理、结构化数据分析、语音分析、视觉分析、文本分析、知识图谱等。数据科学平台围绕数据价值转化过程，将数据科学中的关键技术统一到一个平台上，打通数据集成、数据治理、数据建模、数据分析、数据应用各阶段，让数据科学团队中的每个成员都只需专注于核心业务问题，免于陷入复杂的技术环境。数据科学团队中包含如下角色。

- 数据工程师：可通过平台完成数据采集 / 汇聚、数据存储 / 治理、数据 ETL（Extract-Transform-Load，提取 - 转换 - 加载）等工作。自研集成引擎，支持多种数据源接入，通过可视化方式完成多源异构数据集成；提供数据目录、数据质量、数据血缘等治理能力；支持流批一体计算、低代码开发模式，支持工作流编排。处理后的数据可接入知识生产、知识应用工具中。
- 数据分析师：可通过平台完成数据分析、数据应用等工作。通过利用数据工程师加工后的数据，采用数据探查、数据可视化、行业模型、知识应用等工具展开数据分析，各工具之间数据互联互通，并支持以编码方式进行数据分析。
- 数据科学家：平台提供了自然语言处理（Natural Language Processing，NLP）、智能语音、计算机视觉等方面的预训练模型，同时支持自助式机器学习、AutoML、MLOps 等能力供数据科学家构建机器学习模型，可将训练得到的模型注册为服务，供数据分析使用。

数据科学平台致力于解决数据价值转化过程中的共性需求，为数据工程师、数据分析师和数据科学家群体提供能力全面、交互自然、知识驱动的通用工具，高效构建数据应用。数据科学平台的功能架构如图 1-2 所示。

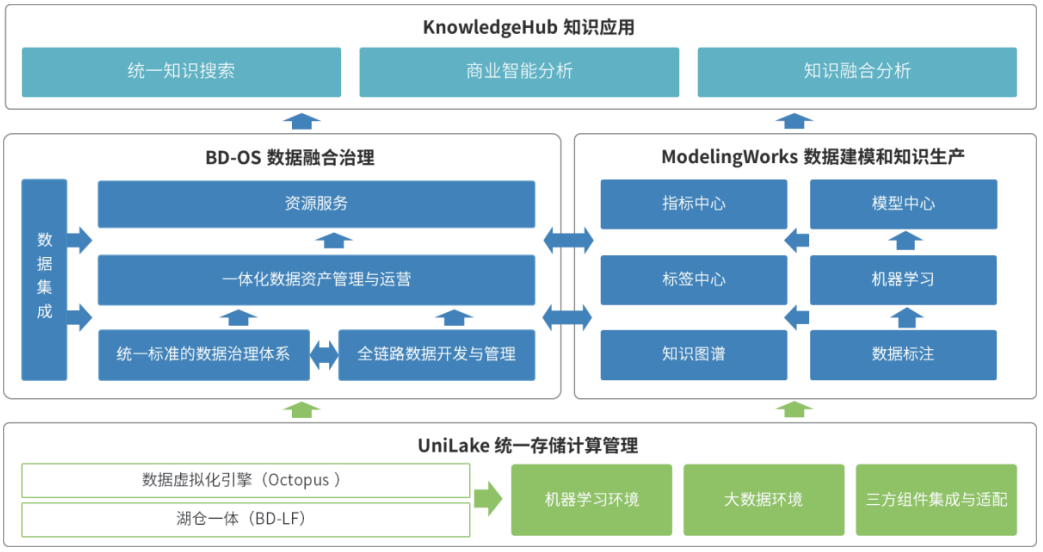

图 1-2 数据科学平台的功能架构

为了让用户专注于数据价值的实现，从各种技术组件的复杂配置和相互适配中解脱出来，数据科学平台提供了统一的湖仓一体架构，屏蔽了底层存储和计算组件的技术细节，并基于此提供了数据融合治理、数据建模与知识生产、知识应用三类专业工具集。平台的主要功能说明如下。

- 湖仓一体架构：支持外部数据源物理入湖和虚拟入湖，支持结构化、半结构化和非结构化数据，并提供统一的元数据、数据权限和安全管理。支持 Hive、Spark、Flink 等大数据处理引擎，以及 R、TensorFlow、PyTorch 等机器学习工具，支持批流一体计算，支持联邦查询。支持通过标准 SQL 访问湖仓数据。
- 数据融合治理：高效集成网络开源数据、业务系统、日志文件、IoT（Internet of Things，物联网）数据、人工填报信息等多种数据源。支持关系数据库的探查和变化数据捕获（Change Data Capture，CDC）。支持多种数据处理任务的开发和统一调度。支持完善的数据治理体系，包括数据标准管理、元数据管理、数据生命周期管理、数据质量管理、主数据管理等。支持数据资产的管理和运营。通过数据资源目录提供数据库、文件、API 等多种数据服务方式。
- 数据建模与知识生产：平台提供了统一的多模态数据建模能力，包括数据标注、机器学习、模型管理等，提供了在线 Notebook 和低代码的模型构建能力，并且内置了多种机器学习算法和开箱即用的 AI 模型。这些模型可以用于构建业务知识，平台支持标签、指标和知识图谱三种常见的知识表示形式，并提供了对应的知识生产和管理能力。
- 知识应用：平台提供了统一知识搜索，可以快速检索数据湖、数据仓库、数据资源、标签、指标、知识图谱，支持全文检索、图文搜索、以图搜图、内容推荐等能力。平台提供敏捷 BI（Business Intelligence，商业智能），可以快速分析数据仓库、标签和指标数据。平台还提供了知识图谱分析和推理工具，支持实体分析、关联分析、时空分析、图谱挖掘等多种数据分析手段。平台统一管理领域内的数据标准、数据加工算子、数据仓库模型、知识图谱本体模型、分析算

法等知识，并运用这些知识增强数据治理、数据建模和知识生产、知识应用的全过程。

1.2.1 数据存储计算

数据存储计算位于数据科学的基础层，它提供了数据存储、计算和分析的基础设施，为数据科学家提供了处理和分析大规模数据的能力，具体包括数据存储、处理、分析和管理等方面的技术，如分布式存储（Distributed Storage）技术、分布式计算技术、分布式分析技术、资源管理技术等。这些技术为数据科学家提供了强大的数据处理和分析能力，可以帮助他们更好地探索和理解数据。

1. 分布式存储系统

分布式存储技术的定义：将数据按照一定的分布算法分散存储在多台独立的存储节点上，实现多节点并行访问的存储技术。

常见的用户数据存储主要是分布式文件及对象存储，支撑不同类型的数据进行原始格式的保障。在分布式文件存储中，文件被分成小块并存储在多台服务器上，通过文件系统进行管理和访问。每个文件块都有一个独立的标识符，可以根据该标识符定位文件块并从多个服务器中获取数据以组合成完整的文件。这种架构通常用于需要大量读取和写入的应用程序，例如视频流、科学计算和大规模数据分析，并且一半的文件系统也支持通过 Fuse 方式挂载成本地磁盘路径，这样使用方可以通过本地文件方式进行访问和使用。常见的文件系统包括 HDFS（Hadoop Distributed File System，Hadoop 分布式文件系统）、Ceph、GlusterFS、FastDFS 等。

1）分布式文件系统

HDFS 为 Hadoop 提供数据存储能力，以 HDFS 为例的典型分布式文件系统架构如图 1-3 所示。

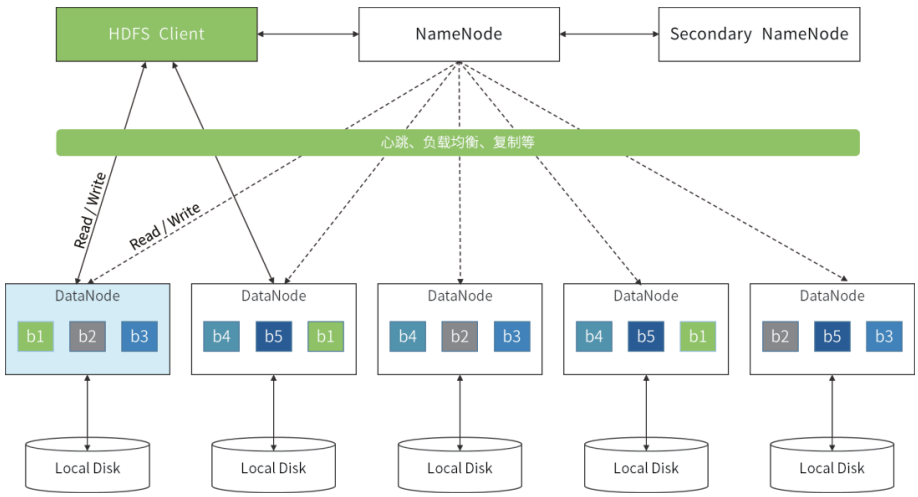

图 1-3　HDFS 系统架构图

在 HDFS 中，一个文件被分成一个或多个数据块，这些数据块存储在一组 DataNode 上。

NameNode 会根据副本数量和数据块所在节点的负载情况来确定数据块的复制位置。这样可以保证在某个 DataNode 节点失效的情况下，数据仍然可用。NameNode 是一个中心服务器，它负责管理文件系统的命名空间（Namespace），包括文件和目录的创建、删除、重命名等操作。此外，NameNode 还负责管理数据块的元数据信息，如数据块的副本数量、所在 DataNode 节点的位置等。NameNode 会将这些元数据信息存储在内存中，并定期持久化到本地磁盘上，以保证集群的可靠性和一致性。

DataNode 是 HDFS 集群中的从属节点，一般每个节点都会有一个 DataNode。DataNode 负责管理它所在节点上的存储，并处理文件系统客户端的读写请求。它会向 NameNode 定期发送心跳信息，以保证它的可用性。此外，DataNode 还负责处理数据块的创建、删除和复制等操作。

2）对象存储

在对象存储中，文件被视为对象，并将对象存储在无须文件系统的分散节点上。每个对象都有一个唯一的标识符和元数据，例如创建时间、大小和内容类型等。这种架构通常用于需要大规模存储和访问数据的应用程序，例如云存储和内容分发网络（Content Delivery Network，CDN）。另外，随着云计算和大数据技术的兴起，存储与计算分离变得越来越普遍。云计算提供了一个可伸缩的基础设施平台，使得用户可以根据需要增加或减少计算资源。同时，大数据技术使得分布式存储和计算变得更加高效和可靠。例如，TensorFlow 分布式计算框架允许用户在多个计算机节点上并行训练深度学习模型，从而加快训练速度。目前主流的计算与分析框架也支持对象存储的数据访问，并且兼容各云厂商的对象存储服务，如阿里云的 OSS、华为云 OBS、Azure Blob Storage、AWS S3 等。常见的开源对象存储系统包括 Ceph、Swift、MinIO 等。

Ceph 是一种可扩展的分布式存储系统，可以提供对象存储、块存储和文件存储等服务，其基础架构如图 1-4 所示。

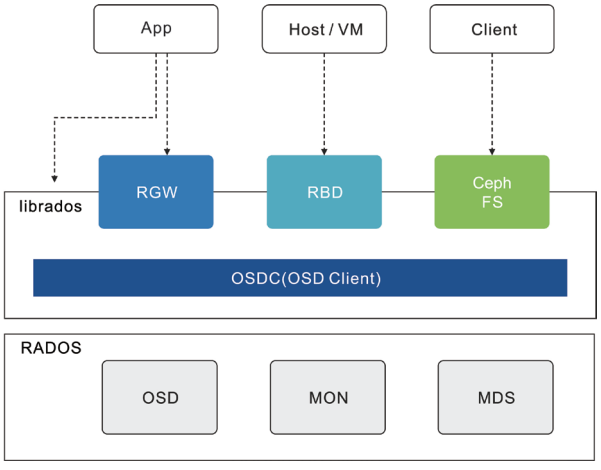

图 1-4 Ceph 系统架构图

以 Ceph 为代表的架构，与 HDFS 不同的地方在于该架构中没有中心节点。

按照其分层的架构，最底层是 RADOS 集群，它提供了高可靠性的数据存储和管理功能。

RADOS 集群实现了分布式的基本特征，包括数据可靠性保护、分布式一致性和故障检测与恢复等。上层的存储形态包括块存储、文件存储和对象存储等，它们都以对象为粒度进行数据存储。Ceph 集群为客户端提供了丰富的访问形式，比如对于块存储可以通过动态库或者块设备的方式访问。其中，动态库提供了 API，用户通过该 API 可以访问块存储系统。

大多数对象存储系统（包括 Ceph、S3、Swift、MinIO）除满足对象存储的能力外，也提供文件存储和块存储的服务能力，以满足不同类型的存储需求。主流的分布式存储对比如表 1-1 所示。

表1-1　存储系统对比表

存储系统	Ceph	Swift	HDFS	MonIO
开发语言	C++	Python	Java	AGPLv3
协议	LGPL	Apache	Apache	Golang
数据存储方式	对象/文件/块	对象	文件	对象
在线扩容	支持	支持	支持	支持
冗余备份	支持	支持	支持	支持
单点故障	不存在	不存在	存在（NameNode）	不存在
跨集群同步	不支持	支持	不支持	不支持
易用性	中等，官方文档详细	复杂，官方文档详细	复杂，官方文档详细	简单，官方文档详细
使用场景	大、中、小文件	OpenStack对象存储	Hadoop底层文件存储	轻量化文件存储
FUSE挂载	支持	支持	支持	支持
访问接口	POSIX	POSIX	不支持POSIX	POSIX

2. 全文搜索

全文检索（Full-Text Search）是一种基于文本的信息检索技术，通过对文本数据建立索引并支持文本查询，可以高效地进行文本信息检索。通常情况下，全文检索引擎会将文本数据切割成若干词项（Term），并对这些词项进行索引。当用户输入查询词（Query）时，全文检索引擎会对索引进行搜索，返回符合条件的文本数据。全文检索不同于关系数据库的模糊查询，它不仅能够精确匹配查询条件，还能够匹配近义词、拼写错误、词形变化等文本变化。在很多需要文本检索的场景下，如搜索引擎、电子商务网站、社交媒体、新闻门户等，全文检索技术都得到了广泛应用。

对于数据科学来说，只要涉及基于关键字的内容搜索都会涉及全文搜索的技术，开源的全文搜索技术一般都是基于 Lucene 实现的。主流的开源搜索框架包括 ElasticSearch 和 Solr。

Lucene 是一个开源的全文检索工具包，提供了构建搜索引擎的基础架构。它可以让开发者快速构建自己的搜索引擎，并支持多种编程语言。ElasticSearch 和 Solr 都是基于 Lucene 的开源搜索框架，它们在 Lucene 的基础上进一步扩展了搜索功能，提供了更完整的搜索引擎解决方案。ElasticSearch 和 Solr 都支持分布式搜索、实时搜索、自动化索引等高级功能，并且提供了友好的 REST API，可以方便地与其他应用程序集成。

目前，开源搜索引擎的选型主要还是以 ElasticSearch 为主，提供实时搜索、分布式搜索、

自动索引、支持多语言等能力。此外，它可以处理大量数据，并且能够在几乎所有类型的数据中执行实时搜索和分析。在 ElasticSearch 中，数据被存储在分片中，并且分片会在多台服务器之间自动分布，以确保数据的可用性和高性能，对外可以通过简单的 RESTful API 进行交互。

此外，ElasticSearch 也提供了强大的数据分析能力，并且与主流的大数据框架进行了打通，如基于 Spark/Flink 访问 ElasticSearch 等场景。

3. 图数据库

维基百科把图数据库定义为一个使用图结构进行语义查询的数据库，它使用点、边和属性来表示和存储数据。这种数据结构使得图数据库在许多应用场景中比关系数据库和其他传统数据库更有效，例如社交网络分析、推荐系统、网络安全等。

目前主流的开源图数据库包括 Neo4j、JanusGraph、HugeGraph、Nebula Graph 等。其中，Neo4j 是最早的图数据库之一，但受限于企业版本闭源、社区版本只能单机模式等诸多限制，很多公司也都基于自身的业务开始图数据库的研发，并提交到开源社区，开源图数据库的发展也日益活跃。以百度开源的 HugeGraph 为例，能够与大数据平台无缝集成，有效解决海量图数据的存储、查询和关联分析需求。HugeGraph 支持 HBase 和 Cassandra 等常见的分布式系统作为其存储引擎来实现水平扩展。HugeGraph 可以与 Spark GraphX 进行链接，借助 Spark GraphX 图分析算法（如 PageRank、Connected Components、Triangle Count 等）对 HugeGraph 的数据进行分析挖掘。

图数据库有原生和非原生存储两种存储方式。原生图数据库使用图专用的存储引擎，对图数据进行优化存储和索引。这样能够更加高效地支持图查询，但是也意味着原生图数据库可能对图数据的处理能力更为有限，例如在扩展性和兼容性方面可能会存在一定的局限性，而非原生图数据库通常使用已有的 NoSQL 存储技术作为存储引擎，相对来说更加灵活，但也需要自行实现图查询的算法和数据结构，可能会对性能和查询能力产生一定的影响。同时，非原生图数据库通常具有更好的扩展性和兼容性，能够更加容易地集成到现有的技术栈中。Neo4j 属于原生图，而 HugeGraph 则属于非原生图的存储形式。

4. NoSQL 数据库

NoSQL 是指非关系数据库（Not only SQL），它是一种提供灵活、可扩展的处理和管理大量结构化和非结构化数据的数据库管理系统。与传统的关系数据库不同，NoSQL 数据库不使用固定的模式来组织和存储数据，而是使用动态模式来实现灵活的数据建模和检索。

NoSQL 数据库有多种类型，包括文档数据库、键值数据库和列式数据库等。文档数据库（如 MongoDB）将数据存储在文档中，这些文档可以是不同的结构和类型。键值数据库（如 Redis）使用键－值对存储数据。列式数据库（如 HBase、ClickHouse）使用列来存储相关的数据，而不是以行的方式，这样存储可以减少数据扫描，提供更高的性能。

NoSQL 数据库通常被用于高并发场景和大数据处理场景，如 Web 应用程序、移动应用程序、物联网、社交媒体和游戏等。相比传统的关系数据库，NoSQL 数据库具有更好的可扩展性、更高的性能和更灵活的数据模型。

主流的 NoSQL 数据库包括 Redis、HBase、MongoDB 等。

1）Redis 的介绍

Redis 是一种开源的、基于内存的数据存储系统，是流行的键值数据库，它支持多种数据结构，如字符串、哈希表、列表、集合和有序集合等。Redis 提供了类似于键－值对的存储方式，可以将数据保存在内存中或者持久化到磁盘中。Redis 提供了快速的读写速度、高可用性和数据持久化等功能，还支持事务、Lua 脚本、发布 / 订阅等高级功能。

Redis 常用于缓存、消息队列、实时计数器、实时消息传递等各种应用场景。Redis 的设计理念是追求简单、高性能、灵活性和可扩展性，并且提供了多种编程语言的客户端库，如 Python、Java、Ruby 等，使得它容易被集成到不同的应用程序中。

2）HBase 的介绍

HBase 是一个开源的、面向列的分布式数据库，它使用 Hadoop 作为其底层的分布式文件系统。HBase 源自于 Google 的 Bigtable 系统，并且被设计用于存储非结构化和半结构化数据。与传统的关系数据库不同，HBase 采用了基于列的存储模式，这意味着它能够高效地处理具有大量列但行数较少的数据集。

HBase 是基于 Google 的 Bigtable 论文实现的，其数据模型由“表”“行”和“列族”组成。表是 HBase 中的基本单元，每个表可以包含多个行和列族。每个行都有一个唯一的行键和多个列族，每个列族可以包含多个列。每个列都有一个唯一的列限定符和一个时间戳，用于标识不同版本的数据。HBase 还支持复杂的数据类型，如数组和嵌套的对象。并且，由于其基于 Hadoop 生态系统，HBase 可以与其他 Hadoop 组件集成，包括 Hive、MapReduce、Spark 等。

在使用上，HBase 可以处理海量的数据，支持横向扩展，可以通过增加节点来扩展集群的容量，HBase 还提供了强大的数据查询和过滤功能，可以使用行键、列限定符、时间戳等条件来查询和过滤数据，因而适合需要快速读写和随机访问大规模数据的存储，如实时数据处理、日志分析、推荐系统和社交网络等场景。

3）MongoDB 介绍

MongoDB 通常被称为文档数据库，其内部定义了一种类似于 JSON 的 BSON 结构来存储数据。它的数据模型由文档（Document）、集合（Collection）、数据库（Database）组成。文档是 MongoDB 中的基本单元，每个文档都是一个 BSON 文档，可以包含不同的字段和值。集合是一组文档的容器，每个集合都有一个唯一的名称，并且可以包含多个文档，可以理解为关系数据库中的表。数据库是 MongoDB 中的一个物理容器，每个数据库可以包含多个集合。MongoDB 不需要预定义数据模式，允许文档具有不同的结构和类型，并且可以直接扩展结构。MongoDB 具有很高的可扩展性和性能，可以处理大量的数据和高并发请求。它支持自动分片和副本集，可以通过添加节点来水平扩展集群的容量和提高数据可用性。MongoDB 还提供了强大的查询和聚合功能，可以使用索引、聚合等功能来查询和分析数据。此外，MongoDB 还支持 ACID（Atomicity、Consistency Isolation 和 Durability，原子性、一致性、隔离性和持久性）事务和多种数据存储引擎。其主要特点如下。

- 面向集合：数据以集合的形式存储，每个集合都有唯一标识名，并且可以包含无限数量的文档。
- 模式自由：集合中的文档可以是任意数据类型，不需要定义固定的模式，即使是不同的数据类型也可以存储在同一个集合中。
- 文档型：MongoDB 存储的数据是键－值对的集合，每个文档类似于关系数据库中的一条记录，可以包含任意数量的键－值对，值可以是任何数据类型。

此外，MongoDB 还提供了文件的存储模块 GridFS，在 GridFS 中，分为 fs.filsy 与 fs.chunks。fs.files 存储文件的元数据信息，包括文件名、文件类型、文件大小、上传时间等。而 fs.chunks 存储文件的数据，将大文件分割成多个小的 chunk（默认 256KB），每个 chunk 作为 MongoDB 的一个文档存储在 chunks 集合中。chunks 集合中的每个文档都包含 chunk 数据、文件 ID 以及当前 chunk 在整个文件中的位置等信息。

在使用上，MongoDB 适用于需要灵活性和可扩展性的应用场景，例如：

- 网站实时数据：MongoDB 非常适合实时地插入、更新与查询，并具备网站实时数据存储所需的复制及高度伸缩性。
- 数据缓存：由于性能很高，MongoDB 也适合作为信息基础设施的缓存层。MongoDB 通过持久化缓存层可以避免数据穿透导致底层数据库压力过大。
- 利用 GridFS 进行大量小文件的存储，如图片、文档的存储等。
- 对象或 JSON 数据存储：MongoDB 的 BSON 数据格式非常适合文档化格式的存储及查询。

5. 数据湖

数据湖（Data Lake）是一个以原始格式存储数据的存储库或系统，具有如下几个特点。

- 存储所有类型和格式的数据：数据湖可以存储结构化、半结构化和非结构化的数据，包括文本、图像、音频和视频等各种类型的数据。
- 无须预定义架构：数据湖不需要在存储数据之前预定义数据结构，这使得数据湖能够处理各种不同类型的数据，包括新的和未知的数据类型。
- 聚合数据：数据湖可以将来自不同来源的数据聚合到一个地方，这使得数据分析师和数据科学家可以使用单个数据源来执行各种分析任务。
- 大规模数据处理：数据湖可以处理大规模的数据，因为它通常运行在云计算环境中，具有可扩展性和弹性，能够自动调整资源来满足工作负载。
- 数据访问控制：数据湖可以为不同用户和应用程序提供不同级别的访问权限，并保护敏感数据不被未经授权的人员访问。

开源的数据湖架构主要基于 Delta Lake、Iceberg 和 Hudi 进行构建，整体架构如图 1-5 所示。数据湖架构整体分为 4 层，分别说明如下。

- 最底层是分布式文件系统，各云厂商通常基于自有的对象存储服务，如阿里的对象存储服务（OSS）、Amazon S3 和华为的对象存储服务（OBS）等。私有云环境，可以选择用户自己维护的 HDFS，或者选择 MinIO 等开源对象存储系统。

- 第二层是数据加速层。数据湖架构是一个存储计算分离的架构，如果所有的数据访问都远程读取文件系统上的数据，那么性能和成本开销都很大。如果能把经常访问的一些热点数据缓存在计算节点本地，这就非常自然地实现了冷热分离，一方面能收获不错的本地读取性能，另一方面还节省了远程访问的带宽，这层一般会选择开源的 Alluxio。

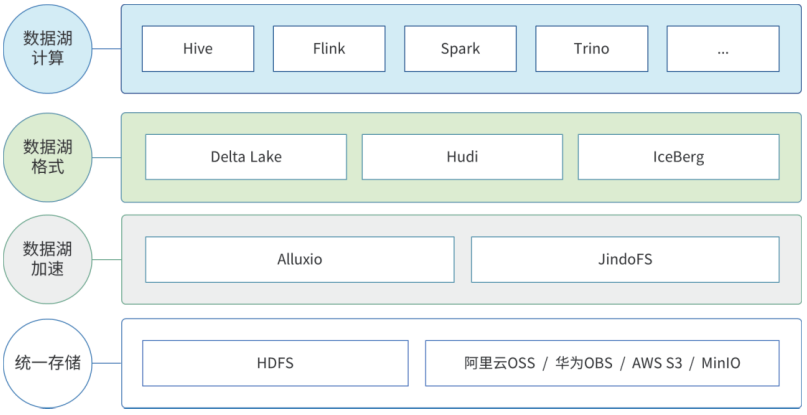

图 1-5　开源数据湖架构图

- 第三层是 Table Format 层，主要是把一批数据文件封装成一个有业务意义的 Table，提供 ACID、Snapshot、Schema、Partition 等表级别的语义。一般对应着开源的 Delta、Iceberg、Hudi 等项目。
- 最上层是不同计算场景下的计算引擎。其中，开源的常见计算引擎包括 Spark、Flink、Hive、Presto、Hive MR 等。这批计算引擎可以同时访问数据湖中同一张表。

从数据湖的技术架构可以看出，Delta Lake、Iceberg 和 Hudi 位于第三层，为大数据分析提供了一种开放的表格式。

首先 Delta Lake 作为开源项目由 Databricks（Apache Spark 的创建者）维护，因此与 Spark 深度集成以实现读写操作。它还支持 Presto、AWS Athena、AWS Redshift Spectrum 和 Snowflake 的读取操作，提供 ACID 事务支持、版本控制、数据修复和一致性保证等功能。而 Hudi 是由 Uber 开源的开源数据技术，在 Hadoop 和 Apache Spark 之上提供了一种高效、可靠和可伸缩的数据湖解决方案，支持对列式数据格式的增量更新，支持从多个来源摄取数据，包括通过 Spark 从外部数据源（如 Apache Kafka）读取数据，支持从 Apache Hive、Apache Impala 和 PrestoDB 等读取数据。Iceberg 作为新兴的数据湖框架之一，抽象出“表格格式”（Table Format）这一中间层。Iceberg 既独立于上层的计算引擎（如 Spark 和 Flink）和查询引擎（如 Hive 和 Presto），也和下层的文件格式（如 Parquet、ORC 和 Avro）相互解耦。Iceberg 的架构和实现并未绑定于某一特定引擎，它实现了通用的数据组织格式，利用此格式可以方便地与不同引擎（如 Flink、Hive、Spark）对接。

Delta Lake、Hudi、Iceberg 三个开源项目都是为了解决数据湖架构中的数据管理和处理问题而设计的。整体功能类似，但它们各自定位又略有差异。

- Delta Lake 的定位是流批一体的数据处理，它能够在批处理和流处理之间自动切换，支持增量更新和查询，并且具有 ACID 事务特性，能够保证数据的一致性和可靠性。
- Hudi 的初衷场景是增量的 Upserts，能够高效地支持数据的插入、更新和删除操作，并且能够通过分区和索引等方式加速数据的查询。它的特点是支持多种存储格式，可以根据数据的特

点和需求选择不同的存储方式。

- Iceberg 的目标是做一个通用的 Table Format，它的设计原则是解耦计算引擎和存储系统，能够支持多种计算引擎和存储格式。Iceberg 的特点是数据版本管理和快照机制，能够实现增量更新和查询，并且支持多版本数据的回溯和恢复。

1.2.2 数据治理

当下数据已经成为政府和企业决策的重要手段与依据，同时近年来政府也多次提出推进和加快政府及企业的数字化转型进程。在数字化转型的建设过程中，数据治理体系建设一直是业界探索的热点。但与传统要素不同的是，数据是无形的，且数据是孤岛林立的、杂乱的，要想发挥数据价值，提升数据治理能力是必要举措。

面对政府和企业数据多样化、数据需求个性化、数据应用智能化的需求，数据治理需求急剧增加，如何做好数据治理以及提升数据治理能力成为政府和企业共同的需求。以数字政府为例，面向数据本身的管理与治理市场还处于大规模的人力投入阶段，相对较离散，正是技术和治理能力颠覆的最佳发力点。我们结合多年政府各个部门及各类企业数据治理项目的经验，提出数据治理项目开展过程中数据治理平台应具备四大能力：聚、治、通、用，以及项目实施总体指导思想：PDCA（Plan-Do-Check-Act，计划、执行、检查和行动）。

数据治理平台的四大能力建设说明如下。

- 聚：数据汇聚能力，面对数据来源各异、数据类型纷繁多样、数据时效要求不一等各类情况，数据治理首先能把各类数据接入平台中，“进的来”是第一步。
- 治：狭义的数据治理能力，包括数据标准、数据质量、元数据、数据安全、数据生命周期、主数据。核心是保证数据标准的统一、借助元数据掌握数据资产分布情况及影响分析和血缘关系、数据质量的持续提升、数据资产的安全可靠、数据资产的淘汰销毁机制以及核心主数据的统一及使用。
- 通：数据拉通整合能力，原始业务数据分散在各业务系统中，数据组织以满足业务流转为前提。后续数据需求是根据实际业务对象开展的而非各业务系统，所以需要根据业务实体重新组织数据。比如政府单位针对人的综合分析通常会涉及财产、教育程度、五险一金、缴税、家庭成员等，需要以身份证号拉通房管局、交通局、教育局、人社局、税务局、卫健委等多个委办局数据。数据拉通整合能力是后续满足多样化需求分析的基础，是数据资产积累沉淀的根基，也是平台建设的另一个重点。
- 用：数据服务能力，数据资产只有真正赋能于前端业务才能发挥实际效用，所以如何让业务部门快速找到并便利地使用所需数据资产是数据治理平台的另一项核心能力。

项目实施总体指导思想 PDCA 包括如下 4 个方面。

- P：Plan，标准、规划、流程制定。
- D：Do，产品工具辅助落地。
- C：Check，业务技术双重检查保证。

- A：Action，持续优化提升数据质量及服务。

结合数据治理项目实际落地实施过程以四大能力构建、PDCA 实施指导思想提出了 PAI 实施方法论，即流程化（Process-Oriented）、自动化（Automation）、智能化（Intelligence）三化论，以逐步递进方式不断提升数据治理能力，为政府和企业后续的数据赋能业务及数据催生业务创新打下坚实基础。

- 流程化将数据治理项目执行过程进行流程化梳理，同时规范流程节点中的标准输入输出，并将标准输入输出模板化。另外，对各流程节点的重点注意事项进行提示。
- 自动化针对流程化之后的相关节点及标准输入输出进行自动化开发，减轻人力负担，让大家将精力放在业务层面及新技术拓展上，避免重复人力工作，如自动化数据接入及自动化脚本开发等。
- 智能化针对新项目或新领域，结合历史项目经验及沉淀给出推荐内容，比如模型创建、数据质量稽核规则等。

1. 数据治理流程化

因数据治理类项目通常采用瀑布式开发模式，核心流程包含需求、设计、开发、测试、上线等阶段，流程化是将交付流程步骤进行详细分解并对项目组及客户工作内容进行提炼及规范，明确每个流程的标准输入、输出内容。其中需求、概要设计和详细设计为执行过程中的核心流程节点，将针对这三部分进行详细讲解。

1）需求调研

（1）需求调研流程

数据调研是整个项目的基础，既要详细掌握业务现状及数据情况，又要准确获取客户需求，明确项目建设目标。数据调研总体分成三个大的时间节点，包括需求调研准备、需求调研实施及需求调研后期的梳理确认。

需求调研准备包括调研计划确定、调研前准备，具备条件的尽量开一次调研需求见面会（项目启动会介绍过的可以不需要再组织）。其中调研前准备需对客户的组织架构及业务情况进行充分的了解，以便在后续的调研实施阶段有的放矢，使调研内容更为翔实，客户需求把控更为准确。

调研实施阶段一般组织两轮调研，第一轮主要是了解业务运转现状、对接业务数据以及客户需求。第二轮针对具体的业务和数据的细节问题进行确认，以及分析后的客户需求与客户确认。对于部分系统的细节问题以线下方式对接，不再做第三轮整体调研。需求调研后期主要是针对客户需求、客户业务及数据现状进行内外部评审并确认签字，以《需求规格说明书》形式明确本期项目建设目录。

（2）需求调研工作事项

表 1-2 描述了需求调研过程关键节点的客户方、项目组工作内容及输入输出，并说明了需求调研阶段的总体原则、调研方式及相关要求。

表1-2 需求调研工作事项

工作项	执行人	工作内容	输入	输出	备注
业务系统初次调研	项目组	负责对模板内容进行培训讲解	《源系统技术侧调研填报模板V1.0》 《源系统业务侧调研模板V1.0》 “问题-系统级”工作表	《源系统技术侧调研填报模板V1.0》 《源系统业务侧调研模板V1.0》 各系统《业务流程图》 各系统《管理流程图》	
	客户方	1. 协调各业务系统管理人员参加模板讲解培训 2. 各业务系统管理人员组织认真填写下发的数据普查模板，按时反馈			
业务系统二次调研	项目组	1. 项目组对收集回来的模板信息进行整理、汇总，并对表、字段级问题进行整理 2. 对疑问进行再次调研、沟通 3. 查看系统真实数据情况	《源系统技术侧调研填报模板V1.0》 《源系统业务侧调研模板V1.0》	《源系统业务侧调研模板V1.0》	源系统技术、业务调研表内容整理，并整理业务调研表级及字段级问题，系统真实数据查看；每个系统真实数据查看问题汇总预计3~5天，再次调研预计1~3天
	客户方	协调对应系统负责人配合项目答疑，讲解数据现状			
系统需求调研	项目组	1. 根据招投标文件整理需求调研模板，细化需调研的问题（如数据接入系统范围） 2. 对系统使用人员进行需求访谈（业务、管理、技术人员） 3. 制定需求追踪矩阵 4. 编制《需求规格说明书》 5. 根据业务需求调整原型	《招标文件》 《投标文件》 《需求调研模板V1.0》系统原型	《需求调研模板》 《需求追踪矩阵模板V1.0》 《需求规格说明书》系统原型	1. 总体原则： a. 需求“宽进严出”，本期不做内容可做二期三期需求输入 b. 我们做方向引导，客户多说，争取把业务现状、问题、改善方案都讲出来 2. 调研方式：问卷、访谈、系统/原型试用等 3. 要求：调研录音、访谈纪要或其他总结内容
	客户方	1. 协调后期系统使用人员配合项目需求调研 2. 确认需求优先级及本期范围			
需求评审及确认	项目组	准备评审所需的材料	《需求规格说明书》系统原型	《需求规格说明书》 需求确认签字 系统原型 《评审会议纪要》	各功能项在需求规格说明书中

（3）需求调研注意事项

具体的需求调研有如下注意事项。

- 需求收集：关键干系人需求，真正用户是谁及其需求，需求获取前置问题，包括客户管什么、重点关注什么、目前如何管理、欠缺什么、重复劳动有哪些。
- 需求验证：3W 验证，包括谁来用、什么场景下用、解决哪些问题以及原型草图。
- 需求管理：核心需求（需求需融入业务流程并发挥实际效用），识别是否行业共性（有余力则做，没有就算了，项目管理角度不需要，行业角度需要）。
- 需求确认：形成文字版需求规格说明书，务必签字确认（后续可以更改，大变更需记录）。

2）概要设计

数据治理项目概要设计主要涵盖网络架构、数据流架构、标准库建设、数据仓库建设 4 部分内容。总体目标是明确数据如何进出数据治理平台（明确网络情况）、数据在平台内部如何组织及流动（数据流架构及数据仓库模型）以及数据在平台内部应遵循哪些标准及规范（标准库）。每部分具体工作事项及输入、输出如表 1-3 所示。

表1-3 概要设计的组成

工作项	执行人	工作内容	输 入	输 出	备 注
网络架构	项目组	1. 根据原型、需求及现有网络情况进行网络架构设计 2. 确认硬件规划及部署方案	现有网络情况后续使用人员及访问方式	《网络架构图》 《集群硬件规划及部署方案》	总体输出《概要设计说明书模板V1.0》
	客户方	1. 提供现有网络情况 2. 后续平台使用人员及访问方式统计			
数据流架构	项目组	根据原型及需求进行数据流设计	《需求规格说明书》系统原型	《数据流程图》 《实时数据标准接入方案》	
	客户方	评审并确认整体数据流程			
标准库范围	项目组	根据需求规格说明书及系统建设过程中涉及的标准收集标准	已有标准	数据库设计与运行管理标准 共享交换标准 接口管理标准 数据元标准 资源目录体系	
	客户方	提供现有系统中已有标准（包括引用标准和内部使用标准）			
仓库模型	项目组	1. 根据业务输入确认核心主题域 2. 梳理业务逻辑，确认核心实体	《源系统业务侧调研模板V1.0》 《源系统技术侧调研填报模板V1.0》 各系统《业务流程图》	《业务架构图》 全部主题域及关键实体CDM	
	客户方	1. 评审并确认主题域主题 2. 评审并确认核心实体逻辑模型			

（1）网络架构

网络架构要明确硬件部署方案、待接入系统网络情况、后续使用人群及访问系统方式，以便满足数据接入及数据服务需求。

（2）数据流示意图

数据流架构要明确各类数据的处理方式及流向，以便确认后续数据加工及存储方式。

（3）数据标准内容示意图

数据标准内容示意图如图 1-6 所示，标准库建设要明确平台所遵循的各类标准及规范，以保证平台建设过程的统一规范，为后续业务赋能打下坚实基础。

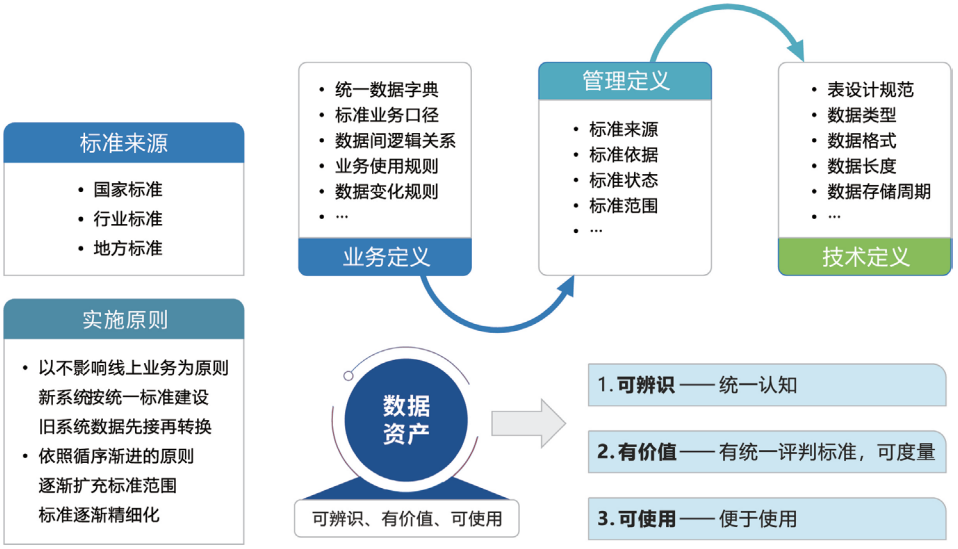

图 1-6 数据标准内容示意图

（4）数据仓库主题域及核心实体示意图

数据仓库建设要明确主题域及关键实体，明确后续数据拉通整合的实体对象，以更好地支撑繁杂多变的数据需求。

原始库是数据仓库和业务系统对接的缓冲，负责数据的抽取，建设时要使用贴源模型作为理论支持，即保证以源系统字段为基础，只增加处理时间的控制字段等，不对数据表结构做任何修改，不做逻辑计算，以保证源系统的数据能够准确、完整、高效地接入数据仓库中。原始库中的数据一般采用分区保存，数据存储时效较短，一般数据保留一周（按日分区或一天一张表）。

资源库从原始库抽取数据，数据结构保持不变，保留全部历史数据记录，会从数据存储空间及后续数据使用角度对数据的存储进行确认，如使用增量、历史拉链等方式记录历史数据。同时，要对数据做标准化以及数据质量的清洗，对于数据质量符合要求的，会放到资源库中存储，不符合要求的会放到问题库，用以反馈给业务系统进行问题处理和数据的清洗。

主题库要求按照主题的方式进行数据处理，数据从资源库获取，把针对同一主题的、联系

较为紧密的数据集中到一起。主题的设计要遵循行业共识，应优先使用行业内通用、具有共识的主题结构，对于没有通用行业主题规范的，应由最终用户和数据仓库设计人员共同设计完成。主题库的建设要遵循主题域模型，数据按照主题进行归纳；数据结构按照三范式的规范对不同来源的数据进行整合，重新进行设计，并且要求保持数据结构的扩展性和兼容性。对于常用的数据公共属性，会补充相应的字段属性做冗余处理，比如获取地区信息后可以补充相对应的省份名称和编码。

专题层要按照业务需求定制开发，数据从主题层获取。专题库下的一个专题至少对应一个具体的应用。专题库常用的是以维度模型存储指标数据，维度模型的结构清晰、条理简单，易于业务任务理解和使用，建设维度模型只需根据业务需求梳理维度和分析指标，即可构建对应的维度表和事实表，常见的维度模型结构为星型模式，数据专题示意图如图 1-7 所示。

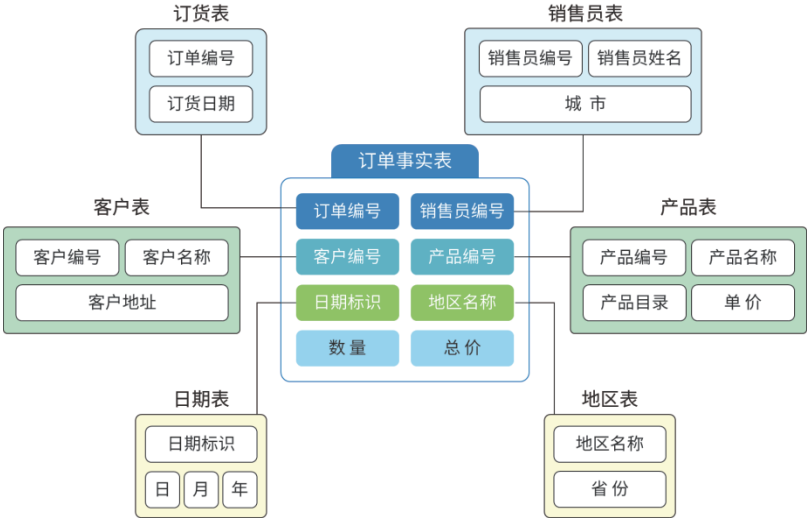

图 1-7　数据专题示意图

配置库是独立于上面 4 个库之外的一个共性库，用于存储整个数据治理过程中共性使用的信息，保证数据治理过程中规范、标准的统一。比如标准代码库，负责维护数据治理过程中需要使用的标准代码，这些代码在资源库的数据清洗、质量稽核以及专题库维度模型的构建中都要使用。

实时数据区的数据结构与业务系统相同，数据时效性要求较高，采取实时及准实时的方式将数据从业务系统中同步进行相应的加工计算，直接为应用提供数据支持。

3）详细设计

详细设计针对项目实际落地的工作模块分别进行设计，明确每部分实现的设计，具体模块、工作内容、输入、输出如表 1-4 所示。

表1-4 详细设计的组成

工作项	执行人	工作内容	输 入	输 出	备 注
数据标准设计	项目组	1. 获取数据元标准（命名、类型、精度、代码值等信息） 2. 与信息部门沟通集成技术，根据讨论结果制定集成标准	《字段命名自动化模板V1.0》 《标准代码表模板V1.0》	《字段命名自动化模板V1.0》 《标准代码表模板V1.0》	输出包含：PDM、标准命名词库、字段匹配、词根命名（标准命名获取包含分词、翻译，最终形成标准命名词库，需梳理标准数量不同、工期不同；目前未包含：共享交换、目录、接口等标准，评审确认后形成终版
	客户方	协调信息部门技术人员参与集成技术、方案沟通讨论、标准制定			
批量数据接入设计	项目组	1. 配置接入模板 2. 配置工作流	《源系统技术侧调研填报模板V1.0》	《数据接入模板V1.0》 《工作流配置模板V1.0》	目前批量数据库对接方式可以直接批量生成（包含工作流调度），其他方式目前需要手动配置，会同步生成初始化全量脚本
	客户方	确认初始化方案（时间、方式），确认日常数据抽取方案	《数据连接方式清单》		
实时数据接入设计	项目组	1. 统计实时接入清单并确认接入方案 2. 画实时数据流程图	《源系统技术侧调研填报模板V1.0》 《实时数据标准接入方案》	《实时接入清单》 《实时接入数据流程图》	提供实时接入标准方案及示例代码，通过配置即可实现数据接入汇总
	客户方	确认历史数据接入方案，接入频率	《数据连接方式清单》		
模型设计	项目组	1. 梳理业务逻辑确认逻辑模型 2. 自动生成物理模型	《源系统业务侧调研模板V1.0》 《源系统技术侧调研填报模板V1.0》 《字段命名自动化模板V1.0》	LDM、PDM（PowerDesigner） 《数据元标准》 《建表DDL》	《数据元标准》代替《数据字典》
Mapping设计	项目组	设计Mapping关系，确认处理算法	逻辑模型 《Mapping关系模板V1.0》 《数据处理配置模板V1.0》	《Mapping关系模板V1.0》 《数据处理配置模板V1.0》	
	客户方	Mapping关系确认			

（续表）

工作项	执行人	工作内容	输 入	输 出	备 注
工作流设计	项目组	根据系统资源、业务优先级、各系统数据到达时间确认并发及优先级及依赖	《集群硬件规划及部署方案》 业务优先级	工作流设计	
	客户方	确认业务优先级			
数据质量稽核	项目组	1. 制定数据清洗方案 2. 配置数据质量稽核规则 3. 配置数据质量工作流	《数据清洗方案》 《字段命名自动化模板V1.0》 《标准代码表模板V1.0》 《Mapping关系模板V1.0》	数据清洗 《数据质量稽核规则库》 数据质量稽核工作流 《数据质量报告》	源系统数据最好可以修改，如果协调有问题可先记录，后期再修改
	客户方	1. 协调技术人员参与清洗方案制定 2. 协调技术人员进行指标加工核对 3. 确认数据稽核规则及是否阻断任务			
主数据	项目组	1. 主数据范围确认 2. 主数据标准制定 3. 主数据管理标准制定 4. 数据集成标准制定 5. 数据清洗	《源系统技术侧调研填报模板V1.0》 《标准代码表模板V1.0》 《数据连接方式清单》 《数据清洗方案》	《主数据范围表》 《主数据流程图》 《主数据标准》 《主数据管理标准》 《主数据系统集成标准规范》	
	客户方	1. 主数据范围确认 2. 协调业务人员，参与分类标准、编码标准和属性标准制定 3. 协调业务人员，参与数据维护、修改、变更流程、模板制定 4. 协调信息部门技术人员参与集成技术、方案沟通讨论、标准制定 5. 协调业务人员评审清洗方案，协调技术人员参与清洗方案制定			
数据安全	项目组	1. 梳理数据分级 2. 收集并梳理数据访问、加密、脱敏清单 3. 配置数据访问策略 4. 配置加密及脱敏规则 5. 数据安全管理标准制定	组织架构及角色清单 《数据字典》 《数据资源目录》	《数据分级》 《数据密级清单》 《加密存储清单》 《脱敏清单》 《数据安全管理标准》	
	客户方	1. 协调任务人员及管理人员确认数据分级 2. 协调业务人员提供数据访问、加密、脱敏清单 3. 组织业务人员统一并确认以上规则 4. 确认数据安全管理标准			

（续表）

工作项	执行人	工作内容	输 入	输 出	备 注
数据生命周期管理	项目组	1. 制定各表数据保留策略 2. 配置各表或层级清理配置 3. 制定数据生命周期管理标准制定	《数据字典》 《数据生命周期配置清单V1.0》 《集群硬件规划及部署方案》	《数据生命周期配置清单V1.0》 《数据生命周期管理逻辑架构图》 《数据生命周期管理标准》	
	客户方	协调业务人员，参与数据全生命周期管理标准制定及确认数据保留策略			
指标库加工	项目组	1. 收集指标及加工过程 2. 整理冲突、有歧义等指标 3. 整理最终指标库	指标集及指标加工规则	《原子指标库》	
	客户方	1. 协调业务人员提供指标加工业务口径 2. 协调技术人员提供指标加工技术口径 3. 组织业务人员统一并确认冲突或歧义指标业务口径计算规则			
数据资源目录设计	项目组	进行数据目录新增、修改、变更等管理标准制定	《需求规格说明书》 数据资源目录管理流程	《数据资源目录》 《数据资源目录管理标准》	
	客户方	协调业务人员，参与数据维护、修改、变更流程、模板制定			
数据共享交换设计	项目组	1. 进行数据新增、修改、变更等全生命周期管理标准制定 2. 梳理共享交换数据清单 3. 梳理每项数据入参、出参信息 4. 数据调用限制策略（每分钟调用数、黑白名单等） 5. 梳理共享交换权限体系	用户、角色清单 《数据共享交换清单》 数据共享交换配置信息	《数据交换配置清单》 《数据共享交换管理标准》 《数据开放共享数据权限配置清单》	
	客户方	1. 协调业务人员，参与数据维护、修改、变更流程、模板制定 2. 协调业务人员参与共享交换数据入参、出参信息制定及调用限制策略 3. 协调业务人员参与权限体系制定			

2. 数据治理自动化

在将数据治理项目流程化以后，整个工作内容及具体工作产出已经比较明确了，但是发现流程中会涉及大量的开发工作，同时发现很多工作具有较高的重复性或相似性，开发使用的流程及技术都是一样的，只是配置不同，因此针对流程化以后各节点的自动化开发应运而生。通过配置任务的个性化部分，然后统一生成对应的开发任务或脚本即可完成开发。

自动化处理一般有两种实现路径，其一是采购成熟数据治理软件，其二是自研开发相应工具。其中数据治理过程中可实现自动化处理的流程节点（如“工序”，标蓝色部分），如图 1-8 所示。

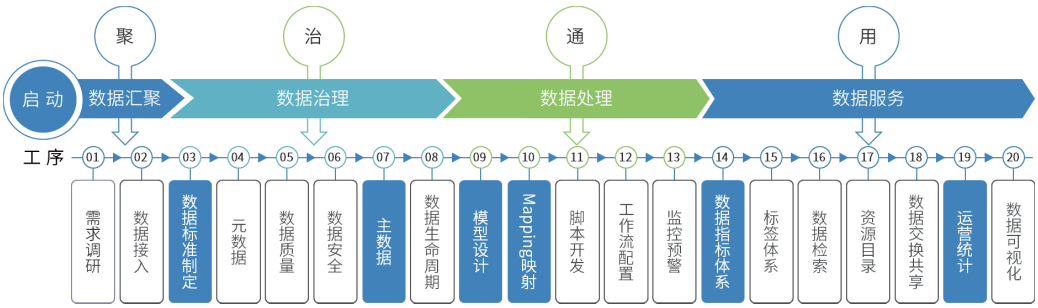

图 1-8　数据治理流程节点

对于需求调研、模型设计等流程节点，因为涉及线下的访谈、业务的理解，更多的是与人的沟通交流，进而获取相应的业务知识及需求，并非单纯的计算机语言，同时“因人而异”的情况也比较常见，所以这部分相关工作暂时还以人工为主。因数据接入、脚本开发及数据质量稽核在日常工作中占用时间较长，下面将详细讲解这三部分内容。

1）批量数据接入

数据接入是所有数据治理平台的第一步，批量数据接入占数据接入工作量的 70%~90%。自动化处理即将任务个性化部分进行抽象化形成配置项，通过配置任务的抽象化配置项，进而生成对应的任务。批量数据接入抽象以后的配置项如下。

- 源系统：源系统数据库类型。
- 源库名：源系统数据库库名称（数据库的链接方式在其他地方统一管理）。
- 源表名：源系统数据库库表名称。
- 目标系统：目标数据库类型。
- 目标库：目标数据库库名称。
- 目标表：目标数据库库表名。
- 增 / 全量：1 表示全量接，0 表示增量接。

使用 Sqoop、DataX 等方式可以批量生成对应命令或配置文件，实现批量生成接入作业，实现自动化数据接入工作，数据接入效率提升 75% 以上，后续只需验证数据接入正确性即可。对于数据接入来说，大数据平台主流的开源工具有 Sqoop、DataX、Kettle 等。

（1）Sqoop

Sqoop 可以理解为连接关系数据库和 Hadoop 的桥梁，主要有两个方面的能力：①将关系数据库（如 MySQL、Oracle、Postgres 等）的数据导入 Hadoop 及其相关的系统中，如 Hive 和 HBase；②将数据从 Hadoop 系统中抽取并导出到关系数据库。Sqoop 工具的底层工作原理本质上是执行 MapReduce 任务。由于实现方式是 MapReduce 任务，因此具体到接入任务控制，可以高效、可控地利用资源，可以通过调整任务数来控制任务的并发度，其采集效率也很高。Sqoop 整体围绕 Hadoop 生态建立，必须依赖 Hadoop，无法独立运行，因此整体服务相对较重。

（2）DataX

DataX 是阿里开源的一个异构数据源离线同步工具，主要用于实现包括关系数据库（MySQL、Oracle 等）、HDFS、Hive、ODPS、HBase、FTP 等各种异构数据源之间稳定高效的数据同步功能。DataX 采用 Framework +Plugin 架构构建，将数据源读取和写入抽象成为 Reader/Writer 插件。这样每接入一套新数据源，该新加入的数据源即可实现和现有的数据源互通。DataX 具有简单化、轻量化、易扩展的特点，因此很多公司的数据平台都在使用，可进行相关的数据源扩展，并作为 ETL 工具的后端引擎集成。

其核心原理如下：

- DataX 完成单个数据同步的作业，我们称之为 Job，DataX 接收到一个 Job 之后，将启动一个进程来完成整个作业的同步过程。DataX Job 模块是单个作业的中枢管理节点，承担了数据清理、子任务切分（将单一作业计算转换为多个子 Task）、TaskGroup 管理等功能。
- DataX Job 启动后，会根据不同的源端切分策略，将 Job 切分成多个小的 Task（子任务），以便于并发执行。Task 便是 DataX 作业的最小单元，每一个 Task 都会负责一部分数据的同步工作。
- 切分多个 Task 之后，DataX Job 会调用 Scheduler 模块，根据配置的并发数据量，将拆分的 Task 重新组合，组装成 TaskGroup（任务组）。每一个 TaskGroup 负责以一定的并发方式运行分配好的所有 Task，默认单个任务组的并发数量为 5。
- 每一个 Task 都由 TaskGroup 负责启动，Task 启动后，会固定启动 Reader → Channel → Writer 的线程来完成任务同步工作。
- DataX 作业运行起来之后，Job 监控并等待多个 TaskGroup 模块任务完成，等待所有 TaskGroup 任务完成后 Job 成功退出。

（3）Kettle

Kettle 是一款基于 Java 的开源的 ETL 工具，它允许管理来自不同数据库的数据，把各种数据放到一个壶里，然后以一种指定的格式流出。Kettle 提供一个图形化的用户操作环境，使用比较方便和简单，具有数据迁移、文件解析、数据关联比对、数据清洗转换能力。由于操作比较方便和简单，并且具有完全可视化的页面，很多厂商的 ETL 工具会基于 Kettle 定制。

以上内容主要针对的是离线接入的场景。在实际项目中，还要考虑到数据实时接入的场景，或者数据提供方不允许使用 SQL 方式访问数据库，防止对原有业务产生影响，这时候就会涉及

CDC（Change Data Capture，变更数据捕获）接入技术，CPC 技术是数据库领域非常常见的技术，主要用于捕获数据库的一些变更，然后把变更数据发送到下游。通常业内聊的 CDC 技术主要是基于数据库日志解析进行实现的。由于是基于日志解析实现的，不同的数据库的形式并不一致，因此没有一个引擎可以处理所有的数据库。目前常用的方案主要通过 Canel 实现 MySQL 的 CDC 能力，通过 Logminer 实现 Oracle 的 CDC 能力。以 Canel 的实现方式为例，其本质是模拟 MySQL Slave 的交互协议，伪装自己为 MySQL Slave ，向 MySQL Master 发送 dump 协议，MySQL Master 收到 dump 请求，开始推送 Binlog 给 Slave（即 Canal），最后 Canal 解析 Binlog 对象。很多数据库 CDC 都是采用这种方案进行实现的。

这里我们重点介绍的是 Debezium 组件，Debezium 是 Apache Kafka Connect 的一组源连接器，使用 CDC 从不同的数据库中获取更改。它可以对接 MySQL、PostgreSQL、SQL Server、Oracle、MongoDB 等多种 SQL 及 NoSQL 数据库，把这些数据库的数据持续以统一的格式发送到 Kafka 的主题，供下游进行实时消费。与 Canel、Logminer 等组件不同，Debezium 支持多类型的数据库，因此，很多公司将其作为基础引擎进行选型。

此外，随着实时处理技术的发展，Flink 也拥有 CDC 能力，Flink CDC 底层封装了 Debezium，支持通过 SQL 方式实现数据库的实时接入。

百分点的 CDC 接入技术底层基于 Debezium 进行日志解析，将数据接入分为数据读取和数据写入两个阶段，全流程可视化地实现 CDC 实时接入。此外，还具有以下特性：

- 支持分布式和高可用的部署方式，在数据量增长的情况下可以快速扩容来处理数据。
- 支持多种接入模型，如增量、全量、断点续传等。
- 提供友好的可视化支持，不需要了解底层技术即可实现 CDC 任务的配置。

2）脚本开发

资源库、主题库的加工脚本占整体开发工作的 50%~80%，同时经过对此部分数据加工方式进行特定分析，结合 Mapping 文档，选定以上数据处理方式的一种即可自动生成资源库或主题库对应的脚本，开发效率得到大幅度提升，整体效率提升 60% 以上（模型及 Mapping 设计尚需人工处理）。

3）数据质量

数据质量是指在业务环境下，数据符合消费者的使用目的，能满足业务场景具体需求的程度。数据质量一般由完整性、有效性、正确性、唯一性、及时性和合理性等特征来描述。

- 完整性是指数据信息是否完整，描述的数据、属性及关系存在或不存在，包括但不限于记录完整性、属性完整性、关系完整性等。
- 有效性，是指数据内容是否遵循了标准规范，满足一定的格式，或符合标准定义的值域要求。
- 正确性，是指数据内容是否符合预定的逻辑要求，如年龄字段不能取负数。
- 唯一性，是指数据信息是否唯一，包含数据的主键信息是否唯一，数据信息的部分或全部是否唯一。

- 及时性，是指及时记录和传递相关数据，满足业务对信息获取的时间要求。
- 合理性，是指数据信息是否符合预定义的业务逻辑要求。

数据质量是数据价值的根本。基于上述特征，数据质量管理贯穿着数据治理的全过程，并且不是一蹴而就的，不同的处理阶段对质量控制的要求也有所不同，表 1-5 所示的内容供读者参考。

表1-5 质量控制要求表

评估项	数据接入	数据处理
记录完整性	√	
属性完整性	√	√
关系完整性	√	√
格式有效性		√
值域有效性		√
接入及时性	√	
更新及时性		√
主键唯一性		√
数据唯一性		√
数据正确性		√
业务合理性		√

数据质量是 PDCA 实施总体指导思想的关键一步，是发现数据问题以及检查数据标准规范落地的必需环节。针对具体的规则，可以通过产品和自助开发来实现，只需进行相应配置即可实现自动化检查。

4）元数据

元数据，最常见的定义就是“描述数据的数据”。元数据管理是对元数据的创建、存储、整合、控制的一整套流程，目的是支持基于元数据的相关需求和应用。通过元数据管理能够让开发和业务人员快速地了解数据的上下游关系及本身的含义，精准定位需要查找的数据，减少数据研究的时间成本，提高效率。

常见的开源元数据管理工具包括 Apache Atlas、LinkedIn DataHub 等。

- Atlas 元数据治理框架，可以为 Hive、HBase、Kafka 等提供元数据管理功能，它为 Hadoop 集群提供了包括数据分类、集中策略引擎、数据血缘、安全和生命周期管理在内的元数据治理核心能力。Atlas 可以帮助企业构建其数据资产目录，对这些资产进行分类和管理，并为数据分析师和数据治理团队提供围绕这些数据资产的协作功能。但其主要管理对象还是围绕 Hadoop 生态的大数据组件，对企业各类业务系统使用的传统数据库生态并不支持。
- DataHub 是由 LinkedIn 的数据团队开源的一款提供元数据搜索与发现的工具，能够从不同的平台（比如 MySQL、Airflow、Superset）将元数据同步到 DataHub，实现了从数据源到 BI 工

具的全链路的血缘打通。DataHub 提供统一的元数据搜索和治理，能降低开发人员的数据探索复杂性。

对于元数据管理系统来说，可以从元数据采集、元数据管理、元数据分析三个方面进行评估。

（1）元数据采集

从各种工具中，把各种类型的元数据采集进来；对于数据科学平台来说，元数据采集一般是从源系统接入的。如果企业已经拥有数仓，那么数仓也可以看作元数据采集的一个数据源，将已有的元数据从数仓接入，还未接入的从源系统接入。

（2）元数据管理

实现元模型统一、集中化管理，提供元模型的查询、增加、修改、删除、元数据关系管理、权限设置等功能，让用户直观地了解已有元模型的分类、统计、使用情况、变更追溯，以及元数据间的血缘联系。其中包含如下几个重要能力。

- 元数据存储：使用相应的存储策略来对元数据进行存储，要求在不改变存储架构的情况下扩展元数据存储的类型。
- 元数据维护：元数据维护就是对信息对象的基本信息、属性、被依赖关系、依赖关系、组合关系等元数据的新增、修改、删除、查询、发布等操作，支持根据元数据字典创建数据目录，打印目录结构，根据目录发现、查找元数据，查看元数据的内容。元数据维护是基本的元数据管理功能之一，技术人员和业务人员都会使用这个功能来查看元数据的基本信息。
- 元数据审核：元数据审核主要是审核已采集到元数据仓库中但还未正式发布到数据资源目录中的元数据。审核过程中支持对数据进行有效性验证并修复一些问题，例如缺乏语义描述、缺少字段、类型错误、编码缺失或不可识别的字符编码等。
- 元数据版本管理：在元数据处于一个相对完整、稳定的时期，或者处于一个里程碑结束时期，可以对元数据定版以发布一个基线版本，以便日后对存异的或错误的元数据进行追溯、检查和恢复。
- 元数据变更管理：用户可以自行订阅元数据，当订阅的元数据发生变更时，系统将自动通知用户，用户可根据指引进一步在系统中查询变更的具体内容及相关的影响分析。

（3）元数据分析

基于采集的各类元数据，通过元数据存储策略提供统一的数据地图，帮助用户全面盘点和整理数据资产。其重要能力通常包含数据资产地图、血缘分析和影响分析。

- 数据资产地图：按数据域对企业数据资源进行全面盘点和分类，并根据元数据字典自动生成企业数据资产的全景地图。该地图可以告诉你有哪些数据，在哪里可以找到这些数据，能用这些数据干什么。同时，也可以展现数据与其他数据的关系，以及它们的关系是怎样建立的。关联度分析是从某一实体关联的其他实体及其参与的处理过程两个角度来查看具体数据的使用情况，形成一张实体和所参与处理过程的网络，如表与 ETL 程序、表与分析应用、表与其他表的关联情况等。

- 血缘分析：血缘分析会告诉你数据来自哪里，经过了哪些加工。其价值在于当发现数据问题时，可以通过数据的血缘关系追根溯源，快速定位到问题数据的来源和加工过程，减少数据问题的复杂度。
- 影响分析：影响分析会告诉你数据去了哪里，经过了哪些加工。其价值在于当发现数据问题时，可以通过数据的关联关系向下追踪，发现谁在使用这些数据，降低问题带来的影响。

最后，元数据管理是数据治理的关键，我们要知道数据的来龙去脉，才能对数据进行全方位的管理、监控和洞察。从自动化的角度，元数据的范围也是贯穿数据治理的整个过程的，在这个过程中，元数据管理除针对数据模型的元数据管理外，也可以针对上述实现的数据接入任务、数据开发任务、DQC 任务的元数据进行识别，并且实现全链路的血缘智能识别和关联。例如，对接入任务的上下游识别、SQL 的血缘自动解析、QQC 及调度的资源关联等。由此可以实现元数据的管理和分析能力。

3. 数据治理智能化

经过自动化阶段后，数据治理流程中的数据仓库模型设计、Mapping 映射等阶段依旧有非常多人工处理工作，这些工作大部分跟业务领域知识及实际数据情况强相关，依赖专业的业务知识和行业经验才可进行合理的规划和设计。如何快速精通行业知识和提升行业经验是数据治理过程中新的“拦路虎”。如何更好地沉淀和积累行业知识，自动提供设计和处理的建议是数据治理“深水区”面临的一个新挑战。数据治理智能化将为我们的数据治理工作开辟一个 “新天地”。

在整个数据治理流程中，智能化发挥作用的节点（如“工序”，标红色部分）如图 1-9 所示。

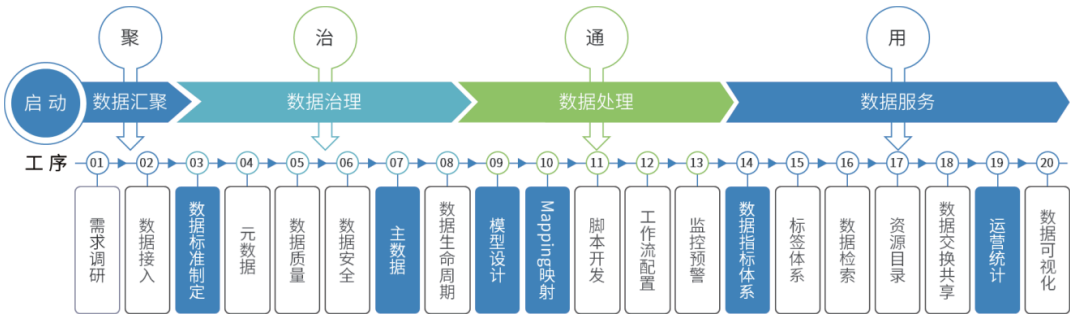

图 1-9 数据治理流程节点

实现智能化的第一步是积累业务知识及行业经验，形成知识库。数据治理知识库应包括标准文件、模型（数据元）、DQC 规则及数据清洗方案、脚本数据处理算法、指标库、业务知识问答库等，具体涵盖如下内容。

- 标准文件：在 2B 和 2G 行业，尤其是 2G 行业，国家、行业、地方都发布了大量的标准文件，在业务和技术层面都进行了相关约束，并对新建业务系统的开发提供了指导。标准文件知识库涵盖几个方面：① 国标、行标、地标等标准的在线查看，② 相关标准的在线全文检索，③ 标准具体内容的结构化解析。

- 数据元（模型）：对于不同行业来说，技术标准中的命名以及模型是目前大家都比较关注的，也是在做数据中各类项目以及数据治理项目比较耗时的地方，在金融领域已经比较稳定的主题模型在其他行业尚未形成统一，所以做 2B 和 2G 市场的企业如何能沉淀出特定行业的数据元标准甚至是主题模型，对于行业理解及后续同类项目交付就至关重要。具体包括实体分类、实体名称、中文名称、英文名称、数据类型、引用标准等。
- DQC（Data Quality Control，数据质量稽核）& 数据清洗方案数据治理的关键点是提升数据治理，所以不同行业及各个行业通用的数据质量清洗方案及数据质量稽核的沉淀就尤为重要，比如通用规则校验，身份证号为 18 位校验（15 转 18），手机号为 11 位校验（如有国际电话需加国家代码），以及日期格式、邮箱格式等。
- 脚本开发：在数据类项目中，数据 Mapping 确认以后就是具体的开发了，由于数据处理方式的共性，可以高度提炼成特定类型的数据处理，比如交易流水一般采用追加的方式，每日新增数据加入进来即可。此过程中的步骤都可以通过自动化程序来实现，同时借助上面沉淀的具体标准内容，进一步规范化脚本开发。
- 指标库：对于一个行业的理解一定程度上体现在行业指标体系的建立，行业常用指标是否覆盖全、指标加工规则是否有歧义是非常重要的两个考核项，行业指标库的建立对于业务知识的积累至关重要。
- 业务知识问答库：行业知识积累的最直观体现是业务知识问答库的建立，各类业务知识都可以逐步沉淀到问答库中，并以问答等多种交互方式更便利地服务于各类使用人员。比如生态环境领域 AQI 的计算规则，空气常见污染因子、各类污染指标的排放限值等，都能够以问答的形式进行沉淀。基于以上知识的不断沉淀积累，在数据治理开展过程中即可进行智能化推荐。在做实体及属性认定时，结合 NLP 技术和知识库规则即可进行相似度认定推荐。并且随着行业知识的不断积累和完善，后期可以直接推荐行业主题模型及主数据模型，以及针对实体及属性的数据标准、数据质量检查规则。

流程化是数据治理工作开展的第一步，是自动化和智能化的基础，将数据治理各节点开展过程中用到的内容进行梳理并规范，包括业务流程图、网络架构图、业务系统台账等，行业知识梳理完善以后形成行业版知识（抽离通用版），如标准文件梳理，包括代码表整理和数据元标准整理（数据仓库行业模型对应标准梳理）。

自动化是将流程化标准后的工作进行自动化开发，涉及仓库模型设计、标准化、脚本开发、DQC、指标体系自动化构建，包括自动化程序生成和自动化检查。自动化程序生成一是解放生产力，提高效率；二是提升开发的规范化。自动化检查包括：

- 发现数据问题，出具质量报告（唯一性、空值等通用问题）。
- 行业知识检查（行业版内置，不同行业关注的重要数据问题，并且会不断完善知识库）。

智能化是在流程化、自动化基础上针对数据拉通整合、主题模型、数据加工检查给出智能化建议，减少人工分析的工作。

总体思路是先解决项目上的标准化执行问题，然后提升建设效率及处理规范化问题（自动

化处理），最后基于业务知识的沉淀最终实现全流程智能化构建。

1.2.3 结构化数据分析

近年来，随着互联网技术的日益发展，用户越来越多地使用多媒体数据进行信息的传输和交流。这为数据科学技术带来了新的挑战。除文本、日志、表格等传统的数据格式外，数据科学的相关技术还必须支持语音、图像以及视频等多媒体数据。如何从这些多媒体数据中获取有用的信息，并且对其进行结构化处理，成为数据科学技术不可缺的组成部分。

1. 结构化与非结构化数据

在信息社会，信息可以划分为两大类：一类信息能够用数据或统一的结构加以表示，我们称之为结构化数据，如数字、符号；另一类信息无法用数字或统一的结构表示，如文本、图像、声音、网页等，我们称之为非结构化数据。

具体而言，结构化数据可以使用关系数据库表示和存储，表现为二维形式的数据，即数据库。其一般特点是数据以行为单位，一行数据表示一个实体的信息，每一行数据的属性是相同的，比如企业 ERP、财务系统、医疗 HIS 数据库、教育一卡通、政府行政审批以及其他核心数据库等。非结构化数据是数据结构不规则或不完整，没有预定义的数据模型，不方便用数据库二维逻辑表来表现的数据，包括所有格式的办公文档、文本、图片、HTML、各类报表、图像和音频 / 视频信息等。相对于传统的在数据库中或者标记好的文件，由于非结构化数据的非特征性和歧义性，因此更难理解。

结构化数据与非结构化数据之间除存储方式不同外，最大的区别是分析两种类型数据的简便程度不同。用于分析结构化数据的工具模型以及方法已经较为成熟，但是用于挖掘分析非结构化数据的工具仍处于新生或发展阶段。

2. 结构化数据分析的常用模型

1）有监督学习

（1）分类模型

分类模型是指通过让机器学习与训练已有的数据，从而预测新数据的类别的一种模型方法。通常在已有分类规则标签，需要预测新数据的类别的情况下使用分类模型，如通过男士的年龄、长相、收入、工作等特征指标数据，预测女士是否见面等。分类模型的常见方法包括：

① 决策树。决策树就是一棵树，一棵决策树包含一个根节点、若干内部节点和若干叶节点；叶节点对应于决策结果，其他每个节点则对应于一个属性测试；每个节点包含的样本集合根据属性测试的结果被划分到子节点中；根节点包含样本全集，从根节点到每个叶节点的路径对应一个判定测试序列。因此，决策树的学习过程可以总结为特征选择、决策树生成和剪枝三个部分。

② KNN 模型。KNN 的分类原理是通过先计算待分类物体与其他物体之间的距离，再统计

距离最近的 K 个邻居，最后对于 K 个最近的邻居，它们属于哪个分类最多，待分类物体就属于哪一类。

K 近邻算法的三个基本要素：K 值的选择、距离度量和分类决策规则。

- K 值的选择：K 取值较小时，模型复杂度高，训练误差会减小，泛化能力减弱；K 取值较大时，模型复杂度低，训练误差会增大，泛化能力有一定的提高。因此，K 值选择应适中，K 值一般小于 20，可以采用交叉验证的方法选取合适的 K 值。
- 距离度量：KNN 算法是利用距离来度量两个样本间的相似度的。距离越大，差异性越大；距离越小，相似度越大。常用的距离表示方法有欧式距离、曼哈顿距离、闵可夫斯基距离、切比雪夫距离、余弦距离。在兴趣相关性比较上，角度关系比距离的绝对值更重要。
- 分类决策规则：KNN 算法一般使用多数表决方法，即由输入实例的 K 个邻近的多数类决定输入实例的类。这种思想也是经验风险最小化的结果。

③ SVM 模型。SVM 就是构造一个“超平面”，并利用“超平面”对不同类别的数进行划分，同时使得样本集中的点到这个分类超平面的最小距离，即分类间隔最大化。

④ 逻辑回归。逻辑回归是一个分类算法，它可以处理二元分类以及多元分类。逻辑回归的假设函数为：

$$h_\theta(X) = g(X_\theta) = \frac{1}{1+e^{-X\theta}}$$

其中 X 为样本输入，$h_\theta(X)$ 为模型输出，θ 为要求解的模型参数。设 0.5 为临界值，当 $h_\theta(X)>0.5$ 时，即 $x_\theta>0$ 时，y=1；当 $h_\theta(X)<0.5$ 时，即 $x_\theta<0$ 时，y=0。模型输出值 $h_\theta(X)$ 在 [0,1] 区间内取值，因此可以从概率角度进行解释：$h_\theta(X)$ 越接近 0，则分类为 0 的概率越高；$h_\theta(X)$ 越接近 1，则分类为 1 的概率越高；$h_\theta(X)$ 越接近临界值 0.5，则无法判断，分类准确率会下降。

⑤ 朴素贝叶斯。朴素贝叶斯分类（Naive Bayes Classifier，NBC）是以贝叶斯定理为基础并且假设特征条件之间相互独立的方法，先通过已给定的训练集，以特征词之间独立作为前提假设，学习从输入到输出的联合概率分布，再基于学习到的模型，输入 X 求出使得后验概率最大的输出 Y。

⑥ 人工神经网络。人工神经网络（Artificial Neural Network，ANN）简称神经网络（Neural Network，NN），是基于生物学中神经网络的基本原理，在理解和抽象了人脑结构和外界刺激响应机制后，以网络拓扑知识为理论基础，模拟人脑的神经系统对复杂信息的处理机制的一种数学模型。该模型以并行分布的处理能力、高容错性、智能化和自学习等能力为特征，将信息的加工和存储结合在一起，以其独特的知识表示方式和智能化的自适应学习能力引起各学科领域的关注。它实际上是一个由大量简单元件相互连接而成的复杂网络，具有高度的非线性，能够进行复杂的逻辑操作，具有非线性关系。神经网络分为以下三种类型的层。

- 输入层：神经网络最左边的一层，通过这些神经元输入需要训练观察的样本，即初始输入数据的一层。

- 隐藏层：介于输入与输出之间的所有节点组成的一层。帮助神经网络学习数据间的复杂关系，即对数据进行处理的层。
- 输出层：由前两层得到神经网络的最后一层，即最后结果输出的一层。

（2）回归分析模型

回归分析是一种预测性的建模技术，它研究的是因变量（目标）和自变量（预测器）之间的关系。这种技术通常用于预测分析时间序列模型以及发现变量之间的因果关系。例如，司机的鲁莽驾驶与道路交通事故数量之间的关系，最好的研究方法就是回归。回归分析的常用方法如下：

① 线性回归。线性回归是利用数理统计中的回归分析来确定两种或两种以上的变量间相互依赖的定量关系的一种统计分析方法，运用十分广泛。其表达形式为 y=w′x+e，e 为误差，服从均值为 0 的正态分布。在回归分析中，只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量，且因变量和自变量之间是线性关系，则称为多元线性回归分析。如果自变量是二次方及以上，就称为多项式回归。线性回归对数据的要求很高，要求自变量相互独立、残差呈正态分布、方差齐性。从业务场景来说，线性回归可以使用的业务场景很多，只要自变量和因变量都是连续变量就可以使用回归模型，如 GDP、淘宝双十一销售额等。

② 岭回归与 LASSO 回归。岭回归是一种专用于共线性数据分析的有偏估计回归方法，实质上是一种改良的最小二乘估计法，通过放弃最小二乘法的无偏性，以损失部分信息、降低精度为代价获得回归系数更为符合实际、更可靠的回归方法，对病态数据的拟合要强于最小二乘法。LASSO 的基本思想是在回归系数的绝对值之和小于一个常数的约束条件下，使残差平方和最小化，从而能够产生某些严格等于 0 的回归系数，得到可以解释的模型。将 LASSO 应用于回归，可以在参数估计的同时实现变量的选择，较好地解决回归分析中的多重共线性问题，并且能够很好地解释结果。

（3）时间序列模型

时间序列（时序）是指将同一个统计指标的数值按其发生的时间先后顺序排列而成的数列。时序分析的主要目的是根据已有的历史数据对未来进行预测。时序数据本质上反映的是某个或者某些随机变量随时间不断变化的趋势，而时序预测方法的核心就是从数据中挖掘出这种规律，并利用其对未来做出估计。时序预测在商业领域有着广泛的应用，常见的应用场景包括销量和需求预测、经济和金融指标预测、设备运行状态预测等。企业参考预测结果制定不同期限的生产、人员和运输计划，以及长期战略规划。时间序列分析的常用方法如下：

① 多元线性回归。时序预测最朴素的方法是使用多元线性回归。具体做法是将下一时期的目标作为回归模型中的因变量，将影响目标的相关因素作为特征。以销量预测问题为例，影响目标的相关因素包括当期的销量、下一时期的营销费用、门店属性、商品属性等。为了捕捉时间信息，我们同时从日期中提取时间特征。以月度频率数据为例，常用的时间特征包括：将月份转化为 0-1 哑变量，编码 11 个变量；将节假日转化为 0-1 哑变量，可简单分为两类，即“有

假日”和“无假日”；或赋予不同编码值，如区分国庆节、春节、劳动节等。

② 时间序列分解。时间序列分解（Time Series Decomposition）主要用于理解时序的特点，也可以用作预测。其基本思想是任何时序中的规律可以分为长期趋势变动（Trend）、季节变动（Seasonality，即显式周期，特点是固定幅度、长度的周期波动）和循环变动（Cycles，即隐式周期，周期长但不具严格规则的波动）。时间序列分解使用加法模型（Additive Model）或乘法模型（Multiplicative Model）将原始时序分解为上述三类规律和不规则变动。在对时序分解之后，我们就可以对这三个部分分别使用平滑模型或回归模型进行建模，最后将这三个部分根据加法或乘法模型合并，得到预测值。

③ 指数平滑。指数平滑法的原理是对时间序列的趋势和季节性建模。指数平滑法由布朗（Robert G. Brown）于 1959 年提出。布朗认为，时间序列的态势具有稳定性或规则性，所以时间序列可被合理地顺势推延，最近的过去态势在某种程度上会持续到最近的未来，所以将较大的权值放在最近的数据上。指数平滑法使用了所有时间序列历史观察值的加权和来对未来进行预测。其中，权值随着数据的远离，逐渐趋近为零，这意味着，越靠近的观察值，其对预测的影响越大，这与实际生活中随机变量的特性是相符的，故指数平滑法是一种有效而可靠的预测方法。

④ ARIMA 模型。ARIMA 模型的目的是对时间序列的自回归特征建模。ARIMA(p, d, q) 包括自回归过程（AR）、移动平均过程（MA）和差分项（I）。参数中，p 为自回归项数，d 为时间序列成为平稳时所做的差分次数，q 为移动平均项数。根据 AIC 准则选择最合适的（p, d, q）组合。ARIMA 模型的一般形式为

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-B)^d y_t = c+(1+\theta_1 B+\cdots+\theta_q B^q)e_t$$

其中，为延迟算子，表示时间序列的上一时刻；为差分的阶数；p 为自回归阶数；q 为移动平均阶数；c 为常数参数。

⑤ 机器学习。机器学习方法是多元线性回归方法的拓展，重点在于使用更加复杂的特征工程技巧提取时序特征。在创建完特征后，使用常用的算法进行拟合，例如 XGBoost 和 LightGBM。

2）无监督学习

（1）聚类模型

聚类算法是对大量未标注数据，按照数据的内在相似性将数据集划分为多个组，这些相似的组被称作簇。处于相同簇中的数据实例彼此相同，处于不同簇中的实例彼此不同。聚类和分类的区别主要如下。

- 聚类（Clustering）：是指把相似的数据划分到一起，具体划分的时候并不关心这一类的标签，目标就是把相似的数据聚合到一起。
- 分类（Classification）：是把不同的数据划分开，其过程是通过训练数据集获得一个分类器，再通过分类器来预测未知数据。

聚类的常用方法如下：

① K-Means 聚类。K-Means 算法的运算原理是每一次迭代都确定 K 个类别中心，将数据点归到与之距离最近的中心点所在的簇，将类别中心更新为它的簇中所有样本的均值，反复迭代，直到类别中心不再变化或小于某个阈值。

② DBSCAN 聚类。DBSCAN 通过检查数据集中每点的 Eps 邻域来搜索簇，如果点 p 的 Eps 邻域包含的点多于 MinPts 个，则创建一个以 p 为核心对象的簇；然后，DBSCAN 迭代地聚集从这些核心对象直接密度可达的对象，这个过程可能涉及一些密度可达簇的合并；当没有新的点添加到任何簇时，该过程结束。

③ 层次聚类。层次聚类的聚类原理是先计算样本之间的距离，每次将距离最近的点合并到同一个类。然后，计算类与类之间的距离，将距离最近的类合并为一个大类。不停地合并，直到合成一个类。其中类与类的距离的计算方法有最短距离法、最长距离法、中间距离法、类平均法等。比如最短距离法，将类与类的距离定义为类与类之间样本的最短距离。层次聚类算法根据层次分解的顺序分为自底向上和自顶向下，即凝聚的层次聚类算法和分裂的层次聚类算法。

④ 高斯混合聚类。高斯混合模型（Gaussian Mixed Model，GMM）指的是多个高斯分布函数的线性组合，理论上 GMM 可以拟合出任意类型的分布，通常用于解决同一集合下的数据包含多个不同的分布的情况（或者是同一类分布但参数不一样，或者是不同类型的分布，比如正态分布和伯努利分布）。

高斯混合聚类采用概率模型来表达聚类原型。GMM 用于聚类时，假设数据服从混合高斯分布（Mixture Gaussian Distribution），那么只要根据数据推出 GMM 的概率分布就可以了，GMM 的 K 个分量实际上对应 K 个 Cluster 。根据数据来推算概率密度通常被称作密度估计。

（2）降维

数据降维即为降低数据的维度，在机器学习领域中，所谓的降维是指采用某种映射方法将原高维空间中的数据点映射到低维度的空间中。降维可应用于很多业务场景中，它可以降低时间复杂度和空间复杂度，节省了提取不必要特征的开销，避免维度爆炸；还可以起到去噪的作用，去掉数据集中夹杂的噪声，例如信号数据处理中，通过降维操作提高数据信噪比来提高数据质量。此外，还可以利用降维实现数据的可视化和对样本进行特征提取。在有些业务场景下，我们可以利用因子分析等降维方法进行权重计算和综合评分。

降维的常用方法如下：

① 主成分分析（Principal Component Analysis，PCA）。构造原变量的一系列线性组合形成几个综合指标，以去除数据的相关性，并使低维数据最大限度保持原始高维数据的方差信息，核心是把给定的一组相关变量（维度）通过线性变换转换成另一组不相关的变量，这些新的变量按照方差依次递减的顺序排序。第一变量具有最大方差，称第一主成分，第二变量的方差次大，称第二主成分。PCA 本质上是将方差最大的方向作为主要特征，并且在各个正交方向上将数据“离

相关”，也就是让它们在不同正交方向上没有相关性。

② 线性判别（Linear Discriminant Analysis，LDA）。线性判别将高维的模式样本投影到最佳鉴别矢量空间，以达到抽取分类信息和压缩特征空间维度的效果，投影后保证模式样本在新的子空间有最大的类间距离和最小的类内距离，即模式在该空间有最佳的可分离性。线性判别通过一个已知类别的“训练样本”来建立判别准则，并通过预测变量来为未知类别的数据进行分类。

③ 因子分析（Factor Analysis，FA）。因子分析是指研究从变量群中提取共性因子的统计技术，因子分析可在许多变量中找出隐藏的具有代表性的因子。将相同本质的变量归入一个因子，可减少变量的数目，还可检验变量间关系的假设。下面对因子分析进行举例说明，从而对因子分析的作用有一个直观的认识。

④ LASSO 变量选择。LASSO 是由 1996 年 Robert Tibshirani 首次提出的，全称是 Least Absolute Shrinkage and Selection Operator。该方法是一种压缩估计。它通过构造一个惩罚函数得到一个较为精炼的模型，使得它压缩一些回归系数，即强制系数绝对值之和小于某个固定值；同时设定一些回归系数为零。因此，保留了子集收缩的优点，是一种处理具有复共线性数据的有偏估计。LASSO 本质上是基于 L1 正则化的原理，当输入的特征变量较多时，通过剔除相关性小的特征变量实现特征选择，减少输入的特征变量个数，从而达到降维的目的。LASSO 回归的基本原理如以下公式所示：

$$Q(\beta)=\|y-X\beta\|^2+\lambda\|\beta\|_1 \Leftrightarrow \operatorname{argmin}\|y-X\beta\|^2 \text{ s.t.}\sum|\beta_j|\leqslant s$$

上述公式的偏差绝对值求和即为 L1 正则化，L1 正则化构建了特征变量系数的稀疏矩阵，使得不重要的特征变量的权重系数为 0。

3. 结构化数据分析的常见流程

1）数据输入

（1）离线数据接入

离线数据接入包括如下类型的数据。

- CSV 类型的数据：所需算法包或函数为 pandas.read_csv。
- DAT 类型的数据：所需算法包或函数为 pandas.read_table。
- Excel 类型的数据：所需算法包或函数为 pandas.read_excel。
- TXT 类型的数据：所需算法包或函数为 open、deadlines、close。

（2）数据框数据接入

① MySQL 数据库。Python 数据库接口支持非常多的数据库，可以选择适合项目的数据库，例如 MySQL、PostgreSQL、Microsoft SQL、Informix、Interbase、Oracle、Sybase。所需算法包

或函数为 PyMySQL，它是在 Python 3.x 版本中用于连接 MySQL 服务器的一个库。连接数据库前需要准备的项目如下：一个 MySQL 数据库，并且已经启动；可以连接该数据库的用户名和密码。有一个有权限操作的 Database。

② Hive 数据库。所需算法包或函数包括 sasl、thrift、thrift_sasl、pyhive 等，其中 sasl 采用 0.2.1 版本，选择适合自己的版本即可。

2）探索性数据分析

（1）探索性数据分析的目的

探索性数据分析（Exploratory Data Analysis，EDA）是指对已有数据在尽量少的先验假设下通过统计、作图、制表、方程拟合、计算特征量等手段探索数据的结构和规律的一种数据分析方法，该方法在 20 世纪 70 年代由美国统计学家 J.K.Tukey 提出。传统的统计分析方法常常先假设数据符合一种统计模型，然后依据数据样本来估计模型的一些参数及统计量，以此了解数据的特征。但实际中往往有很多数据并不符合假设的统计模型分布，这导致数据分析结果不理想。探索性数据分析则是一种更加贴合实际情况的分析方法，它强调让数据自身“说话”，通过这种方法我们可以更加真实、直接地观察到数据的结构及特征，为后续建模提供重要的洞见。

（2）探索性数据分析方法

探索性数据分析的主要方法包括单变量分析、多变量相关性分析、多变量交叉分析。

① 单变量分析。单变量分析是使用统计方法对单一变量进行描述，捕捉其特征。从统计学的角度来看，单变量分析就是对统计量的估计过程。

- 频率和众数：频率（Frequency）可以简单定义为属于一个类别的观测值占总数据的比例，这里类别对象可以是分类模型中不同的类，也可以是一个区间或一个集合。众数（Mode）是指具有最高频率的类别对象。
- 百分位数：对于有序数据，百分位数（Quantile）是一个重要的统计量。给定一个变量 x，x 的 p 百分位数 x_p 满足：这个变量有的 x% 数据小于 x_p。百分位数能让我们了解数据大小分布情况。
- 中心度量：均值和中位数。对于连续数据，均值（Mean）和中位数（Median）是度量数据中心最常用的统计量，其中中位数即 50% 百分位数。均值对数据中的离群点比较敏感，一些离群点的存在能显著影响均值的大小，而中位数能较好地处理离群点的影响，二者视具体情况使用。
- 离散度量：极差和方差。极差（Range）和方差（Variance）是常用的统计量，用来观察数据分布的宽度和分散情况。极差是样本中最大值与最小值的差值，它可以标识数据的最大散布，但若大部分数值集中在较窄的范围内，极差反而会引起误解，此时需要结合方差来认识数据。方差是每个样本值与全体样本值的平均数之差的平方值的平均数。

② 多变量相关性分析。多变量相关性分析的常用统计量有协方差、相关系数。协方差越接近 0 表明两个变量越不具有（线性）关系，但协方差越大并不表明越相关，因为协方差的定义中没有考虑变量量纲的影响。相关系数考虑了变量量纲的影响，因此是一个更合适的统计量。相关系数的取值在 [-1,1] 上，相关系数为 -1 表示两个变量完全负相关，相关系数为 1 表示两个

变量完全正相关，0 则表示不相关。相关系数是序数型的，只能比较相关程度大小（绝对值比较），并不能做四则运算。

3）数据预处理

（1）数据处理的目的

在真实数据中，我们拿到的数据可能包含大量的缺失值，可能包含大量的噪声，也可能因为人工录入错误（比如，医生的就医记录）导致有异常点存在，这对我们挖据出有效信息造成了一定的困扰，所以我们需要通过一些方法尽量提高数据的质量。

（2）数据处理的环节

数据处理主要包含数据预处理和特征工程两部分，特别是对于数据预处理工作，多个数据源的数据集成工作，数据压缩、数据维度立方体构建、行列转换（宽表或纵表）等数据规约工作，数据离散化等数据变换工作，大批量数据、相对标准化的数据处理工作等，可以利用大数据技术进行批量分布式处理，即通过 ETL 流程进行数据预处理，而不必放在特征工程环节进行处理。第一，提高了数据质量；第二，减少了数据量；第三，得到了最想要的规范数据。这样尽可能把“最合适”的数据输出到后续特征工程的过程中，可以让整个建模过程更加模块化，提高模型性能和可读性。

（3）数据处理的定位

① 数据处理的定位。一个 ML 项目大致可以总结成如图 1-10 所示的流程，从图 1-10 可以看到数据贯穿整个建模的核心流程，同时可以看到数据处理包含数据预处理和特征工程两部分。

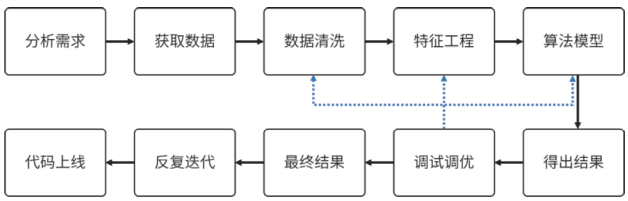

图 1-10　数据建模流程图

数据预处理有 4 个任务：数据清洗、数据集成、数据变换和数据规约。

② 数据处理的重要性。这里引用来自“纽约时报”的一篇报道（Jim Liang 的一个观点），本质上是说数据科学家在他们的时间中有 50%～80%的时间花费在收集和准备不规则数据的更为平凡的任务中，然后才能探索有用的金块。

4）特征工程

（1）特征工程的概述与定位

特征工程是对原始数据进行一系列工程处理，将其提炼为特征，作为输入供算法和模型使用。从本质上来讲，特征工程是一个表示和展现数据的过程。在实际工作中，特征工程旨在去除原始数据中的杂质和冗余，设计更高效的特征以刻画求解的问题与预测模型之间的关系。

（2）特征工程方法论

① 特征理解

- 特征工程的两种常用数据类型分别是结构化数据和非结构化数据。结构化数据可以看作关系

数据库中的一张表，每一列都有清晰的定义，包含数值型和类别型两种基本类型，每一行数据表示一个样本信息。非结构化数据包括文本、图像、音频、视频数据，包含的信息无法用一个简单的数值表示。

- 特征工程的定量和定性数据。定量数据指的是一些数值，用于衡量某件东西的数量。定性数据指的是一些类别，用于描述某件东西的性质。特征工程的定量和定性数据描述如表 1-6 所示。

表1-6 特征工程表

等 级	属 性	案 例	描述性统计	可视化
定 类	离散无序	血型 （A/B/O/AB型） 性别（男女） 货币 （人民币/美元/日元）	频率 占比 众数	条形图 饼图
定 序	有序比较	期末成绩（A/B/C/D） 问卷答案（非常满意/满意/一般/不满意）	频率 众数 中位数 百分位数	条形图 饼图 箱型图
定 距	数据差别	温 度	频率 众数 中位数均值 标准差	条形图 饼图 箱型图 直方图
定 比	连 续	收 入、重 量	均值 标准差	饼图 箱型图 直方图

② 特征构造

目标是增强数据表达，添加先验知识。如果我们对变量进行处理之后，效果仍不是非常理想，就需要进行特征构造了，也就是衍生出新变量。统计量特征扩展，统计量构造新的特征，主要有以下思路：

- 基于业务规则、先验知识等构建新特征。
- 利用 Pandas 的 group by 操作可以创造出以下几种有意义的新特征，如中位数、均值、众数、标准差、最大值、最小值等。仅仅将已有的类别和数值特征进行以上的有效组合，就能够大量增加优秀的可用特征。
- 结合日期与时间型特征构造长、中、短期统计量（如年、月、周、日、时、分、秒），如近 7 日夜间（0:00 ～ 6:00）平均登录次数。

特征分箱包括如下方法。

- 自定义分箱：指根据业务经验或者常识等自行设定划分的区间，然后将原始数据归类到各个区间中。
- 等频分箱：等频分箱如图 1-11 所示，将数据分成几等份，每等份数据里面的个数是一样的。区间的边界值要经过选择，使得每个区间包含大致相等的实例数量。比如说 N=10，每个区间

应该包含大约 10% 的实例。

- 等距分箱：等距分箱如图 1-12 所示，按照相同宽度将数据分成几等份。从最小值到最大值，均分为 N 等份，这样，如果 A、B 分别为最小值和最大值，则每个区间的长度为 W=(B-A)/N，则区间边界值为 A+W,A+2W,…,A+(N-1)W 。这里只考虑边界，每等份里面的实例数量可能不相等。缺点是受到异常值的影响比较大。

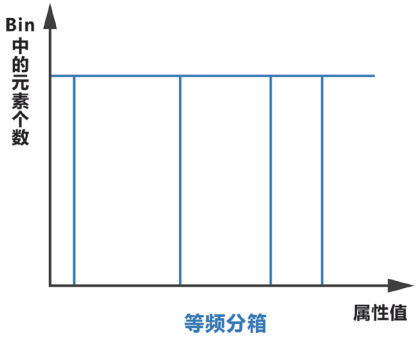

图 1-11　等频分箱图

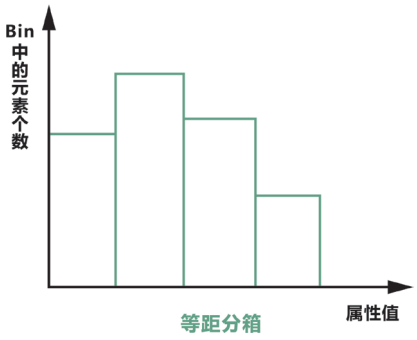

图 1-12　等距分箱图

- Best-KS 分箱：KS（Kolmogorov-Smirnov）用于对模型风险区分能力进行评估，指标衡量的是好坏样本累计部分之间的差距。KS 值越大，表示该变量使得正、负客户的区分程度越大。通常来说，KS>0.2 表示特征有较高的准确率。强调一下，这里的 KS 值是变量的 KS 值，而不是模型的 KS 值。KS 的计算方式：计算每个评分区间的好坏账户数。计算各每个评分区间的累计好账户数占总好账户数的比率（good %）和累计坏账户数占总坏账户数的比率（bad %）。计算每个评分区间累计坏账户比与累计好账户占比差的绝对值（累计 bad% - 累计 good%），然后对这些绝对值取最大值即得到 KS 值。
- 卡方分桶：自底向上的（基于合并的）数据离散化方法。它依赖于卡方检验：具有最小卡方值的相邻区间合并在一起，直到满足确定的停止准则。
- 时间切片特征拓展：在 SQL 中比较容易处理类似“近 n 个月的金额之和 / 最大值 / 最小值 / 平均值”这样的变量，使用 sum(case when date then amount else 0 end) 即可，如果出差在外，只能处理离线数据，不能使用数据库，这个时候就要用 Python 来构造时间切片类的特征。

③ 特征变换

连续值无量纲化，具体包括如下步骤：

- 标准化。转换为 Z-score，使数值特征列的算术平均为 0，方差（以及标准差）为 1。不免疫 outlier（离群值）。

$$x' = \frac{x - \mu}{\sigma}$$

- 归一化。将一列的数值除以这一列的最大绝对值。MinMaxScaler 方法线性映射到 [0，1]，不能免疫异常值。MaxAbsScaler 方法线性映射到 [−1，1]，不能免疫异常值。
- 分布变换：利用统计或数学变换来减轻数据分布倾斜的影响。使原本密集区间的值尽可能分

散，原本分散区间的值尽量聚合。这些变换函数都属于幂变换函数族，通常用来创建单调的数据变换。它们的主要作用在于能够帮助稳定方差，始终保持分布接近正态分布并使得数据与分布的平均值无关。log 变换通常用来创建单调的数据变换。它的主要作用在于帮助稳定方差，始终保持分布接近正态分布并使得数据与分布的平均值无关。因为 log 变换倾向于拉伸那些落在较低的幅度范围内自变量值的范围，倾向于压缩或减少更高幅度范围内的自变量值的范围，从而使得倾斜分布尽可能接近正态分布。所以针对一些数值连续特征的方差不稳定，特征值重尾分布我们需要采用 log 化来调整整个数据分布的方差，属于方差稳定型数据转换。box-cox 变换是另一个流行的幂变换函数簇中的一个函数。该函数有一个前提条件，即数值型值必须先变换为正数（与 log 变换所要求的一样）。一旦出现数值是负的，使用一个常数对数值进行偏移是有帮助的，box-cox 变换可以明显地改善数据的正态性、对称性和方差相等性，对许多实际数据都是行之有效的。

- 离散变量处理：标签编码（Label Encoder）是对不连续的数字或者文本进行编号，编码值是介于 0 和 n_classes-1 之间的标签。独热编码（One Hot Encoder）用于将表示分类的数据扩维。最简单的理解为用 N 位状态寄存器编码 N 个状态，每个状态都有独立的寄存器位，且这些寄存器位中只有一位有效，只能有一个状态。标签二值化（Label Binarizer）功能与独热编码一样，但是独热编码只能对数值型变量二值化，无法直接对字符串型的类别变量编码，而标签二值化可以直接对字符型变量二值化。

④ 特征选择

特征选择的目标是降低噪声，平滑预测能力和计算复杂度，增强模型预测性能。当数据预处理完成后，我们需要选择有意义的特征输入机器学习的算法和模型进行训练。通常来说，从以下两个方面考虑来选择特征。

- 特征是否发散：如果一个特征不发散，例如方差接近 0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。
- 特征与目标的相关性：这点比较显而易见，与目标相关性高的特征，应当优选选择。除方差法外，这里介绍的其他方法均从相关性考虑。

根据特征选择的形式又可以将特征选择方法分为以下 3 种。

- 过滤法（Filter）：按照发散性或者相关性对各个特征进行评分，设定阈值或者待选择阈值的个数来选择特征。
- 包装法（Wrapper）：根据目标函数（通常是预测效果评分），每次选择若干特征或者排除若干特征。
- 嵌入法（Embedded）：先使用某些机器学习的算法和模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征。类似于 Filter 方法，但是通过训练来确定特征的优劣的。

5）模型训练和优化

（1）模型开发迭代

模型的开发和迭代是一个不断试验和尝试的过程。从数据清洗到特征工程再到模型训练是

一个迭代的过程，其中每一步可以视为一种模型设置，都可以尝试。下面列举一些面临的选择：

- 使用多少历史数据进行训练，是否放弃较早的数据？
- 特征工程该怎么做，模型中使用哪些特征？
- 使用哪个算法，超参数如何选择？

对于上述问题，我们要认识到目前没有一套完整的理论支撑应该如何决定每一步的最佳方法，模型开发和科学研究一样，是一个试验和迭代的过程。因此，我们应该系统化地设计和进行试验，验证不同假设，找到最佳方法。

（2）超参数优化

在机器学习模型中，需要人工选择的参数称为超参数。比如随机森林中决策树的个数、人工神经网络模型中隐藏层的层数和每层的节点个数、正则项中的常数大小等。超参数选择不恰当，就会出现欠拟合或者过拟合的问题。而在选择超参数的时候，有两个途径，一个是凭经验微调，另一个是选择不同大小的参数，代入模型中，挑选表现最好的参数组合。挑选超参数的一种方法是手动调整，但是这么做非常耗费开发时间。项目中一般使用自动化的搜索方法优化超参数。常见的优化方法包括网格搜索（Grid Search）和随机搜索（Random Search）。

① 网格搜索。网格搜索即穷举搜索，在所有候选的参数选择中，通过循环遍历尝试每一种可能性，表现最好的参数就是最终的结果。其原理就像是在数组中找到最大值。这种方法的主要缺点是比较耗时。

② 随机搜索。我们在搜索超参数的时候，如果超参数个数较少（三四个或者更少），那么可以采用网格搜索，一种穷尽式的搜索方法。但是当超参数个数比较多的时候，我们仍然采用网格搜索，那么搜索所需的时间将会呈指数级上升。所以有人就提出了随机搜索的方法，随机在超参数空间中搜索几十、几百个点，其中就有可能有比较小的值。这种做法比上面稀疏化网格的做法快，而且实验证明，随机搜索法的结果比稀疏网格法稍好。

（3）模型集成

训练模型时，如果模型表现遇到瓶颈，主要提升方法是增加数据源和特征工程。如果计算资源允许，另一个常用方法是模型集成（Model Ensemble）。模型集成的基本思想是将多个准确率较高但相关性低的基础模型通过某种方式结合成一个集成模型。通俗来讲，每个基础模型可以理解为一个专家的意见，而多个专家通过投票可以做出更加准确的预测。在实际项目中，模型集成一般能够提升效果，是机器学习中最接近“免费的午餐”的方法。当然，模型集成是有代价的，训练多个模型就需要多倍的迭代和训练时间，同时集成模型不易于解释。模型集成具体包括如下方法：

① 投票法（Voting）。投票法针对分类任务。最简单的多数投票法是让所有基础模型的预测结果以少数服从多数的原则决定最终预测结果。除多数投票法外，也可以对基础模型的预测结果进行加权，常用方法是按照基础模型损失加权，即预测结果乘以 1/loss。

② 平均法（Averaging）。平均法针对回归任务，按字面意思就是对所有基础模型的预测结果进行平均。同投票方法，也可以使用加权平均方法，按照基础模型的损失加权。

平均法有一个问题，就是不同的基础模型结果的度量可能不一致，造成波动较小模型的权重较低。为了解决这个问题，人们提出了排序平均（Rank Averaging），也就是先将不同模型的结果归一化之后再进行平均。

③ 堆叠法（Stacking）。堆叠法就是将基础模型（Level 1 模型）的结果输入另一个非线性模型（Level 2 模型），如 GBDT，让模型自行学习最佳的结果合并方法。

6）模型部署

（1）模型结果输出

① 输出为文件

- CSV 文件：使用 pandas.DataFrame.to_csv。
- JSON 文件：使用 pandas.DataFrame.to_json。
- 带有格式的 XLSX 文件：使用 XlsxWriter，该库可以通过 Python 生成带有格式、公式、图表、超链接的 XLSX 文件。
- 序列化（Serialization）：使用 pickle 或 joblib，序列化方法可以保存 Python 中各种类型的对象，例如训练好的模型。

② 输出到数据库

- 关系数据库：使用数据库特有的连接程序库，使用 JDBC 或 ODBC 驱动。
- HDFS 和 Hive：连接 Hive 可以继续使用 JDBC 驱动，但无法批量写入数据，建议自定义函数，先将文件写入本地，再用 Hadoop 命令将文件移动到集群中，最后用 hive 命令加载数据。

（2）模型调用

① 一次性调用

- 适用场景：模型结果一次性输出。

② 定时脚本调用

- 适用场景：模型为固定时间周期调用，例如每日 6 点预测前一日的新数据。
- 使用流程：在工作流中创建 Shell 脚本执行模型主程序、创建工作流、工作流发布和定时上线。

③ API 调用

- 适用场景：模型根据实时 HTTP 请求调用，例如通过页面单击触发模型。
- 使用流程：基于 FastAPI 或 Flask 建立 API。

（3）模型日志

模型在运行时需要利用日志（Log）监控，以便监控模型可能出现的问题。常用的方法是结

合 print 函数和 shell 信息捕捉，或者使用 Python 中的 logging 库。

① print 函数：最简单和直接的日志方法是使用 Python 中的 print 函数打印各类日志信息。例如，一般机器学习任务会拆解为特征工程和模型训练两个阶段。每个阶段可以打印开始和结束信息。当模型部署和自动调度后，由于无法人工观察模型运行情况，因此需要将 Python 程序的输出信息（stdout 和 stderr）进行重新指向，并保存在日志文件中。

② logging 库：使用 print 函数捕捉日志很简单，但当面临较为复杂的程序时有一些限制。首先，无法区分日志信息类型，例如将日志信息分为 MESSAGE 和 BUG。其次，如果想增加时间戳，需要额外定义 wrapper function。如果项目有需求，使用 Python 中的 logging 库可以解决上述问题。

7）模型可视化

（1）代码级可视化

① 为什么需要模型可视化

- 将以下三部分黑箱的内容对客户进行透明化，让整个过程易于客户理解。
- 把现实生活中的问题抽象成数学模型，并且很清楚模型中不同参数的作用（场景和模型的链接）。
- 利用数学方法对这个数学模型进行求解，从而解决现实生活中的问题（建模过程），包括对输入的数据进行数据概览，对经过特征处理的数据进行可视化，对训练之前的参数矩阵进行可视化，训练过程还会有数据可视化的工程，对训练之后的数据进行可视化。
- 评估这个数学模型，是否真正解决了现实生活中的问题，解决的如何（效果和可解释性）。

② 工具介绍

接下来介绍几个 Python 中的可视化数据库。

- Matplotlib：是一个非常基础的 Python 可视化库，其中文学习资料比较丰富，其中最好的学习资料是其官方网站的帮助文档。
- Seaborn：Seaborn 库旨在以数据可视化为中心来挖掘与理解数据，它提供的面向数据集的制图函数主要是对行列索引和数组的操作，包含对整个数据集进行内部的语义映射与统计整合，以此生成信息丰富的图表。
- Pyecharts：是由我国开发人员开发的。与 Matplotlib、Seaborn 等可视化相比，Pyecharts 更符合国内用户的使用习惯。Pyecharts 的目的是实现 Echarts 与 Python 的对接，以便在 Python 中使用 Echarts 生成图表。
- Missingno：通过使用视觉摘要来快速评估数据集的完整性，而不是通过大篇幅的表格。它可以根据热力图或树状图的完成度或点的相关度对数据进行过滤和排序。
- Bokeh：基于 JavaScript 实现交互式可视化，它是原生 Python 语法，可以在 Web 浏览器中实现美观的视觉效果。它的优势在于能够创建交互式的网站图，可以很容易地将数据输出为 JSON 对象、HTML 文档或交互式 Web 应用程序。Bokeh 还支持流媒体和实时数据。

- HoloViews：是一个开源的 Python 库，旨在使数据分析和可视化更加简便，可以用非常少的代码完成数据分析和可视化。
- ggplot：是基于 R 语言的 ggplot2 包和 Python 的绘图系统。ggplot 的运行方式与 Matplotlib 不同，它允许用户对组件进行分层以创建完整的绘图。例如，用户可以从轴开始画，然后添加点，接着添加线、趋势线等。虽然图形语法被认为是绘图的“直观”方法，但经验丰富的 Matplotlib 用户可能需要时间来适应这个新的方式。

8）模型结果可解释性分析

（1）什么是模型的可解释性

模型在训练的过程中会学习到偏差（Bias），导致模型的泛化性能下降。如果模型具有较强的可解释性，就可以协助调试偏差，帮助模型调优。以借贷问题为例，我们的业务目标是向好客户提供贷款，但仅仅使用机器学习方法只能最小化逾期率，同时也要考虑到需要消除对特定客群的偏见，所以我们不仅需要最小化模型的损失函数，还需要在合规和低风险的情况下尽可能地促进业务增长。显然，仅仅使用模型的损失函数来评估是不够的。

（2）为什么进行模型解释

① 模型改进。通过可解释分析可以指导特征工程。一般我们会根据一些专业知识和经验来构建特征，同构分析特征的重要性，可以挖掘更多有用的特征，尤其是在交互特征方面。当原始特征众多时，可解释性分析变得尤为重要。

② 模型的可信度和透明度。在我们构建模型时，需要权衡两个方面，是仅仅想要知道预测的结果是什么，还是想了解模型为什么给出这样的预测。在一些低风险的情况下，我们不一定需要知道决策是如何做出的，比如推荐系统（广告、视频或者商品推荐等）。但是在其他领域，比如在金融和医疗领域，模型的预测结果将会对相关人员产生巨大的影响，因此有时候我们依然需要专家对结果进行解释。从长远来看，更好地理解机器学习模型可以节省大量时间，防止收入损失。如果一个模型没有做出合理的决策，我们可以在应用这个模型并造成不良影响之前就发现这一问题。

③ 识别和防止偏差。方差和偏差是机器学习中广泛讨论的话题。有偏差的模型经常由有偏见的事实导致，如果数据包含微妙的偏差，模型就会学习下来并认为拟合很好。一个有名的例子是，用机器学习模型来为囚犯建议定罪量刑，这显然反映了司法体系在种族不平等上的内在偏差。所以作为数据科学家和决策制定者来说，理解我们训练和发布的模型如何做出决策，可以帮助我们事先预防偏差的增大并及时消除它们。

（3）可解释性的范围

① 算法透明度（Algorithm Transparency）。算法透明度指的是如何从数据中学习一个模型，更加强调模型的构建方式以及影响决策的技术细节。比如使用卷积神经网络（Convolutional Neural Network，CNN）对图片进行分类时，模型做出预测，是因为算法学习到了边或者其他纹理。算法透明度需要弄懂算法知识，而不是数据以及学到的模型。对于简单的算法，比如线性模型，

具有非常高的算法透明度。复杂的模型如深度学习，人们对模型内部了解较少，透明度较差，这将是一个非常重要的研究方向。

② 全局可解释（Global Interpretability）。为了解释模型的全局输出，需要训练模型了解算法和数据。这个层级的可解释性指的是，模型如何基于整个特征空间、模型结构和参数等因素做出决策，哪些特征是重要的，以及特征之间的交互会产生什么影响。模型的全局可解释性可以帮助我们理解不同特征对目标变量分布的影响。在实际情况下，当模型具有大量参数时，人们很难想象特征之间是如何相互作用的，以得到这样的预测结果。然而，某些模型在这个层级是可以解释的。例如，我们可以解释线性模型的权重，以及树模型是如何划分分支和得到节点预测值的。

③ 局部可解释（Local Interpretability）。局部可解释性更加关注单个样本或一组样本。这种情况下，我们可以将模型视为一个黑盒，不再考虑模型的复杂情况。对于单个样本，我们可以观察到模型给出的预测值和某些特征之间可能存在的线性关系，甚至可能是单调关系。因此，局部可解释性比全局可解释可能更加准确。

4. 常见结构化数据分析场景

1）客户管理模型

（1）客户特征细分。一般客户的需求主要是由其社会和经济背景决定的，因此对客户的特征细分，就是对其社会和经济背景所关联的要素进行细分。这些要素包括地理（如居住地、行政区、区域规模等）、社会（如年龄范围、性别、经济收入、工作行业、职位、受教育程度、宗教信仰、家庭成员数量等）、心理（如个性、生活形态等）、消费行为（如置业情况、购买动机类型、品牌忠诚度、对产品的态度等）、客户线上行为（逛、买、比、晒、享）等要素。

（2）客户价值细分。不同客户给企业带来的价值并不相同，有的客户可以连续不断地为企业创造价值和利益，因此企业需要为不同客户规定不同的价值。在经过基本特征的细分之后，需要对客户进行高价值到低价值的区间分隔（例如大客户、重要客户、普通客户、小客户等），以便根据 20% 的客户为项目带来 80% 的利润的原理重点锁定高价值客户。客户价值区间的变量包括客户响应力、客户销售收入、客户利润贡献、忠诚度、推荐成交量等。

（3）客户共性需求细分。依托客户细分的聚类技术，提炼客户的共同需求，以客户需求为导向精确定义企业的业务流程，为每个细分的客户市场提供差异化的营销组合。可以根据不同的数据情况和需要选择不同的聚类算法来进行客户细分。

（4）客户潜客营销模型。企业的发展至关重要，而客户的多少往往在企业发展中占有关键作用。老客户是企业收入来源的稳定部分，但是企业要谋求发展必须靠新的血液，也就是潜在客户。因此，潜在客户的寻找和营销尤为重要。所谓潜在客户，是指对企业的产品或服务存在需求和具备消费能力的待开发客户，这类客户往往很难发现或者不知道如何发掘。而在当今大数据环境下，依靠大量的客户数据及合适的数据挖掘手段，就能构建出一些有效的潜客挖掘模型，利用这些模型能精确地找出潜在客户，从而为营销做好充分准备。

（5）客户生命周期细分。过去几年，各行业获客成本都在直线攀升，部分行业已达数千元

/ 新客。以电商行业为例，2016 年至今，获客成本增长 5 倍。这意味着随着消费升级，人口红利见顶，企业将迎来一场“存量博弈”。但是现在大多企业注重单次交易的购买价值，常以优惠补贴等促销形式，对单次获客投入高成本，以此促进成交来提高业绩，而这种利益驱动下的成交只会使得企业成本居高不下。如果企业能最大限度地利用和挖掘顾客的价值，从而延长客户生命周期，提升客户生命周期价值，有效摆脱对获客与促销高成本投入但低转化的依赖，这时企业将获得更高的利润。

（6）客户细分组合与应用。在对客户群进行细分之后，会得到多个细分的客户群体，但是，并不是得到的每个细分都是有效的。细分的结果应该通过下面几条规则来测试：与业务目标相关的程度；可理解性和是否容易特征化；基数是否足够大，以便保证一个特别的宣传活动；是否容易开发独特的宣传活动等。

2）销量预测与库存预警

梳理影响销售预测的内外部因素，构建销售预测指标体系，采集关键节点数据，建立销售预测模型体系，对总部、分公司、经营部、客户的总量、分尺寸、分型号的销售进行逐级预测。尽可能多因素地考虑，选择最优的预测模型，提高销售预测准确率，进而提高商业库存周转率。建立动态安全库存模型，满足总部、分公司、客户的分型号安全库存建议与预警。对总部、分公司、客户（有商业库存数据的客户），分型号提供合理的补货建议。

1.2.4 语音分析

语音数据作为多媒体数据的一种表现形式，在日常交流中发挥着重要的作用。对语音数据的处理又叫作语音分析，主要包括声纹识别和语音识别，分别说明如下。

1. 声纹识别

近年来，许多智能语音技术服务商开始布局声纹识别领域，声纹识别逐渐进入大众视野。随着技术的发展和在产业内的不断渗透，声纹识别的市场占比也逐年上升，但目前声纹识别需要解决的关键问题还有很多。接下来梳理声纹识别技术的发展历史，并分析每一阶段的关键技术原理，以及遇到的困难与挑战，希望能够让大家对声纹识别技术有进一步的了解。

声纹（Voiceprint）是用电声学仪器显示的携带言语信息的声波频谱。人类语言的产生是人体语言中枢与发音器官之间一个复杂的生理物理过程，不同的人在讲话时使用的发声器官（舌、牙齿、喉头、肺、鼻腔）在尺寸和形态方面有着很大的差异，所以任何两个人的声纹图谱都是不同的。每个人的语音声学特征既有相对稳定性，又有变异性，不是绝对的、一成不变的。这种变异可来自生理、病理、心理、模拟、伪装，也与环境干扰有关。尽管如此，由于每个人的发音器官都不尽相同，因此在一般情况下，人们仍能区别不同的人的声音或判断是不是同一人的声音。因此，声纹也就成为一种鉴别说话人身份的识别手段。

声纹识别是生物识别技术的一种，也叫作说话人识别，是一项根据语音波形中反映说话人生理和行为特征的语音参数自动识别语音说话者身份的技术。首先需要对发音人进行注册，即

输入发音人的一段说话音频，系统提取特征后存入模型库中，然后输入待识别音频，系统提取特征后经过比对打分，从而判断所输入音频中说话人的身份。从功能上来讲，声纹识别技术应有两类，分别为 1:N 和 1:1。前者用于判断某段音频是若干人中的哪一个人所说的，后者则用于确认某段音频是否为某个人所说的。因此，不同的功能适用于不同的应用领域，比如公安领域中重点人员布控、侦查破案、反电信欺诈、治安防控、司法鉴定等经常用到的是 1:N 功能，即辨认音频是若干人中的哪一个人所说的，而 1:1 功能则更多应用于金融领域的交易确认、账户登录、身份核验等。

从技术发展角度来说，声纹识别技术经历了三个大阶段：

- 第一阶段，基于模板匹配的声纹识别技术。
- 第二阶段，基于统计机器学习的声纹识别技术。
- 第三阶段，基于深度学习框架的声纹识别技术。

1）模板匹配的声纹识别

最早的声纹识别技术框架是一种非参数模型，特点在于基于信号比对差别，通常要求注册和待识别的说话内容相同，属于文本相关，因此局限性很强。

此方法将训练特征参数和测试的特征参数进行比较，两者之间的失真（Distortion）作为相似度。例如矢量量化（Vector Quantization，VQ）模型和动态时间规整法（Dynamic Time Warping，DTW）模型。DTW 通过将输入待识别的特征矢量序列与训练时提取的特征矢量进行比较，通过最优路径匹配的方法来进行识别。而 VQ 方法则是通过聚类、量化的方法生成码本，识别时对测试数据进行量化编码，以失真度的大小作为判决的标准。

2）基于统计机器学习的技术框架

由于第一阶段只能用于文本相关的识别，即注册语音的内容需要跟识别语音的内容一致，因此具有很强的局限性，同时受益于统计机器学习的快速发展，声纹识别技术也迎来了第二阶段。此阶段可细分为 4 个小阶段，即 GMM → GMM-UBM/GMM-SVM → JFA → GMM-iVector-PLDA。

（1）高斯混合模型

高斯混合模型（Gaussian Mixture Model，GMM）的特点在于采用大量数据为每个说话人训练（注册）模型。注册要求很长的有效说话人语音。高斯混合模型是统计学中一个极为重要的模型，其在机器学习、计算机视觉和语音识别等领域均有广泛的应用，甚至可以算是神经网络和深度学习普及之前的主流模型。GMM 之所以强大，在于其能够通过对多个简单的正态分布进行加权平均，从而用较少的参数模拟出十分复杂的概率分布。

在声纹识别领域，高斯混合模型的核心设定是：将每个说话人的音频特征用一个高斯混合模型来表示。采用高斯混合模型的动机也可以直观地理解为：每个说话人的声纹特征可以分解为一系列简单的子概率分布，例如发出的某个音节的概率、该音节的频率分布等。这些简单的概率分布可以近似地认为是正态分布（高斯分布）。但是由于 GMM 规模越庞大，表征力越强，

其负面效应也会越明显：参数规模会等比例膨胀，需要更多的数据驱动 GMM 的参数训练才能得到一个更加通用（或泛化）的 GMM 模型。

假设对维度为 50 的声学特征进行建模，GMM 包含 1024 个高斯分量，并简化多维高斯的协方差为对角矩阵，则一个 GMM 待估参数总量为 1024（高斯分量的总权数）+1024×50（高斯分量的总均值数）+1024×50（高斯分量的总方差数）=103424，超过 10 万个参数需要估计。

这种规模的变量即使是将目标用户的训练数据量增大到几个小时，都远远无法满足 GMM 的充分训练要求，而数据量的稀缺又容易让 GMM 陷入一个过拟合（Over-fitting）的陷阱中，导致泛化能力急剧衰退。因此，尽管一开始 GMM 在小规模的文本无关数据集合上表现出了超越传统技术框架的性能，但它却远远无法满足实际场景的需求。

（2）高斯混合背景模型（GMM-UBM）和支持向量机（GMM-SVM）

GMM-VBM 模型的特点在于使用适应模型的方法减少建模注册所需要的有效语音数据量，但对跨信道分辨能力不强。由于前面使用 GMM 模型对数据的需求量很大，因此 2000 年前后，DA Reynolds 的团队提出了一种改进的方案：既然没法从目标用户那里收集到足够的语音，那就换一种思路，可以从其他地方收集到大量非目标用户的声音，积少成多，我们将这些非目标用户数据（声纹识别领域称为背景数据）混合起来充分训练出一个 GMM，这个 GMM 可以看作对语音的表征，但由于它是从大量身份的混杂数据中训练而成的，因此不具备表征具体身份的能力。

该方法对语音特征在空间分布的概率模型给出了一个良好的预先估计，我们不必再像过去那样从头开始计算 GMM 的参数（GMM 的参数估计是一种称为 EM 的迭代式估计算法），只需要基于目标用户的数据在这个混合 GMM 上进行参数的微调即可实现目标用户参数的估计，这个混合 GMM 就叫通用背景模型（Universal Background Model，UBM）。

UBM 的一个重要优势在于它是通过最大后验估计（Maximum A Posterior，MAP）的算法对模型参数进行估计，避免了过拟合的发生。MAP 算法的另一个优势是我们不必再去调整目标用户 GMM 的所有参数（权重、均值、方差），只需要对各个高斯成分的均值参数进行估计，就能实现最好的识别性能。这样待估计的参数一下减少了一半多（103424 → 51200），越少的参数也意味着更快地收敛，不需要那么多的目标用户数据即可完成对模型的良好训练。

GMM-UBM 系统框架是 GMM 模型的推广，用于解决当前目标说话人数据量不够的问题。通过收集其他说话人的数据进行预先训练，并利用 MAP 算法的自适应将预先训练过的模型向目标说话人模型进行微调。这种方法可以显著减少训练所需要的样本量和训练时间（通过减少训练参数）。

但是 GMM-UBM 缺乏对应信道多变性的补偿能力，因此后来 WM Campbell 将支持向量机（Support Vector Machine，SVM）引入到 GMM-UBM 的建模中，通过将 GMM 每个高斯分量的均值单独拎出来，构建一个高斯超向量（Gaussian Super Vector，GSV）作为 SVM 的样本，利用 SVM 核函数的强大非线性分类能力，在原始 GMM-UBM 的基础上大幅提升了识别的性能，同时基于 GSV 的一些规整算法，例如扰动属性投影（Nuisance Attribute Projection，NAP）、类

内方差规整（Within Class Covariance Normalization，WCCN）等，在一定程度上补偿了由于信道易变形对声纹建模带来的影响。

（3）联合因子分析法

联合因子分析法（Joint Factor Analysis，JFA）的特点在于分别建模说话人空间、信道空间以及残差噪声，但每一步都会引入误差。在传统的基于 GMM-UBM 的识别系统中，由于训练环境和测试环境的失配问题，导致系统性能不稳定。于是 Patrick Kenny 在 2005 年左右提出了一个设想：既然声纹信息可以用一个低秩的超向量子空间来表示，那么噪声和其他信道效应是不是也能用一个不相关的超向量子空间进行表达呢？基于这个假设，Kenny 提出了联合因子分析的理论分析框架，将说话人所处的空间和信道所处的空间做了独立不相关的假设，在 JFA 的假设下，与声纹相关的信息全部可以由特征音空间（Eigenvoice）进行表达，并且同一个说话人的多段语音在这个特征音空间上都能得到相同的参数映射，之所以实际的 GMM 模型参数有差异，这个差异信息是由说话人差异和信道差异这两个不可观测的部分组成的，公式如下：

$$M=s+c$$

其中，s 为说话人相关的超矢量，表示说话人之间的差异；c 为信道相关的超矢量，表示同一个说话人不同语音段的差异；M 为 GMM 均值超矢量，表述为说话人相关部分 s 和信道相关部分 c 的叠加，如图 1-13 所示。

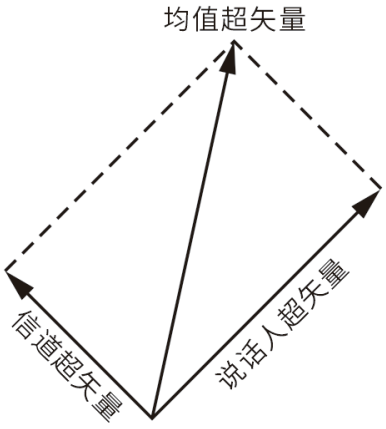

图 1-13　均值超矢量

如图 1-13 所示，联合因子分析实际上是用 GMM 超矢量空间的子空间对说话人差异及信道差异进行建模，从而可以去除信道的干扰，得到对说话人身份更精确的描述。JFA 定义的公式如下：

$$s=m+Vy+Dz$$

$$C=Ux$$

其中，s 为说话人相关的超矢量，表示说话人之间的差异；m 为与说话人以及信道无关的均值超矢量；V 为低秩的本征音矩阵；y 为说话人相关因子；D 为对角的残差矩阵；z 为残差因子；c 为信道相关的超矢量，表示同一个说话人不同语音段的差异；U 为本征信道矩阵；x 为与特定说话人的某一段语音相关的因子。这里的超参数集合 {V,D,U} 即为需要评估的模型参数。有了上面的定义公式，我们可以将均值超矢量重新改写为如下形式：

$$M=m+Vy+Dx+Dz$$

为了得到 JFA 模型的超参数，我们可以使用 EM 算法训练出 UBM 模型，使用 UBM 模型提取 Baum-Welch 统计量。尽管 JFA 对于特征音空间与特征信道空间的独立假设看似合理，但绝对的独立同分布的假设是一个过于强的假设，这种独立同分布的假设往往为数学的推导提供了便利，却限制了模型的泛化能力。

（4）基于 GMM 的 I-Vector 方法及 PLDA

I-Vector 方法的特点在于统一建模所有空间，进一步减少注册和识别所需的语音时长，使用 PLDA 分辨说话人的特征，但噪声对 GMM 仍然有很大影响。N.Dehak 提出了一个更加宽松的假设：既然声纹信息与信道信息不能做到完全独立，那就用一个超向量子空间对两种信息同时建模，即用一个子空间同时描述说话人信息和信道信息。这时候，同一个说话人，无论怎么采集语音，采集了多少段语音，在这个子空间上的映射坐标都会有差异，这也更符合实际情况。这个既模拟说话人差异性又模拟信道差异性的空间称为全因子空间（Total Factor Matrix），每段语音在这个空间上的映射坐标称作身份认证（Identity-Vector，I-Vector）向量，I-Vector 向量的维度通常也不会太高，一般在 400~600。

I-Vector 方法采用一个空间来代替这两个空间，这个新的空间可以成为全局差异空间，它既包含说话人之间的差异，又包含信道间的差异。所以 I-Vector 的建模过程在 GMM 均值超矢量中不严格区分说话人的影响和信道的影响。这一建模方法的动机来源于 Dehak 的又一研究：JFA 建模后的信道因子不仅包含信道效应，也夹杂着说话人的信息。I-Vector 中 Total Variability 的做法（M=m+Tw）将 JFA 复杂的训练过程以及对语料的复杂要求瞬间降到了极致，尤其是将 Length-Variable Speech 映射到了一个固定且低维（Fixed and Low-Dimension）的身份认证向量上。于是，所有机器学习算法都可以用来解决声纹识别的问题了。现在，主要用的特征是 I-Vector。这是通过高斯超向量基于因子分析而得到的，是基于单一空间的跨信道算法，该空间既包含说话人空间的信息，也包含信道空间信息，相当于用因子分析方法将语音从高维空间投影到低维。可以把 I-Vector 看作一种特征，也可以看作简单的模型。最后，在测试阶段，我们只要计算测试语音 I-Vector 和模型的 I-Vector 之间的余弦（cosine）距离，就可以作为最后的得分。这种方法也通常被作为基于 I-Vector 说话人识别系统的基线系统。

I-Vector 简洁的背后是它舍弃了太多的东西，其中就包括文本差异性，在文本无关识别中，由于注册和训练的语音在内容上的差异性比较大，因此我们需要抑制这种差异性。但在文本相关识别中，又需要放大训练和识别语音在内容上的相似性，这时候牵一发而动全身的 I-Vector

就显得不是那么合适了。虽然 I-Vector 在文本无关声纹识别上表现非常好，但是在看似更简单的文本相关声纹识别任务上，I-Vector 表现得并不比传统的 GMM-UBM 框架更好。I-Vector 的出现使得说话人识别的研究一下子简化抽象为一个数值分析与数据分析的问题：任意的一段音频，无论长度怎样，内容如何，最后都会被映射为一段低维度的定长 I-Vector。只需要找到一些优化手段与测量方法，在海量数据中能够将同一个说话人的几段 I-Vector 尽可能分类得近一些，将不同说话人的 I-Vector 尽可能分得远一些。并且 Dehak 在实验中还发现，I-Vector 具有良好的空间方向区分性，即便在 SVM 进行区分，也只需要选择一个简单的余弦核就能实现非常好的区分性。I-Vector 在大多数情况下仍然是文本无关声纹识别中表现性能最好的建模框架，学者们后续的改进都是基于对 I-Vector 进行优化，包括线性区分分析（Linear Discriminant Analysis，LDA）、概率线性判别分析（Probabilistic Linear Discriminant Analysis，PLDA）甚至是度量学习（Metric Learning）等。

概率线性判别分析是一种信道补偿算法，被用于对 I-Vector 进行建模、分类，实验证明其效果最好。因为 I-Vector 中既包含说话人的信息，也包含信道信息，而我们只关心说话人信息，所以才需要做信道补偿。我们假设训练数据语音由 i 个说话人的语音组成，其中每个说话人有 j 段自己不同的语音。那么，我们定义第 i 个人的第 j 条语音为 x_{ij}。根据因子分析，我们定义 x_{ij} 的生成模型为：

$$x_{ij}=\mu+Fh_i+Gw_{ij}+\epsilon_{ij}$$

PLDA 模型训练的目标就是输入一堆数据 x_{ij}，输出可以最大限程度地表示该数据集的参数 $\theta=[\mu,F,G,\Sigma]$。由于我们现在不知道隐藏变量 h_i 和 W_{ij}，因此还是使用 EM 算法来进行求解。

在 PLDA 中，我们计算两条语音是否由说话人空间中的特征 h_i 生成，或者由 h_i 生成似然程度，而不用去管类内空间的差异。下面给出得分公式：

$$\text{score}=\log\frac{p(\eta_1,\eta_2\mid H_s)}{p(\eta_1\mid H_d)p(\eta_2\mid H_d)}$$

以上公式中，η_1 和 η_2 分别是两个语音的 I-Vector 矢量，这两条语音来自同一空间的假设为 H_s，来自不同的空间的假设为 H_d。其中 $p(\eta_1,\eta_2|H_s)$ 为两条语音来自同一空间的似然函数；$p(\eta_1|H_d)$、$p(\eta_2|H_d)$ 分别为 η_1 和 η_2 来自不同空间的似然函数。通过计算对数似然比，就能衡量两条语音的相似程度。比值越高，得分越高，两条语音属于同一说话人的可能性越大；比值越低，得分越低，则两条语音属于同一说话人的可能性越小。

3）基于深度神经网络的技术框架

随着深度神经网络技术的迅速发展，声纹识别技术逐渐采用了基于深度神经网络的技术框架，目前有 DNN-iVector-PLDA 和最新的 End-2-End。

（1）基于深度神经网络的方法（D-Vector）

深度神经网络（Deep Neural Networks，DNN）的特点在于可以从大量样本中学习到高度

抽象的音素特征，同时它具有很强的抗噪能力，可以排除噪声对声纹识别的干扰。在论文 Deep *Neural Networks for Small Footprint Text-Dependent Speaker Verification* 中，作者对 DNN 在声纹识别中的应用做了研究。

DNN 经过训练可以在帧级别对说话人进行分类。在说话人录入阶段，使用训练好的 DNN 用于提取来自最后隐藏层的语音特征。这些说话人特征或平均值即 D-Vector，用作说话人特征模型。在评估阶段，为每个话语提取 D-Vector 与录入的说话人模型相比较进行验证。实验结果表明，基于 DNN 的 D-Vector 与常用的 I-Vector 在一个小的声音文本相关的声纹验证集上相比，具有更良好的性能表现。

深度网络的特征提取层（隐藏层）输出帧级别的说话人特征，将其以合并平均的方式得到句子级别的表示，这种 utterance-level 的表示即深度说话人向量，简称 D-Vector。计算两个 D-Vectors 之间的余弦距离得到判决打分。类似主流的概率统计模型 I-Vector，可以通过引入一些正则化方法（线性判别分析（LDA）、概率线性判别分析（PLDA）等），以提高 D-Vector 的说话人区分性。此外，基于 DNN 的系统在噪声环境中更加稳健，并且在低错误拒绝上优于 I-Vector 系统。最后，D-Vector-SV 系统在进行安静和嘈杂的条件分别以 14％和 25％的相对错误率（Equal Error Rate，EER）优于 I-Vector 系统。

（2）端到端（End-to-End）深度神经网络

端到端（End-to-End）深度神经网络的特点在于由神经网络自动提取高级说话人的特征并进行分类。随着端到端技术的不断发展，声纹识别技术也进行了相应的尝试，百度在论文 *an End-to-End Neural Speaker Embedding System* 中提出了一种端到端的声纹识别系统。

Deep Speaker 是一个系统，包含一个说话人识别的流程，包括语音前端处理 + 特征提取网络（模型）+ 损失函数训练（策略）+ 预训练（算法）。

文中设定了一个 ResBlock，如图 1-14 所示：3×3 的卷积核 + ReLU 激活 +3×3 的卷积核。ResBlock 最后激活函数的输出：a[L+1]=g(z[L+1]+a[L])，残差的核心就体现在这个 a[L] 中。其中，z[L+1] 为输入数据经过块中的（Cov、ReLU、Cov）得到的输出。

layer name	structure	stride	dim	#params
conv64-s	5x5,64	2x2	2048	6K
res64	[3x3,64 / 3x3,64]x3	1x1	2048	41Kx6
conv128-s	5x5,128	2x2	2048	209K
res128	[3x3,128 / 3x3,128]x3	1x1	2048	151Kx6
conv256-s	5x5,256	2x2	2048	823K
res256	[3x3,256 / 3x3,256]x3	1x1	2048	594Kx6
conv512-s	5x5,512	2x2	2048	3.3M
res512	[3x3,512 / 3x3,512]x3	1x1	2048	2.4Mx6
average	-	-	2048	0
affine	2048x512	-	512	1M
1n	-	-	512	0
triplet	-	-	512	0
total				24M

图 1-14 ResCNN

- conv64-s：单纯的卷积层。卷积核尺寸为 5×5（卷积核实际上是 5×5×c，其中 c 为输入数据的通道数）；个数为 64，也就代表着输出数据的第三维；步长为 2×2，会改变数据维度的前 2 维，也就是高和宽。
- res64：是一个 ResBlock（残差块），并不是一层网络，实际层数是这个 ResBlock 中包含的层

数，这里残差块中包含 2 个卷积层：卷积核尺寸为 3×3，个数为 64，步长为 1×1（也就是上文的 Cov+ReLU+Cov，也就是 2 层，中间激活层不算）。后面的乘 3 是指有 3 个 ResBlock。所以说这个 res64 部分是指经过 3 个 ResBlock，而且每一个 ResBlock 中包含 2 个卷积层，其实是 6 层网络。

- average 层，本来数据是三维的，分别代表（时间帧数 × 每帧特征维度 × 通道数），通道数也就是经过不同方式提取的每帧的特征（Fbank 或 MFCC 这种）。将时间平均，这样一段语音就对应一段特征了，而不是每一帧都对应一段特征。
- affine 层：仿射层，就是将维度 2048 的特征变为 512 维。
- ln 层（Length Normalization Layer）：标准化层，特征标准化之后，用得到的向量来表示说话者语音。

关于 dim（维度）这一列，开始时输入的语音数据是三维的：（时间帧数 × 每帧特征维度 × 通道数）。这里时间帧数根据语音长度可变，每帧特征维度为 64，通道数为 3（代表 Fbank、一阶、二阶）。所以输入维度为时间帧数 ×64×3。经过第一层 conv64-s 后：因为卷积层步长 2×2，所以时间帧数和每帧特征维度都减半了，特征维度变为 32，通道数变为卷积核个数 64。32×64=2048，也就是 dim 的值。所以，这里的 dim 指的是除去时间维的频率特征维度。训练的时候，使用 Triplet loss 作为损失函数，如图 1-15 所示。通过随机梯度下降使得来自同一个人的向量相似度尽可能大，不是同一个说话者的向量相似度尽可能小。

该论文在三个不同的数据集上演示了 Deep Speaker 的有效性，其中既包括依赖于文本的任务，也包含独立于文本的任务。其中一个数据集 UIDs 包含大约 250 000 个说话人，这是目前所知文献中最大规模的。实验表明，Deep Speaker 的表现显著优于基于 DNN 的 I-Vector 方法，具体实验数据如图 1-16 所示。

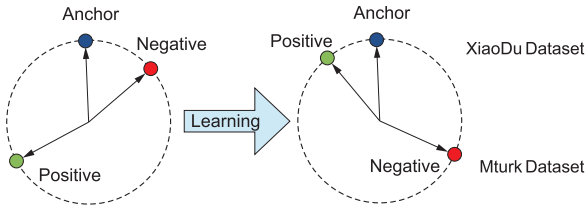

图 1-15　Triplet loss

UIDs Dataset

	#spkr	#utt	#utt/spkr	dur/utt
Train250k	249,312	12,202,181	48.94	3.72s
Train50k	46,835	2,236,379	47.75	3.66s
Eva200	200	3,800	19	4.25s

XiaoDu Dataset

	#spkr	#utt	#utt/spkr	dur/utt
train	11,558	89,227	7.72	1.56s
test	844	10,014	11.86	1.48s

Mturk Dataset

	#spkr	#utt	#utt/spkr	dur/utt
train	2,174	543,840	250.16	4.16s
test	200	4,000	20	4.31s

图 1-16　Deep Speaker 实验结果

比如，在一个独立于文本的数据集上，Deep Speaker 在说话人验证任务上达到了 1.83% 的等错误率（EER），并且在 100 个随机采样的候选者的说话人识别任务上得到了 92.58% 的准确度。相比基于 DNN 的 I-Vector 方法，Deep Speaker 的 EER 下降了 50%，准确度提高了 60%。

总结：从声纹识别技术发展综述中，不难看出，声纹识别的研究趋势正在快速朝着深度学习和端到端方向发展，其中最典型的就是基于句子层面的做法。在网络结构设计、数据增强、

损失函数设计等方面还有很多的工作要做，还有很大的提升空间。

2. 语音识别

在人工智能飞速发展的今天，语音识别技术成为很多设备的标配，过去 5 年间，语音识别的需求逐渐爆发。然而，目前语音识别相关的应用及使用场景仍具有局限性，因此，国内外众多企业纷纷开始探索语音识别的新算法新策略。接下来从技术发展的角度出发，深入分析语音识别技术不同发展阶段的模型构建和优化，以及未来发展趋势。

简单地说，语音识别技术就是将计算机接收到的音频信号转换为相应的文字。语音识别技术从 20 世纪 50 年代出现，发展到现在已有半个多世纪的历史。经过多轮技术迭代，语音识别已经从最早的孤立数字识别发展到今天复杂环境下的连续语音识别，并且已经应用到各种电子产品中，为人们的日常生活带来许多便利。

从技术发展的历史来讲，语音识别技术主要经历了三个时代，即基于模板匹配的技术框架、基于统计机器学习的技术框架和最新的端到端技术框架。近年来，得益于深度学习技术突破性的进展，以及移动互联网的普及带来的海量数据的积累，语音识别已经达到了非常高的准确率，在某些数据集上甚至超过了人类的识别能力。

随着识别准确率的提升，研究者们的关注点也从语音识别的准确率渐渐转移到了一些更加复杂的问题上，比如多语种混合语音识别。该问题涉及多语种混合建模、迁移学习和小样本学习等技术。对某些小语种来说，由于无法获得足够多的训练样本，因此如何从小样本数据中构建可靠的语音识别系统成为一个待解决的难题。

接下来将重点介绍语音识别技术不同发展阶段经历的重要技术框架，包括传统的 GMM-HMM 和 DNN-HMM，以及最新的端到端方法等。

1）GMM-HMM/DNN-HMM

从 GMM-HMM 开始讲起，GMM-HMM 基本使用 HTK 或者 Kaldi 进行开发。在 2010 年之前，整个语音识别领域都是在 GMM-HMM 中做一些文章。

我们的语音通过特征提取后，利用高斯混合模型（Gaussian Mixed Mode，GMM）来对特征进行建模。这里的建模单元是 cd-states。建模单元在 GMM-HMM 时代，或者 DNN-HMM 时代，基本没有太多创新，大多使用 tied triphone，即 senone。

2）DNN-HMM

在 2010 年前后，由于深度学习的发展，整个语音识别的框架开始转变成 DNN-HMM。其实就是把原来用 GMM 对特征进行建模，转换成用神经网络建模。由于神经网络从 2010 年至今不断发展，各种不同的结构不断出现，也带来了不同的效果。DNN 模型可以是纯 DNN 模型、CNN 模型或 LSTM 模型等。整个模型层只是在 GMM 基础上进行替换。在这个时代，模型结构整体上都是各种调优，最经典的模型结果就是谷歌的 CLDNN 模型和 LSTM 结构。*Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition* 是公认的第一篇研究 DNN-HMM 的论文。而后，谷歌、微软等公司在这一算法上不断推进。相对传统的

GMM-HMM 框架，DNN-HMM 在语音识别任务上可以获得全面的提升。DNN-HMM 之所以取得巨大的成功，通常被认为有三个原因：第一，DNN-HMM 舍弃了声学特征的分布假设，模型更加复杂精准；第二，DNN 的输入可以采用连续的拼接帧，因而可以更好地利用上下文的信息；第三，可以更好地利用鉴别性模型的特点。

3）端到端语音识别

端到端语音识别是近年来业界研究的热点，主流的端到端方法包括 CTC、RNN-T 和 LAS。

传统的模型训练还是比较烦琐，而且特别依赖 HMM 这套架构体系。真正脱离 HMM 的是 CTC。CTC 在一开始是由 Hinton 的博士生 Grave 发现的。CTC 框架虽然在学习传统的 HMM，但是抛弃了 HMM 中一些复杂的东西。CTC 从原理上就解释得比 HMM 好，因为强制对齐的问题存在不确定因素或者状态边界有时分不清楚，但 HMM 必须要求分一个出来。

而 CTC 的好处就在于，它引入了 blank 的概念，在边界不确定的时候就用 blank 代替，用尖峰来表示确定性。所以边界不准的地方就可以用 blank 来替代，而我们觉得确信的东西用一个尖峰来表示，这样尖峰经过迭代就越来越强。CTC 在业界的使用有两个办法，有人把它当作声学模型使用，有人把它当作语音识别的全部。但目前工业界系统都只把 CTC 当作声学模型来使用，其效果更好。纯端到端的使用 CTC 进行语音识别效果还是不够好。

这里讲一下 Chain 模型，Chain 模型源自 Kaldi。Kaldi 当时也想做 CTC，但发现在 Kaldi 体系下 CTC 效果不好，但 CTC 的一些思想特别好，后来 Dan Povey 发现可以在此基础上做一些优化调整，于是就把 Chain 模型调好了。但在 Kaldi 体系里，Chain 模型的效果的确比原来模型的效果要更好，这个在 Dan Povey 的论文中有解释。

CTC 时代的改进让语音识别技术朝着非常好的方向发展，CTC 还有一个贡献就是前面提到的建模单元，CTC 把建模单元从原来的 cd-states 调整为 cdphone，或到后面的音节（Syllable），或到后面的字级别（Char）。因此，端到端的语音识别系统中很少使用前面细粒度的建模。目前很多公司的线上系统都是基于 LSTM 的 CTC 系统。

CTC 在业界用得最成功的论文是 *Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition*，论文里探索出来在 CTC 领域比较稳定的模型结构是 5 层 LSTM 的结构。这篇文章从 LSTM 是单向还是双向，建模单元是 cdstate、ciphone 还是最终的 cdphone 等问题进行探究。性能最优的是 cdphone 的双向 LSTM 的 CTC 系统。但是由于双向在线上流式处理不好处理，因此单向 LSTM 的性能也是可以接受的。

整体 CTC 阶段，以 Alex Graves 的论文为主线，论文中从 timit 小数据集，到最终谷歌上万小时的数据集，一步一步验证了 CTC 算法的威力，引领了语音界的潮流。CTC 是语音界一个比较大的里程碑的算法。

接下来介绍注意力（Attention）机制。注意力机制天然适合 Seq2Seq 模型，而语音天然就是序列问题。LAS 的全称为 Listen，Attended and Spell，此模型拉开了纯端到端语音识别架构的序幕。LAS 目前应该是所有网络结构里面最好的模型，性能也是最好的，这点毋庸置疑，超过

了原来基于 LSTM-CTC 的 Baseline。然而，LAS 的主要限制在于它需要接收所有的输入，这对于流式解码来说是不允许的，这一致命的问题影响了这种算法的推进，也引起了众多研究者的关注。当然，最好的办法就是减少 Attention 对输入的依赖，因此推出了一个叫 Mocha 的算法，该算法以后有机会再介绍。

CTC 算法虽然是一个里程碑式的算法，但 CTC 算法也有缺陷，比如要求每一帧是条件独立的假设，比如要想性能好，需要外加语言模型。一开始的 LAS 模型效果也不够好。后来谷歌的研究者们经过各种算法演练，各种尝试，最终提出了流式解码，性能也更好。但是严格来说，谷歌的流式模型也不是 LAS 模型，如果不考虑流式解码，LAS 模型结构肯定是最优的。

RNN-T：和 LAS 模型类似的还有一个 RNN-T 算法，它天然适合流式解码。RNN-T 也是 Grave 提出的，此算法在 2012 年左右就提出来了，但是并没有受到广泛关注，直到谷歌把它运用到 Pixel 手机里才开始流行起来。RNN-T 相比 CTC，继承了 blank 机制，但对原来的路径做了约束。相比 CTC，RNN-T 的约束更合理，所以整体性能也比 CTC 好。但是 RNN-T 较难训练，一般需要把 CTC 模型当作预训练模型的基础进行训练。此外，RNN-T 的显存极易爆炸，因此有很多人在改进显存的应用。谷歌在 2020 ICASSP 的论文中写着用 RNN-T 结合 LAS，效果超过了基于 LSTM-CTC 的 Baseline 方案。

Transformer/Conformer（变换器 / 整形器）：Transformer 和 Conformer 是目前性能最好的模型。Transformer 模型是从 NLP 借鉴到 ASR 领域的，在 ESPnet 的论文里证明，Transformer 模型在各个数据集上效果比 RNN 或者 Kaldi 模型都好。同样，在谷歌的论文 *FastEmit: Low-latency Streaming ASR with Sequence-Level Emission Regularization* 中，同样在 LibriSpeech 上，Conformer 模型比 LSTM 或者 Transformer 模型好。

最后，为什么大家都去研究端到端模型，其实可以从两方面来考虑：第一，端到端模型把原来传统的模型简化到最简单的模型，抛弃了传统的复杂的概念和步骤；第二，其实整个端到端模型用很小的模型结构就可以达到原来几十吉字节模型的效果。谷歌论文的原文里写着：

In this section, we compare the proposed RNN-T+LAS model (0.18G inmodel size) to a state-of-the-art conventional model. This model uses alow-frame-rate (LFR) acoustic model which emits contextdependent phonemes(0.1GB), a 764k-word pronunciation model (2.2GB), a 1st-pass 5-gramlanguage-model (4.9GB), as well as a 2nd-pass larger MaxEnt language model(80GB). Similar to how the E2E model incurs cost with a 2nd-pass LASrescorer, the conventional model also incurs cost with the MaxEnt rescorer. Wefound that for voice-search traffic, the 50% computation latency for the MaxEntrescorer is around 2.3ms and the 90% computation latency is around 28ms. InFigure 2, we compare both the WER and EP90 of the conventional and E2E models.The figure shows that for an EP90 operating point of 550ms or above, the E2Emodel has a better WER and EP latency tradeoff compared to the conventionalmodel. At the operating point of matching 90% total latency (EP90 latency + 90%2nd-pass rescoring computation latency) of E2E and server models, Table 6 showsE2E gives a 8% relative improvement over conventional, while being more than400-times smaller in size.

但端到端模型真正与业务相结合时，遇到的问题还是很明显的，比如不同场景下模型如何调整？遇到一些新词的时候 LM 如何调整？针对此类问题，学术界和工业界都在寻找新的解决方案。

1.2.5 视觉分析

除语音外，图像和视频同样是多媒体数据的重要表现形式。从图像和视频中提取有用的信息对数据科学领域来说非常重要。比如，通过人脸识别或者物体识别，可以将包含相同人脸或者物体的照片进行归类保存。通过内容识别，可以有效地判断视频中包含的场景，从而有效地对视频进行管理。对图像和视频数据的处理又叫作视觉分析。

计算机视觉（Computer Vision，CV）是一门综合性的学科，是极富挑战性的重要研究领域，目前已经吸引了来自各个学科的研究者参加到对它的研究之中。本小节梳理计算机视觉技术的基本原理和发展历程，针对其当前主要的研究方向及落地应用情况进行深入剖析，并分享百分点科技在该领域的技术研究和实践成果。

计算机视觉是人工智能的一个领域，它与语音识别、自然语言处理共同成为人工智能最重要的三个核心领域，也是应用最广泛的三个领域，计算机视觉使计算机和系统能够从数字图像、视频和其他视觉输入中获取有意义的信息，并根据这些信息采取行动或提出建议。如果人工智能使计算机能够思考，那么计算机视觉使它们能够看到、观察和理解。计算机视觉的工作原理与人类视觉大致相同，只是人类具有领先优势。人类视觉具有上下文生命周期的优势，可以训练如何区分对象，判断它们有多远、它们是否在移动，以及图像中是否有问题等情况。计算机视觉训练机器执行这些功能，不是通过视网膜、视神经和视觉皮层，而是用相机、数据和算法，能够在更短的时间内完成。因为经过培训以检查产品或观察生产资产的系统可以在一分钟内分析数千个产品或流程，发现不易察觉的缺陷或问题，所以它可以迅速超越人类的能力。

1. 计算机视觉的工作原理

计算机视觉需要大量数据，它一遍又一遍地运行数据分析，直到辨别出区别并最终识别出图像。例如，要训练计算机识别咖啡杯，需要输入大量咖啡杯图像和类似咖啡杯的图像来学习差异并识别咖啡杯。现在一般使用深度学习中的卷积神经网络（Convolutional Neural Networks，CNN）来完成这一点，也就是说最新的科研方向和应用落地绝大多数都是基于深度学习的计算机视觉的。

CNN 通过将图像分解为具有标签或标签的像素来帮助机器学习或深度学习模型“观察”，使用标签来执行卷积（对两个函数进行数学运算以产生第三个函数）并对其“看到”的内容进行预测。神经网络运行卷积并在一系列迭代中检查其预测的准确性，直到预测开始成真，然后以类似于人类的方式识别或查看图像。就像人类在远处观察图像一样，CNN 首先识别硬边缘和简单形状，然后在运行其预测的迭代时填充信息。

2. 计算机视觉的发展历程

60 多年来，科学家和工程师一直在努力开发让机器查看和理解视觉数据的方法。实验始于 1959 年，当时神经生理学家向一只猫展示了一系列图像，试图将其大脑中的反应联系起来。他

们发现它首先对硬边或线条做出反应，从科学上讲，这意味着图像处理从简单的形状开始，比如直边。

大约在同一时期，第一个计算机图像扫描技术被开发出来，使计算机能够数字化和获取图像。1963 年达到了另一个里程碑，当时计算机能够将二维图像转换为三维形式。在 1960 年代，人工智能作为一个学术研究领域出现，这也标志着人工智能寻求解决人类视觉问题的开始。

1974 年引入了光学字符识别（Optical Character Recognition，OCR）技术，该技术可以识别以任何字体或字样打印的文本。同样，智能字符识别（Intelligent Character Recognition，ICR）可以使用神经网络破译手写文本。此后，OCR 和 ICR 进入文档和发票处理、车牌识别、移动支付、机器翻译等常见应用领域。

1982 年，神经科学家 David Marr 确定视觉是分层工作的，并引入了机器检测边缘、角落、曲线和类似基本形状的算法。与此同时，计算机科学家 Kunihiko Fukushima 开发了一个可以识别模式的细胞网络。该网络称为 Neocognitron，在神经网络中包含卷积层。

到 2000 年，研究的重点是物体识别，到 2001 年，第一个实时人脸识别应用出现。视觉数据集如何标记和注释的标准化出现在 2000 年。2010 年，李飞飞所带领的团队为了提供一个非常全面、准确且标准化的可用于视觉对象识别的数据集创造出了 ImageNet。它包含跨越 1000 个对象类别的数百万个标记图像，并以此数据集为基础每年举办一次软件比赛，即 ImageNet 大规模视觉识别挑战赛（ILSVRC），为当今使用的 CNN 和深度学习模型奠定了基础。2012 年，多伦多大学的一个团队将 CNN 输入图像识别竞赛中。该模型称为 AlexNet，它是由 Yann LeCun 于 1994 年提出的 Lenet-5 衍变而来的，显著降低了图像识别的错误率。第二名 TOP-5 错误率为 26.2%（没有使用卷积神经网络），AlexNet 获得冠军，错误率为 15.3%。在这一突破之后，错误率下降到只有几个百分点（到 2015 年分类任务错误率只有 3.6%）。

3. 计算机视觉的主要研究方向

人类应用计算机视觉解决的最重要的问题是图像分类、目标检测和图像分割，按难度递增，其中图像分割主要包含语义分割、实例分割、全景分割。

在传统的图像分类任务中，我们只对获取图像中存在的所有对象的标签感兴趣。在目标检测中，我们更进一步，尝试在边界框的帮助下了解图像中存在的所有目标以及目标所在的位置。图像分割通过尝试准确找出图像中对象的确切边界并将其提升到一个新的水平，以下用图例简单地介绍它们是如何工作的。

1）图像分类

图像分类（Image Classification）识别图像中存在的内容，即图像所属类别，通常结果为一个带有概率的分类结果，一般取概率最高的类别为图像分类结果。

2）目标检测

目标检测（Object Detection）将物体的分类和定位合二为一，识别图像中存在的内容和检测其位置。

3）图像分割

语义分割（Semantic Segmentation）是将图像中属于某个类别的每个像素进行分类的过程，因此可以将其视为每个像素的分类问题。

实例分割（Instance Segmentation）是目标检测和语义分割的结合，在图像中将目标检测出来，然后对每个像素打上标签，实例分割与语义分割的不同之处在于它只为检测出的目标像素打上标签，不需要将全部像素打上标签，并且语义分割不区分属于同类别的不同实例，实例分割需要区分同类别的不同实例，为每个目标打上 id 标签（使用不同颜色区分不同的人和车）。

全景分割（Panorama Segmentation）是语义分割和实例分割的结合，即要将图像中的每个像素打上类别标签，又要区分出相同类别中的不同实例。

4. 计算机视觉的技术原理

图像分类和目标检测是很多计算机视觉任务背后的基础，接下来将简单地说明一下它们的运行原理。

1）图像分类的运行原理

图像分类的实现主要依靠基于深度学习的卷积神经网络。卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络（Feedforward Neural Networks，FNN），是深度学习的代表算法之一。卷积神经网络具有表征学习（Representation Learning）能力，能够按其阶层结构对输入信息进行平移不变分类（Shift-Invariant Classification），因此也被称为平移不变人工神经网络（Shift-Invariant Artificial Neural Networks，SIANN）。

那么人工神经网络（Artificial Neural Networks，ANN）又是什么呢？它是一种模仿生物神经网络（动物的中枢神经系统，特别是大脑）结构和功能的教学模型或计算模型，用于对函数进行估计或近似。人工神经网络由大量的人工神经元相互连接进行计算，大多数情况下人工神经网络能在外界信息的基础上改变内部结构，是一种自适应系统。

典型的人工神经网络具有以下三个部分。

- 网络结构：定义了网络中的变量和它们的拓扑关系。
- 激活函数：定义了神经元如何根据其他神经元的活动来改变自己的激励值。
- 学习规则：定义了网络中的权重如何随着时间推进而调整。

了解完人工神经网络后，接下来继续了解卷积神经网络，它分为输入层、隐藏层和输出层，具体如图 1-17 所示。

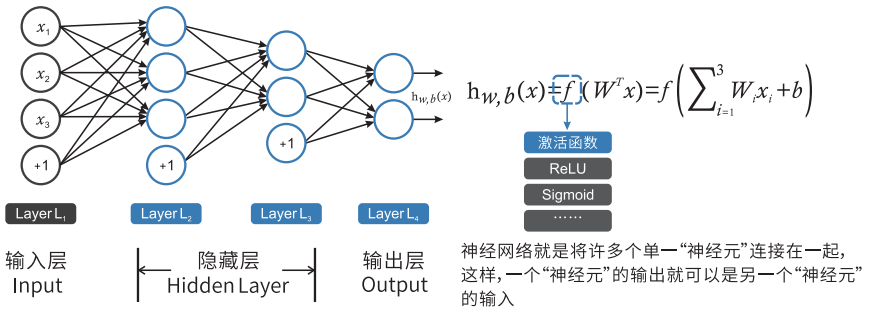

图 1-17 卷积神经网络原理图

输入层接收的是图像的三维数组，数组的形状大小为图像宽度、图像高度、图层数，数组的值为每一个图像通道逐个像素点的像素值。隐藏层主要包括卷积层（Convolutional Layer）、池化层（Pooling Layer）和全连接层（Fully-Connected Layer）。卷积层的功能是对输入数据进行特征提取，其内部包含多个卷积核，组成卷积核的每个元素都对应一个权重系数和一个偏差量，类似于一个前馈神经网络的神经元。卷积层内每个神经元都与前一层中位置接近的区域的多个神经元相连，区域的大小取决于卷积核的大小，也称作感受野，其含义可类比视觉皮层细胞的感受野。卷积核在工作时会有规律地扫过输入特征，在感受野内对输入特征做矩阵元素乘法求和并叠加偏差量。卷积层还包括卷积参数和激励函数，使用不同的参数或函数，可以用来调节卷积层卷积后获得的结果。在卷积层进行特征提取后，输出的特征图会被传递至池化层进行特征选择和信息过滤。池化层包含预设定的池化函数，其功能是将特征图中单个点的结果替换为其相邻区域的特征图统计量，可降低图像参数，加快计算，防止过拟合。卷积神经网络中的全连接层等价于传统前馈神经网络中的隐藏层（或称为隐含层）。全连接层位于卷积神经网络层隐藏层的最后部分，并只向其他全连接层传递信号。特征图在全连接层中会失去空间拓扑结构，被展开为向量并通过激励函数。在一些卷积神经网络中，全连接层的功能可由全局均值池化（Global Average Pooling）取代，全局均值池化会将特征图每个通道的所有值取平均，可降低计算量，加快运行速度，防止过拟合。卷积神经网络中输出层的上游通常是全连接层，因此其结构和工作原理与传统前馈神经网络中的输出层相同。对于图像分类问题，输出层使用逻辑函数或归一化指数函数（Softmax Function）输出分类标签。

卷积就像是拿扫描仪（滤波器，Filter）扫描图片，扫描仪扫完的结果一个一个压在一起，后面再用扫描仪接着扫，这里只展示了一个滤波器，实际上图片是有厚度的，也就是通道数（Channel），有图层，同样滤波器也是有厚度的，会获取不同通道的特征图。

图像分类网络的发展经历了 LeNet、AlexNet、GoogleNet、VGG、ResNet、MobileNet、EfficientNet 等阶段，其中由何恺明提出的 ResNet（残差神经网络）对近几年的图像分类乃至计算机视觉发展起到了至关重要的作用。ResNet 的产生简单来说就是为了更好地获取特征，人们希望使用更深层次的网络来实现，但是随之而来会出现很多问题，如梯度弥散（Vanishing Gradient）、梯度爆炸（Exploding Gradient）等，还会出现网络的退化，反向传播（Back Propagation）时无法有效地将梯度更新到前面的网络层，导致前面的网络层无法正确更新参数，

实际上超过 20 层的网络的效果反而不如之前，于是何恺明提出了 ResNet 来解决这个问题。

由于 ResNet 有残差连接（Skip Connection），梯度能够畅通无阻地通过各个残差块（Res Blocks），使得深层次的卷积神经网络也能够正常有效地运行。

2）目标检测的运行原理

目标检测可分为两个关键的子任务：目标分类和目标定位。目标分类任务负责判断输入图像或所选择图像区域（Proposals）中是否有感兴趣类别的物体出现，输出一系列带分数的标签表明感兴趣类别的物体出现在输入图像或所选择图像区域中的可能性。目标定位任务负责确定输入图像或所选择图像区域中感兴趣类别的物体的位置和范围，输出物体的边界框、物体中心或物体的闭合边界等，通常使用边界框（Bounding Box）来表示物体的位置信息。

算法模型大体可以分成两大类别：

- One-Stage 目标检测算法，这类检测算法不需要 Region Proposal 阶段，可以通过一个 Stage 直接产生物体的类别概率和位置坐标值，如 YOLOV3/4/5/X、SSD、RetinaNet。
- Two-Stage 目标检测算法，这类检测算法将检测问题划分为两个阶段，第一个阶段首先产生候选区域（Region Proposals），包含目标大概的位置信息，然后第二个阶段对候选区域进行分类和位置精修，如 Faster R-CNN。除此之外，还有 Anchor-Free 检测方法，构建模型时将目标作为一个点，即目标 BBox 的中心点。我们的检测器采用关键点估计来找到中心点，并回归到其他目标属性，不再需要进行非极大值抑制（Non-Maximum Suppression，NMS）等操作，如 CenterNet 等。

目前目标检测在落地应用中主要还是以 One-Stage 的 YOLO 系列为主，它既能够充分地保证项目运行的实时性，又能保证较高的模型准确率。其中 YOLOV3 可谓是目标检测发展的里程碑。YOLOV3 的主干网络为 Darknet53，主要由不同尺寸参数的卷积层和残差块组成，并进行多尺度融合训练，能够适应更多不同尺寸大小的图片，拥有较快的运行速度。现在广泛应用的 YOLOV4/V5/X 都是由 YOLOV3 衍变而来的。

5. 计算机视觉前沿技术

随着计算机视觉技术的不断发展，除图像分类、目标检测、图像分割等主要方向外，还有很多新的技术不断产生，生成式对抗网络（Generative Adversarial Networks，GAN）就是其中一个非常有代表性 的技术。GAN 是一种深度学习模型。模型通过框架中（至少）两个模块：生成模型（Generative Model）和判别模型（Discriminative Model）的互相博弈学习产生相当好的输出。原始 GAN 理论上并不要求 G（Generator）和 D（Discriminator）都是神经网络，只需要能拟合相应生成和判别的函数即可。但实际应用中一般使用深度神经网络作为 G 和 D 。一个优秀的 GAN 应用需要有良好的训练方法，否则可能由于神经网络模型的自由性而导致输出不理想。

GAN 的基本原理其实非常简单，这里以生成图片为例进行说明。假设我们有两个网络：G 和 D。正如它的名字，它们的功能分别是：G 是一个生成图片的网络，它接收一个随机的噪声 z，通过这个噪声生成图片，记作 G(z)。D 是一个判别网络，用于判别一幅图片是不是“真实的”。

它的输入参数是 x，x 代表一幅图片，输出 D(x) 代表 x 为真实图片的概率，如果为 1，就代表 100% 是真实的图片，而输出为 0，代表不可能是真实的图片。在训练过程中，生成网络 G 的目标就是尽量生成真实的图片来欺骗判别网络 D。而 D 的目标就是尽量把 G 生成的图片和真实的图片区分开来。这样，G 和 D 构成了一个动态的“博弈过程”。最后博弈的结果是什么？在最理想的状态下，G 可以生成足以“以假乱真”的图片 G(z)。对于 D 来说，它难以判定 G 生成的图片究竟是不是真实的，因此 D(G(z)) = 0.5。这样我们的目的就达成了：我们得到了一个生成式的模型 G，它可以用来生成图片。

由 GAN 和 GAN 衍生的网络在近几年也迸发出许多创新性的应用，例如风格迁移、图像修复、图像生成等。下面这组图展示了使用风格迁移技术完成时间风格迁移、智能上色、人脸风格迁移等功能。

6. 计算机视觉落地应用

计算机视觉是目前人工智能应用中最为广泛与普遍的，且早已深入日常生活与工作的多方面，典型的应用如生物特征识别中的人脸识别。人脸识别已经广泛应用在人证比对、身份核验、人脸支付、安防管控等各个领域。

现在主流的人脸识别技术多使用黄种人和白种人的面部特征进行模型开发和训练，并且样本中的光照条件良好，因此可以从图片中较好地识别和分析黄种人和白种人的人脸，但是，由于深肤色（如黑人）人脸图像的纹理特征较不明显，并且反光较强，因此现有的人脸识别方法对深肤色人脸的识别和分析存在缺陷，尤其无法应对中偏重黑人人脸光照不佳的情况，不能在视频和照片中很好地识别和分析深肤色人脸，也就是说，现有的人脸识别方法很难对深肤色人脸的特征进行有效分析和提取，从而导致对深肤色人脸的识别准确率较低。

我们采用了创新的方法，如增加拉普拉斯变换融合到图像图层等，采用基于深度学习的图像识别技术较好地解决了深肤色人种人脸识别的问题。主要流程如下。

1）人脸检测

人脸检测和关键点检测步骤采用了级联结构的卷积神经网络，可以适应环境变化和人脸不全等问题，且具有较快的检测速度。该方法规避了传统方法劣势的同时，兼具时间和性能两个优势。一幅图片中绝大部分区域容易区分出非人脸区域，只有少部分区域包含人脸和难以区分的非人脸区域。为加快检测速度，我们设计使用三级分类器，使得性能逐级提高。一般算法中只包含一个回归器，如果候选框与真实人脸框相差较大，则无法进行有效的回归。我们使用多级回归器，每一级回归器皆可让结构更接近真实人脸框，在多级回归后结果更准确。为实现对人脸/非人脸分类、人脸框回归、人脸关键点回归等预测，我们设计出基于多任务的深度学习模型。多任务学习提升了各个子任务的性能，达到 1+1 大于 2 的效果。在实现多任务共享深度学习模型参数的同时，较大地减少了模型运算量，大幅提高了人脸检测速度。

2）人脸识别

特征提取步骤采用了深层次的残差卷积神经网络，且具有优化的损失函数，对比传统方法

可以更快、更好地提取人脸特征，增加类间距，减少类内距，获得更好的人脸识别效果。该技术在残差卷积神经网络的基础上增加了更多的 Shortcut 网络连接与 Highway 卷积层连接，保证了特征生成网络中能够兼具挖掘出人脸样本图像中的浅层与深层纹理特征，并将多重纹理特征进行组合，生成可分性更强的人脸特征，增强人脸识别的准确率。同时，在特征生成网络中加入了多尺度融合机制，在卷积层中加入了多尺度视觉感受，保证了同一人在多方位图片中的人脸特征空间距离接近，有效提升同一分类的图像产生更好的聚类，进而提高人脸识别的准确率。

针对不同肤色人种的人脸识别，尤其是深肤色人种，我们使用提取纹理，通过对原始人脸图像进行拉普拉斯变换后得到的变换人脸图像描述人脸图像中的纹理强度，因此，由该原始人脸图像及该变换人脸图像进行拼接后得到的四通道人脸图像，相较于原始人脸图像来说，人脸纹理特征较明显，这样在基于深肤色人脸的四通道人脸图像进行特征提取时，可以更高效地获取人脸图像的纹理特征，进而可以提高深肤色人脸识别的准确率，业务流程示例如图 1-18 所示。

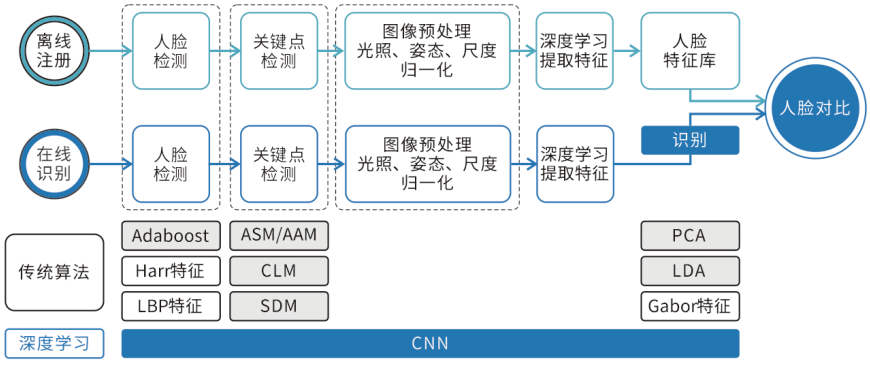

图 1-18　人脸识别业务流程示例

计算机视觉在光学字符识别领域的应用同样很广泛，如文档识别、证件识别、票据识别、视频文字识别、车牌识别等。车牌识别即识别图像中包含车牌的车牌号。基于深度学习的方法获取车牌在图像中的区域并进行光学文字识别。

车牌检测和关键点检测步骤采用了级联结构的卷积神经网络，可以适应车牌的不同倾斜角度、环境变化和车牌不全等问题，还能够将倾斜的车牌进行矫正对齐，方便车牌号识别，且具有较快的检测速度。车牌号识别采用深层卷积神经网络并加以连接式时序分类损失函数，可准确地识别不定长文字序列，具有较强的泛化性。同时，我们还采用了细粒度识别技术，能够精确地识别车型，甚至能够区分具体车型，如红旗 H9。

总结一下，如今计算机视觉已经广泛应用于人们日常生活的众多场景中。随着深度学习的飞速发展，计算机视觉融合了图像分类、目标检测、图像分割等技术，已在工业视觉检测、医疗影像分析、自动驾驶等多个领域落地应用，为各行各业捕捉和分析更多信息。

1.2.6　文本分析

数据科学是一门研究如何从数据中提取有用信息的学科。而文本分析则是数据科学领域中

一个非常重要的分支，可以帮助我们从大量的文本数据中提取有用的信息，了解文本数据中隐藏的洞见，并发现趋势和模式。数据科学和文本分析的结合能够让我们更好地利用文本数据中的信息，做出更加准确的决策。

数据科学和文本分析密不可分，文本数据是人类生产和交流信息的主要方式之一，随着数字化和互联网技术的发展，海量的文本数据不断涌现。数据科学旨在从数据中提取价值信息，以支持决策和创新。文本分析是数据科学中的一个重要领域，它致力于利用自然语言处理和机器学习技术对文本数据进行处理、分析和挖掘，从中提取有用的信息。

数据科学和文本分析的关系也可以从技术的角度来看。数据科学需要处理大量的数据，文本数据是其中一个重要的组成部分，但与结构化数据相比，文本数据的处理更加复杂和困难。文本数据具有天然的不确定性、模糊性和主观性，它们往往需要经过预处理、特征提取、模型训练和评估等多个环节才能够得到有用的信息。数据科学家需要掌握自然语言处理、机器学习和深度学习等多种技术，才能够有效地处理文本数据。

本书中文本分析的重要技术主要介绍预训练模型、多语种文本分析、文本情感分析、文本机器翻译、文本智能纠错等。

1. 预训练模型

在文本分析技术中，预训练模型是一种非常重要的技术手段。随着深度学习技术的不断发展，预训练模型已经成为自然语言处理领域的热门研究方向之一。本书中将介绍文本分析技术中预训练模型的相关内容。预训练模型是指在大规模语料库上进行无监督训练，从而学习出一定的语言表示，并将其作为下游自然语言处理任务的初始化参数或特征。预训练模型的思想是将大规模数据的统计规律内化到模型的权重参数中，通过对预训练模型进行微调，可以使得模型在目标任务上表现出更好的性能。

在文本分析技术中，预训练模型主要应用在两个方面：语言模型和表示学习。语言模型是指给定一个文本序列，预测下一个单词或者生成一段新的文本。通过预训练语言模型，可以学习到一个通用的语言表示，这个表示可以用于下游任务，例如文本分类、命名实体识别等。表示学习是指学习文本的向量表示，通过预训练模型可以学习到高质量的文本表示，这些表示可以用于相似度计算、聚类分析等任务。

预训练模型的应用越来越广泛，当前比较热门的预训练模型有 BERT（Bidirectional Encoder Representations from Transformers）、GPT（Generative Pre-trained Transformer）、RoBERTa（Robustly Optimized BERT Pre-Training Approach）等。BERT 是一种双向 Transformer 编码器，通过对大量语料进行预训练，得到了一个通用的语言表示。GPT 则是一种单向 Transformer 编码器，它可以生成新的文本序列。RoBERTa 是在 BERT 的基础上进行改进，通过更大规模的语料和更长的训练时间来提高模型的性能。预训练模型是文本分析技术中的重要手段之一，它能够学习到通用的语言表示，并可以用于下游任务的初始化和特征提取。当前已经有很多高质量的预训练模型可供使用，可以根据实际需求进行选择和微调，以得到更好的效果。

2. 多语种文本分析

多语种文本分析是指对多种语言的文本进行分析和处理的技术。随着全球化的不断深入和国际交流的增多，多语种文本分析变得越来越重要。多语种文本分析需要处理不同语言之间的差异和变化。语言之间的差异包括语法、词汇和发音等方面，这些都需要考虑到。因此，多语种文本分析需要使用自然语言处理技术来解决这些问题。自然语言处理技术可以处理语言中的语法和词汇问题，同时可以进行文本分类、关键词提取和情感分析等任务。

多语种文本分析在很多领域都有着广泛的应用。在商业领域中，多语种文本分析可以帮助企业了解不同语言用户的需求和反馈，从而改进产品和服务。在政府领域中，多语种文本分析可以帮助政府了解不同语言区域的民意和情况，从而制定更好的政策和措施。在文化领域中，多语种文本分析可以帮助人们了解不同语言的文化背景和特点，从而促进文化交流和沟通。

多语种文本分析技术在现代社会中具有重要的作用和意义。随着全球化的不断发展，这种技术将变得越来越重要，并且将在更多的领域得到应用和发展。

3. 文本情感分析

情感分析是文本分析领域的一项重要技术，旨在通过自然语言处理和机器学习等技术手段识别和分析文本中的情感色彩，如正面、负面或中性等情感倾向。情感分析技术被广泛应用于社交媒体分析、品牌管理、市场调研等领域，以帮助企业了解消费者对其品牌或产品的情感倾向，从而制定更有效的营销策略和改进方案。

在社交媒体分析领域，情感分析技术可以帮助企业快速了解消费者对其品牌或产品的看法，从而及时回应消费者的需求和投诉。例如，一些社交媒体监测工具可以通过对社交媒体上用户的评论和反馈进行情感分析，帮助企业发现并解决用户的不满意，提高用户满意度和口碑。

在品牌管理领域，情感分析技术可以帮助企业了解其品牌在公众心目中的形象和评价，从而进行品牌定位和形象塑造。例如，在一些消费品牌的宣传和广告中，情感分析技术可以帮助企业更好地了解消费者的需求和情感倾向，制定更有效的广告策略和宣传口径，提高品牌形象和知名度。

在市场调研领域，情感分析技术可以帮助企业了解其产品在市场上的表现和评价，从而进行产品改进和创新。例如，在一些产品上市前，企业可以通过情感分析技术对市场上的相关产品进行评价和比较，找到自身的优势和不足，从而进行产品的改进和创新，提高市场竞争力。

虽然情感分析技术在文本分析领域具有广泛的应用前景和优势，但它也面临一些挑战和限制。例如，情感分析技术需要对不同的语境和文化背景进行适配和调整，以保证情感分析的准确性和有效性；此外，情感分析技术也需要不断地进行模型训练和优化，以适应不断变化的语言和文本环境。尽管如此，随着技术的不断发展和进步，相信情感分析技术在未来将会有更为广泛和深入的应用。

4. 文本机器翻译

机器翻译是文本分析技术中的一项重要技术，它能够将一种自然语言翻译成另一种自然语

言。这项技术广泛应用于多语种文本分析、跨语言信息检索、国际化网站和应用程序等领域。

文本机器翻译早期采用的是基于规则的方法，但由于规则过于复杂，难以覆盖所有语言规则，导致翻译质量不佳。近年来，随着深度学习技术的发展，文本机器翻译逐渐采用基于神经网络的方法，其中以 Transformer 模型最为知名。Transformer 模型在机器翻译中的应用已经取得了极大的成功。该模型利用自注意力机制来学习源语言和目标语言之间的关系，避免了基于规则的传统翻译方法中所面临的规则复杂、翻译不准确等问题，进一步提高了翻译质量。

机器翻译技术的应用非常广泛，它可以帮助企业、政府机构和个人快速翻译各种文本，例如新闻报道、政策文件、商业合同、用户手册、社交媒体帖子等。机器翻译技术还可以帮助企业实现全球化，扩大市场，降低翻译成本，提高效率。此外，机器翻译技术还可以与其他文本分析技术结合使用，例如情感分析、实体识别和关系抽取等，以提高翻译质量和准确性。

机器翻译技术在文本分析领域具有重要的应用价值，它可以帮助人们更快速、更准确地理解和传递信息。随着人工智能和自然语言处理技术的不断发展，机器翻译技术的性能和应用范围也将不断扩大。未来，机器翻译技术将更加智能化，能够理解更复杂的语言结构和上下文，并能够根据不同的应用场景进行个性化的翻译。因此，机器翻译技术将成为未来文本分析领域不可或缺的一部分。

5. 文本智能纠错

文本智能纠错是指利用自然语言处理技术自动检测文本中的错误，并进行修正的技术。这项技术在许多场景下都具有重要意义，比如在撰写邮件、新闻稿、论文等文本时，错误的出现会影响读者对文本的理解和信任度。智能纠错能够对文本中出现的语法错误、拼写错误、语义错误等进行自动纠错。

智能纠错技术的应用场景非常广泛。在商业领域，智能纠错技术可以帮助企业提高客户服务质量、改善用户体验等。例如，在在线客服系统中，智能纠错技术可以快速地识别并自动纠正用户输入的问题描述，减少用户的烦恼和困扰。在学术研究中，智能纠错技术可以帮助研究人员进行文本数据的分析和挖掘，发现其中隐藏的信息和规律。

智能纠错技术相比传统的人工纠错方式具有很多优势。首先，智能纠错技术可以大幅度提高纠错的效率，尤其是在大规模文本处理方面。其次，智能纠错技术能够提高纠错的准确性，避免了人工纠错中的主观因素和误差。

当前，利用预训练模型来实现文本智能纠错的方法已经被广泛应用。例如，Google 在 2018 年推出的 BERT 模型中加入了一个任务——掩码语言模型（Masked Language Model，MLM），通过该任务可以让模型自动学习纠错的能力。同时，一些基于 BERT 的文本智能纠错工具，如 Microsoft 的 Editor 和 Grammarly 等，也在市场上广受欢迎。

文本分析技术已经成为数据科学中不可或缺的一部分。随着技术的不断进步，文本分析的应用场景也越来越广泛。预训练模型、多语种文本分析、文本情感分析、文本机器翻译、文本智能纠错等技术的不断发展为文本分析的应用提供了强有力的支持和推动。随着人工智能技术的不断发展，文本分析技术还将有更广泛的应用和更深入的研究。

1.2.7 知识图谱

在数据科学的实践中，行业客户对知识图谱的应用诉求愈发强烈，核心需求是将行业数据知识化，并通过搜索、推荐、问答，以及用知识辅助进行更加智能的决策。因此，如何将结构化和非结构化数据有效地治理起来，进行数据和知识挖掘，提取当中有价值的信息，并以可视化分析为政府和企业决策提供支持，成为当今亟待解决的问题。知识图谱作为大数据知识工程的典型代表，知识图谱技术近年来取得了长足进步，并在一系列实际应用中取得了显著效果。知识图谱之所以备受关注是因为业界普遍认为知识图谱是实现机器认知智能的基础。

但随着应用的深化，知识图谱的落地过程单靠其所代表的知识智能本身这套技术体系和范式已经难以解决很多问题：一是数据获取和治理困难；二是在知识层面，小样本、低资源情况下知识的表示和获取代价仍然非常大；此外，获取知识之后，在应用、服务能力方面也存在很多挑战。因此，未来破题的关键在于要突破以知识图谱为代表的知识智能的边界，向认知智能这样的智能新形态发展。认知智能作为数据智能、知识智能的融合创新产物，将是知识图谱等知识工程技术发展的必然归宿。近些年，人工智能逐渐从感知智能向认知智能发展，知识图谱则是实现认知智能的关键技术方法，在构建出知识图谱后，可以实现各种智能场景应用。未来知识图谱一定会深入各行各业，只有掌握通用的人工智能技术，并将技术和业务需求对应起来，才能真正发挥出知识图谱的价值，解决行业问题。

目前，半自动化结合人工是业内构建知识图谱所采用的主流方式，从长远来看，完全靠机器自动化，一点都不投入人工，目前不现实，也不可能存在。现在有很多知识图谱构建工程化的工具，在解决如何高效地抽取实体关系，如何做出映射、如何融合，以及如何通过预训练模型减少需要标注数据的数量等问题方面，只能说随着技术的发展和工具的发展，人工的工作量会逐渐降低，人工的效率会越来越高。但到什么时候，采用机器构建的比例比人工构建更多，现在还不好衡量，这是一个逐渐发展的过程。

另外，“人在闭环”是认知智能行业落地的必由之路，即在知识图谱构建和应用的过程中，人必须参与。必须要有人在，这是一个责任问题。机器适合做数据密集型和经验密集型的工作。而人适合做价值判断型或情感密集型的工作。任何一个在现实中有意义的业务，它的价值一定来自人。如果没有人的话，这个东西是没有价值的，所以不可能离开人。当前，已经进入一个从数据到知识的“智变”时代，随着大数据、知识图谱、自然语言处理（Natural Language Processing，NLP）等数据智能技术的进一步成熟，数据中的价值将不断被挖掘利用，帮助人们做出合理决策。

1.3 本章小结

数字经济迎来战略机遇期。数字经济正在成为重组全球要素资源、重塑全球经济结构、改变全球竞争格局的关键力量，发展数字经济是把握新一轮科技革命和产业变革新机遇的战略选

择。“十四五”的重点是产业数字化，即数字技术赋能传统行业，这是对数据科学的极大利好。

数据科学是为数字经济提供基础与技术支撑的学科，是有关数据价值链实现过程的基础理论与方法学。它运用建模、分析、计算和学习杂糅的方法研究从数据到信息、从信息到知识、从知识到决策的转换，并实现对现实世界的认知与操控。

数据科学的关键技术包括数据存储计算、数据治理、结构化数据分析、语音分析、视觉分析、文本分析、知识图谱等。数据科学平台围绕数据价值转化过程，将数据科学中的关键技术统一到一个平台上，打通数据集成、数据治理、数据建模、数据分析、数据应用等阶段。本章对这些关键技术进行了基本介绍，在接下来的章节中会介绍每一项关键技术的技术原理、方案并进行具体的源代码讲解，帮忙读者更好地掌握这些关键技术。

1.4 习　题

1. 什么是数据科学，数据科学与人工智能有什么联系和区别？

2. 了解关系数据库和非关系数据库之间的区别，并举例说明这两种数据库的使用场景和优劣势。

3. 在数据治理的背景下，“流程化”“自动化”和“智能化”分别指什么？如何应用这三个方面以增强数据治理实践？

4. 数据分析中有哪些监督模型？它们都有什么优劣势？

5. 在语音分析领域有什么典型的模型？

6. 什么是目标检测？请列举几种常见的目标检测算法并说明其原理。

7. 在文本分析中，预训练模型有什么特点？和普通的模型有什么不同？

8. 为了突破知识图谱的局限性，未来的发展方向是什么？为什么将认知智能作为知识图谱等知识工程技术发展的必然归宿？

1.5 本章参考文献

[1] https://docs.google.com/presentation/d/1RFfws_WdT2lBrURbPLxNJScUOR-ArQfCOJlGk4NYaYc/ edit?usp=sharing.

[2] McLaughlin J, Reynolds D A, Gleason T. A study of computation speed-ups of the GMM-UBM speaker recognition system[C]//Sixth European conference on speech communication and technology. 1999.

[3] Kenny P, Boulianne G, Ouellet P, et al. Joint factor analysis versus eigenchannels in speaker recognition[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(4): 1435-1447.

[4] Dehak N, Kenny P J, Dehak R, et al. Front-end factor analysis for speaker verification[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2010, 19(4): 788-798.

[5] Variani E, Lei X, McDermott E, et al. Deep neural networks for small footprint text-dependent speaker verification[C]//2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014: 4052-4056.

[6] Li C, Ma X, Jiang B, et al. Deep speaker: an end-to-end neural speaker embedding system[J]. arXiv preprint arXiv:1705.02304, 2017.

[7] https://www.cnblogs.com/wuxian11/p/6498699.html.

[8] https://blog.csdn.net/weixin_38206214/article/details/81096092.

[9] https://blog.csdn.net/KevinBetterQ/article/details/85476575.

[10] Sainath T N, He Y, Li B, et al. A streaming on-device end-to-end model surpassing server-side conventional model quality and latency[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 6059-6063.

[11] https://github.com/hirofumi0810/neural_sp.

[12] https://github.com/cywang97/StreamingTransformer.

[13] https://speech.ee.ntu.edu.tw/~hylee/dlhlp/2020-spring.html.

[14] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[15] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[16] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[17] Bolya D, Zhou C, Xiao F, et al. Yolact: Real-time instance segmentation[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 9157-9166.

[18] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[19] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4401-4410.

[20] https://github.com/deepinsight/insightface.

[21] https://github.com/facebookresearch/detectron2.

[22] https://venturebeat.com/2021/11/22/nvidias-latest-ai-tech-translates-text-into-landscape-images/.