《数据采集与预处理》

----复习资料

数据科学与大数据技术

编写：小林

日期：2026.05.16

目录

1. 概述 1
   1.1 数据 1
      1.1.1 数据的概念 1
      1.1.2 数据类型 1
      1.1.3 数据组成形式 1
      1.1.4 数据的价值性 1
      1.1.5 数据爆炸 1
   1.2 数据分析过程 1
   1.3 数据采集与预处理的任务 2
   1.4 数据采集 2
      1.4.1 数据采集概念和数据采集的三大要点 2
      1.4.2 数据采集的数据源 2
   1.5 数据清洗 2
      1.5.1 数据清洗的内容 2
      1.5.2 数据清洗的注意事项 3
      1.5.3 数据清洗基本流程与评价标准 3
   1.6 数据集成 3
   1.7 数据转换 3
   1.8 数据脱敏 4
2. 认识爬虫 4
   2.1 什么是网络爬虫 4
   2.2 网络爬虫的应用场景 4
   2.3 网络爬虫的工作原理和流程 4
   2.3.1 通用网络爬虫的工作原理 4
   2.3.2 聚焦网络爬虫的工作原理 4
   2.3.3 反爬虫应对策略 5
3. 解析网页数据 5
   3.1 Beautiful Soup 5
   3.2 浏览器加载网页过程 5
   3.3 正则表达式 5
   3.4 XPath 6
4. 提升网络爬虫速度 6
   4.1 网络爬虫速度提升方案 6
   4.2 多线程爬虫 6
5. 抓取动态网页数据 6
   5.1 Selenium 6
6. 数据仓库中的数据集成 6
   6.1 数据仓库概念 6
   6.2 数据集成 7
   6.3 ETL 7
   6.4 CDC 7
7. 分布式消息系统 Kafka 7

7.1 消息队列.....7
7.2 Kafka 的应用场景.....7
7.3 Kafka 的消息传递模式.....7
8. 日志采集系统 Flume.....8
9. 使用 Pandas 进行数据清洗.....8
9.1 NumPy 的常见函数.....8
9.2 Pandas 数据结构.....8

1. 概述

1.1 数据

1.1.1数据的概念

1. 数据是记录了客观事物的性质、状态以及相互关系的可识别、抽象的符号。

2. 数据与信息

数据是构成信息的基本单位，信息是数据的有序排列组合。

1.1.2数据类型

数据类型分为**文本、图片、音频和视频**。

1. 文本：以字符、文字、符号形式表示的信息
2. 图片：以像素矩阵形式表示的视觉信息，常见有点阵图和矢量图
3. 音频：以声波信号数字化形式表示的声音信息
4. 视频：以连续帧序列形式表示的动态视觉信息

1.1.3数据组成形式

数据按照数据结构分类分为：**结构化数据、半结构化数据和非结构化数据**

1. 结构化数据：具有预定义的数据模型，以行列形式组织，存储在关系型数据库中。

2. 半结构化数据：具有一定结构，但无严格模式定义，结构灵活可变。

3. 非结构化数据：没有预定义的数据模型，不便于用数据库二维表表示。

数据库的发展阶段经过了层次数据库、网状数据库、关系数据库和 NoSQL 数据库，当前时代的数据有 90%的非结构化数据。

1.1.4数据的价值性

为了某个特定的目的而被收集的数据，常有如**描述现状、解释原因、总结原因、进行预测和决策支撑**等作用。

数据的价值会因为不断重组而产生更大的价值，在不同领域可以重复使用，数据已经具备了资本的属性，可以用来创造经济价值。

1.1.5数据爆炸

人类社会正经历第二次数据爆炸，“**数据爆炸**”称为大数据时代的鲜明特征。数据爆炸也对科学研究提出更高要求，需设计出更灵活高效的数据存储、处理和分析工具，页将带来云计算、数据仓库、数据挖掘等技术和应用的提升或者根本性质改变。

1.2 数据分析过程

数据分析过程为：**数据采集与预处理、数据存储与管理、数据处理与分析和数据可视化**。

1. 数据采集与预处理：将外部各种数据源数据进行实时或非实时的采集、预处理以及利用。

2. 数据存储与管理：使用计算机硬件和软件技术，将数据进行存储和应用，发挥数据作用。

3. 数据处理与分析：使用统计学、机器学习和数据挖掘等方法，分析手机的数据，从中提取有用的数据。

4. 数据可视化：将数据以图形图像表示，可用于数据分析和开发工具。

1.3 数据采集与预处理的任务

1. 数据采集

从传感器和智能设备、企业在线系统、企业离线系统、社交和互联网平台等获取数据的过程。

2. 数据预处理

为后续的数据分析工作提供可靠和高质量的数据，减少数据集规模，提高数据抽象程度和数据分析效率。

1.4 数据采集

1.4.1数据采集概念和数据采集的三大要点

1. 概念

数据采集是指从各种数据源中获取原始数据，并将其转换为适合后续处理和分析形式的过程。

2. **全面性、多维性和高效性**是数据采集的三大要点

全面性：数据量足够具有分析价值；数据面足够支撑分析需求。

多维性：数据能满足分析需求（自定义数据多种属性和类型）。

高效性：技术执行的高效性、团队内部成员协同的高效性以及数据分析需求和目标实现的高效性。

1.4.2数据采集的数据源

常见数据采集的数据源为：传感器数据、互联网数据、日志文件和企业业务系统数据。

1.5 数据清洗

1.5.1数据清洗的内容

数据清洗主要是对**缺失值、重复值、异常值和数据类型有误**的数据进行处理，其内容如下。

1. **缺失值处理：**由于掉渣、编码和录入误差，数据中可能存在一些缺失值，需要给予适当的处理。常见的处理方法有估算、整例删除、变量删除和成对删除。

2. **异常值处理：**根据每个变量的合理取值范围和相互关系，检查数据是否合乎要求，发现超出正常范围、逻辑上不合理或者相互矛盾的数据。

3. **数据类型转换：**数据类型往往会影响到后续的数据处理分析环节，因此，需要明确每个字段的数据类型。

4. **重复值处理：**重复值的存在会影响数据分析和挖掘结果的准确性，所以，在数据分析和建模之前需要进行数据重复性检验，如果存在重复值，还需要进行重复值的删除。

1.5.2数据清洗的注意事项

1. 数据清洗是有限进行缺失值、异常值和数据类型转换的操作，最后进行重复值的处理。

2. 在对缺失值、异常值进行处理时，要根据业务的需求进行处理，这些处理并不是一成不变的，常见的填充包括：统计值填充、前/后值填充、零值填充。

3. 在数据清洗之前，最为重要的对数据表的查看，要了解表的结构和发现需要处理的值，这样才能将数据清洗彻底。

4. 数据量的大小关系着数据的处理方式。

5. 在导入数据表后，一般需要将所有列一个个地进行清洗，来保证数据处理的彻底性。

1.5.3数据清洗基本流程与评价标准

1. 基本流程

数据分析（对于原始数据源中存在的数据质量问题需要检测分析）、定义数据清洗的策略和规则（根据“脏数据”情况，指定策略和规则，选择合适的算法）、寻找并确定错误示例（自动检测属性错误和检测重复记录的算法）、纠正发现的错误（某些特定领域根据发现的错误模式，编制程序或者借助外部标准数据源文件、数据字典等，修正错误或者自动修正）。

2. 评价标准

数据清洗的评价标准包含：**数据的可信性、数据清洗的代价和数据的可用性**。

数据的可信性：精确性、完整性、一致性、有效性、唯一性。

数据的可用性：时间性、稳定性。

1.6 数据集成

1. 模式集成问题

由于不同数据源的数据模式（如属性名、数据类型、结构定义）存在差异，需要将它们统一为全局模式的过程。

2. 冗余问题

多个数据源中可能存在重复或可以推导出的相同信息，导致数据存储浪费和结果偏差，需要进行检测和消除。

3. 数据值冲突检测与消除

同一实体在不同数据源中的属性值不一致（如同一人的年龄在不同表中不同），需要识别并解决这些矛盾值。

1.7 数据转换

数据转换是将数据转换或归并成一个合适数据处理的描述形式。一下是常见转换策略。

1. 平滑处理：分箱、回归和聚类等。

2. 聚集处理：对数据进行汇总操作，这一操作常用于构造数据立方体或对数据进行多粒度的分析。

3. 属性构造处理：根据已有属性集构造新的属性，后续数据处理直接使用新增的属性。

4. 规范化处理：Min-Max 规范化、Z-Score 规范化和小数定标规范化等。

5. 数据泛化处理：用更抽象的概念来取代低层次的数据对象。

1.8 数据脱敏

数据脱敏是指对敏感信息进行变形、替换或加密处理，使其在保留数据可用性的同时，无法识别或追溯到特定个人或实体，从而保护数据隐私和安全。

2. 认识爬虫

2.1 什么是网络爬虫

网络爬虫又称网络蜘蛛、网络机器人，它是一种按照一定规则，自动浏览万维网的程序或脚本。与真人浏览万维网相比，网络爬虫能过够浏览的信息量更大，效率也更高。网络爬虫按照系统结构和实现技术大致可以分为 4 种类型：**通用网络爬虫、聚焦网络爬虫、增量式网络爬虫、深层网络爬虫**。

1. 通用网络爬虫：General Purpose Web Crawler 又称全网爬虫（Scalable Web Crawler），是指访问全互联网资源的网络爬虫。通用网络爬虫是“互联网时代”早期出现的传统网络爬虫，它是搜索引擎（如百度、谷歌、雅虎等）抓取系统的重要组成部分，主要用于将互联网中的网页下载到本地，形成一个互联网内容的镜像备份。

2. 聚焦网络爬虫：Focused Crawler 又称主题网络爬虫（Topical Crawler），是指有选择性地访问那些与预定主题相关网页的网络爬虫，它根据预先定义好的目标，有选择性地访问与目标主题相关的网页，获取所需要的数据。

3. 增量式网络爬虫：Incremental Web Crawler 是指对已下载的网页采取增量式更新，只抓取新产生或者已经发生变化的网页的网络爬虫。

4. 深层网络爬虫：Deep Web Crawler 是指抓取深层网页的网络爬虫，它要抓取的网页层次比较深，需要通过一定的附加策略才能够自动抓取，实现难度较大。

2.2 网络爬虫的应用场景

大多数依赖数据支撑的应用场景都离不开网络爬虫，包括**搜索引擎、舆情分析与监测、聚合平台、出行类软件**等。

2.3 网络爬虫的工作原理和流程

2.3.1 通用网络爬虫的工作原理

通用网络爬虫的采集目标是整个互联网上的所有网页，它会从一个或多个初始 URL 开始，获取初始 URL 对应的网页数据，并不断从该网页数据中抽取新的 URL 放到队列中，直至满足一定的条件后停止。

流程：获取初始 URL；抓取页面，并获得新 URL；将新 URL 放入 URL 队列；读取新 URL；是否满足停止条件。

2.3.2 聚焦网络爬虫的工作原理

聚焦网络爬虫面向有特殊需求的人群，它会根据预先设定的主题顺着某个垂直领域进行抓取，而不是漫无目的地随意抓取。

流程：根据需求确定好聚焦网络爬虫的采集目标，以及进行相关的描述；获取初始 URL；根据初始 URL 抓取页面，并获得新 URL；从新 URL 中过滤掉与采集目标无关的 URL；将过滤后的 URL 放到 URL 队列中；根据一定的抓取策略，从 URL 队列中确定 URL 优先级，并确定下一步要抓取的 URL；从下一步要抓取的 URL 中读取新 URL，以准备根据新 URL 抓取下一个网页；若聚焦网络爬虫满足设置的停止条件，或没有可获取的 URL 时，停止采集，若网络爬虫没有满足设置的停止条件，则继续根据新 URL 抓取对应的网页。

2.3.3 反爬虫应对策略

网络爬虫会采取一些应对策略继续访问网站，常见的应对策略包括添加 User-Agent 字段、降低访问频率、设置代理服务、识别验证码。

1. User-Agent 检测：检查请求头中的浏览器标识，拒绝非浏览器或已知爬虫工具的请求。
2. 访问频率限制：设置时间窗口内的最大请求次数（如每秒/每分/每小时 N 次）
3. IP 限制：限制单个 IP 的访问频率，超过阈值则封禁或要求验证码。
4. 验证码机制：对可疑请求弹出图形验证码、滑动验证、点选验证等。

3. 解析网页数据

3.1 Beautiful Soup

Beautiful Soup 是一个用于从 HTML 和 XML 文档中解析和提取数据的 Python 第三方库，它能够将复杂的网页源代码转换为易于操作的 Python 对象树，并提供了简洁直观的 API 来快速定位、遍历和提取页面中的特定元素及其属性，常与 Requests 库配合使用，是 Python 网络爬虫开发中最流行的 HTML 解析工具之一。

Beautiful Soup 的使用步骤

1. 创建 Beautiful 类的对象，根据 HTML 或 XML 文档或片段创建一个 Beautiful Soup 类的对象；
2. 定位节点，通过 Beautiful Soup 类的对象的查询方法或 CSS 选择器定位节点；
3. 提取节点，通过访问节点的属性或节点的名称提取文本。

3.2 浏览器加载网页过程

1. 浏览器通过域名系统（Domain Name System，DNS）服务器查找百度服务器对应的 IP 地址。
2. 浏览器向 IP 地址对应的 Web 服务器发送 HTTP 请求。
3. Web 服务器接收 HTTP 请求后进行处理，向浏览器返回 HTML 页面。
4. 浏览器对 HTML 页面进行渲染呈现给用户。

3.3 正则表达式

正则表达式是对字符串操作的一种逻辑公式，它会将实现定义好的一些特定字符，以及这些特定字符的组合，组成一个规则字符串，并且通过这个规则字符串表达对给定字符串的过滤逻辑。

3.4 XPath

XPath 是 XML 路径语言，用于从 HTML 或 XML 格式的数据种提取所需的数据，适合处理层次结构比较明显的数据，它能够基于 HTML 或 XML 的节点树确定目标节点所在的路径，顺着这个路径便可以找到节点对应的文本或属性值。

4. 提升网络爬虫速度

4.1 网络爬虫速度提升方案

影响网络爬虫速度的因素主要是网络 I/O 操作，原因在于网络 I/O 操作的速度赶不上 CPU 的处理速度。在 Python 中，**提升网络爬虫程序运行速度主要有三种方案，分别是多进程、多线程和协程**。

线程是**执行的最小单位**，而**进程**是**操作系统分配资源的最小单位**。一个进程由一个或多个线程组成，线程是一个集成中代码的不同执行路线。

多进程适用于 CPU 密集型的代码，如循环处理、大量的密集并行计算等；对线程适用于 I/O 密集型的代码，如文件处理、网络交互等；协程适用于大量不需要 CPU 的操作，如网络 I/O。

4.2 多线程爬虫

多线程爬虫的运行流程如下。

1. 构建一个网址队列，用于存放网络爬虫待爬取数据的所有网址；
2. 开启多个线程抓取网页；
3. 将抓取到的网页源码存储到网页源码队列中；
4. 开启多个线程对网页源码队列中的网页数据进行解析；
5. 将解析之后的数据存储到网页数据队列中；
6. 将网页数据队列中的数据进行存储。

5. 抓取动态网页数据

5.1 Selenium

Selenium 是一个开源的、便携式的自动化检测工具，它最初是为网站自动化测试而开发的，Selenium 支持与所有主流的浏览器（如 Chrome、Firefox、Edge、IE 等）配合使用，也包括注入 PhantomJS、Headless Chrome 等无界面的浏览器。

6. 数据仓库中的数据集成

6.1 数据仓库概念

数据仓库（Data Warehouse）是一个**面向主题的、集成的、相对稳定的、反映历史变化**的**数据集合**，用于支持管理决策。

实时主动数据仓库是一个集成的信息存储仓库，既具备批量和周期性的数据加载能力（采用 ETL 技术），也具备数据变化的实时探测、传播和加载能力（采用 CDC 技术），并能结合历史数据和新颖数据实现查询分析和自动规则触发，从而提供对战略决策和战术决策的双重支持。

6.2 数据集成

1. 实时主动数据仓库中数据集成方式：数据整合、数据联邦、数据传播、混合方式。

2. 数据分发方式：推（push）和拉（pull）、周期和非周期、一对一和一对多。

6.3 ETL

ETL 是将业务系统的数据经过抽取（Extract）、清洗转换（Transform）之后加载（Load）到数据仓库的过程，目的是将企业中的分散、零乱、标准不统一的数据整合到一起，为企业的决策提供分析依据。

ETL 的基本模块：**数据抽取、数据清洗与转换、数据加载**。

ETL 四种实现模式：**触发器模式、增量字段、全量同步、日志比对**。

市场上主流的 ETL 工具：Kettle、DataPipeline、Talend、Informatica、Datax、Oracle Goldengate。

6.4 CDC

CDC 的三个特性：**没有宕机时间、保持数据新颖性、减少系统开销**。

CDC 三个组成部分：**变化捕捉代理、变化数据服务、变化分发机制**。

变化数据捕捉技术的两个典型的应用场景：面向批处理的 CDC（pull CDC）和面向实时的 CDC（push CDC）。

7. 分布式消息系统 Kafka

7.1 消息队列

消息队列（Message Queue）是一种用来存储信息的队列，由生产者将数据从一端放入消息队列，由消费者按顺序从另一端进行去除使用。

7.2 Kafka 的应用场景

1. **应用解耦合**：多应用间通过消息队列对同一消息进行处理，避免调用接口失败导致整个过程失败；

2. **异步处理**：多应用对消息队列中同一消息进行处理，应用间并发处理消息，相比串行处理，减少处理时间；

3. **限流消峰**：广泛应用于秒杀或抢购活动中，避免流量过大导致应用系统挂掉的情况；

4. **消息驱动的系统**：系统分为消息队列、消息生产者、消息消费者，生产者负责生产消息，消费者（可能有多个）负责对小心进行处理。

7.3 Kafka 的消息传递模式

一个消费系统负责将数据从一个应用穿戴到另外一个应用，应用只需关注于数据，无需关注数据在两个或多个应用间是如何传递的。分布式消息传递基于可靠的消息队列，在客户端应用和消息系统之间一部传递消息。对于消息系统而言，一般有**两个主要的消息传递模式：点对点传递模式和发布订阅模式**。Kafka 就是一种发布订阅模式。

8. 日志采集系统 Flume

1. 通用日志采集系统组成

1) **日志采集：**主要负责提供多种方式、方法进行采集日志；

2) **存储分析：**主要实现分析统一存储和定制的场景分析日志；

3) **结果应用：**实现将日志分析结果提供服务接口或者默认的管理功能，供各种应用功能使用。

2. 系统日志分类

1) **用户行为日志：**记录系统用户使用系统是的一系列操作信息。

2) **业务变更日志：**根据需要记录特定业务场景。

3) **系统运行日志：**记录系统运行时，服务器资源、网络和基础中间件的实时运行状态，定期从不同设备采集信息进行记录。

3. 日志分析场景

1) **完善和提升系统：**应用程序通过分析用户行为，调查什么功能最受欢迎，对用户信息进行分析，得出某区域、某类用户为何种用户群体。

2) **内容推荐：**根据用户平时的阅读内容，收集相关日志并进行分析，通过完善算法，自动向用户推荐其感兴趣的内容，提高用户粘性。

3) **系统审计：**实时、完整、可靠地收集各类系统操作日志和安全事件日志，为后续的安全分析、合规审计、故障追溯和责任认定提供数据基础。

4) **自动化操作和维护：**针对系统运行环境日志进行自动采集和分析，根据预定规则可以实现服务器资源的预警和动态分配，有利于快速定位和故障排除。

9. 使用 Pandas 进行数据清洗

9.1 NumPy 的常见函数

函数	作用
zeros()	创建指定形状的全 0 数组
ones()	创建指定形状的全 1 数组
empty()	创建指定形状的未初始化数组（值随机，取决于内存状态）
eye()	创建 N×N 单位矩阵（对角线为 1，其余为 0）
full()	创建指定形状并用指定值填充的数组
random.random()	生成 [0, 1) 区间内均匀分布的随机浮点数数组
random.randint()	生成指定范围内的随机整数数组
ruandom.rand()	生成 [0, 1) 区间内均匀分布的随机浮点数（参数直接传维度，不用元组）
random.randn()	生成服从标准正态分布（均值 0，标准差 1）的随机浮点数数组

9.2 Pandas 数据结构

1. Series

Series 是一维带标签数组，可以存储任意数据类型（整数、字符串、浮点数等），由一组数据和对应的索引（Index）组成，类似于一维数组或 Excel 的一列数据。

2. DataFrame

DataFrame 是二维带标签数据结构，由多列 Series 组成，每列可以是不同的数据类型，类似于 Excel 表格或 SQL 数据表，是 Pandas 中最常用的数据结构。

3. 索引对象

Pandas 的索引（Index）对象负责管理轴标签和轴名称等，构建 Series 或 DataFrame 时，所用到的任何个数组或其他序列的标签都会转换成一个 Index 对象，该 Index 对象不可修改。