# 数据分析

本章具体描述了数据分析：介绍了本文中数据分析的概念，并引出针对不同类型数据的处理方法。

## 数据分析概述

需要提前说明的是，我们这部分指的数据分析不是当前火爆的从大量原始数据中进行分析，得出未知信息的数据分析，而是指对不同格式不同类型的源数据进行处理，使其格式统一化，方便后期数据存储处理。也许用数据提取或数据交换形容会更贴切些。在目前大部分针对普通用户的项目场景下，录入的数据格式多种多样，这些不同格式，不同结构的数据也被称为异构数据，异构数据源间的数据集成和转换过程几乎是必要的。只有对这些数据进行良好的合并整合，才能使得碎片数据发挥应用的作用。

### 数据分析的基本概念

### 数据分析方法

## 针对不同类型数据的分析方法

### Txt数据

### Excel数据转XML格式

### Xml/json格式

## 本章小结

这里我们简单考虑几种原始数据格式的处理情况，例如excel, txt, xml, json. 当它们同时存在时，我们应该统一化为什么形式？分别应该如何处理？

Microsoft Excel是微软公司发开的一款电子表格软件。直观的界面、丰富的功能模块和图表工具使Excel成为最流行的个人计算机数据处理软件。它简单易用，主要适用于非IT人员对数据的准备，编辑和存储，是保存统计数据最流行的工具。但相对专业的关系型等数据库，它在数据后期处理如查询，统计上仍是非常低效和麻烦，而且因为先天限制，数据冗余问题也很严重。而在我们假设的食物领域项目中，可以预想到，原始数据格式将以excel为主，txt为辅。所以我们需要首先处理Execl数据，将其转换为方便操作的其他格式。

XML一般指可扩展标记语言，主要用于标记电子文件和数据，使其具有结构性。结构化的数据可以方便搜索，分类，传输，这也是它被设计的目的。XML的示例如” <country><name>中国</name><province><name>黑龙江</name></province><province><name>广东</name></province></country> ”。它的标记均为自定义，所有标记成对出现，用/来区别一对标签的开始和结尾，用标记间的嵌套关系显示数据结构，整体是树形。它的结构非常简单，甚至有点繁琐，但就是这样机械重复的结构，可能不方便人使用，但是非常适合机器处理的。因为它的简单明了，在出现后迅速就成为了W3C提出的国际标准语言。总之，XML易于实现，可以用灵活清晰的方式管理数据整体结构，同时作为数据传输的标准语言受到广泛支持。

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它采用结构化的文本格式来存储和表示数据。层次结构简洁清晰，易于人编写阅读，易于机器解析或生成，还可以提升网络传输效率，是一种理想的数据交换语言。它的对象主要键值对表示，用方括号标记数组，花括号标记对象，对象间用逗号隔开。示例如” {"people":["firstName": "Brett","lastName":"McLaughlin"}, {"firstName":"Jason","lastName":"Hunter"}]} ”

由以上分析也可以看出XML和json的定位非常相近，都是一种数据交换格式，都使用结构化方法来标记数据。分析其不同点主要是JSON在Javascript场景下更合适；Json解析在事先知道数据结构下非常方便，但如果不知道结构，则费时费力。而XML解析则无此限制；同时，json的有效数据率高于xml。在选取使用的格式时，应根据具体情况进行选择。

目前转换excel数据为xml格式的方式主要有两种。一：使用软件。比如，我们可以先将excel文件导出为csv文件，这种用来交换电子表格的常用格式可以在浏览器中通过头文件加” application/vnd.ms-excel”进行显示或下载。或者用MySQL，SQLserver等功能齐全的数据库里包含的excel文件导入导出功能。 二：用编程语言开发api。这种方法比使用软件灵活度和性能高得多，但开发代码的过程较久，相比而言费时费力。在本文场景下，我们需要转换的excel格式不同，转换后存储的数据库也不同，使用软件限制太多，所以我们主要采取灵活性高的方案二，即使用编程语言开发代码解析excel数据，这里我使用的语言是python，它有一个方便的库xlrd，可以用来读取和格式化（format）excel数据。而XML库则可以处理XML文件。

Excel文件数据由表头，列名和对应数据组成。其中的表头列名部分是静态的。为了提高效率，我们先通过XML配置文件进行结构解析。这里我们以方便面属性为例。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 厂商 | 厂商编号 | 方便面编号 | 方便面名称 | 规格 | 是否加宽 |
| 五谷道场 | 001 | 20070528761 | 庖丁时蔬 | 袋装75g | 1 |
| 统一 | 002 | 20071011SY | 浓香排骨 | 袋装75g | 0 |

其对应的XML配置文件应如：。

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <manuInstNoodle>  < manufacturer >  < value excelCol=”厂商” colIndex=”1” colName=”manufacturerName” excelType=”LABEL” />  < value excelCol=”厂商编号” colIndex=”2” colName=”manufacturerId” excelType=”NUMBER” />  </ manufacturer >  < instantNoodel >  < value excelCol=”方便面编号” colIndex=”3” colName=”instantNoodelId” excelType=”LABEL” />  < value excelCol=”方便面名称” colIndex=”4” colName=”instantNoodelName” excelType=”LABEL” />  < value excelCol=”规格” colIndex=”5” colName=” specification” excelType=”LABEL” />  < value excelCol=”是否加宽” colIndex=”6” colName=”ifWiden” excelType=”NUMBER” />  </ instantNoodel >  </ manuInstNoodle > |

当我们要使用python进行数据转换时，先根据XML配置文件进行映射。读出每个单元格的数据，根据以上config文件里excelCol列名名称和colIndex列名索引进行对应。分析以上的配置文件，我们可以通过定义三个对象，分别是厂商，方便面，和他们间的关系来建立映射，也可以使用Python强大的数据类型进行组织，即

|  |
| --- |
| [ [ { manufacturerName : manufacturerNameValue}, { manufacturerId : manufacturerIdValue } ] , [ { instantNoodelId : instantNoodelIdValue }, { instantNoodelName : instantNoodelNameValue }, { specification : specificationValue } ,{ ifWiden : ifWidenValue } ] ] |

然后再通过def ListmapXML函数进行映射。这里方法二相对简单，我采取的就是这种方法。

Txt文件的数据提取我们则可以使用正则提取的方法，通过匹配完成结构化。这里我们举例如下，这是机器生成的文件machineDatas.txt。

|  |
| --- |
| TestID: B1J1A001;Force-1: 222.871;Area-FT: 1166.289; TestID: B1J1A002;Force-1: 222.27;Area-FT: 1093.455; |

可以写正则如:

|  |
| --- |
| machineData = data.findall(r”TestId(.+?)TestId”,str)  TestId = machineData.find(r“TestId:();”,str)  Area-FT = machineData.find(r“Area-FT:();”,str) |

通过python处理后即可映射为

|  |
| --- |
| <machineDatas>  < machineData>  <TestId>  B1J1A001  </TestId>  < Force-1>  222.871  </ Force-1>  < Area-FT >  1166.289  </ Area-FT >  </ machineData >  < machineData>  <TestId>  B1J1A001  </TestId>  < Force-1>  222.871  </ Force-1>  < Area-FT >  1166.289  </ Area-FT >  </ machineData >  </machineDatas> |

同时我们还应该考虑到，一个项目数据的录入时由多人做的，而每个人录入的每份数据可能格式有细微的差别。比如在文件夹“时间强度评价”下，提供的时间强度评价表一般有16列，包括产品id一列，从0s以30s为间隔到420s。但有个别文件缺失了360-420s的记录，有个别文件多了420-450s的记录。对于这种不完全统一的数据，我们在配置文件上应留下一些缓冲度，或在程度中体现出健壮性。例如：

|  |
| --- |
| #file1 |
| #file 2 |