第5章 数据分析

学习目标

- 了解数据分析；
- 了解数据仓库；
- 掌握 Hive 的操作；
- 掌握 HQL 语句的使用。

大数据价值链中最重要的一个环节就是数据分析，其目标是提取数据中隐藏的数据，提供有意义的建议以辅助制定正确的决策。通过数据分析，人们可以从杂乱无章的数据中萃取和提炼有价值的信息，进而找出研究对象的内在规律。本章将介绍如何通过数据分析技术对第 4 章预处理后的数据进行相关分析。

5.1 数据分析概述

数据分析是指用适当的统计分析方法对收集来的大量数据进行分析，从行业角度看，数据分析是基于某种行业目的，有针对性地进行收集、整理、加工和分析数据的过程，通过提取有用信息，从而形成相关结论，这一过程也是质量管理体系的支持过程。数据分析的作用包含推测或解释数据并确定如何使用数据、检查数据是否合法、为决策提供参考建议、诊断或推断错误原因以及预测未来等作用。

数据分析的方法主要分为单纯的数据加工方法、基于数理统计的数据分析、基于数据挖掘的数据分析以及基于大数据的数据分析。其中，单纯的数据加工方法包含描述性统计分析和相关分析；基于数理统计的数据分析包含方差、因子以及回归分析等；基于数据挖掘的数据分析包含聚类、分类和关联规则分析等；基于大数据的数据分析包含使用 Hadoop、Spark 和 Hive 等进行数据分析。本书通过使用基于大数据方法的数据分析技术的 Hive 对某招聘网站的职位数据进行分析。

5.2 Hive 数据仓库

5.2.1 什么是 Hive

Hive 是建立在 Hadoop 分布式文件系统上的数据仓库，它提供了一系列工具，能够对存储在 HDFS 中的数据进行数据提取、转换和加载，这是一种可以存储、查询和分析存储在

Hadoop 中的大规模数据的工具。

Hive 定义了简单的类 SQL 查询语言，称为 HQL，它可以将结构化的数据文件映射为一张数据表，允许熟悉 SQL 的用户查询数据，也允许熟悉 MapReduce 的开发者开发自定义的 mapper 和 reducer 来处理内建的 mapper 和 reducer 无法完成的复杂的分析工作，相对于 Java 代码编写的 MapReduce 来说，Hive 的优势更加明显。

由于 Hive 采用了类 SQL 的查询语言 HQL，因此很容易将 Hive 理解为数据库。其实从结构上来看，Hive 和数据库除了拥有类似的查询语言，再无类似之处。我们以传统数据库 MySQL 和 Hive 的对比为例，通过它们的对比来帮助读者理解 Hive 的特性，具体如表 5-1 所示。

表 5-1　Hive 与传统数据库对比

对　比　项	Hive	MySQL
查询语言	HQL	SQL
数据存储位置	HDFS	块设备、本地文件系统
数据格式	用户定义	系统决定
数据更新	不支持	支持
事务	不支持	支持
执行延迟	高	低
可扩展性	高	低
数据规模	大	小
多表插入	支持	不支持

5.2.2　设计 Hive 数据仓库

在数据仓库设计中，一般会围绕着星状模型和雪花模型来设计数据仓库的模型。在这里，针对招聘网站的职位数据分析项目，我们将 Hive 数据仓库设计为星状模型，星状模型是由一张事实表和多张维度表组成。接下来，通过一张图来讲解设计的星状模型数据仓库，具体如图 5-1 所示。

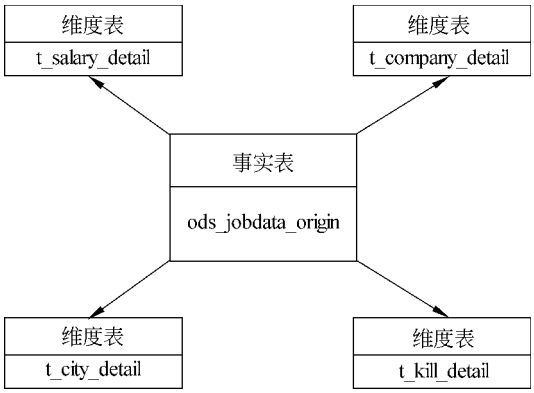

图 5-1　星状模型示意图

在图 5-1 中的，事实表为 ods_jobdata_origin 表(俗称窄表)，主要用于存储业务的主体数据；维度表有 t_salary_detail、t_company_detail、t_city_detail 以及 t_kill_detail 表，主要用于存储业务分析结果数据。

下面我们详细讲解数据仓库中事实表和维度表的表结构，具体如下。

1. 事实表 ods_jobdata_origin

事实表 ods_jobdata_origin 主要用于存储 MapReduce 计算框架清洗后的数据，其表结构如表 5-2 所示。

表 5-2 事实表 ods_jobdata_origin 的表结构

字 段	数 据 类 型	描 述
city	String	城市
salary	array<String>	薪资
company	array<String>	福利标签
kill	array<String>	技能标签

从表 5-2 可以看出，上述字段即为 MapReduce 初步预处理后的数据字段。

在表 5-2 中，事实表 ods_jobdata_origin 的表名前缀为 ods(Operational Data Store)，指的是操作型数据存储，作用是为使用者提供当前数据的状态，且具有及时性、操作性、集成性的全体数据信息。

2. 维度表 t_salary_detail

维度表 t_salary_detail 主要用于存储薪资分布分析的数据，其表结构如表 5-3 所示。

表 5-3 t_salary_detail 表

字 段	数 据 类 型	描 述
salary	String	薪资分布区间
count	Int	区间内出现薪资的频次

3. 维度表 t_company_detail

维度表 t_company_detail 主要用于存储福利标签分析的数据，其表结构如表 5-4 所示。

表 5-4 t_company_detail 表

字 段	数 据 类 型	描 述
company	String	每个福利标签
count	Int	每个福利标签的频次

4. 维度表 t_city_detail

维度表 t_city_detail 主要用于存储城市分布分析的数据，其表结构如表 5-5 所示。

表 5-5　t_city_detail 表

字　　段	数 据 类 型	描　　述
city	String	城市
count	Int	城市频次

5. 维度表 t_kill_detail

维度表 t_kill_detail 主要用于存储技能标签分析的数据，其表结构如表 5-6 所示。

表 5-6　t_kill_detail 表

字　　段	数 据 类 型	描　　述
kill	String	每个技能标签
count	Int	每个技能标签的频次

5.2.3　实现数据仓库

由于我们使用基于大数据分析方法的 Hive 对招聘网站的职位数据进行分析，因此需要将采集到的职位数据进行预处理后，加载到 Hive 数据仓库中，后续进行相关分析。Hive 数据仓库的部署步骤，具体如下。

1. 创建数据仓库

启动 Hadoop 集群后，在主节点 hadoop01 上启动 Hive 服务端，创建名为“jobdata”的数据仓库，命令如下。

```
hive >create database jobdata;
```

创建成功后通过 use 命令使用 jobdata 数据仓库，按照 5.2.2 节介绍的项目数据仓库模型，创建相应的表结构。

2. 创建事实表

创建存储原始职位数据的事实表 ods_jobdata_origin，命令如下。

```
hive >CREATE TABLE ods_jobdata_origin(
        city string COMMENT '城市',
        salary array<String>COMMENT '薪资',
        company array<String>COMMENT '福利',
        kill array<String>COMMENT '技能')
        COMMENT '原始职位数据表'
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        COLLECTION ITEMS TERMINATED BY '-'
        STORED AS TEXTFILE;
```

上述命令中，创建事实表语法的各项参数说明如下。

(1) CREATE TABLE：创建一个指定名字的表。

(2) COMMENT：后面跟的字符串是给表字段或者表内容添加注释说明的，虽然它对于表之间的计算没有影响，但是为了后期的维护，所以实际开发都是必须要加COMMENT的。

(3) ROW FORMAT DELIMITED：用来设置创建的表在加载数据的时候支持的列分隔符。不同列之间默认用一个'\001'分隔，集合(例如array，map)的元素之间默认以'\002'隔开，map中key和value默认用'\003'分隔。

(4) FIELDS TERMINATED BY：指定列分隔符，本数据表指定逗号作为列分隔符。

(5) COLLECTION ITEMS TERMINATED BY：指定array集合中各元素的分隔符，本数据表以“-”作为分隔符。

(6) STORED AS TEXTFILE：表示文件数据是纯文本。

3. 导入预处理数据到事实表

由于第4章数据预处理程序的运行结果将预处理完成的数据存储在HDFS上的/JobData/output/part-r-00000文件中，因此通过Hive的加载命令将HDFS上的数据加载到ODS层的事实表ods_jobdata_origin中，命令如下。

```
hive > LOAD DATA INPATH '/JobData/output/part - r - 00000' OVERWRITE INTO TABLE ods_jobdata_
origin;
```

通过select语句查看表数据内容，验证数据是否导入成功，命令如下。

```
hive > select * from ods_jobdata_origin;
```

如果返回数据信息，则证明数据加载成功，否则需要查看数据文件中分隔符是否与表设置的分隔符匹配、文件目录是否正确等细节问题。执行上述命令后的效果如图5-2所示。

从图5-2可以看出，执行完查询命令后，返回了数据，则说明数据加载成功。

4. 明细表的创建与加载数据

创建明细表ods_jobdata_detail，用于存储细化薪资字段的数据，即对薪资列进行分列处理，将薪资拆分形成高薪资、低薪资两列，并新增一列为平均薪资，平均薪资是通过最低薪资和最高薪资相加的平均值得出，命令如下。

```
hive > create table ods_jobdata_detail(
city string comment '城市',
salary array<String> comment '薪资',
company array<String> comment '福利',
kill array<String> comment '技能',
low_salary int comment '低薪资',
high_salary int comment '高薪资',
```

```
avg_salary double comment '平均薪资')
COMMENT '职位数据明细表'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

图 5-2　查询表数据

创建完明细表后，就可以向 ods_jobdata_detail 表中加载数据，加载数据的命令如下。

```
hive >insert overwrite table ods_jobdata_detail
select city,salary,company,kill,salary[0],salary[1],
(salary[0]+salary[1])/2 from ods_jobdata_origin;
```

需要注意的是，select 语句中字段的顺序应与明细表创建时的字段顺序一致。

5. 创建中间表

(1) 对薪资字段内容进行扁平化处理，将处理结果存储到临时中间表 t_ods_tmp_salary，命令如下。

```
hive > create table t_ods_tmp_salary as select explode(ojo.salary) from ods_jobdata_
origin ojo;
```

上述命令使用 explode()函数将 salary 字段的 array 类型数据进行遍历，提取数组中的每一条数据。

(2) 对 t_ods_tmp_salary 表的每一条数据进行泛化处理，将处理结果存储到中间表 t_ods_tmp_salary_dist，命令如下。

```
hive >create table t_ods_tmp_salary_dist as
select case when col>=0 and col<=5 then "0-5"
when col>=6 and col<=10 then "6-10"
when col>=11 and col<=15 then "11-15"
when col>=16 and col<=20 then "16-20"
when col>=21 and col<=25 then "21-25"
when col>=26 and col<=30 then "26-30"
when col>=31 and col<=35 then "31-35"
when col>=36 and col<=40 then "36-40"
when col>=41 and col<=45 then "41-45"
when col>=46 and col<=50 then "46-50"
when col>=51 and col<=55 then "51-55"
when col>=56 and col<=60 then "56-60"
when col>=61 and col<=65 then "61-65"
when col>=66 and col<=70 then "66-70"
when col>=71 and col<=75 then "71-75"
when col>=76 and col<=80 then "76-80"
when col>=81 and col<=85 then "81-85"
when col>=86 and col<=90 then "86-90"
when col>=91 and col<=95 then "91-95"
when col>=96 and col<=100 then "96-100"
when col>=101 then ">101" end from t_ods_tmp_salary;
```

上述命令使用条件判断函数 case，对 t_ods_tmp_salary 表的每一条数据进行泛化处理，指定每一条数据所在区间。

(3) 对福利标签字段内容进行扁平化处理，将处理结果存储到临时中间表 t_ods_tmp_company，命令如下。

```
hive >create table t_ods_tmp_company as select explode(ojo.company) from ods_jobdata_
origin ojo;
```

上述命令使用 explode()函数将 company 字段的 array 类型数据进行遍历，提取出数组中的每一条数据。

(4) 对技能标签字段内容进行扁平化处理，将处理结果存储到临时中间表 t_ods_tmp_kill，命令如下。

```
hive >create table t_ods_tmp_kill as select explode(ojo.kill) from ods_jobdata_origin ojo;
```

上述命令使用 explode()函数将 kill 字段的 array 类型数据进行遍历，提取出数组中的每一条数据。

6. 创建维度表

(1) 创建维度表 t_ods_kill，用于存储技能标签的统计结果，命令如下。

```
hive >create table t_ods_kill(
every_kill String comment '技能标签',
count int comment '词频')
COMMENT  '技能标签词频统计'
ROW FORMAT DELIMITED
fields terminated by ','
STORED AS TEXTFILE;
```

(2) 创建维度表 t_ods_company,用于存储福利标签的统计结果,命令如下。

```
hive >create table t_ods_company(
every_company String comment '福利标签',
count int comment '词频')
COMMENT '福利标签词频统计'
ROW FORMAT DELIMITED
fields terminated by ','
STORED AS TEXTFILE;
```

(3) 创建维度表 t_ods_salary,用于存储薪资分布的统计结果,命令如下。

```
hive >create table t_ods_salary(
every_partition String comment '薪资分布',
count int comment '聚合统计')
COMMENT '薪资分布聚合统计'
ROW FORMAT DELIMITED
fields terminated by ','
STORED AS TEXTFILE;
```

(4) 创建维度表 t_ods_city,用于存储城市的统计结果,命令如下。

```
hive >create table t_ods_city(
every_city String comment '城市',
count int comment '词频')
COMMENT '城市统计'
ROW FORMAT DELIMITED
fields terminated by ','
STORED AS TEXTFILE;
```

5.3　分析数据

在实际开发中,统计指标可能会不断地变化。这里指定的统计指标为招聘网站的职位区域分布、职位薪资统计、公司福利分析以及职位的技能要求统计。本节将通过编写 HQL 语句对职位区域、职位薪资、公司福利以及职位技能要求数据进行分析。

5.3.1　职位区域分析

通过对大数据相关职位区域分布的分析,帮助读者了解该职位在全国各城市的需求状

况，通过对事实表 ods_jobdata_origin 提取的城市字段数据进行统计分析，并将分析结果存储在维度表 t_ods_city 中，命令如下。

```
hive >insert overwrite table t_ods_city
select city,count(1) from ods_jobdata_origin group by city;
```

查看维度表 t_ods_city 中的分析结果，使用 sort by 参数对表中的 count 列进行逆序排序，命令如下。

```
hive >select * from t_ods_city sort by count desc;
```

执行上述语句后，效果如图 5-3 所示。

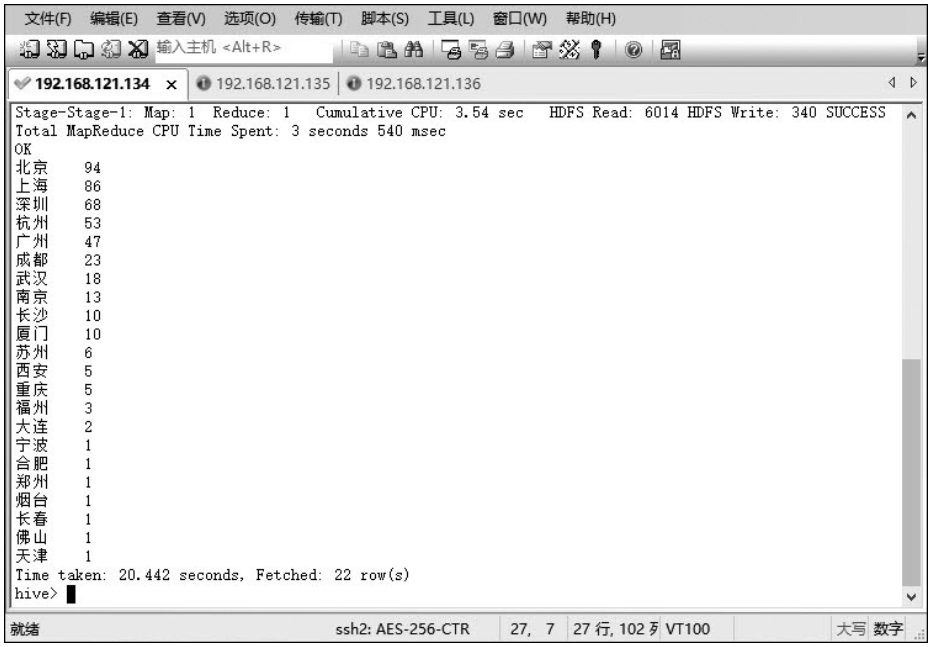

图 5-3 城市分布分析结果

通过观察图 5-3 的分析结果，可以得出如下三条结论。

（1）大数据职位的需求主要集中在大城市，其中最多的是北京，其次分别是上海和深圳。

（2）一线城市（北上广深）占据前几名的位置，然而杭州这座城市对大数据职位的需求也很高，超越广州，次于深圳，阿里巴巴这个互联网巨头应该起到不小的带领作用。

（3）四座一线城市北上广深加上杭州这五座城市的综合占总体需求的 77%（北上广深杭频次总和/所有城市频次总和），甩开其他城市很大一段距离，想参加大数据相关职位的从业者可先从这几个城市考虑，机遇相比会高出很多。

5.3.2 职位薪资分析

通过对职位薪资分析，了解大数据职位在全国以及在全国各城市的薪资情况，本节主要从三个分析点对薪资数据进行分析，具体操作流程如下所示。

1. 全国薪资分布情况

通过中间表 t_ods_tmp_salary_dist 提取薪资分布数据进行统计分析，将分析结果存储在维度表 t_ods_salary 中，命令如下。

```
hive >insert overwrite table t_ods_salary
select `_c0`,count(1) from t_ods_tmp_salary_dist group by `_c0`;
```

因为创建临时表 t_ods_tmp_salary_dist 时使用的是“create table as select”语句，该语句不可以指定列名，所以默认列名为 _c0、_c1，在访问的时候需要加上 ` 符号，需要注意的是这里的符号是 ` 而不是'，所以应该这样写：select `_c0` from xxx。

查看维度表 t_ods_salary 中的分析结果，使用 sort by 参数对表中的 count 列进行逆序排序，命令如下。

```
hive >select * from t_ods_salary sort by count desc;
```

执行上述语句后，效果如图 5-4 所示。

图 5-4　薪资分布结果

通过观察图 5-4 的分析结果，可以了解到全国大数据相关职位的月薪资分布主要集中在 11k～30k，在总体的薪资分布中占比达 77%（11～30 区间频次总和/全部频次总和），其中出现频次最高的月薪资区间在 11k～15k。

2. 薪资的平均值、中位数和众数

通过明细表 ods_jobdata_detail 提取 avg_salary（高薪资＋低薪资的平均值）数据进行