大数据：

**搜狗海量用户日志记录的分析系统**

项目背景：

搜狗搜索引擎每天有上百万条的搜索记录，通过对这些海量用户日志进行分析挖掘，对用户行为画像、智能推荐等方面具有重大意义。

主要职责：

主要对数据从以下几个方面做处理

* **数据获取**：搜狗实验室下载的五百万用户的搜索日志，主要是网页搜索用户查询及点击记录
* **数据清洗**：去除空白数据，转化部分数据，通过写shell脚本来扩展数据（将时间字段拆分并拼接，添加年、月、日、小时字段拆分）、数据过滤（过滤第二个字段uid或者第三个字段搜索关键词为空的行）、转换文件的分隔符号等
* **数据分析**：

**Hadoop：**

1.将数据上传到HDFS中

2.通过MapReduce程序实现统计频度最高的30词

3.基于Hive构建日志数据的数据仓库，创建外部表，灌入数据

通过HiveQL 实现对网站点击率（统计点击率最高的前十个网站）、点击次数与网站排名（点击排名的网站的排序）、查询时间（统计用户使用搜狗搜索的时间，）、URL搜索访问、查询关键词（查询频度最高的前50词，分析最受用户欢迎的搜索关键词）多个维度对数据分析

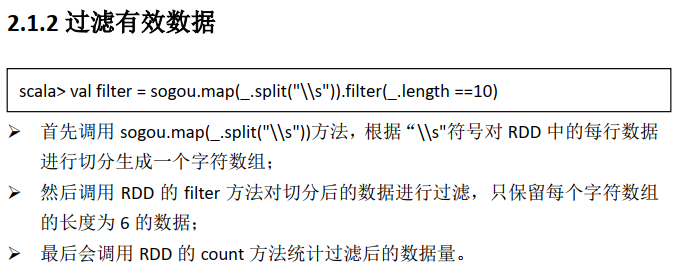
4.实时数据分析：使用sqoop将hive表里的数据导入mysql，

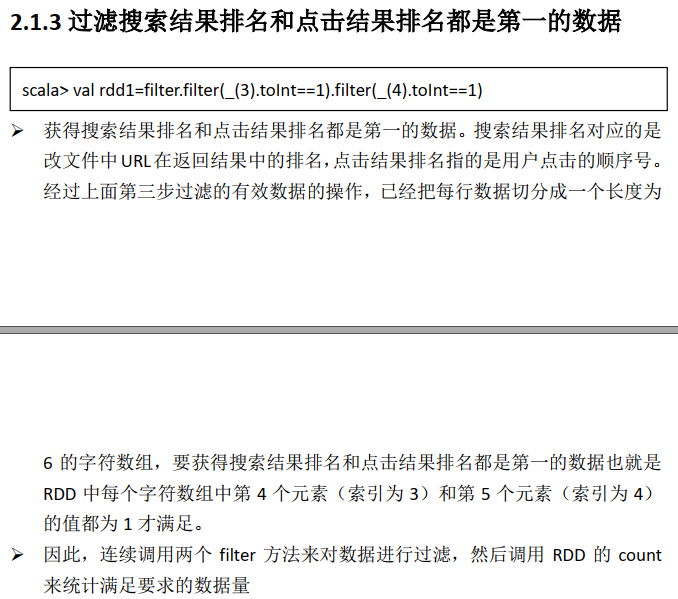
5.通过python的matlab库对数据实现可视化

**Spark： 详见spark实验报告**

1. 基于Spark，通过scala读取数据，过滤有效数据，编写Spark RDD实现同上的数据分析需求
2. 通过matplotlib对结果可视化





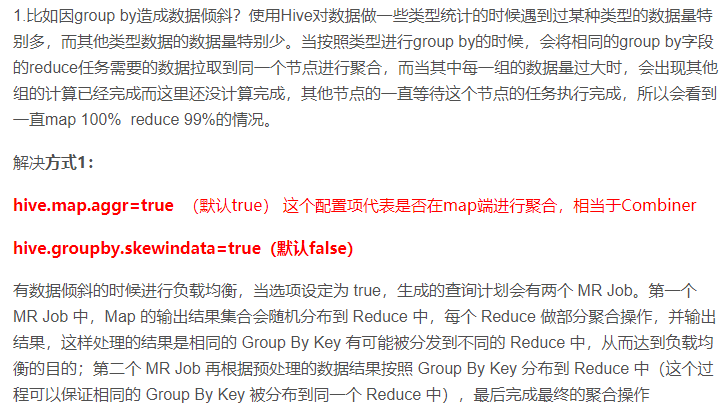


* **数据展示**：
* **问题及解决：**Hive实现对数据分析时出现数据倾斜问题（），通过优化HiveQL和MapReduce来解决

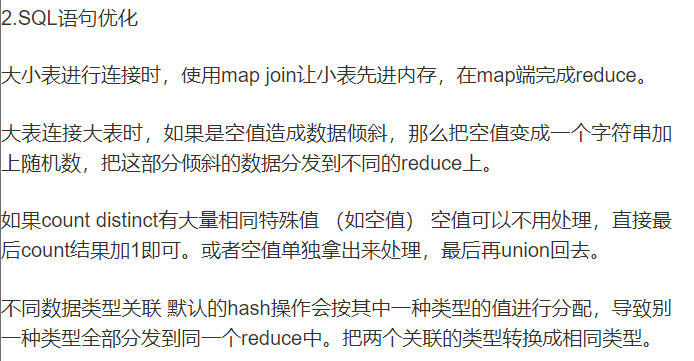
数据倾斜的原因：

基本都是发生了shuffle操作，在shuffle的过程中，出现了数据倾斜，或者某些key对应的数据，远远的高于其他的key。

发生在group by和join过程







**豆瓣社区网站用户的行为分析系统**

项目背景：

豆瓣，是一个社区网站，其描述及评论都由用户提供产生，是一个典型的UGC（User-Generated Content）网站，是一个由用户主导的社区网站，对用户进行分析画像可以更好地为用户提供服务，主要由三部分组成：读书、电影和音乐。主要的页面有两个：作品页面和用户个人主页。本项目主要是对用户行为做分析，故只爬取和分析用户主页的信息

主要职责：

负责通过python、Redis数据库、Echart等技术来实现用户数据的动态爬取、存储、构建数据仓库、处理分析以及可视化；通过关联规则Apriori算法为用户推荐电影。

两个问题：

* 如何爬取用户信息

第一个问题主要解决的是对于单个用户，我们需要爬取哪些信息以及如何爬取这些我们想要的信息。这个问题非常简单，只要通过HTTP协议获取到用户页面的HTML代码，通过正则表达式就可以获取特定的信息。这一部分我们将使用Python结合特定的库（BeautifulSoup库）进行实现。

* 如何让爬虫一直运行下去

第二个问题则是解决，如何一直获取稳定的用户ID。根据豆瓣的用户个人主页URL的生成规则，只要有了用户ID，我们就可以获取到用户个人主页的URL。为了让可视化的结果更加客观，我们需要对爬取的用户列表有一定的要求。有一定的数量，同时足够随机。为了实现这样的效果，我们将爬取用户信息与爬取用户列表进行了分离，这一部分我们将采用共享Redis数据库来实现。

* 数据获取：爬虫的基本流程：

1) 从 Redis待爬用户池中随机获取一个用户ID

2) 拼接用户ID成为一个URL

3) 根据URL发起HTTP请求，获取HTML代码文件

4) 对HTML进行解析，匹配需要的数据

5) 对数据进行序列化，存入Redis

**需要考虑的情况**：

 异常情况处理（需要登录、用户无效）

 伪造HTTP Header

 随机事件 sleep，防止爬取速度过快

 动态IP地址

**异常情况小结**：

1. 学院的公网IP只有三个，需要使用动态IP代理、洋葱网络或是使用手机流量的方式才能满足基本的爬取需求。
2. 一些数据也许有直接的接口，就不需要再去HTML代码里正则匹配了。
3. 尽可能避免那些需要登录的情况，也不要带着Cookie去爬，否则就是封号。

* 数据处理

数据处理主要是从Redis中读取原始数据、导入到MySQL进行初步分析、使用Python进行数据清洗、处理和导出成JSON格式文件。

**基本思路**：由于原始数据本身存在Redis中，为了方便进行一个初步的分析，我们将它导入到了MySQL并使用SQL语句进行一个简单的分析和处理。最后再用Python对数据进行扩展和导出。

* 在Redis中保存的是以**JSON**格式存储的原始信息。其从前到后的分别是：

 用户ID

 用户名

 用户头像地址

 注册时间

 常居地

 关注人数

 被关注人数

 加入小组数

 电影想看

 电影在看

 电影看过

 读书想看

 读书在看

 读书看过

* Mysql

将数据导入MySQL，用MySQL客户端进行查看树和初步分析

* Json

使用python对数据进行清洗、扩展和导出成适合可视化使用的json格式数据

* **数据可视化**

使用了jQuery + Fullpage.js + Echarts来展示

**用户加入时间**数据可视化 年、月、日

**用户常居地**数据可视化 用户主要集中在东部沿海地区、华中省会城市和直辖市

上海、背景占据了用户来源的top2，深圳作为四大一线城市之一在文艺青年的数量上稍逊一筹

海外用户中，一半来自于美国、英国、澳大利亚

**用户电影及月度数据**可视化

**用户关注及被关数据可视化**

**用户名和ID数据可视化 、**

用户昵称里，用的最多的中文字符是小、阿、一、大、我

英文字符最多的是 S M A s L

**综合数据**可视化

用户的各项数据随着加入时间的变长逐渐增大

北京用户最爱关注别人也最爱受别人关注、并且最喜爱阅读

上海人喜欢看电影和加小组

广州人最喜欢评论

电影与城市，北上差距不大，但【看过】的电影远超过其他城市

**基于深度学习的智能水表识别系统**

**项目背景**：

为解决传统手工抄表带来的各种弊端，本项目基于深度学习算法实现水表数字的智能识别，

主要负责通过 OpenCV、 Yolo、 SpringBoot 技术实现对水表图片的预处理、目标检测以及系统显示

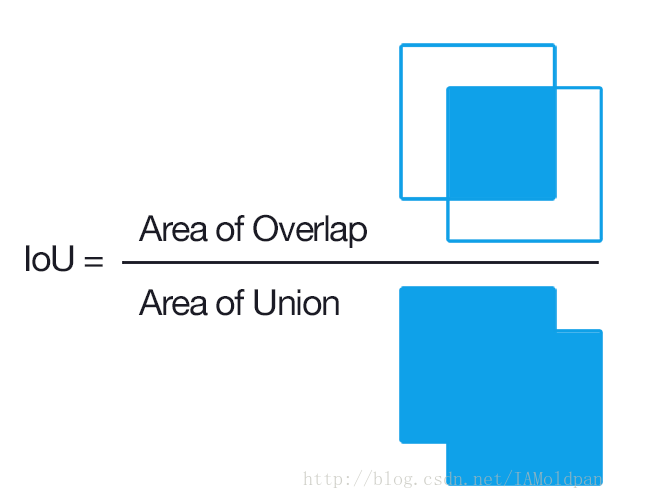
**核心思想：**

定位数字区域，opencv旋转后再做数字识别。

**问题及解决：**

1. 采用yolo来做第一步的切割，是由于图像输入太大4000\*3000，需求方说之后可能会加水表号识别，而且需求方说一张图片从输入到输出最多0.5s，yolo速度快、分类任务准确
2. 数字样本分布不均匀，0特别多，其他数字特别少，而且水表所处的环境恶劣，导致水表表面有杂物，光线不好，拍摄距离、角度、反光灯问题。通过opencv的高斯模糊、椒盐噪声、曝光处理，把两个问题结合一起，增强了数据集，1:3的比例，最后训练10000张
3. 水表数字与数字之间是凹陷关系，中间凹陷部分容易被判断成1，根据数字距离关系，良好的过滤掉了这个问题
4. 部分数字被遮挡，容易识别出两个框，比如7如果上半部分被遮挡，那么被识别出来7 1两个类，根据数字的实际情况，两个数字必然是不可能重叠，优化更新了NMS算法（非极大值抑制，抑制不是极大值的元素），适合这个项目本身。首先要过滤到夹缝的1，以IOU为阈值，设置为0.5，判断类之间是否重合，再取得置信度高的
5. 最后一层过滤，进位问题，由于仪表进位，会使得难以分辨，例如49进位，进位到一半，容易判断50 49 判断50可以，但是不能40 59.因此采用负反馈技术，若末位为9，发生进位关系，前一个框会识别出4 5两个数字，根据负反馈，应该选取小的那个，所以是49，这里负反馈无视置信度，即使是5高也要取4（假设最后两个十位数识别出两个框4 5，个位数识别出两个或者一个框 9 0，则可以判断发生了进位，那么在个位取9的情况下，十位必须取4，即时5的置信度高，也要取4）
6. 数字识别错误情况，要考虑对某个数字做数据集？现在是在loss中设置，比如0有100个，1有10个，正常来说，判错一个数字，损失记为1，但因为样本失调，会导致1的错误率很高，所以修改loss，如果0判错记作1,1判错记作10，**网络永远朝着最小损失的方向**，所以这个网络宁可0判错，也不会让1判错，这样就可以将1的正确率拉上来，达到平衡

Yolo的置信度 p\*iou （p是该类的概率）



**成果**：

经过对模型的不断的调整，使得图片数字区域目标定位 IOU （Intersection over Union）为 0.85+， 水表图片数字识别的准确率（除去认为判断错的，没识别出来或识别错了就算错，统计正确的除以总数）达 97+%，图片输入到输出速度为 0.5s。