|  |
| --- |
|  |
| 基于认知计算的就业咨询智慧服务系统 |
| 技术报告书 |
|  |

项目组成员：唐新晨、司舟、叶玉婷、汪昕濛、笪戴伟

目 录

[一、目的和意义 3](#_Toc448912126)

[二、技术方案 4](#_Toc448912127)

[2.1 业务流程 4](#_Toc448912128)

[2.2 方案设计 5](#_Toc448912129)

[三、平台设计及代码介绍 6](#_Toc448912130)

[3.1 数据采集单元设计方案 6](#_Toc448912131)

[3.1.1 Scrapy简介及其框架 6](#_Toc448912132)

[3.1.2 爬虫中遇到以下问题 8](#_Toc448912133)

[3.1.3 代码介绍 9](#_Toc448912134)

[3.2 数据计算平台设计方案 11](#_Toc448912135)

[3.2.1 朴素贝叶斯算法 11](#_Toc448912136)

[3.2.1.1 朴素贝叶斯算法介绍 11](#_Toc448912137)

[3.2.1.2 朴素贝叶斯算法的一般执行流程 12](#_Toc448912138)

[3.2.1.3 朴素贝叶斯算法的原理 12](#_Toc448912139)

[3.2.1.4 算法代码介绍 13](#_Toc448912140)

[3.2.1.5 构建算法中遇到的问题汇总 14](#_Toc448912141)

[3.2.2 SVM（支持向量机）算法部分 15](#_Toc448912142)

[3.2.2.1 SVM算法介绍 15](#_Toc448912143)

[3.2.2.2 SVM算法的一般执行流程 15](#_Toc448912144)

[3.2.2.3 SVM算法的原理 15](#_Toc448912145)

[3.2.2.4 算法代码介绍 21](#_Toc448912146)

[3.2.3 随机森林算法 27](#_Toc448912147)

[3.2.3.1 分布式存储与运算 27](#_Toc448912148)

[3.2.3.1.1 Hadoop 27](#_Toc448912149)

[3.2.3.1.2 HDFS 27](#_Toc448912150)

[3.2.3.1.3 MapReduce 28](#_Toc448912151)

[3.2.3.1.4 Spark 28](#_Toc448912152)

[3.2.3.2 随机森林算法 29](#_Toc448912153)

[2.1.3.3 算法实现 30](#_Toc448912154)

[3.3 用户服务平台设计及代码介绍 31](#_Toc448912155)

[3.3.1 Web前端 31](#_Toc448912156)

[3.3.1.1 AdminLTE框架的简介 31](#_Toc448912157)

[3.3.1.2 Bootstrap框架和jQuery库介绍 32](#_Toc448912158)

[3.3.1.3 AdminLTE框架的特性及使用 33](#_Toc448912159)

[3.3.2 Web后台 34](#_Toc448912160)

[3.3.2.1 web后端简介 34](#_Toc448912161)

[3.3.2.1.1 为什么要选择SSH框架以及MongDB数据库 34](#_Toc448912162)

[3.3.2.1.2 基于MVC的框架—Struts2 35](#_Toc448912163)

[3.3.2.1.3 基于持久层的框架—Hibernate 35](#_Toc448912164)

[3.3.2.1.4 基于IOC以及AOP的框架—Spring 35](#_Toc448912165)

[3.3.2.2 web后台的详细实现 35](#_Toc448912166)

[3.3.2.2.1 SSH框架的搭建 35](#_Toc448912167)

[3.3.2.2.2 Scrapy数据的采集（Data Collection） 36](#_Toc448912168)

[3.3.2.2.3 数据的处理（DataPresentation） 37](#_Toc448912169)

[3.3.2.2.4 求职岗位数据的展示 37](#_Toc448912170)

[3.3.2.2.5 求职简历信息的后台处理 39](#_Toc448912171)

[3.4 数据存储单元设计与实现方案 39](#_Toc448912172)

[3.4.1 MongoDB数据库简介 39](#_Toc448912173)

[3.4.2 数据库集群设计图 39](#_Toc448912174)

[3.4.3 副本集的设计 40](#_Toc448912175)

[3.4.3.1 副本集原理 40](#_Toc448912176)

[3.4.3.2 项目设计思路 40](#_Toc448912177)

[3.4.3.3 服务器配置 40](#_Toc448912178)

[3.4.4 读写分离设计 41](#_Toc448912179)

[3.4.4.1 设计思路 41](#_Toc448912180)

[3.4.4.2 效果展示 42](#_Toc448912181)

[四、系统功能展示 43](#_Toc448912182)

[4.1 软件概述及功能 43](#_Toc448912183)

[4.2 软件安装说明 43](#_Toc448912184)

[4.3 软件使用说明 44](#_Toc448912185)

[4.3.1 数据的采集过程 44](#_Toc448912186)

[4.3.2 数据的处理过程 45](#_Toc448912187)

[4.3.3 数据的展示过程 46](#_Toc448912188)

[4.4 针对个人简历来提供匹配合适的招聘信息 47](#_Toc448912189)

[五、总结 48](#_Toc448912190)

# 一、目的和意义

拥有百年历史的IBM于2015年12月18日在乌镇举行的第二届世界互联网大会上正式提出产业转型，将把认知计算确立为其重要战略支柱之一。在国外认知计算早在2002年就受到了政府、研究机构的高度重视。IBM在先后十几年间发力认知计算系统，并于2013年宣布成立“认知计算研究联合会”。我国于2013年10月11日在北京举办的以“从大数据到认知计算”为主题的认知计算研讨会，达成“我们已经进入了认知计算的新时代”的共识，在MDCC 2013移动开发者大会上中国工程学院院士李德毅发表题为《大数据时代的认知计算》的演讲，充分体现了当前我国学者对认知计算技术方面研究的高度重视。

认知计算（Cognitive Computing）是模拟人脑的计算机系统的人工智能，认知计算试图解决生物系统中的确定、不确定和部分真实的问题，以实现不同程度的感知、记忆、学习、语言、思维和问题解决等过程。目前随着智慧服务系统的发展和大数据时代的到来，如何实现类似人脑的认知与判决，实现新的关联和模式，从而做出正确的决策，显得尤为重要，这给认知计算技术的发展带来了新的机遇和挑战。

如今就业竞争愈发激烈，通过前期调研我们发现应届生存在因为对自身实力定位不当而造成就业困难的问题，而优秀的毕业生往往也会因为收获众多的offer而难以抉择。因此应当构建基于认知计算的就业咨询智慧服务系统，该系统具备理解（Understanding）能力和决策（Decision）能力：具有卓越的观察力和理解能力，能够定量分析往届毕业生简历内容（如：精通技能、项目经历等）与最终就职岗位的内在联系；具备洞察与发现（Discovery）能力：能够通过实时分析互联网上海量就业需求数据完成岗位分类，能够预测就业方向的变化等；具备辅助（Assistance）功能：能为院校学生求职岗位的选择提供智能化的辅助咨询服务。因此构建该系统能够有效为院校学生就业岗位的选择提供咨询与帮助。

# 二、技术方案

## 2.1 业务流程

该系统具体业务流程介绍如下图：

 图2.1-1 系统业务流程介绍图

（1）、Scrapy从各大招生信息网，获取就业实习信息，并进行数据预处理；

（2）、存储处理后的数据至数据存储单元；

（3）、通过走访、问卷调查、联合社团以及院校合作的方式获取前几届毕业生简历信息（包含掌握技能、社团经历等）和就职岗位信息，经过数据预处理后，存进数据存储单元；

（4）、数据计算平台取出数据，利用认知计算的相关算法，使用就业实习岗位需求信息生成“就业岗位智慧分类模型”，使用简历与就职岗位数据生成“就业智慧决策树模型”；

（5）、数据计算平台运用Model生成结果：使用“就业岗位智慧分类模型”，对岗位信息分类：技术类和非技术类，其中技术类分为：开发、测试和技术支持，并通过无监督式学习发现就业方向变化趋势；使用“就业智慧决策树模型”量化简历信息与岗位信息的内在联系，并能够通过建立信息，决策最终就职的岗位。所分岗位为：技术类和非技术类，其中技术类分为：开发、测试和技术支持。

（6）、数据计算平台存储结果数据

（7）、用户在用户服务平台输入简历信息

（8）、用户服务平台前端提取简历信息给平台后端

（9）、用户服务平台运用数据计算平台的Model得出适合岗位分类：用户服务平台使用“就业智慧决策树模型”为用户就业岗位提出意见

（10）、数据计算平台返回用户服务平台相关岗位信息

（11）、用户服务平台查询数据：用户服务平台依据得到的是技术类还是非技术类的相关岗位推荐，从数据存储单元查询相对应分类的实习和就业信息

（12）、用户服务平台取出数据：将对应分类的实习和就业信息返回至用户服务平台

（13）、短信推送给用户相关数据

（14）、Web显示相关数据

## 2.2 方案设计

本项目的技术方案设计包括五部分：

（1）设计并搭建数据采集单元：通过走访、问卷调查、联合社团与院校合作等方式选择获取近几年来高质量南京邮电大学在内的研究生简历以及最终就业单位、岗位信息。通过Scrapy爬虫框架，爬取各大就业信息网（南京邮电大学招生就业创业网、南大小百合BBS等）的就业信息，并对进行数据预处理。

（2）设计并搭建数据计算平台：使用认知计算中机器学习监督式学习多类别支持向量机、朴素贝叶斯算法，构造“就业岗位智慧分类模型”，对所提取的就业信息进行数据分类；采用随机森林算法，智能对用户简历信息进行数据分析，构造“就业智慧决策树模型”，洞察简历信息与就业岗位的内在联系，完成用户岗位信息的预测判决。搭建Hadoop+Spark分布式计算平台，提升模型分析处理能力。

（3）设计并搭建用户服务平台，使用SSH框架完成人机交互服务与业务逻辑设计，数据展示等。

（4）搭建数据存储单元，采用MongoDB数据库，完成数据存储，并配置用户登陆、副本集等功能，保障数据安全和冗余备份。

（5）迭代与验证模型环节：选择2016级应届生春招就业数据作为测试和再学习数据，验证就业咨询智慧服务系统：计算就业推荐岗位预测错误率和岗位分类错误率，并不断学习数据修正“就业岗位智慧分类模型”和“就业智慧决策树模型”。

# 三、平台设计及代码介绍

## 3.1 数据采集单元设计方案

### 3.1.1 Scrapy简介及其框架

目前有不少开源的网络爬虫的框架供我们使用。基本可分为三类：

（1）分布式爬虫（Nutch)：Nutch 是为搜索引擎设计的爬虫，其运行流程中，有三分之二是为了搜索引擎而设计的，对精确抽取没有太大意义。此外，Nutch依赖hadoop运行，hadoop本身会消耗很多时间，集群机器较少时爬取速度反而不如单机快。用Nutch进行爬虫的二次开发，爬虫的编写和调试所需的时间，往往是单机所需时间十倍以上。

（2）JAVA单机爬虫（Crawler4j、WebMagic、WebCollector）：把JAVA单独分为一类，是因为JAVA在网络爬虫已非常完善，相关资料也很健全。

（3）非JAVA单机爬虫（Scrapy）: Scrapy是一个使用Python编写的，轻量级的，简单轻巧，并且使用起来非常的方便。相比于JAVA，python可以用30行代码完成JAVA 50行代码干的任务。在针对爬取规模不大、爬取业务不复杂，使用scrapy爬虫是可以轻松完成爬虫任务的。

基于以上分析，我们选择scrapy作为此次项目的爬虫框架。

Scrapy是一个快速,高层次的屏幕抓取和web抓取框架，用于抓取web站点并从页面中提取结构化的数据。Scrapy用途广泛，可以用于数据挖掘、监测和自动化测试。Scrapy吸引人的地方在于它是一个框架，任何人都可以根据需求方便的修改。它也提供了多种类型爬虫的基类，如 BaseSpider、sitemap爬虫等，最新版本又提供了web2.0爬虫的支持。  
 所谓网络爬虫，就是抓取特定网站网页的HTML数据。抓取网页的一般方法是，定义一个入口页面，然后一般一个页面会有其他页面的URL，于是从当前页面获取到这些URL加入到爬虫的抓取队列中，然后进入到新页面后再递归的进行上述的操作。  
 Scrapy 使用 Twisted这个异步网络库来处理网络通讯，架构清晰，并且包含了各种中间件接口，可以灵活的完成各种需求。

下图显示了Scrapy的大体框架，其中包含了它的主要组件及系统的数据处理流程（绿色箭头所示）。

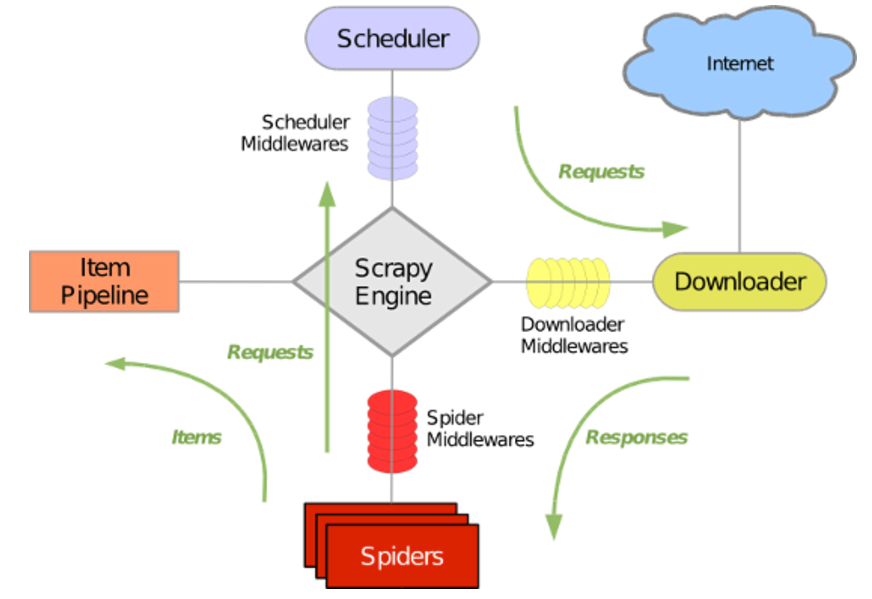


图3.1-1 Scrapy框架图

Scrapy主要包括了以下组件：

1. 引擎，用来处理整个系统的数据流处理，触发事务。

（2）调度器，用来接受引擎发过来的请求，压入队列中，并在引擎再次请求的时候返回。

（3）下载器，用于下载网页内容，并将网页内容返回给蜘蛛。

1. 蜘蛛，蜘蛛是用来制订特定域名或网页的解析规则。

（5）项目管道，负责处理有蜘蛛从网页中抽取的项目，主要任务是清晰、验证和存储数据。当页面被蜘蛛解析后，将被发送到项目管道，并经过几个特定的次序处理数据。

（6）下载器中间件，位于Scrapy引擎和下载器之间的钩子框架，主要是处理Scrapy引擎与下载器之间的请求及响应。

（7）蜘蛛中间件，介于Scrapy引擎和蜘蛛之间的钩子框架，主要工作是处理蜘蛛的响应输入和请求输出。

（8）调度中间件，介于Scrapy引擎和调度之间的中间件，从Scrapy引擎发送到调度的请求和响应。

使用Scrapy可以很方便的完成网上数据的采集工作，它为我们完成了大量的工作，而不需要自己费大力气去开发。

### 3.1.2 爬虫中遇到以下问题

（1）抓取到的json输出文件用unicode显示，中文可读性差。解决方法是在代码调试期间print爬取结果，方便调试和更改代码。

（2）需要同时运行多个scrapy。使用批处理文件.bat将多个运行语句汇总完成并行多个scrapy操作。

（3）动态网页的爬取。先通过url是否包含“?”判断一个网页是否是动态网页。初爬取时发现抓取的源代码与网页展现的内容不相符。经过调查发现，页面是JS动态生成的。

解决方法：在网上查询相关JS动态爬取方法，发现可以使用selenium来模拟生成动态页面，然后再静态爬取生成的页面。selenium是一个web自动化测试工具，可以通过pip install -U selenium语句来安装。安装完成后，selenium还需要调用浏览器才能实现动态页面的生成。这里选择chrome，下载chrome的驱动chrome driver使得selenium能够调用chrome浏览器。

（4）分页爬取及递归爬取。分页爬取采用递增url中的page值完成所需页面的遍历；递归爬取则要求在当前页面获得链接后进一步深度爬取，通过Request实现递归抓取。

（5）爬取页面时发现有访问限制，访问速度太快会导致禁止访问。

解决方法为在爬取间隔中加入访问速度控制语句time.sleep(1) 使得每次爬取都延迟1秒，降低读取速度。

### 3.1.3 代码介绍

项目中我们采用Scrapy完成招聘数据的大量采集，采集的目标网站为南京邮电大学就业创业网站（http://njupt.91job.gov.cn/job/search）、南京理工大学紫霞湖畔BBS（http://zixiahupan.com/forum.php?mod=forumdisplay&fid=107），东大虎踞龙蟠BBS（http://bbs.seu.edu.cn/bbsdoc.php?board=JobExpress），南京航空航天大学论坛求职招聘专栏（http://www.hkhtu.com/forum-44-1.html）。

以南邮网站为例介绍代码内容。文件夹名为NJUPT\_Intern目录中有以下文件：

NJUPT\_Intern/

scrapy.cfg

NJUPT\_Intern /

\_\_init\_\_.py

items.py

pipelines.py

settings.py

spiders/

\_\_init\_\_.py

...

这些文件分别是:

scrapy.cfg: 项目的配置文件

NJUPT\_Intern /: 该项目的python模块。之后将在此加入代码。

NJUPT\_Intern /items.py: 项目中的item文件，用于创建scrapy.Item类。

NJUPT\_Intern /pipelines.py: 项目中的pipelines文件。当Item在Spider中被收集之后，它将会被传递到Item Pipeline，一些组件会按照一定的顺序执行对Item的处理。item pipeline的一些典型应用：清理HTML数据、验证爬取的数据(检查item包含某些字段)、查重(并丢弃)、将爬取结果保存到数据库中。

NJUPT\_Intern /settings.py: 项目的设置文件。

NJUPT\_Intern /spiders/: 放置spider代码的目录。

Spider代码，保存在NJUPT\_Intern /spiders目录下的intern.py文件中。进入项目的根目录，执行下列命令启动spider：

scrapy crawl njuptintern

下面具体介绍NJUPT\_Intern /spiders/intern.py中代码。主要实现对南京邮电大学就业创业网站（http://njupt.91job.gov.cn/job/search）的页面信息爬取。网页显示如下图：



图3.1-2 爬取网址页面示意图

起始爬虫网址设置：

start\_urls = [*"http://njupt.91job.gov.cn/job/search?d\_category%5B0%5D=0&d\_category%5B1%5D=101&d\_category%5B2%5D=102&page=1"*]

分页爬取采用递增url中的page值完成所需页面的遍历：

page\_num = 34

for i in range(1,page\_num): next\_url=*"http://njupt.91job.gov.cn/job/search?d\_category%5B0%5D=0&d\_category%5B1%5D=101&d\_category%5B2%5D=102&page="* + str(i)

yield scrapy.Request(next\_url,callback = *self*.parse\_info)

爬取到的数据以.txt格式统一存储：

global file

file = open(*'D:\\Data\_Classification\\bayes\\test\\scrapy\\NUPT.txt'*,*'w'*)

提取页面的以下信息：职位名称（intern\_info）；发布日期（intern\_date）；链接网址（intern\_url）；详细信息（intern\_detail）， 其中，职位信息、发布日期、链接网址在本页提取，采用基于XPath和CSS的表达机制：Scrapy Selectors。使用选择器提取需要的信息，存入NjuptInternItem() 中：

item = NjuptInternItem()

item[*'intern\_info'*]=*''*.join(info.xpath(*"li[@class='span1']/a/@title"*).extract())

item[*'intern\_url'*]=*"http://njupt.91job.gov.cn"*+*''*.join(info.xpath(*"li[@class='span1']/a/@href"*).extract())

item[*'intern\_date'*]=*''*.join(info.xpath(*"li[@class='span4']/text()"*).extract())

其中，详细信息在递归页面中提取，通过Request实现递归抓取。

deep\_url=*"http://njupt.91job.gov.cn"*+*''*.join(info.xpath(*"li[@class='span1']/a/@href"*).extract())

yield Request( deep\_url, meta={*'item'*:item}, callback=*self*.parse\_item)

……

def **parse\_item**(*self*, response):

item = response.meta[*'item'*]

sel = Selector(response)

……

return item

最后完成的NUPT.txt显示如下图：

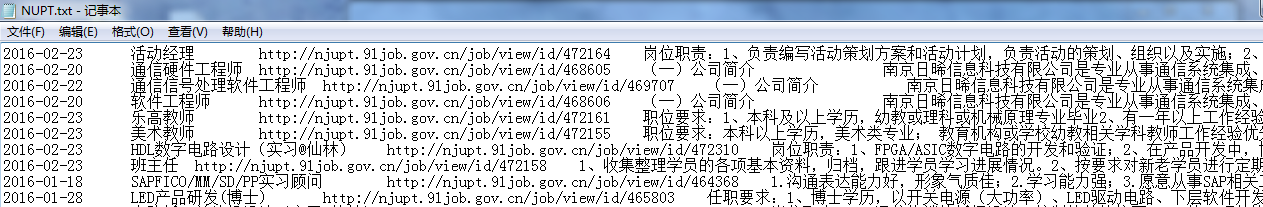


图3.1-3 爬取得到的信息示意图

至此，我们成功的完成了实习信息的爬取，整理和汇总。

## 3.2 数据计算平台设计方案

数据平台的设计用到以下三个算法：朴素贝叶斯算法，多类别SVM算法，随机森林算法。使用朴素贝叶斯算法完成岗位信息的技术与非技术分类；使用多类别SVM完成与技术相关的开发、测试、技术支持和其他的分类；使用随机森林算法完成职位预测功能。

### 3.2.1 朴素贝叶斯算法

#### 3.2.1.1 朴素贝叶斯算法介绍

机器学习的一个重要应用就是文档的自动分类。在文档分类中，整个文档（如一封电子邮件）是实例，而电子邮件中的某些元素则构成特征。我们可以观察文档中出现的词，并把每个词的出现或者不出现作为一个特征，这样得到的特征数目就会跟词汇表中的词目一样多。朴素贝叶斯是贝叶斯分类器的一个扩展，是用于文档分类的常用算法。它在数据较少的情况下仍然有效，并且可以处理多类别问题。

#### 3.2.1.2 朴素贝叶斯算法的一般执行流程

朴素贝叶斯算法的执行流程如下：

(1)收集数据：可以使用任何方法。

(2)准备数据：需要数值型或者布尔型数据。

(3)分析数据：有大量特征时，绘制特征作用不大，此时使用直方图效果更好。

(4)训练算法：计算不同的独立特征的条件概率。

(5)测试算法：计算错误率。

(6)使用算法：一个常见的朴素贝叶斯应用是文档分类。可以在任意的分类场景中使用朴素贝叶斯命类器，不一定非要是文本

#### 3.2.1.3 朴素贝叶斯算法的原理

朴素贝叶斯方法，是指基于贝叶斯定理的情况下，特征向量之间彼此独立。原理叙述如下：

根据贝叶斯定理，对一个分类问题，给定样本特征x，样本属于类别y的概率是

p(y|x)=p(x|y)p(y)p(x) （1）  
其中x是一个特征向量，将设x维度为M。因为朴素的假设，即特征条件独立，根据全概率公式展开，公式（1）可以表达为

p(y=ck|x)=∏Mi=1p(xi|y=ck)p(y=ck)∑kp(y=ck)∏Mi=1P(xi|y=ck) （2）  
其中只需分别估计出，特征xi 在每一类的条件概率就可以了。类别y的先验概率可以通过训练集算出，同样通过训练集上的统计，可以得出对应每一类上的，条件独立的特征对应的条件概率向量。

下面简单介绍一下，数据共分为训练集和测试集两个部分。

第一部分是使用训练集训练分类模型。训练集TrainingSet={(x1,y1),(x2,y2),...,(xN,yN)} 包含N条训练数据，其中 xi=(x(1)i,x(2)i,...,x(M)i)T 是M维向量，yi∈{c1,c2,...cK} 属于K类中的一类。

首先计算公式（2）中的p(y=ck)

p(y=ck)=∑Ni=1I（yi=ck）N （3）

其中I(x) 为指示函数，若括号内成立，则计1，否则为0。

其次计算分子中的条件概率，设M 维特征的第j 维有L 个取值，则某维特征的某个取值ajl ，在给定某分类ck 下的条件概率为：

p(xj=ajl|y=ck)=∑Ni=1I(xji=ajl,yi=ck)∑Ni=1I(yi=ck) （4）

经过上述步骤，我们就得到了模型的基本概率

第二部分是分类功能的实现：通过学到的概率，给定未分类新实例X ，就可以通过上述概率进行计算，得到该实例属于各类的后验概率p(y=ck|X) ，因为对所有的类来说，公式（2）中分母的值都相同，所以只计算分子部分即可，具体步骤如下：

（1）、计算该实例属于y=ck 类的概率

p(y=ck|X)=p(y=ck)∏j=1np(X(j)=x(j)|y=ck) （5）（2）、确定该实例所属的分类y

y=argmaxckp(y=ck|X) （6）

从而得到分类结果

#### 3.2.1.4 算法代码介绍

本课题使用了朴素贝叶斯算法，对采集到的实习就业信息进行“技术与非技术”的样本分类。

算法的具体执行过程分为以下几步：

第一步进行数据准备过程：

（1）进行训练集和测试集数据预处理：在handledata.py和SplitWord.java使用了MMAnalyzer的分词组件对训练集数据进行分词处理，并提取训练集中的label标签，构建vocabulary字典。Bayes\_SplitWord.java中对爬取得测试数据进行分词处理。

（2）在bayes\_1.py中使用loadDataSet()载入处理好的数据集，createVocabList()方法读取生成的vocabulary字典。

第二步训练算法：我们在bayes\_1.py中的trainNB0()重写了贝叶斯准则，将之前的x,y替换为w（w表示一个向量）。



之后对每个类计算该值，然后比较这两个概率值得大小。通过类别i(技术或非技术)中文档数除以总的文档书来计算。在计算时我们使用了朴素贝叶斯的条件独立性假设。将写为 认为每个向量之间彼此独立，上式可进一步写为。

具体的代码如下：

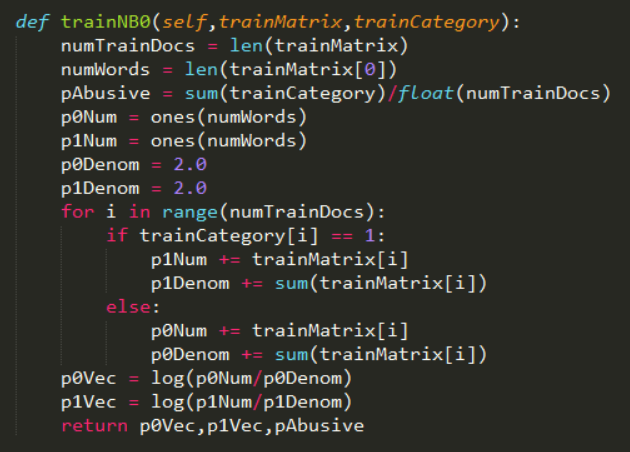


图3.2-1 代码示例图

第三步测试算法：在bayes\_1.py中使用classifyNB完成分类的预测

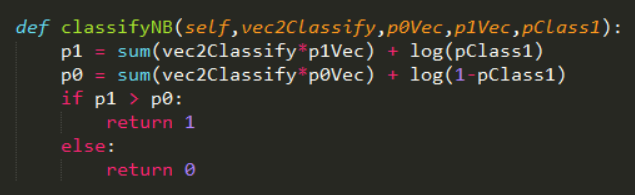
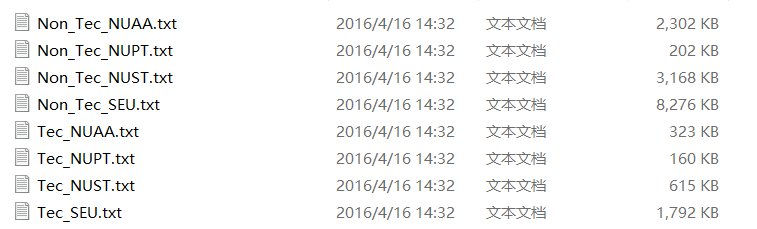


图3.2-2代码示例图

分类完成的数据存放于D:\Data\_Classification\bayes\outcome中



#### 3.2.1.5 构建算法中遇到的问题汇总

（1）当使用贝叶斯分类器对文档进行分类时，要计算多个概率的乘积以获得文档属于某个类别的概率，比如。其中某一个概率若为0的话，那么最后的乘积也为0。为降低这种影响，可以将所有词的出现数初始化为1，并将分母初始化为2。

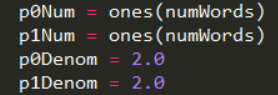


图3.2-3

（2） 当乘积较小时由于大部分因子较小，会造成下溢出，因此使用ln(a\*b) = ln(a)+ln(b)，通过求取对数从而避免了下溢出或者浮点数舍入导致的错误。



图3.2-4

### 3.2.2 SVM（支持向量机）算法部分

#### 3.2.2.1 SVM算法介绍

支持向量机(support vector machine)是一种分类算法，通过寻求结构化风险最小来提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。通俗来讲，它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

#### 3.2.2.2 SVM算法的一般执行流程

⑴收集数据：可以使用任意方法。

(2)准备数据：需要数值型数据。

(3)分析数据：有助于可视化分隔超平面。

(4)训练算法：SVM的大部分时间都源自训练，该过程主要实现两个参数的调优。

(5)测试算法：十分简单的计算过程就可以实现。

(6)使用算法：几乎所有分类问题都可以使用SVM，值得一提的是，SVM本身是一个二类分类器，对多类问题应用SVM需要对代码做一些修改。

#### 3.2.2.3 SVM算法的原理

具体原理：

1. 在n维空间中找到一个分类超平面，将空间上的点分类。如下图是线性分类的例子。



图3.2-5 线性分类的例子

一般而言，一个点距离超平面的远近可以表示为分类预测的确信或准确程度。SVM就是要最大化这个间隔值。而在虚线上的点便叫做支持向量Supprot Verctor。

分割超平面的形式可以写为，若计算任意一点到达超平面的距离就是。这里的常数b是指截距。在实际的分类过程中，通过使用下类似海维赛德阶跃函数（即单位阶跃函数）的函数对作用得到 f(),其中当u<0时£(u)输出-1, 反之则输出+1。因此SVM的核心就是寻找具有最小间隔的数据点，并利用这些点对间隔进行最大化，从而确定w和b的值。这样具体的表达式可以写为：

，通过引入拉格朗日乘子，使用基于约束条件表达式对目标函数进行进一步的解析，最终可以写为：



约束条件为：

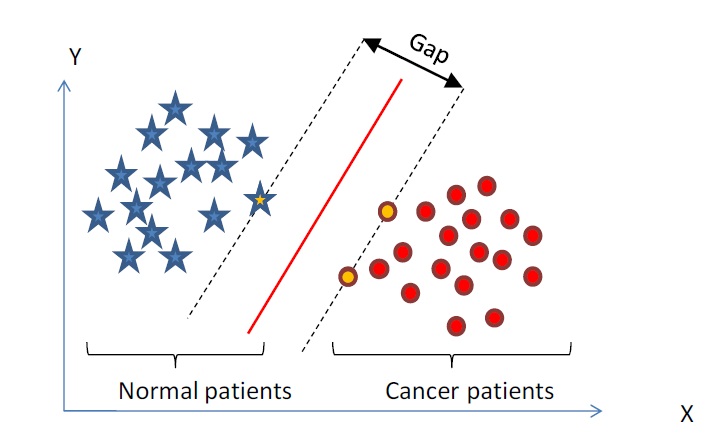


图3.2-6

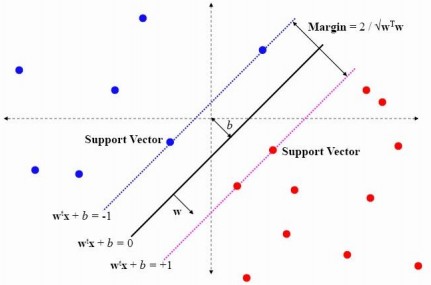


图3.2-7

3. 实际中，我们会经常遇到线性不可分的样例，此时，我们的常用做法是把样例特征映射到高维空间中去(如下图)；

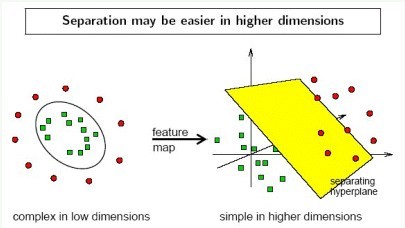


图3.2-8

线性不可分映射到高维空间，可能会导致维度大小高到可怕的(19维乃至无穷维的例子)，导致计算复杂。核函数的价值在于它虽然也是讲特征进行从低维到高维的转换，但核函数绝就绝在它事先在低维上进行计算，而将实质上的分类效果表现在了高维上，也就避免了直接在高维空间中的复杂计算。

4.使用松弛变量处理数据噪音

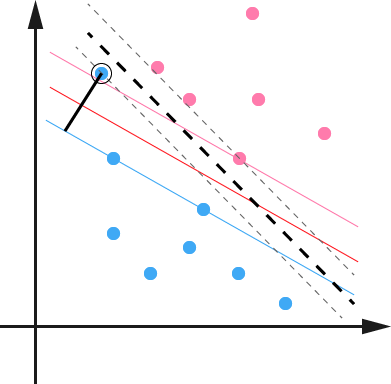


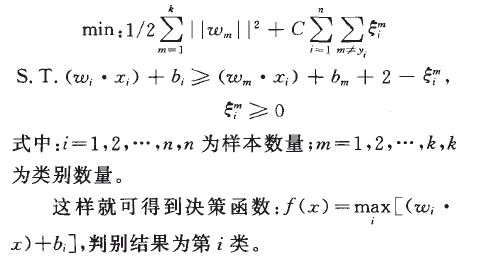
图3.2-9

此时需要将约束条件改为，其中常数C用于控制“最大化间隔”和“保证大部分点的函数间隔小于0.1”这两个目标的权重。

5.SVM多类分类方法

SVM多类分类方法的实现根据其指导思想大致有两种：

（1）将多类问题分解为一系列SVM可直接求解的两类问题，基于这一系列SVM求解结果得出最终判别结果。

（2）通过对前面所述支持向量分类机中的原始最优化问题的适当改变，使得它能同时计算出所有多类分类决策函数，从而“一次性”地实现多类分类。原始问题可以改写为：  
[](http://photo.blog.sina.com.cn/showpic.html#blogid=5eef0840010147pa&url=http://s1.sinaimg.cn/orignal/5eef0840g79582f0fde10)  
虽然第(2)种指导思想看起来简单，但由于它的最优化问题求解过程太复杂，计算量太大，实现起来比较困难，因此未被广泛应用。而基于第(1)种指导思想的SVM多类分类方法主要有5种。

1、一对其余法

     一类对余类法(One versus rest，OVR)是最早出现也是目前应用最为广泛的方法之一，其步骤是构造k个两类分类机(设共有志个类别)，其中第i个分类机把第i类同余下的各类划分开，训练时第i个分类机取训练集中第i类为正类，其余类别点为负类进行训练。判别时，输入信号分别经过k个分类机共得到k个输出值fi(x)=sgn(gi(x))，若只有一个+1出现，则其对应类别为输入信号类别；实际情况下构造的决策函数总是有误差的，若输出不只一个+1(不只一类声称它属于自己)，或者没有一个输出为+1(即没有一个类声称它属于自己)，则比较g(x)输出值，最大者对应类别为输入的类别。

    这种方法的优点是，对k类问题，只需要训练k个两类分类支持向量机，故其所得到的分类函数的个数(k个)较少，其分类速度相对较快。

2、一对一

     该方法在每两类问训练一个分类器，因此对于一个k类问题，将有k(k-1)／2个分类函数。当对一个未知样本进行分类时，每个分类器都对其类别进行判断．并为相应的类别“投上一票”，最后得票最多的类别即作为该未知样本的类别。决策阶段采用投票法，可能存在多个类的票数相同的情况，从而使未知样本同时属于多个类别，影响分类精度。

3、DAG方法（有向无环图）

     DAG-SvMS是由PIatt提出的决策导向的循环图DAG导出的，是针对“一对一"SvMS存在误分，拒分现象提出的。这种方法的训练过程类似于“一对一”方法，k类别问题需要求解k(k-1)／2个支持向量机分类器，这些分类器构成一个有向无环图。该有向无环图中含有k(k-1)／2个内部节点和k个叶结点，每个节点对应一个二类分类器。

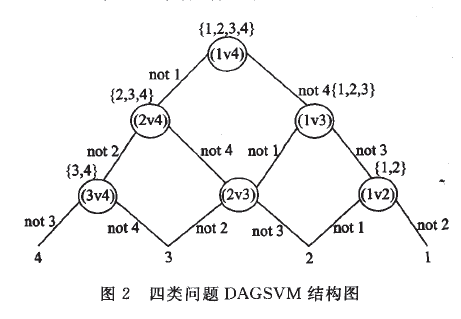
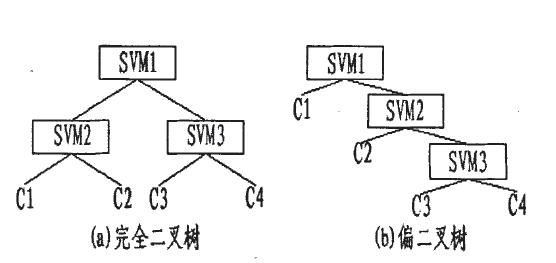
[](http://photo.blog.sina.com.cn/showpic.html#blogid=5eef0840010147pa&url=http://s4.sinaimg.cn/orignal/5eef0840gbd72fde8d163)

图3.2-10

DAG-SVMS简单易行，只需要使用k一1个决策函数即可得出结果，较“一对一"方法提高了测试速度，而且不存在误分、拒分区域；另外，由于其特殊的结构，故有一定的容错性，分类精度较一般的二叉树方法高。然而，由于存在自上而下的“误差积累”现象是层次结构固有弊端，故DAG-SVMS也逃脱不掉。即如果在某个结点上发生了分类错误，则会把分类错误延续到该结点的后续结点上.

4、决策树方法

      决策树的基本思想是从根节点开始，采用某种方法将该节点所包含的类别划分为两个子类，然后再对两个子类进一步划分，如此循环，直到子类中只包含一个类别为止，这样，就得到了一个倒立的二叉树。最后，在二叉树各决策节点训练支持向量机分类器，实现对识别样本的分类。决策树支持向量机多分类方法有很多种，不同方法的主要区别在于设计树结构的方法不同。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=5eef0840010147pa&url=http://s13.sinaimg.cn/orignal/5eef0840xbd763e243b4c)  
     图3.2-11

完全二叉树结构分类时使用的平均分类器数目为log2k，偏二叉树使用的平均分类器数为(k+1)／2-1／k，具有其他层次结构的二叉树使用的分类器平均值介于二者之间。完全二叉树分类时所需要的分类器数目最少，因此具有较少支持向量的完全二叉树的分类器速度也是较快的。  
5、纠错输出编码法（ECOC）

对于K类分类问题，可以根据不同方法构造一系列的两类分类问题，对于每个两类分类问题可以建立一决策函数。共得到L个决策函数，如果这些决策函数完全正确，K类中的每一类都对应一个元素为-l或+1的长度为L的数列，按照K类中的第一类、第二类，...，第K类的顺序，把这些数列排列起来，便可得到一个K行L列的编码矩阵，若要判断一个测试输入点的归属，首先用所得到的L个决策函数，得到一个元素为-l或l的长度为L的数列，然后将此数列与先前得到矩阵比较，相应于矩阵中有一行且仅有一行向与此数列相同，这个行数就是输入点的归属类；若矩阵中没有一行与该数列相同，可以通过计算汉明距离找出最近的一行，改行对应的类别即为该点的类别。

6.LIbSVM简介

LibSVM是台湾林智仁(Chih-Jen Lin) 教授2001年开发的一套支持向量机的库，该库运算速度快，可以很方便的对数据做分类或回归。由于libSVM程序小，运用灵活，输入参数少，并且是开源的，易于扩展，因此成为目前国内应用最多的SVM的库。

#### 3.2.2.4 算法代码介绍

我们通过SVM算法完成对技术信息的职位的再次细分为“开发、测试、技术支持、其他”共四大类。由于LibSVM已经提供了非常完善的SVM库能够实现多类别分类过程，因此我们的工作大致分为以下几个步骤：

（1）选择文本训练数据集和测试数据集：训练集和测试集都是类标签已知的；

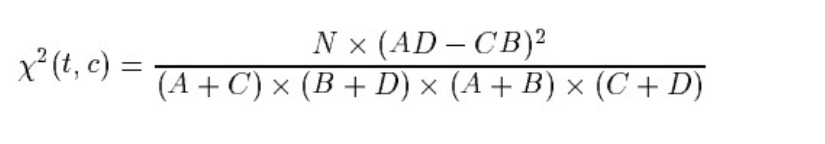
这里的文本训练集是由Scrapy从网上爬取得各大招生就业信息，经过朴素贝叶斯分类后形成的所有技术相关的就业信息，大家使用人工的方式先将技术相关的就业信息分类，并用“0：开发、1：测试、2：技术支持、3：其他”进行标注。测试数据集同样由Scrapy从网上爬取。

（2）训练集文本预处理：这里主要包括分词、去停用词、建立词袋模型（倒排表）；我们使用了MMAnalyzer完成分词的操作，使用停用词字典完成停用词去除，并将字典保存于vocab变量中。



图3.2-12

选择文本分类使用的特征向量（词向量）：最终的目标是使得最终选出的特征向量在多个类别之间具有一定的类别区分度，可以使用相关有效的技术去实现特征向量的选择，由于分词后得到大量的词，通过选择降维技术能很好地减少计算量，还能维持分类的精度；

基于论文《A comparative study on feature selection in text categorization》，我们选择基于卡方统计量（chi-square statistic， CHI）技术来实现选择，这里根据计算公式：  
  
其中，公式中各个参数的含义，说明如下：

* N：训练数据集文档总数
* A：在一个类别中，包含某个词的文档的数量
* B：在一个类别中，排除该类别，其他类别包含某个词的文档的数量
* C：在一个类别中，不包含某个词的文档的数量
* D：在一个类别中，不包含某个词也不在该类别中的文档的数量

使用卡方统计量，为每个类别下的每个词都进行计算得到一个CHI值，然后对这个类别下的所有的词基于CHI值进行排序，选择出最大的topN个词；最后将多个类别下选择的多组topN个词进行合并，得到最终的特征向量。对每个向量进行量化，并需要对训练集中的文档进行量化处理。

具体代码如下：

common.FeatureMap.java完成整个过程的调度；

train.FeatureTermVectorSelector.java文件文件提供了具体的方法，

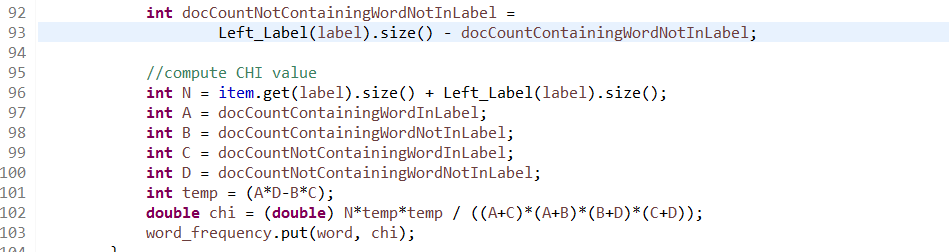
**public** Map<String, Double> processOneLabel(**int** label)中提供了计算卡方统计量的函数  


图3.2-13

**public** Map<String,Double> sortmap(Map<String, Double> word\_frequency)函数将Map的值按照chi进行排序。

**public** Map<String,Double> topN(Map<String,Double> sortedMap,Double n)函数依据chi从排序好的map中选取N个word作为该label的特征值。

产生的特征向量编号如下：

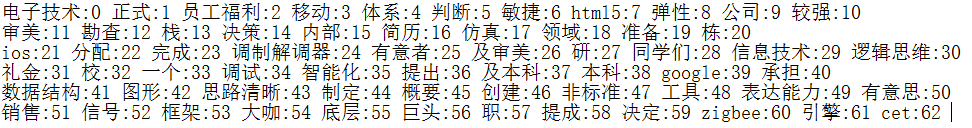


图3.2-14

DifferentSchoolAnalyzer中会调用component.DocumentTFIDFComputation.java文件的compute完成TF-IDF的计算

图3.2-15

（4）输出libsvm支持的量化的训练样本集文件：类别名称、特征向量中每个词元素分别到数字编号的映射转换，以及基于类别和特征向量来量化文本训练集，能够满足使用libsvm训练所需要的数据格式；

DifferentSchoolAnalyzer中会调用util.Handle\_LibSVM\_Input.java文件的loadData方法完成数据的格式处理

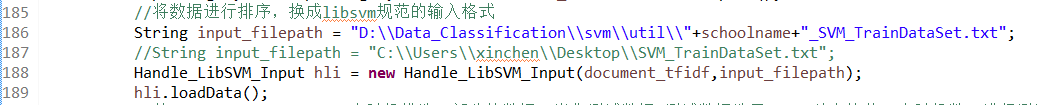


图3.2-16

各个特征向量对应的TF-IDF如下：

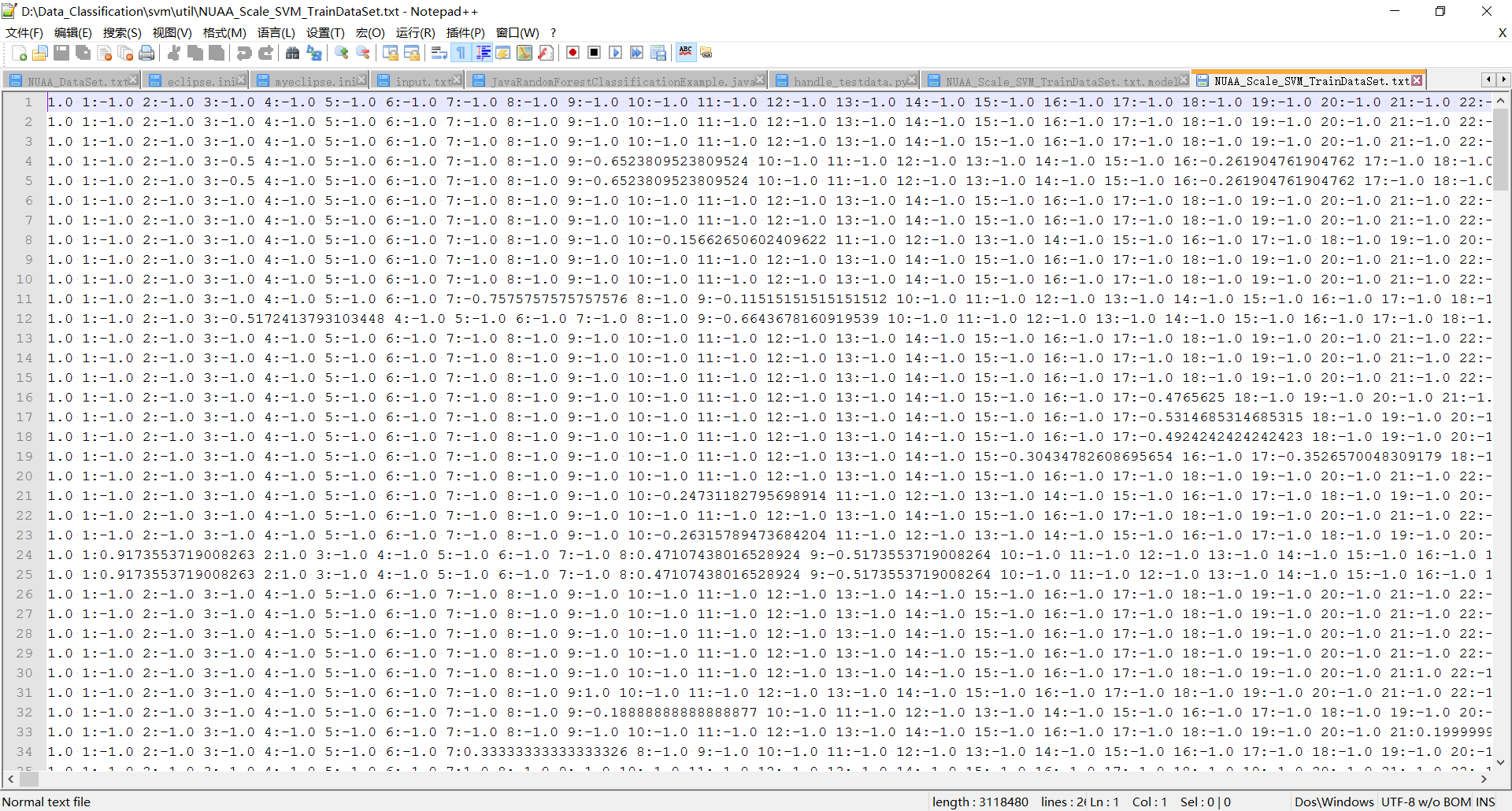


图3.2-17

（5）测试数据集预处理：同样包括分词（需要和训练过程中使用的分词器一致）、去停用词、建立词袋模型（倒排表），但是这时需要加载训练过程中生成的特征向量，用特征向量去排除多余的不在特征向量中的词（也称为降维）；

（6）输出libsvm支持的量化的测试样本集文件：格式和训练数据集的预处理阶段的输出相同。

（7）使用libsvm训练文本分类器：使用训练集预处理阶段输出的量化的数据集文件，最终输出分类模型文件

我们在这里首先对每个特征向量数据都对应着的TF-IDF值做归一化操作，之后输出模型文件XX.model文件，

具体执行代码图下：



图3.2-18

Model文件内容如下：

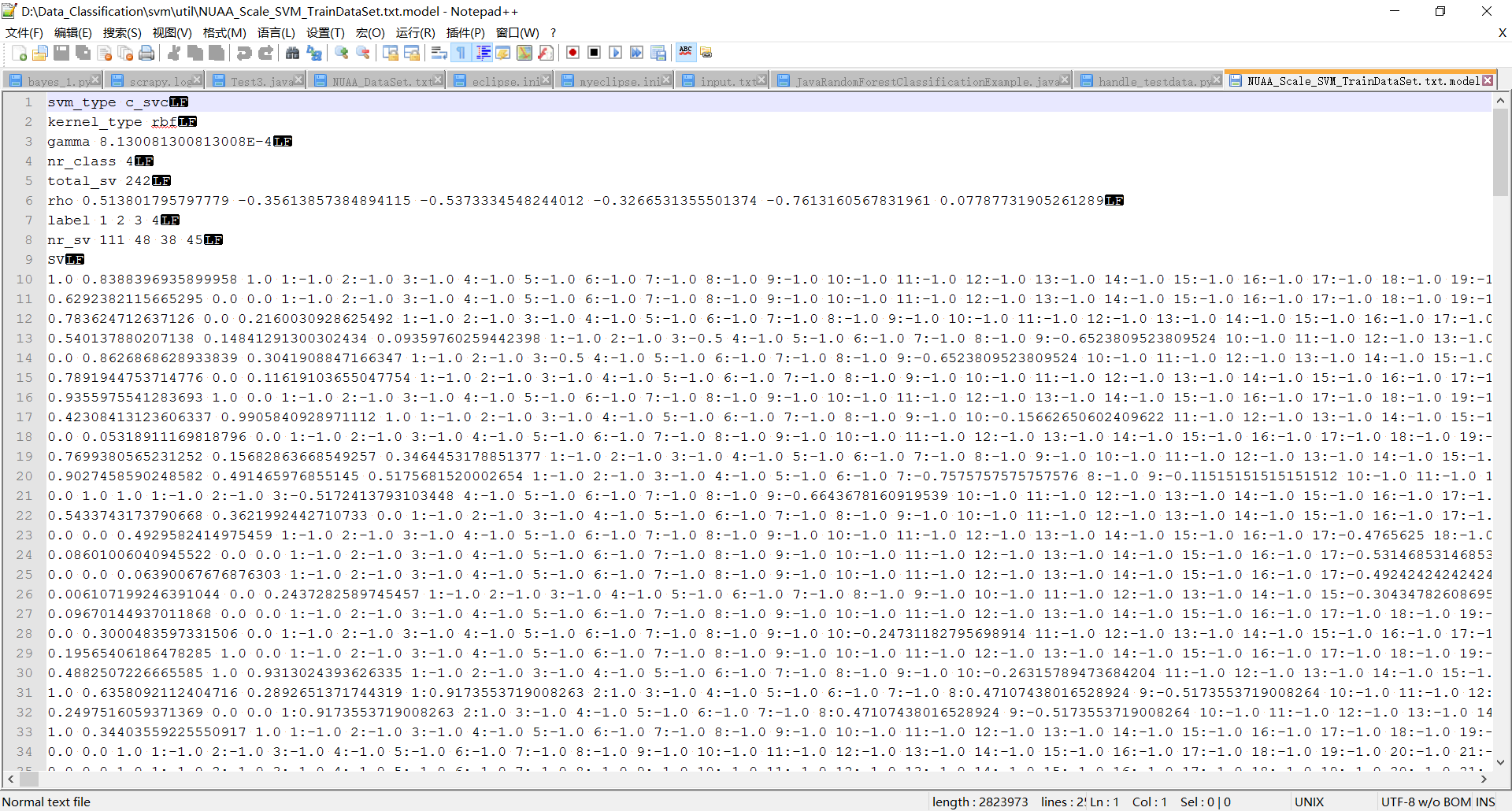


图3.2-19

（8）使用libsvm验证分类模型的精度：使用测试集预处理阶段输出的量化的数据集文件，和分类模型文件来验证分类的精度。

在主要的参数设置上，我们SVM类型采用C\_SVC、RBF核函数、多项式核中degree值为3、惩罚系数为1、损失函数中e为0.1,交叉验证的次数为10次。

我们从训练数据中随机抽取30条记录作为调优的测试数据，下图展示了预测数据的精度:

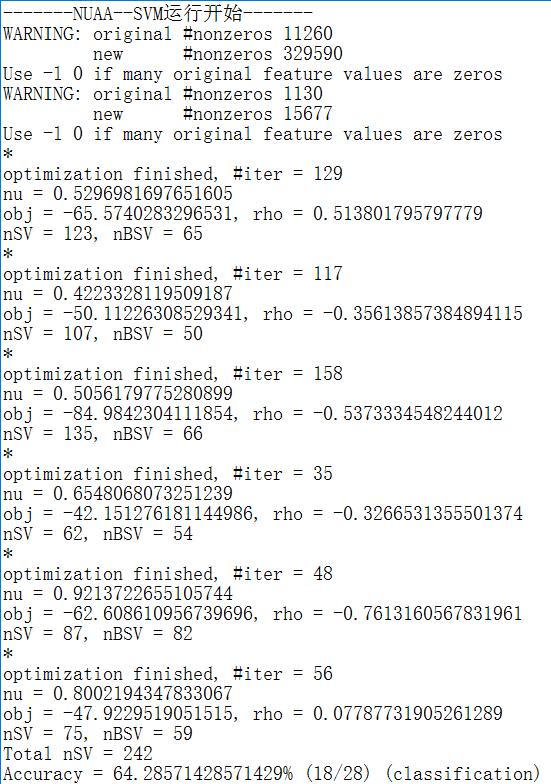
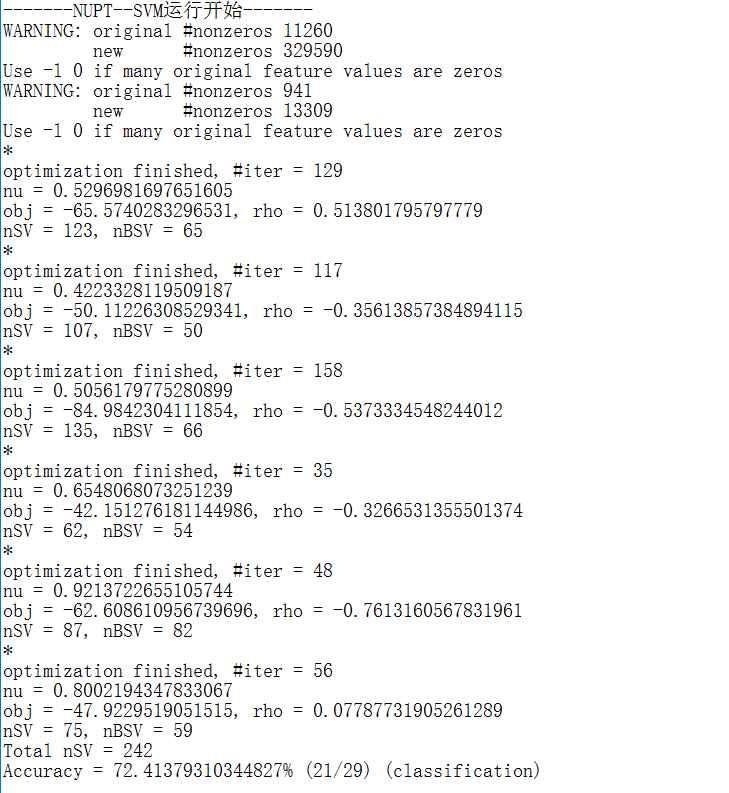
预测数据的精度：

从NUPT获取的数据分类后的精度为72.41%

从NUAA获取的数据分类后的精度为64.28%

从NUST获取的数据分类后的精度为74.04%

从SEU获取的数据分类后的精度为80.00%



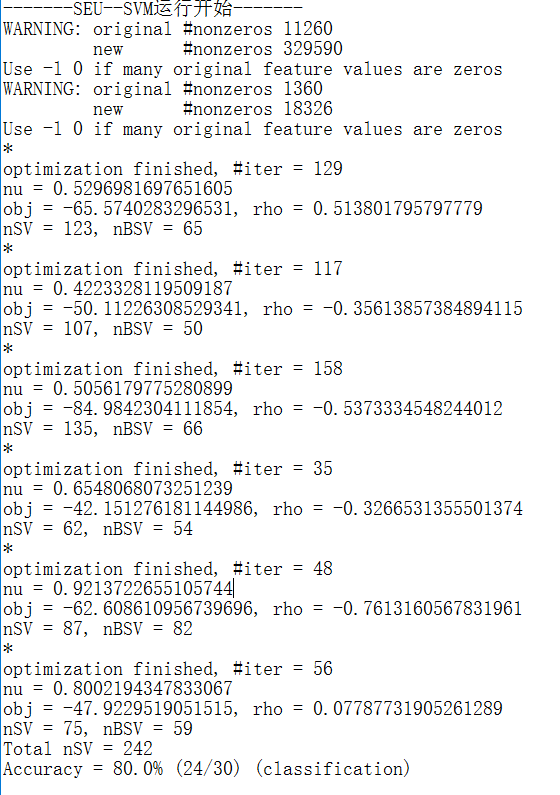
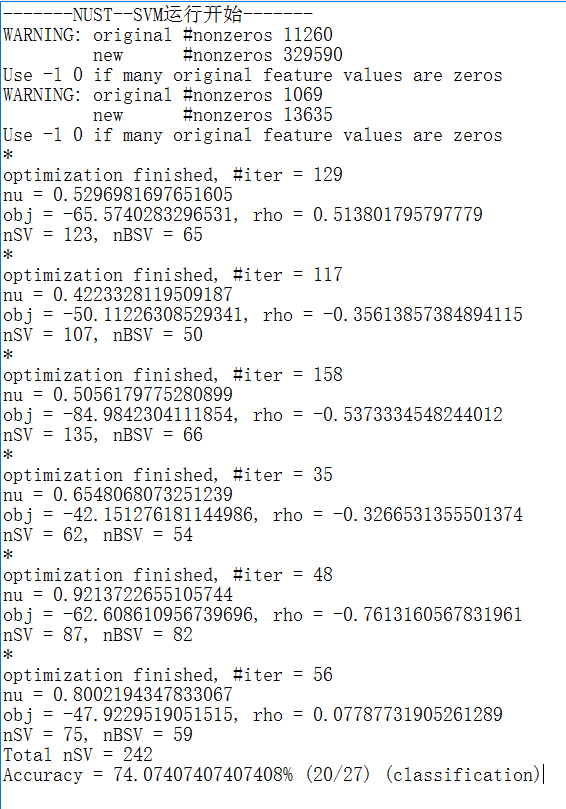


图3.2-20

1. 将最终的预测结果存进数据库中，代码如下：

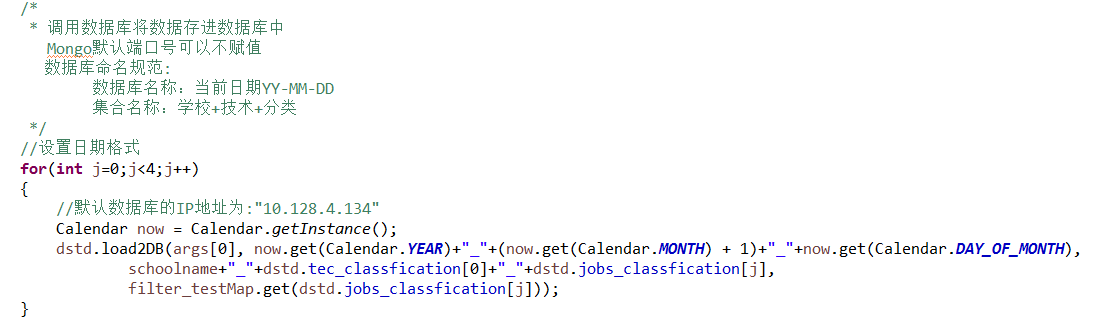


图3.2-21

### 3.2.3 随机森林算法

#### 3.2.3.1 分布式存储与运算

由于从各处收集爬取的招聘与简历信息种类繁多，数目庞大，这些数据的存储与处理就成为了一个问题。经过调研，发现可以使用开源的分布式架构Hadoop来进行海量的数据存储与处理，同时还能够兼顾到数据集的可靠性与容错性。

##### 3.2.3.1.1 Hadoop

Hadoop是一个由Apache基金会所开发的分布式系统基础架构能够对大量数据进行分布式处理。用户可以在不了解分布式底层细节的情况下，开发分布式程序,充分利用集群的威力进行高速运算和存储。

Hadoop 以一种可靠、高效、可伸缩的方式进行数据处理，它维护多个工作数据副本，确保能够针对失败的节点重新分布处理；它以并行的方式工作，通过并行处理加快处理速度。此外，Hadoop依赖于社区服务，因此它的成本比较低，任何人都可以使用。

Hadoop的核心是HDFS和MapReduce，Hadoop旗下有很多经典子项目，比如HBase、Hive等，这些都是基于HDFS和MapReduce发展出来的。要想了解Hadoop，就必须知道HDFS和MapReduce是什么。

##### 3.2.3.1.2 HDFS

HDFS的全称是Hadoop Distributed File System，分布式文件系统，它为海量的数据提供了存储。HDFS的设计特点是：

1、大数据文件，非常适合上T级别的大文件或者一堆大数据文件的存储。

2、文件分块存储，HDFS会将一个完整的大文件平均分块存储到不同计算器上，它的意义在于读取文件时可以同时从多个主机取不同区块的文件，多主机读取比单主机读取效率要高得多得都。

3、流式数据访问，一次写入多次读写，这种模式跟传统文件不同，它不支持动态改变文件内容，而是要求让文件一次写入就不做变化，要变化也只能在文件末添加内容。

4、廉价硬件，HDFS可以应用在普通PC机上，这种机制能够让给一些公司用几十台廉价的计算机就可以撑起一个大数据集群。

5、硬件故障，HDFS认为所有计算机都可能会出问题，为了防止某个主机失效读取不到该主机的块文件，它将同一个文件块副本分配到其它某几个主机上，如果其中一台主机失效，可以迅速找另一块副本取文件。

HDFS的主要结构如下图所示：

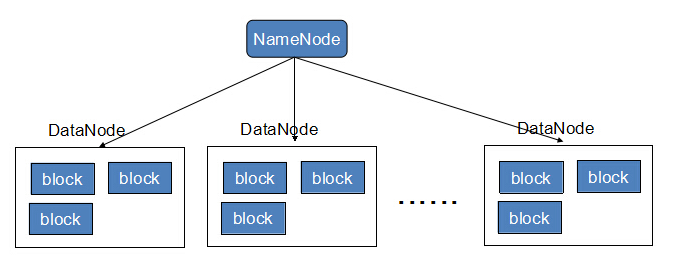


图3.2-22

Block：将一个文件进行分块，通常是64M。

NameNode：保存整个文件系统的目录信息、文件信息及分块信息，这由唯一一台主机专门保存，当然这台主机如果出错，NameNode就失效了。在Hadoop2.\*开始支持activity-standy模式——如果主NameNode失效，启动备用主机运行NameNode。

DataNode：分布在廉价的计算机上，用于存储Block块文件。

##### 3.2.3.1.3 MapReduce

MapReduce是Hadoop中的分布式运算框架，为海量的数据提供了计算。MapReduce是一套从海量源数据提取分析元素最后返回结果集的编程模型，将文件分布式存储到硬盘是第一步，而从海量数据中提取分析内容就是MapReduce做的任务了。MapReduce的基本原理是将大的数据分析分成小块逐个分析，然后再将提取出来的数据汇总分析，最终获得我们想要的内容。

##### 3.2.3.1.4 Spark

客观的来说，MapReduce是Hadoop中的运算部分，与Hadoop是相对独立的，而且由于MapReduce存在很多缺陷，使用起来比较困难，所以我们也可以使用更好的分布式运算框架来代替MapReduce进行数据处理。经过比较分析，我们最后确定使用Spark框架与Hadoop结合来实现快速的数据存储与运算。

Spark是UC Berkeley AMP lab所开源的类Hadoop MapReduce的通用并行框架，Spark拥有MapReduce所具有的优点，但不同于MapReduce的是Job中间输出结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的MapReduce算法；Spark启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。

Spark在机器学习方面有着无与伦比的优势，特别适合需要多次迭代计算的算法。同时Spark拥有非常出色的容错和调度机制，确保系统的稳定运行，Spark目前的发展理念是通过一个计算框架集合SQL、Machine Learning、Graph Computing、Streaming Computing等多种功能于一个项目中，具有非常好的易用性。尽管创建 Spark是为了支持分布式数据集上的迭代作业，但是实际上它是对Hadoop的补充，可以在Hadoop文件系统中并行运行。

#### 3.2.3.2 随机森林算法

在给求职者呈现职位信息的时候，系统需要实现一个职位推荐功能，从预先收集的简历信息中提取训练出与职位分类的关系，然后再根据求职者输入的简历讯息进行比较运算，最后在页面呈现推荐结果。为了达到这一目的，我们需要选出合适的数据处理算法。经过分析比较，选择随机森林算法作为实现这一功能的主要算法。

随机森林算法是机器学习、计算机视觉等领域内应用极为广泛的一个算法，它不仅可以用来做分类，也可用来做回归，即预测。随机森林机由多个决策树构成，相比于单个决策树算法，它分类、预测效果更好，不容易出现过度拟合的情况。

随机森林算法基于决策树，在介绍随机森林算法之前，先来介绍决策树的原理。决策树是数据挖掘与机器学习领域中一种非常重要的分类器，算法通过训练数据来构建一棵用于分类的树，从而对未知数据进行高效分类。在构建决策树的过程中，最重要的是如何找到最好的分割点。寻找最好的分割点是通过量化分割后类的纯度来确定的，目前有三种纯度计算方式，分别是 Gini 不纯度、熵（Entropy）及错误率，它们的公式定义如下：



公式中的 P(i) 表示记录中第 i 类记录数占总记录数的比例，上面的三个公式均是值越大表示越“不纯”，值越小表示越“纯”。实际中最常用的是 Gini 不纯度公式。如果一个分割点能够将整个记录准确地分为两类，那该分割点就可以认为是最好的，此时被分成的两类是相对来说是最“纯”的。

决策树的构建是一个递归的过程，理想情况下所有的记录都能被精确分类，即生成决策树叶节点都有确定的类型，但现实这种条件往往很难满足，这使得决策树在构建时可能很难停止。即使构建完成，也常常会使得最终的节点数过多，从而导致过度拟合。过度拟合的典型表现是决策树的节点过多，决策树对训练数据错误率很低，而对测试数据其错误率却非常高。在实际应用中需要设定停止条件，当达到停止条件时，直接停止决策树的构建。但这仍然不能完全解决过度拟合问题，常常需要对构建好的决策树进行枝叶裁剪。随机森林算法的出现能够较好地解决过度拟合问题。

随机森林是由多个决策树构成的森林，算法分类结果由这些决策树投票得到，当在基于某些属性对一个新的对象进行分类判别时，随机森林中的每一棵树都会给出自己的分类选择，并由此进行“投票”，森林整体的输出结果将会是票数最多的分类选项；而在回归问题中，随机森林的输出将会是所有决策树输出的平均值。决策树在生成的过程当中分别在行方向和列方向上添加随机过程，行方向上构建决策树时采用放回抽样得到训练数据，列方向上采用无放回随机抽样得到特征子集，并据此得到其最优切分点，这便是随机森林算法的基本原理。同单一的决策树分类不同的是，随机森林通过多个决策树投票结果进行分类，算法不容易出现过度拟合问题。

随机森林算法能解决分类与回归两种类型的问题，并在这两个方面都有相当好的估计表现；在对缺失数据进行估计时，随机森林是一个十分有效的方法，就算存在大量的数据缺失，随机森林也能较好地保持精确性；当存在分类不平衡的情况时，随机森林也能够提供平衡数据集误差的有效方法。

MLlib 是Spark对常用的机器学习算法的实现库，同时包括相关的测试和数据生成器。MLlib 目前支持四种常见的机器学习问题：二元分类，回归，聚类以及协同过滤，同时也包括一个底层的梯度下降优化基础算法。使用者能够很方便的使用这个函数库来实现各种机器学习算法，包括随机森林算法。下一小节就对系统中随机森林算法的实现进行一个简要的说明。

#### 2.1.3.3 算法实现

在生成运算模型前，先对spark运算系统进行一些设置，使其适用于java环境。



图3.2-23

将简历数据转换为标准格式以后，从文件中读取训练用数据。



图3.2-24

将读取的数据分为两部分，70%用于训练，获取模型，30%数据用于最终模型正确性测试。



图3.2-25

设置好运算模型所需要的参数。

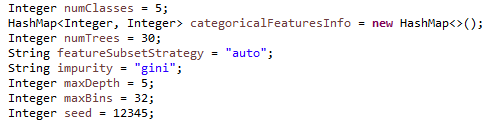


图3.2-26

模型中各种参数的调整与修改十分重要，通过查阅文献，在此给出各种参数的说明：

numClasses : 分类的个数。

categoricalFeaturesInfo : 特征指数与特征数量对应的map类型，不需要明确指定categoricalFeaturesInfo算法仍将运行并得到合理的结果，然而若能正确指定分类功能那么运算的性能将会更好。

numTrees : 随机森林中决策树的个数。增加树的数目将减少预测的方差，提高模型的测试精度。但是训练时间也会在树的数量上呈线性增加。

featuresubsetstrategy：候选节点在每个树节点分裂时所使用的特征个数。减小这个值可以加速训练，但是如果太低则会影响运算性能。“auto”表示由算法自行确定该值。

Impurity : 决定在候选分割之间选择的纯度测量方法。

max\_depth : 每棵决策树的最大深度。增加树的深度可以使得模型具有更强大的表现力。然而，深度大的树需要更长的时间来训练，也更容易出现过度拟合。

maxBins : 将连续特征离散化是使用的bin的数量。增加maxbins允许算法考虑更多的分割候选人，做出细粒度分裂决策。但是同时它也会增加计算和通信的负担。

seed : 指定随机数种子。

在设定好参数以后就可以调用方法计算模型。



图3.2-27

接下来是通过事先预留的测试数据计算错误率并打印。

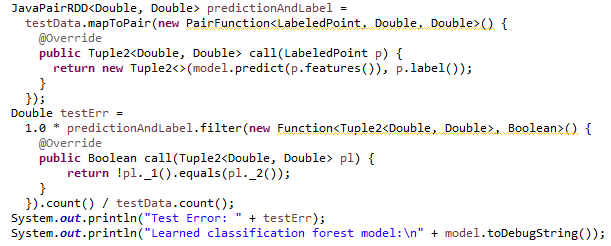


图3.2-28

最后可以通过运算出来的模型对输入数据进行分类。



图3.2-29

## 3.3 用户服务平台设计及代码介绍

### 3.3.1 Web前端

#### 3.3.1.1 AdminLTE框架的简介

AdminLTE 框架是受欢迎的开源的管理仪表盘和控制面板的WebApp模板。它是基于Bootstrap 3的CSS框架，反应灵敏的HTML模板。利用所有Bootstrap的组件对大部分使用插件进行设计和调整风格，创建出可以用作后端应用程序的用户界面一致性设计。AdminLTE是基于模块化设计，很容易在其之上定制和重制。目前对于AdminLTE框架支持的浏览器包括IE9+、FireFox、Chrome、Safari、Opera。下面就是一个简单的AdminLTE框架的主页面：

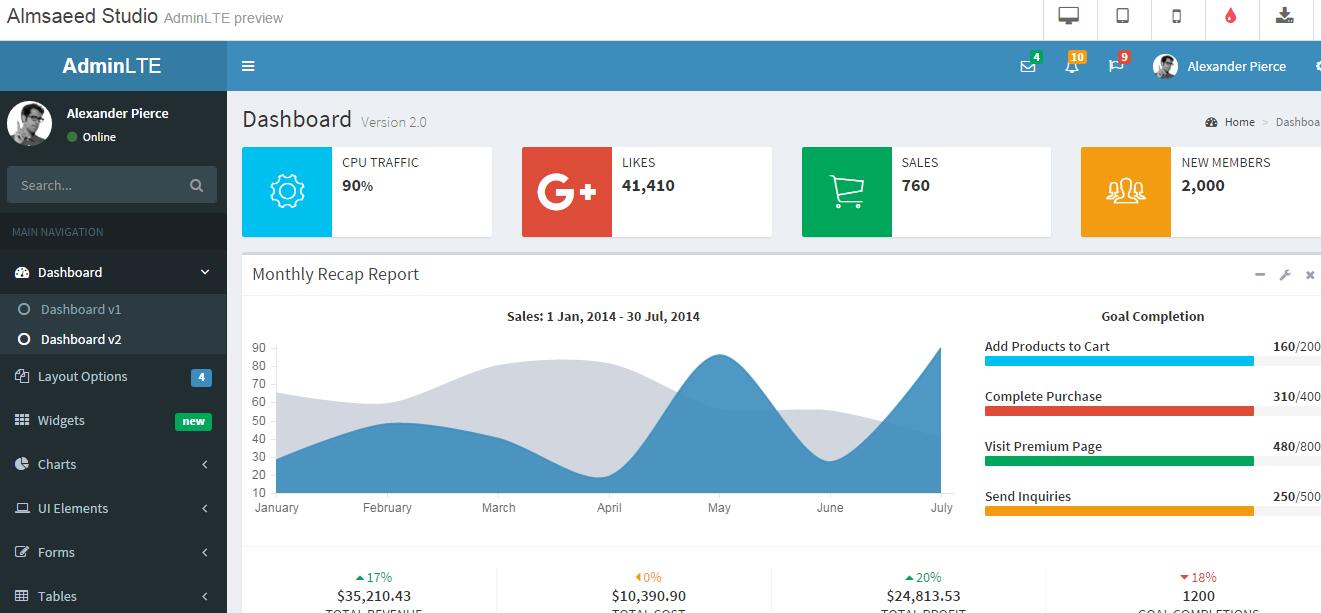


图3.3-1 AdminLTE框架的主页面

#### 3.3.1.2 Bootstrap框架和jQuery库介绍

AdminLTE框架主要依赖于两个框架，Bootstrap框架和jQuery库。下面将对这两个框架作简单的介绍：

Bootstrap框架，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言LESS写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。Bootstrap是基于HTML5和CSS3开发的，它在JQuery的基础上进行了更为个性化和人性化的完善，形成一套自己独有的网站风格，并兼容大部分jQuery插件。Bootstrap中包含了丰富的Web组件，根据这些组件，可以快速的搭建一个漂亮、功能完备的网站。其中包括以下组件：下拉菜单、按钮组、按钮下拉菜单、导航、导航条、导航路径、分页、排版、缩略图、警告对话框、进度条、媒体对象等。Bootstrap自带了13个Jquery插件，这些插件为Bootstrap中的组件赋予了“生命”。其中包括：模式对话框、标签页、滚动条、弹出框等。Bootstrap作为前端框架，具有很多优秀的特性：（1）Bootstrap支持跨浏览器，支持跨设备，可以实现对手机和平板的完美支持。（2）从Bootstrap 2开始，提供完整的响应式特性。所有的组件都能根据分辨率和设备灵活缩放，从而提供一致性的用户体验。（3）样式化的文档，与其他[前端开发](http://www.2cto.com/kf/qianduan/)工具包不同，Bootstrap优先设计了一个样式化的使用指南，不仅用来介绍特性，更用以展示最佳实践、应用以及代码示例。（4）不断完善的代码库。尽管经过gzip压缩后，Bootstrap只有10KB大小，但是它却仍是最完备的前端工具包之一，提供了几十个全功能的随时可用的组件。

jQuery是继[prototype](http://baike.baidu.com/view/1217697.htm)之后又一个优秀的[Javascript](http://baike.baidu.com/view/16168.htm)库。它是轻量级的js库 ，它兼容[CSS3](http://baike.baidu.com/view/1713027.htm)，还兼容各种浏览器。jQuery使用户能更方便地处理[HTML](http://baike.baidu.com/view/692.htm)（[标准通用标记语言](http://baike.baidu.com/view/5286041.htm)下的一个应用）、events、实现动画效果，并且方便地为网站提供[AJAX](http://baike.baidu.com/subview/1641/5762264.htm)交互。Jquery还有一个比较大的优势是，它的文档说明很全，而且各种应用也说得很详细，同时还有许多成熟的[插件](http://baike.baidu.com/view/18979.htm)可供选择。jQuery能够使用户的html页面保持代码和html内容分离，也就是说，不用再在html里面插入一堆js来调用命令了，只需要定义id即可。jQuery是一个兼容多浏览器的[javascript](http://baike.baidu.com/view/16168.htm)库，核心理念是write less,do more(写得更少,做得更多)。jQuery在2006年1月由美国人[John Resig](http://baike.baidu.com/view/3141971.htm)在纽约的[barcamp](http://baike.baidu.com/view/1135515.htm)发布，吸引了来自世界各地的众多JavaScript高手加入，由Dave Methvin率领团队进行开发。如今，jQuery已经成为最流行的javascript库，在世界前10000个访问最多的网站中，有超过55%在使用jQuery。jQuery是免费、开源的，使用[MIT](http://baike.baidu.com/subview/74918/8382747.htm)许可协议。jQuery的语法设计可以使开发更加便捷，例如操作[文档](http://baike.baidu.com/view/55621.htm)对象、选择[DOM](http://baike.baidu.com/view/14806.htm)元素、制作动画效果、事件处理、使用[Ajax](http://baike.baidu.com/view/1641.htm)以及其他功能。除此以外，jQuery提供API让开发者编写插件。其模块化的使用方式使开发者可以很轻松的开发出功能强大的静态或动态网页。jQuery作为新一代的[Javascript](http://baike.baidu.com/view/16168.htm)库，具有以下特性：（1）轻量级jQuery非常轻巧，采用Dean Edwards编写的Packer压缩后，大小不到30KB。（2）出色的DOM操作的封装。jQuery封装了大量常用的DOM操作，使开发者在编写DOM操作相关程序的时候能够得心应手。（3）出色的浏览器兼容性。jQuery能够在IE6.0+,FF 2+,Safari2.+和Opera9.0+下正常运行。（4）丰富的插件支持。 jQuery的易扩展性，吸引了来自全球开发者来编写jQuery的扩展插件。目前已经有超过几百种官方插件支持，而且还不断有新插件面试。（5）完善的文档。jQuery的文档非常丰富，现阶段多为英文文档，中文文档相对较少。很多热爱jQuery的团队都在努力完善jQuery中文文档，例如jQuery的中文API。

#### 3.3.1.3 AdminLTE框架的特性及使用

在AdminLTE框架中，使用了Bootstrap框架和jQuery库。下面将对AdminLTE框架做一个简单的介绍。（1）依赖性。AdminLTE框架依赖于Bootstrap框架和jQuery库，在使用AdminLTE框架的时候，已经包含了Bootstrap框架和jQuery的源文件，可以实现对Bootstrap框架和jQuery库的完美支持。（2）布局。AdminLTE的布局主要包括四个层面，wrapper层是页面整体布局，main-header，包含的是logo和导航栏布局，sidebar-wrapper包含用户面板和侧边栏菜单，content-wrapper包含的是页面的头部和内容部分。（3）皮肤。AdminLTE设计了丰富多彩的皮肤供使用者使用，主要包含skin-blue、skin-yellow、skin-green等12种颜色的皮肤以供渲染。（4）组件。在AdminLTE中，继承了大量可用的组件供使用者选择。Main Header部分包含的是logo和导航菜单。Sidebar部分包含的是侧边栏部分。Info Box包含了一些常用的联系信息的片段图标。Direct chat可以创建动态的聊天窗口。

在我们的项目中，我们选择AdminLTE模板作为我们的前端应用程序模板。针对我们的项目，我们在AdminLTE模版的界面上做了一定的修改。主题修改为智慧就业信息系统，我们的后端应用程序展示界面主要包括两个部分。一个就是HOME主页，HOME主页部分主要的包含的是我们的项目名称、项目组成员。第二个部分就是Backstage Show页面部分，这部分主要是用来展示我们的项目运行的结果，包含两个与后台进行交互的操作，分别是Data collection，这部分主要所做的工作就是后代对数据库中的数据进行收集分类的操作。

另外一个很重要的操作就是Data Presentation，就是后台对数据库中的数据处理完毕后，会通过这一步操作返回到前台来。对于后台接收到的就业信息数据，我们进行了分类，分别是DEV（研发）、TEST(测试)、IT（技术支持）、OTHER（未分类），对于接收来的大批数据，我们采用iframe技术进行展示在页面窗口中。整个前端页面的性能依赖于后台的执行时间，对于后端发送来的数据，可以进行有效的分类并且展示在页面窗口中，可以对就业信息数据进行一定的过滤和筛选，筛选出用户关心的信息。

### 3.3.2 Web后台

#### 3.3.2.1 web后端简介

现在，数据存储到数据库了，但是如何给求职者展示所要的求职信息是一个技术难题。为了给求职者提供一个基于web的求职信息平台，我们考虑使用现在比较流行的SSH(即：struts2，hibernate，spring)框架以及MongoDB数据库来构建 web后台环境。

##### 3.3.2.1.1 为什么要选择SSH框架以及MongDB数据库

目前，针对Web后台的开发主要有Struts2的MVC框架以及SpringMVC的

MVC开发框架。我们之所以选用基于Struts2的MVC框架，主要是它有以下几点优势：

(1) 实现了MVC模式，层次结构清晰，使程序员只需关注业务逻辑的实现;

(2) 丰富的标签库，大大提高了开发的效率;

(3) Struts2提供丰富的拦截器实现;

(4) 通过配置文件，就可以掌握整个系统各个部分之间的关系;

(5) 异常处理机制，只需在配置文件中配置异常的映射，即可对异常做相应的处理;

数据库这里我们采用的是一个基于分布式文件存储的数据库，即MongDB数据库，它是一种非关系型数据库(NoSQL)，和我们传统的关系型数据库(比如MySQL)相比，有如下几点优势：

弱一致性（最终一致），更能保证用户的访问速度。

基于内存存储方式，能够更快速的获取数据。

内置GridFS，支持大容量的存储。GridFS是一个出色的分布式文件系统，可以支持海量的数据存储。

内置Sharding,提供基于Range的Auto Sharding机制：一个collection可按照记录的范围，分成若干个段，切分到不同的Shard上。

第三方支持丰富。(这是与其他的NoSQL相比，MongoDB具有的优势)。

##### 3.3.2.1.2 基于MVC的框架—Struts2

Struts2和SpringMVC是目前比较流行的MVC Web后台框架，两者的基本原理都一样。并且都规范，封装了Servlet的开发，使得在开发web后台这块效率得到大大提升。M的意思即Model(模型)，它相当于一个java实体类，主要实现对事物的抽象及封装。在Hibernate里，它与数据库表形成了对象关系映射。V的意思是View，也就是视图的意思。它主要提供用户界面的展示，这里我们一般使用jsp页面提供对用户的展示。C的意思是Controller，也就是控制器的意思。在这里，我们主要是指action，它主要完成对前台传送过来的数据进行接收以及业务的处理。另外，Struts2还提供了拦截器以及校验器的功能，这样我们可以实现权限的验证，以及输入校验。当然，Struts还提供了文件上传与下载的功能，强大的标签库以及程序的国际化功能。

##### 3.3.2.1.3 基于持久层的框架—Hibernate

Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了轻量级的封装，使得Java程序员可以随心所欲的使用对象编程的思维来操纵数据库，并且它提供了对MySQL，Oracle，DB2，SQL server数据库的基本操作。在这里，我们主要使用它的对象关系映射，Session会话以及Session缓存，SessionFactory的二级缓存以及Hibernate的查询缓存。当然在数据源那里，我们使用了C3P0的数据库连接池。利用好Hibernate提供的这些性能，我们可以加快对数据库的CRUD操作。除此之外，Hibernate还提供了对数据库的事务操作以及锁操作，主要分为JDBC的数据库事务以及JTA的数据库事务，确保我们操作的原子性。通过ORM，我们可以使用HQL语句来实现面向对象的数据库操作。

##### 3.3.2.1.4 基于IOC以及AOP的框架—Spring

Spring也是一个轻量级Java开发框架，是轻量级的IoC和AOP的容器框架。主要是针对javaBean的生命周期进行管理的轻量级容器，它可以单独使用，也可以和Struts2以及Hibernate框架等组合使用。Spring的主要功能是提供了对象之间的解耦，简化开发，以及Aop编程，声明式事务的支持。其中IOC的意思是控制反转或依赖注入，也就是说创建对象的任务并不是由调用者去完成，而是由外部的Spring容器来完成的，也可以认为调用者要依赖Spring Ioc容器来获得对象实例，所以我们也称Ioc为依赖注入。另外一个重要特点是动态代理，它可以使得我们面向切面编程，从而实现代码的织入性强。通过AOP，我们可以实现事务的管理，通知以及日志的管理。另外，Spring提供了一些更强大的功能，比如事务的管理，它封装了所有对事务处理的功能。以及基于Restful风格的代码编写和编解码时乱码的处理。

#### 3.3.2.2 web后台的详细实现

在这一节，我们主要讲解我们利用SSH框架的所实现的功能以及具体实现过程。

##### 3.3.2.2.1 SSH框架的搭建

在构建SSH框架，我们所使用的开发工具是Eclipse EE，它是开源的，主要用来开发J2EE项目。SSH框架我们使用的是Struts2.3，Hibernate3.0以及Spring3.0.6。

首先，我们得导入三大框架的jar包，因为包与包之间可能存在冲突，所以我们得首先调查好jar之间的关系，导入开发项目必须要用到的jar包。之后，我们要新建web.xml配置文件，它是web项目的全局配置文件，在里面配好Struts2的 DTD约束以及Struts2的主控制器，这样就可以使得我们在浏览器器端输入地址可以进入Struts2框架。

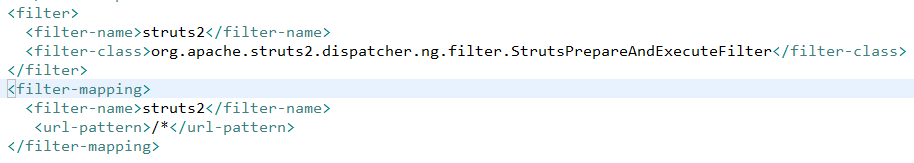


图3.3-2 Struts2的主控配置

配置好Struts2，我们就可以去配置数据源DataSource，因为我们使用的Hibernate已经封装了数据库的一系列操作。所以我们可以在Hibernate里配置好数据源。同时，为了使用好Spring容器，我们需要在web.xml里面配置好监听器，使得web容器在初始化的过程中就可以加载由Spring容器管理实体类。并且，我们Struts2的action控制器类的生成应该转交到Spring容器实现。

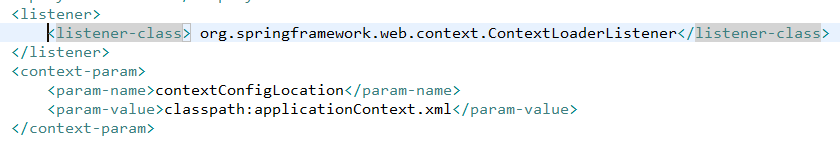


图3.3-3 Spring的配置及加载

##### 3.3.2.2.2 Scrapy数据的采集（Data Collection）

在web前端做好的前端页面，我们点击Data Collection按钮，将会调用Scrapy爬虫框架进行各大门户网站招聘信息的采集过程。



图3.3-4 浏览器前端页面（Data Collection）

它将发生如下一系列事件：

首先调用start.bat批处理文件，执行scrapy crawl进行各大门户网站实时招聘信息的采集。

同时，通过Jquery的ajax方法异步调用后台的scrapy控制器，并返回OK给前端的浏览器页面。

在采集即将完成时，会在浏览器上显示OK，并过三秒后消失，通知操作者数据采集完成。

另外，我们将通过python脚本，将采集到的招聘信息按照统计的格式存储到MongoDB数据库里。

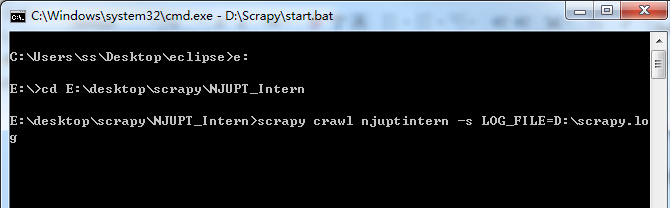


图3.3-5 调用Scrapy框架进行数据的采集

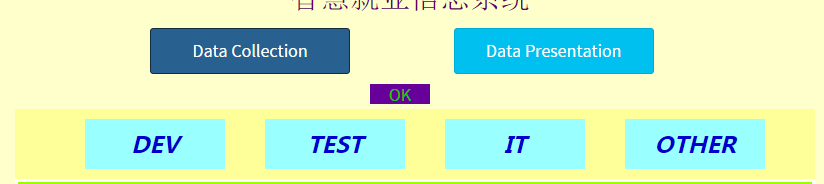


图3.3-6采集完成后，页面显示“OK”

##### 3.3.2.2.3 数据的处理（DataPresentation）

在web前端做好的前端页面，我们点击Data Presentation按钮，将会机器学习算法对数据的梳理，并将处理过的数据存放到数据库里。通过点击Data Presentation按钮，将发生如下几件事情：

（1）调用数据处理模块以及机器学习算法模块对数据进行处理。首先它会将你的请求通过异步调用的方式传递给后台的resumeaction去处理。

（2）resumeaction接到请求后，将调用execute()方法去处理浏览器的请求。并返回Map格式的数据，并通过Struts2的格式转换，变为Json格式进行返回到前端的页面。

（3）数据分析，处理成功后，将通过页面展示“OK”的方式提醒操作者，数据的分析以及处理结束。

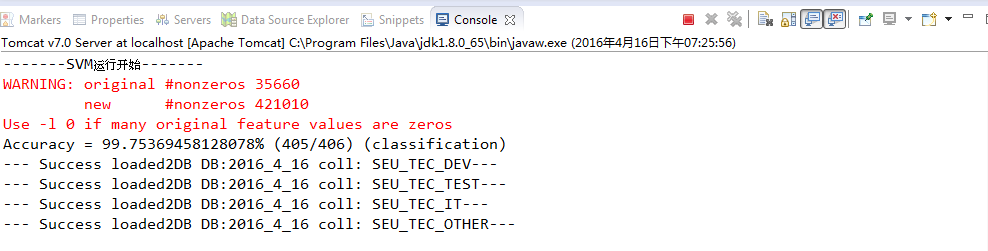


图3.3-7机器学习算法处理数据

##### 3.3.2.2.4 求职岗位数据的展示

针对不同的求职岗位，我们将展示出针对不同求职岗位的实时的招聘信息，提供给求职者需要的招聘信息。这边我们主要是基于angular Js里的ngtable技术，进行信息的展示。当然，在后台处理那块，我们还是使用的SSH框架进行处理，并且操控MongoDB数据库，进行不同岗位的招聘信息的获取。在这里，我们主要将招聘信息分为四大类：研发，测试，技术支持，其他。通过前端页面的点击，将会调用后台不同的action去处理，获取MongoDB不同的数据表。并展示在angular Js的Ngtable表格中。

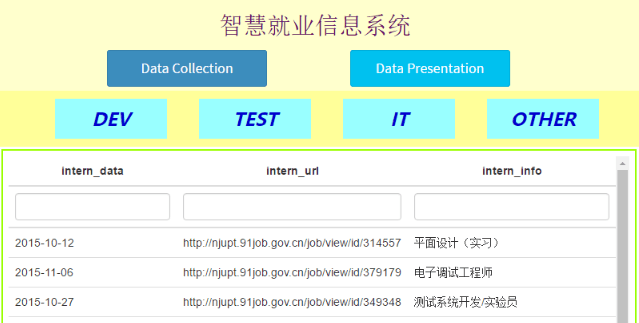
比如，当我们点击DEV按钮时，将会通过http请求进入到struts2的主控制器里面，并且通过Struts2的Struts.xml配置文件将匹配到对应的action，通过action的处理之后，将我们所需要的岗位信息在不同的angular Js的ngtable上显示出

图3.3-8 TEST测试岗位信息的web页面



图3.3-9 DEV开发岗位信息的Web页面

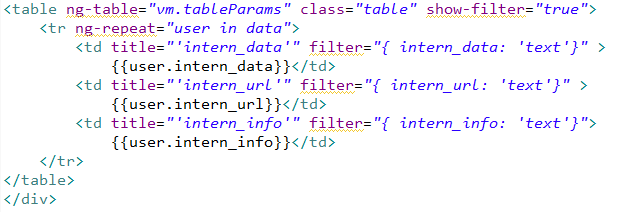


图3.3.10 Anjular Js的部分ngtable代码用于展示研发类求职信息

##### 3.3.2.2.5 求职简历信息的后台处理

输入http://localhost:8080/resume/对应的在后端GetResumeDataAction将发生如下一系列事件：

1. 收集页面信息，处理数据格式
2. 调用随机森林算法接口，算法运行并输出预测结果
3. 根据预测的结果从MongoDB数据库中获取信息，并至前端ngtable展示。

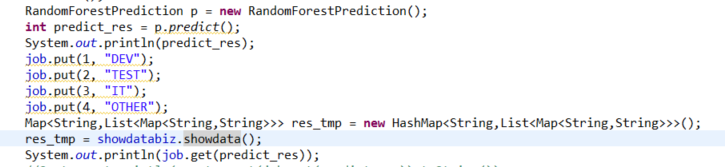


图3.3.11 调用随机森林算法获取预测结果部分代码展示

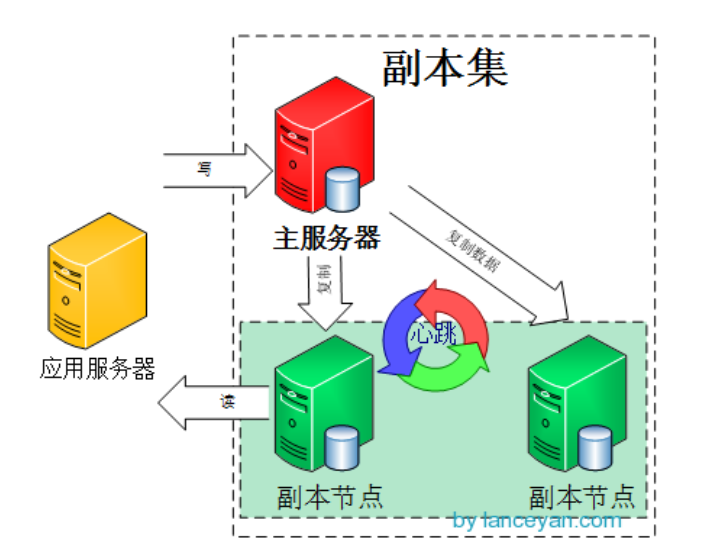
## 3.4 数据存储单元设计与实现方案

### 3.4.1 MongoDB数据库简介

数据库采用的是基于分布式文件存储的数据库，即MongDB数据库，它是一种非关系型数据库(NoSQL)，和我们传统的关系型数据库(比如MySQL)相比，有如下几点优势：

1. 弱一致性（最终一致），更能保证用户的访问速度。
2. 基于内存存储方式，能够更快速的获取数据。
3. 内置GridFS，支持大容量的存储。GridFS是一个出色的分布式文件系统，可以支持海量的数据存储。
4. 内置Sharding,提供基于Range的Auto Sharding机制：一个collection可按照记录的范围，分成若干个段，切分到不同的Shard上。
5. 第三方支持丰富。(这是与其他的NoSQL相比，MongoDB具有的优势)。
6. 在本次项目中为了解决大数据量、高扩展性、高性能、灵活数据模型、高可用性，我们搭建了高可用的MongoDB数据库集群，设计思路如下：

### 3.4.2 数据库集群设计图

图3.4-1 数据库集群设计图

### 3.4.3 副本集的设计

#### 3.4.3.1 副本集原理

主服务器负责整个副本集的读写，副本集定期同步数据备份，一但主节点挂掉，就会利用心跳机制在集群内发起主节点的选举机制，自动选举一位新的主服务器。

#### 3.4.3.2 项目设计思路

（1）本次实验环境中主服务器选用一台性能卓越的机架式服务器。IP为10.128.6.245，id=1。

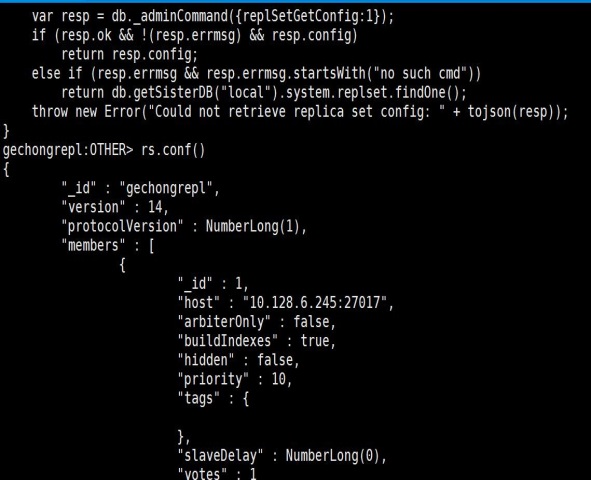
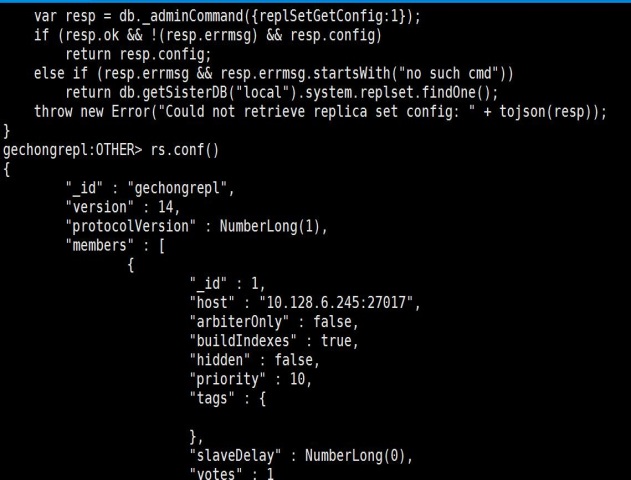
从服务器为两台PC，IP分别为：10.128.7.97和10.128.4.10，id值分为2和3。

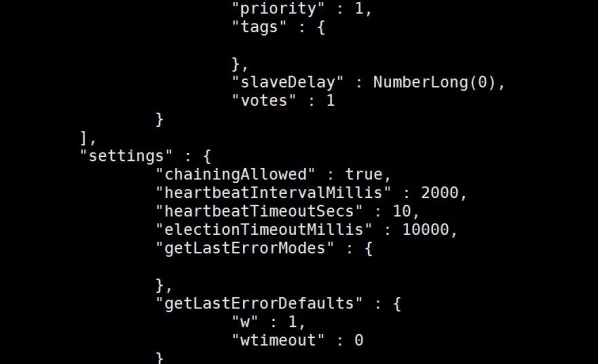
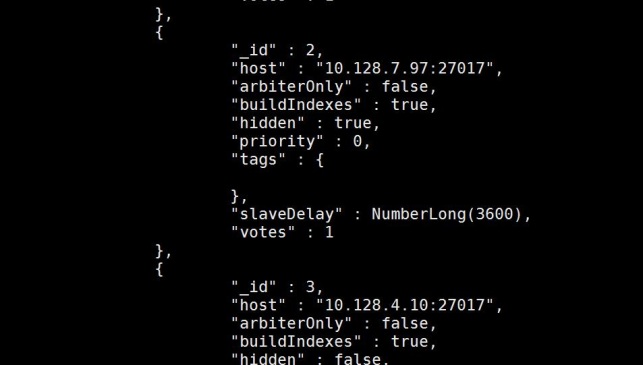
因为在主服务器上进行写操作，为了防止数据因为误删等人工原因造成数据丢失，配置id为3的从服务器(slaveDelay:0)实时同步于主服务器，id为2的从服务器(slaveDelay:3600)每隔3600s同步于主服务器。

为了保证数据的安全性，设计id为3的从服务器(hidden:true),从而不能被外界程序访问，并且设置(prority：0)表示当主服务器宕机后，该从服务器将不参与新的主服务器的选举。

#### 3.4.3.3 服务器配置

下图为数据库集群的参数设置：



图3.4-2 DEV开发岗位信息的Web页面

### 3.4.4 读写分离设计

#### 3.4.4.1 设计思路

在主服务器(id=1)上进行写操作，配置了MongoExpress WEB服务器于从节点（id=3）上读取与展示数据。

#### 3.4.4.2 效果展示

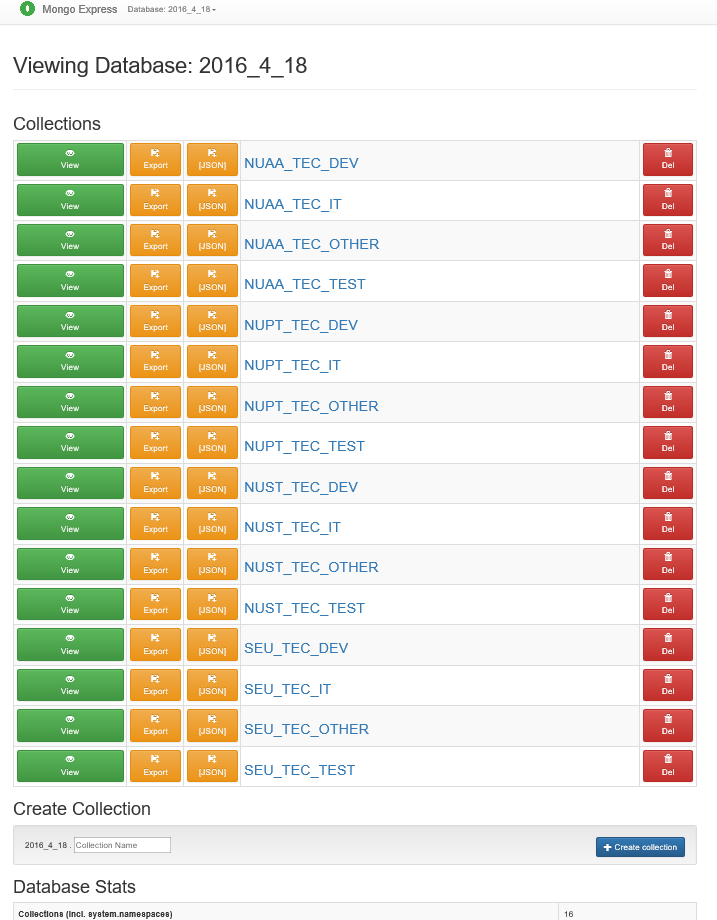


图3.4-3 2016\_4\_18的数据库中的集合展示

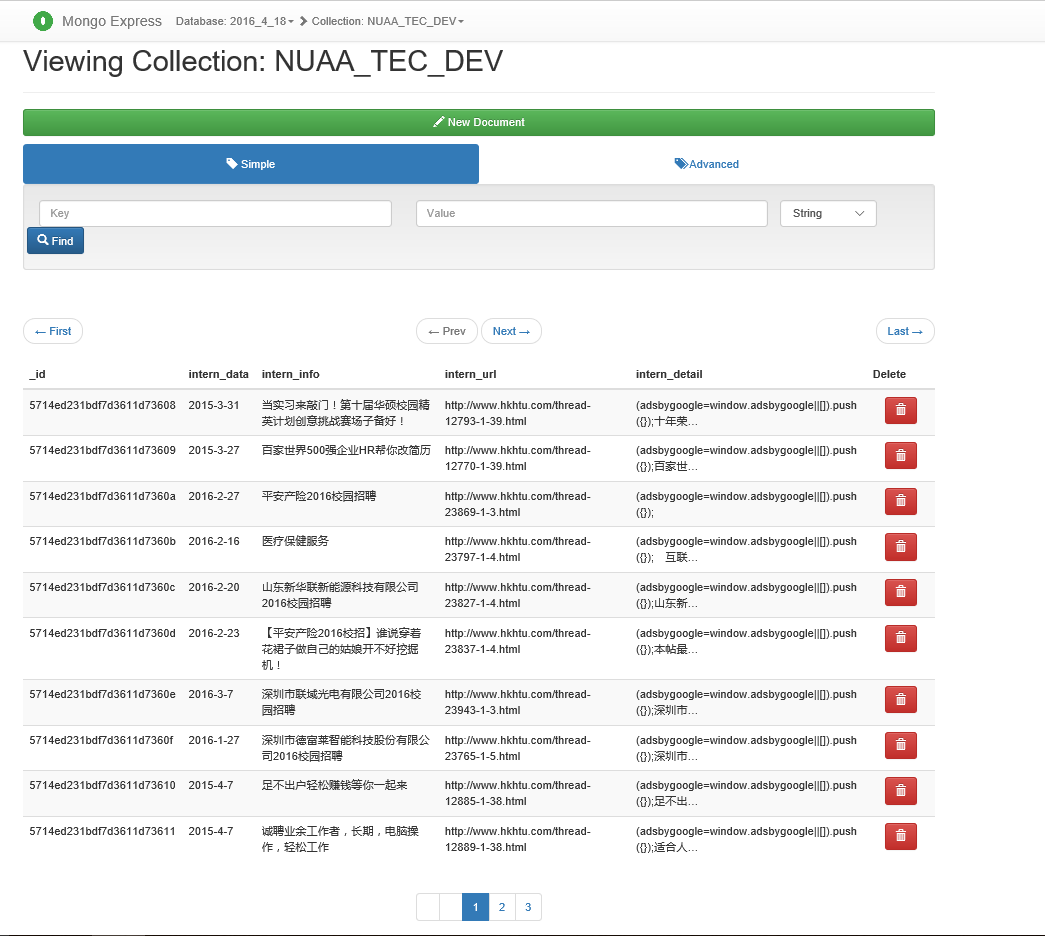


图3.4-4 NUAA\_TEC\_DEV集合中的文档展示

# 四、系统功能展示

## 4.1 软件概述及功能

本款软件主要服务对象是想求职，应聘IT行业的大学生，研究生或者社会上的求职者，主要提供的服务是为各类求职者提供他们所需要的求职信息，包括研发（开发）类，测试类，技术支持类以及其他岗位的招聘岗位信息，并且可以为你填写的简历匹配合适的岗位，并提供实时的招聘信息展示。

## 4.2 软件安装说明

开发环境在Eclipse环境下开发，JAVA部分代码量8000行，Python部分代码量250行。

运行环境：JRE/JDK、python2.7.3、Scrapy(http://scrapy.org/)、MongoDB数据库(https://www.mongodb.org/)

运行前需要将软件包中的Scrapy文件夹以及Data\_Classification文件夹拷贝至D盘。并启动软件包中的Tomcat。

## 4.3 软件使用说明

本款软件是基于浏览器/服务器的软件，所以使用工具只要一个IE/Chrome的浏览器。

首先，我们应该在浏览器里输入http://服务器的IP址:8080/resume/index1.html，然后回车（Enter），出来Web软件的主页面，如图4.3-1。



图4.3-1 软件的主页面

### 4.3.1 数据的采集过程

首先，在Web软件的主页面上，点击左侧的导航栏（Backstage Show），你会看到如下页面:(如图4.3-2)



图4.3-2 功能主页面

此时，你应该先点击Data Collection，这将调用服务器端的Start.bat批处理文件，在客户端会调用CMD窗口，而服务器端会调用Scrapy框架进行各大门户网站招聘信息数据的采集。在采集完成即将时，Web页面上会出现“OK”，并停留3秒后，消失。如图4.3-3和4.3-4所示。

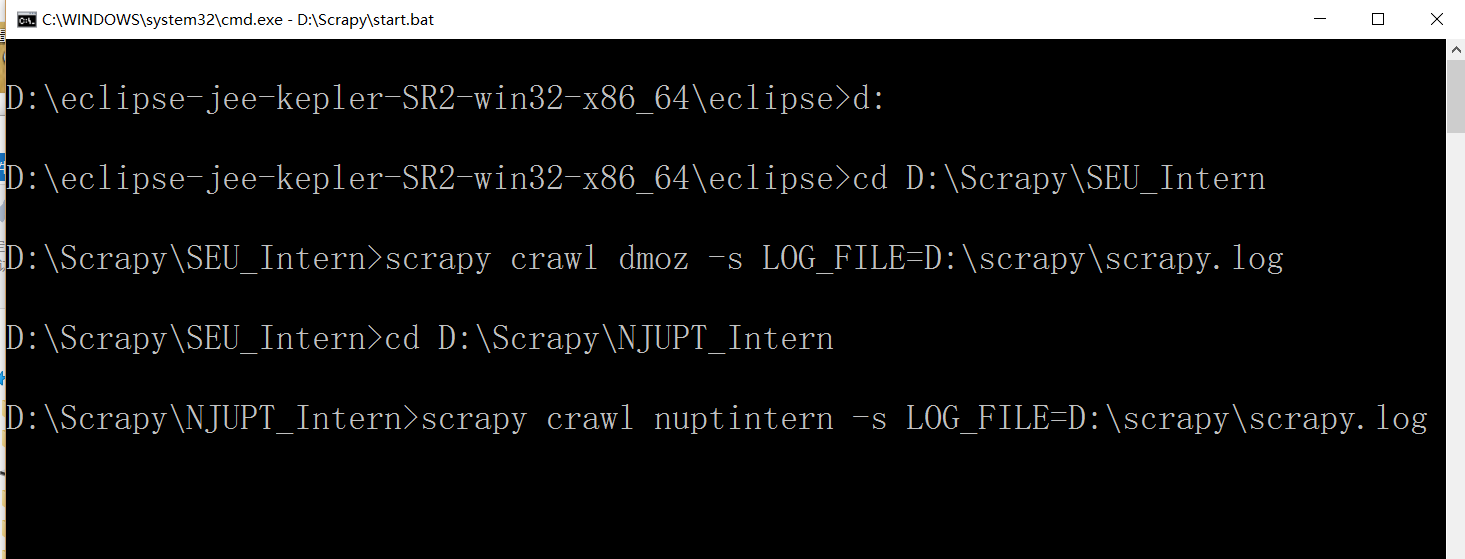


图4.3-3 执行start.bat文件调用Scrapy框架



图4.3-4 数据采集完成，显示“OK”

### 4.3.2 数据的处理过程

之后，我们应该点击Data Presentation来完成数据的处理。同样，在数据完成时，Web页面上将会出现“OK”，并停留3秒后，消失。如图4.2-5和4.2-6。

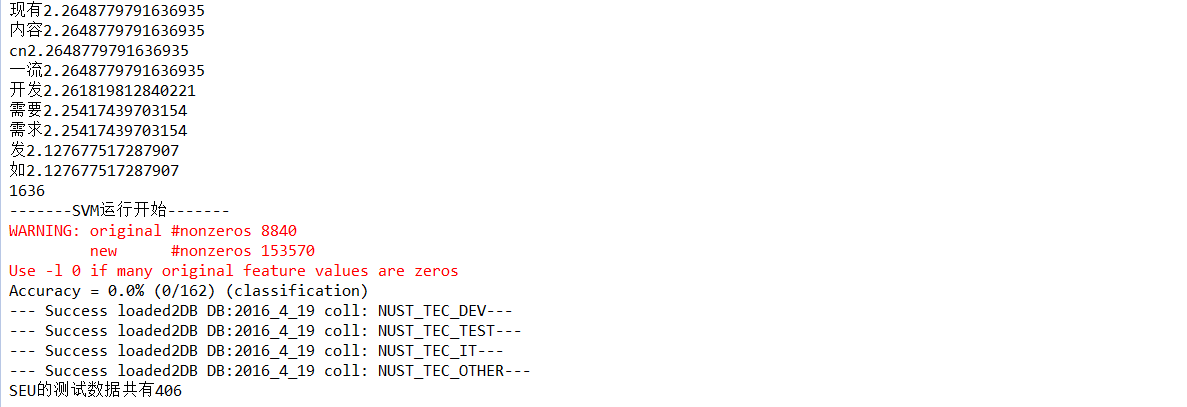


图4.3-5 数据处理阶段，eclipseEE控制台显示



图4.3-6 数据处理完成，显示“OK”

### 4.3.3 数据的展示过程

在经过以上两个步骤后，其实服务端已经完成数据的采集，处理以及存储了！之后，我们就可以点击下面DEV，TEST，IT，OTHER来完成研发（开发），测试，技术支持以及其他招聘信息岗位的展示。如图4.2-7，4.2-8，4.2-9，4.2-10。

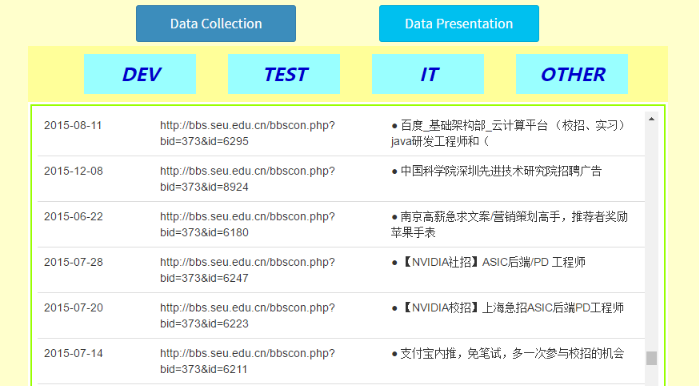


图4.2-7 研发/开发（DEV）招聘信息显示

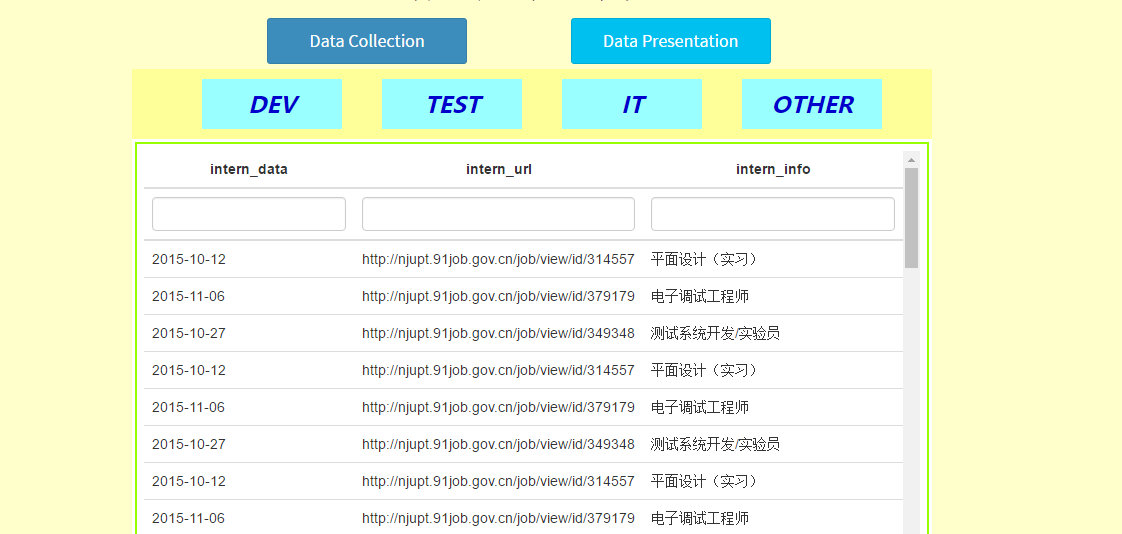


图4.3-8 测试（TEST）招聘信息显示

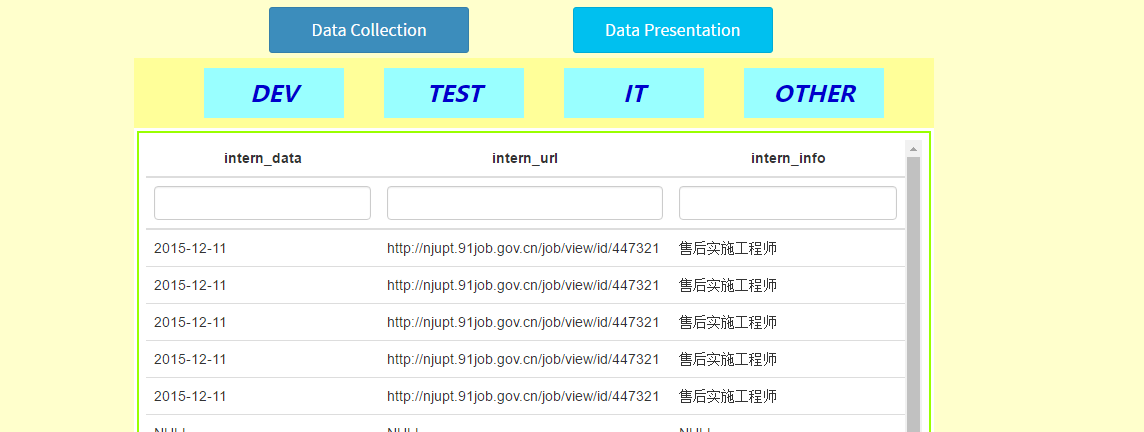
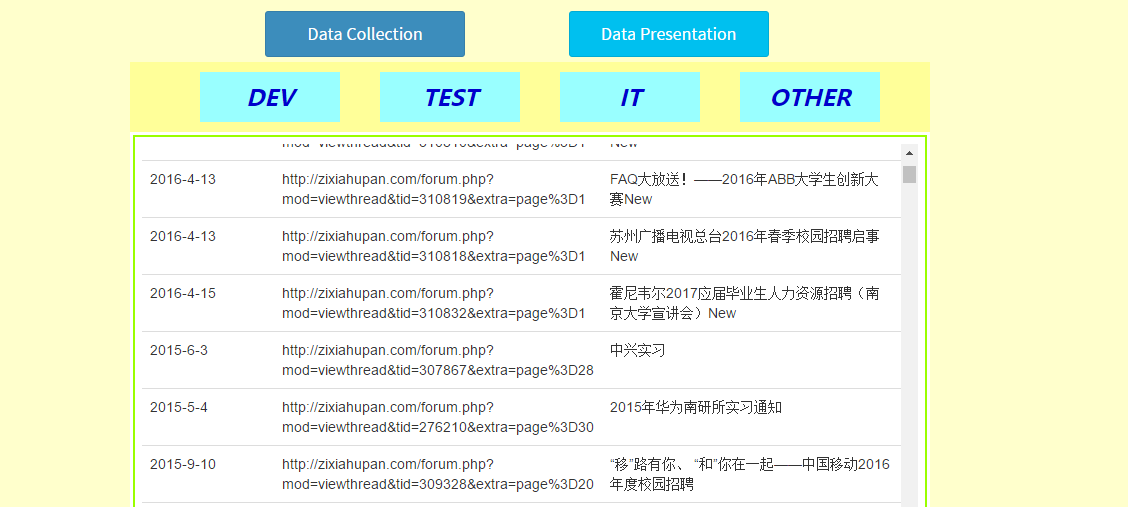


图4.3-9 技术支持（IT）招聘信息显示

图4.3-10 其他（OTHER）招聘信息显示

## 4.4 针对个人简历来提供匹配合适的招聘信息

首先，我们可以在浏览器上输入http://服务器的IP地址:8080/resume/，这时会出现个人简历页面（如图4.4-1），求职者应该在上面完成个人简历的填写，其中，你必须选择好论文发表等级。然后，你可以提交简历，交给服务器端进行数据的处理以及招聘信息的展示。在网页的标题上会出现你所适合的岗位，然后在页面上出现符合你岗位的最近招聘信息，如图4.4-2。



图4.4-1 个人简历填写页面

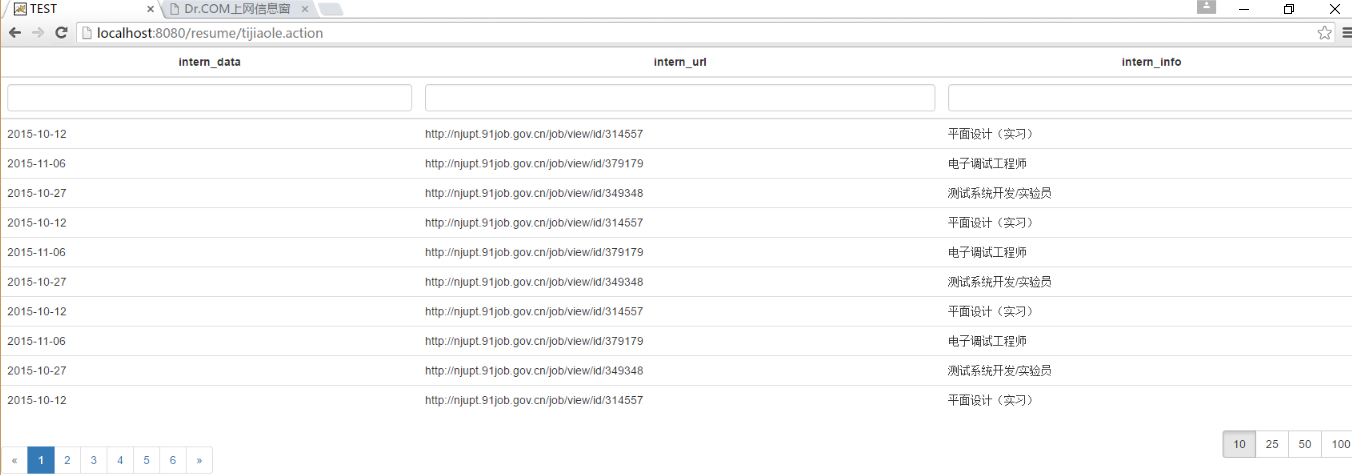


图4.4-2 与你简历匹配的的求职招聘信息展示

# 五、总结

感谢王老师在项目过程中给予的指导帮助，感谢组委会给予的项目开展机会。

本项目开发的就业咨询智慧服务系统，通过计算认知所涉及的相关算法完成分类与预测功能的建模，并使用自然语言输出，可以有效为院校学生求职岗位的选择提供智能化的辅助咨询服务。该系统使用Hadoop与Spark平台所提供的分布式大数据处理功能，能够保证数据的充分挖掘,使用SSH作为用户服务平台能够提高人机交互体验，使用MongoDB数据库将有效完成数据存储、冗余备份、数据加密的功能。

在设计与实现该系统的过程中，存在数据编码、格式处理、Tomcat与Spark运行环境冲突等诸多问题，都是在组员讨论中得到解决。感谢组内每一位成员的辛勤努力，再次感谢组委会给予的项目开展机会。