**A07 基于大数据的岗位画像和求职者画像设计**

**新增： 前台的四个页面以及用户数据收集。**

**/\***

**通过网络爬虫爬取智联招聘、51job、拉勾网等招聘网站上，大数据相关职位的招聘信息。**

**\*/**

1. 需求分析流程（√已完成，\*进行中，其余未开始）：
   1. 了解‘岗位画像’和‘求职者画像’√
   2. 查找同类型产品\*
   3. 分析本项目的基本流程√
   4. 分析爬虫需要设置的初始条件√
   5. 分析爬虫需要爬取的公司信息和职位信息（根据各大招聘网站分析）√
   6. 查找如何结构化处理数据的资料\*
   7. 提取‘可能’的岗位工资影响因素√
   8. 设计‘岗位需求能力图谱’的属性√
   9. 分析求职者如何匹配公司（或根据求职者的信息如何推荐公司给求职者）
   10. 设计‘岗位画像’与‘求职者画像’
   11. 分析如何展示分析结果及两个画像
   12. 分析已有需求仍存在哪些问题
2. 用户画像： <https://www.cnblogs.com/cescyang/p/6017608.html>
   1. 用户画像是对现实世界中用户的数学建模，它包括两方面：
      1. 一方面是描述用户，没有说人，是说明它跟业务密切相关，它是从业务中抽象出来的，因此来源于现实，高于现实。
      2. 第二个是用户画像它是一种模型，是通过分析挖掘用户尽可能多的数据信息得到的，它是从数据中来，但对数据做过了抽象，比数据要高，后面所有用户画像的内容都是基于这个展开的。比如刚刚说的月光族，这个肯定是挖掘分析出来的，不是说原来的数据中包含月光族这个标签，所以说这是它的两层含义。
   2. 理解：
      1. 画像对应类，分为岗位画像、求职者画像
      2. 显性属性对应爬去信息
      3. 隐形属性对应分析结果
3. 基本流程：
   1. 设置爬虫条件（页面）
      1. 管理员对网站列表增删查改
         1. 选择爬取的招聘网站（可多选，网站是确定的）
         2. 增加、修改爬虫的职位或单个url：
            1. 当管理员输入职位关键字时，异步获取相关的url，爬取并显示职位目录列表
         3. 删除爬虫的职位及其相联url或仅删除单个url
         4. 显示结果：
            1. 显示url，爬取成功显示√，失败显示‘爬取失败’。
            2. 爬取失败的url移入‘黑名单’，服务器空闲时重新爬取‘黑名单’的url,若成功则移除。
      2. 设置开始爬取的时间、爬取的频率（每天一次或者每天两次等）
      3. 爬虫支持深度优先或广度优先策略，要求提供自研算法。
   2. 爬取数据（网络爬虫）
      1. 提取出其中的关键数据，包括但不限于职位名称、职位待遇、
      2. 职位描述、公司介绍、公司规模、公司性质等信息。

注：comp全称为company

* + - 1. 公司名称（compName）：根据公司名去爬取百科的信息
      2. 公司性质(compNature)
      3. 公司规模(compSize)
      4. 公司行业(compIndustry)
      5. 联系方式(contactInfo):电话/邮箱都可以
      6. 公司介绍(compIntroduce)
      7. 公司链接(compLink)

* + - 1. 职位名称(workName )
      2. 工作地点（省市结合）(workPlace)
      3. 工资待遇(workSalary)
      4. 福利待遇（五险一金、补充医疗保险等）
      5. 职位信息(workInfo)：根据职位要求的关键字（设置权值）匹配
      6. 招聘人数(workNeed)
      7. 发布时间(releaseTime )
      8. 学历要求(educationRequire)
      9. 经验要求(experienceRequire )
      10. 专业技能要求(skillRequire)
      11. 职位招聘链接(workLink)
  1. 结构化处理数据（结构化）
  2. 分析、挖掘数据（算法分析）
     1. 数据分析流程：
        1. http://python.jobbole.com/81133/
        2. 数据导入
        3. 导入本地的或者web端的CSV文件；
        4. 数据变换；
        5. 数据统计描述；
        6. 假设检验
        7. 单样本t检验；
        8. 可视化；
        9. 创建自定义函数。
     2. 提供优质挖掘算法或解析规则
     3. 分析：
        1. 岗位工资可能的影响因素
           1. 公司规模（规模越大，可能工资越高）
           2. 工作地点（所在城市越好工资可能越高）
           3. 融资阶段（只有部分招聘网站有）
           4. 公司行业（高新技术行业可能工资高于传统行业）
           5. 公司性质（国企、民企、外企等）
           6. 岗位性质（全职、兼职or实习）
           7. 招聘人数与求职人数（供求关系）
        2. 岗位能力需求图谱
           1. 专业技能
           2. 工作经验
           3. 学历要求
           4. 是否相关专业
           5. 个人品质（团队协作能力）
           6. 获奖经历（待定）
        3. 岗位的招聘企业画像
           1. 公司性质
           2. 公司规模
           3. 公司行业及行业发展前景
           4. 同类型公司排名
           5. 公司介绍
           6. 公司成立时间
           7. 公司投资方
           8. 福利待遇
  3. 智能推荐岗位（算法推荐）
     1. 当求职者输入学历、专业、学校、求职地、工作年限、技能、岗位名称等基本信息后，系统将智能分析出该职位的待遇水平、求职者的待遇区间、可能去的公司、公司性质和规模、行业、匹配概率等信息。要求提供：求职者画像及岗位个性化推荐算法。
     2. 岗位画像
        1. ？（显）职位的待遇水平：所有公司该岗位的平均待遇水平
        2. 建表打分（五险一金 补充医疗保险 交通补贴 餐饮补贴 通讯补贴 定期体检等）
     3. 求职者画像
        1. （隐）求职者的待遇区间（岗位的隐形+求职者的显性得出）
        2. \*\*\* 根据岗位能力需求图谱（岗位的隐性属性）进行评分
  4. 展示分析、推荐结果（页面）

1. 问题
   1. 是否只需要大数据相关职位
   2. 对每个招聘信息的展示方式不同，不好提取
2. 技术
   1. 版本控制(github)
   2. 开发语言：python
   3. 网络爬虫（python+scrap+redis）
   4. 大数据挖掘（？？）
   5. 大数据分析（分析算法）
   6. python的后端框架（Flask、Django之一）
   7. 前端框架（angular,bootstrap,easyui(后台页面),jQuery,html+css+js）
   8. 数据库（mysql+redis）
   9. 日志记录
3. 整理需要学习的东西（文档、视频）发布到群公告
   1. python学习教程链接、github教程链接
   2. 题目文档、画像介绍的博客、python日志使用
4. 问题
   1. 爬虫自研算法，怎样提高效率，如何优化？
   2. 快速匹配，精准推荐给用户岗位
   3. 如何结构化数据，爬取的数据构建成表（python提供）
   4. 前端的效果，页面的构成部分
   5. 匹配算法（消耗时间）
   6. 求职者画像（ 提供标签自选，录入基础数据（ 薪资，地点 ） ）
   7. 展示页面的数据，几个页面，团队
   8. #
5. 封面效果
6. 界面
   1. 首页
      1. 导航栏 + 轮播（广告，网站公告（java培训广告， 热门职业（用户搜索最多））， ） + 推荐同类型求职者的职位匹配
   2. 个人详情
      1. 个人信息：真实姓名， 性别（男女其它）， 出生日期（日历本）， 联系方式， 地点（接口）， 邮箱， 学历（多选）。
      2. 工作经验：工作年限（单选），职位（手动输入），公司（手输入）
      3. 学历模块： 毕业学校（手输入） 学历（多选） 毕业年限（多选） 专业（手输入） 外语水平（暂时英语四六级）， 获奖（不匹配，投简历可用）
   3. 匹配数据页面
      1. 岗位名称 – 地点 – 薪资 – 公司
   4. 注册页面

昵称，手机号， 密码， 邮箱， 手机验证码。调至个人信息完善页面

* 1. #

1. #