5 评估

本章节将对本篇论文进行一个评估。评估内容涵盖了项目实施的方法、项目结论以及将结果与现有的文献进行比较以分析整个项目的优缺点。

项目评估

对于整个项目所有的里程碑都已经被证实是完成的。在实际实施过程中，数据的复杂程度远远超出了我的预计因此更多的时间都花在了数据处理上。项目实施的过程中，总体上分为了三个主要的阶段它们分别是数据搜集、数据处理以及数据分析。

在数据搜集阶段使用scrapy框架对网络招聘的数据进行挖掘。在实验中同一套代码能够同时应对两个不同的网站。在进行数据挖掘的过程中与文献中进行的内容有所改进。我在数据搜集的时候就按照每一个岗位的URL来判断是否是之前已经存储过的数据，在这一步骤中我就完成了数据的去重处理。在数据存储方面我选择将数据存储在了数据库之中，这种存储方式更加安全可靠。在应对反爬虫系统的时候，我选择与文献中相同的方法即添加头请求以及增加虚拟cookie的形式来实现。在本项目的第三章节详细记录了这些内容的实施过程。Chang et al (2019)就提出使用直接输出Excel的文件来实现数据输出。他们所采用的方式是在程序中添加断点来避免从头开始获取数据。但他们的程序有一大缺点就是很容易造成数据重复。因为网站的招聘信息每一天都在更新，如果一天之内不能够完成任务那么第二天搜集到的数据就会有重复的。在反爬虫方面Hu et al (2019)就提出了可以使用ip代理池的方式来实现，但是我在实际实验过程中发现这一措施并没有实际的作用。对于获取网页内容Yin and Zhang (2019) 以及Chang et al(2019)都使用了正则表达式来获取内容，而我所使用的是xpath方法从复杂度层面来说两者并无实质区别。在自动获取数据的方法中Sun et al(2019)等人所使用的方式是判断是根据评论数量来判断是否有下一页的存在。我所使用的方法是直接循环生成下一列的链接来实现。原因是我发现页面的总数总是固定的因而不需要再进行复杂的判断了。

数据清理阶段，由于是两个完全不一样的网站因此数据的结构也是完全不一样的。在我看来最主要的工作就是统一所有的数据。对于工作地点这一属性对于英国与中国两个不同的招聘网站我都去除了一些小城市并将它们归入相应的省份或郡中。Yin和Zhang(2019)采用按照城市规模将城市分别划分为了一线城市、二线城市和三线城市。在中国的招聘网络中对于岗位分类与企业分类也进行了大规模的精简工作。在51job的java数据中它们分别有160和120个不同的数据。然而这其中有些变量仅仅只有几个数据，而一旦这些数据都运用到了接下来的数据分析中去那么对于最终的结果是缺乏参考价值的。因此我仅仅保留了那些含有数量较多的变量，对于那些极少数的变量都划分进了‘others’属性中。对于划分工资等级的工作花费了非常多的时间。我在实际工作中总共进行了三次实验最终确定了按照如下的划分方式来进行划分。原本计划将工资等级划分为三个不同的等级，但是在实际的分析过程中我发现，收入最高的那个等级很少能够被正确地识别出来因此这一划分的方法被证明是错误的。岗位信息在这里划分为了岗位内容与应聘要求。在51job中挖掘这一类数据的时候我用正则表达式通过识别关键字来获取这些数据。在实际获取数据集了之后我发现有大约4000条数据并有被正确地划分，因此我在Excel中再次对这一部分的数据进行更深一步的处理最终搜集到了5848条岗位内容与5548条应聘要求。这一规模的数据量对于接下来的关键词提取起到了关键的作用。而对于英国网站的这一类数据由于它们本身就缺乏统一的结构因此在搜集的时候就不再进行分开搜集了。

在数据可视化中Wang(2019)给出了薪资与城市的关系，通过饼图来挥之各城市之间的岗位数量，绘制直方图来展示各个城市的薪资水平。从他的结论中看出上海、深圳、北京、广州、杭州等城市的java岗位人数是最多的。而且上海的需求量是远远大于其他城市的。这与本项目所得出的结论相同。Yang(2019)绘制了薪资与工作经验的关系图得出了工资随着工作经验的增长而增长。这与我的结论是一样的。同时也证明了我的结论是正确的。Wang(2019)给出了关键词提取，根据默认参数提取了前二十个关键词

对于本项目中所建立的模型评估如下。在本项目中总共运用两种软件建立了四种不一样的模型，它们分别是决策树模型、关联算法以及聚类算法。我分别使用了python与weka来建立决策树模型。在weka中分析决策树模型用了ID3算法与J48算法，其中发现ID3算法造成的决策树模型极其庞大原因是它并没有进行剪枝操作。因此实际的实验中仅保留了J48模型。通过比较可以发现模型的预测准确度是python的决策树模型略高一些，但实际上并没有相差过多。

对于总体而言，决策树模型要优于关联算法，关联算法要优于聚类算法。因为在数据集中数据的属性非常多，而聚类算法所使用的KNN模型给出的聚类并不能很好地代替大多数因此在本项目中这一算法给出的结果实际意义并不是特别明显。由于属性集的不足与样本数据的缺乏，在本项目中没有对jobtotal的数据集以及51job中的python数据集建立模型来寻求相关的模式。

总结

总而言之，本项目无论从结果和过程来看都是成功的。本项目通过挖掘现有的网络招聘的数据通过模型与编程获得了预期的想知道的内容。在本项目中回答了如下三个问题

1. 程序员应该掌握何种技术来使自己在就业中更加具有竞争力
2. 哪些参数会使得一个程序员更有可能获得中高收入
3. 在何处更加具有发展前景与机遇

本项目基于 python 实现对网络招聘信息数据挖掘，根据分类模型对招聘信息进行待遇值预测。本次数据挖掘主要采用 python语言、openrefine软件及weka软件对数据挖掘过程中的数据采集、数据处理、数据可视化以及数据建模均有实现。 从本次研究的过程中可以看到多个方法共同协作对于数据挖掘提供了强大的支持。通过运用python语言在jupyter notebook中进行可视化操作，相较于文献中的片面性我完成了对所有属性之间的信息挖掘。除此之外我还更关注于岗位内容的信息挖掘，这一点是之前没有一篇文献提到的。本次对网络招聘信息薪资的数据挖掘主要立足于求职者角度去分析，提出招聘待遇新特性，既可提高求职者招聘效率，又能一定程度减少企业招聘成本。此外还对打算就业的人提供了有价值的信息参考。

但是有一点遗憾的是对于英国互联网招聘的数据并没有做得和中国互联网招聘数据分析一样详尽。这其中一个主要的原因是网站本身所提供的数据就较少，很多数据是缺乏实际分析价值的。原本计划对两国所获得的招聘数据有一个详细的比较，但由于这一原因获得的数据集量过少而不得不放弃。不得不承认的是现如今各大互联网企业都已经意识到了数据的重要性因此传统的爬虫方式已经很难应付这一点。原本计算法对于英国招聘网站和中国的招聘网站各找两个网站。然而在实际的过程中我原本备用的几个网站都因为反爬虫措施的完备而让我望而却步。如今所获得的数据集是现存仅有的既能保证数据的可靠性又有数据的可获得性。这也是本项目中的一个比较可惜之处。

总结与未来工作

总结

项目的起源是来自于我发现现在网络招聘非常地热门，网站在发布它们的招聘信息的同时也提供了岗位信息大数据供人分析。而很多的网咯招聘信息分析都仅仅局限于网站实际给出的数据却忽略了在岗位信息与岗位要求中仍然存在与大量的数据可以分析。通过使用常用的scrapy框架来获取原始数据、使用openrefine及Python来清理与整合数据以及使用weka与jupyter notebook来分析数据。总体上来说，本项目完成了需要实现的主要目标与目的。

数据采集阶段，我使用 python 的 scrapy 爬虫库并充分结合 Web 数据挖掘原理，在招聘信息网中采集到了本项目所需的数据集。该基于scrapy框架的爬虫程序能够适配中国的招聘网站与英国的招聘网站。与此同时我还成功地运用了反爬虫技术来获取数据。

数据预处理阶段，同时使用openrefine、python以及Excel来对数据进行清理与整合。利用python绘制数据可视化来给数据清理工作提供参考。使用Excel来整合相同的属性。利用openrefine来进行数据填充。同时还使用openrefine来改变数据类型例如将numeric value转换为nominal value反之亦然。

数据可视化阶段，使用Python语言在Jupyter Notebook中进行绘图来寻找不同数据属性之间的相互关系。此外还利用python关键词提取功能对岗位信息内容进行了一个提取并使用Wordcould包来对提取到的关键词按照词频来显示出来。

数据建模阶段，我使用python 的机器学习库 sklearn，通过简短的代码实现了决策树算法的模型构建。我还使用weka软件中的J48算法、Apriori算法以及simpleKmeans算法分别实现了分类算法、关联算法以及聚类算法。

项目评估阶段，评估了之前根据预测结果生产混淆矩阵，计算出模型的准确率和错误率。对两种不同软件中的分类算法进行了对比评估，得出python决策树模型具有较高的预测准确度的结果。对于岗位信息的挖掘得出了无论是java程序员亦或者是python程序员都需要掌握数据库操作能力这一结论。除此之外对于java来说目前在各大企业中最广泛使用的模型就是spring模型。本项目还对于程序员如何获得一份体面的收入也给出了一个综合的分析。这是一个非常好的建议提供给正在学习计算机编程语言的年轻人以及在找工作的程序员。

未来工作

在实际的数据搜集过程中，我已经发现了现有网站对于自家数据的保护意识是非常强烈的。我所设计的基于scrapy框架的爬虫程序已经很难应付大多数网站的反爬虫机制。我发现目前有些网站对于页面生成已经使用动态的形式呈现。传统的分析网页源代码形式挖掘数据已经举步维艰。因此我希望在接下来的学习过程中能够寻求一种方法在合法合规的原则上进行网站信息搜集。在信息搜集的过程中在未来可能会更加依赖于网络服务器。因为在自己实验的过程中占用了自己电脑的大量内存从而导致电脑除了运行程序而无其他事情可做。

此外，对于英国网站的数据分析较差的原因就是网站本身将大量有用的信息例如工作经验以及学位要求都放入了岗位信息描述中。这点与中国馆网络招聘网站将关键信息单独列出来不同。我使用传统的通过词频来获取这一类关键信息的方式是很难收获预期的成效的。通过了解得知这一部分的知识是属于自然语言处理的范围内。因此在未来我想通过学习自然语言处理来将关键的信息提取出来再来对招聘信息有一个全面地彻底地分析以此来找出一个收益最大化的模式。

摘要

随着互联网的迅速发展，大数据时代的来临，数据挖掘在从海量数据中探查潜 在的价值信息起到了重要的作用，成为当下热门的研究和实践方向之一。python 作为数据挖掘领域中较为热门的程序语言，其丰富的技术库和强大的科学计算能 力成为数据挖掘过程中不可或缺的工具。本次研究主要是基于 python 语言对智联 招聘网的数据进行数据挖掘分析和建模，进而得出招聘信息薪资待遇预测分类模 型。 本次研究主要分为如下步骤：数据源选择、数据采集、数据存储、数据预处理、 数据建模和模型评估，通过算法构建了近邻和决策树两种分类模型，其次对两种 模型的混淆矩阵数据进行计算，比较模型的预测准确率，最终得出准确率较高的 数据模型。 本次研究所得的分类模型，可以帮助被招聘者在浏览网站招聘信息时预测薪资待遇水平，有效的评估招聘内容是否合适，以及对招聘岗位提供的薪资待遇是 否感到满意，进而有效的提高求职者在寻求招聘岗位时的效率。 同时，该模型对于企业优化招聘信息也起到反馈性作用。企业可以根据模型 对市场中现有招聘信息薪资水平进行有效分类，从而得到市场不同岗位的薪资水 平分布现状，进而对企业招聘内容进行合理的调整，提高企业招聘效率，节约招 聘成本，优化企业人才结构，提高企业在行业的竞争力。

随着互联网的迅速发展，我们目前正处在一个大数据时代。数据挖掘在从海量数据中探查潜 在的价值信息起到了重要的作用，成为当下热门的研究和实践方向之一。此外现如今人们在寻找工作的时候不再像过去那样参加大型线下招聘会。企业会在综合招聘网站发布相关的招聘信息，并通过接收简历的形式来选拔未来的员工。成千上万条的招聘信息就这样被发布出来。本次研究主要是对51job和jobtotal两个不同招聘数据网站进行数据挖掘，进而得出招聘信息薪资待遇预测模型。此外还通过数据可视化的方式来展示各个招聘属性与薪资之间的关系。本次研究主要分为如下步骤：数据采集、数据预处理、数据可视化、数据建模和项目评估。

首先使用Python中的scrapy框架来搜集互联网招聘信息。紧接着我使用Openrefine、Python和Excel三种软件共同协作来进行数据处理。通过Python和Weka通过代码构建了决策树分类模型，然后对两种工具得出的混淆矩阵数据进行计算，比较模型的预测准确率，最终得出准确率较高的数据模型。此外，还使用Weka软件中的Apriori算法和SimpleKmeans算法分别实现了关联算法和聚类算法，并列举出所获得的规则。本次研究所得的模型，可以帮助被招聘者在浏览网站招聘信息时预测薪资待遇水平，有效的评估招聘内容是否合适，极大地缩短了寻找岗位的效率。除此之外还利用大数据来寻找当前那种技术是一个程序员应该掌握的。

Data clean tools

OpenRefine是一个数据转换工具（IDTS），其能够对数据进行可视化操作处理。它很像传统的excel软件，但其工作方式更像数据库，因其并不是处理单独的单元格，而是处理列和字段。这意味着OpenRefine对于增加新行内容表现不佳，但对于探索、清洗、整合数据却功能强大。Openrefine也提供一些自定义筛选选项，其可以给大多数用户提供有用的筛选功能。其次，它还可以进行单元格格式的转换，通过使用Blank down菜单并再选择transform即可。点击Categories列下拉菜单选择“编辑单元格” |“转换”会出现转换对话框：方框内可以输入小脚本，用来修改值。Language可以让我们选择表达式的编程语言，现在支持的语言有General Refine Expression Language(GREL), Jython (Java环境的Python语言)。

OpenRefine is a data conversion tool, which can perform visual operation processing on data. It is much like traditional Excel software, but it works more like a database because it does not deal with individual cells, but with columns and fields. This means that OpenRefine does not perform well for adding new content, but it is powerful for exploring, cleaning, and integrating data. Openrefine also provides some custom filtering options, which can provide useful filtering functions for most users. Secondly, it can also convert the cell format by using the Blank down menu and selecting transform. Click the drop-down menu in the Categories column and select "Edit Cell" | "Transform". A transformation dialog box will appear: a small script can be entered in the box to modify the value. Language allows us to choose the programming language for expressions. Currently supported languages are General Refine Expression Language (GREL), Jython (Python language in Java environment).

Wang(2019)通过使用Python中的Pandas库来进行数据清洗。Pandas是基于Numpy构建的含有更高级数据结构和工具的数据分析包。数据缺失在大部分数据分析应用中都很常见，Pandas使用浮点值NaN表示浮点和非浮点数组中的缺失数据。处理NA的方法有四种：dropna,fillna,isnull,notnull。可以利用fillna实现许多别的功能，比如可以传入Series的平均值或中位数。DataFrame的duplicated方法返回一个布尔型Series，表示各行是否是重复行。

首先非常感谢我的导师彭涛新博士，论文的选题、 研究内容、组织结构、系统的研究进展以及各个阶段的研究方向都离不开梁老师 的悉心指导，老师的严谨治学、孜孜不倦的研究态度我将一直谨记于心。我还要感谢Pete, Barclay博士，他在项目初期以及中期都给我提供了非常宝贵的建议。

最后，我要感谢所有对我有过帮助和鼓励的老师、同学和家人， 祝大家身体健康、事业顺心。