Python-based recruitment data collection and analysis

Yijia Sun

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
BSc Computing

School of Computing

April 2020

Authorship Declaration

I, Yijia Sun, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed:

Date:

21.3.2020

Matriculation no:

40450455

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Abstract

With the rapid development of the Internet, we are now in a big data era. Data mining plays an important role in exploring potential value information from massive data, and becomes one of the hot researches and practice directions. In addition, people are no longer going to large-scale offline job fairs when they are looking for jobs. Enterprises will publish relevant recruitment information on the comprehensive recruitment website, and select future employees by receiving resumes. Thousands of recruitment information were released. This research is to mine the data of 51job and jobtotal, and then get the salary forecast model of recruitment information. In addition, data visualization is used to show the relationship between recruitment attributes and salary. This research is mainly divided into the following steps: data collection, data pre-processing, data visualization, data modelling and project evaluation. Firstly, I use the scrapy framework in Python to collect Internet recruitment information. Then I use OpenRefine, Python and Excel three kinds of software to work together for data processing. The decision tree classification model is built by Python and Weka through code, then the confusion matrix data obtained by the two tools are calculated, the prediction accuracy of the model is compared, and finally the data model with high accuracy is obtained. In addition, Apriori algorithm and SimpleKmeans algorithm in Weka are used to implement the association algorithm and clustering algorithm respectively, and the rules obtained are listed. The model obtained in this study can help the recruiter to predict the salary and treatment level when browsing the recruitment information of the website, effectively evaluate whether the recruitment content is appropriate, and greatly shorten the efficiency of job search. Besides, using big data to find what current technology is a programmer should master.

Contents

List of Tables

List of Figures

Acknowledgements

First of all, I'd like to thank my supervisor, Dr. Peng Taoxin. The topic selection, research content, organizational structure, systematic research progress and research directions at all stages are inseparable from your careful guidance. I will always keep in mind the rigorous research attitude and tireless research attitude of the teacher. I would also like to thank Dr. Pete Barclay, who gave me very valuable advice at the beginning and middle of the project.

Finally, I would like to thank all the teachers, classmates and family members who have helped and encouraged me. I wish you all good health and a happy career.

1 Introduction

1.1 Project Background

With the continuous development of modern information technology, most countries in the world have entered the Internet era. Nowadays, most young people mainly apply for jobs through the Internet, and the main way for enterprises to recruit is gradually based on the Internet. Therefore, a variety of comprehensive recruitment websites have emerged on the Internet, such as India and Liepin. The enterprise publishes position information on these websites, and the candidates choose according to their needs. However, there are also disadvantages in the process of internet recruitment. There will also be a release time for each published position information on the company's website. The longer the publishing time is from now, the two problems will be explained. One is that talents in this position are indeed scarce, and the other is that enterprises do not like to use Internet recruitment. Through the actual observation of several comprehensive recruitment websites, I found that they lack comprehensive display area for position information data. At present, with the continuous accumulation of comprehensive recruitment website data, there are many people want to mine useful information for analysis. D. Smith collects recruitment information about program developers from the recruitment website, and uses keyword index technology to study the demand trend of enterprise for the programming ability of job seekers, so as to provide important reference for the curriculum setting of computer major in Colleges and universities. (Smith 2014) Jia obtains the position information by retrieving the position information of data analyst in China's Lagoon network. His analysis is not only a single job but only a recruitment website, which will lead to incomplete data. (Jia 1019) In addition, most of the relevant researches at home and abroad adopt the traditional sampling method to sort out the recruitment website data by artificial statistics. (Jiang et.al 2016) The data volume is small and the efficiency is low, which is not suitable for the rapid and intelligent data collection and analysis in the big data environment.

1.2 Project Aims & Objectives

The aim of the project is to investigate online job market in order to provide support to both job seekers and companies. This will be achieved by conducting a detailed literature review of the current methods and tools used to obtain the key data from the recruitment website and analyze this data. This project will provide some reference for those who are planning their careers to keep up with the trend of the times. This project will allow the job seekers to know which positions are popular and which positions are in the greatest demand and so on through the data collected. A set of suggestions will be generated, which will provide assistant to both job seekers, and companies. Let job seekers know where they can find a satisfactory job in their current major or skills.

The objectives are:

- Literature review-learn Python web information mining technology, data analysis and data science.

- Data collection and integration- Compile a web data mining program according to the knowledge learned from reading literature and explain how to collect the integrate data.

- Analyze data- Visualize the acquired data to observe whether there is a certain relationship between the data.

- Analyze and evaluate findings- Evaluate results from implantation of the proposed method and experiments, analyze findings and compare the results.

- Finish essay- Write the paper strictly according to the specified format and summarize my findings.

1.3 Project Outline

Chapter 1 Introduction: Contains an overview of the project and provides background for the project. Also contains a breakdown of the projects aims, objectives and plan.

Chapter 2 Literature Reviews: In this chapter, it introduces the existing methods of network data mining and how to use Python for data mining and analysis.

Chapter 3 research of data mining method: In this chapter, it introduces the research of data mining method of recruitment information network. The content includes the use of Python to write a crawler software, data collection and data storage.

Chapter 4 Data mining implementation: This chapter will introduce how to collect and preprocess data.

Chapter 5 Data analysis: This chapter introduces data mining modeling and evaluation of the established model. In addition, this chapter also uses machine learning to predict the data.

Chapter 6 Conclusion: This section provides a review of the entire project, explaining the problems encountered throughout the project and how to overcome them. It also provides a summary of project outcomes and deliverables, including recommendations for people in the choice of employment.

1.4 Project plan

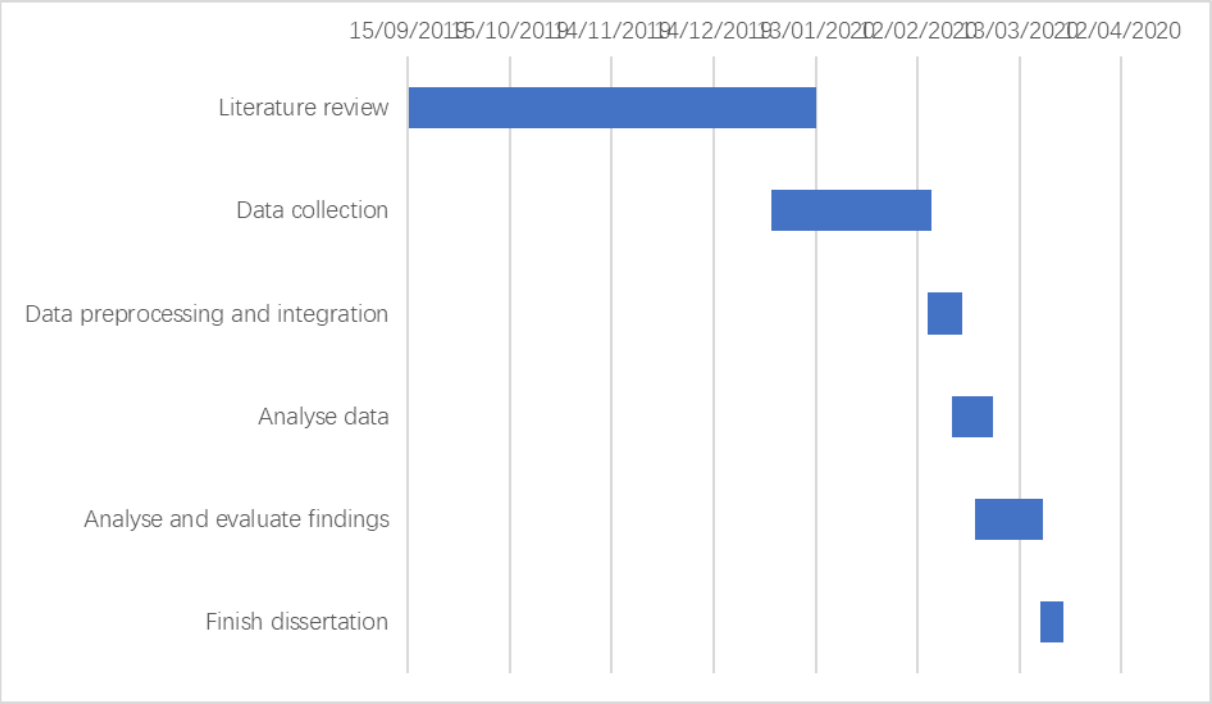


Figure 1 G-chart of the project

2 Literature review

This chapter provides a literature review of the project and is carried out in five parts. The first part is to learn the background of this project, which can clearly understand the current operation process of online recruitment and the existing shortcomings. The second part is the knowledge learning of network data mining, to understand the existing methods of network data mining and how to realize them. The third part is data processing, which is how to deal with the data obtained before and prepare for the analysis. The fourth part is data analysis, which introduces the existing data analysis methods for the next project analysis work. The fifth part is the introduction of data analysis tools to understand how to realize the data analysis in the previous section. Finally, it summarizes the whole literature review and evaluates the feasibility in this project.

2.1 Internet recruitment

Internet recruitment refers to the recruitment activities carried out through the Internet, through publishing position information on the recruitment website, collecting and sorting out resumes, and providing online or offline interview and evaluation for candidates. Online recruitment originated in the United States and has become the main way of talent recruitment in many countries. According to the statistics of Fortune magazine, more than 88% of the global top 500 enterprises use online recruitment. (Zhao et al 2012). It can be seen that although online recruitment has become the mainstream recruitment mode, the traditional recruitment mode has not been completely abandoned. In the traditional way of recruitment, enterprises need to design job description and determine the geographical area and social position of job seekers in the labour market. Then, companies need to decide on feasible ways to attract job seekers, such as advertising in newspapers or finding high-quality resumes from recruitment agencies. This process requires a lot of money to the advertising media and middlemen. (Li 2019). After the establishment of the collection, the human resources department will spend a lot of time to select the right potential candidates to meet the requirements. They are then assessed to see if they need to be given an interview. The traditional recruitment mode greatly increases the

workload of recruiters, and a lot of manual information is also included in the recruitment process. Online recruitment has many advantages, such as no geographical restrictions, wide coverage, low recruitment cost, strong pertinence, convenience and timeliness. (Liu 2017) At present, there are two forms of enterprise online recruitment: one is to publish recruitment information on the official website of the enterprise and build a recruitment system; the other is to cooperate with the professional recruitment website to publish recruitment information and carry out recruitment activities through the professional website. （Dai 2012）The former is suitable for high-profile large-scale enterprises, because their company names have been passed on from person to person. However, for the vast majority of companies, their popularity is not so high, so the professional recruitment website is particularly important. In Xing's description, Internet recruitment websites use Internet channels to build an information intermediary platform for recruitment enterprises and job seekers, which is used as an information intermediary recruitment platform to obtain traffic and revenue. (Xing 2017). In addition, he also mentioned that in the current recruitment market, the revenue of online recruitment has exceeded that of paper recruitment, and a number of comprehensive Internet recruitment websites have been listed in many places.

2.1.1 Current situation of recruitment market

Comprehensive recruitment website is the most mainstream recruitment website business model at present. According to Analysys, by 2019, China's Internet recruitment market is expected to reach 10.31 billion yuan. 51job, Zhilian recruitment and 58 market share are nearly 80%. With the gradual maturity of information technology such as mobile Internet, artificial intelligence and big data technology, the construction of Internet recruitment products will focus on the upgrading of search matching algorithm and user experience. (Xing 2017) However, the application's automatic filtering function may exclude talents with special skills from the employment list, which is the human resources that can greatly help the company. (Li 2019) The recruitment information published by the recruitment website can best reflect the market demand for talents, including the requirements of enterprises for all aspects of the skills of job seekers, but the recruitment information exists in the web page, and the detailed requirements for job seekers are mostly semi-structured or unstructured text information in the form. At present, in the data analysis of recruitment websites, there is no need to analyze the skills of job seekers. More

importantly, it analyzes the relationship between salary and regional distribution, as well as between salary and education.

2.2 Web data mining

Web data mining technology is mainly to extract and regularize valuable data from a large number of complex web information. Through data transformation, data analysis and data modeling, the potential information in web page information is mined. We evaluate the current situation according to the data presentation results, and even make predictive judgments, which is of great commercial and scientific value.

2.2.1 Summary of Web data mining

Web mining aims to discover useful information or knowledge from the Web hyperlink structure, page content, and usage data. It is the application of data mining technology in the web environment. It applies data mining technology to the web, and finds the implied, unknown, potential application value and non-trivial patterns from a large number of Web document collections and the relevant data browsing in the site. It deals with static web pages, web databases, web structures, user usage records and other information (Han 2011). Through the mining of these information, we can get the information that can't be obtained only by text retrieval. The research object of Web mining is the web centered on semi-structured and unstructured documents. There is no uniform pattern for these data. The content and representation of the data are intertwined. The data content basically has no semantic information to describe, and only relies on HTML syntax to describe the data structure. In order to analyze and process the semi-structured data, web mining must be combined with its research methods. (Li 2008)

2.2.2 Web data mining classification

According to the different data categories used in Web mining, we divide them into three categories. They are web structure mining, web content mining and Web usage mining. (Liu 2011).

2.2.2.1 Web structure mining

It refers to mining information from the organization structure and link relationship of web pages. Web structure mining is to mine the structure between web pages. At present, it mainly aims at link structure pattern According to the link relationship between web files, we can mine useful information except web content. (Li et al 2013)

Traditional data mining does not perform such tasks because there is usually no link structure in a relational table. (Liu 2011) Through web structure mining, we can effectively find out the important web page link information. All in all, the main process of structure mining is to analyze the link relationship and page structure in detail, find out the useful information, and do a good job in link and relationship classification, so as to make the page clear.

2.2.2.2 Web content mining

Web content mining can be seen as the combination of Web Information Retrieval and information mining. It refers to the summary, classification, clustering, association analysis and trend prediction of a large number of document collections on the web. It is a process of extracting knowledge from web document content or its description, mainly divided into text information mining and multimedia information mining. The current research focuses on the use of word frequency statistics, classification algorithm, machine learning, metadata, part of HTML structure information discovery, hidden patterns between data discovery and generation of extraction rules, and the separation of concept and entity data from the page. (Li 2008) For example: job information introduction, user evaluation, etc. In this project, the technology widely used is web content mining. On this basis, the data mining algorithm is used to analyze the potential needs of users. This classification can be divided into text mining and multimedia mining. The former is the process of obtaining information from Web text documents and obtaining their hidden patterns through feature extraction. The latter is the collection and mining analysis of image, audio and video data on the network.

2.2.2.3 Web usage mining

Web usage mining refers to the discovery of user access patterns from Web usage logs, which record every click made by each user. Web usage mining applies many data mining algorithms. One of the key issues in Web usage mining is the pre-processing of clickstream data in usage logs in order to produce the right data for mining. (Liu 2011) Extract the content of interest to users and finally achieve user clustering by Mining user access information. Through the analysis of log files and users' browsing behavior, users' usage information can be found. But in this project I will not use this data mining classification

2.2.3 The general process of Web Data Mining.

In the process of Web data mining, we can roughly summarize the main process of its work.

Firstly, we search for data from the target web resource. In this stage, the original data needed in data mining analysis will be collected. But there are usually many redundant data and incorrect data in this data set. Therefore, we need to preprocess the data next. This stage mainly includes data cleaning, data noise reduction, dimension specification, discretization and other methods. Through preprocessing, the data quality will be greatly improved, which can effectively reduce the time and cost of data mining and analysis.

2.3 Data Preparation and Integration

2.3.1 Crawler search method

2.3.1.1 Breadth-first search

Breadth-first traversal is a search strategy widely used by crawlers. Its process is to have a URL queue first, pop up the URL in the queue, then extract the sub URL in the pop-up links, and put them back in the original URL queue to wait for pop-up. The URL that have been searched will be put into a table similar to the collection. Each time a new pop-up URL is processed, a judgment will be made first to see if there is a URL in the searched table. If there is one, skip it and carry out the next operation. The advantage of this operation is that it reduces the repetition rate without repeated crawling. The disadvantage is that the information update will not be timely after the page update, and each judgment will consume more resources and time. (Wang et al 2019)

2.3.1.2 Depth-first traversal algorithm

Depth-first traversal is often used in early crawler development. The general process is as follows: Firstly, find the first hyperlink URL from an HTML page. Then extract the URL, and then extract the first URL within the URL. It is always a single chain mode to dig deep until there is no next URL, and then return to the first HTML interface, and perform the same operation from the second URL. That is to say, there will be a complete single chain search before the next capture. When there is no next step for all links, the search ends. (Wang et al 2019) The advantage of this approach is that it can find all the deep URL, but it also has disadvantages, because once the

search starts, it may fall into a permanent deep search and cannot jump out. (Zhao et al 2017, Wang et al 2019)

2.3.1.3 Design and implementation of Crawler Based on scratch framework

Scrapy is a fast and high-level application framework written in Python for crawling website data and extracting structural data. At present, it is the most famous and widely used framework among all crawler frameworks. Scrapy is now capable of interacting with APIs in order to extract data. There are many reasons that contribute to the use of Scrapy as the WebCrawler. (Shi et al 2016)

Instead of designing a crawler framework from scratch, we can learn how to use Python's scratch framework simply and efficiently. (Liu 2018) Developers only need to develop a few specific modules to write a stable and efficient web crawler. (Sun et al 2019) Its detailed framework and operation process are shown in the figure. (Li et al 2017, Wang et al 2019)

2.3.1.4 The running process of the Scrapy Crawler

The Engine gets URLs to scrape from the Spider and schedules them in the Scheduler, as Requests. Then, the Engine asks the Scheduler for URLs to crawl, as Requests, and send them to the Downloader, passing through the Downloader Middleware. Once the page finishes downloading, the Downloader generates a Response and sends it to the Engine. The generated response contains the copy of the static HTML of the web page. The information extracted is stored in Items that are data holders of the framework. Then through the use of the Item Pipeline, the Items or data can be saved in suitable Formats including but not limited to CSV or SQL database. (Bassam 2016, Shi et al 2016)

2.3.1.5 Modules of the scrapy framework

1) Scrapy Engine: It is responsible for the transfer of regulatory data between modules in the system and calling corresponding functions to respond to specific events.

2) Scheduler : It is responsible for the unified management of all URL resources to be crawled. For example, insert the URL resources submitted by the receiving scrapy engine into the request queue. Then, the URL is taken from the queue and sent to the scrapy engine in response to the URL request from the scrapy engine.

3) Downloader Middlewares : It is responsible for delivering the URL request sent by the scrapy engine to the downloader module and the HTTP response sent by the downloader module to the scrapy engine.

4) Downloader : It is responsible for downloading the data on the web page and finally sending it to the crawler module through the scrapy engine.

5) Spider : It analyzes the data obtained from the downloader module, and then extracts the item or relevant URL resources.

6) Spider Middlewares : It is responsible for the input and output of the crawler module.

7) Item Pipeline : Process items extracted and sent by the crawler module through data cleaning, data validation, data persistence and other operations. (Cattell 2011, Li 2017)

2.3.2 Strategy of anti-crawler Technology

Nowadays, many websites prohibit crawlers from crawling data. Websites use headers, user behavior, website directory, data loading and other ways to anti crawl, so as to increase the difficulty of crawling. Thus, there are several strategies:

2.3.2.1 Set download_delay parameter

If the download waiting time is too long, the task of large-scale data grabbing in a short time will not be completed, and too short will increase the probability of being prohibited from crawling data. So we set DOWNLOAD_DELAY = 2 in settings.py (Li 2017)

2.3.2.2 Disable cookies,

This prevents crawler behavior from being detected by sites that use cookies to identify crawler tracks. So we need to set COOKIES_ENABLES = False. (Li 2017)

2.3.2.3 User-agent

User agent also refers to browser, including hardware platform, system software, application software and user's personal software preferences. (Chen 2016) Every browser and regular web crawler has a fixed user agent. Camouflage user agent can judge the category of the website visitors by violating rules. For camouflage browser and famous crawler, camouflage browser is more recommended. Compared with the

crawler, the browser has no fixed IP and can be anyone, while the crawler has a fixed IP. Camouflage browser can improve multiple user agents. Each time a request is sent, a user agent can be randomly selected to set the code according to the specific needs. (Liu 2019) We need to set DOWNLOADER_MIDDLEWARES = { ' scrapy.contrib.downloadermiddleware.useragent.UserAgentMiddleware':None,' HouseInfoSpider.spiders.rotate_useragent.RotateUserAgentMiddleware' : 400 , } in settings.py. (Li 2017)

2.3.3 Screening technology of Python based web crawler

A web crawler is a web robot. It is an important part of search engine. It is a program that can automatically extract the content of specific pages on the Internet. As for data mining, the first step is to determine the location of data storage, which can be successfully extracted when data is found. When browsing the web page, all kinds of elements on the web page are composed of data, which needs to be mined out from the web page. The workflow is summarized as follows: (Wang et.al 2019)

(1) Grabbing the page code through URL;

(2) Obtaining the useful data or URL on the page through regular matching;

(3) Processing the acquired data or entering the next round of grabbing cycle through the acquired new URL.

There are four ways to filter web crawlers in Python.

2.3.3.1 Regular expression grabs website data

Regular expressions, also known as normal representation, are often used to retrieve the text that conforms to a certain pattern. It first sets some special words and character combinations, and filters the expression through the combined "rule string", so as to obtain or match the specific content we want. It has the advantages of flexibility, logicality and functionality. It can quickly find the required information from the string through the expression. (Fu 2019). Specifying a Regex in a selector allows the matching the extracted data or string to the Regex logic and, thus filtering out the precise data. (Farooq et al 2016)

In Python, we can use the built-in RE module to use regular expressions. The most common find method we use in Python is the findall(). When we use the findall (), we can simply get a list of all the matching patterns.

Regular expressions with the same meaning and different writing methods are as follows: (Wang et.al 2019)

2.3.3.2 Xpath

Xpath is a language for finding information in XML documents, which is used to navigate through elements and attributes in XML documents. Using Xpath can easily locate the interested nodes in HTML documents. Lxml library is the third-party library of python, which supports the standard Xpath specification. A language that can navigate and extract tags from XML documents. In Xpath, there are five basic types of nodes: document (root) node, category, text, attribute, element. In order to distinguish between the root node and the category, the Xpath language stipulates that special symbols should be added before the name. The etree package needs to be imported from the Lxml library before use. (Wang et.al 2019) Besides navigating through the HTML tag hierarchy, Xpath selectors can also look at the content residing within the HTML tags, making them extremely useful when crawling a heavy content-based website. (Farooq et al 2016)

2.3.3.3 Beautiful Soup

Beautiful soup does not have its own parser. It passes in a parameter when building an object to specify the parser. Among these parsers supported, you can use your favorite one(Wang et.al 2019)

2.3.3.4 Urllib library in Python

Urllib is the library used for network requests in Python standard library The library has four modules: urllib.request, urllib.error, urllib.parse and urllib.robotparser。其 Urllib.request and urllib.error are two libraries that are frequently used in crawlers. The urllib.request module that people need to use to simulate a browser to send an HTTP request. The function of urllib.request is not only to initiate the request, but also to get the return result of the request. (Chi 2019)

2.3.4 Data storage

The crawled data can be stored locally or in the database.

2.3.4.1 Local storage

JSON files can be directly created in pipeline to write data, but the readability of JSON files is poor. Therefore, we can save the JSON file as a readable excel file after further processing. The extracted data table is a two-dimensional table structure.

The Pandas library is the main data processing tool in Python. With the help of Dataframe, we can store two-dimensional table structure. First, build a dictionary, then a two-dimensional dictionary priceall, save it as a Dataframe, then build an index of Dataframe with a list, and finally save it as an excel file using the to_excel() method of dataframe (Li et al 2018)

2.3.4.2 Database storage

The data model defined in scrapy establishes tables in the database, By using pymysql, a third-party module of python, to log in to the database, and executes SQL statements to insert data into the database. In order to achieve incremental crawling, a verification field needs to be created in the data table. (Li et al 2018)

2.3.5 Data cleaning

The main purpose is to correct errors, standardize formats, and eliminate duplicate and abnormal data. Data cleaning includes filling in missing values, identifying or removing outliers. (Si et al 2018) Data cleaning can be divided into two categories: repeated data cleaning and missing values imputation. In order to improve the speed and accuracy of data mining, it is necessary to remove the duplicate records in the data set. If two or more instances represent the same entity, they are duplicate records. In order to find duplicate instances, it is usually done to compare each instance with other instances to find the same instance. For numerical attributes in an example, statistical methods can be used to detect. According to the mean value and standard deviation value of different numerical attributes, confidence intervals of different attributes can be set to identify the records corresponding to abnormal attributes, identify the duplicate records in the data set, and eliminate them. Similarity calculation is a common method in the process of repeated data cleaning. By calculating the similarity of each attribute of records, and considering the different weight values of each attribute, the similarity of records can be obtained after weighted average. If the similarity of two records exceeds a certain threshold, the two records are considered to be matched; otherwise, the two records are considered to point to different entities. (Luengo et al 2016)

2.3.6 Handling missing values

Most of the missing data is due to the wrong operation of manual input, the need for confidentiality of some information or the unreliable data source, which makes the content of the data set incomplete. When the wrong data mining model is applied to

the front-end decision-making system, it will lead to serious deviation between the analysis results and the implementation decision. Missing values imputation can be regarded as incomplete data that can be ignored. It can ignore incomplete data directly by deleting attributes or instances. (Galar el at 2012) Missing values interpolation can also be a missing value interpolation algorithm based on filling technology. (Kong et al 2018) We use numerical information to fill in missing values. The simplest method is the average filling method. It takes the arithmetic mean of all complete data as the value of the missing data. The disadvantage of this method is that it may affect the original correlation between missing data and other data. If the missing values of large-scale data sets are all filled with the average method, because there are too many median values, more peak frequency distribution may mislead the mining results. (Galar el at 2012, Kong et al 2018)。Another method is to fill in the missing values by classification and clustering. Common missing value filling algorithms include EM algorithm (expectation maximization algorithm), MI algorithm (multiple imputation) and KNNI algorithm (k-nearest neighbor imputation) In the expectation maximum algorithm, the probability model is created to find the maximum likelihood estimate or the maximum posterior estimate. The success of the probability model depends on the unobservable hidden variable. (Gao et al 2011, Sotoca et al 2010)

2.3.7 Data cleaning tools

2.3.7.1 OpenRefine

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). (Anovana 2018)

2.3.7.2 Python

Wang (2019) performs data cleaning by using the Pandas library in Python. Pandas is a data analysis package built with Numpy and containing more advanced data structures and tools. Missing data is common in most data analysis applications. Pandas uses the floating-point value NaN to represent missing data in floating-point and non-floating-point arrays. There are four ways to deal with NA: dropna, fillna, isnull, notnull. You can use fillna to achieve many other functions, such as the average or median of the Series. The duplicated method of DataFrame returns a Boolean Series indicating whether each row is a duplicate row.

2.3.8 Data transformation and integration

In this stage, the pre-processed data is formatted and stored in the database for subsequent data mining. According to the designed data warehouse structure, the pre-processed data is loaded into the database. After that, it will be more convenient to add, delete, modify and query the collected data, and improve the efficiency of subsequent data mining. (Si et al 2018） Data integration is to merge the heterogeneous data in the multi file or multi database environment to solve the semantic ambiguity. This part mainly involves data selection, data conflict and inconsistent data processing. (Kong et al 2018)

2.3.8.1 Data transformation

Data transformation is to find the characteristic representation of data. It uses dimensional transformation or conversion to reduce the number of valid variables or find invariants of data, including normalization, switching and projection operations. Data transformation is to transform the data into a form suitable for various mining patterns. According to the data mining algorithm used later, we decide which data transformation method to use. Common transformation methods include: function transformation, using mathematical functions to map each attribute value; normalizing the data, scaling the attribute value of the data, as far as possible falling into a small specific interval Standardization not only helps to implement all kinds of classification and clustering algorithms, but also avoids over dependence on measurement units, and avoids the occurrence of weight imbalance. (Guan 2015, Kong et al 2018)

2.3.8.2 Data reduction

It is based on the understanding of the discovery task and the content of the data itself, looking for the useful features of the expression data that depend on the discovery target, in order to reduce the data model, so as to simplify the data as much as possible and promote the more efficient big data mining on the premise of keeping the original data as possible. (Kong et al 2018）

2.3.8.3 Dimensionality reduction

The technologies involved include feature selection and space transformations. The core of dimension reduction is to reduce the number of random variables or attributes. The purpose of eigenvalue selection is to obtain the attributes that can describe the key features of the problem. By removing irrelevant and redundant attributes, the machine learning process is faster and the memory consumption is less. The focus of quantity reduction is to reduce the amount of data and select a smaller data representation from the data set. (Kong et al 2018） The main numerical reduction techniques include log linear model, histogram, clustering, sampling and so on. Common algorithms include LVF (Las Vegas filter), MIFs (mutual information feature selection), MRMR (minimum redundancy maximum relevance), Relief algorithm. Space transformations is another way to reduce data dimensions. Popular algorithms include LLE (locally linear embedding), PCA (principal components analysis), etc. (Wang et al 2010, Kong et al 2018)

2.3.8.4 Instance reduction

This is now a very popular algorithm to reduce the size of data set is the instance reduction algorithm. While reducing the amount of data, it does not reduce the quality of knowledge acquisition. By removing or generating new instances, the data scale is greatly reduced. The technologies involved include instance selection and instance generation. Good instance selection algorithm can generate a minimum data set, remove noise data and redundant data, independent of the subsequent data mining algorithm. Common algorithms include CNN (Condensed Nearest Neighbour), ENN (Edited Nearest Neighbour), ICF (iterative case filtering), drop (decremental reduction by ordered projections), etc. Instance generation establishes various prototypes for instance generation, involving algorithms such as LVQ (learning vector quantification). (Perezortiz et al 2015, Kong et al 2018)

2.3.8.5 Discretization

Its purpose is to reduce the number of given continuous attribute values. Before discretization, we first estimate the scale of discrete data, then sort the continuous data, and then specify several split points to divide the data into multiple intervals. All continuous data falling in the same interval are mapped to the same discrete data by a unified mapping method (Prati et al 2015). According to the different identification methods of split points, discretization can be divided into top-down and bottom-up. According to whether to use classified information, it can be divided into two categories: supervised and unsupervised. At present, most discretization methods are divided into two directions: one is to discretize based on the importance of attributes, and the other is to map based on the resolution matrix. (Kong et al 2018) Common algorithms include: MDLP (minimum description length principle), CAIM (class attribute interdependency maximization), etc. (Angiulli et al 2007)

2.3.8.6 Imbalanced learning

When using supervised learning of machine learning to form data model, it is easy to produce huge priority differences in different types of data sets. Many standard classification learning algorithms often tend to ignore the priority class. (Bacardit et al 2012) Data pre-processing technology can avoid the imbalance of type distribution. The main methods are under sampling and over sampling. The former is to remove most instances as much as possible when creating a subset of the original dataset as data mining. The latter is to copy many of the same instances or create new ones during sampling. Among many sampling algorithms, the most complex and famous genetic algorithm is SMOTE (synthetic minority oversampling technique). (Kong et al 2018)

2.4 Data Analytics

2.4.1 Introduction

As Michael introduced, computer technology is breaking through every year, which makes it possible for us to collect and store a large amount of data at an effortless and low cost.(Berthold et al 2010) Pyne and other scholars believe that big data has different characteristics from traditional data sets. The world of big data is often nourished by dynamic sources such as intense networks of customers, clients, and companies, and thus there is an automatic flow of data that is always available for analysis.(Pyne et al 2016) Thus we can use this kind of data to capture complex

dynamic phenomena. Data analysis refers to the use of appropriate statistical analysis methods to analyse a large number of collected data, in order to maximize the development of data functions and play the role of data. Big data analytics is the process of exploring Big data, to extract hidden and valuable information and patterns. (Russom et al 2011) Data analysis application can be divided into descriptive analysis and predictive analysis. Descriptive analysis goes deep into historical data to detect patterns such as changes in operating costs, sales of different products, and customer purchase preferences. (Pusala et al 2016) Predictive analysis is based on historical data for future development and change. Experts assess the future by predicting trends, generating forecasting models and ratings. (Pusala et al 2016) Data analysis is a process of studying and summarizing data in detail in order to extract useful information and form conclusions. (Tao 2017) In the following sections, I will introduce four commonly used data analysis algorithms, which are classification, regression, association, cluster.

2.4.2 Classification

Classification predict the outcome of an experiment with a nominal target class attribute. Decision tree is the most popular classification method. There are two commonly used algorithms in decision tree: ID3 and C4.5. The ID3 (Iterative Dichotomizer 3) algorithm proposed by Quinlan in 1986 is the representative of the decision tree algorithm. After that, many decision tree algorithms are improved on the basis of ID3 algorithm. Based on the evolution of ID3 algorithm, Quinlan proposed C4.5 algorithm in 1993. Compared with ID3 algorithm, this algorithm has lower computational complexity and higher computational efficiency. C4.5 algorithm is an algorithm to construct decision tree classification rules, which is an extension of ID3 algorithm. ID3 algorithm can only deal with discrete descriptive attributes, while C4.5 algorithm can deal with continuous descriptive attributes. Classification is a supervised learning. (Quinlan 1986 1993)

2.4.3 Regression

Regression (especially logical regression) is also a popular method to solve the two-classification problem. Regression is the process of fitting a real-valued function of a given class to a given data set by minimizing some cost functional, most often the sum of squared errors. (Berthold et al 2010) Regression analysis is an algorithm that can be used for prediction. When we need to predict, we need to use this method when the variables we need to predict are numbers. Generally, regression analysis

can be divided into linear regression and nonlinear regression. Linear regression analysis is a statistical analysis method based on the least square principle, which is the optimal linear unbiased estimation under the statistical assumption. Regression from a statistical point of view, quantitative analysis of the law of change between variables, and through constant correction and adjustment, with quantitative relations to reflect this law. The quantitative output of continuous variable prediction is called regression.

2.4.4 Association

Association rule is to find out the frequent patterns between data from massive data. The research of association rules is originated from the research of the behaviour rules of supermarket shopping. The researchers find out the historical purchase data of customers and try to find out the combination of products when customers purchase so as to arrange the goods on the shelves reasonably. Association rules can predict any attribute, not just the class, and this gives them the freedom to predict combinations of attributes too. (Witten et al 2012) Among them, the most famous one is Apriori algorithm proposed by Agrawal et al. This algorithm uses support-based pruning technology to control the exponential growth of candidate sets. (Agrawal et al 1993) In Witten's research, weather data is introduced to show that in association rules, researchers are only interested in association rules with high coverage. In addition, Berthold proposed FP-growth and several other frequent item set mining algorithms all rely on the described basic recursive processing scheme. They differ mainly in how they represent the conditional transaction databases.
(Berthold et al 2010)

2.4.5 Clustering

Clustering techniques apply when there is no class to be predicted but rather when the instances are to be divided into natural groups. (Witten et al 2012) Bertold et al. classified clustering into Hierarchical Clustering, Prototype-Based Clustering and Density-Based Clustering. Hierarchical clustering is only suitable for small data sets. Prototype based clustering technology is very effective, even for larger datasets, the results can be interpreted. Density based clustering allows groups of arbitrary shapes, but the cost of this flexibility is that it is difficult to explain the obtained clusters. One of the most classical clustering algorithms is k-means algorithm. The algorithm divides the data set into clusters by setting the number of clusters K value. (Wang et al 2012) The specific operations are as follows: (Witten et al 2012)

1. Specifies the number of clusters K to search.

2. Randomly select k points as clustering centre. According to the Euclidean distance metric, all instances are assigned to their nearest clustering centres.

3. Next, calculate the centroid or mean value of each instance in the cluster, which is the "mean value" part. These centres of mass are considered as new centre values for their respective clusters.

4. Finally, repeat the whole process with the new cluster centre. The iteration continues until the same points are assigned to each cluster in successive rounds, during which the cluster centre is stable and will remain unchanged forever.

This clustering method is simple and very effective. As the value of K cannot be determined at the beginning, we choose to increase gradually from 2 to find the best value of K.

2.5 Data Analytical Tools

The following are two different tools for data analysis. At present, Python is widely used because it has a large number of toolkits to use. However, Weka is also a good data analysis tool because of its simple operation.

2.5.1 Weka

Weka (Waikato Environment for Knowledge Analysis) is an open source data mining platform researched by Waikato University, which integrates a large number of machine learning algorithms that can undertake data mining tasks, including data pre-processing, association rule mining, classification, clustering, and provides rich visualization functions. (Witten et al 2017) The process of data mining on Weka platform is as follows: First input the data set to be tested, then pre-process the data to be tested, and then place the processed data set in a learning scheme and analyse its results, or predict the unknown instance by the learned model, or place the data set in different learning schemes, In order to find out the best learning scheme, we evaluate the learning results of each scheme. (Chen et al 2008) For ease of use, firstly the CSV file is imported into Weka, and then it is exported in. arff format. Weka integrates nearly 50 algorithms. Id3 algorithm (in the simpleEducationalLearningSchemes package) and J4.8(implementation of C4.5) algorithm can be applied to decision tree. In association rules we can use FPGrowth

(frequent pattern trees) and Apriori. LinearRegression and logistic are available to realize regression analysis. SimpleKMeans can be used to do cluster analysis. (Witten et al 2017)

2.5.2 Python

2.5.1 Python Common Library

Python has Numpy, Pandas, Matplot lib, Scikit learn and other libraries with complete functions and unified interfaces, which can provide great convenience for data analysis. (Zhai 2018)

2.5.1.1 NumPy

This library, whose name means Numerical Python, actually constitutes the core of many other Python libraries that have originated from it. Indeed NumPy is the foundation library for scientific computing in Python since it provides data structures and high-performing functions that the basic package of the Python cannot provide. In fact, NumPy defines a specific data structure that is an N-dimensional array defined as ndarray. The knowledge of this library is revealed in fact essential in terms of numerical calculations since its correct use can greatly influence the performance of a computation. (Nelli 2015) Besides，Numpy also has the following features: (Zhai 2018)

1) It has functions that perform element level calculation on arrays and directly perform mathematical operations on arrays.

2)It can integrate C, C++ code into Python.

3) It can be used as a container for transferring data between algorithms.

2.5.1.2 Pandas

This package provides complex data structures and functions specifically designed to make the work on them easy, fast, and effective. This package is the core for the data analysis with Python. The fundamental concept of this package is the DataFrame, a two-dimensional tabular data structure with row and column labels.Pandas combines the high performance properties of the NumPy library to apply them to the manipulation of data in spreadsheets or in relational databases (SQL database). In fact, using sophisticated indexing it will be easy to carry out many operations on this kind of data structures, such as reshaping, slicing, aggregations,

and the selection of subsets. (Nelli 2015) In addition, Pandas can also be used for data preprocessing, such as data consolidation, data cleaning, data standardization and data conversion. (Zhai 2018)

2.5.1.3 Matplot-lib

This package is the Python library that is currently most popular for producing plots and other data visualizations in 2D. Since the data analysis requires visualization tools, this is the library that best suits the purpose. (Nelli 2015) Matplotlib consists of four parts: (Li 2018)

(1) The basic figure type of Matplotlib;

(2) Adjust the style and color of figure;

(3) Add notes to the drawing (including coordinate axis range, length width ratio or coordinate axis, etc.);

(4) Other complex figures.

2.5.1.4 Scikit-learn

Scikit-learn is a simple and effective data mining and analysis tool. It is based on Numpy, SciPy and Matplotlib, and encapsulates some common algorithms. Its main modules are data pre-processing, model selection, classification, clustering, data reduction and regression and other machine learning algorithms, which can help users to quickly build models in the process of data analysis, and the model interface is unified, which is very convenient to use. (Zhai 2018)

2.5.1.5 WordCloud

The system can import Wordcloud to generate the specified word cloud, and remove keywords without substantive statistical significance in the continuous optimization process. (Guo 2018)

2.6 Summary

In this chapter I have read a lot of literature, which lays a theoretical foundation for the following chapters. The literature shows that 51job and Zhilian recruitment occupy the main share of Internet recruitment in China recently, so they will be used for analysis when obtaining the content of Chinese Internet recruitment website. At

present, anti-crawler settings have been carried out for each website, so in the design of crawler, the way of adding request header will be used to simulate the real browser to visit the web page. This is a simple and general way to deal with the anti-crawler strategy of the website. In addition, in order not to cause load to the server, the access gap is set during access. There are three ways to get web page information mentioned in the literature, which are beautiful soup, Xpath and re (regular expression). In practice, I will choose the form of Xpath as the main and re as the auxiliary. This is because the browser I am using now has the Xpath helper plug-in, which can help me judge whether the Xpath code I wrote is correct or not to avoid errors in the actual running of the program. RE is a good filter for keyword matching. It can get what I need in the job information very well. It also points out the methods that should be used in the process of data cleaning. For example, the median can be used to fill in the default value. I will use the combination of Python, OpenRefine and Excel to clean the data. Python can detect duplicate data, OpenRefine can convert data format, and Excel can add columns. There are four main analysis methods for data analysis: classification, expression, cluster and association. I will use them in the actual analysis process because they cover descriptive analysis and predictive analysis. When using the analysis software, we will first consider the use of Weka, which is also mentioned in the literature. In addition, python analysis is also used as a reference, especially when drawing images.

3 Data processing

In the process of data processing, there are three stages: data collection, data cleaning and data integration. Data collection is to grab data from recruitment website by using Python scrapy framework, and store the data to local disk of computer. Data cleaning is the default value processing and outliers processing of the acquired data. Data integration is to unify data format for data analysis. These three steps are interrelated, and the specific operation steps are shown in the UML:

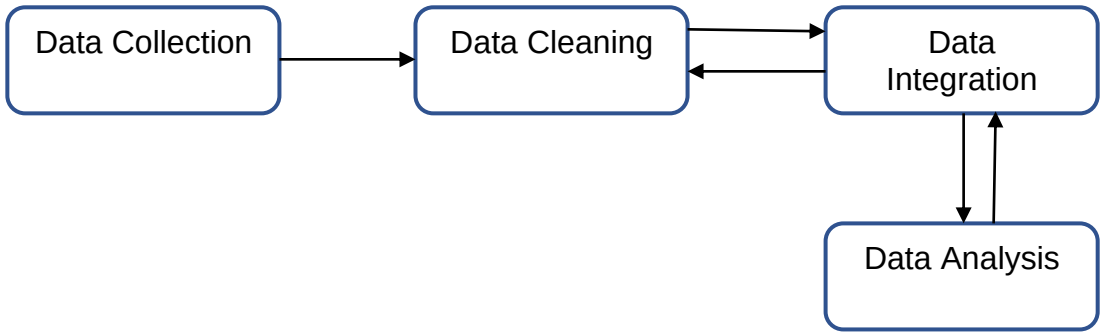


Figure 2 UML of the data processing

Firstly, I collect data, then clean the collected data, and then integrate the data. We can see that there are two-way arrows between data integration and data cleaning, and between data integration and data analysis. The purpose of this design is to find out that the data does not have the expected results when data analysis or data integration is carried out, so it needs to go back and reprocess.

3.1 Data collection

In this section, I use the summary framework for data collection. The data to be collected are from two recruitment websites, one is 51job in China and the other is job total in the UK. Although it's two different websites, the collection method is the same. What's different is that the Xpath used is different because of the different structure of the website. In addition, URLs are different. Therefore, only these two parameters need to be changed in the process of collection. The way of collecting data will be described in this section.

3.1.1 Scrapy based crawler design

3.1.1.1 Build scrapy framework

In order to gather the data from recruitment website page, Python is needed to program. At present, scrapy framework is a very popular framework for capturing page data. It integrates page grabbing, data storage and setting. In this framework, I don't need to do too much design, just need to set some specific parameters in it. Because the scrapy package is not Python's default package. First of all, you need to use the PIP instruction to wrap the necessary scrapy framework into the python installation directory.

After the scrapy toolkit is installed, execute the following instructions in Windows CMD to create a crawler directory (for the acquisition of page information of two websites is the same, so take the Jobtotal website as an example only):scrapy startproject jobtotal

After running the above instructions, a project folder of jobtotal will be automatically generated in the directory. According to the further prompt instructions, running ‘scrapy genspider crawl jobsite jobtotal.com’ to generate a crawler file named jobsite. The goal of this step is to limit the domain name of the crawler page to jobtotal.com. At this point, a complete framework of scrapy is completed. In this file, a self-defined crawler class named jobsite, will be generated according to the crawler template. This class inherits the methods and properties of the scrape.spider class. Here we can fully show the ease of use and high encapsulation of Python's scrapy framework. It makes the crawler project easier to start, reduces the repetitive and tedious preliminary work, and improves the development efficiency.

3.1.1.2 General solution to anti crawler of website in this project

With the development of big data technology, people begin to pay more and more attention to and demand the value of data itself, and page crawler tools in the network are also increasing. Web data has become one of the most important resources for everyone to compete with each other. At the same time, the website has begun to pay more attention to protecting its own data resources and preventing competitors from obtaining core data. Because the crawler robot can visit the website in a large quantity in a short time, it is easy to cause the overload of the website server. The worst-case scenario is that multiple people visiting the site in large

quantities at the same time will cause the server to crash. Therefore, the key goal of anti-crawler is to prevent mass access to website information.

In the HTTP request, the header information of the data request body will carry the identification information of the browser during the normal browsing process of the user through the web browser. In order to simulate the request in the crawler, I visited the page to be visited through Chrome browser firstly, and put the intercepted header information into setting.py. Setting.py is the global configuration file of the crawler, which will be loaded automatically at runtime by setting the content. For example, in the user agent of headers, the information displayed is as follows : 'Mozilla/5.0(WindowsNT10.0;Win64;x64)AppleWebKit/537.36(KHTML,likeGecko)Chrome/79.0.3945.130Safari/537.36'. It shows that the request was initiated through Chrome browser with additional information such as cookies. Through such head processing, the effect of camouflage browser request is achieved. In the process of crawling data, the web server will recognize the request as a regular request of the browser. At the same time, due to the number of requests, you can further prepare a group of browser header list. Each subsequent request is randomly assigned with different request headers, which makes the website think that it is initiated from multiple browsers when it recognizes the large number of requests of crawler items.

In addition, in order to avoid the anti-crawler processing caused by high-frequency requests, in the scrapy crawler project, I also increase the interval time of multiple requests by setting the request frequency in setting.py to avoid abnormal request behavior. Due to the setting of request interval, although it will affect the efficiency of requesting web pages, it can avoid unnecessary load on the other server. As a result, every request I make is successful. This operation has been highly encapsulated in the scrapy project and can be set globally in setting.py. The specific settings are as follows:

Set download delay time: DOWNLOAD_DELAY = 1

Set the maximum request delay time: AUTOTHROTTLE_MAX_DELAY = 60

Through the above design of anti-crawler strategy, I found that both Jobtotal and 51job websites can successfully solve the anti-crawler strategy.

3.1.1.3 Gathering information from the website

For jobtotal and 51job, I use the same strategy to capture information. First of all, using keyword search for positions, in order to ensure the comparability of data in this project, four keywords are selected. They are Python, Java, JavaScript and PHP. The purpose of choosing them is that Python and Java are the two most commonly used object-oriented languages for programmers recently. They are characterized by a wide range of job applications and career opportunities. The latter two are two common programming languages in the process of making web pages. After searching for keywords, the website will display the position information in the form of a list. However, the amount of information obtained in this list is not particularly detailed, and more job information is still reflected in the position details page. This is also in line with the normal operation of browsing a position information in the browser. Therefore, in the position list page, I choose to get the URL of the details page of each page first and store them in the container first. After obtaining the URLs of all position information detail pages of this list page, scrapy will traverse and visit the detail page again, and the interval of each visit is 1 second.

In this project, I use the Xpath method to read the web page information. Xpath is a path language used to determine the information of XML structured documents. In the process of crawling web data this time, Xpath will read the information of HTML tree structure. According to the difference of node information structure, it is mainly divided into three categories: element node, attribute node and text node. According to the specified path information, Xpath looks for the corresponding node in the DOM structure of HTML. In this way, the path is used to locate the required page node, through which the useful information of the node can be obtained, and then the required source data can be obtained by sorting out the information.

Table 1 The Xpath statement corresponding to the Jobtotal collection attribute

jobname	//h1[@class="brand-font"]/text()
jobissue	//li[contains(@class,"date")]/div/span/text()
company_location	//div[contains(@class,"col-xs-12 col-sm-7")]/ul
joblocation	//li[contains(@class,"location")]/div
salary	//li[contains(@class,"salary")]/div/text()
company_name	//li[contains(@class,"company")]/div/a/text()
job_content	//div[@class="job-description"]/ul[1]
job_description	//div[@class="job-description"]/ul[2]
job_style	//li[contains(@class,"job-type")]/div/text()

Table 2 The Xpath statement corresponding to the 51job collection attribute

jobname	//div[@class="cn"]/h1/text()
company_name	//p[@class="cname"]/a[1]/text()
salary	//div[@class="cn"]/strong/text()
joblocation	//div[@class="cn"]/p[2]/text()[1]
experience	//div[@class="cn"]/p[2]/text()[2]
education	//div[@class="cn"]/p[2]/text()[3]
job_issue	//div[@class="cn"]/p[2]/text()[5]
company_style	//div[@class="com_tag"]/p[1]/text()
company_scale	//div[@class="com_tag"]/p[2]/text()
job_industry	//div[@class="com_tag"]/p[3]/@title
job_classification	//div[@class="mt10"]/p
job_content	/html/body/div[3]/div[2]/div[3]/div[1]/div

In practice, add '.get (default = '')' after the Xpath statement, otherwise the obtained information will be empty.

After analyzing the required data information and how to parse the node information, I started to write spider module of the scrapy crawler project. Since the custom class inherits to the spider class of scrapy, the class will define the logical judgment of getting the initial action of crawling and whether to follow up the crawling or not. The specific steps are as follows:

1. Select the initial page crawled, and fill in the URL value of the initial page in the start_url attribute

2. The parse function will get the data returned by the specified URL request. In this method, the detail page connection will be realized. The detail page data will continue to be requested based on the existence of valid links. In this request, the callback function parseDetail will be specified to specify its return data, which is used to process the resolution operation of the returned data. In addition, the URL splicing of the next list page will be implemented in this function. Through many experiments, I found that the URL of the next page in the 51job web page is only related to the 111th number in the start_url, so I only need to replace this character each time. In the jobtotal web page, I only need to replace the last character. The last step is to set the page turning times by the maximum number of pages observed, so that the program can realize the automatic page turning function.

```python
def parse(self, response):
    s1 = response.url[:110]
    s2 = response.url[111:]
    num = int(response.url[110:111]) + 1
    print(num)
    new_url = s1+'{}'.format(num)+s2
    selectors = response.xpath('//div[@class="el"]')
    for selector in selectors:
        next_url=selector.xpath('./p/span/a/@href').get()
        if next_url:
            yield scrapy.Request(next_url, callback=self.parseDetail)
    if num<10:
        yield scrapy.Request(new_url, callback=self.parse)
```

Figure 3 The code of getting the next page URL

3. The parseDetail function will get the return data of the detail page, and extract the required data information in the return value response through the Xpath path shown in the table above. After simple processing, it is bound to the defined item class property and returned to the pipelines component for the next process processing.

```python
items={
    'html_url':html_url,
    'jobname':jobname,
    'company_name':company_name,
    'salary':salary,
    'salary_min':salary_min,
    'salary_max':salary_max,
    'salary_method':salary_method,
    'salary_style':salary_style,
    'joblocation':joblocation,
    'experience':experience,
    'education':education,
    'job_issue':job_issue,
    'company_style':company_style,
    'company_scale':company_scale,
    'job_industry':job_industry,
    'job_classification':job_classification,
    'job_content':job_content,
    'job_responsibility':job_responsibility,
    'job_demand':job_demand
}
yield items
```

Figure 4 Define the item

Because the data content structure of 51job website is relatively uniform, so the data is processed simply when it is gathered. In this process, since the salary part of the collected data is a string of interval segments, I split them to obtain the minimum salary, the maximum salary, and the payment method of salary (pay by year, pay by month, pay by day).

```
if (salary):
    sal = salary.split('-')
    # print(len(sal))
    if len(sal) == 1:
        salary = salary
        salary_min = sal[0]
        salary_max = float(sal[0].split('/')[0][:-1])
        salary_method = sal[0].split('/')[0][-1:]
        salary_style = salary[-1:]
    else:
        salary = salary
        salary_min = float(sal[0])
        salary_max = float(sal[1].split('/')[0][:-1])
        salary_method = sal[1].split('/')[0][-1:]
        salary_style = salary[-1:]
else:
    salary = None
    salary_min = None
    salary_max = None
    salary_method=None
    salary_style = None
```

Figure 5 Extract the number and payment method in salary attribute

In addition, it is not difficult to find in the job description part that most job contents are divided into job responsibilities and job contents. By using regular expressions, the job responsibilities and the job demand can be distinguished from the job contents.

```
a = re.split(r'(工作要求：|任职要求：|任职资格：|岗位要求：|岗位职责：|工作职责：|岗位职能：|我们期望：|工作内容：'
             r'|职位描述：|职位要求：|岗位职责:|招聘要求：|任职资格:|职责要求：|主要职责1)', job_content, 3)
job_demand=''
job_responsibility=''
if len(a) == 5:
    if a[1] in ("岗位职责：", "工作职责：", "岗位职能：", '工作内容：', '关于职位：','任职资格：',
                '职位描述：','职位要求','岗位职责:','招聘要求：','任职资格:','职责要求：','主要职责1'):
        job_responsibility = a[2]
        job_content = ""
    elif a[1] == 0:
        job_demand =  a[2]
        job_content = ""
        #print(job_demand)
    if a[3] in ("岗位职责：", "工作职责：", "岗位职能：", '工作内容：', '关于职位：','任职资格：'
                ,'职位描述：','职位要求：','岗位职责:','招聘要求：','任职资格:','职责要求：','主要职责1'):
        job_responsibility =  a[4]
        job_content = ""
    else:
        job_demand =  a[4]
else:
    pass
```

Figure 6 Filter position responsibilities and application requirements in job content

3.1.1.4 Local storage of web page information

The data gathered from the above methods can be better used only by storing it in a local file. In the operation of scrapy, if there is no local output, all data will be displayed in the CMD one by one, which is of no value to my next analysis work. The scrapy framework has its own way of data output. When running scrapy, add -o filename.csv, and I can directly output the web page information obtained in CSV format. But this method is easy to cause data redundancy. For example, when I search for the position whose keyword is python, some positions may have the keyword Java. Then when I search for Java, the recruitment information will also have the keyword python. But they are actually the same recruitment message from the same company. The scrapy framework also provides a way to store information in a database.

Through the spider module, the requested data has been returned to the pipelines module. In this module, data integration, database connection, transfer and other operations will be carried out. According to the processing of the request data in the previous section, the data information of a single request has been stored in the custom properties of items. After confirming the data structure, connect to the database. In the module, pymysql package is introduced, through which the local MySQL database is connected for the transfer operation. The local database can be added by simple code, and the data documents can be transferred. The specific process is as follows:

1. Import pymysql package and global configuration

```
import pymysql
from scrapy.exceptions import DropItem
```

2. Connect local MySql database

```
connect = pymysql.connect(
    host='localhost',
    user='root',
    passwd='123456',
    charset='utf8',
    db='liepin',
    use_unicode=False)
```

Figure 7 Connect to MySql Database

3. In the process_item function of pipelines, format the acquired item data

```
data = {'html_url':item['html_url'],
        'jobname': item['jobname'],
        'jobissue':item['jobissue'],
        'joblocation':item['joblocation'],
        'company_location':item['company_location'],
        'salary':item['salary'],
        'company_name':item['company_name'],
        'job_content':item['job_content'],
        'job_description':item['job_description'],
        'job_style':item['job_style']}
```

Figure 8 Get data from item

4. Execute db_distinct function to determine whether it is the data written previously. In this step, we need to complete the data deduplication work first to ensure that each piece of data recorded in the MySQL database is unique. The deduplication method is based on the URL of the position details page. The reason why this is done is that there is only one position details page for each position in a company.

```python
def db_distinct(self, html_url):
    connect = pymysql.connect(
    host='localhost',
    user='root',
    passwd='123456',
    charset='utf8',
    db='liepin',
    use_unicode=False)
    cur = connect.cursor()
    sql = 'select * from jobsite where html_url ="{}";'.format(html_url)
    cur.execute(sql)
    data = cur.fetchone()
    cur.close()
    if data == None:
        return True
    else:
        return False
```

Figure 9 Retrieve if previously stored

5. Save the formatted data into MySQL database by using SQL statements. Execute db_distinct function, and write a try function to determine whether the data is written successfully.

```python
try:
    re = self.db_distinct(item['html_url'])
    if re:
        try:
            cur.execute('insert into jobsite(html_url,jobname,jobissue,joblocation,company_location,salary,company_name'
                        ',job_content,job_description,job_style)values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)',
                        [html_url,jobname,jobissue,joblocation,company_location,salary,company_name,job_content,job_description,job_style])
            connect.commit()
            print('sql insert success')
        except:
            raise DropItem('sql run error')
    else:
        raise DropItem('data exists')
except:
    connect.rollback()
    cur.close()
    connect.close()
```

Figure 10 The code of storing in the MySql database

Therefore, the data gathering from the website is stored into my local database. Then, the data can be exported from the Mysql Workbench 8.0 which is a Mysql Database visualization tool.

3.1.1.5 The Nature of two Datasets

Table 3 The nature of 51job

51job nature		
Attribute Name	Value	Code description
_id	numeric	No practical significance is only automatically generated when the database stores data
html_url	string	No practical significance is only automatically generated when the database stores data
jobname	string	position name
company_name	string	company name
salary	string	The original salary display form
salary_min	numeric	The minimum salary for the position
salary_max	numeric	The maximum salary for the position
salary_method	string	Salary payment method
salary_style	string	Salary counting method
joblocation	string	work place
Internship	string	Determine whether the position is only available
education	string	Academic qualification
experience	string	work experience
company_style	string	Company Size
company_scale	string	Company Type
job_industry	string	Industry to which the job belongs
job_classification	string	Job type
job_content	string	Descriptive information of the position
job_responsibility	string	Job responsibility of the position
job_demand	string	Ability requirements for the position

Chinese job site 51job has a total of 20 attributes. Some of the data is useful in the process of data collection but it is meaningless for data analysis.

Table 4 The nature of jobtotal

jobtotal nature		
Attribute Name	Value	Code description
html_url	string	No practical significance is only automatically generated when the database stores data
jobname	string	position name
jobissue	string	Recruitment information release time
Location	string	work place
joblocation	string	work place
salary	string	The original salary display form
min salary	numeric	The minimum salary for the position
max salary	numeric	The maximum salary for the position
salary_type	string	Salary payment method
welfare	string	Extra rewards received by employees in addition to salary
company_name	string	company name
job_content	string	Descriptive information of the position
job_description	string	Descriptive information of the position

Jobtotal has a total of 13 different attribute names. However, due to the poor data structure of the website. In fact, location and joblocation both indicate geographic location, but there are two different ways of expressing this information on the

website. Job_description is a supplement to job_content. The reason for this is that the data structure of the website does not have a unified standard in terms of job information description like the Chinese website.

3.1.1.6 Summary

Through the data set obtained, we can find that there are more attributes and the total number of data sets in China is also much more, so this data set has more reference significance for the process of data analysis. And too few data attributes and numbers on the UK recruitment website may not be conducive to the next analysis process. Both of these data sets have null values, so the next stage of data cleaning is needed.

3.2 Data clean

Through the use of scrapy framework to mine online recruitment data, a large amount of recruitment data was obtained. In this chapter, the previously collected data will be cleaned and the wrong data collected will be corrected. In the cleaning process, OpenRefine and numpy and panda libraries in Python will be used for operation.

3.2.1 51job data clean

Through the previous information mining, a total of 10253 Java-related professional information and 1492 Python-related job information were obtained. First, I exported the data stored in the local database to the computer desktop in the form of csv format. Then use the OpenRefine tool to read the data set. It is not difficult to find through visual inspection that the biggest problem with the data set is the reversal of the two data records of educational background and work experience. One of the reasons for this is that the data displayed on the website does not have a strict and uniform structure, and the data is captured using a general method in the process of data capture. The types of dirty data in the two columns can be divided into three categories: missing data, opposite data records, and completely incorrect data content. For data whose educational background is opposite to the storage location of work experience, my job is only to swap them left and right.

3.2.1.1 Data unification

Using text facet to view salary_style and salary_method in OpenRefine, we can find that there are three ways to display salary, they are paid by day, paid by month and

paid by year. There are two different methods for the settlement of salary, which are thousands and tens of thousands. In order to make the data unified for subsequent analysis, here the data will be unified as annual payment and the unit is "ten thousand". The specific operation is as follows. Use the filter in Excel to view salary_style and salary_method, and ignore the default value at this time. For the salary payment method is monthly payment and the settlement form is "ten thousand" data, I multiply these data by 12 and then multiply by 10000 to achieve the result of annual payment. For the salary payment method is monthly payment and the settlement form is "thousand", I multiply these data by 12 and then by 1000 to achieve the result of annual payment. Regarding the salary payment method as the daily payment data, I simultaneously multiply these data by 12 and then by 22 to achieve the result of annual payment. The reason for taking 22 here is because the number of working days per month is calculated as 22 days. For the salary paid annually, because the settlement units are all ‘ten thousand’, they are simply multiplied by 10,000 to process. In addition, use text facet in OpenRefine to view company_scale and job_industry. In the process, you can find that they have 228 and 319 different distributions, respectively. However, many of the data are redundant due to too detailed classification, so the data subdivided in the major categories are unified into major categories. Finally, I used text facet to check the location data in OpenRefine. I also found that many cities are divided too carefully. For example, some work locations are shown as' Shanghai-Baoshan District '. In this case, they are modified to' Shanghai '. Other city names with similar situations are handled in this way. When using OpenRefine to process location, company_scale and job_industry, all data sets are translated at the same time, and they are translated from the original Chinese to English. The reason for this operation is that when Weka software is used for data analysis, it cannot perform a good encoding analysis on Chinese characters. Therefore, it will lead to the situation that all Chinese characters are garbled after being imported into Weka.

3.2.1.2 Default value processing

Through the panda library in python to view all the data, we can find that whether it is java recruitment data or python recruitment data, there is a default value in the salary part. There are 124 data missing in the java part and 32 data missing in the python part. In addition, there are cases where there are default values at the time of posting of job information and detailed descriptions of occupations, but no treatment will be

done for these two parts. The first reason is that the release time of job information is meaningless for this analysis. Second, the job description is unique and unique, so it is not filled. Next, the OpenRefine tool will also be used to fill in the missing data.

	jobname	company_name	salary
count	10253	10253	10129
unique	3251	8757	418

	jobname	company_name	salary	salary_min
count	1491	1492	1460	1460.000000
unique	672	1274	176	NaN

Figure 11 Overview of the raw data

By checking, we can find that there is missing ‘salary’ data in every region. For a city with a large amount of data itself, the method used for the default value filling is to first use text facet in OpenRefine to view the salary distribution of the city. For example, using text facet to view shanghai and salary_max at the same time, it can be seen that there are a total of 127 default data. Due to the large amount of data in Shanghai, the mode 1.5 in salary_max is selected to fill the default data and use the same The way to fill the salary_min data at the same time. For cities with a default value greater than 10, the above method is used to fill in the default value. For cities with a default value less than 10, the filling method is different. At this time, the overall sample size of the city is relatively small and the sample data is scattered, so in this case, the significance of using mode filling is not so obvious. In addition to this, this operation may also give wrong guidance to the final result. At this time, choose to operate on the data in Excel. First, calculate all the average salary in Excel without including the default value, and then use this average value to fill the salary_min and salary_max of the remaining cities.

3.2.1.3 Outliers processing

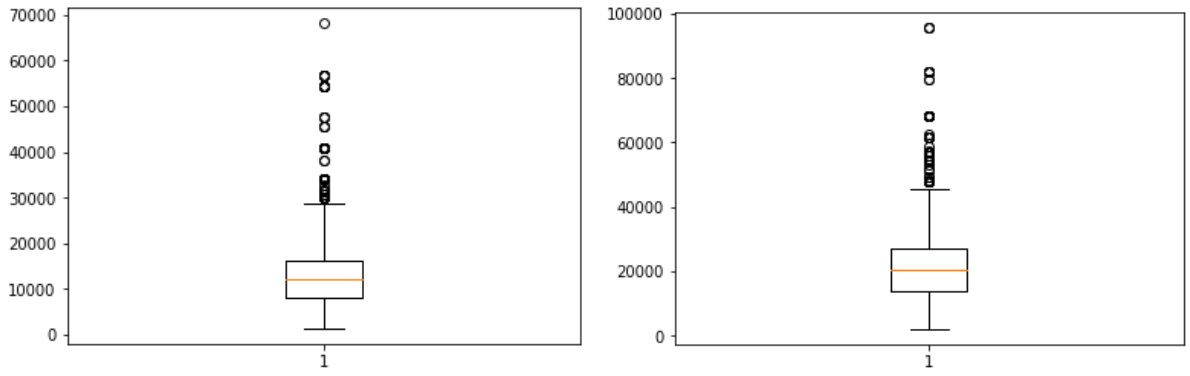


Figure 12 Show outliers in boxplot the left one is Java dataset and the right one is Python dataset

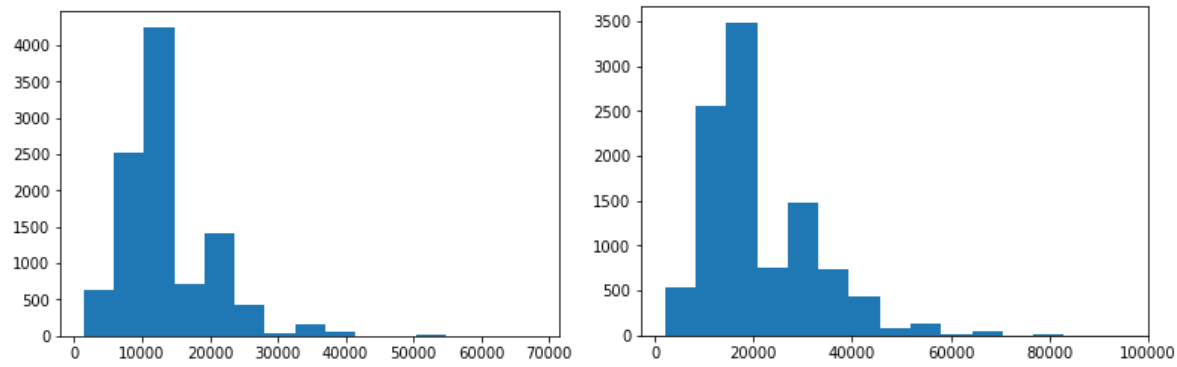


Figure 13 Salary distribution of Java data set and Python data set

Through the use of box plots in Python, we can find that some of the values are considered outliers, but fortunately, the values of all lattices do not appear to be less than 0. By using the histogram, we can see the specific distribution of salary more intuitively. Because it is the distribution of salary, we should allow a value greater than zero to appear here. Therefore, no other changes are made in this section.

3.2.2 UK recruitment data clean

2036 Java related job information and 404 Python related job information were obtained through the previous web crawler. Because the data structure of the website is relatively simple, there are only differences in the attribute of address. Therefore, in the process of data acquisition, two different kinds of Xpath are used to obtain location data.

3.2.2.1 Unification of data

By using the text facet in OpenRefine to view the salary payment types, you can find that there are three payment methods similar to China's recruitment data, which are daily payment, monthly payment and annual payment. In order to unify the data format for subsequent analysis, here I will pay all the salaries on an annual basis. Secondly, in order to visually observe the data better, replace k with '000' for all salaries with k in the data. For salaries paid on a monthly basis, I will multiply all of them here by 11 to convert to annual salary. The main reason for multiplying by 11 here rather than by 12 is that I found that the overall salary level after the monthly payment is multiplied by 12 is relatively high, so this phenomenon will cause the overall annual salary to be increased. For salaries paid on a daily basis, I will multiply them by 20 to indicate working 20 days a month, and then multiply by 11 for the same reason as before. Here I found that the daily salary have generally increased

greatly after being replaced by annual salaries. Here I analyze one of the main reasons is because some projects are only short-term labor workers demanded by enterprises. In this way, to recruit the company's short-term needs in part of the business and conduct a recruitment activity. In addition, for the unification of the company's location. First, in Excel, job_location and company_location are merged into location. Because the website has only two types of storage for the location during data acquisition, there will be no data coverage during the merge process. In addition, I also found that most of the location information is presented as 'city, county where the city is located'. In order to prevent the data from being divided too carefully at this time, the data is separated according to the characters ‘,’ in Excel, so as to achieve the purpose of keeping only the county name.

3.2.2.2 Default value processing

By using python to view the data of each attribute, you can see that there are default values in salary, salary type and job description. Job description Because of the disorganization of information structure in the early stage of the website information acquisition process or the company's original failure to provide job information, the defect is caused. Therefore, for the job description and salary type part, no adjustment will be made here. The way of filling the salary attribute is mainly similar to the previous filling method. First create a project in OpenRefine to view the data. Then use text facet to view the entire data set, select the blank part of the data, and then use text facet to view the location attribute. My strategy here is roughly the same as the data filling method of Chinese recruitment websites. For cities with a large amount of data, such as London, I use the mode of the data in their respective attributes of the highest salary and the lowest salary to fill in the data. For cities with a small amount of data, I calculated in Excel the average value of the data excluding the default value. Finally, use this average to fill in the data. At the same time, I also found that in the original salary attribute, many companies present this attribute data in the form of salary plus benefits. So in Excel, I separate all the numbers and letters in it to list the company benefits in a new attribute. So far, the default value of the salary for the UK recruitment website has been filled.

3.2.2.3 Outliers processing

Similarly, I draw box chart and histogram in Python to see if there is any abnormal value. The main judgment basis is that all salaries must be greater than zero, so all

non positive integer data is not desirable. Fortunately, no data in this sample is less than or equal to zero. It's acceptable for too large a value. After all, some industries offer very high salaries, which is not nonexistent in reality.

3.3 Data Integration

In order to facilitate the subsequent data analysis in Python or Weka. I need to format the data in a unified format. Here I will generate two files in CSV format, one file is all numeric value and the other file is nominal value. This process will be implemented in OpenRefine using the transform function. For the data of 51job, temporarily delete job_content, job_responsibility and job_demand at this time. Because these three columns of data contain a large number of Chinese characters, not only will the computer memory pressure be displayed when reading the data, but it will also display garbled characters in the program after reading.

Because of the cities of 51job data set are divided too detailed, they are divided into four categories: First tier city, Other First tier city, Second tier city and Third tier city. The division is based on the city's position and economic strength in China. In the two properties of job classification and job industry, their original variables are 62 and 46 respectively. In order to simplify their scale, the first 13 and 10 variables are reserved according to their frequency. When all the nominal values are converted into numerical values, they are numbered from 1 according to their frequency. In other words, the higher the frequency, the smaller the value. See the appendix for specific conversion. Finally, according to the size of salary, the data set is divided into two categories. Data sets with an average salary of less than £ 11000 are classified as low income. Figures with an average salary greater than £ 11000 are classified as medium to high income. The average salary here is half of the maximum and minimum salary for each data.

At the same time, for the UK recruitment data, the positions with an annual income of less than 20000 pounds are also marked as "low", the positions with an annual income of between 20000 pounds and 40000 pounds are marked as "medium", and the positions with an annual income of more than 40000 pounds are marked as "high". At this point, the division of salary grade is completed.

4 Data Analysis

4.1 Visualization

In this section I use Python to draw graphs in Jupyter Notebook to explore the factors related to the number of jobs and the factors related to salary. In order to the relationship I draw bar charts and box charts. Three packages in Python are used here, which are matplotlib, pandas and numpy.

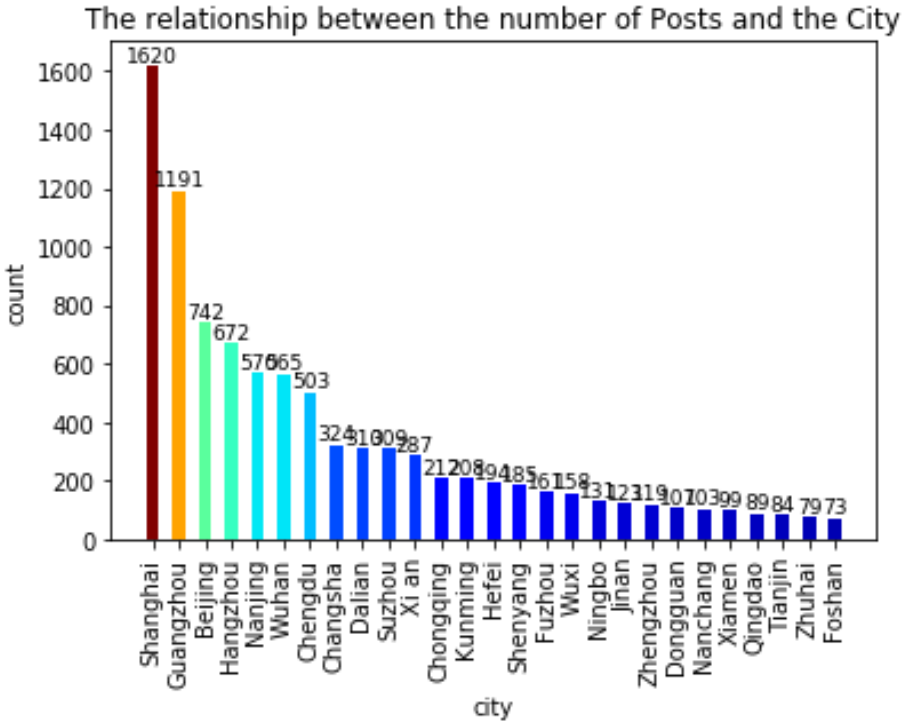


Figure 14 The relationship between the number of posts and the city

Due to space constraints, this figure shows only the number of Java jobs in the top 25 cities. Through the bar chart, we can directly find that the demand for jobs is closely related to the size of the city. The demand of municipalities directly under the central government is greater than that of provincial capital cities, which is greater than that of general prefecture level cities. The demand for Java jobs in Shanghai is far ahead of other cities in China. The greater the demand, the greater the opportunity for the city for those with Java skills, and the better for the future development of a programmer. Although the captured data is only a data record of more than one month, it is enough to reflect the demand of local enterprises.

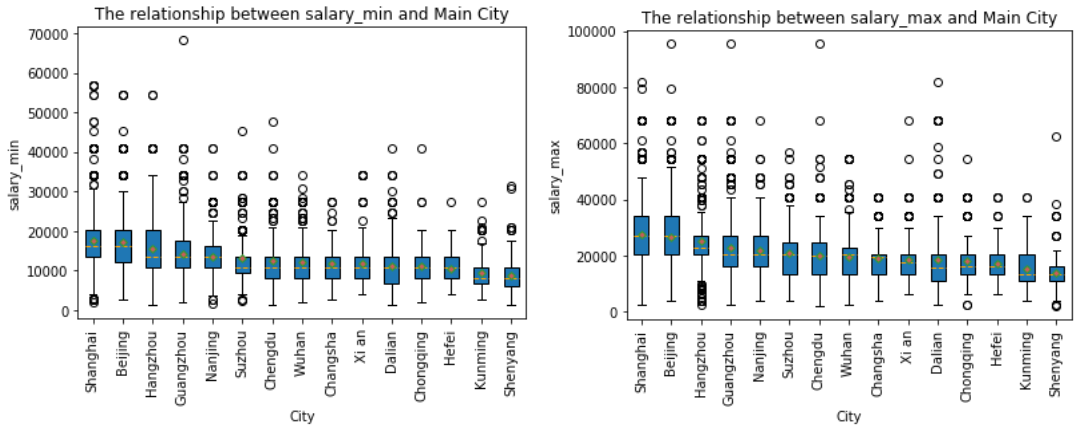


Figure 15 The relationship between salary and main cities

Through the above relationship between the number of jobs and cities, let's take a look at the relationship between salary and cities. Similarly, due to space, only the first 15 cities are showed here. For better and more comprehensive observation, the box diagram is used here. The general situation is similar to the distribution of previous positions. However, Guangzhou dropped from second to fourth. So while the city's demand is large, its overall salary is not that high. In addition, its job demand is even 1.5 times or higher than the latter. Hangzhou's salary level is second only to the first two municipalities directly under the central government, and its salary is more decentralized. There are outliers in every city, which should refer to those high-end technical occupations. Although Hangzhou's super high salary is not as high as Beijing and Guangzhou's, its number is the largest. So it can be said that Hangzhou welcomes more high-end technical talents to take root here.

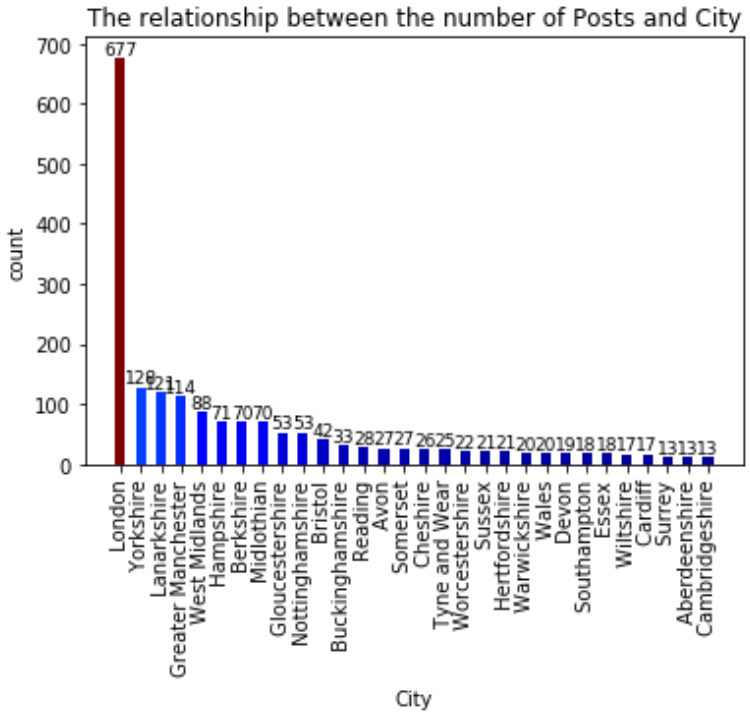


Figure 16 The relationship between the number of posts and city in UK

Through the data in the figure, it can be found that Java jobs in the UK have strong aggregation. Jobs are mostly concentrated in the London area. The number of Java jobs in London is five times that of the second place. We can see the agglomeration effect of super big cities here, so if we want to have better development in the UK, we should go to London, where there are more opportunities and better prospects for development.

Next, let's look at the relationship between experience, education and the city.

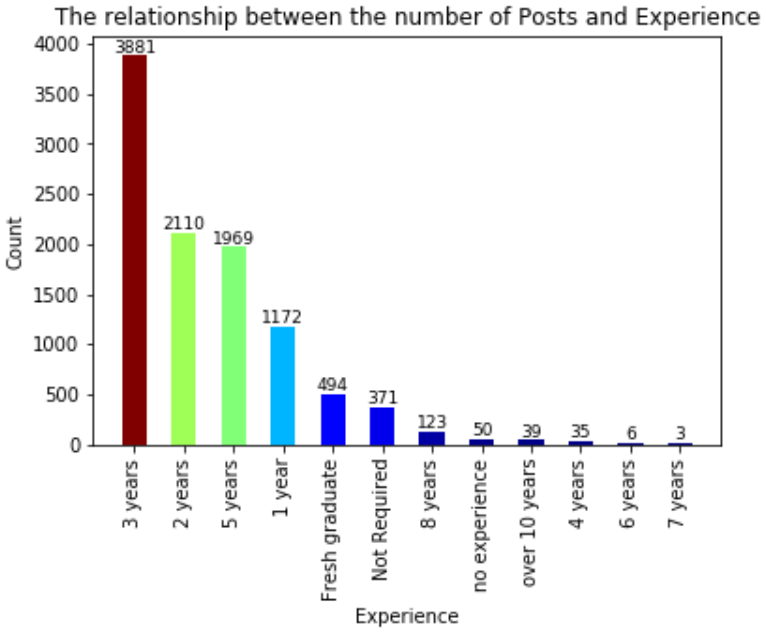


Figure 17 The relationship between the number of posts and experience

On the whole, the most popular programmers are those with three years of project experience, almost twice as many as those with two years of experience and five years of experience. Therefore, it can be explained that more than three years of project experience is more favored by the company. For fresh graduates or inexperienced people, the number of jobs offered by the company can not be too many or too few, but it can also be said that 51job is a tool for such people to find jobs. At the same time, for the young people who lack working experience, working experience is hard strength in the current large employment environment. Do not change a job casually unless you are already strong enough to compete with others. The demand for more than six years' experience is relatively small, which also reflects people's reluctance to change a new company in a relatively stable working environment.

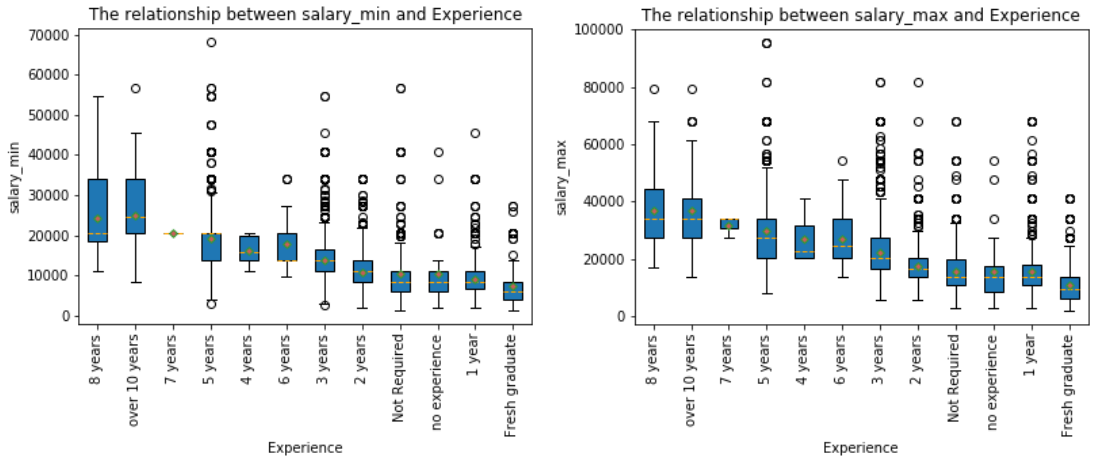


Figure 18 The relationship between salary and experience

The box chart is used here to more clearly reflect the relationship between salary and work experience. Through the box chart of minimum salary and maximum salary, we can find that the salary of programmers increases with the increase of working experience. Therefore, work experience is positively related to salary. For a company, it is sure to hope that the higher the working experience, the better, but in order to get this kind of employees, it will pay a considerable price. For a newcomer, the opportunity of good study is as important as the salary. There are many outliers in the each stage.

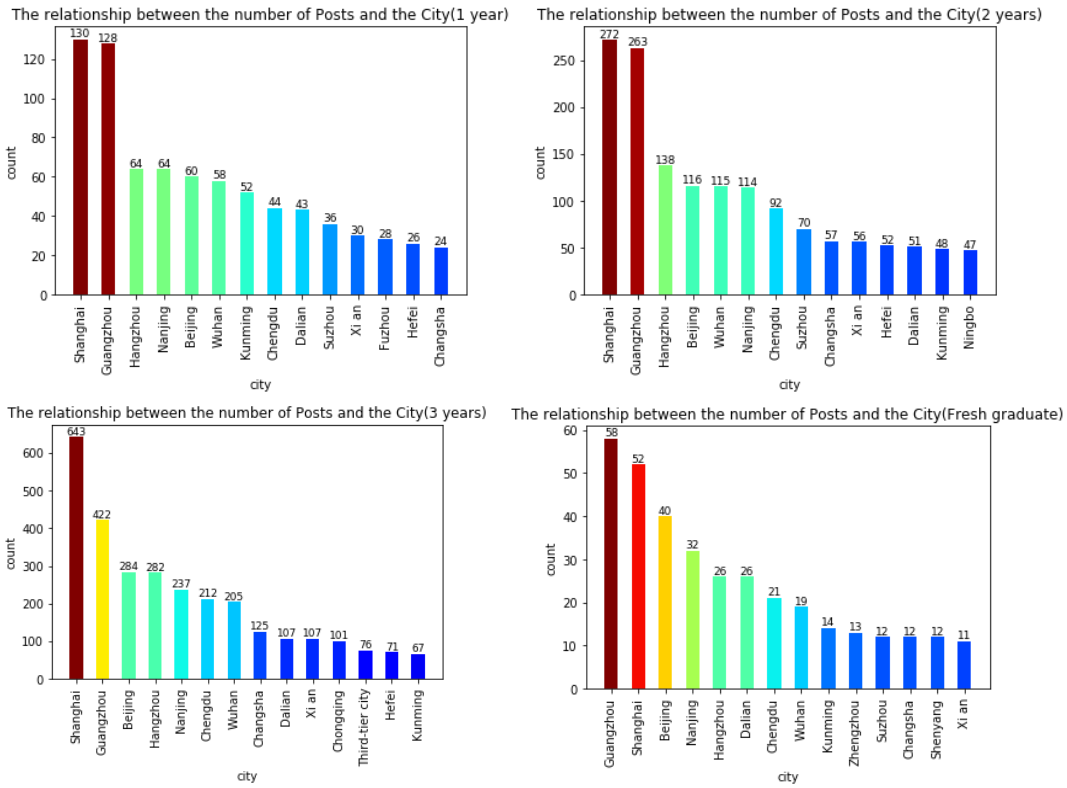


Figure 19 The relationship between the number of posts and city in different experience years

The four subplots are the demand for people with different work experience in different cities. It can be found from the pictures that for young people with fresh graduate and work experience of only 1 year, the demand of Guangzhou and Shanghai is far more than other cities. This explains why Guangzhou's job demand is high but its overall salary is very low. Because we have also reached a conclusion before that is the positive correlation between salary and work experience. For all stages, Shanghai is still the first choice for programmers not only because of its relatively high overall salary but also because it welcomes people with different experiences to work. However, we also know the fact that the high consumption level in big cities leads to high work pressure, so it is also a good choice for young people who are new to the workplace, such as Hangzhou, Nanjing, Wuhan and other capital cities.

Let's take a look at the demand for academic qualifications.

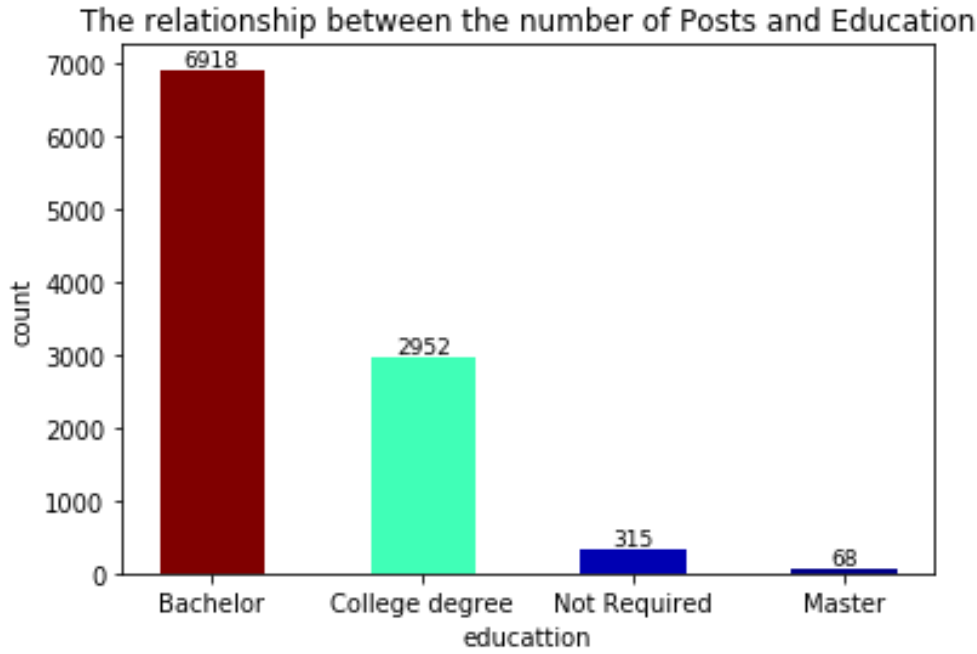


Figure 20 The relationship between the number of posts and education

Through the bar chart, you can find that the demand for undergraduate degree is the most and the number is more than twice that of the college degree. Only 2.99% of the positions are for applicants without academic qualifications. And there are only 68 positions with master's degree and above. It can be seen that for java positions, work experience is more important than education.

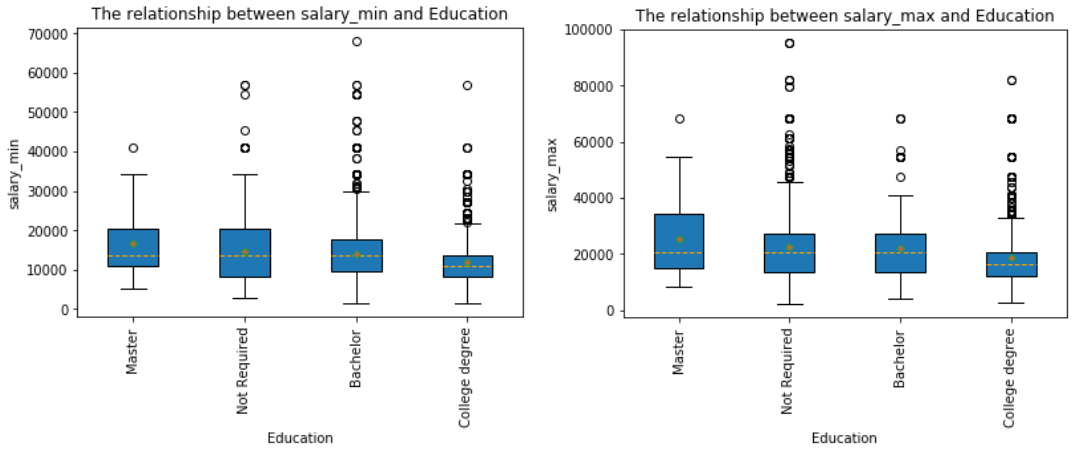


Figure 21 The relationship between salary and education

I use the box chart to find the relationship between education and salary. Judging from the situation reflected in the two pictures above, education and salary are positively correlated. In other words, the higher the qualifications of a programmer, the higher the salary he gets. In addition, it can be seen that the salary of a position without academic qualifications is close to that of a position with an undergraduate degree. Therefore, this type of education can also be classified into a group with an undergraduate degree.

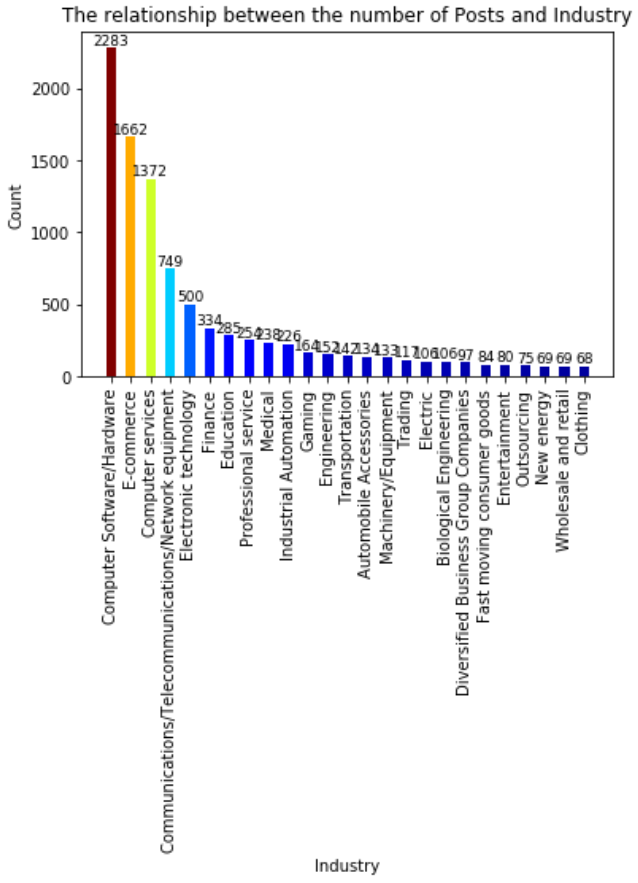


Figure 22 The relationship between the number of posts and industry

Through the bar chart, we can see that different industries have different requirements for java jobs. Among them, computer-related industries occupy the main position, followed by electronic technology, finance, professional services and other industries. The remaining industries have very little demand for Java programmers.

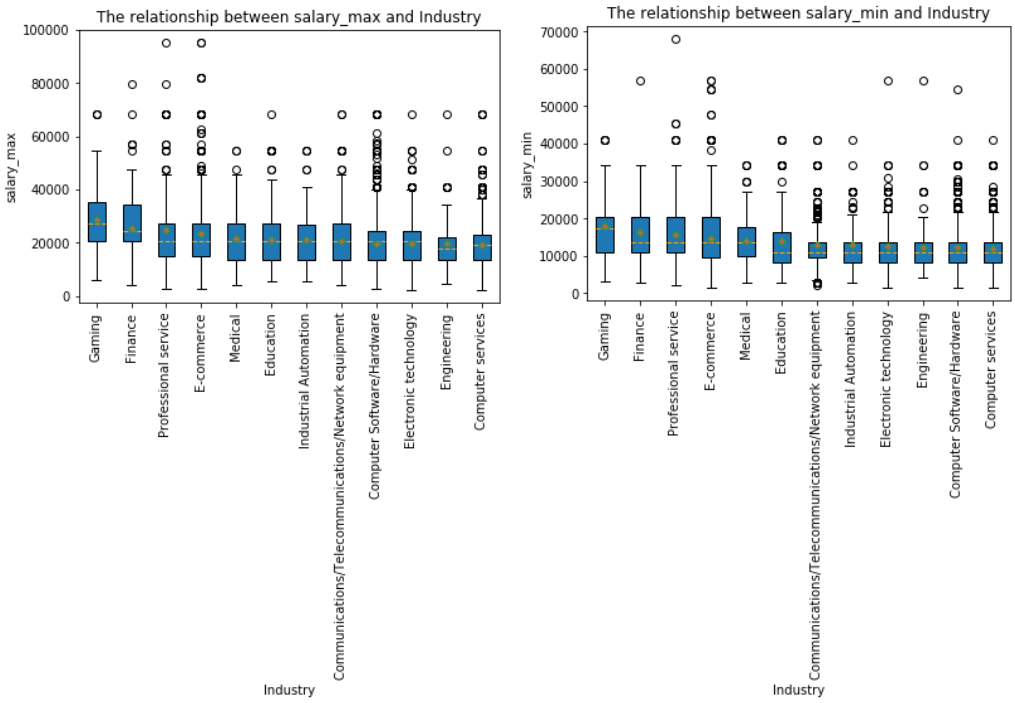


Figure 23 The relationship between salary and industry

Box charts are used here to observe the salary structure between different industries. Due to space factors, only the top 12 industries are shown here. Unlike the previous distribution, the salary of the computer-related industries here is ranked lower. The top rankings are the gaming industry, financial industry and professional services industry. The demand for java programmers in these industries is not high but the salary given are relatively high. This reflects a phenomenon in which single-type talents receive lower salaries, while compound programmers are more competitive in today's professional environment.

Finally, let's take a look at the company size.

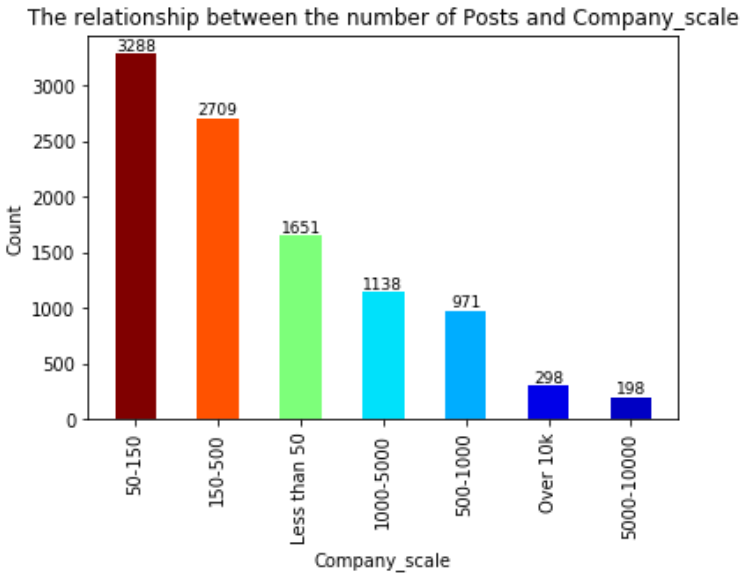


Figure 24 The relationship between the number of posts and company scale

The bar graph shows that the number of companies with fewer than 150 people and the number of companies with more than 15 people are similar. Most companies are small and medium-sized.

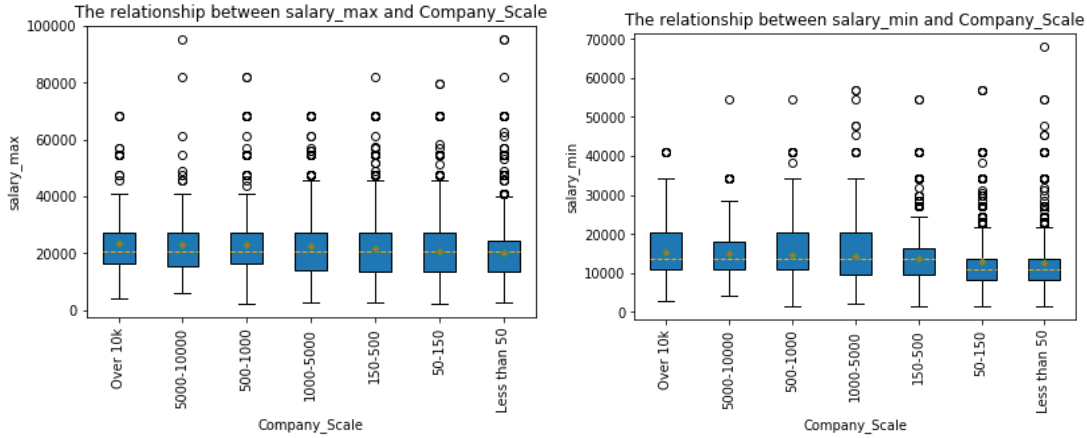


Figure 25 The relationship between salary and company scale

It can be clearly seen from the box chart that the larger the company is, the larger the average salary is. The last three companies are all companies with less than 150 employees. Therefore, this is also in line with the mentality of most people. Everyone hopes to be able to enter a large company to start their own career.

4.2 Keyword extraction for job requirements

Extract keywords by using the jieba package in Python and to generate a word cloud based on the word frequency by using the wordcloud package. First of all, I integrate the previously obtained job requirement. Here we need to combine each piece of data into a string. The high frequency words are automatically divided by using the

Jiebao package. And count the frequency of each word according to the number of times it appears. Only the first 23 pieces of data are displayed here for spatial reasons.

Table 5 Key technologies and their frequency

Tech	Num
Java	6817
Mysql	3387
Spring	2606
Oracle	2285
Mybatis	2195
Sql	1846
Springmvc	1771
Linux	1670
Redis	1668
Springboot	1636
Web	1489
Html	1478
Javascript	1475
J2ee	1298
Tomcat	1251
Jquery	1240
Hibernate	1240
Css	1226
Springcloud	1161
Sqlserver	882
Ajax	847
Maven	686
Dubbo	629

Through the statistical data, it is not difficult to find that Java is the most popular word, but there is no practical significance in this word. Mysql, Oracle and SQL are the most frequently used words. They are all database related. This shows that to be a java programmer, you need to be proficient in using the database. Spring, mybatis, springmvc and spring boot are the four most frequently used framework nouns. This shows that they are very mainstream Java frameworks so far. Through the above conclusion, we can show what the enterprises need at present, and give a enlightenment to the students who are learning Java technology to avoid learning the knowledge that is not popular without keeping up with the trend of the times. Generate a word cloud using the python code in the Jupyter notebook through the previously generated word frequency.

```
frequencies = {}
for line in open('C:/Users/HP/Desktop/honors project/word.txt'):
        arr = line.split(" ")
        frequencies[arr[0]] = float(arr[1])
wordcloud = WordCloud(
   font_path="C:/Windows/Fonts/simfang.ttf",
   background_color="white",width=2000,height=1500,max_words=100).generate(cut_text).fit_words(frequencies)

plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Figure 26 The code of getting wordcloud

It can be seen here that the size of the font is positively correlated with the frequency of occurrence. Therefore, the fonts generated with more occurrences are more eye-catching.

Figure 27 Wordcloud of 51job Java data set

Through word cloud, we can find that the data of theUK recruitment website is slightly different from that of Chinese recruitment website. Top of the list are agile projects and cloud computing. In addition, cloud services such as AWS and azure are mentioned most. The spring framework mentioned in 51job is not as popular here as you might think.

Figure 28 Wordcloud of UK recruitment website jobtotal

In the python position of 51job, Linux is the most mentioned technology, and in Java, Linux is also the most mentioned technology. Similarly, the database is quite a top thing. Besides MySQL and Oracle, mongodb and PostgreSQL need to be mastered by Python programmers. Python is widely used in the network through the listed technical aspects. The Django framework has been mentioned 308 times.

Table 6 Key technologies and their frequency of 51job Python data set

Python	1451
Linux	485
Mysql	430
Web	312
Django	308
Redis	264
Flask	235
Java	179
Sql	150
Mongodb	148
Http	123
Html	116
Tornado	116
Shell	105
Css	99
Javascript	99
Git	95
Tcp	92
None1	85
Ip	75
Js	72
Pythonweb	71
Postgresql	71
Oracle	70
Vue	69
Docker	69

Figure 29 Wordcloud of 51job Python dataset

4.3 Modelling

Data analysis is carried out by using the data processed before. Three different algorithms will be used. They are classification algorithm, association algorithm and cluster algorithm, and select a certain number of rules from each algorithm.

4.3.1 Classification algorithm

Classification algorithm is a supervised machine learning algorithm which is used to determine which categories are known in the target data. The process of classification is to classify each record into corresponding categories. I will use two methods to implement the classification algorithm. The first method is to use scikit-learn in Python to build a decision tree model. The second method is to use j48 algorithm to implement the classification algorithm in Weka. J48 is also a decision tree model in essence, but it has more pruning than ID3. I hope to be able to find a combination of low-income and middle-income through this algorithm.

4.3.1.1 Implementing decision tree in Python

First of all, add a new package sklearn to Python. The data used here are all numerical data after previous integration. Although unconformity data can also be used, it only needs to be coded before analysis. But in order to simplify the process, we still use the integrated data.

The first step is to read the data, and divide the read data into training set and test set. Here, 75% of the data is selected as the training data set and 25% as the test data set.

```
X = java_data.drop('salary_grade',axis=1)
y = (java_data['salary_grade'] == 1).astype(np.int32)
train_X,test_X,train_y,test_y=train_test_split(X,y,test_size=0.25, random_state = 125)
```

Figure 30 The code of select test data set and training data set

The second step is to define the index corresponding to the variable name and create a feature selector to calculate the F value of each prediction variable for the target variable.

```
sp = SelectPercentile(f_classif, 80)
sp.fit(train_X, train_y)
scores=pd.DataFrame({'feature':np.array(features), 'score':sp.scores_})
scores.sort_values("score", ascending = False, inplace=True)
print(scores)
```

Figure 31 The code of selecting the top 80% with the highest correlation

	feature	score
3	experience	1836.602122
0	joblocation_num	778.358154
1	Internship	714.775007
2	education	136.766268
5	company_scale	75.665750
7	job_classification	46.492079
6	job_industry	38.261838
4	company_style	15.721116

Figure 32 The top 80% with the highest correlation

The results show that the nature of the company has little to do with the predicted results. In other words, they don't have the ability to predict. Therefore, the first seven most important characteristic variables were selected.

The third step is to find the most suitable parameters. Set the maximum depth of the grid search parameters to 3,4,5,6 and the minimum doping reduction ratio in the range of [0.001-0.003]. Finally, set the resampling strategy to 3 fold cross validation.

```
clf = DecisionTreeClassifier()
param_grid = {'max_depth': [3,4,5,6], 'min_impurity_decrease': [0.0015,0.003]}
gs = GridSearchCV(clf, param_grid, 'roc_auc', cv = 3)
gs.fit(train_X, train_y)
cv_results = pd.DataFrame(gs.cv_results_)
print(gs.best_estimator_)
```

Figure 33 The code of creating a decision tree model

The results show that the optimal prediction effect can be obtained when the maximum depth is 6.

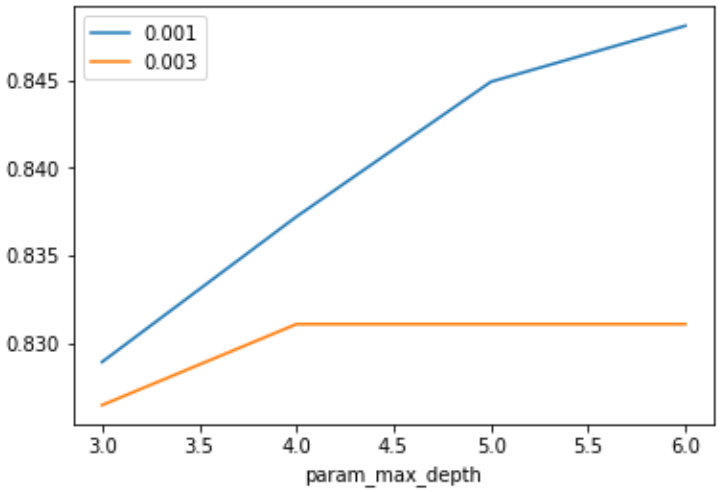


Figure 34 Show the optimal depth of the tree

Finally, the decision tree model is generated by using the search depth parameter 6.

```
dot_data = export_graphviz(clf, out_file=None, max_depth=6,
                           feature_names=np.array(features)[sp.get_support()].tolist(),
                           class_names=['medium','low'],
                           filled=True, rounded=True)
graph = graph_from_dot_data(dot_data)
graph.write_png('tree.png')
img = plt.imread('tree.png')
fig = plt.figure()
plt.imshow(img)
plt.axis('off')
plt.show()
```

Figure 35 The code of generating a decision tree graph

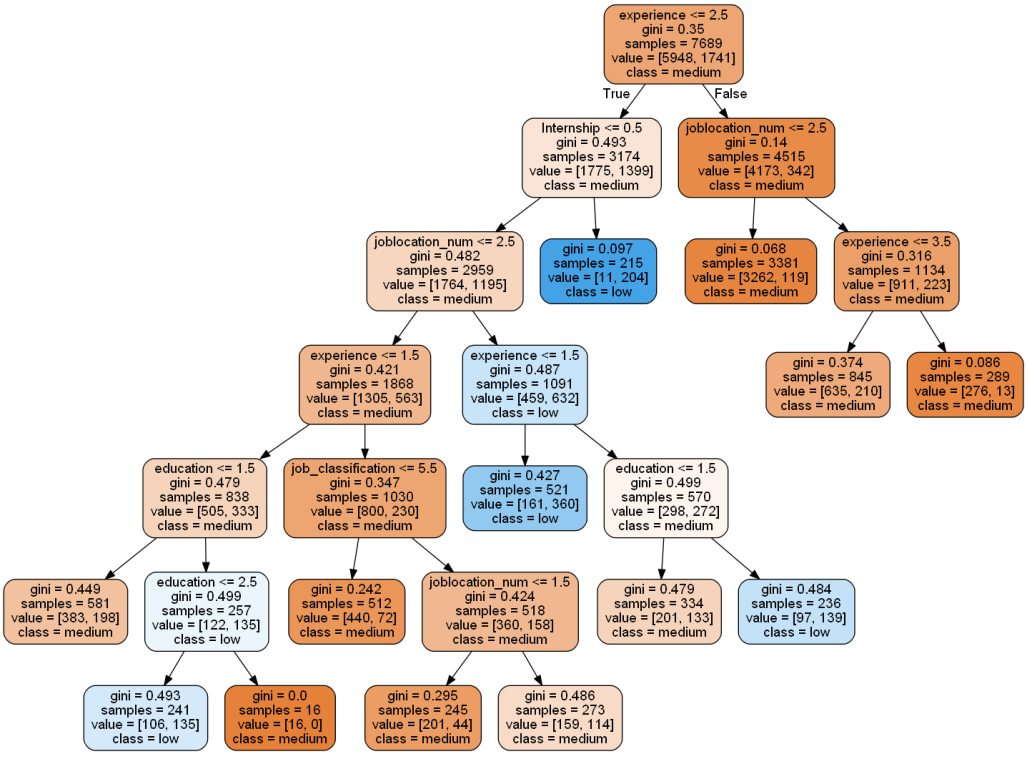


Figure 36 The decision tree

Based on this decision tree, I find out four rules:

Rule1: IF experience<=2.5 AND Internship >0.5 Then class=low (204/215)

This rule indicates that if the work experience is less than two years and he is an intern, the salary value he receives will be very low, 204 of the 215 records will be correctly marked. This is also in line with the normal logic. The income of an inexperienced intern is bound to be low.

Rule2: IF experience>2.5 AND joblation_num<=2.5 Then class= medium(3262/3381)

It showed that if the working experience is more than or equal to three years and he is in a municipality directly under the central government or a provincial capital city, then his salary is of medium and high level. A total of 3381 records, 3262 of which were correctly marked. This is also the most numerous of all the rules, which can also indicate the widespread application of this rule.

Rule3: IF joblocation_num>2.5 AND experience>3.5 Then class=medium(276/289)

This rule shows that a Java job, even in a second tier or third tier city, with a candidate's work experience of more than or equal to 4 years, he/she will earn medium and high income. Of the 289 rules, 276 were correctly marked. This rule shows that experience is more important than work city.

Rule4: IF Internship<=0.5 AND joblocation<=2.5 AND experience<=1.5 AND job_classification <=5.5 Then calss=medium(440/512)

This rule shows that if a candidate is a software engineer or Senior Software Engineer or Java developer in a non internship position in a first tier city, even if he is inexperienced, his income will not be low. The rule has 512 records, 440 of which are correctly marked. This rule shows that a good position is also an extremely important choice.

Finally, the test data set is used to test the reliability of the model P = 84.6%. The results show that the confidence level of the model is quite high.

4.3.1.2 Implementing classification algorithm in Weka

The operation process of Weka is more convenient than that of Python. Because it is highly encapsulated in the program, it can be realized only by importing the integrated data. First, we change the CSV format file to ARFF suffix file after Weka. The purpose of this is to facilitate the reading of files. The essence of this algorithm is ID3 algorithm. Also select use training set in test option. Then change the default parameter, because I found that there are too many default parameter leaves. Here, the confidence is changed to 0.1, and the remaining parameters remain unchanged.

The leaves are 70, the size of the tree is 85, and the depth of the number is 5. The accuracy of the model is 83.87%. Four rules with high confidence and quantity are also found here:

Rule1: IF joblocation_num=First-tier city AND Other First-tier city Then class=medium(1362/1655)

The rules show that most of the income in the first tier cities is of medium and high level. Here are 1655 data, 1362 of which are correctly marked.

Rule2: IF joblocation_=Second-tier city AND education=College degree Then class=low(176/253)

This rule indicates that in the second tier cities and with a college degree, his salary level is low. There are 253 records, 176 of which are correctly marked.

Rule3: IF experience=3 years AND 5 years Then class=medium(5850/6304)

The rule states that a candidate with 3 to 5 years of experience will be paid at a medium to high level. There are 6304 records, 5850 of which are correctly marked. The rule is also one of the most widely used.

Rule4: IF experience=1 year AND joblocation_num=Second-tier city and Third-tier city Then class=low(478/625)

This rule shows that he lacks working experience and his salary is low in the second and third tier cities. The rule has 625 records, 478 of which are correctly marked.

4.3.1.3 Association algorithm

In association algorithm, I choose to use Apriori algorithm to realize data mining. Apriori algorithm is also the most classical algorithm in association algorithm. Apriori is based on breadth-first search. The data I use in this algorithm is the nominal value data set described in the previous section. After associating with the default parameters, I found that the number of rules displayed in the result did not reach the preset 10 due to too high confidence. Therefore, I adjusted the default parameter value. I adjusted minMetric to 0.85, and numRules to 15, while the other parameters remained the default.

The following rules are 5 of 15 rules I chose.

Rule1: education=Bachelor job_classification=Senior software engineer 2283 ==> Salary_grade=medium 2106 conf:(0.92)

This rule shows that the salary level of Senior Software Engineer with bachelor degree is at the middle and high level, 2106 of 2283 data are correctly marked, and the confidence degree reaches 0.92, which has high availability.

Rule2: education=Bachelor experience=3 years 2657 ==> Salary_grade=medium 2428 conf:(0.91)

This rule shows that the salary of undergraduates with three years of experience is at a medium and high level, and 2428 items are correctly marked. The number and credibility of the rules are quite high.

Rule3: joblocation_num=First-tier city education=Bachelor 2560 ==> Salary_grade=medium 2364 conf:(0.92)

This rule shows that the income of undergraduate Java programmers in the first tier cities is also at a medium to high level. 2364 were correctly marked. The number and credibility of the rules are quite high.

Rule4: joblocation_num=First-tier city experience=3 years 1349 ==> Salary_grade=medium 1328 conf:(0.98)

This rule shows that the income of Java programmers with three years of experience in the first tier cities is also at a medium to high level. The number and credibility of the rules are quite high.

Rule5: joblocation_num=First-tier city job_classification=Senior software engineer 1214 ==> Salary_grade=medium 1175 conf:(0.97)

The rules show that the salary level of senior software engineers in the first tier cities is medium to high. The number and credibility of the rules are quite high.

4.3.1.4 Cluster algorithm

Cluster algorithm is a statistical analysis method to study classification problems. In Weka, I choose to use simplekmeans algorithm for clustering analysis. It belongs to unsupervised learning. Its algorithm idea is to take any k values as the center points, and then compare the Euclidean distance from each point to each center point in the data set. At last, it divides the close distance into a cluster. When the default number of clusters is 2, the part with low sum of salary level has been successfully obtained, but the accuracy is poor. So I adjusted the number of clusters to 4. Each cluster represents a rule.

Table 7 Show the clusters

Attribute	cluster0	cluster1	cluster2	cluster3
joblocation	Other First-tier city	First-tier city	First-tier city	Second-tier city
education	College degree	Bachelor	Bachelor	College degree
education	2 years	3 years	3 years	1 year
salary_grade	low	medium	medium	medium
company_scale	150-500	50-150	500-1000	50-150
job_industry	Computer Software/Hardware	Computer Software/Hardware	Computer services	Computer Software/Hardware
job_classification	Software Engineer	Software Engineer	Software Engineer	Senior software engineer

Cluster0 shows that this category of people is likely to receive low salary. If you have a Java software engineer with two years of work experience in a first-tier city other than Beijing, Shanghai and Guangzhou, and work in a small or medium scale, then he may get a lower salary. The total number of such people reached 2447 very close to 2308 in the original data.

The last three clusters show that these three categories of people are very likely to be middle-high income. The situation of 1 and 2 is very similar. They are all software engineers working in first-tier cities, working in medium-sized computer service companies and small-scale computer software and hardware companies.

People in the third category are highly likely to receive middle and high incomes. These people are senior software engineers working in small computer software and hardware companies in second-tier cities, although their academic qualifications are only college degrees.

5 Evaluation

This chapter will provide an assessment of the project. The evaluation covers the methods of project implementation, project conclusions and comparison of the results with the existing literature to analyze the advantages and disadvantages of the whole project.

5.1 Project Evaluation

All milestones for the whole project have been proved to be completed. In the actual implementation process, the complexity of data far exceeds my expectation, so more time is spent on data processing. In the process of project implementation, there are three main stages: data collection, data processing and data analysis.

In the data collection stage, scrapy framework is used to get the data of online recruitment. In the experiment, the same set of code can cope with two different websites at the same time. In the process of data mining, the content in the literature has been improved. When I was collecting data, I used the URL of each job to judge whether it was data that had been previously stored. In this step, I completed the deduplication of the data. In terms of data storage, I chose to store the data in a database, which is more secure and reliable. When dealing with anti-reptile systems, I chose the same method as in the literature, that is, adding header requests and adding virtual cookies. The implementation process of these contents is recorded in detail in the third chapter of this project. Chang et al (2019) proposed to directly output Excel files to realize data output. The way they adopted is to add breakpoints in the program to avoid getting data from the beginning. But their program has a major disadvantage is that it is easy to cause data duplication. Because the recruitment information of the website is updated every day, if the task cannot be completed within one day, then the data collected the next day will be repeated. In terms of anti-reptiles, Hu et al (2019) proposed that it can be implemented using the ip proxy pool, but I found that this measure has no practical effect during the actual experiment. Both Yin and Zhang (2019) and Chang et al (2019) use regular expressions to obtain content, and I use the Xpath method to make no difference in

terms of complexity. In the method of automatically obtaining data, the method used by Sun et al (2019) and others is to judge whether there is a next page based on the number of comments. The method I used is to directly loop to generate the link of the next column. The reason is that I find that the total number of pages is always fixed, so no need to make complicated judgments.

In the data cleaning stage, the structure of the data is completely different because it is two completely different websites. In my opinion, the main job is to unify all the data. For the attribute of work location, I have removed some small cities and classified them into corresponding provinces or counties for two different recruitment websites in the UK and China. Yin and Zhang (2019) use cities to divide cities into first-tier cities, second-tier cities, and third-tier cities. In China's recruitment network, large-scale streamlining of job classification and enterprise classification has also been carried out. In the 51job java data, they have 160 and 120 different data respectively. However, some of these variables have only a few data, and once these data are used in the subsequent data analysis, then there is no reference value for the final result. Therefore, I only keep those variables with a large number, and those very few variables are divided into the ‘others’ attribute. It takes a lot of time for the job of dividing salary grades. In my actual work, I conducted a total of three experiments and finally decided to divide according to the following division method. It was originally planned to divide the salary grade into three different grades, but in the actual analysis process, I found that the highest-income grade can rarely be correctly identified, so this division method proved to be wrong. The job information is divided into job content and application requirements here. When mining this type of data in 51job, I use regular expressions to obtain these data by identifying keywords. After actually obtaining the data set, I found that there are about 4000 data items that have been correctly divided. Therefore, I performed further processing on this part of the data in Excel and finally collected 5848 job contents and 5548 job requirements. The amount of data of this scale plays a key role in the subsequent keyword extraction. And for this kind of data on British websites, because they lack a unified structure, they will not be collected separately when they are collected.

In the data visualization, Wang (2019) gives the relationship between wages and cities. The number of posts between cities is displayed through a pie chart, and a histogram is drawn to show the salary level of each city. From his conclusions, it can

be seen that the number of java posts in Shanghai, Shenzhen, Beijing, Guangzhou, Hangzhou and other cities is the largest. And the demand in Shanghai is far greater than in other cities. This is the same conclusion as this project. Yang (2019) draws a graph of the relationship between salary and work experience and concludes that wages increase as work experience increases. This is the same as my conclusion. It also proved that my conclusion is correct. Wang (2019) gave keyword extraction, extracting the first twenty keywords according to the default parameters.

The evaluation of the model established in this project is as follows. In this project, a total of four different models were established using two kinds of software. They were decision tree model, association algorithm, and clustering algorithm. I used python and Weka to build the decision tree model. In Weka, the ID3 algorithm and J48 algorithm were used to analyze the decision tree model. It was found that the decision tree model caused by the ID3 algorithm was extremely large because it did not perform pruning. Therefore, only the J48 model was retained in the actual experiment. Through comparison, we can find that the prediction accuracy of the model is slightly higher than that of Python's decision tree model, but in fact there is not much difference.

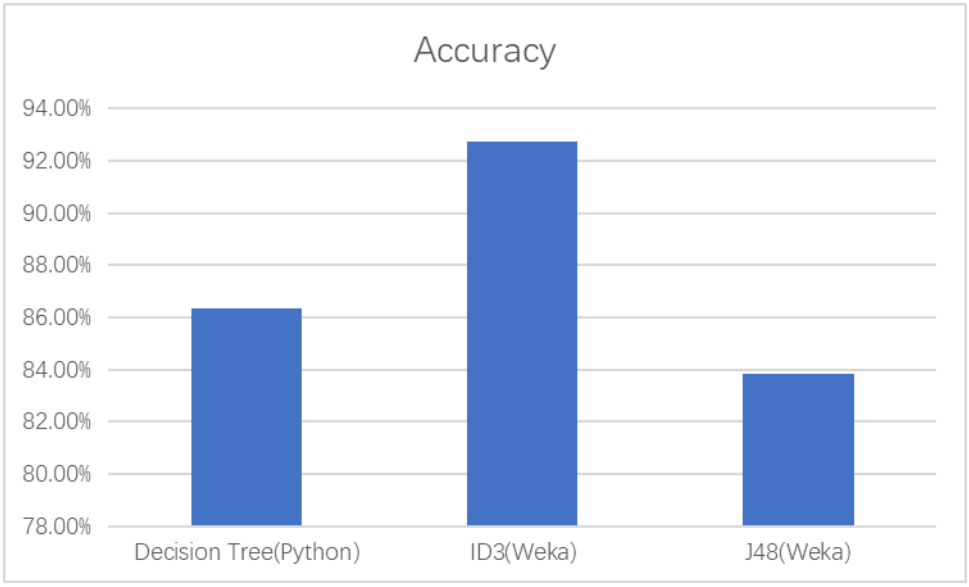


Figure 37 The accuracy of three different decision tree models

In general, decision tree model is better than association algorithm, and association algorithm is better than clustering algorithm. Because there are many attributes in the data set, and the clustering algorithm used by KNN model can not replace most of them very well, so the practical significance of the results given by this algorithm in

this project is not particularly obvious. Due to the lack of attribute set and sample data, there is no model for job total data set and python data set in 51job in this project.

5.2 Summary

All in all, the project is successful in terms of results and process. This project obtains the expected content through the model and programming by mining the existing network recruitment data. In this project, the following three questions are answered

1. What technology should programmers master to be more competitive in their jobs?
2. What parameters make a programmer more likely to earn medium and high income?
3. Where is the development prospect and opportunity?

This project is based on Python to realize data mining of online recruitment information, and to predict the treatment value of recruitment information according to the classification model. This data mining mainly uses python language, OpenRefine software and Weka software to realize data collection, data processing, data visualization and data modelling in the process of data mining. From the process of this research, we can see that multiple methods work together to provide powerful support for data mining. By using python language to perform visual operations in Jupyter notebook, compared to the one-sidedness in the literature, I completed the information mining between all attributes. In addition, I also pay more attention to the information mining of the content of the post, which is not mentioned in the previous literature. The data mining of online recruitment information and salary this time is mainly based on the analysis of job seekers, and proposes new characteristics of recruitment treatment, which can not only improve the recruitment efficiency of job seekers, but also reduce the cost of enterprise recruitment to a certain extent. In addition, it also provides valuable information for those who intend to work.

But it's a pity that the data of Internet recruitment in the UK is not as detailed as the data analysis of Internet recruitment in China. One of the main reasons is that the website itself provides less data, many of which are lack of practical analysis value. It was planned to have a detailed comparison of the recruitment data obtained by the two countries, but due to this reason, the data set obtained was too small and had to be abandoned. It has to be admitted that nowadays all the major Internet enterprises

have realized the importance of data, so the traditional crawler method has been difficult to deal with this. The original calculation method for the British recruitment website and China's recruitment website to find two sites. However, in the actual process, several websites that I used to use as backup have been deterred by the completion of anti crawler measures. The data set obtained now is the only one that can guarantee the reliability and availability of data. This is also a pity in this project.

6 Conclusion and Future Work

6.1 Conclusion

The origin of the project comes from the fact that I found that online recruitment is very popular now. The website not only publishes their recruitment information, but also provides big data of position information for analysis. But a lot of online recruitment information analysis are only limited to the actual data given by the website, but ignore that there are still a lot of data to be analysed in the position information and position requirements. I use scrapy framework to obtain raw data, use OpenRefine and Python to clean and integrate data, and use Weka and Jupyter notebook to analyse data. In general, the project has achieved the main goals and objectives to be achieved.

In the data collection phase, I used Python scrapy crawler package and fully combined with the principle of Web data mining to collect the data set required by this project in the recruitment information network. The scrapy framework crawler program can adapt to both Chinese recruitment websites and British recruitment websites. At the same time, I also successfully used anti-crawler technology to obtain data.

In the data pre-processing stage, OpenRefine, python, and Excel are used to clean and integrate the data. I use python to make data visualization to provide a reference for data cleaning and use Excel to integrate the same attributes. I use OpenRefine for data filling. I also use OpenRefine to change the data type such as converting numeric value to nominal value and vice versa.

In the data visualization phase, I use Python language to draw graphs in Jupyter notebook to find the relationship between different data attributes. In addition, I also use Python keyword extraction function to extract the position information content and use Wordcould package to display the extracted keywords according to word frequency.

In the data modelling phase, I use Python machine learning library named sklearn to build the model of decision tree algorithm with short code. I also use j48 algorithm,

Apriori algorithm and SimpleKmeans algorithm in Weka software to implement classification algorithm, association algorithm and clustering algorithm respectively.

In the project evaluation stage, the production confusion matrix based on the prediction results is evaluated, and the accuracy and error rate of the model are calculated. By comparing and evaluating the classification algorithms in two different softwares, we get the result that Python decision tree model has high prediction accuracy. For the mining of position information, it is concluded that both Java programmers and python programmers need to master the database operation ability. In addition, the most widely used model for Java is spring model. This project also gives a comprehensive analysis of how programmers can get a decent income. This is a very good suggestion for young people who are learning computer programming language and are looking for a job.

6.2 Future Work

In the actual data collection process, I have found that the existing websites have a strong sense of protection for their own data. The crawler I designed based on the scrapy framework has been difficult to cope with the anti-crawler mechanism of most websites. I found that some websites have used dynamic presentation for page generation. It is difficult to mine data by analysing the source code of web page. Therefore, I hope that in the next learning process, I can find a way to collect website information on the principle of legal compliance. In the process of information collection, it may be more dependent on the network server in the future. Because in the process of their own experiments, they took up a lot of memory of their own computer, resulting in the computer has nothing to do but run programs.

In addition, the reason for the poor data analysis of the UK website is that the website itself puts a lot of useful information such as work experience and degree requirements into the job description. This is different from the key information listed separately by the online recruitment website of China Pavilion. It's hard for me to get the expected results by using the traditional way of getting this kind of key information through word frequency. Through understanding, we know that this part of knowledge belongs to the scope of natural language processing. So, in the future, I want to learn natural language processing to extract the key information, and then have a comprehensive and thorough analysis of recruitment information to find out a model of maximizing revenue.

References

Madia, S. (2011) A Best practice for using social media as a recruitment strategy. ***Strategic HR Review***; ***10(6)*** 19-24.

Kluemper, D. H & Rosen, P.A. (2009) Future employment selection methods: evaluating social networking web sites. ***Journal of Managerial Psychology, 24(6),*** 567-580.

Wang, J. L. & Wang, X. H. (2019, September) Implementation of crawler technology based on Python. ***Software Development and Application,*** p.18-20

Fu, L. M. (2019) Application of regular expression in Python crawler. ***Computer Knowledge and Technology, 15(25)*** 253-254

Mark, L. (Ed.), (2013) *Learning Python.* Farnham: O'Reilly Media

Smith D. & ALI A. (2014) Analyzing computer programming job trend using web data mining. ***Informing Science and Information Technology***, ***11***: 203-214.

Karakatsanis, I., Alkhader, W.& Maccrory, F. (2017) Data miningapproach to monitoring the requirements of the job market: A case study. ***Information Systems, 65(4)***: 1-6.

Jia, N.Y. (2019) data analysis based on Python crawler ***Information Technology and Informatization 4***: 64-66

Wei, T.T. Fang, H.Y. & Song, S.L. (2019) The Employment Characteristic Mining of Data Analysis Class Position in the Context of Big Data. ***Modern computer 25***:14-17

Jiang L., Zhu S.C., & Fu P.H. (2016) The influencing factors of job seekers' satisfaction in comprehensive recruitment websites. ***Electronic Commerce (11)***,90-92

Ma, Z.Y. Problems and Countermeasures of Internet recruitment under the era of "Internet +". ***Modern marketing (09)***200-201

Li, A.N., Zhang, X. & Zhang, B.Y (2017) Research on performance evaluation method of public cloud storage system. ***Journal of Computer Applications 37(5)*** : 1229 － 1235

Zhao, L., Hu, S.Y., &Wu, Y.L. (2017) Research of Aliyun ECS-Based Animal Medicine Examination System. ***Journal of Domestic Animal Ecology 38(9)*** : 94 － 96 .

Guo, L.R. (2018) Design of research data analysis platform based on Scrapy ***Electronic Technology & Software Engineering (23)*** :136-137

Shi, Z., Shi, M., & Lin, W. (2016). The Implementation of Crawling News Page Based on Incremental Web Crawler.

Cattell R. (2011) Scalable SQL and NoSQL data store ***ACM SIGMOD Record 2011(2)*** : 12－27

Chen, L.T. (2016) Anti Crawler Technology in the Era of Big Data ***Computer and Information Technology ,24(6)***:60-61.

Han, J.M. Meng, X.F. & Wang, J. (2011) RESEARCH ON WEB MINING: A SURVEY, J***ournal of Computer Research and Development,27(4)***:15- 18.

Garcia S. Luengo J & Herrera F. (2016) Tutorial on practical tips of the most influential data preprocessing algorithms in data mining ***Knowledge −Based Systems 98***: 1－29 .

Galar M，Fernandez A，& Barrenechea E(2012) A review on ensembles for the class imbalance problem: bagging-，boosting-，and hybrid- based approaches ***IEEE Transactions on Systems，Man，and Cybernetics，Part C: Applications and Reviews 42(4)*** : 463－484 .

Triguero I，Peralta D & Bacardit J. MRPR (2015) A Map Reduce solution for prototype reduction in big data classification ***Neurocomputing 150:*** 331－345

Gao, M., Hong X. & CHEN S (2011) A combined SMOTE and PSO based RBF classifier for two－class imbalanced problems ***Neurocomputing 74(17)*** : 3456－3466

Sotoca, J. M. & Pla, F. (2010) Supervised feature selection by clustering using conditional mutual information-based distances ***Pattern Recognition 43(6)*** : 2068－2081

Wand H, & Wang S. (2010) Mining incomplete survey data through classification ***Knowledge and Information Systerms 24(2)*** : 221－233

Perezortiz M，Gutierrez P & Martinez C.H. (2015) Graph－based approaches for over－sampling in the context of ordinal regression ***IEEE Transactions on Knowledge and Data Engineering 27(5) :*** 1233－1245

Prati R.C. , Batista G.E.A.P.A & Silva D.F. (2015) Class imbalance revisited: a new exper－imental setup to assess the performance of treatment methods ***Knowledge and Information Systems 45(1) :*** 247－270 .

Angiulli F & Folino G. (2007) Distributed nearest neighbor-based condensation of very large data sets ***IEEE Transcactions on Knowledge and Data Engineering，2007，19 (12) :*** 1593－1606 .

Bacardit J. , Widera P. & Chamorro A.E.M (2012) Contact map prediction using a large－scale ensemble of rule sets and the fusion of multiple predicted structural features ***Bioinformatics，2012，28(19) :*** 2441－2448

Raicu, I. (2019) Financial Banking Dataset for Supervised Machine Learning Classification ***Informatica Economica Vol. 23(1):*** 37-49

Bassam, F., Mohd S.H., & Mohammad, S. (2018)Crawling of Japanese Real-Estate Websites Using Scrapy ***International Journal of Advanced Research in Computer Science 9,(2):*** 64-67.

Dimitrios K.L. *Learning Scrapy : learn the art of efficient web scraping and crawling with Python* Birmingham Packt Publishing

Fabio Nelli *Python Data Analytics Data Analysis and Science using pandas, matplotlib and the Python Programming Language* Berkeley, CA Imprint: Apress

Bing Liu *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data* Berlin Heidelberg : Springer Berlin Heidelberg

Kong, Q., Ye, C.Q. & Sun Y. (2018) Research on Data Preprocessing Methods for Big Data ***Computer Technology and Development 28(5):*** 1-4

Li, Y. Data mining based on Python (2018) ***Computer Knowledge and Technology 14(23)***: 15-20

Shen, R.F., Shi, X.J. & Wu, Y.H. (2005) Data Preprocessing Procedure Model based on Data Warehouse ***Computer & Digital Engineering*** 33(9):73-75

Wang, F., Zhang R. & Gong H.R. (2019) Design and implementation of a distributed web crawler base on scrapy framework ***Information Technology 3:***96-101

Sun, H.J. (2014) Research of Web Data Mining based on Cloud Computing ***Intelligent Computer and Applications 4(05):*** 103-105

Gao, Y. & Hu, J.T. (2002) Principles,Methods and Application of Web Mining ***New Technology of Library and Information Service 3:*** 51-53
Li, G.H. (2008) The Reasearch of Web Data Mining ***Computer Knowledge and Technology 4:***592-595
Sun J.Y. Ma Y.X. & Wu W.J. (2019) Network Crawler System Based on Python ***Computer Knowledge and Technology:*** 15(26),61-63
Wang F. (2019) Information crawling and data analysis of recruiting website based on Python ***Information Technology and Network Security*** 38(08),42-46
Yin L.F. & Zhang H.R. (2019) Crawling and analysis of online recruitment information based on Python ***Electronic Design Engineering*** 27(20),22-26
Guan X.H. Huang S.Q. & Wei L. (2020) Design and implementation of job search information collection and analysis system using Python ***Computer Era*** (03),32-34
Yang Z. (2019) Analysis on Visualization of Recruiting Information Based on Python Language ***Computer & Network*** 46(02),61-64
Yang Y. (2019) Web Data Mining and Analysis Based on Python Language ***Modern Information Technology*** 3(23),63-65
Li Z.H. (2019) Comparative Research on Internet Recruitment and Traditional Recruitment ***E-Business Journal*** (01),93-94
Dai Y. (2015) An Analysis of Recruitment Ways in the Internet Age ***Business*** (40) 216
Liu Z. (2017) Analysis of the current situation and development trend of enterprise online recruitment ***China Journal of Commerce*** (01),78-79
Xing X.L. (2017) China Internet Recruitment Development History ***The Internet Economy*** (03),92-97
Zhao Q.B. Ji H.L. & Liu D.B. (2012) China's online recruitment industry: development status, trends and strategies ***Commercial Research*** (09)43-49
Wang H. & Peng Q. (2017) Analysis of Recruiting Websites' Web Attention Based on Baidu Index:A Case Study of Zhilianzhaopin ***Journal of Hunan University of Technology(Social Science Edition)*** (06) 29-35
Huang W. (2007) The Algorithm of Decision Trees: ID3 and C4.5 ***Sichuan University of Arts and Science Journal*** (05) 16-18
Cheng F.F. Wang Z.N. & Hou L.Z. (2015) Data Mining Application in Weka Platform Based on Decision Tree Classification ***Microcomputer Applications*** 31(06),63-65
Dong T. (2015) The application of association rules in weka based on data mining ***Machine Design and Manufacturing Engineering*** 44(12),78-80
Chen H.P. Lin L.L. Wang J.D. & Miao X.R. (2008) Data mining platform-WEKA and secondary development on WEKA ***Computer Engineering and Applications*** 44(19) 76-79

Appendix 1 Project Overview

Initial Project Overview

SOC10101 Honours Project (40 Credits)

Title of Project: Python-based recruitment data collection and analysis

Overview of Project Content and Milestones

Nowadays, there is a phenomenon in society that companies cannot find the employees they want and job seekers cannot find the enterprises they want. This seemingly contradictory phenomenon seems to me to be caused by people's lack of understanding of the current employment environment. Thus, my honours project is to collect, categorize and integrate the job information on the job network. In order for university students to have a clear understanding of social needs in their career planning, make career planning as soon as possible, and choose career direction in time. Accurate and comprehensive access to recruitment information can enable university students to avoid employment risks in a timely manner. Therefore, my honours project will be divided into several important stages. First of all, search and review relevant references. Secondly, collect and integrate data for analysing. Thirdly, apply data mining techniques to the dataset collected, to find interest patterns. Then, findings will be analysed and evaluated. Finally, a set of suggestions will be generated, which will provide assistant to both job seekers, and companies.
Milestones include: data collection and integration; generating patterns by applying data mining techniques on the data collected; Analysing findings

The Main Deliverable(s):

The project will eventually show in the form of a report where the most jobs are available today and which jobs are sought after by others. In addition, the future trend of some industries will be forecasted. The ultimate goal is to provide some reference for those who are planning their careers to keep up with the trend of the times.

The Target Audience for the Deliverable(s):

Students, young people or people who are looking for a job

The Work to be Undertaken:

Do a literature review on relevant topics
Collect and integrate data
Analyse data by the use of data mining techniques, in Weka, Jupyter notebook or R studio
Analyse and evaluate findings

Additional Information / Knowledge Required:

Learn existing data mining techniques to collect data
Using Panda Library in Python to do the data analysis
Using Scikit-learn Library in Python to do machine learning
The above two are what I have learned and will broaden my knowledge in these 2 fields.

Information Sources that Provide a Context for the Project:

Google Scholar
School library
“Learning Python” Mark Lutz
“Python for Data Analysis” Wes McKinney
“Second-hand house data crawling and analysis based on python” Lyucao Zhao
“Python-based web big data collection and data analysis” Le Xiao

The Importance of the Project:

Previously, only use Python to make data analysis on given data. But in this project, I want to use data mining technology to collect data and make analysis and prediction by myself.

The Key Challenge(s) to be Overcome:

How to collect data.
How to integrate data for analysing
Which techniques are better for such an analysing

Appendix 2 Second Formal Review Output

SOC10101 Honours Project (40 Credits)

Week 9 Report

Student Name: Yijia Sun
Supervisor: Taoxin Peng
Second Marker: Peter Barclay
Date of Meeting: 27 Nov
Can the student provide evidence of attending supervision meetings by means of project diary sheets or other equivalent mechanism? **yes**

~~If not, please comment on any reasons presented~~

Please comment on the progress made so far

- ⇨ Good progress so far. Good references in report. Literature review concentrates mainly on technology side rather than application area (recruitment data), this could be balanced to give a more project-specific review.

Is the progress satisfactory? **yes**
Can the student articulate their aims and objectives? **yes**
If yes then please comment on them, otherwise write down your suggestions

- ⇨ Yes, but greater clarity would help progress of the project and the evaluation.

Does the student have a plan of work? **yes**
If yes then please comment on it, otherwise write down your suggestions

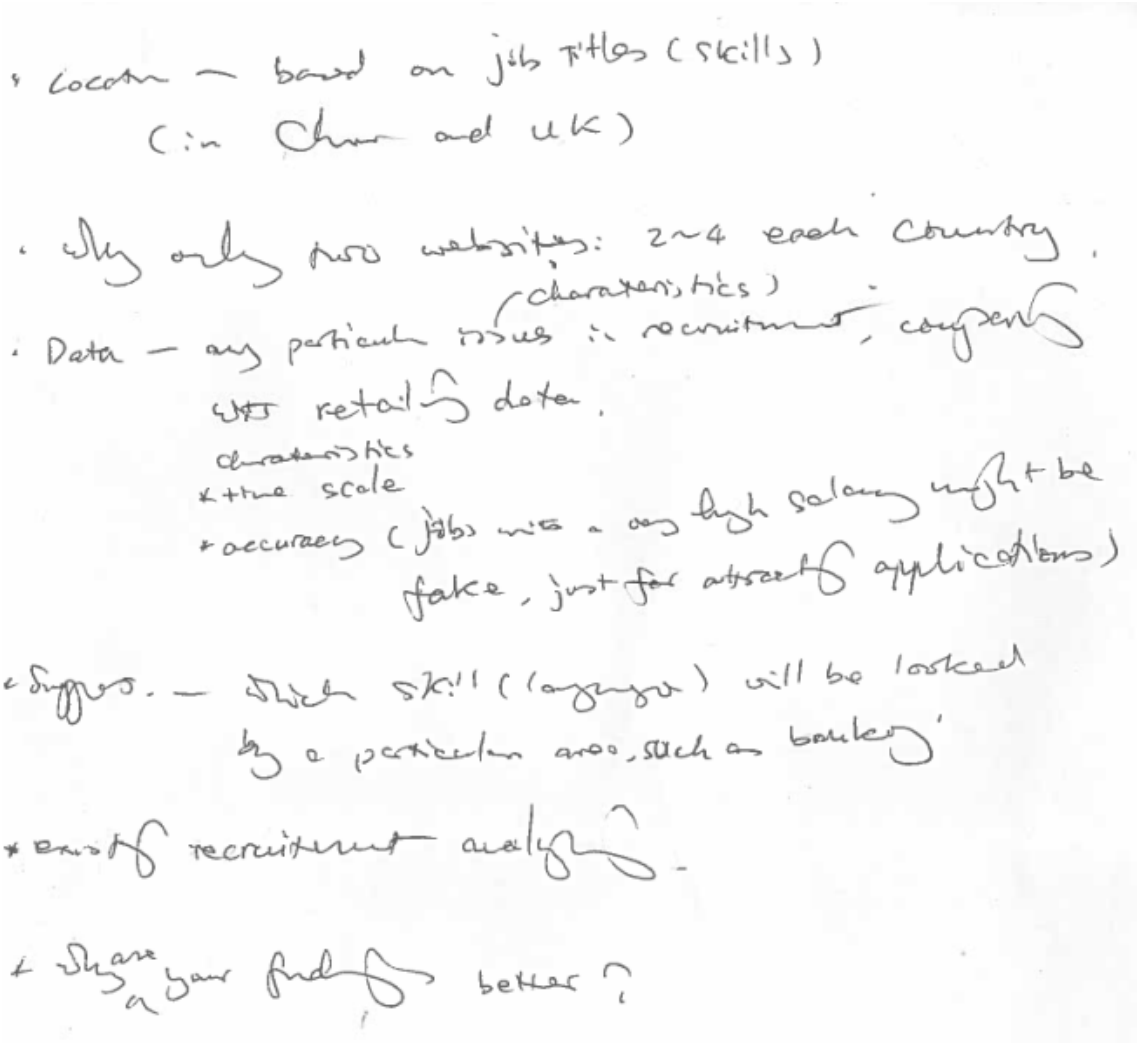
- ⇨ Plan is quite high level, perhaps identify more specific milestones.

Does the student know how they are going to evaluate their work? **no***
If yes then please comment otherwise write down your suggestions.

- ⇨ Project approach is good overall, but Yijia needs to think further about how to evaluate.

Any other recommendations as to the future direction of the project

- ⇨ Availably and reliability of data may be an issue.



Signatures: Supervisor Second Marker

Student Sun Yijia

The student should submit a copy of this form to Moodle immediately after the review meeting; A copy should also appear as an appendix in the final dissertation.

Appendix 3 Diary Sheets (or other project management evidence)

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Yijia Sun **Supervisor: Dr Taoxin Peng**

Date: 16/10/2019 **Last diary date: First**

Objectives:

1. Define aims and objectives of the project
2. Create a G-Chart (project plan)
3. Start literature review
   a. Search references
   b. Background study
4. Reference list

Progress:

1. Done
2. Done
3. a Partial
   b Done
4. Partial

Supervisor's Comments:

A good start with aims and objectives clearly defined. The project chart is indicative, might need to be refined at a later stage.
However, The background study is far from "done". This is just the beginning of a literature review. Documents provided on moodle are a good source of information, which need to look into carefully.

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Yijia Sun **Supervisor: Dr Taoxin Peng**

Date: 23/10/2019 **Last diary date: 16/10/2019**

Objectives:

1. Refine aims and objectives of the project
2. Download relevant documents on moodle, such as how to do a literature review. Also find a sample dissertation for references.
3. Continue -- literature review
   a. Search references
   b. Background study
4. Reference list

Progress:

1. Done
2. Done
3. Partial
4. Partial

Supervisor's Comments:

The aims and objectives are clearly defined. This is a good start. However, the progress of literature review has been too slow. Also far more references are required for this project.

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Yijia Sun **Supervisor: Dr Taoxin Peng**

Date: 23/10/2019 **Last diary date: 16/10/2019**

Objectives:

1. Refine aims and objectives of the project
2. Download relevant documents on moodle, such as how to do a literature review. Also find a sample dissertation for references.
3. Continue -- literature review
   a. Search references
   b. Background study
4. Reference list

Progress:

1. Done
2. Done
3. Partial
4. Partial

Supervisor's Comments:

The aims and objectives are clearly defined. This is a good start. However, the progress of literature review has been too slow. Also far more references are required for this project.

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Yijia Sun **Supervisor: Dr Taoxin Peng**

Date: 20/11/2019 **Last diary date: 30/10/2019**

Objectives:

1. Continue -- literature review
    a. Background study- include a description of what is online job market, how does it work, any problems and issues with the current system, current methods, approaches to these problems, data mining techniques, data collection/integration techniques
    b. Add more details
2. Reformat the document, using the template
3. Prepare the interim document (within the template), mainly on literature review, also have all other designed chapters, with a brief description for each of the chapters.
4. Reference list

Progress:

1 Partial
2 Done
3 Done
4 Partial

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Yijia Sun **Supervisor: Dr Taoxin Peng**

Date: 06/02/2020 **Last diary date: 20/11/2019**

Objectives:

1. Finalise -- literature review, adding sections, such as data analytics and a summary at the end. See comments on the document.

2. Complete -- Data gathering and preparation – data gathering, cleaning, integration. Gathering data from both UK and China markets.
3. Start - Data Analysing
4. Modify G-Chart

Progress:

1. Partial
2. Partial Finish data gathering start doing data cleaning and integration
3. Partial
4. Done

Supervisor's Comments:

It is good that you have already collected enough data, two sets, one from UK and one from China. However, the written up work should be done in parallel. The structure of literature review needs to be re-organised to four mail sections: background about recruitment approaches, issues; Web data mining (in general); Data Processing; Data Analysis. Adding a summary at the end.

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Yijia Sun **Supervisor: Dr Taoxin Peng**

Date: 12/02/2020 **Last diary date: 06/02/2020**

Objectives:

1. Finalise -- literature review, re-organise sections and add a summary at the end. See comments on the document.
2. Complete (in written) -- Data gathering and preparation – data gathering, cleaning, integration. Gathering data from both UK and China markets.
3. Modify G-Chart

Progress:

1. Done
2. Partial Finish written data gathering. Data cleaning is writing and data integration is processing.
3. Done

Supervisor's Comments:

Very good progress.

Appendix 4 The results of algorithms in Weka

The results of J48 in Weka

```
J48 pruned tree
------------------

experience = 2 years
|   joblocation_num = First-tier city: medium (651.0/81.0)
|   joblocation_num = Other First-tier city: medium (711.0/212.0)
|   joblocation_num = Second-tier city
|   |   education = Bachelor: medium (280.0/97.0)
|   |   education = College degree: low (176.0/77.0)
|   |   education = Not Required: medium (6.0/1.0)
|   |   education = Master: medium (0.0)
|   joblocation_num = Third-tier city
|   |   company_scale = 1000-5000: medium (33.0/15.0)
|   |   company_scale = 500-1000: low (20.0/7.0)
|   |   company_scale = Less than 50: low (56.0/14.0)
|   |   company_scale = 150-500
|   |   |   education = Bachelor: medium (40.0/15.0)
|   |   |   education = College degree: low (26.0/7.0)
|   |   |   education = Not Required: low (1.0)
|   |   |   education = Master: medium (1.0)
|   |   company_scale = 50-150: low (90.0/42.0)
|   |   company_scale = Over 10k: medium (11.0/1.0)
|   |   company_scale = 5000-10000: medium (8.0/4.0)
experience = 3 years: medium (3881.0/426.0)
experience = 5 years: medium (1969.0/28.0)
experience = 1 year
|   joblocation_num = First-tier city
|   |   education = Bachelor: medium (213.0/62.0)
|   |   education = College degree: low (95.0/42.0)
|   |   education = Not Required: medium (4.0)
|   |   education = Master: medium (6.0)
|   joblocation_num = Other First-tier city
|   |   job_classification = System architect: medium (2.0/1.0)
|   |   job_classification = Software Engineer: low (173.0/77.0)
|   |   job_classification = Internet Software Development Engineer
|   |   |   company_scale = 1000-5000: low (3.0/1.0)
|   |   |   company_scale = 500-1000: low (2.0)
|   |   |   company_scale = Less than 50: low (6.0/2.0)
|   |   |   company_scale = 150-500: medium (14.0/5.0)
|   |   |   company_scale = 50-150
|   |   |   |   education = Bachelor: medium (7.0/1.0)
|   |   |   |   education = College degree: low (11.0/3.0)
|   |   |   |   education = Not Required: medium (0.0)
|   |   |   |   education = Master: medium (0.0)
|   |   |   company_scale = Over 10k: low (1.0)
|   |   |   company_scale = 5000-10000: medium (1.0)
|   |   job_classification = Java Developer: medium (49.0/18.0)
```

```
|   |   job_classification = Web front-end development: low (10.0/2.0)
|   |   job_classification = Senior software engineer: medium (77.0/28.0)
|   |   job_classification = Others: medium (12.0/5.0)
|   |   job_classification = Manager: medium (2.0/1.0)
|   |   job_classification = ERP: low (4.0/1.0)
|   |   job_classification = Technical support/maintenance engineer: low (2.0)
|   joblocation_num = Second-tier city: low (289.0/94.0)
|   joblocation_num = Third-tier city: low (189.0/53.0)
experience = no experience
|   company_scale = 1000-5000: medium (4.0/1.0)
|   company_scale = 500-1000: medium (5.0/1.0)
|   company_scale = Less than 50: low (4.0/1.0)
|   company_scale = 150-500: low (15.0/7.0)
|   company_scale = 50-150: low (17.0/4.0)
|   company_scale = Over 10k: medium (4.0)
|   company_scale = 5000-10000: medium (1.0)
experience = Not Required
|   Internship = No
|   |   education = Bachelor: medium (158.0/48.0)
|   |   education = College degree: low (81.0/24.0)
|   |   education = Not Required
|   |   |   joblocation_num = First-tier city: medium (19.0/2.0)
|   |   |   joblocation_num = Other First-tier city: medium (25.0/7.0)
|   |   |   joblocation_num = Second-tier city
|   |   |   |   company_scale = 1000-5000: low (1.0)
|   |   |   |   company_scale = 500-1000: low (0.0)
|   |   |   |   company_scale = Less than 50: low (0.0)
|   |   |   |   company_scale = 150-500: medium (3.0)
|   |   |   |   company_scale = 50-150: low (6.0)
|   |   |   |   company_scale = Over 10k: medium (2.0)
|   |   |   |   company_scale = 5000-10000: low (0.0)
|   |   |   joblocation_num = Third-tier city: low (14.0/5.0)
|   |   education = Master: medium (10.0/2.0)
|   Internship = Yes: low (52.0/2.0)
experience = 8 years: medium (123.0)
experience = Fresh graduate: low (494.0/127.0)
experience = 4 years: medium (35.0)
experience = 7 years: medium (3.0)
experience = over 10 years: medium (39.0)
experience = 6 years: medium (6.0)

Number of Leaves  :         70

Size of the tree :         85
```

```
=== Summary ===

Correctly Classified Instances        8599               83.8681 %
Incorrectly Classified Instances      1654               16.1319 %
Kappa statistic                          0.5016
Mean absolute error                      0.2379
Root mean squared error                  0.3449
Relative absolute error                 68.2006 %
Root relative squared error             82.5871 %
Total Number of Instances            10253

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.925     0.460     0.874       0.925   0.899      0.507    0.839     0.930     medium
               0.540     0.075     0.678       0.540   0.601      0.507    0.839     0.609     low
Weighted Avg.  0.839     0.373     0.830       0.839   0.832      0.507    0.839     0.858

=== Confusion Matrix ===

    a    b   <-- classified as
 7353  592 |    a = medium
 1062 1246 |    b = low
```

The results of ID3 in Weka

```
Correctly Classified Instances        9507               92.7241 %
Incorrectly Classified Instances       746                7.2759 %
Kappa statistic                          0.7751
Mean absolute error                      0.0919
Root mean squared error                  0.2143
Relative absolute error                 26.329  %
Root relative squared error             51.314  %
Total Number of Instances            10253

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.983     0.264     0.928       0.983   0.954      0.783    0.981     0.994     medium
               0.736     0.017     0.925       0.736   0.820      0.783    0.981     0.936     low
Weighted Avg.  0.927     0.208     0.927       0.927   0.924      0.783    0.981     0.981

=== Confusion Matrix ===

    a    b   <-- classified as
 7808  137 |    a = medium
  609 1699 |    b = low
```

The results of Apriori algorithms in Weka

```
 1. experience=5 years job_classification=Senior software engineer 1102 ==> Average_salary=medium 1095    <conf:(0.99)> lift:(1.28) lev:(0.02) [241] conv:(31.01)
 2. education=Bachelor experience=5 years 1491 ==> Average_salary=medium 1474    <conf:(0.99)> lift:(1.28) lev:(0.03) [318] conv:(18.65)
 3. experience=5 years 1969 ==> Average_salary=medium 1941    <conf:(0.99)> lift:(1.27) lev:(0.04) [415] conv:(15.28)
 4. joblocation_num=First-tier city experience=3 years 1349 ==> Average_salary=medium 1328    <conf:(0.98)> lift:(1.27) lev:(0.03) [282] conv:(13.8)
 5. joblocation_num=First-tier city job_classification=Senior software engineer 1214 ==> Average_salary=medium 1175    <conf:(0.97)> lift:(1.25) lev:(0.02) [234] conv:(6.83)
 6. experience=3 years job_classification=Senior software engineer 1275 ==> Average_salary=medium 1183    <conf:(0.93)> lift:(1.2) lev:(0.02) [195] conv:(3.09)
 7. joblocation_num=Other First-tier city job_classification=Senior software engineer 1207 ==> Average_salary=medium 1117    <conf:(0.93)> lift:(1.19) lev:(0.02) [181] conv:(2.99)
 8. joblocation_num=First-tier city education=Bachelor 2560 ==> Average_salary=medium 2364    <conf:(0.92)> lift:(1.19) lev:(0.04) [380] conv:(2.93)
 9. education=Bachelor job_classification=Senior software engineer 2283 ==> Average_salary=medium 2106    <conf:(0.92)> lift:(1.19) lev:(0.03) [336] conv:(2.89)
10. education=Bachelor experience=3 years 2657 ==> Average_salary=medium 2428    <conf:(0.91)> lift:(1.18) lev:(0.04) [369] conv:(2.6)
11. joblocation_num=First-tier city 3553 ==> Average_salary=medium 3208    <conf:(0.9)> lift:(1.17) lev:(0.04) [454] conv:(2.31)
12. joblocation_num=Other First-tier city experience=3 years 1440 ==> Average_salary=medium 1300    <conf:(0.9)> lift:(1.17) lev:(0.02) [184] conv:(2.3)
13. job_classification=Senior software engineer 3219 ==> Average_salary=medium 2897    <conf:(0.9)> lift:(1.16) lev:(0.04) [402] conv:(2.24)
14. experience=3 years 3881 ==> Average_salary=medium 3455    <conf:(0.89)> lift:(1.15) lev:(0.04) [447] conv:(2.05)
15. experience=3 years company_scale=50-150 1261 ==> Average_salary=medium 1096    <conf:(0.87)> lift:(1.12) lev:(0.01) [118] conv:(1.71)
```