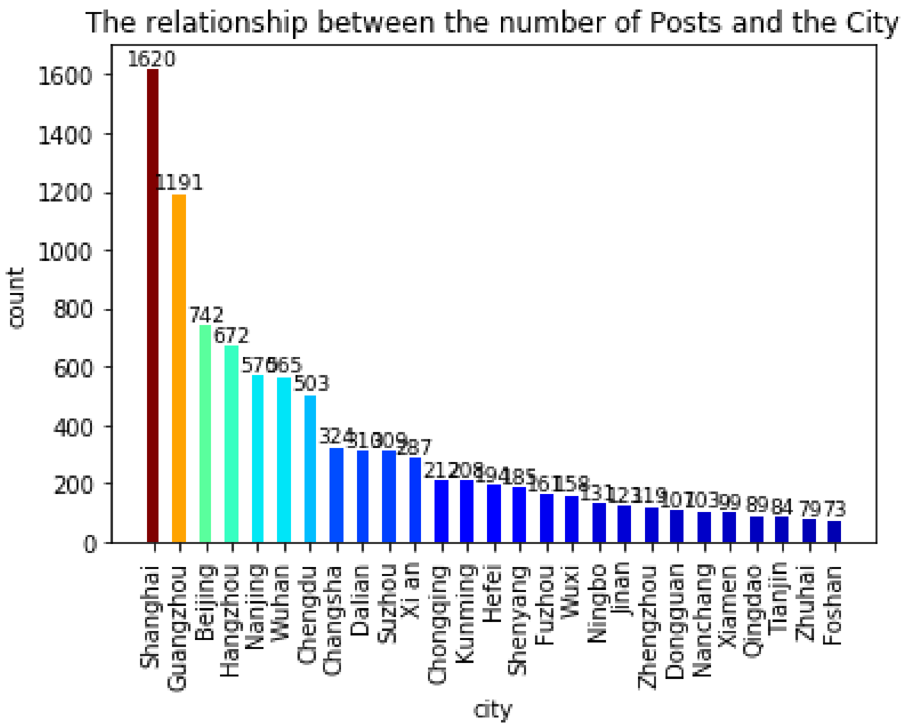
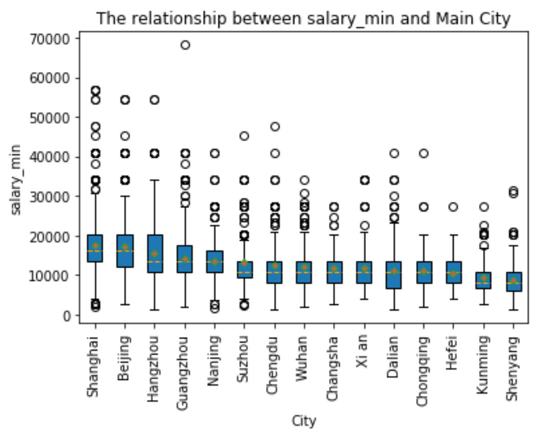
# 1 Data Analysis

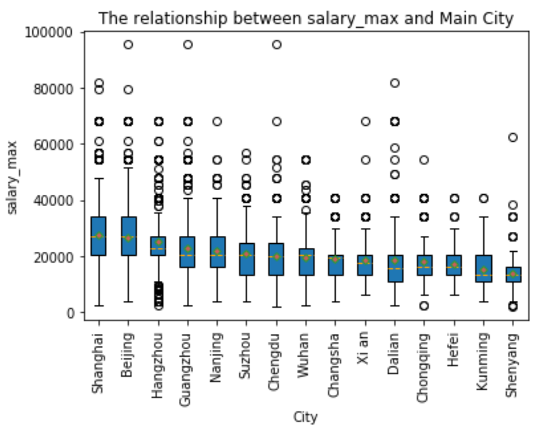
## 1.1 Visualization

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.

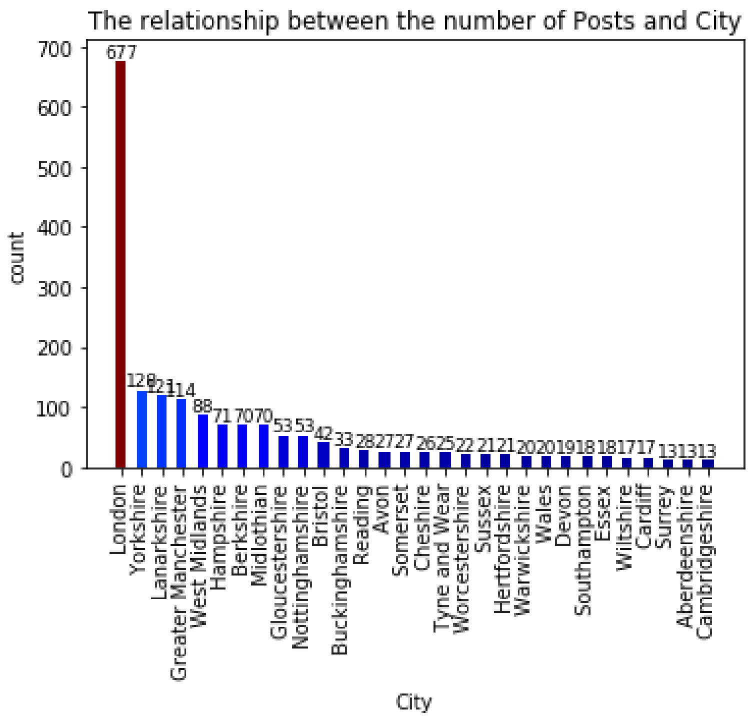


Due to space constraints, this figure shows only the number of Java jobs in the top 25 cities. Through the bar chart, we can directly find that the demand for jobs is closely related to the size of the city. The demand of municipalities directly under the central government is greater than that of provincial 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.



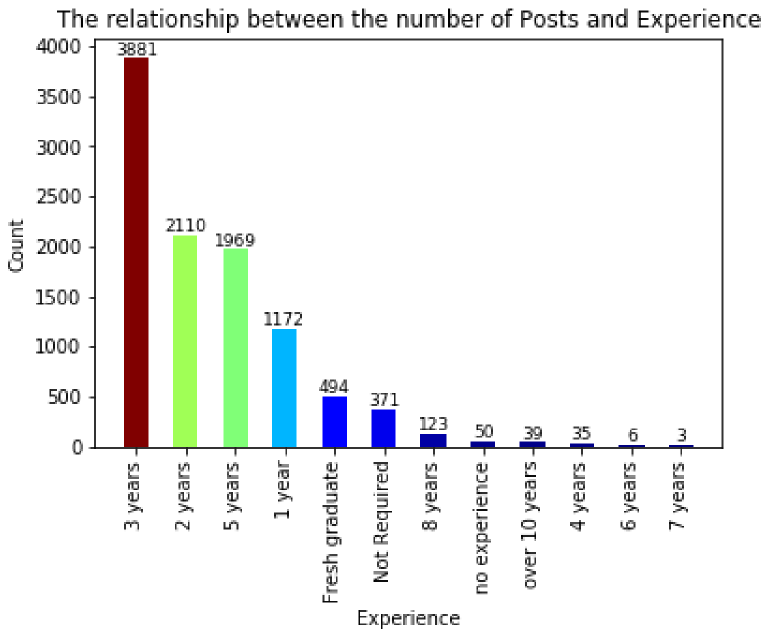


Through the above relationship between the number of jobs and cities, let's take a look at the relationship between salary and cities. Similarly, due to space, only the first 15 cities are showed here. For better and more comprehensive observation, the box diagram is used here. The general situation 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.

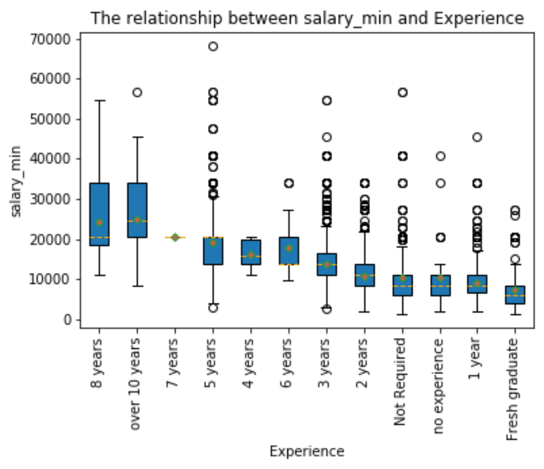


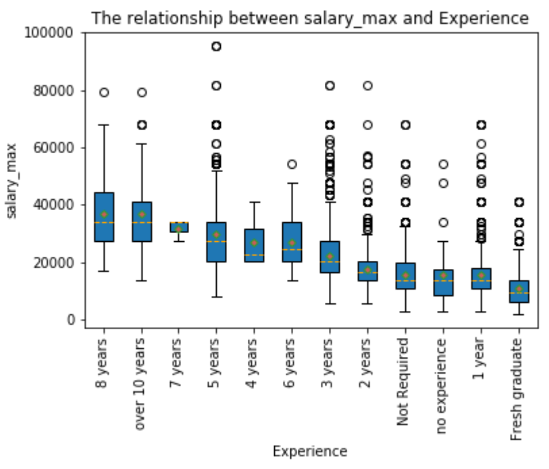
Through the data in the figure, it can be found that Java jobs in the UK have strong aggregation. Jobs are mostly concentrated in the London area. The number of Java jobs in London is five times that of the second place. We can see the agglomeration effect of super big cities here, so if we want to have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

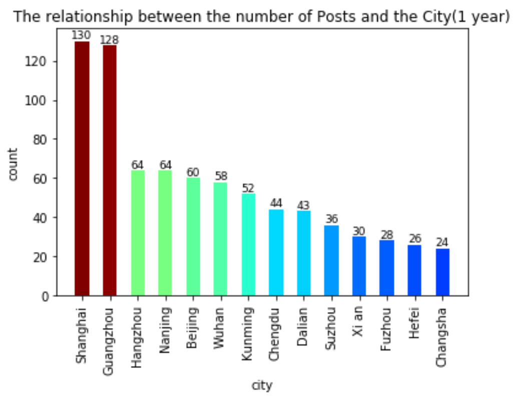


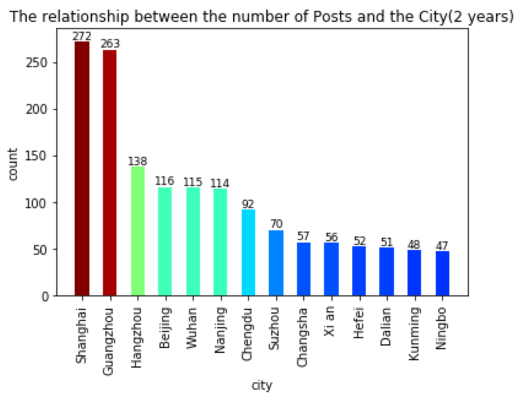
On the whole, the most popular programmers are those with three years of project experience, almost twice as many as those with two years of experience and five years of experience. Therefore, it can be explained that more than three years of project experience is more favored by the company. For fresh 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.

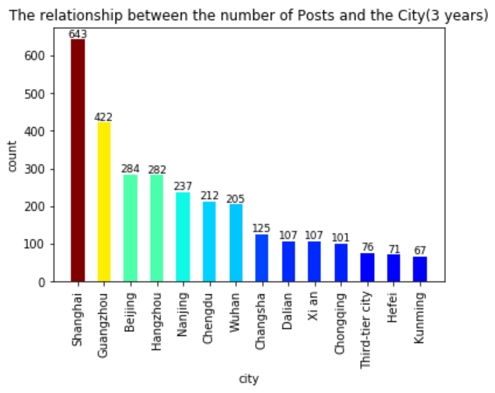


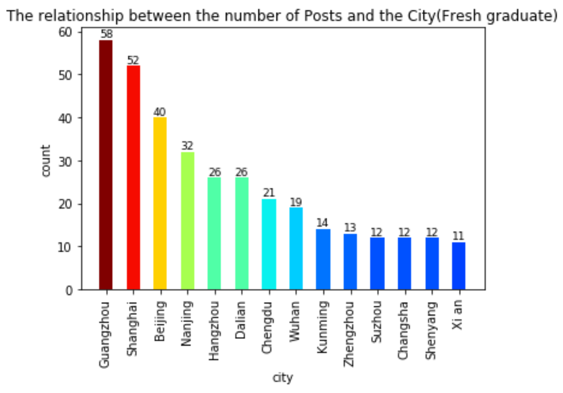


The box chart is used here to more clearly reflect the relationship between salary and work experience. Through the box chart of minimum salary and maximum salary, we can find that the salary of programmers increases with the increase of working experience. Therefore, work experience is positively related 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.

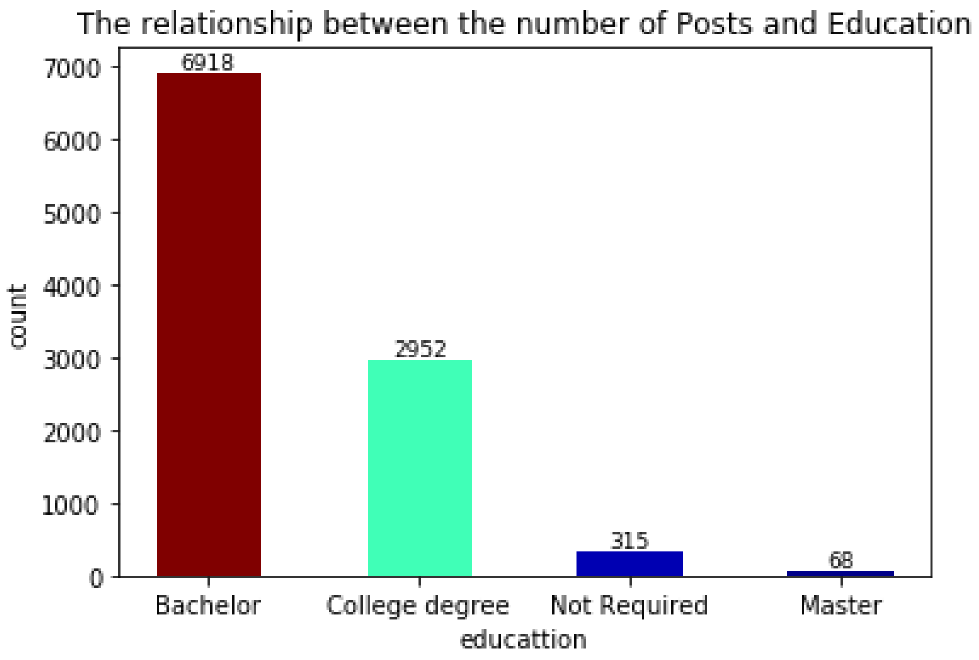




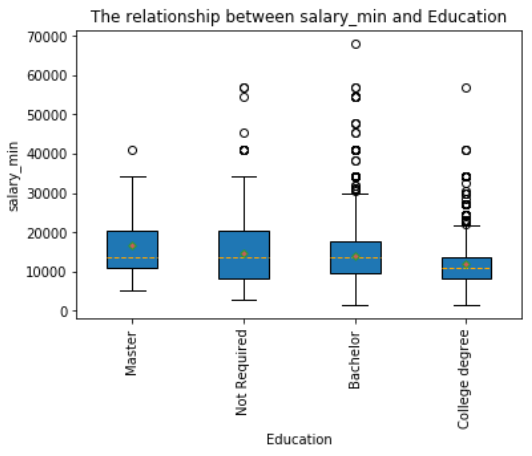


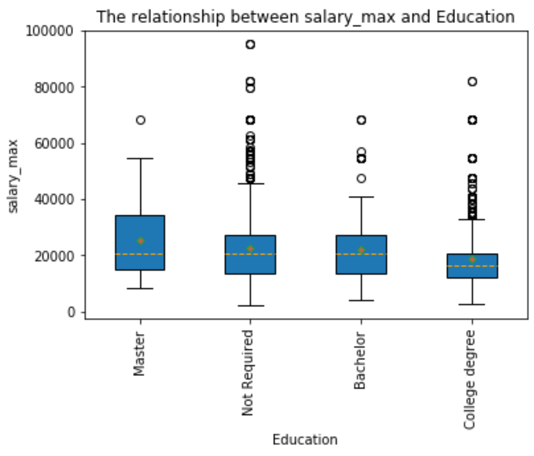


The four subplots are the demand for people with different work experience in different cities. It can be found from the pictures that for young people with fresh graduate and work experience of only 1 year, the demand of Guangzhou and Shanghai is far more than other cities. This explains why Guangzhou's 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.

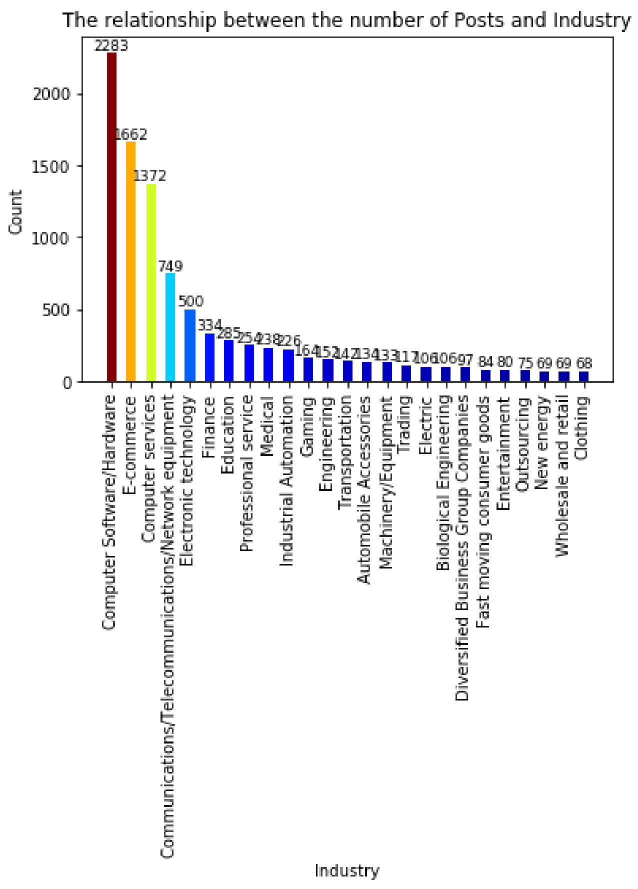


Through the bar chart, you can find that the demand for undergraduate degree is the most and the number is more than twice that of the college degree. Only 2.99% of the positions are for applicants without academic qualifications. And there are only 68 positions with master's degree and above. It can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

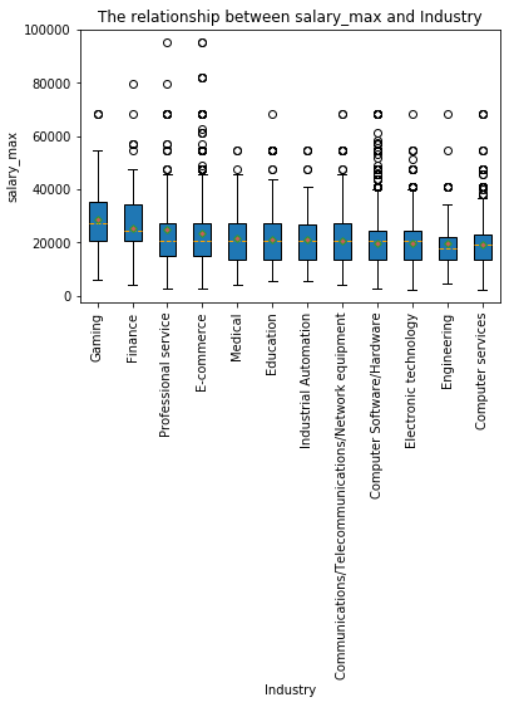


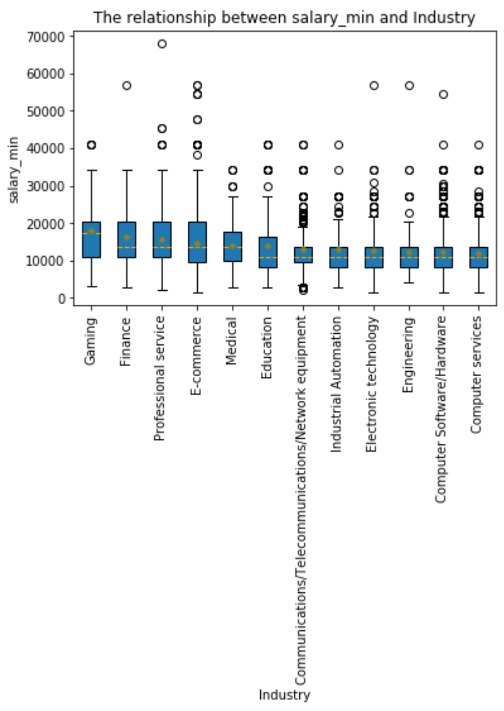


I use the box chart to find the relationship between education and salary. Judging from the situation reflected in the two pictures above, education and salary are positively correlated. In other words, the higher the qualifications of a programmer, the higher the salary he gets. In addition, it can 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.



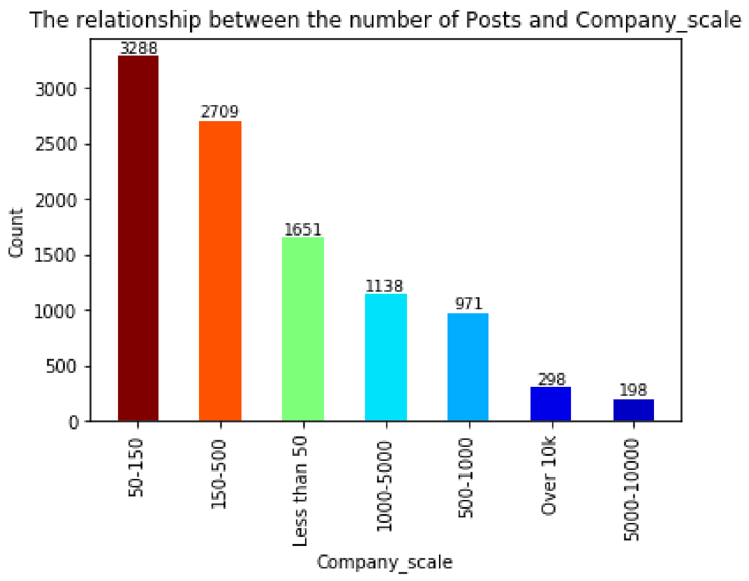
Through the bar chart, we can see that different industries have different requirements for java jobs. Among them, computer-related industries occupy the main position, followed by electronic technology, finance, professional services and other industries. The remaining industries have very little demand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



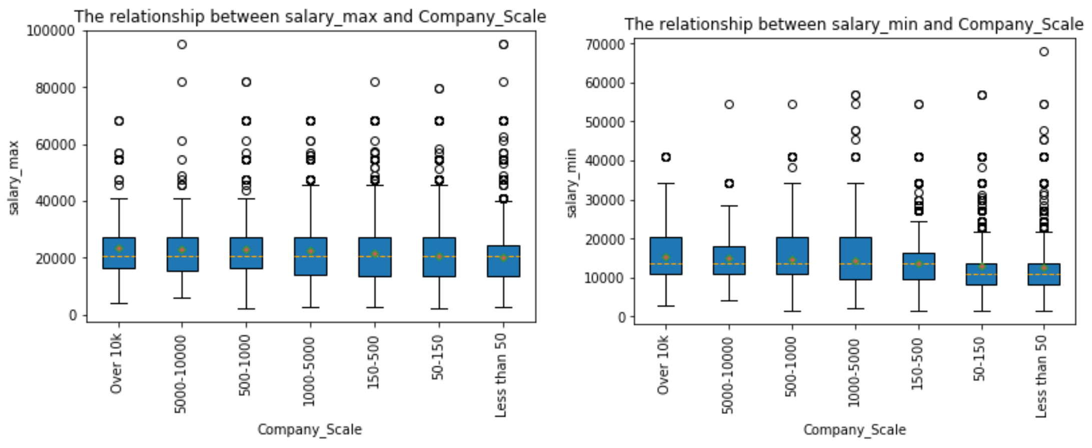


Box charts are used here to observe the salary structure between different industries. Due to space factors, only the top 12 industries are shown here. Unlike the previous distribution, the salary of the computer-related industries here is ranked lower. The top rankings are the gaming industry, financial 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.



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.



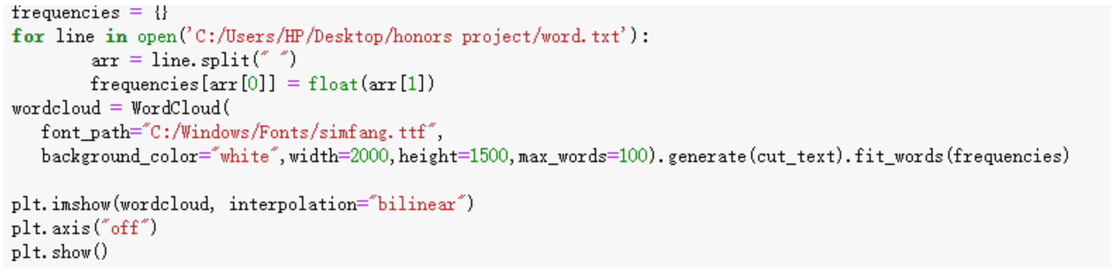
It can be clearly seen from the box chart that the larger the company is, the larger the average salary is. The last three companies are all companies with less than 150 employees. Therefore, this is also in line with the mentality of most people. Everyone hopes to be able to enter a large company to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

## 1.2 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.

| 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 |

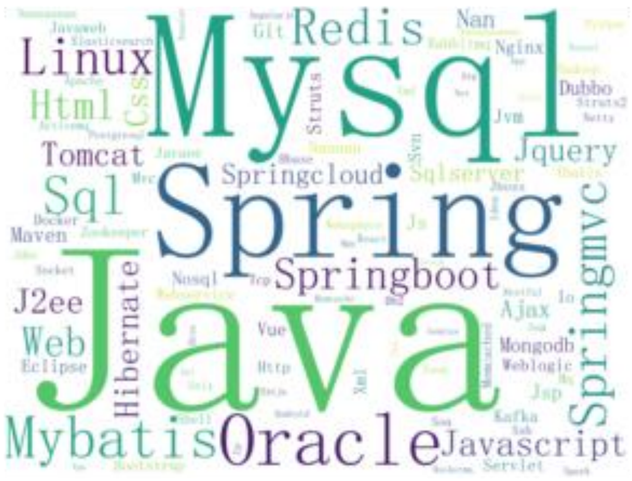
Through the statistical data, it is not difficult to find that Java is the most popular word, but there is no practical significance in this word. Mysql, Oracle and SQL are the most frequently used words. They are all database related. This shows that to be a java programmer, you need to be proficient 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.



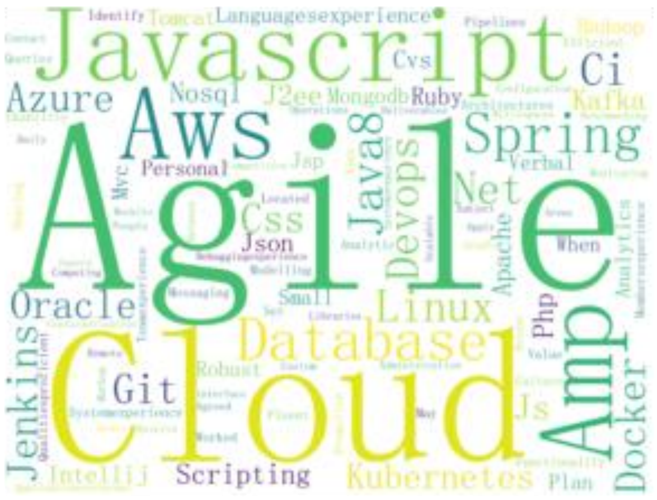
```
frequencies = {}
for line in open('C:/Users/HP/Desktop/honors project/word.txt'):
        arr = line.split(" ")
        frequencies[arr[0]] = float(arr[1])
wordcloud = WordCloud(
   font_path="C:/Windows/Fonts/simfang.ttf",
   background_color="white",width=2000,height=1500,max_words=100).generate(cut_text).fit_words(frequencies)

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

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.



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.



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.

| | |
|---|---|
| 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 |

## 1.3 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.

### 1.3.1 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.

#### 1.3.1.1 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)
```

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)
```

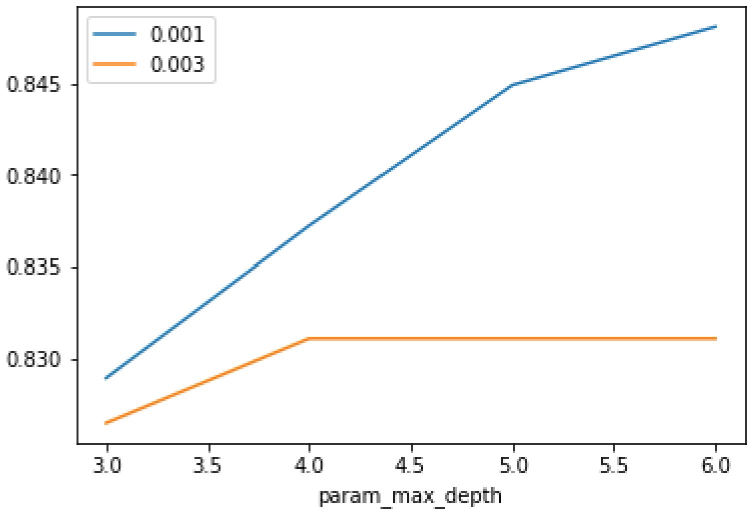
```
              feature        score
3          experience  1836.602122
0     joblocation_num   778.358154
1          Internship   714.775007
2           education   136.766268
5       company_scale    75.665750
7  job_classification    46.492079
6        job_industry    38.261838
4       company_style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

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.

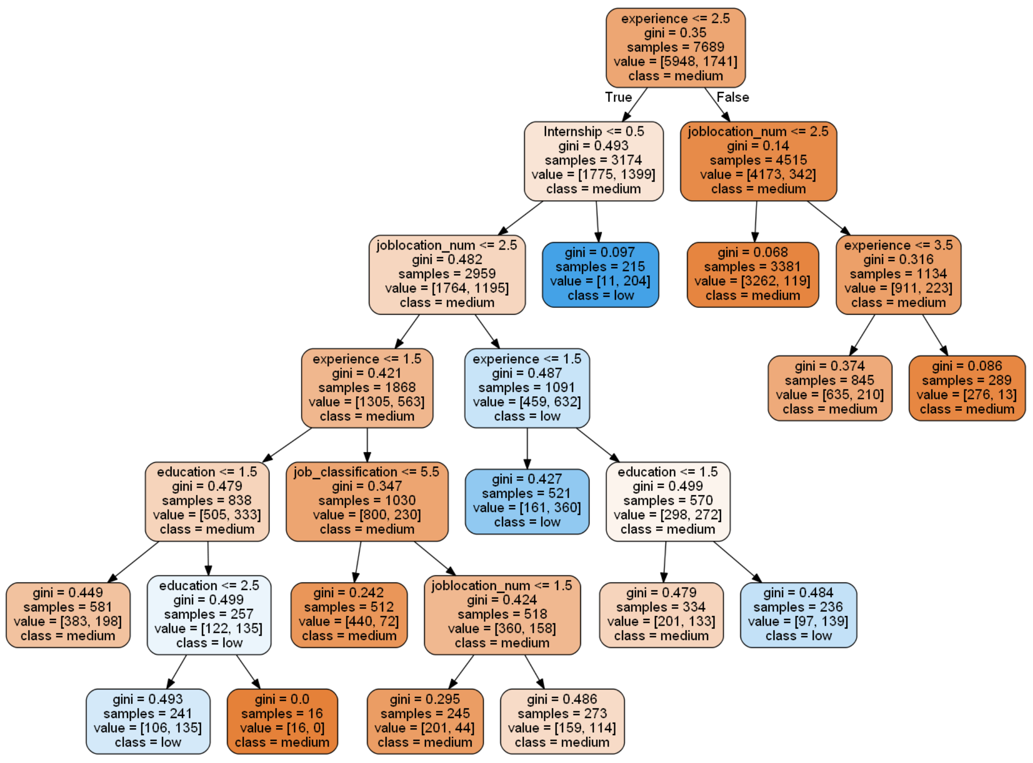
```
clf = DecisionTreeClassifier()
param_grid = {'max_depth': [3,4,5,6], 'min_impurity_decrease': [0.0015,0.003]}
gs = GridSearchCV(clf, param_grid, 'roc_auc', cv = 3)
gs.fit(train_X, train_y)
cv_results = pd.DataFrame(gs.cv_results_)
print(gs.best_estimator_)
```

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



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()
```



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.

#### 1.3.1.2 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.

#### 1.3.1.3 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.

#### 1.3.1.4 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.

| 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.