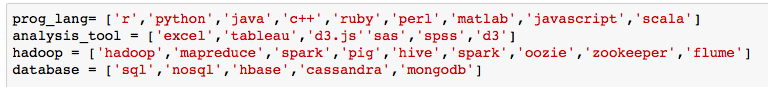
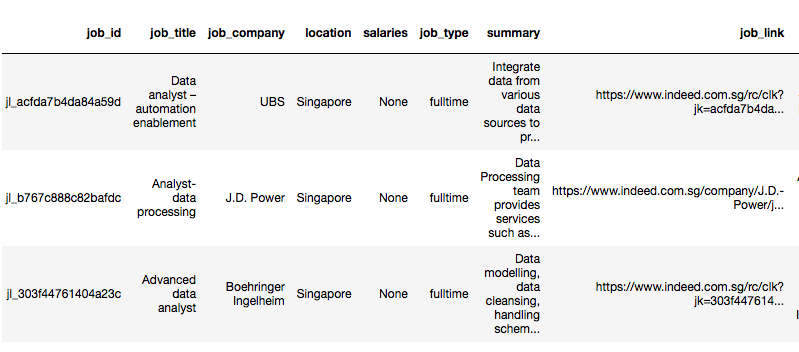
In this project, I will leverage the various skills I learnt from class to scrap dataset of job posting from Indeed.com, structuring the text and then segregate the data by salary below median and above the median. In part 1, I will explain how I collect and clean dataset. In part 2, I’ll explain how I used machine learning algorithms to determine the industry factors that are most important in predicting the salary amounts for data analysts and data scientist job postings.

I conducted my scraping using “requests” and “BeautifulSoup” libraries in python to scrape the data that contains “data scientist” and “data analyst” related job posting in Singapore. The function I defined parsed the HTML on each page of posting to extract the urls of job posting that contain “data analyst”, “data scientist”, “fulltime”, “permenant”, “contract”,”internship”,”temporary”. To avoid time out error, I used for loop to loop through each keywords. Total 2,008 jobs are found. In order to extract the urls, I used a “requests” module to fetch the HTML and convert it to string. Then, using “find” or “findall” function to query data from HTML object. Total 9 items are extracted: Job\_id, Job\_title, company, location, salary, job\_type, link, job description. Job\_id is extracted to remove all duplicated records. From the job description, I applied NLP technique to further categorized the keywords into four category: programming language, analysis tool, hadoop and database skills. The complete vocabulary is shown in below snapshot.

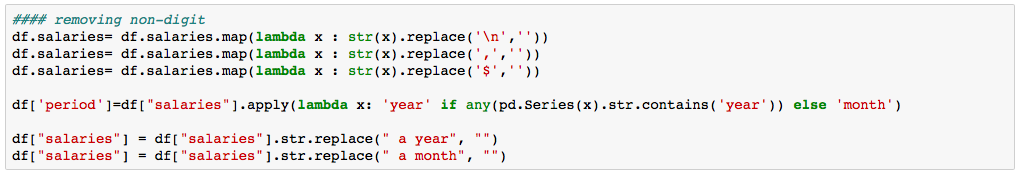


Next, all this information are stored into a Data Frame or csv file.

Before applying model, I needed to perform some initial analysis to extract useful feature and cleaning of scraped data. The following part will explain the technique I applied in my cleaning process. First, I utilized python “pandas” module to read data from csv file, and then used “drop duplicated” function to drop duplicated records. My dataset was reduced from 2,008 jobs to 1,432 unique jobs.



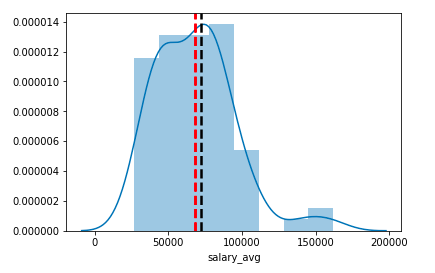
Next, the salary data is formatted. Characters such as “\n” ,”$”,”year”,”month” are removed with “replace” function. And for those salary provided in range or “-” is detected in value, an average value is obtained.



After, the average value is obtained. Next, the jobs are transformed to same scale, annual scale for further analysis. Total 2 jobs are reported in annual scale whereas the rest of the jobs are in monthly scale. For those jobs with monthly scale, I transformed the data to annual scale by multiply salary with 12 multiplier. Once the data is cleaned, then we are ready to do some initial analysis.

Only 77 jobs contained salary information. For those jobs without salary information, I applied logistic regression to predict salary information based on the model I got from these 77 jobs.

For this dataset, the target of interest variable is “salary” and the rest of columns are feature variables. To assist visualization of data, python libraries such as ‘seaborn’ , ‘matplotlib’ are imported.



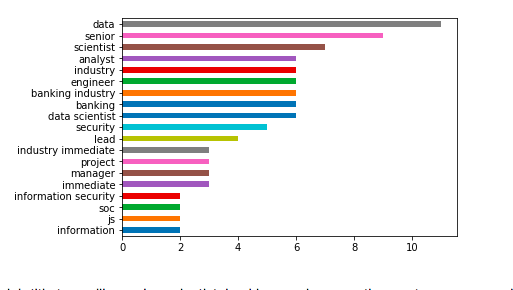
From the distribution plot above, I can tell that the data is in left-skewed distribution because median (black line) is greater than mean(red line). And most of salary are fall in range of 40,000 to 90,000.

In order to frame this as a classification problem, labels are created from the salary column. For those job, the salary is lower or equal to median, the label is “0” whereas the label is “1” for those jobs have salary greater than median. From the bar graph below, 47 out of 77 jobs have salary lower or equal to median.

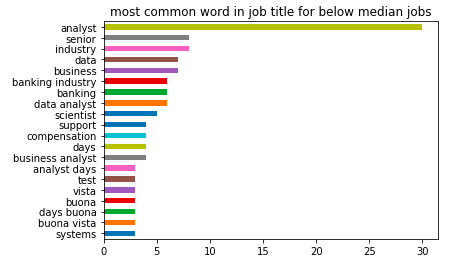


Next, columns such as job\_title, company, location, job type are analyzed in greater depth to examine which feature is the most indicative of salary of job.

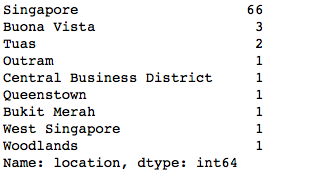
First, ‘job\_title’ values are analyzed with NLP technique to count what’s most frequent word appeared in label “1” or salary greater than median.



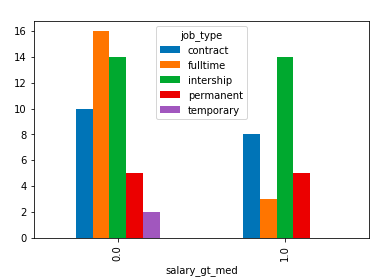
From the bar plot above, job title terms like senior, scientist, banking and engineer are the most common words whereas the job title term like analyst appear most frequent in below median job posting as shown in bar plot below.



For the location column, it doesn’t seem like a good predictor, as it has more than 90% or 66 out of 77 of job postings are in Singapore, therefore, I decided not to include location in modeling.



For the job type feature or bar plot below, temporary job type can be categorized as below median salary while the rest of job types are categorized as above median salary.



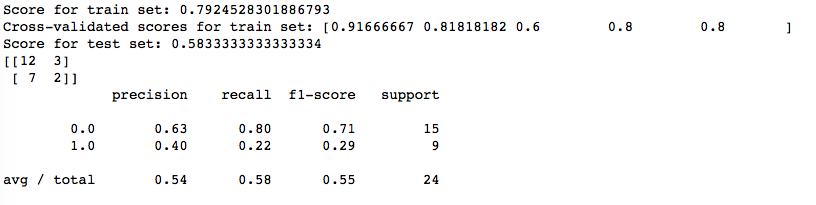
Now all the relevant feature are determined, I can build a predictive model based on those features.

From the analysis above, I decided to build a model with 6 binary features derived from each job posting to predict job posting above-median or below median.

1. Are keywords such as “senior, scientist, banking and engineer” in job title? If yes, then “good job title=1” else 0.
2. Are keywords such as “analyst, business” in job title? If yes, then “bad job title=1” else 0.
3. Does job type contain “temporary” in job type? If yes, then bad job type is 1 else 0.
4. Does job type contain “temporary” in job type? If no, then good job type is 1 else 0.
5. Does ‘analysis’ skill in skill column? If yes, then good skill is 1 else 0.
6. Does ‘analysis’ skill in skill column? If no, then bad skill is 1 else 0.

Now my dataset only left with 6 binary variables (good job title, bad job title, bad job type, good job type, good skill, bad skill) and 1 target variable (salary), I applied logistic regression, random forest, decision tree and SVM models. Before applying the model, the dataset is splitted. 70 % of dataset is for training our model whereas the rest of 30% (test data) are used to validate our model. Below are classification report for four different models:

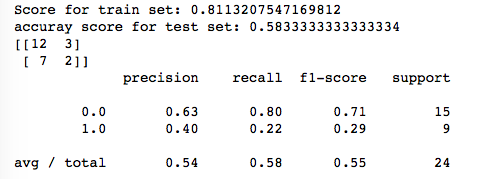
1. Logistic Regression



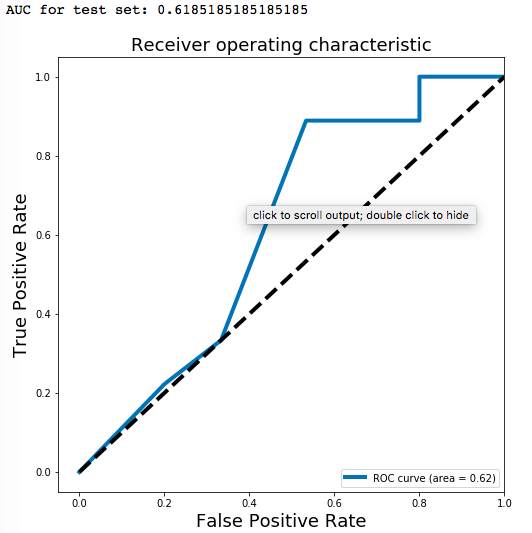
The logistic model score quite well for train set, 79% accurate with a range between 60% to 92% . For the test set, the logistic model score quite well, 58% accurate.

For the confusion matrix, 7 (33%) below median job salary were incorrectly predicted as above median job, whereas 2 (40%) above median job salary were incorrectly predicted as below median job. Therefore, the model performs well for jobs with below median job salary as compared to above median job salary.

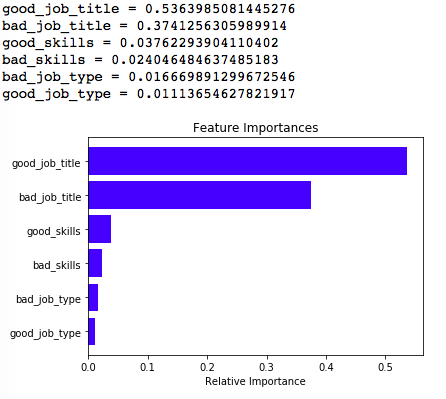
1. Random Forest



The score for random forest is higher than logistic regression, ~0.811 for train set. Same results (test score, confusion matrix, precision score, recall, f1) are obtained for test data.

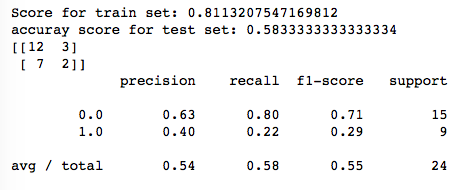


Next, AUC graph is plotted to determine if the model is better than baseline or not. The baseline is in black dashed line. From the plot, we can see that, the model is better than baseline model.

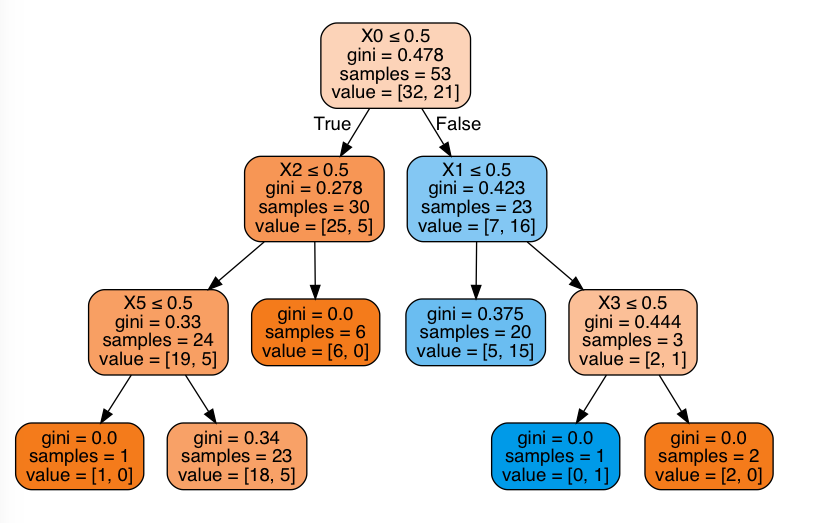


The good thing for random forest is there is a module called ‘feature important’ which can tell you which feature is important feature in predicting salary. From the bar plot above, good job title (such as: senior, scientist, banking, engineer ) is an important feature.

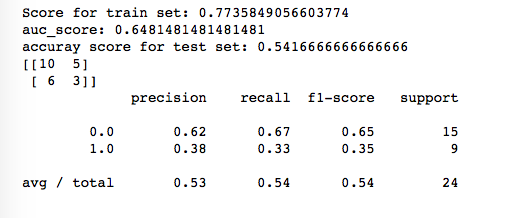
1. Decision Tree



Decision Tree returned same results as random forest models for train score, test score and classification report. This might due to small dataset issue, as I only have 77 job postings contained salary information .

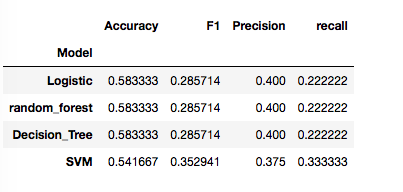


1. SVM



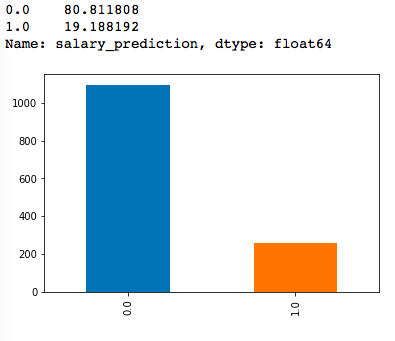
Score for train set, test set, classification report of SVM model is slightly lower than previous 3 models.

Next, results of all four models are compared in table below.



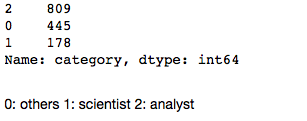
From the result, it is clear that logistic, random\_forest and decision tree models giving same scores on accuracy, F1 score, precision, and recall metric. This might due to small dataset issue, as I only have 77 job posting which contained salary information.

Hooray, we got the model. Next, I applied the decision tree model to predict those job postings without salary information.

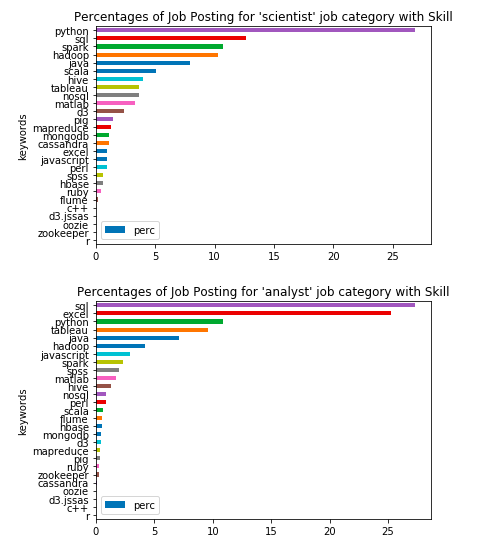


Total 1355 job postings are without salary information. 80% of jobs were predicted to have salary below median whereas only 19.1% were predicted to be above median after with decision tree model.

From the above analysis, we can tell that ‘job title’ is the most important feature to predict whether salary of the job posting is above median or below median. Next, we can analyze the data set further to identify what’s skill is required to distinguish job title from data scientist and data analyst. To begin, I categorized jobs into three categories: data scientist, data analyst or others by finding keywords such as ‘scientist’, ‘analyst’ in job title. From the result, there are total 809 jobs are categorized as ‘analyst’ whereas only 178 jobs are categorized as ‘scientist’.



Next, I applied NLP to examine frequencies of words such as ‘python, sql, spark, hadoop, java,, scala’ and etc in job description. Looking at plot below, it seems that python is definitely the most commonly requested skills for data scientist job and SQL language are used far more commonly than nosql language. For the data analyst related jobs, excel are the most common requested skill.

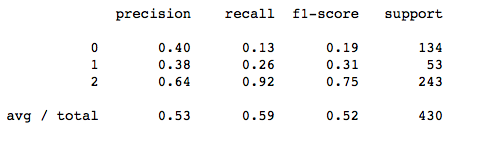


In summary, here are the top 4 requested skills for:

1. Scientist: python,sql, spark, hadoop
2. Analyst: sql, excel, python, tableau

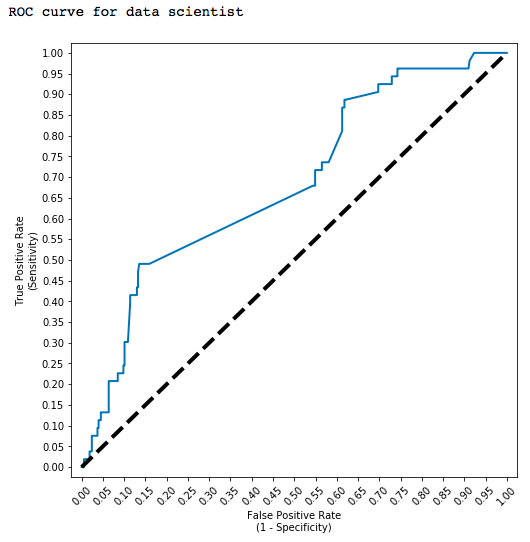
Next, logistic regression and Multinomial NB models are applied to predict job title based on the frequency of these keywords in job description. The dataset is splitted to 70% train data and 30 % test data.

1. Logistic Regression
   1. The test score is around 0.59
   2. Classification report

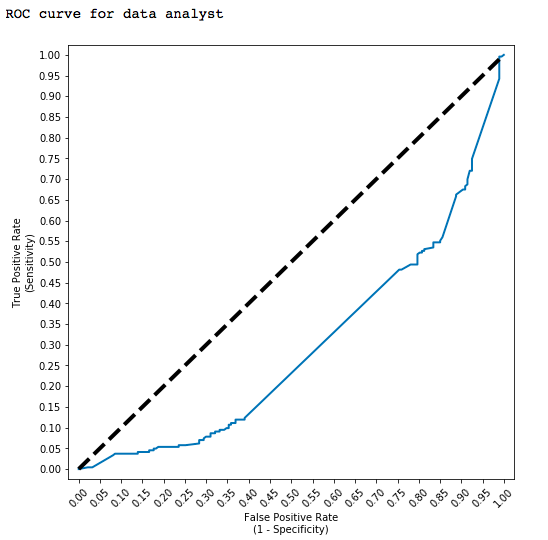


There are total 809 analysts, whereas only 178 data scientist jobs. Therefore, the dataset is imbalanced, therefore, it is not recommended to look at F1 or precision score. By looking at the ROC curve below, the plot is greater than 0.5. The model performs better than baseline model.

* 1. ROC plot
     1. Data scientist

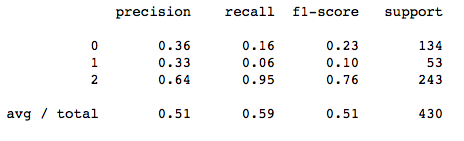


* + 1. Data Analyst



2. MultinomialNB

1. Test score is around 0.59 which is about the same as logistic regression
2. Classification report:



Precision score is slightly lower than logistic regression model.