**Assumption**

Data provided by recruiters at the job website is inconsistent.

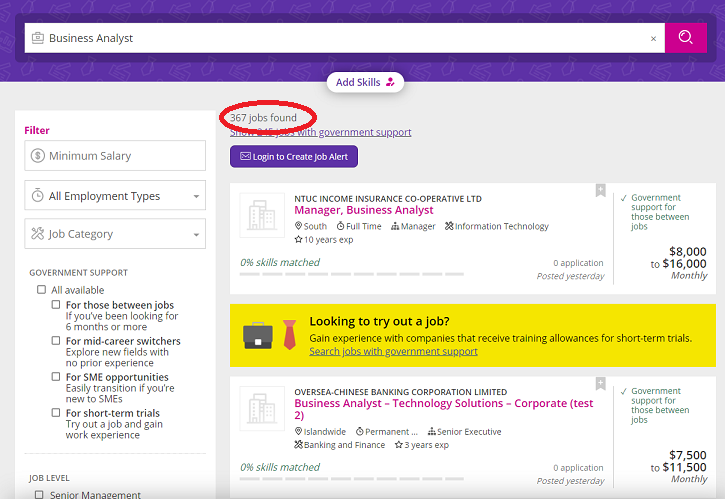
Data scientist, Data analyst roles are very inconsistent. Especially Data analyst roles can be Data Scientist when we read the roles, responsibilities.

Employment type column is very inconsistent. For example , values can be permanent, temporary, contract on the same field. Care has been taken to impute correctly to the best of our knowledge

**Scraping**

Mycareersfuture.sg as compared to the rest ( Indeed, LinkedIn etc. ) is because the salary is indicted for most of the job. Besides that, every job search is serialized with a number suffix.

<https://www.mycareersfuture.sg/search?search=Business%20Analyst&sortBy=new_posting_date&page=0>



Meaning if 367 jobs are found and each listing has 20 jobs. We need to scrape 367/20 number of pages of the particular job title – in this case it’s 19 page ( after rounding up ) . We’d be using selenium, scrappy and X-path. Selenium is used because the page is running JavaScript and a 3s wait command is required in order for the page to load. Scrappy to scrape the page and X-path is used in tandem to extract necessary information from the HTML tags.

After obtaining the index url of each job search, we need to scrape the individual URL of each job. Hence, in this example we’d have 367 individual URL.

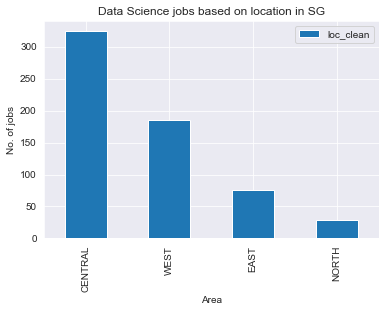
Next, we need to scrape the rest of the relevant job titles ( i.e. Data Scientist, Data Analyst ) and this will lead us to a lot of more the individual URL. With the consolidated URL, we will scrape individual information from each of the job titles. During extracting information from each of the fields as pointed by the X-path, we need to try our best to clean to a certain degree in order that lesser cleaning would be required when we load the DataFrame with pandas.

**EDA and Data Cleaning**

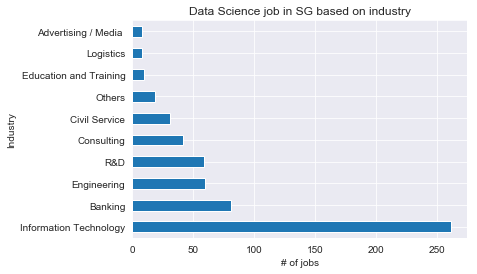
First, we need to remove duplicates based on job\_id and remove columns that are totally irrelevant such as job\_id , date\_close, url and date\_scrape.

Next, we need to manually sort the job title, employment type and of each job location as it’s extremely messy with the aid of functions, if, elif and else.

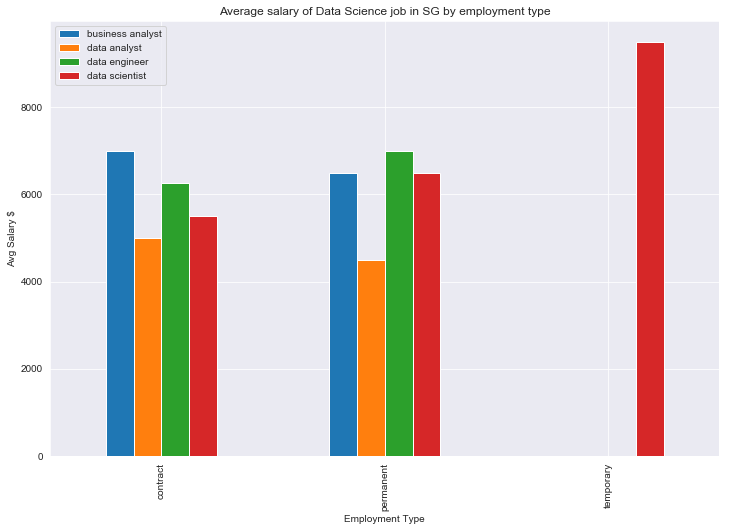
The most important column which is the salary needs to be feature engineered as it has a range , some values are null and required to be inputted , and an average salary is feature engineered.



We found out that Central and West has the most companies that require Data Science expertise.

****

This is because there’s a lot of IT in the West ( i.e Google and Block 71 ) and a lot of banks at Central



Nothing surprising as data analyst is the most junior role as compared to the rest. Data Engineer tipped the Data Scientist as there is a higher demand for them as compared to Data Scientist. Data Engineer is required as they need to ‘get’ the data for Data Scientist. The outlier for temporary is actually a mislabeled consultation job.

Q1 . Distinguish between low and high salary based on median.

Firstly, we calculated the mean of 0.52 where we need to beat and we quickly ran logistic regression and obtain a cross val score of 0.63 and precision of 0.65. Precision metric is used because we care about how many jobs with high salary are accurately predicted . ( Precision = TP / TP + FP )

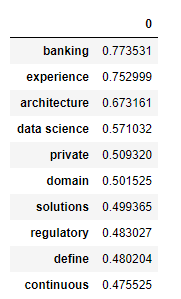
Test score is 0.68. Before fitting we need to dummify some categorical columns and before train\_test\_split we need to use stratify = y as we need to preserve the percentage of each class.

Next we ran a model based on Logistic Regression using columns such as Roles, Jobs Description and Skills tags. We actually combined these 3 columns and we perform Count Vectorizer using English stop words , ngram\_range of (1,2 ) and transform it to TFIDF. Cross val score increased to 0.67 and to precision to 0.73.

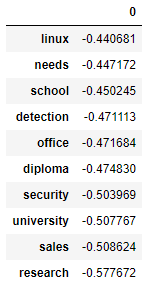
We also did grid search with Random Forest, Naïve Bayes with the same features and also combined it with the dummified features but at the end we selected the Logistic Regression with the 3 NLP columns.

Some of the features that contributed to higher are the following keywords:

It provides hint that banking, experience, data science and domain ( expert ) contributed to it



Those that contribute negatively such as diploma, university and school. These keywords indicates those that are fresh from education.

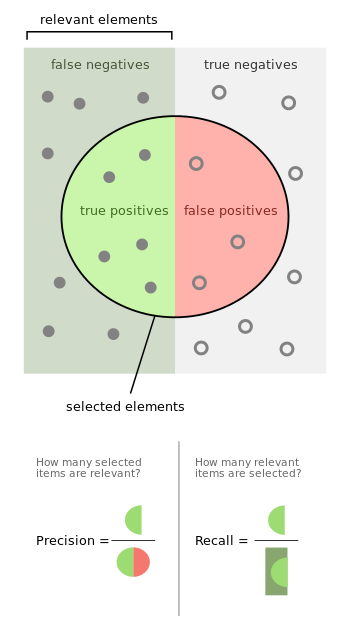


Q2. Distinguish between data analyst(minority) and the rest

As this dataset is imbalanced, hence we are unable to model in a normal manner. We need to upsample the data so that it’s more balanced any metrics used will not be appropriate. For our case, class 1 ( is\_data\_analyst) is the one that we’re interested has only 20 labels as compared to the class 0 which has 563. Upon inspecting the classification report, precision and recall is ZERO. This proves the fact that we can’t perform normal modelling.

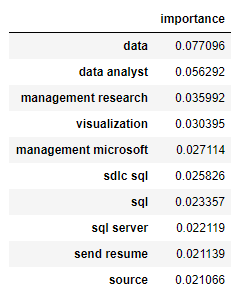
Hence, we use SMOTE to upsample the minority class. Meaning SMOTE will assist in ‘balancing’ the labels for the minority by increasing it to the level of the majority class. Then we can do our modelling as per normal.

We selected Random Forest as we’re looking at recall metrics for this ‘problem’ , we need to detect all the ‘Data Analyst’ . Precision is also important but it ranks second as compared to Recall. It’s analogous to a disease scenario, we want to ensure that we detect all personnel that has the disease ( data analyst ) even if they do not have it ( the other class ).



Hence, Random Forest model is selected as the recall is 0.75 and precision is 1.00. Feature importance include the following:

I agree with it because of the keywords of ‘data analyst,’ ‘vizualization’ and especially SQL.



**Future improvements**

Scrape using ‘data’ field in order to obtain more data and filter out accordingly.

Combining data from other job sites( indeed, linkedin ) and impute median salary based on current job site

Use PCA for dimentionality reduction.

Use LSTM for classifying as it’s more accurate.