Studying the Impact of Job Descriptions on Data Science Salaries

Team Humilty

CHLOE NEO TZE CHING ENG JING KEAT GOH WEI LUN GLENN
KANEKO YOSHIKI TAN WEI KEONG

2022-11-18

Abstract

Millions of workers worldwide are no longer satisfied with just a fair paycheck, instead, they wish achieve job satisfaction through meaningful career opportunities. This project aims to interpret the relationship between data science salaries and keywords in job descriptions. The data is cleaned by preparing job description text for analysis by converting it into a document-term matrix. The models used are Multiple Linear Regression (MLR), Random Forest (RF), XGBoost (XGB), Multivariate Adaptive Regression Splines (MARS), and Partial Least Squares (PLS), where their performances will be assessed in relation to one another. We found that MLR has the lowest Root Mean Square Error (RMSE) of 16.2. Thus MLR is the best performing model that accurately predicts salary, and we identified keywords used in this model.

Contents

1 Introduction to the problem

1.1 Literature review

A rapid shift towards digitalisation of businesses has radically changed the employment landscape in Singapore, which means Singaporeans need to keep up with the changes if they wish to stay competitive at work (My Skills Future, 2021). Employers in Singapore are starting to place an emphasis on skills rather than education (Tan, 2021). New hires today are assessed not just by their qualifications and work history, but also by their soft skills as there are a variety of soft skills in demand (The Straits Times, 2021).

According to a new report by Instant Offices, 73% of Singaporean workers are dissatisfied with their jobs (Arora, 2022). When asked if they planned to change jobs over the following six months, 31% of respondents responded "yes" (Chong, 2022). Millions of workers worldwide are no longer willing to return home with just a fair paycheck. They prefer to know how well they are progressing towards a meaningful career, which is wellness, freedom, security, and experience at work, so as to achieve job satisfaction. As a result, job seekers should take all these factors into consideration when applying for a job.

These factors can be broken down and identified as keywords in job descriptions. Keywords are crucial to job adverts because they allow job seekers to narrow their search related to a role, skill, or industry for suitable employment (Alexander, 2019). Suitable individuals are more likely to find the job post when employers and recruiters add key terms and phrases that are relevant to a particular role. This increases the percentage of successfully matching job seekers with their ideal jobs.

1.2 Objective

This project aims to interpret the relationship between data science salaries and keywords in job descriptions. Doing so will give insight to keywords that have a significant impact on salaries. Job seekers will also be able to set healthy salary expectations based on keywords that reflect their skill set and values. Identified keywords can also give insight to areas that job seekers can look to upskill in.

2 Data set

2.1 Description of data set

Click here to access our data set. The owner scrapped the data from Glassdoor and pre-processed it. It contains 41 variables, including the average data science salary measured in thousand, and has 742 observations.

Below is a sample of our dataset:

```
## Rows: 742 Columns: 42
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): Job Title, Salary Estimate, Job Description, Company Name, Locatio...
## dbl (25): index, Rating, Founded, Hourly, Employer provided, Lower Salary, U...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Table 1: A sample of our dataset.

Avg Salary(K)	company_txt	Job Location	Age	Python	spark
72.0	Tecolote Research	NM	48	1	0
87.5	University of Maryland Medical System	MD	37	1	0
85.0	KnowBe4	FL	11	1	1
76.5	PNNL	WA	56	1	0
114.5	Affinity Solutions	NY	23	1	0
95.0	CyrusOne	TX	21	1	0

2.2 Exploratory data analysis

2.2.1 Feature selection

Since this project focuses on job description and its relationship to salary, we will only keep the "Job Description" variable and the "Avg Salary(K)" variable. In the next section, we process our "Job Description" variable so that it is usable for our models.

2.2.2 Data cleaning

We clean our "Job Description" variable by creating a new variable "cleaned_text" that stores only alphanumeric characters and converts all the text to lowercase. We then only keep the "Avg Salary(K)" and "cleaned_text" variables.

2.2.3 Data visualizations

To get a better understanding of the data we are working with, we perform some data visualizations. First, we take a look at the distribution of the average salary (in thousands).

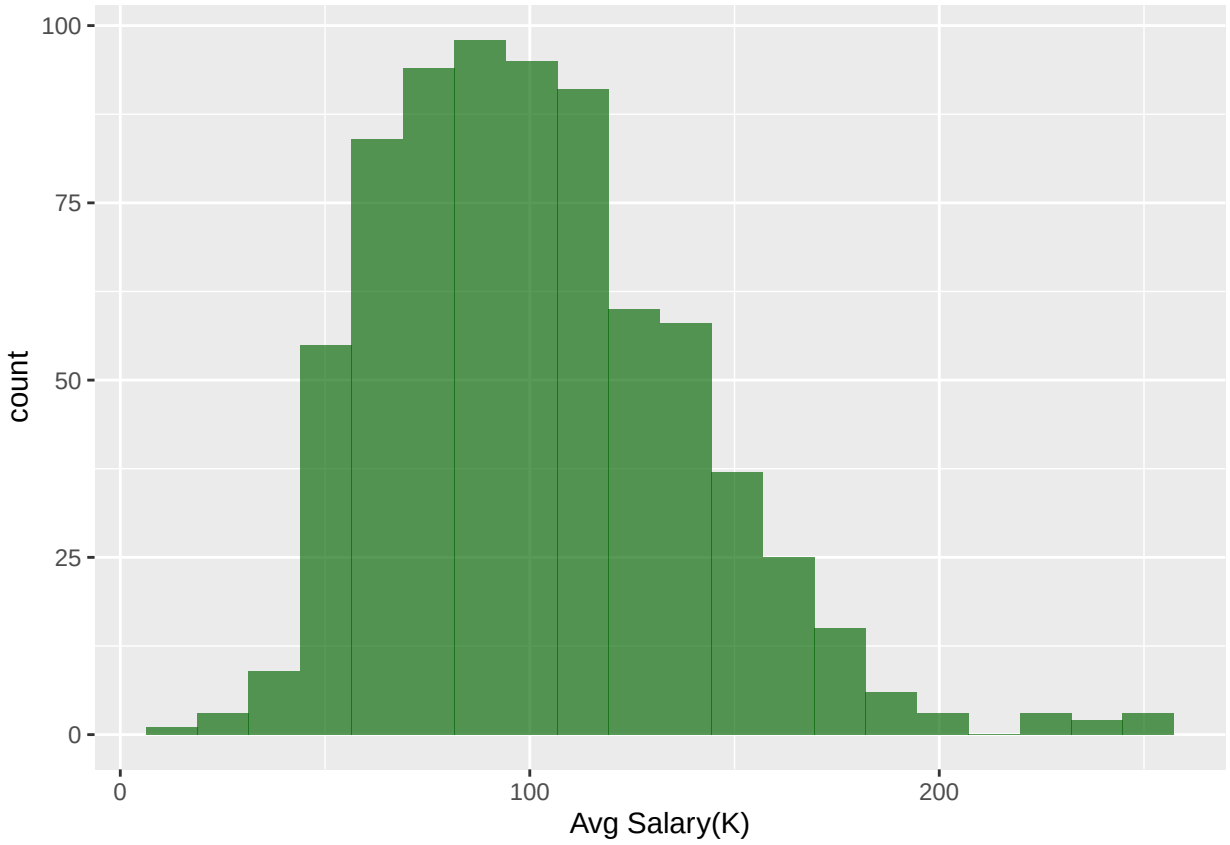

Figure 1: Histogram of salaries

From our histogram, we see that our average salary (in thousands) roughly follows a normal distribution. Hence we will predict the raw annual salary without further transformations.

Secondly, we visualize the job descriptions with a word cloud after removing words that occur less than 40 times, and stopwords such as "the", "he" etc. as these are common words that do not store any informational value in our analysis. Below we show 100 of the most common words.

Figure 2: Job description word cloud

2.2.4 Document-term matrix

The document-term matrix (DTM) is a table reflecting the frequency of each word where the column names represent words and the row names represent documents, i.e., the reviews.

We create a DTM for our job descriptions, setting minimum word frequency to 40.

```
Setting minimum word frequency to 40, we retain 1238 words out of the original 10314 words.
```

```
Our document-term matrix consists of 742 job descriptions against 1238 words present in our vocab after
```

We create a new data frame for our DTM and produce a sample of it below.

Table 2: A sample of our DTM.

data	machine	python	analytics	science
11	1	1	0	1
7	2	2	7	1
18	2	1	3	1
6	4	1	0	7
8	2	1	1	2
16	1	1	1	0

3 Modelling

3.1 Feature selection using LASSO

We will use LASSO regularization to select features to prepare our data for three of our models: Multiple LInear Regression, Random Forest, and XGBoost. The last two models use their own methods of feature selection, so we will not use features selected by LASSO for those two models, but the entire dataset instead.

The LASSO procedure is as follows:

1. Use cross-validation to find the best value of lambda (approx. 0.359)
2. Store our variables selected by LASSO
3. Prepare our final dataset to be used for MLR, RF, and XGB

```
knitr::kable(head(D_final[, c('Y_salary', 'predictive', 'education', 'learn',
                              'machine', 'experience', 'analytics', 'support')]),
             caption = "A sample of our final dataset.")
```

Table 3: A sample of our final dataset.

Y_salary	predictive	education	learn	machine	experience	analytics	support
72.0	0	2	0	1	3	0	0
87.5	3	1	0	2	7	7	2
85.0	2	0	0	2	7	3	1
76.5	1	0	1	4	9	0	1
114.5	0	0	1	2	1	1	3
95.0	0	0	0	1	5	1	0

We also prepare our training and test data to be used, and the dimensions are as follows:

```
Dimensions of the training set are 585 355
```

```
Dimensions of the test set are 157 355
```

From this, we can see that what started with over 10,000 unique words in our job descriptions has been narrowed down to 354 words.

3.2 Models

3.2.1 Multiple Linear Regression

The first model we use is Multiple Linear Regression, using the 354 features selected by LASSO.

`RMSE: 16.21`

After using cross-validation to find our lambda value for regularization, our MLR model's RMSE value (rounded to two decimal places) is 16.21, the lowest we will achieve in this project.

Below is a bar plot of the top 20 variables with the highest absolute coefficients. The sign of their coefficients reflect a positive or negative relationship with the predicted salary.

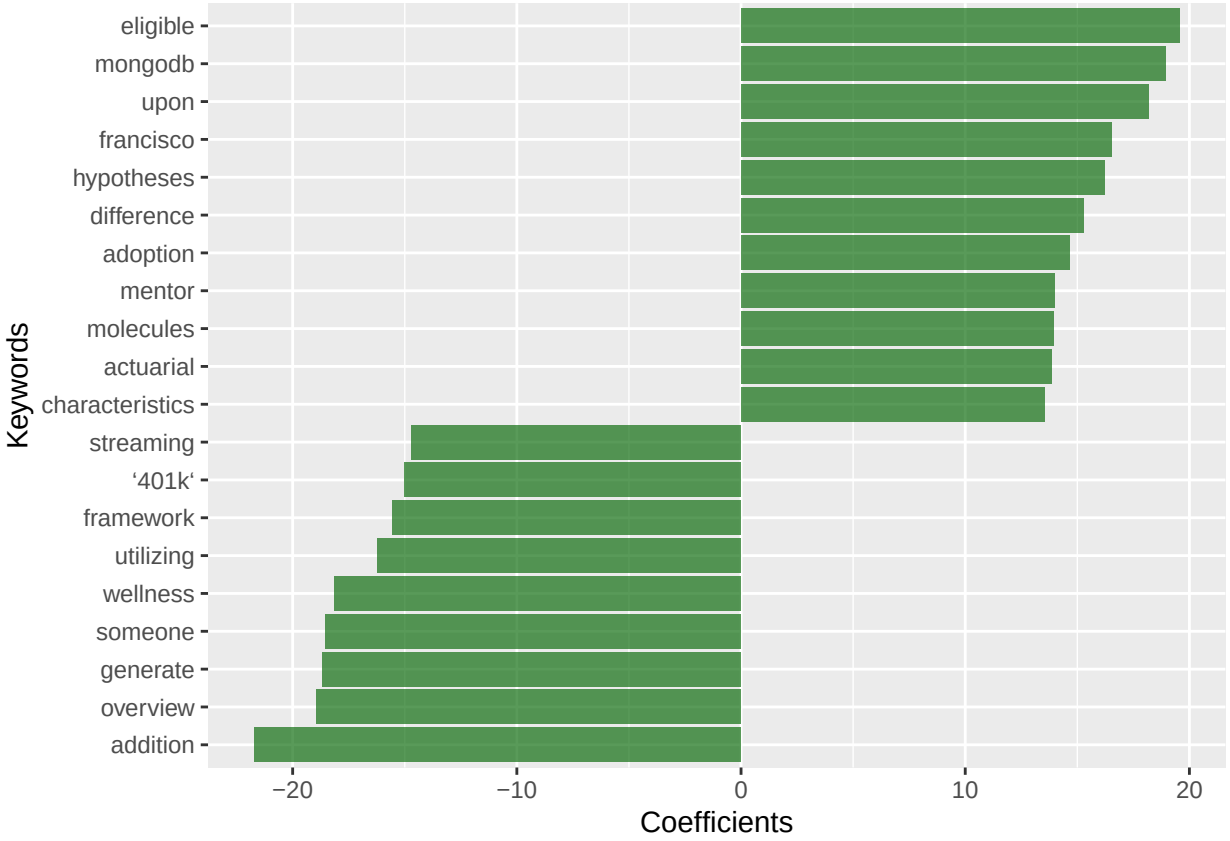

Figure 3: Top 20 variables according to absolute coefficients

This plot shows that the presence of words such as "eligible" and "molecules" in the job description will result in a higher predicted salary, whereas words such as "streaming" and "wellness" in the job description will lower the predicted salary.

3.2.2 Random Forest

The next step is to visualize our data using random forest decision trees. We first train the data to obtain its optimal hyper-parameters. We then perform grid search on the optimal hyper-parameter values to minimize the out-of-bag error.

The optimal values of the hyper-parameters are:

	mtry	splitrule	min.node.size
5	178	extratrees	5

So we retrain the model with the selected hyper-parameters, and fit our Random Forest model on the training data set.

Our RMSE for the Random Forest model (rounded to two decimal places) is as shown below:

```
RMSE: 21.59
```

The bar graph below shows the top 20 most important variables in this prediction. Based on the bar graph, the top 5 most important variables are "machine", "analyst", "give", "phd", and "scientists".

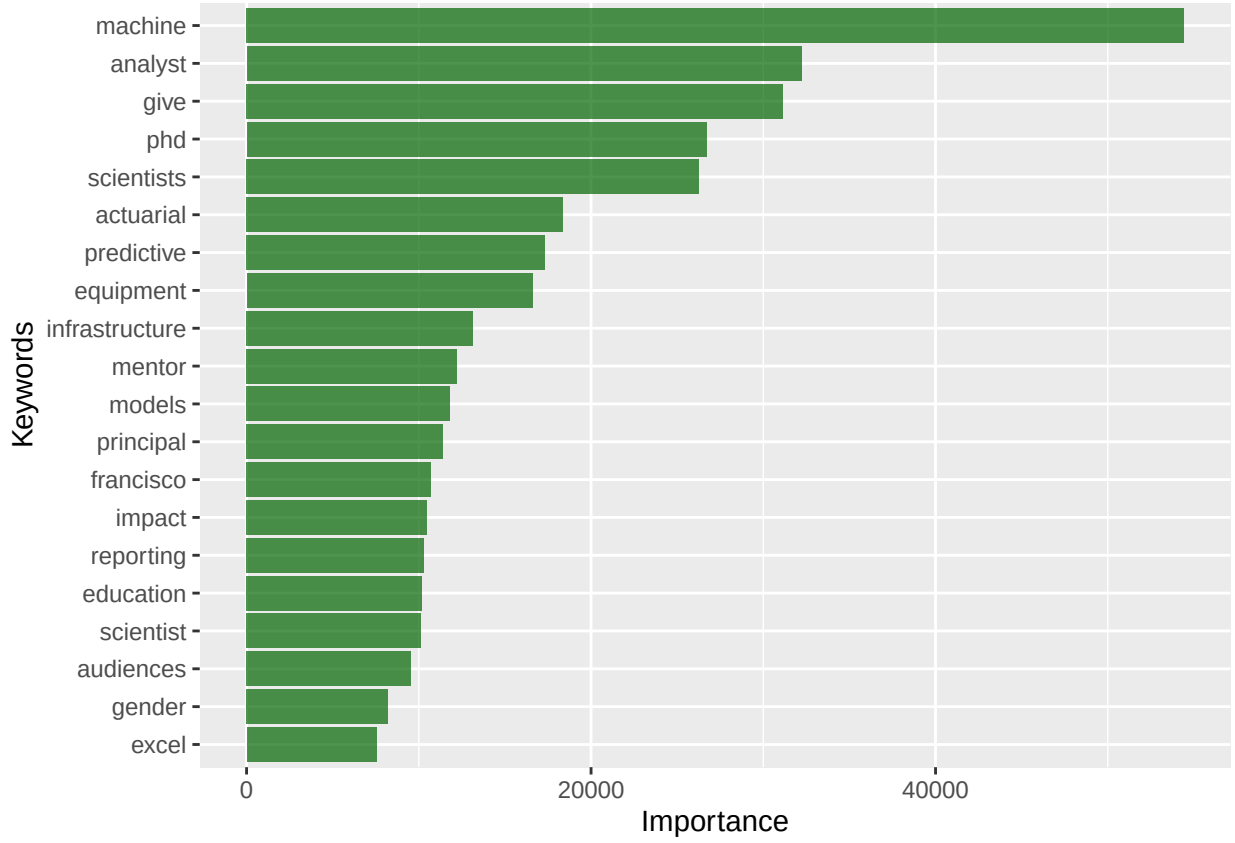

Figure 4: Random forest top 20 most important variables

3.2.3 XGBoost

Like random forests, gradient boosting machines are based on decision trees. However, while random forest builds an ensemble of deep (i.e complex) trees independent of one another, gradient boosting builds shallow trees sequentially, where each tree learns and improves from the previous tree. This is achieved by starting with a weak model and subsequently boosting its performance by allowing each new tree to focus on training data where the previous tree had the largest errors (or residuals) in prediction. Each tree in the sequence is thus fitted according to the residuals of the previous tree.

Moreover, it computes the second-order gradients, i.e. second partial derivatives of the loss function, which provides more information about the direction of gradients and how to get to the minimum of our loss function while gradient boost uses the loss function of simple decision tree model as a proxy to minimize the error of the overall model.

The objective function (loss function and regularization) at an iteration t that we need to minimize is as such:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

where $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$

The XGBoost objective is trained in an additive manner. l is a differentiable convex loss function that measures the difference between prediction $\hat{y}_i$ and target y_i. The second term Ω penalizes the complexity of the model. From the equation, we greedily add the f_t that improves our model the most. Second-order approximation can be used to quickly optimize the objective in the general setting.

We create our XGBoost model from the caret library, using hyper-parameters as shown below:

in our model, we have used these hyperparameters:

1. *gamma*: Pseudo-regularisation hyperparameter that controls the complexity of each tree.
2. *nrounds*: Number of decision trees in the final model
3. *eta*: Learning rate; determines the contribution of each tree on the final outcome and also how quickly the algorithm goes down the gradient descent.
4. *max_depth*: Depth of each tree
5. *min_child_weight*: Minimum number of observations in terminal nodes; controls complexity of the trees
6. *colsample_bytree*: subsample of columns used for each tree (repeated for every tree)
7. *subsample*: subsampling ratio of training data for growing trees to prevent over-fitting

The hyperparameters were tuned using 5-fold cross validation and grid search to find the best model, and we arrived at the optimal values for the hyperparameters.

Our chosen values for our hyper-parameters (through cross-validation) and our RMSE for the XGBoost model (rounded to two decimal places) are as shown below:

eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds
0.21	3	0.04	1	1	1	650

```
RMSE: 16.59
```

Overall, XGBoost gave an RMSE value of 16.59.

We extracted the 20 most important features (words) from the XGBoost model. The bar graph below shows the top 20 most important words in this model. Based on the bar graph, the top 5 most important variables are “machine”, “give”, “education”, “analyst”, and “phd”.

```
Attaching package: 'xgboost'

The following object is masked from 'package:dplyr':

    slice
```

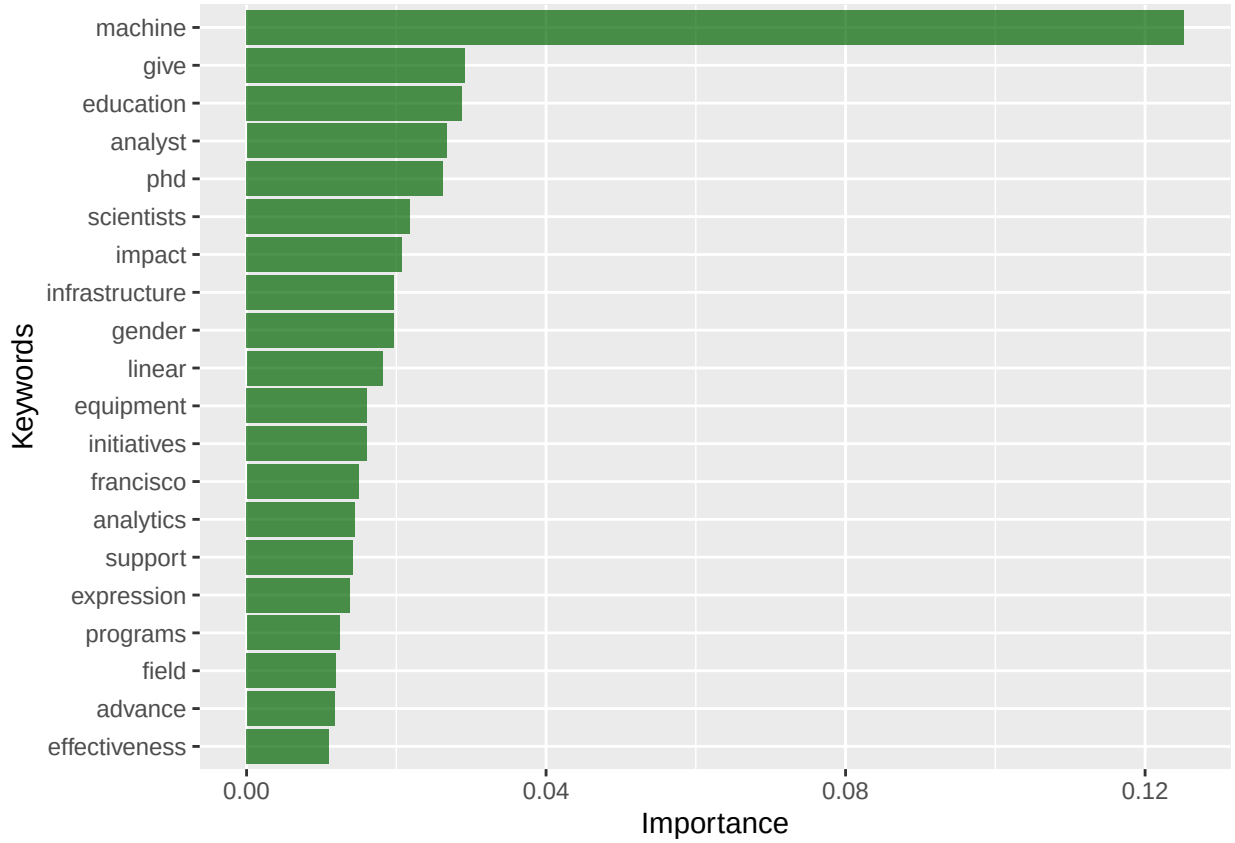

Figure 5: XGBoost top 20 most important variables

3.2.4 Multivariate Adaptive Regression Splines (MARS)

We used multivariate adaptive regression splines (MARS) (Friedman 1991) model here, it is an approach that automatically generates a piecewise linear model that serves as an understandable stepping stone into non-linearity after learning the notion of multiple linear regression.

By evaluating cutpoints (knots) similar to step functions, MARS offers a practical method to capture the nonlinear relationships in the data. This method evaluates every data point for every predictor as a knot and builds a linear regression model using the candidate feature(s).

Consider non-linear, non-monotonic data where $Y = f(X)$. The MARS method will initially search for a single point within a range of X values where two distinct linear relationships between Y and X provide the lowest loss. The outcome is referred to as a hinge function $h(x - a)$, where a is the cutpoint value.

For example, if $a = 1$, our hinge function is $h(x - 1)$ such that the linear models for y are:

$$y = \begin{cases} \beta_0 + \beta_1(1 - x)\,, & x < 1 \\ \beta_0 + \beta_1(x - 1)\,, & x > 1 \end{cases}$$

After the first knot is identified, the search for a second one begins, and it is discovered at $x = 2$. Now the linear models for y are:

$$y = \begin{cases} \beta_0 + \beta_1(1 - x)\,, & x < 1 \\ \beta_0 + \beta_1(x - 1)\,, & 1 < x < 2 \\ \beta_0 + \beta_1(2 - x)\,, & x > 2 \end{cases}$$

This process is repeated until several knots are identified, leading to the creation of a highly non-linear prediction equation. Even if using a lot of knots could help us fit a particularly excellent relationship to our training data, it might not perform well to unseen data. Once all of the knots have been found, we may systematically eliminate knots that do not significantly improve predictive accuracy. This is pruning process, and we may use cross-validation to determine the optimal number of knots.

We will use the following packages. First of all, we divided the dataset into training dataset and test dataset:

```
Dimensions of the MARS training dataset are 512 1239
```

```
Dimensions of the MARS test dataset are 230 1239
```

MARS model have two hyper-parameters: the maximum degree of interactions and the number of terms retained in the final model. To achieve the optimal combination of these tuning parameters, we must conduct a grid search that minimize the error of prediction.

We built up a grid with 30 different combinations of interaction complexity (degree) and the number of terms to include in the final model (nprune).

We performed required grid search by using 10-fold cross-validation to determine our parameters:

degree	nprune	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	11	31.87122	0.3140381	25.10864	2.810016	0.0945591	2.344299

The backwards elimination feature selection process used in MARS models seeks for reductions in the generalized cross-validation (GCV) estimate of error when each additional predictor is introduced to the model. The variable importance is based on this overall reduction. MARS effectively accomplishes automated feature selection since it will automatically include and remove variables throughout the pruning phase.

After pruning, a predictor's significance value is 0 if it was never used in any of the MARS basis functions in the final model. There are only 9 features have importance values greater than 0, whereas the other features all have importance values of zero since they were excluded from the final model.

We also kept track of how the residual sums of squares (RSS) change when terms are added. However, we noticed that there is no much difference between these two measures.

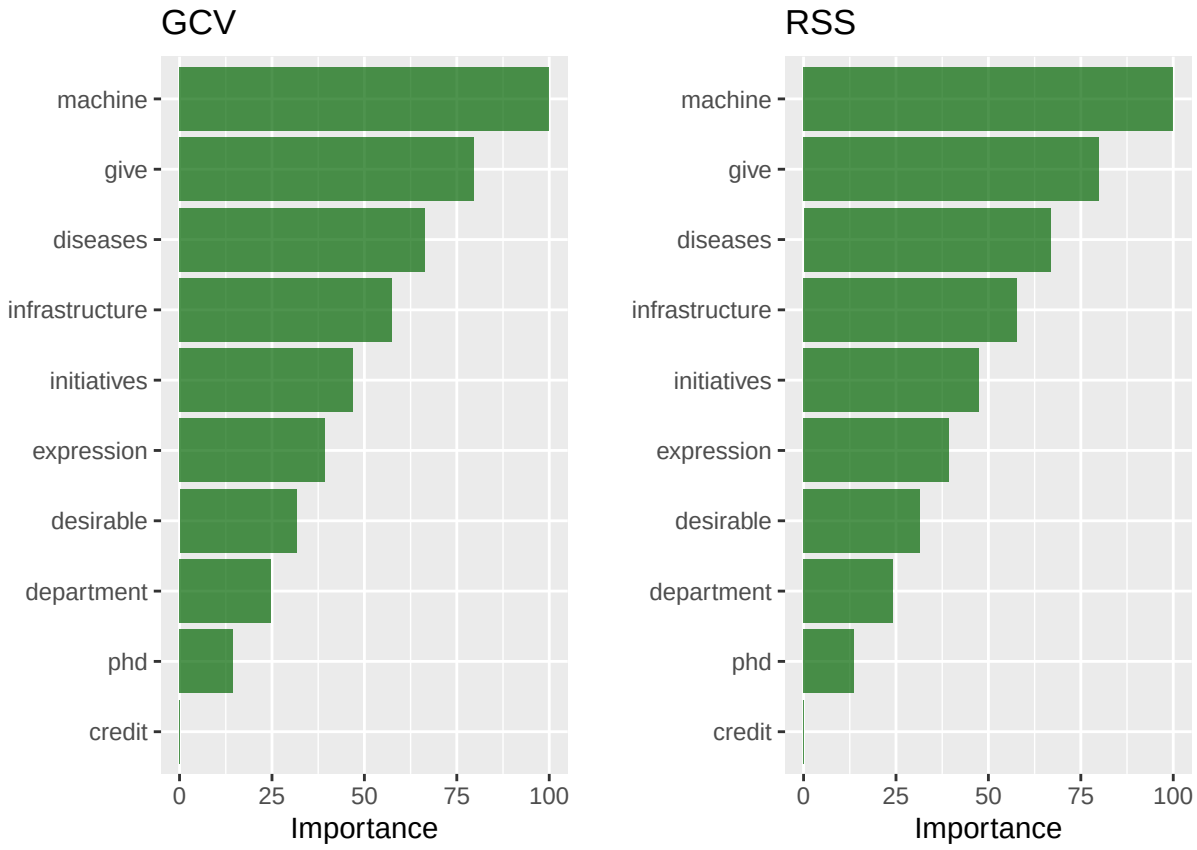

Figure 6: MARS top 20 most important variables

We used the optimal hyper-parameters to train the model and then calculated the RMSE:

```
RMSE: 30
```

3.2.5 Partial Least Squares (PLS)

Partial Least Squares (PLS) is a common technique to analyse relative importance when the data includes more predictors than observations. It is an useful dimension reduction method which is similar with principal component analysis (PCA).

We do a regression against the response variable inside the narrower space created by mapping the predictor variables to a smaller set of variables. The response variable is not taken into account during the dimension reduction process in PCA. PLS, on the other hand, seeks to select newly mapped factors that best describe the response variable.

Similar to MARS, we divided the dataset into training dataset and test dataset first:

```
Dimensions of the MARS training dataset are 522 1239
```

```
Dimensions of the MARS test dataset are 220 1239
```

The hyper-parameter for PLS model is the number of components used in the model (ncomp) .We conduct a grid search that minimize the prediction error to achieve the optimal hyper-parameter. The grid search was conducted by 10-fold cross-validation, and we used the optimal hyper-parameter to train the model and calculated the RMSE as well:

```
RMSE: 23.88
```

The barplots below show that 'addition', 'generate', 'hypotheses', 'framework' and 'characteristics' are positive predictors, while 'streaming', 'francisco', 'difference', 'mongodb' are negative predictors:

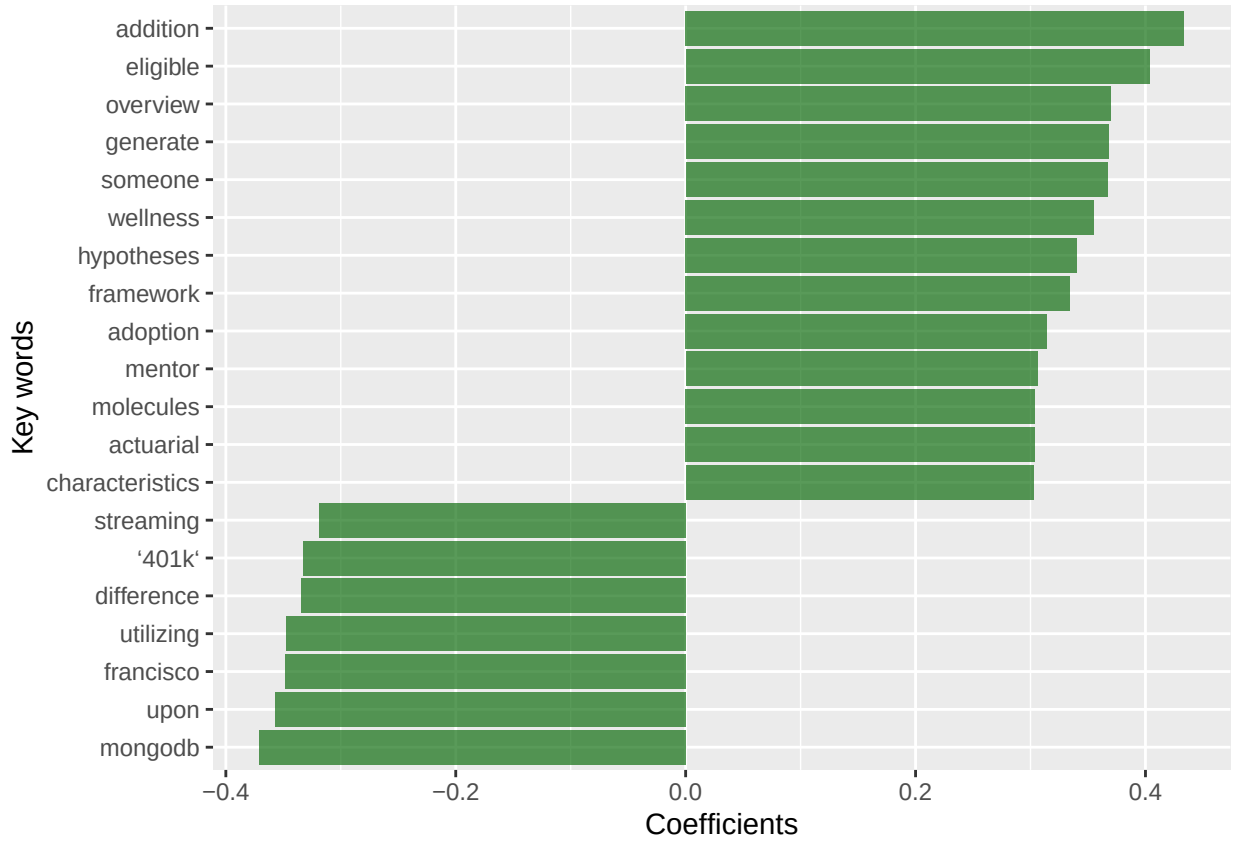

Figure 7: PLS top 20 most important variables

3.3 Summary of results

Table 7: Accuracy of models.

Model	RMSE
MLR	16.21
Random Forest	21.59
XGBoost	16.59
MARS	30.00
PLS	23.88

4 Conclusion

Recommendation engines are a commonly found solution applied to job search portals, such as the Singapore government's national jobs portal, MyCareersFuture. However, more can be done to bridge the gap between job seekers and job satisfaction. In this project, we have produced models that predict the expected average salary earned given a job description. Our best performing model, Multiple Linear Regression, can be used to help workers set expectations of salary based on keywords they value as important in search of a job. This will help job seekers focus on searching for their desired job role rather than focus on maximizing salary earned, which will hopefully increase job satisfaction.

MLR also identified key terms such as "eligible", "mongodb", "upon", "francisco", and "hypotheses". Some of these keywords may not seem to make sense, and understanding the importance of these words is unclear. Some of these words, on the other hand, give insight into areas that job seekers can focus on, be it upskilling (for example, learning MongoDB), or narrowing their job search to sectors such as actuarial science or molecular chemistry in order to maximize their potential salary.

Based on RMSE, our best model uses Multiple Linear Regression, and it has identified key terms that have a significant impact on data science salary. However, Multiple Linear Regression does not identify the same important features (words) as our other models. Comparing identified important keywords between models was also not feasible within this project as each models used different methods of ranking the importance of variables. Different models also identified different keywords as important. Further research is required to better understand the difference between models and why they identify vastly different features as important.

5 References

1. Alexander, L. (2019). The importance of keywords in job ads. SEEK. Retrieved November 16, 2022, from https://www.seek.com.au/employer/hiring-advice/the-importance-of-keywords-in-job-ads
2. Arora, P. (2022, August 29). What's keeping Singapore employees unhappy at work? - ETHRWorldSEA. HR News Southeast Asia. Retrieved November 7, 2022, from https://hrsea.economictimes.indiatimes.com/news/employee-experience/whats-keeping-singapore-employees-unhappy-at-work/93833788
3. Chong, C. (2022, May 17). Nearly 1 in 3 workers in S'pore plans to change employers in first half of 2022: Survey. The Straits Times. Retrieved November 4, 2022, from https://www.straitstimes.com/singapore/jobs/nearly-1-in-3-workers-in-spore-plan-to-change-employers-in-first-half-of-2022-survey
4. My Skills Future. (2021, June 21). 5 Crucial Skills You Need to Remain Employable in the Wake of Covid-19 | Myskillsfuture.gov.sg. MySkillsFuture. Retrieved October 14, 2022, from https://www.myskillsfuture.gov.sg/content/portal/en/career-resources/career-resources/education-career-personal-development/5-crucial-skills-you-need-to-remain-employable-during-covid.html
5. The Straits Times. (2021, December 22). It's a match: How skills-based hiring fits in the future of work. The Straits Times. Retrieved October 14, 2022, from https://www.straitstimes.com/singapore/jobs/its-a-match-how-skills-based-hiring-fits-in-the-future-of-work
6. Tan, E. (2021, April 14). S'pore employers prioritise skills over education, experience: LinkedIn survey. The Straits Times. Retrieved October 14, 2022, from https://www.straitstimes.com/singapore/jobs/singapore-employers-prioritise-skills-over-education-experience-linkedin-survey