Project 2: Draft

Econ 1680: Machine Learning, Text Analysis, and Economics

Lena Kim

April 7, 2024

github link: https://github.com/wkim31/Econ1680-TA-Project

1 Introduction

The job market is overcrowded. Google receives millions of applications for its positions a year, and by that point it is simply impossible to read each and every application and resume to its fullest. As such, many companies now rely on algorithms to sift through and pick the resumes of best fit, motivating questions of algorithmic fairness in the hiring process- are the right people being hired, or are they simply being chosen based on certain words in their resume?

In fact, a New York Times coverage of the ever-increasing reliance on resume algorithms has even led to identification of certain keywords and skills to include in your resume (sectioned off by position type) to make it past this resume sifter (Weed (2021)). As a Post article states, including even one of these key words can mean the difference between getting hired and being discarded (Abril (2023).

Taking inspiration from this trend, I aim to approach it from the other side of the table and accurately predict the salary levels of a job based on keywords used in its job description. With some companies remaining tight-lipped in regards to salary transparency, this text analysis project may be useful for job seekers who want to know more about compensation before taking any steps toward subjecting themselves to a cold, algorithmic hiring process.

Research Question: Using keywords and skills found in the job description, can we accurately predict the salary of the job?

To this end, I will first use sentiment analysis to see what words in the job description may contribute to a "positive" or "negative" job description, and see if those labels themselves are correlated with the job salary. Next, I will move on to using Machine Learning methods to answer my formal research question; a Ridge Regression will be implemented to see which words have the largest magnitude coefficients on our salary regression. Next, a neural

network will be implemented to capture the complex relationships between many input features (tfidfs) and predicted salary. My main evaluation metric for this regression task will be Mean Squared Error, of which Ridge will minimize.

2 Data Sources and Descriptions

I use APIFY, a cloud platform for web scraping, to scrape recent jobs data from the employment website Indeed.com. My dataset originally included 700 jobs (white collar, blue collar, service jobs, and skilled trades) with locations in New York; however, in my exploratory analysis I discovered that it had also scraped some Spanish-language jobs. After filtering out jobs not in the English language and additional data preprocessing to filter out potential duplicate posts, my final set consisted of 639 unique jobs posted on Indeed in 2024 ready for our text analysis.

The main input features I use are: 'company', 'position', and 'description'. The main target variable is 'yearlysalary', which is a manually created column based on the 'salary' column that could take the form of hourly salaries, daily salaries, and yearly salaries. Descriptive statistics- in this case, only applicable for our quantitative target variable hourly salary- is shown below:

Table 1: Summary Statistics

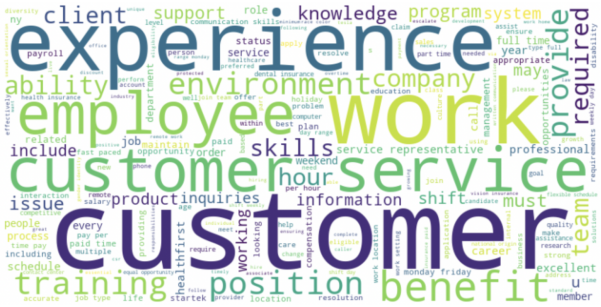
	yearly_salary
Count	632
Mean	69343.35
Stdv	45391.71
Min	325.00
Median	57510.00
Max	440000

Note: Descriptive statistics for yearly_salary.

In addition to these numerical descriptive statistics above, the text data in the job descriptions were also explored in the exploratory data analysis step. Word clouds are shown below, in Figures 1 and 2.

We can see that certain jobs require certain skills, and as such words more common to that job type appear more often. For example, "service" and "customer" appear more often in client-facing roles, and "data" and "team" appear more in analyst roles. Across the board, "experience" appears in all of the wordclouds, meaning experience is desired no matter the role.

Figure 1: Wordclouds by Job Type



Note: Top words for client-facing job descriptions. Top words for analyst job descriptions

Figure 2: Wordcloud for Overall Job Descriptions

Note: Top words overall, for all positions.

3 Method

The main methods I use for my analyses are: Sentiment Analysis, Ridge Regression, and Neural Networks.

I first use pre-trained sentiment analysis models to see the sentiments of different job descriptions. Because jobs advertisements on Indeed are exactly that- advertisements to encourage job seekers to apply- I expected mostly positive results, and was not surprised by the results: 624 positives, 13 negatives, and 2 neutral. The main reason I chose to implement sentiment analysis was not merely to confirm these suspicions, but to see if certain words are predictive of lower salaries. The pretrained model indeed gives a significant finding: words associated with service jobs ("dishwashing", "ironing") are predictive of lower salaries. See Figure 3 for details.

Significantly, these results were found after implementing a logit-lasso classification to predict job description sentiments, which has close to a 0.99 accuracy score due to the class imbalance. The central finding of this classification problem is not the logit-lasso itself, but the words that have the highest magnitude coefficients from that logit-lasso. Here, the finding detailed above and in Figure 3 are found, and certain words are found to be predictive of a lower sentiment, and often lower associated salaries. These results are preliminary results as of yet, so I include them here in the methods section.

Figure 3: Top 5 Negative Words

	word	coef	abs coef
8148	osi	-30.963059	30.963059
11411	toiletry	-23.447377	23.447377
3861	dishwashing	-21.384278	21.384278
6903	linens	-19.405892	19.405892
6435	ironing	-7.661715	7.661715

Note: Highest magnitude coefficients are all negative.

.

For my main regression task of predicting salaries from job descriptions, I am using a Ridge Regression of the form defined in equation (1). In this example,

$$(\text{predicted salary})_i = \beta_0 + \beta_1(\text{TFIDF vector})_i + \varepsilon_i + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{1}$$

$(predictedsalary)_i$ is our target variable of interest, the yearly predicted salary for job description i. X_i is our main independent variable, the vector of coefficients (weights) corresponding to the TF-IDF vector constructed for job description i. The TF-IDF vector itself consists of measurements evaluating how relevant a job descriptor word is to that document/job description i, which is itself in the larger corpus. ε_i is the error term for each job description regression, and $\lambda \sum_{j=1}^{p} \beta_j^2$ represents the regularization term added to prevent overfitting.

The *conditions under which* this analysis is valid rests on common assumptions of linearity, as ridge regressions are regularized forms of linear regressions. In the current example, we would require some form of conditional independence assumption to hold. We can express this as

$$E[\varepsilon_i | X_i] = 0 \tag{A1}$$

This means that once we know the values of our independent variable (the TF-IDF vector constructed for job description i), the expected value of the error term should be 0. That is, the error term is not systematically related to the values of the TF-IDF vector; there are no systematic biases left in the residuals between the obeserved and predicted salaries. Given that each document of job descriptions seems to be independent of each other and our large enough sample size, we will proceed with our Ridge regression.

β_1 is our main metric of interest- a vector of coefficients corresponding to the TF-IDF vector. Each element within β_1 corresponds to the contribution of a particular word in the TF-IDF towards the predicted yearly salary. These estimates are presented in Section 4.

Finally, I use a neural network architecture to predict yearly salaries from job descriptions. Neural networks complement my regression task well, as they may be more effective

at capturing the complex relationship between the input features of the TF-IDF vector and yearly-salary. Although currently I am in the early stage of the neural network, the architecture consists of multiple layers of neurons interconnected by weighted edges. I employ a feedforward neural architecture with three layers: an input layer, a hidden layer, and an output layer of interest.

The input layer is fed the constructed TF-IDF vector for job description i. The hidden layer is composed of 64 neurons with a nonlinear ReLU activation function, defined as: f(x)=max(0,x). This layer is the primary processing unit where the network learns the complex patterns between the TF-IDF and the yearly salary. The output layer consists of one neuron representing the predicted yearly salary that is outputted, which is necessarily activated by a linear activation function for our regression task.

All of this information is encapsulated in the equation:

$$(\text{predicted salary})_i = \text{ReLU}(\beta_0 + \beta_1(\text{TFIDF vector})_i) + \varepsilon_i$$

where the ReLU is applied element-wise to the linear combination of the input features and their corresponding weights.

Our metric of interest is again β_1, and is interpreted in much the same way as our ridge regression: the impact of each word in the TF-IDF vector on the predicted salary. A one unit change in X represents a higher relevance of that word in the job description, and an increase in this is associated with a β_1 increase/decrease (based on if β_1 is positive/negative) in the associated salary. Job descriptions containing more relevant terms or words are more likely to be associated with higher predicted salaries.

4 Results or Expected Results

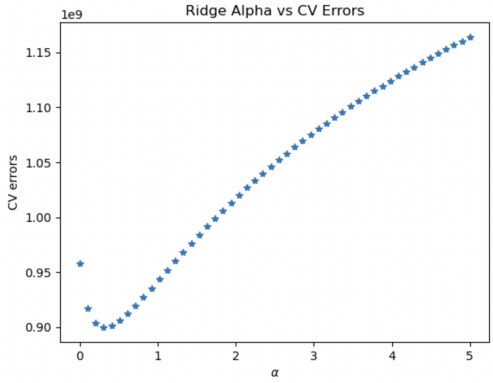
I will expand upon the results found so far for our Ridge Regression and Neural Network here.

First, after constructing TF-IDF vectors for each job description, I trained a Ridge Regression on an evaluation metric of Mean Squared Error, which we seek to minimize. Looping over the alpha parameter found an optimal alpha of around 0.3, and this graph is shown below in Figure 4.

Figure 6 specified the highest magnitude positive and negative words. Positive words are associated with higher salaries according to our ridge specification; of note is the pattern of tech and finance-related words, such as "ai", "risk", and "business". This suggests that most positions that contain higher salaries in our dataset are likely to be finance, business, or technology-related.

By contrast, the negative words are much more service-related. For example, "shift", "hour", "week", and "customer" seem to be related more to service or part-time jobs, with

Figure 4: Ridge Alpha Plot



Note: Optimal Alpha of 0.3.

Figure 5: Ridge Coefficients

6867	management	19025.781729
954	ai	17078.131791
5947	info	15680.692006
1145	anthropic	15285.377074
6527	lead	14754.615668
9498	risk	14516.375279
3556	develop	13526.747383
2697	compensation	13008.824596
2011	business	12824.403413
9314	research	12158.195567

Top magnitude positive words.

	var	var_ridge
9952	shift	-16359.918868
5638	hour	-13543.910505
7367	must	-13073.772226
3248	customer	-12897.677279
5346	guests	-10931.255592
3949	duties	-10929.621474
9302	required	-10511.365903
9876	service	-10131.617852
11816	week	-10062.748386
643	able	-9919.959307

Top magnitude negative words

their emphasis on customer-oriented, scheduled work.

The neural network, with a ReLU activation function, has been trained with 10 epochs and an Adam optimizer. The weights for the neural network for each word is outputted below in Figure 6. More analysis is needed, and I will consider making a SHAP feature importance chart to see which words hold the most weight to our yearly-salaries target variable.

5 Conclusion

So far, my Ridge Regression results hold the most power in answering my original research question of predicting salaries from job descriptions. It has been hypertuned with the optimal alpha, and produced concrete results in identifying the highest magnitude positive and

Figure 6: Weights for Neural Network

.	zoladz	zone	zones	zoo	zoology	zoom	zpi	zr	zulawski
.	0.005924	-0.015482	0.010595	-0.014703	-0.008793	-0.016011	0.012822	0.008451	-0.009268
.	0.056588	0.081490	0.080958	0.052740	0.072716	0.104175	0.038457	0.049132	0.059690
.	0.039971	0.077908	0.116833	0.049912	0.084635	0.107394	0.043960	0.065577	0.054786
.	0.033501	0.038067	0.065612	0.047877	0.031938	0.071969	0.041096	0.054183	0.031445
.	0.073783	0.067889	0.117569	0.029527	0.078869	0.106414	0.052329	0.042125	0.073879

Note: A snippet of the weights determined for each word.

negative results in determing salaries. For next steps, I will finish-up a similar in-depth analysis for my neural network to determine if it concludes different words. The limitation of my research rests on the fact that my dataset is data focused on jobs in New York, and a similar project may not be generalizable to the rest of the nation. Beyond this class, I may consider addressing this limitation by reproducing the project on data outside the field of New York, and out into all corners of America.

References

Abril, Danielle (2023) "Your résumé isn't the only thing popular job sites evaluate," *Washington Post*.

Weed, Julie (2021) "Résumé-Writing Tips to Help You Get Past the A.I. Gatekeepers," *New York Times*.