Predicting Salary Based on Job Descriptions

Anonymous

1 Introduction

Predicting the salary of a job based on its description is a useful task for job seekers and employers who want to know the market value of a job. However, this task is challenging due to the unstructured and noisy nature of text data and the scarcity and imbalance of labelled data. Therefore, it is important to explore how different machine learning techniques that require different levels of supervision can leverage the labelled and unlabelled portions of data sets to predict the salary of a job. It is also important to experiment with different feature representations to compare the performance and quality of the models. In this report, we address the following research question: How do different machine learning paradigms, such as supervised, unsupervised, and semi-supervised learning, perform on salary prediction? Furthermore, can we improve the performance by combining labelled and unlabelled data in a hybrid approach? We also briefly investigate how different feature representations (TF-IDF and sentence embedding) affect the performance and quality of the models.

2 Literature Review

In this section, we review some related literature on the topic of salary prediction using machine learning techniques. We first introduce the data set that we use for this project, then we discuss some previous studies that have applied different machine learning paradigms to the task of salary prediction.

2.1 Dataset

We use a data set of 13,902 job descriptions, of which 8,000 are labelled with the salary bin (a categorical label indicating the salary band) and the mean salary (a numerical value indicating the expected salary). The remaining 5,902 descriptions are unlabelled. The data set also contains a development set of 1,738 labelled job descriptions and a test set of 1,737 unlabelled job descriptions. The data set is derived from the resource published by Bhola et al. (2020), who proposed a language model based extreme multi-label classification framework for retrieving skills from job descriptions. The dataset is collected from a Singaporean government website, consisting of over 20,000 richly structured job posts.

2.2 Previous Studies

Machine learning involves different techniques for learning from data and making predictions. Generally, Machine Learning can be grouped into three main groups: supervised, unsupervised, and semi-supervised. Supervised learning uses labelled data to predict an output, while unsupervised learning discovers hidden patterns in unlabelled data. Semi-supervised learning combines labelled and unlabelled data to improve performance, especially when labelled data is scarce.

Several studies have applied different machine learning paradigms to the task of salary prediction using various data sources and features. Huang (2023) used machine learning to predict salary from 15 features of the 1994 census database. He compared random forest, decision tree, and logistic regression and found that random forest was the most accurate. He also used Pearson correlation to study the feature-salary relationship.

Duarte and Burton (2023) compared semi-supervised text classification methods based on how they use labelled and unlabelled data. They classified the methods into four types (self-training, co-training, graph-based methods, generative models) and tested them on benchmark datasets.

3 Methods

In this project, we explored three different approaches which contributed to achieving the goal of creating a model which classifies salary ranges based on job descriptions with good accuracy. We employed supervised, unsupervised, and semi-supervised techniques and compared their

performances using different features and models.

Before anything, we first analyzed whether the labelled training data and validation data were balanced or not. It was apparent there was only a slight imbalance in the data labels, which could be countered by giving weights to classes when training the data.

For the supervised approach, two models, Neural Networks and Random Forest, were considered. This is because they can capture complex relationships and high-dimensional data in different ways: Neural Networks can learn hierarchical representations and extract relevant features, while Random Forest can handle missing data and noisy features and is robust to overfitting. Both models were trained on the provided TF-IDF vectors and sentence embeddings, and the hyperparameters were tuned by using a grid search which evaluated the accuracy on the validation set. In the neural network models, the batch size, number of hidden layers, and layer size were tuned, while the number of epochs was set to 30 with early stopping and a patience of 5, meaning that training would stop if the validation loss did not improve for 5 continuous epochs (the weights would also be reset to that of the epoch in which since improvement halted). For the random Forest model, the tuned hyperparameters were the number of decision trees, the maximum tree depth, and the minimum number of samples required to split an internal node. To measure the quality of a split, Gini impurity was used over entropy, since both lead to similar decision trees and comparable predictive performance. However, Gini impurity is computationally more efficient since it avoids the logarithmic calculation required by entropy:

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Unsupervised approaches do not directly solve classification problems but instead reveal underlying patterns or structures in the data, such as identifying groups of job descriptions with similar characteristics. Initially, the labelled portion of the training data was visualized by reducing the data to a 2D space using PCA and estimating the density of each salary bin in the 2D embedding space using gaussian kernel density estimation (KDE). After which, this data was clustered using K-Means with K=10, and the quality of the clustering algorithm was evaluated using the Rand-index metric, which provides an objective measure of performance, especially since ground truth labels are not used in the clustering process. Additionally, clustering using DBSCAN was attempted, but the high density and close proximity of data points resulted in insignificant clusters.

For the semi-supervised section of the project, we considered self-training. In self-training, a model is trained on the labelled portion of the dataset and then used to make predictions on the unlabelled portion of the dataset. In each iteration, elements of the unlabelled portion which have been predicted with high confidence are added to the labelled dataset, and the model is retrained on the expanded labelled dataset. This process is repeated until the model's performance stops improving or reaches a predefined threshold. case in real-world datasets. The self-training model used sentence embeddings of job descriptions and was trained on neural network and random forest models. The use of sentence embeddings was based on promising results from the supervised learning section (discussed in detail in the next section). Different confidence thresholds were tested, and the best-performing model found in supervised section was used as the baseline model.

4 Results

4.1 Preprocessing

Upon visualizing the class distribution of the labelled section of the training set and validation set, it became evident that there existed a minor imbalance (Figure 1).

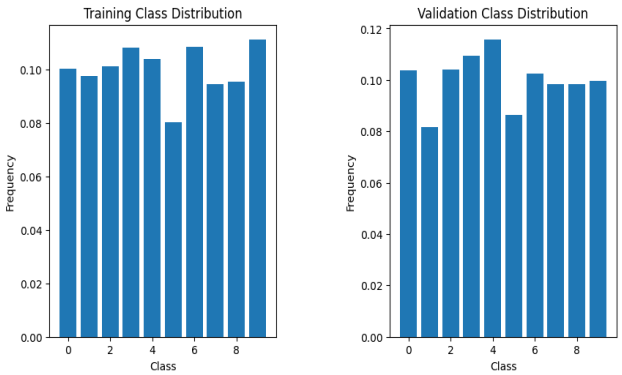

Figure 1- Salary bin class distribution

This led us to create class weights for the training data which would be used during model training. We note that we should not consider the validation set when creating and using these weights, since we want to account for possible data imbalance in the test set.

4.2 Supervised Learning

In this project, various machine learning models were trained using different embeddings. Neural Networks and Random Forest models were the primary models used, and the hyperparameters for each model was optimized using grid search.

4.2.1 Neural Networks

We compared the performance of the neural networks which gave the best results (lowest validation loss) for each of the different embeddings. We look at how the validation loss changed across epochs for each of the models, and where early stopping occurs (figure 2).

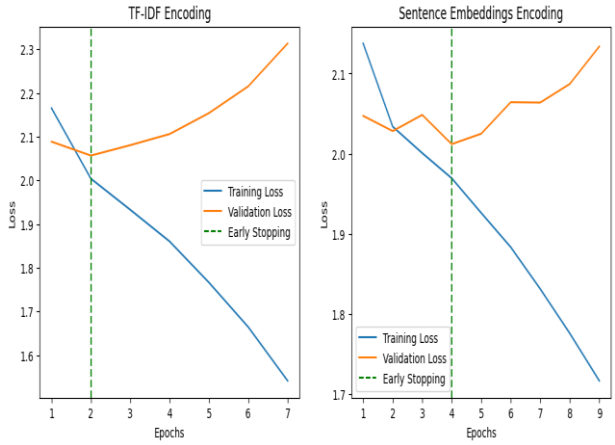

Figure 2- Loss/Epochs for NN models

and the validation performance of these models was as follows:

Model Encoding	Validation Loss	Validation Accuracy
TF-IDF	2.056	0.2360
Embeddings	2.011	0.2412

Table 1- performance of NN models

We also have the classification reports:

TFIDF

	precision	recall	f1-score	support
0	0.44	0.60	0.51	180
1	0.22	0.23	0.22	142
2	0.23	0.11	0.15	181
3	0.19	0.08	0.11	190
4	0.17	0.49	0.25	201
5	0.09	0.04	0.06	150
6	0.00	0.00	0.00	178
7	0.22	0.06	0.10	171
8	0.19	0.18	0.19	171
9	0.29	0.51	0.37	173
accuracy			0.24	1737
macro avg	0.20	0.23	0.20	1737
weighted avg	0.20	0.24	0.20	1737

Figure 3- TF-IDF NN report

EMBEDDINGS

	precision	recall	f1-score	support
0	0.42	0.63	0.51	180
1	0.17	0.22	0.19	142
2	0.18	0.15	0.16	181
3	0.19	0.18	0.18	190
4	0.17	0.15	0.16	201
5	0.13	0.14	0.13	150
6	0.18	0.19	0.18	178
7	0.23	0.15	0.18	171
8	0.26	0.04	0.07	171
9	0.33	0.55	0.42	173
accuracy			0.24	1737
macro avg	0.23	0.24	0.22	1737
weighted avg	0.23	0.24	0.22	1737

Figure 4- Embeddings NN report

Even though both classes have low accuracy, the embeddings model seems to have slightly better precision and recall for classes compared to the TF-IDF model, which suggests it is the better model.

4.2.2 Random Forest

The classification reports of the RF models with the best results are as follows:

TFIDF

	precision	recall	f1-score	support
0	0.37	0.67	0.48	180
1	0.20	0.18	0.19	142
2	0.20	0.16	0.18	181
3	0.25	0.21	0.23	190
4	0.23	0.14	0.18	201
5	0.16	0.05	0.08	150
6	0.19	0.19	0.19	178
7	0.24	0.20	0.22	171
8	0.32	0.19	0.24	171
9	0.27	0.60	0.37	173
accuracy			0.26	1737
macro avg	0.24	0.26	0.23	1737
weighted avg	0.24	0.26	0.24	1737

Figure 5- TF-IDF RF report

EMBEDDINGS

	precision	recall	f1-score	support
0	0.40	0.64	0.49	180
1	0.23	0.27	0.25	142
2	0.17	0.15	0.16	181
3	0.20	0.16	0.18	190
4	0.20	0.12	0.15	201
5	0.24	0.12	0.16	150
6	0.23	0.23	0.23	178
7	0.23	0.19	0.21	171
8	0.30	0.23	0.26	171
9	0.27	0.51	0.36	173
accuracy			0.26	1737
macro avg	0.25	0.26	0.24	1737
weighted avg	0.25	0.26	0.24	1737

Figure 6- embeddings RF report

The results show that both models have slightly higher accuracy and f1-scores for most classes than the neural net models, which means they are slightly better at predicting salary ranges. This may be due to its ensemble technique that combines decision trees, which reduces expected variance error. Ensembling a Neural Network to the same degree would be more time-consuming and resource intensive.

4.3 Unsupervised Learning

In this section we only use the sentence embeddings to gather information regarding the data. Before the learning, we visualize the labelled data in a 2d plot by lowering the data dimensions using PCA, while showing the density centers for each of the salary binds with KDE:

Figure 7- Ground truth with density centres

Interesting information can be gathered from this. For example, it can be seen that salaries in the first two bins have similar words, while the last salary bin is further away from all others.

Now, by performing K-Means with K=10, we have the following:

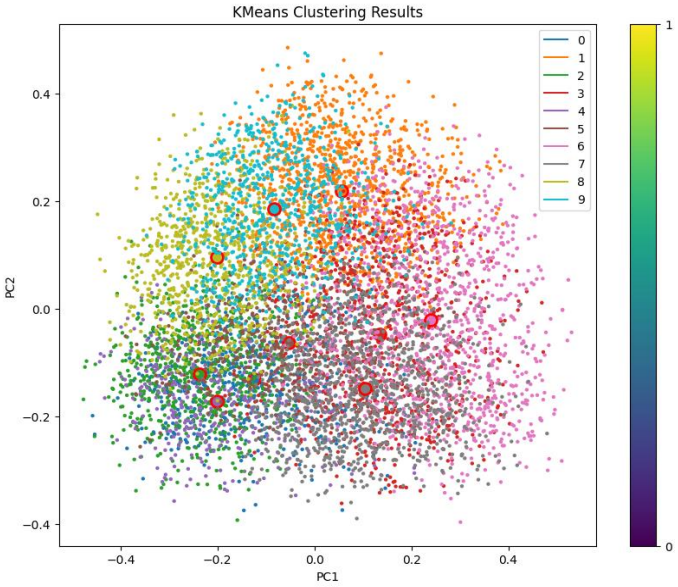

Figure 8- K-Means with centroids shown

If the cluster centroids were closely aligned to the density centers, it would suggest that the embeddings are capturing the information related to the salary bins. However, that doesn't seem like the case. By analyzing the Rand-Index between the ground truth labels and the clusters, we get the value of 0.0322 indicating a weak agreement between the two.

4.4 Semi-Supervised Learning

Self-training models with different confidence thresholds were trained. The models add labels to the unlabelled portion based on this confidence. The configuration used was based on the best model and hyperparameters found in the supervised section (Random Forest with sentence embeddings).

Confidence Threshold	#Added labels	#Iterations	Validation Accuracy
0.4	322	10	0.2498
0.5	36	6	0.2440
0.6	15	3	0.2573
0.7	0	0	0.2521

Table 2- Self-Training models on Random Forest

The model with the confidence threshold of 0.6 has the highest accuracy, and its classification

report is as follows:

	precision	recall	f1-score	support
0	0.39	0.63	0.48	180
1	0.20	0.25	0.22	142
2	0.18	0.15	0.16	181
3	0.22	0.17	0.19	190
4	0.21	0.15	0.18	201
5	0.22	0.12	0.16	150
6	0.25	0.24	0.25	178
7	0.22	0.18	0.20	171
8	0.26	0.16	0.20	171
9	0.26	0.51	0.35	173
accuracy			0.26	1737
macro avg	0.24	0.26	0.24	1737
weighted avg	0.24	0.26	0.24	1737

Figure 8- threshold=0.6 classification report

The semi-supervised approach yielded similar results to the supervised approach, but with slightly lower F1-Scores, indicating slightly worse performance.

5 Discussion

After examining supervised, unsupervised, and semi-supervised models, we can draw some insights about our data and the problem.

In the supervised section it's reasonable to conclude that TF-IDF is a less meaningful representation because it created sparse arrays without being able to capture semantic information from the sentences (since it only counts word frequencies based on the top 500 words, with most of them being 0's). However, using sentence embeddings only produced slightly better accuracy and F1 score, which may indicate a weak relationship between job description data and salary. If this is the case, the performance of the models may not improve significantly, regardless of the representation used. Analyzing the ground truth salary plot was performed using PCA on the sentence embeddings, with this analysis revealing that there were no discernible patterns, besides trivial ones, in the data, further indicating a weak relationship between job description data and salary. Furthermore, the clustering of the data points using the rand-index was found to be very close to zero, suggesting that the data points do not have any significant clustering structure.

The semi-supervised approach based on self-training did not achieve satisfactory results, since low confidence thresholds were required to label the unlabelled data. This approach may have resulted in adding noise to the labelled data, which could negatively impact the performance of the model. Given all the previous observations, it can be inferred that the self-training approach was not able to label data with higher confidence thresholds due to the lack of structure in the data.

6 Conclusion

The task of predicting salary based on job descriptions proved to be a formidable challenge. Although we utilized a training set of 13,902 job descriptions, of which 8,000 were labelled with salary levels, the data did not capture factors that may be an important influence on salary, such as years of experience required or job location.

Our analysis of supervised models revealed that changing the representation of the job description only led to small performance improvements. Additionally, the unsupervised approach showed lack of patterns between job descriptions and salary, Indicating weak relationship between the two. The lack of structure in the data prevented the semi-supervised approach from labeling the unlabelled data with high confidence, and reducing the confidence threshold would introduce noise to the data.

Overall, our findings suggest that job descriptions may not be the sole factor in determining salary, and alternative features or data sources may need to be considered to improve the performance of the models. While our study provides insights into the limitations of various approaches, further research could explore other factors that may influence salary and develop more effective models for predicting salary.

References

Bhola, A., Halder, K., Prasad, A., & Kan, M.-Y. (2020). Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 5832-5842). Barcelona, Spain (Online): International Committee on

Computational Linguistics.

Huang, Zelin. (2023). Salary Prediction with Analyzing Affected Elements by using Pearson Correlation. BCP Business & Management. 44. 786-792. 10.54691/bcpbm.v44i.4955.

Duarte, J. M., & Berton, L. (2023). A review of semi-supervised learning for text classification. Artificial intelligence review, 1–69. Advance online publication. https://doi.org/10.1007/s10462-023-10393-8