Machine Learning For Assessing Point Factor Analysis Performance

Patrice West
School of Data Science and Society
University of North Carolina - Chapel Hill
Chapel Hill, NC 27599
pmwest@unc.edu

Abstract

This research seeks to apply machine learning techniques, specifically FastText, to compare job descriptions and assess the performance of the measurement tool, using data from the State of Virginia Department of Human Resources (DHRM) compensation and classification website. By utilizing the FastText algorithm, this study aims to improve the accuracy, efficiency, and scalability of job classification systems, particularly in public sector organizations. This approach aligns with the ongoing trend of integrating advanced machine learning techniques into HR functions to streamline processes and provide data-driven insights for better decision-making.

1 Introduction

Compensation management is a key component of human resources (HR) management. Fair, consistent, and transparent pay practices impact morale, performance, and organizational success. To establish an equitable pay structure, jobs must be classified correctly to ensure that comparable jobs across work units are similarly compensated. Job classification is both an art and a science. The aspects of this work that are quantitative, such as statistical analysis and computations are straight forward. While there is no subjectivity in performing the computation of a mean, deciding which values to include in the calculation can be. There is lots of qualitative work involved in compensation, and this work is subjective. While measures are generally taken to reduce subjectivity (which often introduces bias), machine learning techniques may be able to pinpoint imbalances in classification structures.

One of two methods are generally employed when classifying jobs, point factor analysis and factor comparison. This research will focus on the point factor analysis method. This method mimics many machine learning techniques that apply weights to factors based on importance. For each factor, like job complexity, education and experience, scope and impact, working conditions, and level of supervision received, there are levels. These levels correspond to a point range, and the factors themselves are weighted. The total points generally correspond to a pay grade on the pay scale. Assessing job factors and determining levels is subject to the interpretation of the evaluator, and this is where inconsistencies can arise.

The primary motivation for this project is to explore how machine learning, particularly FastText, can be applied to categorize and classify job roles within a compensation plan, focusing on the alignment between job descriptions and point factors. FastText, known for its speed and ability to process text data efficiently and within context, is ideal for processing and classifying large volumes of textual job descriptions. The goal is to develop a tool that can assist HR professionals in classifying jobs based on a standard point factor system, thereby reducing the potential for inconsistencies or human error.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

2 Related Work

Job classification can be time-consuming and prone to inconsistencies. Recent studies have sought to automate this process using machine learning techniques, such as naïve Bayes, decision trees, and support vector machines. A study by Nasser and Alzaanin (2020) compared various machine learning methods to compare outcomes. Using a dataset of real and phony job postings, text classification was performed using multinomial naïve Bayes, support vector machine, decision tree, k-nearest neighbors and random forest models. Preprocessing the data included converting the text into vectors using term frequency-inverse document frequency (TF-IDF). In a similar way to the point factor analysis method, TF-IDF applies weights to terms based on importance. Using accuracy, precision, recall , and F1-scores as model performance metrics, they found random forests to be the best at predicting the appropriate class.

In another study, Das and Diaz-Garcia (2001) created a Visual Basic program to identify the most relevant classification factors and appropriate levels. Statistical methods were used to analyze the classification model to pinpoint problematic factors that improperly influenced the points assignment. This resulted in the removal of overlapping, correlated factors, and adjusted levels for proper distribution across different job classes. The final tool improved accuracy and consistency and had future applications in assessing and redesigning classification models.

A more recent study from 2019 used random forest and extra tree models to classify job titles into occupational groups. Ikudo et. al (2019), obtained employee data from public universities and combined it with Census data. Job titles have many variations. By grouping titles under occupational families, necessary skills can be more easily identified. Other machine learning methods were tested, but random forests had the highest predictive accuracy. The researchers concluded that the tool could be useful in aiding with manual classification of jobs, but that it was not ready to be launched a fully automated process.

While these studies highlight the potential of using machine learning to improve job classification, none have specifically applied word embeddings to measure the relatedness of jobs. This research aims to assess parity between jobs in similar pay grades to confirm that jobs are properly classified.

3 Methods

The methodology for this project involves several key steps: data collection, preprocessing, model development, and evaluation. The original design plan was to use FastText, a text classification algorithm developed by Facebook's AI Research (FAIR) team, to convert the text in job descriptions and factors into vectors. In job descriptions, words carry weight. FastText considers word forms and decomposes words into smaller sub words (ngrams).This additional layer in the conversion of words into vectors allows the model to generalize well and compare new words that may not have been present in the training data, as it simply looks for the components of the word and assigns value based on reconstruction. The overarching factors for determining job classification within the State of Virginia Department of Human Resources Management (DHRM) system are complexity, accountability, and results. The original plan was to use these features, along with role descriptions, and convert them into sentence vectors. Complexity was the only feature used due to time constraints. Tensorflow Text Vectorization was used to convert the complexity feature into vectors, due to compatibility issues with FastText. Despite consulting literature and FastText and genism libraries, errors arose at various stages of the preprocessing and deployment phases, across multiple IDEs and with the corpus converted into various formats. In the end, a neural network and k-neighbors classifier were used to evaluate similarity between job complexity and pay grade. The hypothesis was that roles with like levels of complexity would belong to the same pay grade, being as though the resulting vectors should be relatively close to each other.

3.1 Data collection

The dataset used in this study consisted of job descriptions obtained from the DHRM salary and job structure site. The web scraping package BeautifulSoup was used to gather hyperlinks for each job class, with the intention of using Excel Power Query to extract and organize each job title. This method was abandoned, and the data was manually collected and saved in csv format. The corpus contained a total of 3,301 unique words. The longest complexity description was 247 words and

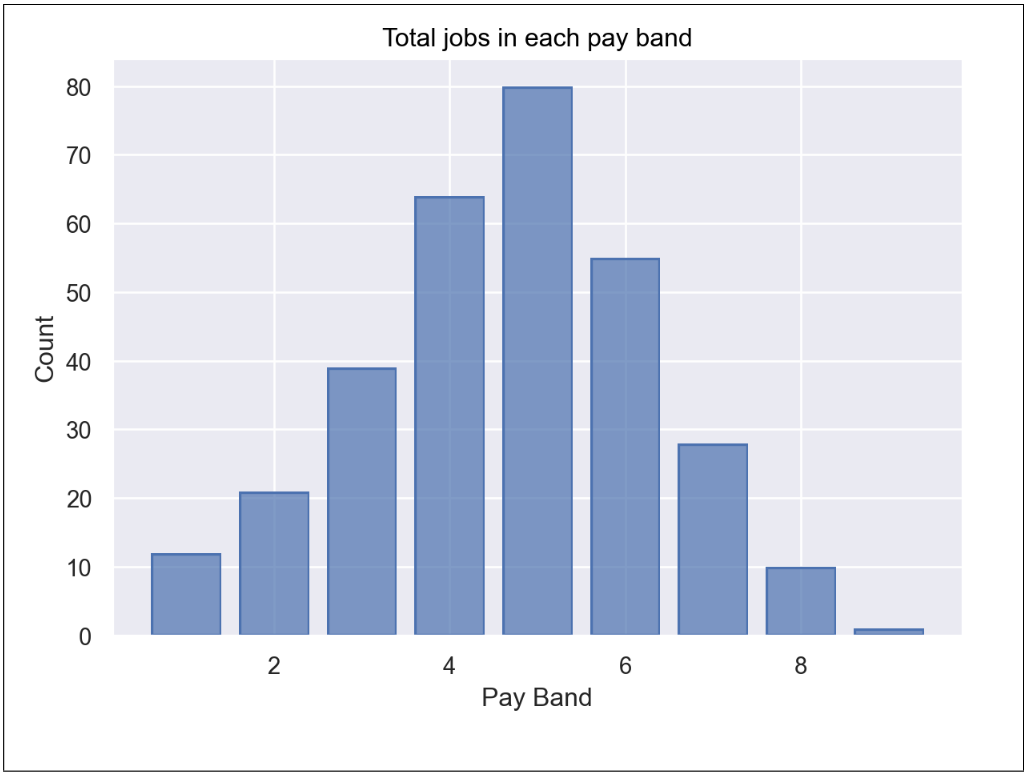

Figure 1: Class distribution.

the shortest was 8. After stop words were removed, the word knowledge appeared at the highest frequency (535) and applies was second (452).

3.1.1 Data preprocessing

Upon initial import, the dataset was ordered by job category. To ensure training and testing datasets contained random samples of jobs, the rows of the data frame were shuffled using the pandas data frame sample method. As seen in Figure 1, the class frequency follows a normal distribution. This was concerning as class imbalance can affect model performance. A modified dictionary of English stop words was created to remove words of little value and retain words contained in the original vocabulary that may hold weight in a job description. The dataset was split into training and testing variables, and the feature sets were converted to tensors. The cross-entropy metric used within the neural network expects labels to be integers; therefore, the label values were reduced by one to avoid errors caused by a mismatch in output layer dimensions. The final preprocessing step was performed using Tensorflow Text Vectorization. The default arguments of the preprocessing layer removes punctuation and converts all characters to lowercase. Words are split using white space and encoded to utf-8. The number of max tokens parameter was set to 3,250, to account for unique words in the corpus.

3.1.2 Model development

The neural network model consisted of five layers. The embedding layer creates a dense vector which is a unique representation of each word. The layer takes in vectors of integers with values no higher than the size of the vocabulary. The output_dim argument (length of the embedding vector for each word) was set to 32. (Initially, this parameter was set to 12, allowing for unique representations of up to 4,096 words, but subsequent runs decreased performance). The mask_zero argument was set to true to prevent sending sparse vectors to the gated recurrent unit (GRU) layer. The GRU layer optimizes performance on large datasets, and preserves important values while the full sequence is processed in a manner similar to a recurrent neural network. GRUs are good at detecting patterns. The output layer was activated using the softmax function, producing a probability distribution of

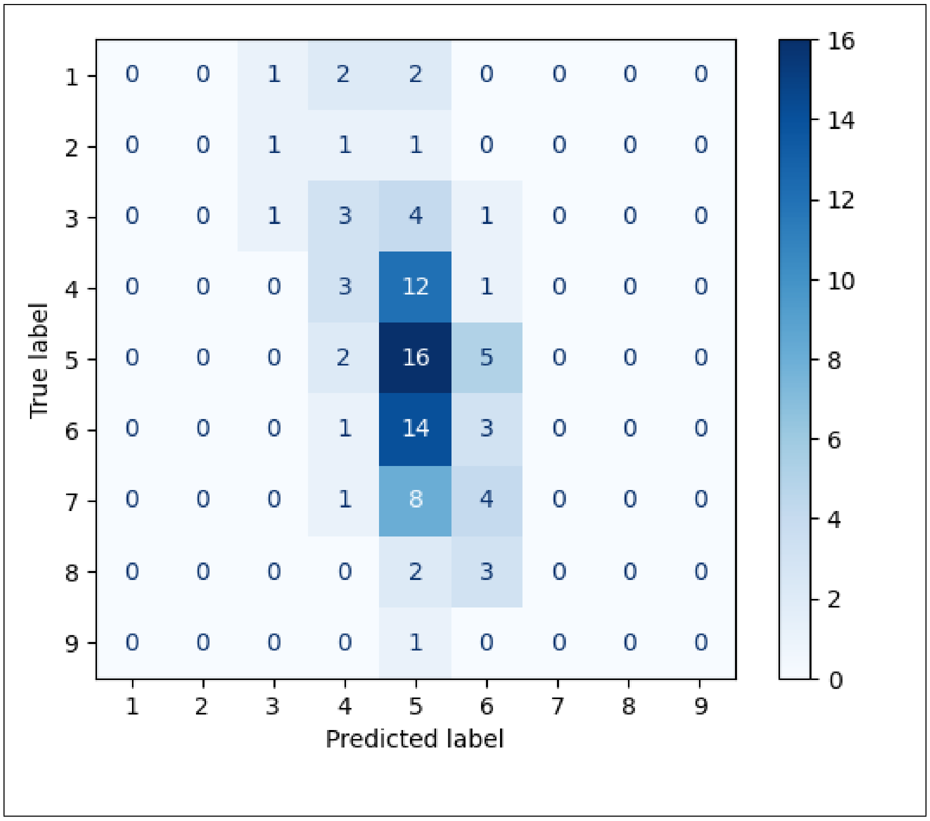

Figure 2: Confusion matrix for k-neighbors k=5.

class predictions for each instance. Since the classes are integers and not sparse encodings, sparse categorical metrics were used to evaluate accuracy and loss during model compilation. The adam optimizer, based on stochastic gradient descent, is used to minimize loss, and accuracy and F1 score are used as additional performance evaluation metrics.

The k-Neighbors model received text vectorized using the Text Vectorization function with output set to term frequency inverse document frequency (TFIDF). To address the imbalance in paygrade representation, hyperparameters for weights was set to distance to give higher weights to closest neighbors.

3.1.3 Model development

4 Results

The model did not perform well. In the text classification study performed by Nasser and Alzaanin (2020), KNN recall and precision rates were 98.2 and 99.5 respectively. With the exception of stemming, similar preprocessing methods were used. The first significant difference between the dataset used for this study is the multilabel class. The dataset used in the aforementioned study was evaluated using binary classification methods. Their study also suffered from imbalanced data, but with a different effect. The Kaggle dataset used contained 18,000 job descriptions, with only 800 instances being labeled as fake. The prediction success rate may simply be attributed to the high probability of predicting the correct label. For this paper, the imbalance led to a low accuracy, F1 score and high loss rates. The best accuracy attained on a model fit was 39%. Loss was low, but the F1 score was low as well. The model does not generalize well, and like the class distribution, predictions follow the same shape with pay grade 5 at the center and pay grades 4 and 6 being the only other classes predicted.

As seen in Table 1, the K-neighbors classifier had an even worse performance with k=5 and no additional hyperparameters selected. Tuning a model using cosine similarity and k=39, a 25%

Table 1: Classification Report

Class	Precision	Recall	F1 Score	Support
1	0.00	0.00	0.00	5
2	0.04	0.00	0.00	3
3	0.00	0.00	0.00	9
4	0.17	0.44	0.25	16
5	0.00	0.00	0.00	23
6	0.00	0.00	0.00	18
7	0.00	0.00	0.00	13
8	0.00	0.00	0.00	5
9	0.00	0.00	0.00	1
Accuracy			0.10	93
Macro avg	0.02	0.12	0.04	93
Weighted avg	0.03	0.10	0.04	0.04

accuracy rate was obtained. It is clear that the class imbalance in the data was too much for the model to overcome.

5 Conclusion

There is much to be explored in constructing a model that can be used to classify jobs. The first phase will be to successfully install FastText and GloVe to evaluate machine learning models using different text vectorization techniques. Curating data to increase the size of the vocabulary and balance the dataset should also greatly improve model performance. The state of Virginia only had 310 jobs, but three additional features were not included in the dataset; results, accountability, and job descriptions. The addition of these features may influence model performance despite the class imbalance.

Another area of exploration includes enhancing the dataset with jobs from other jurisdictions with comparable features, and scaling the target variable according to salary ranges. Since pay grades are a proxy for salary ranges, equivalencies can be made between organizations using metrics such as the midpoint for each range or other statistical measures. Developing an efficient means of gathering this data from the web will improve efficiency as manually constructing the dataset was time consuming.

Text classification results in high dimensionality. Exploring other machine learning algorithms is another potential performance enhancer. Principal component analysis (PCA) and support vector machines (SVM) may better identify differences between classes.

Tuning hyperparameters is another technique that needs to be applied to the current models to ensure optimization. This can be accomplished using cross-validation techniques. Evaluating the effectiveness of various feature scaling methods on vectorized text is also a next step in the model improvement process.

Evaluating the competitiveness of a salary structure involves surveying an adopted market and collecting data on positions. It has been demonstrated that variations in job titles and descriptions complicate the process of establishing job classes and determining parity between roles. Market surveys are based on the premise that the jobs selected for comparison are indeed comparable, usually by a predetermined metric e.g., seventy-five percent (75%). If a successful model can be created, this methodology could be extended to more easily identify job matches between organizations.

References

Ao, Z., Horvath, G., Sheng, C., Song, Y., Sun, Y.(2022, November 28). Skill requirements in job advertisements: A comparison of skill-categorization methods based on wage regressions.

Das, B., Garcia-Diaz, A. (2001, July). Factor selection guidelines for job evaluation: A computerized statistical procedure. *Computers & Industrial Engineering, 40*(3), 259 – 272. https://doi.org/10.1016/S0360-8352(01)00028-6

De Mauro, A., Greco, M., Grimaldi, M., Ritala, P., (2018). Human resources for Big Data professions: A systematic classification of job roles and required skill sets. *Information Processing & Management, 54*(5), 807-817. https://doi.org/10.1016/j.ipm.2017.05.004

Ikudo, A., Lane, J. I., Staudt, J., Weinberg, B. A. (2019, May 1). Occupational classification: A machine learning approach. *Journal of Economic and Social Measurement, 44*(2-3), 57-87. https://doi.org/10.3233/JEM-190

Nasser, I., Alzaanin, A. (2020, September). Machine learning and job posting classification: A comparative study. *International Journal of Engineering and Information Systems (IJEAIS), 4*(9), 6 – 14. http://ijeais.org/wp-content/uploads/2020/9/IJEAIS200903.pdf

Shet, S. V., Poddar, T., Samuel, F. W., Dwivedi, Y. (2021). Examining the determinants of successful adoption of data analytics in human resource management – A framework for implications. *Journal of Business Research, 131*, 311-326. https://doi.org/10.1016/j.jbusres.2021.03.054