Using Machine Learning to Classify Assessment Center Text Data

Sarah Haidar, Georgi Yankov, Sukesh Kumar, and Jimmy Zheng

Development Dimensions International

INTRODUCTION

Assessment centers (AC) are very labor intensive and can cost over $5,000 per participant (Thornton, Murphy, Everest, & Hoffman, 2000). Moreover, assessors might be highly cognitively loaded when evaluating AC participants on multiple dimensions, and this might affect the quality of their ratings (Gaugler & Thornton, 1989). Thus, if some part of assessors' job can be automated, the cost of AC might drop, and its usage may increase.

Some assessment center excercises are scored on open-ended responses from candidates. In psychology, when it comes to analyzing the content of open-ended text, the convention is to rely on human coding (Iliev, Dehghani, & Sagi, 2014). However, the process could become tedious and prone to human error. Instead, machine learning (ML) applied to assessment centers can be used for training assessors or optimizing scoring. ML can provide a second data point to each assessor's exercise rating, and, thus, improve it. Our study aims to demonstrate the use of ML for classifying performance (e.g., needs development, proficient, strength) on text data gathered in AC.

Therefore, our main research question was whether ML text classification methods achieve high accuracy in predicting decision-making?

METHOD

- We used an archival dataset containing evaluations of 7,523 US-based candidates, applying for executive leadership positions. This data was collected via AC exercises which contained high-fidelity situations that were designed to assess various leadership competencies, including decision-making. These exercises took the format of email exchanges, to which the candidates had to respond to via writing. After the completion of these day-long exercises, trained assessors reviewed candidates' responses and provided an overall decision-making rating for each candidate, using the following 3-point scale: 1 (behavior not present), 2 (good performance), and 3 (strong behavior performance). The score frequencies were the following: 827 1's, 4976 2's, and 1720 3's.
- We used the Python programming language and leveraged its NLP packages NLTK (Loper & Bird, 2004) and Scikit-learn (Pedregosa et al., 2011). To create our bag of words, we cleaned the data from punctuation and lemmatized the words (e.g., "worked" became "work"). We started with 59123 unique words and limited them to 3000 parsimonious words for modelling.

RESULTS

For classification modeling, apart from logistic regression (LR), we also used two state-of-the-art classifiers: gradient boosted trees (GBT) and Naïve Bayes (NB). To evaluate the performance of our models, we split the data between Training (90%) and Testing (10%). For evaluation metrics we used confusion matrices (i.e., rows representing instances of the predicted classes and columns representing the instances in the actual class) and three indices these matrices produce: precision (true positive divided by true positives and false negatives), recall (true positives rate), and F1 score.

Our classification accuracy and Type II errors for the D's (i.e., 1's) and S's (i.e., 3's) were quite high initially. Then we removed the P's (i.e., 2's), which left us with 2547 observations. This increased classification accuracy dramatically to 85%, and the GBT method performed the best.

After re-running the classifiers with the two-ratings removed, all evaluation metrics were very good. However, GBT delivered the best values, two of its most notable improvements over LR and NB being the precision for the one-ratings and the recall for the three-ratings. Using only the GBT classifier, we did 5-fold cross validation with the following hyperparameters: boosting stages of 500, learning rate of .1, maximum tree depth of 3, minimum observations in each leaf of 1, minimum observation in a node to split of 2, tolerance for early stopping of .0001. The classification accuracy of this final GBT model was 84.9% with confidence interval of 80.3 to 89.2, which means that the model could classify almost nine out of ten people correctly.

However, NB allowed to trace back the most informative words used in its classification modelling, and the top ten words were: "lobbying", "deploy", "focusing", "fixing", "continuing", "factor", "revisit", "access", "understands", "expanding".

Initial Classification Accuracy

Corpus	Method	Rating	Precision	Recall	F1
3000 words	LR	1	0.20	0.16	0.18
		2	0.71	0.75	0.73
		3	0.37	0.33	0.35
	GBT	1	0.41	0.09	0.15
		2	0.72	0.91	0.80
		3	0.51	0.26	0.34
	NB	1	0.22	0.80	0.35
		2	0.75	0.34	0.47
		3	0.39	0.61	0.48

Classification Accuracy after Removing the Middle Ratings

Method	Rating	Precision	Recall	F1
LR	1	0.69	0.72	0.70
	3	0.85	0.83	0.84
GBT	1	0.83	0.72	0.77
	3	0.86	0.92	0.89
NB	1	0.61	0.84	0.71
	3	0.89	0.71	0.79

DISCUSSION

- Our study demonstrated empirically the successful leveraging of ML and natural language processing for classification of AC text data. Specifically, our final tree-based GBT classifier achieved cross-validated accuracy of 85% in classifying our executive leadership candidates into deficient and proficient decision-makers.
- This result is promising for applied practice as decision-making is a foundational leadership competency and assisting assessors in scoring it appropriately can have tangible results in company performance by accurately assessing relevant competencies and enhancing quality of hires.
- Our study has the limitation of predicting only extreme ratings. This was mostly because of our imbalanced data set in favor of good (i.e., the two-ratings) decision-making. Ideally, ML classifiers require bigger data than ours, data than can be up-sampled and down-sampled to achieve equal representation of the predicted outcome's levels.
- During the preprocessing stage we lose data from words with spelling mistakes. Thus, another recommendation for future research is to perform automated spelling correction before assembling the initial corpus. Pressure during the AC sometimes leaves no time for individuals to proof-check their responses.

REFERENCES

Gaugler, B.B. and Thornton, G.C. III. (1989). Number of assessment center dimensions as a determinant of assessor accuracy. Journal of Applied Psychology, 74, 611–618.

Iliev, R., Dehghani, M., & Sagi, E. (2015). Automated text analysis in psychology: Methods, applications, and future developments. Language and cognition, 7(2), 265-290.

Loper, E., & Bird, S., 2002. NLTK, the Natural Language Toolkit. In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia, PA: Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Thornton, G. C., Murphy, K. R., Everest, T. M., & Hoffman, C. C. (2000). Higher cost, lower validity and higher utility: Comparing the utilities of two tests that differ in validity, costs and selectivity. International Journal of Selection and Assessment, 8(2), 61–75.