STEPHEN FOX, PhD

Bellevue, WA 98006 ☐ 646.265.7722 ☐ sfox@alum.mit.edu
www.androidalpha.com ☐ www.linkedin.com/in/stephen-fox ☐ www.github.com/sfox1975

April 7, 2017

Subject: HackerRank Machine Learning Contest: Tagging Raw Job Descriptions

Summary

I enjoyed participating in this contest, even though my total score was not all that impressive. I only discovered the contest a day before it was due to end, so unfortunately I was unable to put in as much effort as I would have liked.

A RandomForest Decision Tree was used to tag the raw job descriptions with any of 0 to 12 tags per job. The algorithm was tuned using gridsearch cross validation. The final model achieved an F1 score of 0.515. This was below the top score on the leader board by 0.235, but I believe by employing some of the ideas discussed below, I could have achieved a better score.

Training Data Set Quality

I did not run any data set quality checks in the interest of time, since I only had a few hours to build a working model. In hindsight, I would have liked to run consistency checks on the dataset, as a minimum. For example, certain tags are mutually exclusive. The same job description cannot ask for 5+ years experience and 1-2 years experience. I would have liked to scan the data for incompatible tags, and then depending on the frequency of issues, discard those data points from the model development process.

Data Preprocessing Steps

Extraneous symbols and stop words were removed from the job description and text was transformed to lower case only. A 'bag-of-words' approach was used for transforming the text documents into a word frequency matrix. A maximum features setting of 5,000 and an ngram_range of (1,4) was used.

Given more time, I would have used more intelligent preprocessing steps. I think much would have been gained by engineering a few additional features. For example, I would have liked to have searched the job descriptions for instances of a number followed by the text 'year' occurring within a certain number of characters, as I think this would be a strong indicator of an experience duration requirement (e.g. "…5 or more years…" would be a good signal for a '5-plus-years-experience-needed' tag).

Model Choice

In the interest of time, I opted for testing a few standard models from the sklearn library. I tested decision tree, KNN and random forest approaches. Given initial performance, I opted to proceed with the random forest and conducted parameter tuning on it.