Domantas Zilenas

Felix Campbell

Hunter Negron

Dustin Van Tate Testa

October 30, 2018

The algorithm we’ve developed to screen candidates is primarily concerned with academic success. First, any candidate who did not receive a 100 in their CS 100 introductory course is removed from the final list of candidates to review. Then, any candidate who does not have a 75 as their overall GPA is also dropped from the list. The algorithm then uses a weighted average to score candidates, with the weights assigned to courses that Moogle would deem important for their developers to succeed in. In this case, Moogle places high importance on data structures and algorithms, since they separate engineers who have good problem solving abilities and those that do not. From here, they are sorted according to the score calculated by the weighted average, and the top 20 candidates are returned.

We leveraged modern JavaScript’s functional programming features such as map to implement our algorithm. At first, we tried implementing it in Dustin’s own scripting language, YodaScript. However, a few issues with the implementation, namely its incompatibility with Microsoft Windows and its unoptimized runtime made it a less attractive option. JavaScript was then the most obvious option, due to its platform independence, and functional programming features. The original JavaScript we wrote would have required installation of the Node.JS runtime because of the use of file I/O, but it was rewritten to work as a web page.

Our algorithm is purely merit based and does not pay attention to soft skills, gender, or other non-quantitative data. Grades aren’t perfect predictors of an employee’s performance in the workplace. An interview where a few descriptors are applied to the interviewee would give the algorithm additional data to work with when selecting candidates. We also do not consider an upwards trend in grades, and immediately disqualify those who don’t do perfectly in their introductory course. By excluding those people, the algorithm disqualifies people who have not studied computer science before entering university, and the sort of person who has previous experience likely has had more resources than someone who is learning programming for the first time. So, the overall ethical concern of our algorithm is its lack of concern with qualitative features and that having this sort of data available to the algorithm could yield candidates that are more interesting and may have different viewpoints.