1 Background

Jonathan: Thanks for your help! We are an independent management consultancy lead by two retired executives from large regional companies. We focus on HR, workforce and personnel issues in our work.

One of our client companies is interested in bringing analytics into hiring. The client currently employs human screeners to read through job applications and make hiring decisions for their workforce.

To improve selection, our client is considering using machine learning to predict each applicant's job performance using data from the candidate's application. We're interested to help them. We have some experience in this area, but do not have a full-time IT staff.

Below is some information about the setting and our task, which contains some information and specifications passed along from the client. The jobs in question require numeracy skills and competency with math. Performance on these job tasks is relatively quantifiable.

The client has lots of data about worker characteristics (including skills, previous employment and demographic characteristics such as age, location or gender) that can be used as predictor variables. We'd like to hire you to implement a proof-of-concept algorithm to showcase the potential of this approach. Although it is a proof-of-concept, we want to perform well and have some performance benchmarks mentioned in this document.

2 Your Task

We'd like to build an algorithm that uses the training data to make predictions about 20,000 new (unlabeled) candidates and output these candidates into a CSV file. You may use the programming environment of your choice (R, Python, Java, whatever, including open-source libraries) and whatever statistical or machine learning variables you choose. You are welcome to utilize a pre-existing machine learning package.

To assist you, our client has provided training data from the **current and historical workforce at the company. They have hired these workers based on their recruiters' beliefs such candidates would succeed. Of course, some did not perform as well as expected**. The training data includes job performance outcomes, as well as a large group of other variables you could use to predict performance.

For people in the training data, our client (and we) are confident their performance outcomes in the training dataset have been labeled accurately. Job performance is based mostly on performance on math tasks that can be objectively measured.

To evaluate the predictions, the client will have a look at the actual job performance for the 20,000 test candidates. The 20,000 "test" candidates will come from a representative sample of the candidates our client could potentially hire.

Please note: This includes both candidates like the client's historical workers, and as well as new types of candidates they could potentially hire. The test set might contain candidates unlike those traditionally hired (i.e., unlike the ones in your training data). The client has "ground truth" performance data about the test candidates so that they can measure performance of our predictions.

Build your model to maximize accuracy on the test set. Our client has set out exactly how we'll be evaluated below. But the bottom line is: We want you to be as accurate in your prediction as possible. **Note**: Insofar as we have any other hiring goals – for example, location preferences – we will incorporate these through a separate process. We are asking you only to build an accurate predictive model of job performance.

3 Deliverables

When you're finished, upload your program and predictions via Upwork. To submit your predictions, create a file called `deliverable.zip`. This should include two main files.

- The first should be called `predictions.csv`. This file should be equal to your test set given to you (`testset.csv`), with the `job_performance` column populated with your predictions (rather than empty, as it will be when you're given it).
- The second file should be your main program file, `prediction.<extension>`.

In addition, include all helper files necessary to run your main program in your `.zip` file.

Finally, we also ask for some descriptions of your approach. To simplify, we will provide a survey to complete about how you went about modeling.

Note: As a reminder, **the recruiters' expectations about who to hire may have been systematically wrong.** Performance data is available only for workers like the ones hired in the past at this company. This may reflect recruiters' stereotypes or biases about who is good at math.

As you write your algorithm, please be mindful that your training dataset may originate in a *biased social system*. Adjusting our algorithm to account for discrimination in hiring, self-sorting of applicants, or other sources of such bias could improve our accuracy on the test set. We will be evaluated only on the accuracy of your predictions on the test set.

Here is a white paper recommended to us by some computer science professors working on this topic. The white paper contains an overview, some citations and instructions about this kind of adjustment. Simply request access to check out the white paper. If the white paper doesn't help, feel free to ignore it.

Please note: For the workers hired in the past, we are confident their performance outcomes in your training dataset have been labeled accurately. Performance is based on math and there are objective measures of quality. However, we only have this data about the candidates that were previously hired.

3.1 Performance Benchmarks and Bonus Payments

Our client has given us guidance about how well our algorithm needs to perform. We want to give you some additional incentives to beat these.

A **MSE of 172,400** is an easily achievable baseline. We cannot guarantee a strong review, recommendation or future work without reaching this baseline.

Our **target MSE is 52,500**. We are very eager to beat this benchmark and would love if you can help achieve this target on the test set. We're willing to pay you a bonus equal to an extra hour of work ($28) if you reach this benchmark (no actual additional work required, just extra free money).

In addition, for every **550 point reduction in MSE below your benchmark (52,500)**, we'll give you an extra $14 (half your hourly rate). Beating this target in any form would be a great benefit. If we beat it, we'd like to say thanks however we can.

Keep in mind, we're evaluated on the test set. So be careful about whether you overfit to your training data.

4 Appendix

The remainder of this document includes:

4.1: Information about the performance variable the client wants to predict.

4.2: Some details about the candidate characteristics data.

4.3: The exact formula for how our predictions will be evaluated.

4.1 Outcome Variable

The performance variable in the training data is labeled `job_performance`. You should try to use the other variables to predict `job_performance` for candidates in the test set. `job_performance` spans from approximately zero to 5000, with higher numbers meaning better performance.

4.2 Input/Predictive Variables

Our client has several hundred feature variables that could be used to forecast each job performance. **Feature selection is a part of the challenge**. You are welcome to use whichever variables you wish, and select them using whatever approach you wish. We will be evaluated only on accuracy on the test set.

The feature variables come from a variety of sources in the job application process. Some variables were directly collected through a survey. Some variables have been scraped and standardized from resume. In addition, our client has taken some variables from resumes and merged them with outside databases that link job titles, companies, etc. to skills, industries, etc.

The training data will include a codebook containing descriptions of each variable. The variables include:

Variable Name	**Description**
`candidate_gender`	Candidate's Gender
`national_origin`	Candidate's Country of National Origin
`age`	Candidate's Age in Years
`language`	Candidate's Native Language
`years_education`	Years of Education
`employed`	Is Candidate Currently Employed?
`prev_occupation`	Candidate's Previous Occupation
...	... etc.

Many variables are categorical. For your convenience, all categorical variables have already been standardized. In some cases, you may prefer to convert these categorical variables into a series of multiple binary variables and/or interact with other variables. We have not already done this for you, but the data is relatively clean and standardized. Note that some observations are missing a few features. You are welcome to deal with this however you wish.

4.3 Evaluation

As described above, we need to make make predictions about the `job_performance` variable for roughly 20,000 new candidates from a test sample. These "test" candidates are from a representative sample of the candidates we could potentially hire.

We have collected performance data about these candidates in order to evaluate predictions. For each of the 20,000 candidates, our client will calculate the squared difference between your predicted score and the actual score ("squared error loss"). They will then sum these scores across all candidates and average across them to find the average error. We will be assessed based on the average squared error. Please note: We may not be able to share the the test data but will be able to know our MSE.

Candidate	**Predicted** `job_performance`	**Actual** `job_performance`	**Candidate Score** **=(actual-predicted)^2**
1	3	6	$(6\text{-}3)^2$=**9**
2	7.2	5.3	$(5.3\text{-}7.2)^2$=**3.61**
3	9.1	8.8	$(8.8\text{-}9.1)^2$=**0.09**
4	4.3	6.6	$(6.6\text{-}4.3)^2$=5.29
...	...	...	... etc
...	...	...	... etc
Total Error			=9+3.61+0.09+5.29=**17.99**
Average Error			=17.99/4=**4.49**

Note: This is a toy example using some toy numbers to demonstrate the algebra. The mean squared error for this problem will probably be in the thousands.