Instructions for Candidate

Background:

We've collected data of Just Eat users with an old-fashioned pen-and-paper survey, and the data was subsequently digitised using a very old OCR software that our boss found on some old five-inch floppy disks. In addition to the survey data, we know how much money these users have spent with us during the last two years. During the last year, we also experimentally tested a new personalised marketing campaign on some users. This was a pretty standard randomised controlled trial, where first a population was defined, which was then randomly split into 'test' and 'control'.

Data:

- The CSV file attached called dataset.csv.
- The first variable is a unique user identifier.
- The variable great_customer_class is the "class" variable of whether the customer reported that he/she had a great user experience or not.
- The variable 'spent_18' is the amount of money that each customer has spent last year.
- The variable 'spent_17' is the amount of money that each customer has spent two years ago.
- The variable 'test' indicates if the user has received our personalized marketing campaign last year.
- The other variables are features extracted about the user.

Instructions

There are two parts: Part A focusses on data science / machine learning and software engineering best practices. Part B focuses on statistics and causal inference. We do not expect candidates to shine in every domain, and for this role, it is fine to put more effort in Part B, but at the same time it is not advisable to neglect Part A completely.

Part A: Programming and Machine Learning

- Use Python, any version >=3.
- There is some existing code (only a couple classes) that we would like you to use. Use the abstract classes provided for your models and data loader.
- Write a unit test for something. We don't really mind what, even if you just write a test for the checking the file exists or loading it, we'll be happy.
- Predict the binary class 'great_customer_class' using two prediction models. Use a random forest, then afterwards a second approach of your choice. This could be something to show your knowledge about problem datasets like support vector machines, or could be something fancy like NN.
- Determine ways to evaluate your prediction model.
- Outline any concerns with the data and how you approached them.
- Explain the dominant features for classifiers if it's possible.

Part B: Statistics and Causal Inference

- Did the personalised marketing campaign change the amount of money spent? Please show us what kind of statistical analysis you would do if asked to evaluate an experiment like this.
- Is there something you would check about how the experiment has been set up? How?
- Can you think of (a) different way(s) to leverage any suitable pre-experiment data for this test? If so, how are the results affected?
- Our personalisation experts suggested there might be subgroups among the customers for which the experiment worked better than for the rest. What kind of statistical analysis / causal modelling would you do to investigate this possibility?
- Did the campaign affect how many users reported a great customer experience?

- You learn that only about a third of all personalised messages were actually sent, but we don't know which users received them. All sent messages were read, though. Does this change what you think about the impact of the personalised messages on spending?

During the panel, we'll be going through this code with you and asking questions about your approach. We're looking for how you might handle this type of problem in a real life scenario, and demonstrate some maturity at software engineering. We will also ask further questions around statistics / causal inference relevant to our setting.

Feel free to write me at andre.richter@justeattakeaway.com with any questions. If something in the test is vague or not well defined, please use your best judgement.