Job Assessment - Machine Learning Engineer

Thank you for engaging with this assessment!

This step is required of all candidates**.** By completing this assessment we hope to introduce you to the work we do and give you a chance to show off your skills. This is a **strengths finding** assessment.

You will be writing a **machine learning framework and giving a presentation**.

Here's how we will evaluate your submission:

1. Did you read the instructions and build a working framework?
2. Did you display a high degree of technical acumen when it comes to
    a. Supporting common patterns
    b. Ensuring good science
    c. Making something easy to use
3. Did you communicate your work well to a technical audience?

We respect your time, **please don't spend more than 5 hours on this assessment**. Explaining what you would do with more time is an expected part of the process and can contribute to demonstrating the various proficiencies listed above. Function stubs, shallow interfaces/protocols, and partial functionality is fine. The important part of this assessment is having a jumping off point for our discussions during the presentation, we already assume you are a good MLE, we want to see how you think about enabling other engineers/data scientists.

Your presentation will be a **60 minute** discussion-style meeting with up to four people, similar to a technical peer review. We expect you to lead the discussion for ~30 minutes, but we may run out of time if we have a lot of questions.

Please submit full working code as part of the submission. You can pick one or multiple mediums – it can be a notebook, slides, a live demo, a repository (preferred), or something else.

Please complete this assessment **within seven days**, and reach out to the recruiter if you have a conflict and/or need more time. Thank you!

Problem statement

One way that we (HubSpot's Data Science and ML team) help HubSpot grow better is by building models for marketing & sales teams to better understand and prioritize outreach to

prospective customers. We rely on efficient scalable frameworks to iterate and deliver on ML models. As the MLE on the team, your goal is to support the data scientists by building a framework to accomplish the following:

1. Gather and transform data
2. Rapidly build and iterate on ML models
3. Host and deploy models
4. Measure model impact

For this assessment we want to see your initial version of this framework.

Make sure to keep in mind how members of your team will use and expand your framework over time.

Your framework should:

1. Enable reproducible science, including a reproducible environment. This could include
    a. Containerization, anaconda environment files, pip requirements files, or poetry lock files, etc.
    b. Enabling experimentation with config files, environment variables, CLI arguments, or composite classes like pipelines, etc.
    c. Saving artifacts such as model weights, metric containers, schemas, tensorboards, etc.
2. Abstract common patterns your teammates will follow (e.g. data loading and artifact creation), while allowing for potential extensions/modifications in the future (e.g. new data sources or sinks)
    a. You may consider using patterns like inheritance, protocol classes, pipelines, etc.
3. Communicate its API well
    a. Consider type hinted function signatures, docstrings, comments, markdown files, etc.
4. Demonstrate good software design. For example this could include:
    a. Creating an installable package, cookiecutter, template repository, or some kind of platform
    b. Enabling automatic linting and formatting of your code
    c. Unit testing
    d. Automatic documentation

Your framework should not do everything listed above. We are not asking you to, for example, containerize *and* support anaconda *and* support poetry. Don't make an installable package *and* a cookiecutter *and* a template repository. Make a good framework, not a poor framework that includes everything.

In your presentation we will ask you about:

1. Trade offs you make, for example
    a. Functionality you chose not to implement yourself and instead install/purchase or
    b. The medium that a user interacts with your framework (e.g. package vs cookiecutter vs template repository vs …)
2. The user experience, and how one might do work within this framework. For example how a user
    a. Adds/removes features or defines new transformations
    b. Implements a custom model
3. How artifacts (e.g. model weights, training metrics, notebooks, etc.) produced by your framework might be used by other systems at the company such as a model serving platform, or a back end team's service, or other data science teams etc.

Your task is not to train a model. We will not ask you about model accuracy, feature importance, train/test splits, etc. (though, if relevant, we may ask how your framework allows a user to do these things).

Good luck!

Data & definitions

The assessment includes data for the sake of giving you some to work against. You will not be assessed on the depth of your knowledge of customer relationship management (CRM) technology, or your ability to do data science. But, to avoid any confusion, here is information about the data within.

Assumptions about the data:

- Users of HubSpot's free tier are "prospects" as they are not yet paying "customers".
- Hubspot defines MRR (see data dictionary below) as our measure of "customer spend."
- A "conversion" means transitioning from a non-paying prospect to a paying customer.
- Hubspot's CRM consists of 3 main objects: Contacts, Companies, and Deals. You can also send emails through the CRM which are captured in a separate Email object.
- Assume that the datasets below come from a snapshot in time, moments after the latest timestamp amongst them.

Attached datasets:

- **customers.csv.** This file includes a sample list of our paying customers.
- **noncustomers.csv.** This file includes a list of companies that are not currently paying customers (i.e. prospects).
- **usage_actions.csv.** This file includes usage data for our users.
    - Note: HubSpot is a freemium-model business, meaning a company can use HubSpot before converting and becoming a paying customer. That is, noncustomers appear in this table. Furthermore, customer usage records can

exist from before they converted.

Columns in these files are defined as:

- **ID**: unique company identifier; you may also treat it as the company's portal id for usage actions listed below
- **CLOSEDATE:** date when they became a customer
- **MRR:** acronym for "monthly recurring revenue", a monthly payment amount customers make
- **ALEXA_RANK:** a score given by Alexa considering many aspects of a business, such as traffic, performance etc. For example, Google has a score of 1, Facebook has a score of 4 etc.
- **EMPLOYEE_RANGE:** min and max number for employee size of the company
- **INDUSTRY:** industry of the company
- **WHEN_TIMESTAMP:** timestamp when usage activity was logged ● **ACTIONS_CRM_CONTACTS:** total number of actions logged on the Contacts object across all users in the portal for the specified timestamp
- **ACTIONS_CRM_COMPANIES:** total number of actions logged on the Companies object across all users in the portal for the specified timestamp
- **ACTIONS_CRM_DEALS:** total number of actions logged on the Deals object across all users in the portal for the specified timestamp
- **ACTIONS_EMAIL:** total number of actions logged on the Email object across all users in the portal for the specified timestamp
- **USERS_CRM_CONTACTS:** total number of users that have logged actions on the Contacts object in the portal for the specified timestamp
- **USERS_CRM_COMPANIES:** total number of users that have logged actions on the Companies object in the portal for the specified timestamp
- **USERS_CRM_DEALS:**total number of users that have logged actions on the Deals object in the portal for the specified timestamp
- **USERS_EMAIL:** total number of users that have logged actions on the Email object in the portal for the specified timestamp