Machine Learning Engineer Assignment v2

This document details the technical take-home assessment for the MLE team at Callsign. The assessment is an opportunity for you to demonstrate your software engineering ability. Other than the requirements below, feel free to be as creative as you like. Please attempt and return your solution within one week of receiving it. If this is not going to be possible, please let the team know as soon as possible.

Exercise

The task for you is to build an inference server for the provided model in the form of a RESTful API. Your server should be wrapped in a Docker image so it can be deployed anywhere. Please write code that you would be comfortable deploying into production. You should include unit tests for your code, as well as full documentation. Assume that predictions will be part of a synchronous journey, so <200ms p99 response time should be targeted. We recommend using either the Falcon or FastAPI frameworks, but put no constraint on this.

As a part of this assignment, you are also required to prepare your project for cloud deployment using Terraform. This involves creating Terraform scripts to define the neccessary infrastructure for hosting your Python server and model on any cloud platform. Your configuration should be adaptable to different cloud environments, showcasing your ability to manage and automate cloud resources effectively. If you are unable to test Terraform scripts on a cloud then please include detailed comments to demonstrates understanding and application principles.

Model Object and Request

1. We have provided a trained model object `trained_model.pkl`. This is a binary file containing a logistic regression model `sklearn.LinearModel.LogisticRegression`. The model is used to predict the probability that a given transaction was performed by a bot rather than a genuine user, with 0 indicating a genuine user and 1 indicating a bot.
2. Also provided is a sample POST request body for the api in `sample-body.json`. This is the schema your API should be designed to accept. Features are in the correct order to be passed into the model.

Requirements

- scikit-learn==1.1.2
- The server should run on port 8887
- The server should at a minimum expose a POST endpoint at …/bot-score
- The response from this endpoint should be a json containing at least the probability that the given transaction was a bot, i.e. { “p_bot”: 0.123 }
- You should save every request and response on any cloud storage for future analytics
- The server should be deployable anywhere

- The server should have an overall latency of <200ms p99
- Your code should be unit tested and follow conventional good practice

Please include a README.md with instructions on running the server and a section on: - how you would deploy your solution - monitor it - other considerations for making the server production-ready

Submission

Please submit your response by email with a .zip file containing your solution, and CC mle-intel@callsign.com. Note: Please do not upload solutions to public repos on Github etc. Be prepared to discuss your solution in the following interview. We estimate that this should take no longer than 4-5 hours, and if you have any questions please don't hesitate to get in touch. Good luck!