ML Eng Challenge

Introduction

One of the most common tasks of an ML Engineer at AAG/Spike is to take predictive models into a production environment. The story goes like this: the code is initially written by a Data Scientist in the exploratory phase of the project. Then, some modifications are added to the code and it starts to grow and grow. Without noticing it, the code that once was beautiful and organized, now is ugly and messy.

Now it comes your part of the job. You have to take the model to production. It has to be deployed as an auto-scalable service, and it has to be documented. The documentation has to be easy to understand by another technical team. It has to be self-explanatory so that you don’t need to schedule endless meetings to explain the client how things work.

Querying the model should be as easy as making an HTTP request and getting back the model’s response. Also, we would like to have logs and be able to monitor the model to check if it is still working as should. Sadly, the only thing we have is a bunch of Jupyter Notebooks written by your team mate. We are under running out of time and you need to transform those notebooks in something that meets our expectations as best as possible. We would love to do everything, but we have to prioritize. If we have to choose, building the API is more urgent than implementing some kind of monitoring. Now, if you can do both things before the deadline, even better.

The challenge

In this challenge you will be working with real code, written by a member of our team (a Spiker, an AAG member) when they where applying for the position of Data Scientist. In that challenge, the main goal was to build a Machine Learning model to predict the price of basic supplies in Chile. The applicants had to deliver a Jupyter Notebook with all the

code to train the model and perform predictions. Now your job is to transform this code into something that can be put into production in any platform (either cloud or on-premise).

More specifically, you have to do the following:

1. Separate the training code from the prediction code.
2. Build a training pipeline. This pipeline must have at least two distinct phases. Por example: pre-processing and training. The final output of this pipeline must be a file with a serialized model.
3. Build an API that uses the serialized trained model and exposes an endpoint to obtain predictions. You must provide everything necessary to containerize this service and execute the container.

Tools

1. Pipeline

You can use any library or framework to make data pipelines or ML pipelines, as long as they are open source and they do not depend on a particular cloud vendor. Although it is not a hard requirement, we privilege tools that can be easily tested locally and do not need big infrastructure (for example, a cluster) to work.

2. API

You have to use a framework that allows you to create an API and take your model to production. The only requirement is that it has to be open source.

You have to include everything that is necessary to build a container and executed, either on a cloud server or locally.

Base files

In this repository, you will find everything you need to do you challenge.

https://github.com/SpikeLab-CL/ml-engineer-challenge

The file `data_scientist_past_challenge/data_scientist_challenge_answers.ipynb` has the answers of an old challenge for a Data Scientist position, submitted by an applicant who passed it and was hired as a Spiker afterwards. The Data Science challenge involved answering questions and visualizing data, besides training a model. This can be useful to you to understand the context, but it is not needed to be included on your pipeline. However, you do have to include any treatment made to the variables.

The folder `data` has all the data associated to the challenge.

Submission

You have to share us a link to a git repository with all your code. This repository must also include a Readme.md file, where you have yo explain how you addressed the problem and how we can execute your code locally (your pipeline and your API).

It is very important that you write a good readme, so that someone that is not familiarized with your code can execute it in their computer without making you any extra questions.

Remember you have 7 days to complete the challenge and you have to submit it to:

To: mariapaz.salvatierra@bain.com

Cc: aline.andrade@bain.com

Subject: “MLE Challenge”