UDACITY

Return to Classroom

Operationalize a Machine Learning Microservice API

REVIEW

CODE REVIEW 4

HISTORY

Requires Changes

1 specification requires changes

All the best for your DevOps Journey!!!

Files Submitted

The submitted repository includes a .circleci folder, a README.md file, a Dockerfile and Makefile, as well as an app.py file, a prediction script, and the necessary scripts to run and upload a microservice on Docker and Kubernetes.
There should also be two output text files: `docker_out.txt` and `kubernetes_out.txt` that include the log output after a prediction is made, given some sample input data.

NOTE: Before submitting a link to your complete, project repository, make sure you have included all required and complete files (including run_kubernetes.sh, run_docker.sh, docker_out.txt, kubernetes_out.txt, and a .circleci build directory).

nice work including all the documents, scripts, and output files 👏👍

A `.circleci` folder is included in the Github repository. The directory holds a `config.yml` that checks the project code for errors. Your project should pass, as indicated by a CircleCI status badge in the repository README.

Your project passed Circle CI tests as indicated by a CircleCI PASSED badge in the repository README.

Code Quality & Enhancement

Add an additional logging statement to `app.py` that prints as “info” the output prediction for some given input data.

Fantastic work on logging the output👏🏼

The README file includes a summary of the project, how to run the Python scripts and web app, and an explanation of the files in the repository.

Although you did an excellent job here, I highly recommend reviewing this free course at Udacity about https://www.udacity.com/course/writing-readmes--ud777

Both the Dockerfile and the python file pass linting using pylint and hadolint. This may involve selectively customizing lint overrides in both tools. The lint should be run for both tools via the command `make lint`. Circleci build server validates step.

Both the Dockerfile and the python file PASSED linting using pylint and hadolint in this project.

Docker Configuration

The Dockerfile should create a working directory, install the necessary dependencies, expose port 80, and specify that `app.py` run at container launch.

It's nicely has been done by creating a /app working directory, then install the necessary dependencies by using requirements.txt file and finally expose port 80, and specify that app.py run at container launch.

The Dockerfile should pass `make lint` without errors. Circleci build server validates step.

The Dockerfile has been PASSED make lint without errors.