Operationalize a Machine Learning Microservice API

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Great Assignment! You really showed up you competence on deploying a ML-Model to docker and kubernetes. Perhaps you also want to try Heroku - a platform for serverless applications: www.heroku.com
Here ist a blog-post describing how to deploy a ML model with flask/starlette on Heroku: https://medium.com/analytics-vidhya/put-deep-learning-image-classifier-in-action-a956c4a9bc58

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).

.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

✓ 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.

Great! Here are some interesting links: https://circleci.com/docs/2.0/hello-world/
What is Continuous Integration and Why is it Important? An Overview from CircleCI https://youtu.be/wliXoh8OL-c
CircleCI twitter https://twitter.com/circleci

Code Quality & Enhancement

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

check. Logs are very helpful for debugging and checking that your code is running, as expected.

✓ 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.

Great!
Here is a Youtube link on wrting READMEs: https://www.youtube.com/watch?v=PC05prd2usY
Or if you feel more comfortable reading:
https://www.makeareadme.com/
https://www.youtube.com/watch?v=2dAK42B7qtw

✓ 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.

Great

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.

Good choice using one of the latest versions of Python 3.x.x. The dockerfile ensures successfully installing all the packages.

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

I can see from the status badge that your project passes all lint tests.

✓ Build, list, and run steps are completed in `run_docker.sh`.

✓ While running the docker container, call the prediction script, `make_predictions.sh`; the log output, which includes the output prediction value, should be included in your submission as a text file, `docker_out.txt`.

The docker_out.txt file includes all the expected log statements for running your app locally. You could even include more log statements to see how the input is processed and scaled before a prediction is made.!

✓ The built docker image is uploaded to your own personal Docker ID, as indicated by a complete `upload_docker.sh`.

Great job!

Kubernetes Configuration

✓ This script runs a docker image with kubernetes, lists the kubernetes pod(s), and forwards the container port to a host, using `kubectl` appropriately.

I can see that you've con gured your containerized application to run in one kubernetes cluster, as can be seen in your kubernetes_out.txt.

✓ While running on kubernetes, call `make_predictions.sh`; the terminal output should be included in your submission as a text file, `kubernetes_out.txt`.

DOWNLOAD PROJECT

RETURN TO PATH