

ML-Model-Flask-Deployment

UseCase:

In this project, I deployed a deep neural network(resnet18) on the cloud. Machine Learning models are powerful tools to make predictions based on available data. To make these models useful, they need to be deployed so that it can be easily accessed through an API.

Technologies used:

- **Pytorch, Pickle** for initializing, loading, and saving the resnet18 model.
- **Flask** for creating access Apis for the model.
- **Pytest**, a testing framework before deployment to the server.
- **Docker** for making the application portable.
- **Github** for version control.
- **Travis-CI** for checking the build and testing the app before deployment.
- **Heroku** cloud platform for deployment and CI/CD.

Project Structure:

This project has six parts:

- 1.) `TL_resnet18.ipynb`: It has code for DNN resnet18 which is used for classification of images.
- 2.) `application.py`: Contains flask APIs which receive an image, classify it through the model and returns the output.
- 3.) `Dockerfile`: Docker config file to create docker image of the app.
- 4.) `test_application.py`: Runs tests for `application.py` which are used as check before deployment.
- 5.) `.travis.yml`: Config file for travisCI, includes dependencies which to be installed before the software can be built and tested.
- 6.) `heroku.yml`: It is a manifest to define the heroku app. It allows us to build docker images on heroku.

Workflow:



Github repo:

- https://github.com/mayank18garg/Classification_ML

Running the API:

- URL: <https://classificationmayank.herokuapp.com/>
- At root, it just prints the text.
- To test the API, run client.sh script on terminal.
 - \$bash client.sh <image-path as an argument>
 - Client.sh file is present in github repo which post image using curl command.
 - Test images are there in data/test/

Future Scope and Improvisations:

- To include functional tests for the resnet18 neural network.
- Input validation on image size to prevent DDOS attack.