### Week 5: Cloud and API deployment

### Data Glacier Virtual Internship

Name: Junfei Liu

Batch Code: LISUM16

Submission Date: Jan 5 2022

Submitted to: Data Glacier

1. Introduction
   1. Data

The dataset I used to train the model is a csv file containing 50 pairs of randomly generated x, y values, which can be found in Week4/dataset/A.

* 1. Model

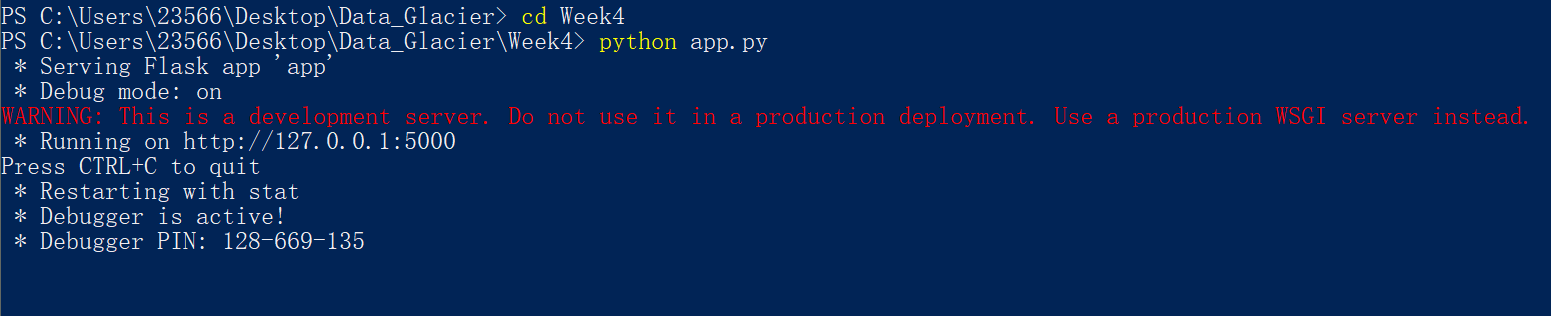
I imported the linear regression model in Scikit-learn and trained the model on the data in csv file introduced above. The model is saved to model.pkl using Pickle. This part is in model.py.

* 1. Deployment

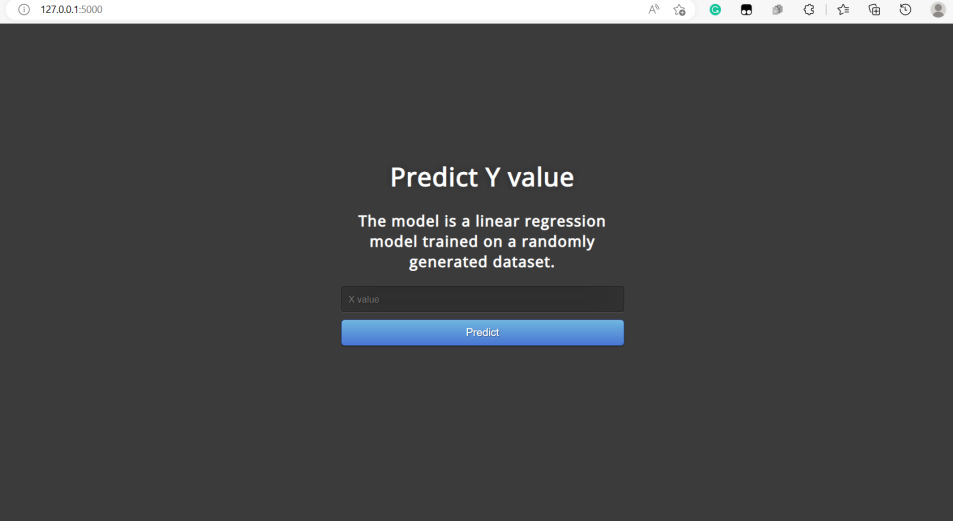
The ML model is deployed through Flask. This part is mainly implemented in app.py.

1. Snapshots

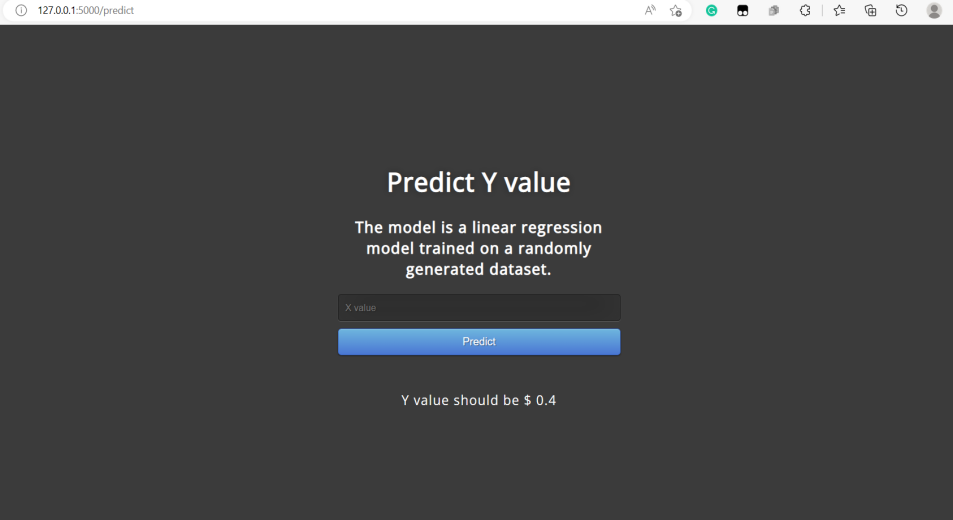
First, start running the app in command line:



Now it is successfully running. The web page can be accessed on local host at <https://127.0.0.1:5000/>. The page is functioning like below:



By entering 1 in the input line, the model will successfully predict the result and present the result to users as below:

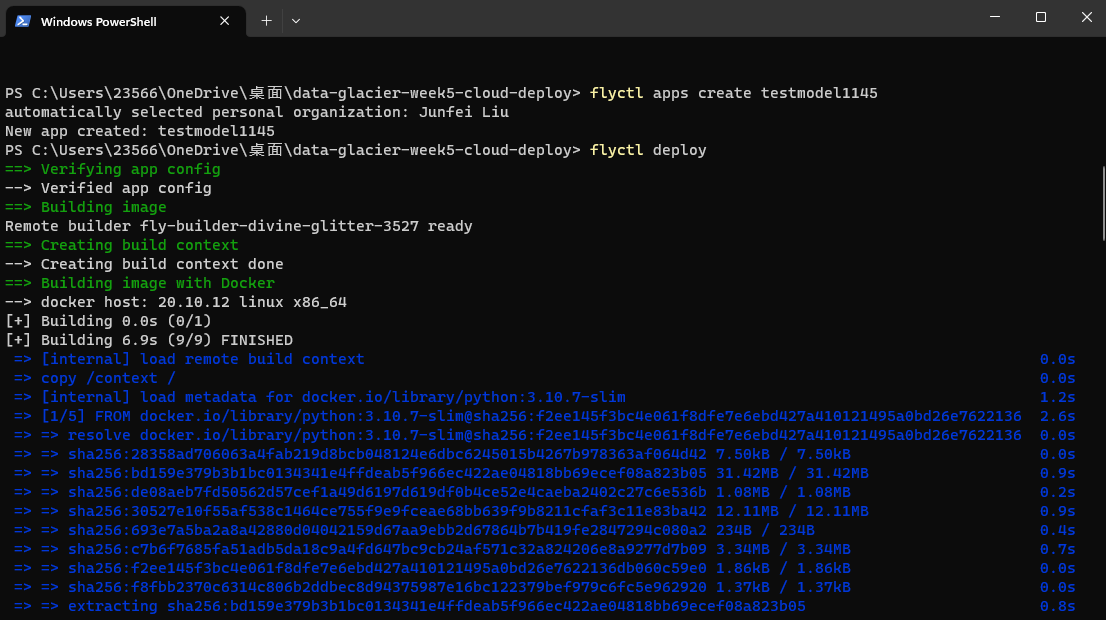


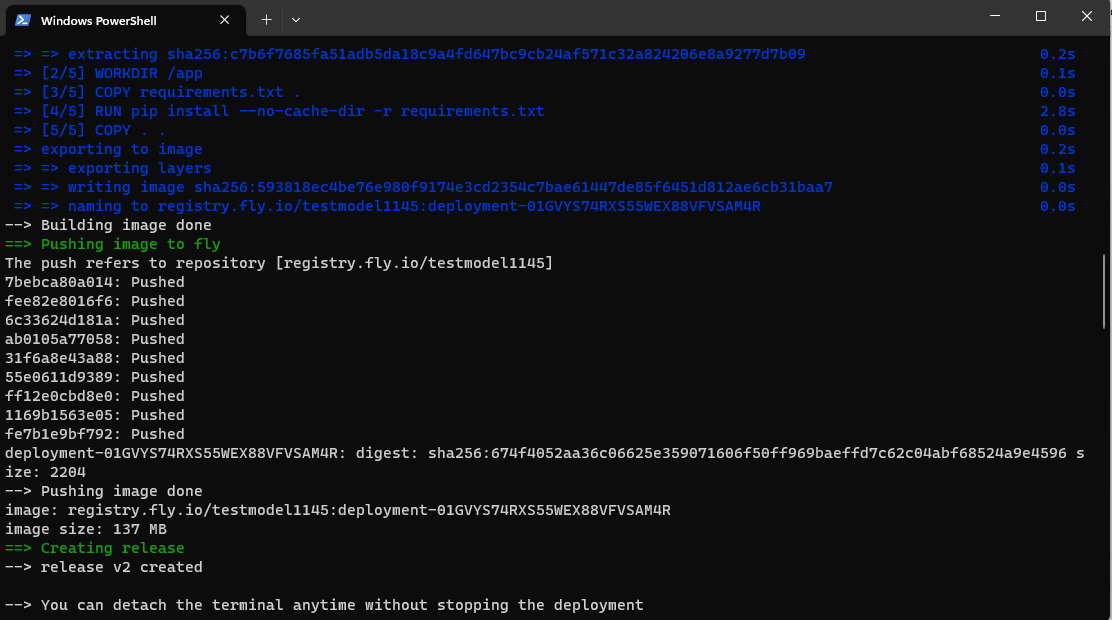
1. Deployment using Fly.io

Because Heroku has switched to no longer providing free plan, I will use Fly.io to deploy the app created in week4.

Fly.io allows to deploy containerized applications using Docker. It is required to have Python, Flask, and Docker installed on the machine to deploy an app using Fly.io. In this task, the app deployed is the linear prediction model created in Week4 using Flask.

‘requirements.txt’, ‘Dockerfile’, ‘.platform.app.yaml’, and ‘fly.toml’ are created as the app deployment configuration with necessary information such as python version, app name, etc. Then, the app can be deployed with ‘flyctl apps create <app-name>’ and ‘flyctl deploy’.





The deployment was successful as the docker image has been created.

Deployment related configuration files are included in git repo.

1. Summary

The deployment of model using Flask is successful in general as the web page can normally function to take inputs and the model can yield predictions.