Cloud and API Deployment

Name: Huu Thien Nguyen

Submission date: 4th June 2022

Submitted to: Data Glacier canvas platform

Internship Batch: LISUM09

**Methods**

The description below contains the detailed approach, including a snapshot of each step of the deployment process.

1. Select data

The Iris data was chosen for this implementation due to this particular dataset's simplicity and low resource cost. The Iris flower data set, often known as Fisher's Iris data set, is a multivariate data set first published in 1936 by British statistician, eugenicist, and biologist Ronald Fisher as an example of linear discriminant analysis in his paper The use of numerous measurements in taxonomic issues. Sepal length, sepal width, petal length, and petal width are the four independent aspects of sizes. The output is the flower's class.



Figure 1: Iris dataset features

1. Save the model

The model was coded using IntelliJ IDE. It read the input file and the Iris dataset, performed the feature engineering with the Standard Scaler library, and then applied the Random Forest model with the split train and test data. After running the python code, a package file was created and ready to be deployed.

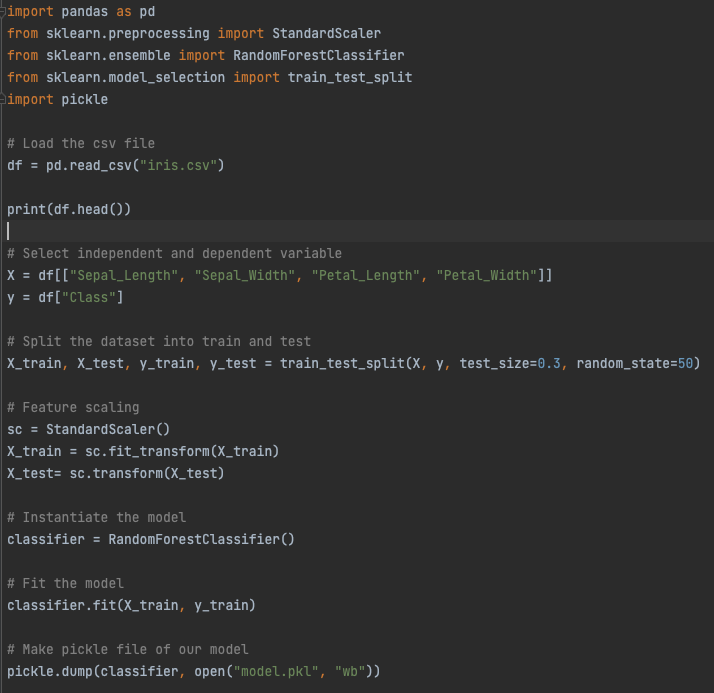


Figure 2: Model creation using IntelliJ IDE

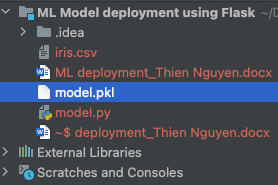


Figure 3: A package file was created

1. Prepare Flask application

The app.py contains the Flask code to build the application. Flask framework and code were included in the flask library. The homepage, route settings, and prediction are specified in each function.

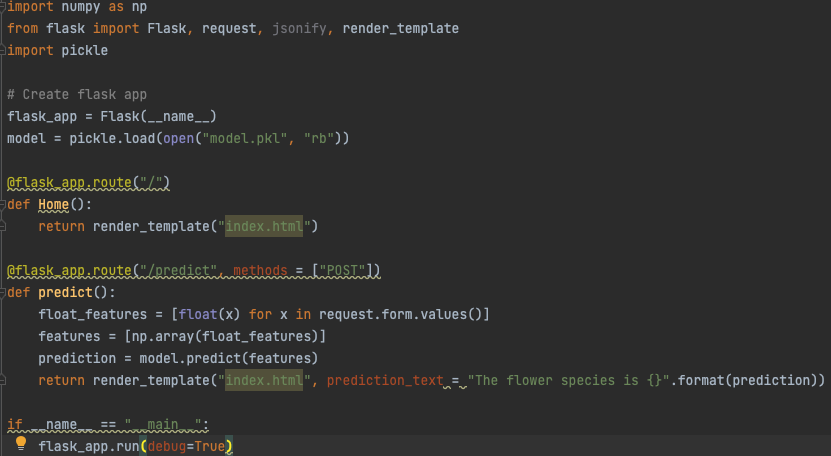


Figure 4: Flask application deployment

1. Webpage setup

The front end of the homepage website was designed in the index.html file.

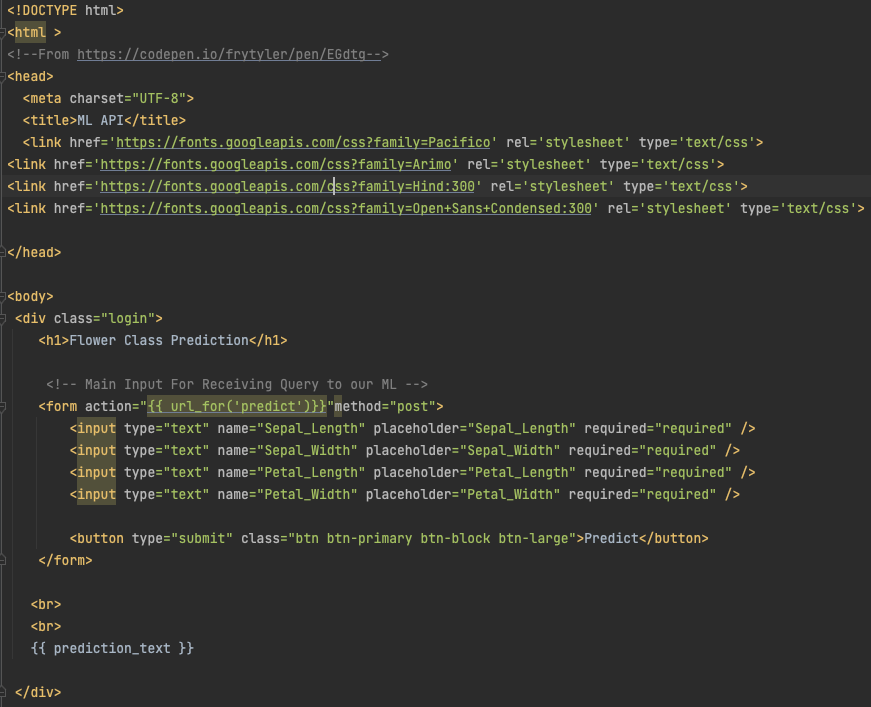


Figure 5: Homepage in HTML code

1. Create an account on Heroku

Create a new account, log in on Heroku, and create a new app.

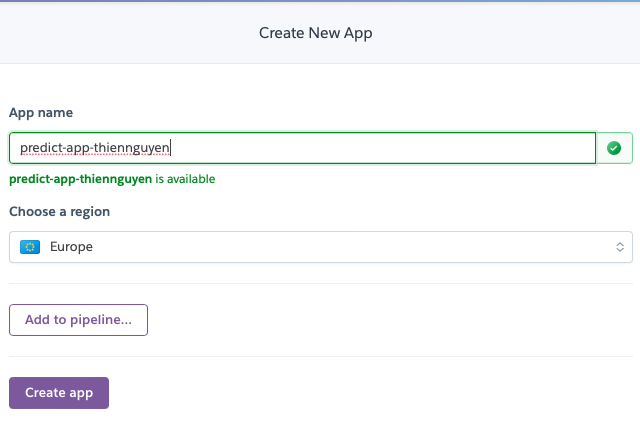
****

Figure 6: Create a new app on Heroku

1. Connect to the GitHub repository

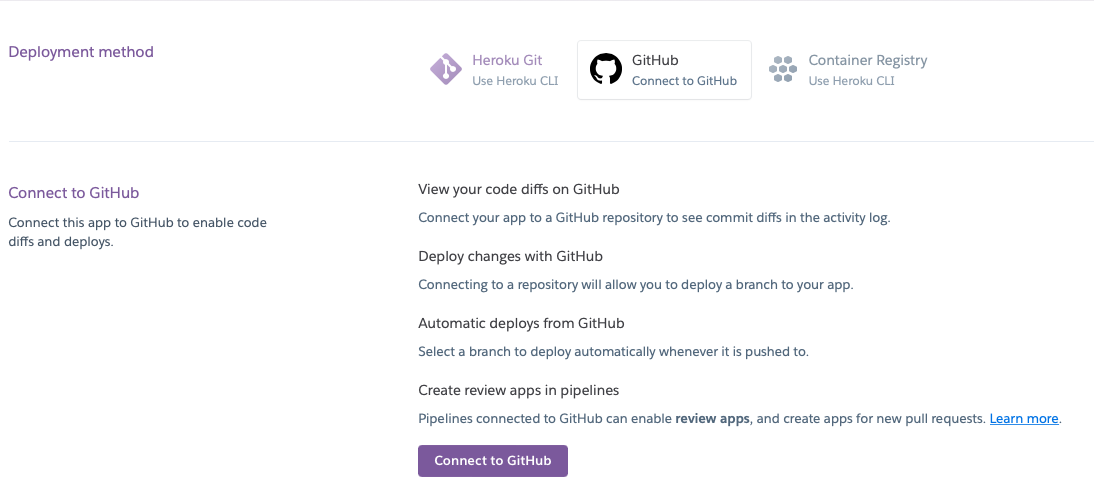


Figure 7: Connect to the GitHub repository

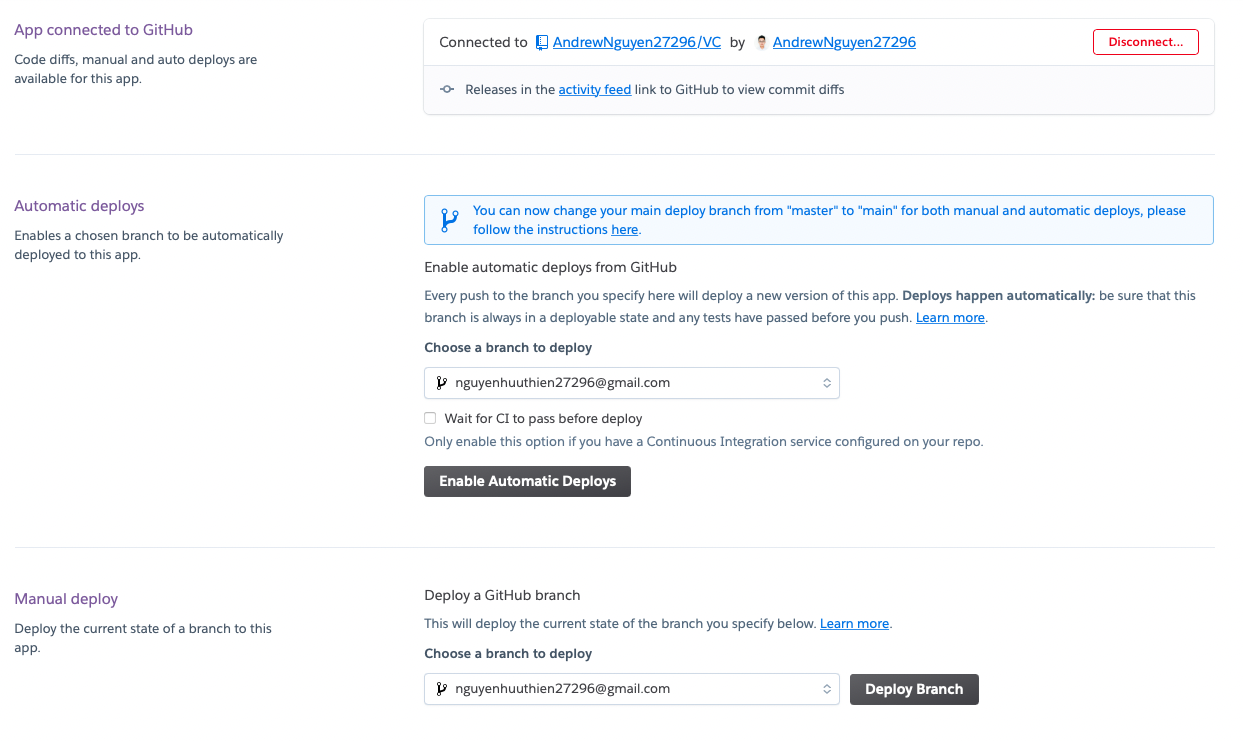


Figure 8: Connect success

1. Deploy project to Heroku

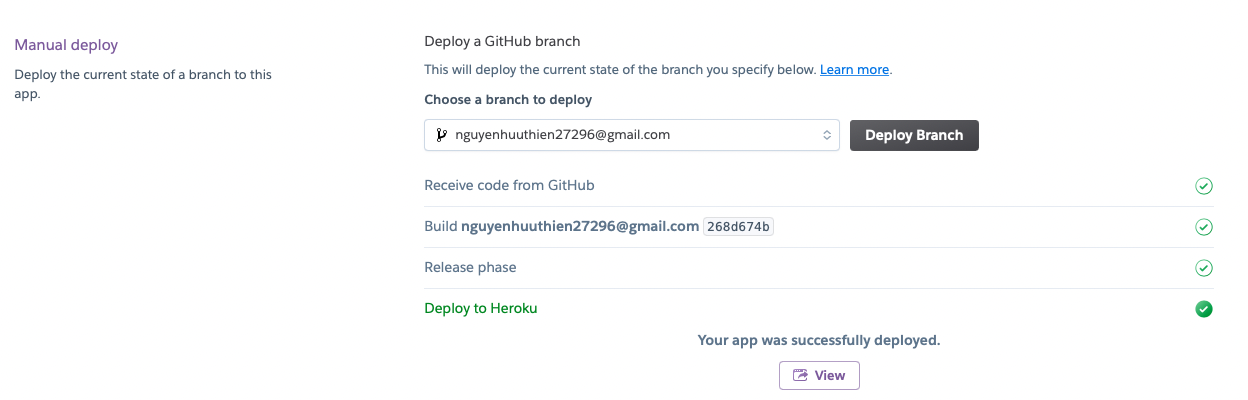


Figure 9: Deploy success



Figure 10: Website front-end