



Data Glacier

Your Deep Learning Partner

Deployment on Heroku

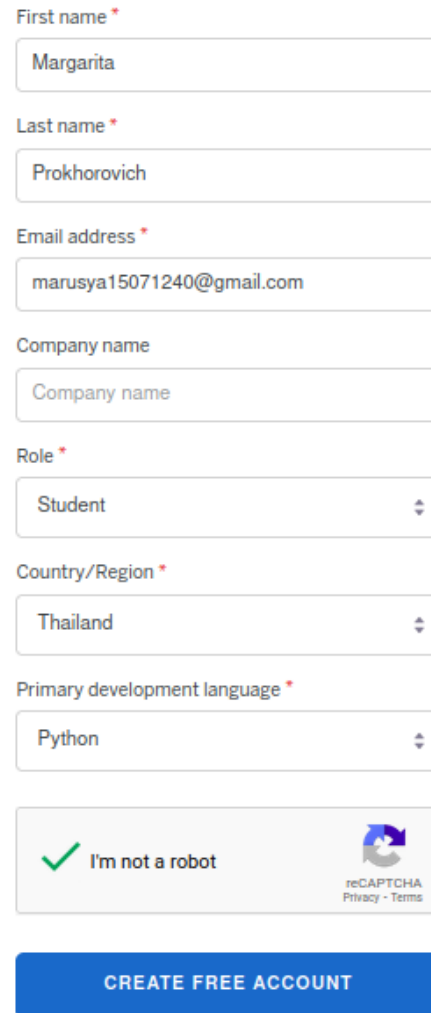
Raisin type prediction app

Name - Margarita Prokhorovich,
Batch code - LISUM09,
Submission date – 4 June, 2022,
Submitted to – Data Glacier
Email: marusya15071240@gmail.com

Creating Heroku account and app

In this project I use a model and app from the previous project – raisin type prediction app. I [described](#) model building and deployment on Flask sections in a previous week project. Here I'll describe a further development on Heroku.

First step is to log in in Heroku filling the form.



First name *

Last name *


Email address *

Company name

Role *

Country/Region *

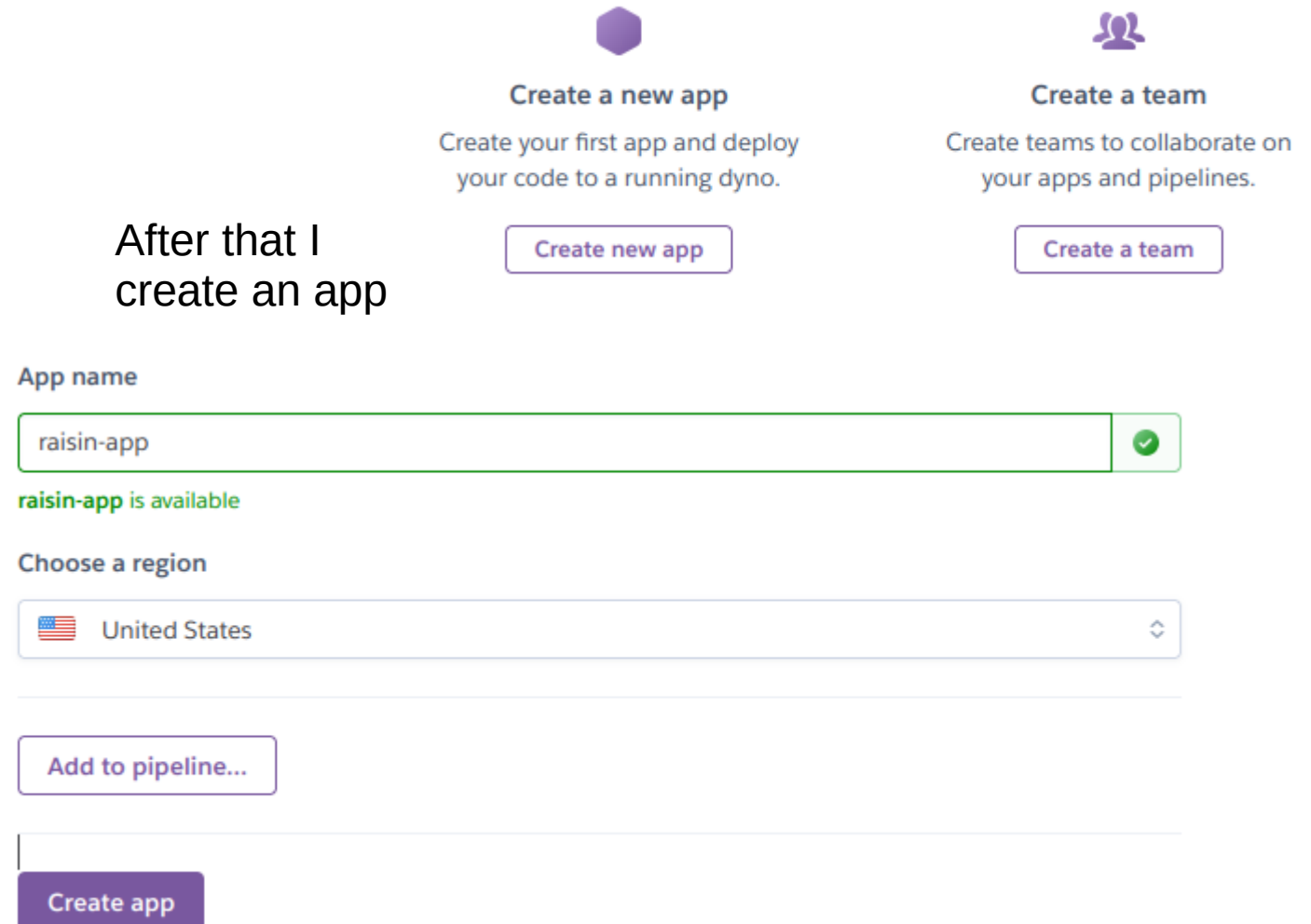
Primary development language *


☒ I'm not a robot 


[Privacy](#) [Terms](#)

CREATE FREE ACCOUNT

After that I create an app



 **Create a new app**
Create your first app and deploy your code to a running dyno.

 **Create a team**
Create teams to collaborate on your apps and pipelines.

[Create new app](#) [Create a team](#)

App name

raisin-app is available

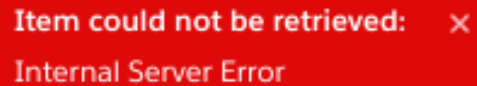
Choose a region

[Add to pipeline...](#)

Create app

Linking the app to a GitHub repository and start of the deployment

I choose GitHub as a deployment method. When pressing 'Connect to GitHub', an error occurs. Since there's no possibility to deploy the app through GitHub, I'll deploy it through Heroku Git.



Item could not be retrieved:
Internal Server Error

An error that occurs when trying to connect to GitHub.

Type 'heroku login' in the console.

```
nyak@nyak-ThinkPad-X220:~/Desktop/scripts$ heroku login
heroku: Press any key to open up the browser to login or q to exit
:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/2fb9ffb4-8e85-47e7-90a8-2e42991e9523?requestor=SFMyNTY.g2gDbQAAAw00S4yMzcuOC4yMTVuBgD4i6P2gAFiAAFRgA.zWhGZUaoEeCCiKfA1fkuzbrt_q0B-CkQGCH8e2N1sHQ
heroku: Waiting for login...
```

Logged in on the site.



Logged In

You can close this page and return to your CLI. It should now be logged in.

```
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ pip install
pipreqs
Defaulting to user installation because normal site-packages is
not writeable
Requirement already satisfied: pipreqs in /home/nyak/.local/lib/
python3.8/site-packages (0.4.11)
Requirement already satisfied: docopt in /home/nyak/.local/lib/p
ython3.8/site-packages (from pipreqs) (0.6.2)
Requirement already satisfied: yarg in /home/nyak/.local/lib/pyt
hon3.8/site-packages (from pipreqs) (0.1.9)
Requirement already satisfied: requests in /usr/lib/python3/dist
-packages (from yarg->pipreqs) (2.22.0)
WARNING: You are using pip version 22.0.4; however, version 22.1
is available.
You should consider upgrading via the '/usr/bin/python3 -m pip i
nsta ll --upgrade pip' command.
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ pipreqs .
INFO: Successfully saved requirements file in ./requirements.txt
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$
```

In addition to app and model files, templates and style files, it's also needed to add requirements.txt and Procfile files. To generate requirements.txt file, I can use pipreqs module. Since an error occurs, I also add gunicorn to the requirements. Also I create runtime.txt to specify python version.

Model deployment

I initialize a new git repository in a project week. Next step is to connect the app to git remote. After that I add and commit the files.

```
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ git init
Initialized empty Git repository in /home/nyak/Desktop/scripts/VC/Week5/.git/
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ heroku git:remote -a raisin-app
set git remote heroku to https://git.heroku.com/raisin-app.git
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ git add .
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ git commit -m 'Created heroku app'
[master (root-commit) aaacb91] Created heroku app
15 files changed, 833 insertions(+)
create mode 100644 .~lock.Deployment_on_Flask_snapshots.odp#
create mode 100644 .~lock.Raisin_Dataset.xlsx#
create mode 100644 Deployment_on_Flask_snapshots.odp
create mode 100644 Procfile
create mode 100644 README.md
create mode 100644 Raisin_Dataset.xlsx
create mode 100644 app.py
create mode 100644 model.ipynb
create mode 100644 model_raisin.pkl
create mode 100644 requirements.txt
create mode 100644 runtime.txt
create mode 100644 standart_scale.pkl
create mode 100644 static/css/style.css
create mode 100644 static/images/schema.png
create mode 100644 templates/index.html
```

After that I push the changes to heroku master branch. Process is completed successfully.

After that I type 'heroku open' to open the app and test it.

```
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ git push heroku master
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 4 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (21/21), 470.41 KiB | 14.70 MiB/s, done.
Total 21 (delta 0), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Python app detected
remote: -----> Using Python version specified in runtime.txt
remote: -----> Installing python-3.9.13
remote: -----> Installing pip 22.0.4, setuptools 60.10.0 and wheel 0.37.1
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: Collecting Flask==1.1.1
remote: Downloading Flask-1.1.1-py2.py3-none-any.whl (94 kB)
```

```
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$ heroku open
nyak@nyak-ThinkPad-X220:~/Desktop/scripts/VC/Week5$
```

Additional: deployment via GitHub


Since a week later the error with GitHub connection disappeared, I also try this method of deployment.

1. Chose a repository

[Connect to GitHub](#)


Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

 maryprokh

repo-name [Search](#)

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

 maryprokh/Machine-learning-examples [Connect](#)

 maryprokh/data_analysis_examples [Connect](#)

 maryprokh/VC [Connect](#)

2. Chose a branch for manual deployment

[Manual deploy](#)

Deploy the current state of a branch to this app.

Deploy a GitHub branch


This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

 heroku_app [Deploy Branch](#)

3. Deployment is in progress

[Receive code from GitHub](#) 

Build heroku_app 6aec3ce8 

```
-----> Building on the Heroku-20 stack
-----> Using buildpack: heroku/python
-----> Python app detected
-----> Using Python version specified in runtime.txt
-----> No change in requirements detected, installing from cache
-----> Using cached install of python-3.9.13
-----> Installing pip 22.0.4, setuptools 60.10.0 and wheel 0.37.1
-----> Installing SQLite3
```

☒ Autoscroll with output

[View build log](#)

4. Deployment is complete

[Receive code from GitHub](#) 

Build heroku_app 6aec3ce8 

Release phase 

[Deploy to Heroku](#) 

Your app was successfully deployed.

[View](#)

Testing the app

The [app](#) opens in a new tab, predictions work. Deployment on Heroku is completed.

←

→

↺

🔒

https://raisin-app.herokuapp.com/predict

☆

🛡️

Predict Raisin Type

53698

348.223231

197.752897

0.823104

56089

0.731341

938.705

Predict

Raisin type should be Kecimen

Features that determine type of raisin are morphological features that were extracted based on Image processing:

- Area: Gives the number of pixels within the boundaries of the raisin grain.
- Perimeter: It measures the environment by calculating the distance between the boundaries of the raisin grain and the pixels around it.
- MajorAxisLength: Gives the length of the main axis, which is the longest line that can be drawn on the raisin grain.
- MinorAxisLength: Gives the length of the small axis, which is the shortest line that can be drawn on the raisin grain.
- Eccentricity: It gives a measure of the eccentricity of the ellipse, which has the same moments as raisins.
- ConvexArea: Gives the number of pixels of the smallest convex shell of the region formed by the raisin grain.
- Extent: Gives the ratio of the region formed by the raisin grain to the total pixels in the bounding box.

[Full study text](#)

The diagram illustrates the hardware setup for the raisin image processing system. It includes a desktop computer with a monitor, keyboard, and mouse. The monitor displays an 'IMAGE SAMPLE' of raisins. A 'CAMERA' is positioned above a 'BOX' containing the raisins, connected by a cable. A 'LIGHTING SYSTEM' is also connected to the box. The entire setup is connected to the 'COMPUTER'.

Thank You

Raisin type prediction app
Deployment on Heroku