

Name: Benjamin Prud'homme

Batch: LISUM12

Date: September 4, 2022

Submitted to: Data Glacier

Deployment on Heroku

Step 1: Download Iris dataset

Id	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa

- Features: SepalLength, SepalWidth, PetalLength, PetalWidth
 - All floats, in inches
- To predict: Species (Iris-setosa, Iris-versicolor, Iris-virginica)

Step 2: Build ML classification model (model.py)

```
model.py 5 ×  app.py 2  index.html
Users > benjaminprudhomme > Downloads > Flask-Deployment > model.py > ...
1  # Importing the libraries
2  import numpy as np
3  import pandas as pd
4  import pickle
5
6  from sklearn import datasets
7  iris = datasets.load_iris()
8
9  # df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
10
11 # df["target"] = iris.target
12
13 x = iris.data
14 y = iris.target
15
16 from sklearn.model_selection import train_test_split
17 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.5)
18
19 from sklearn import neighbors
20 classifier=neighbors.KNeighborsClassifier()
21
22 classifier.fit(x_train,y_train)
23
24 pickle.dump(classifier, open('model.pkl', 'wb'))
25 model = pickle.load(open('model.pkl', 'rb'))
```

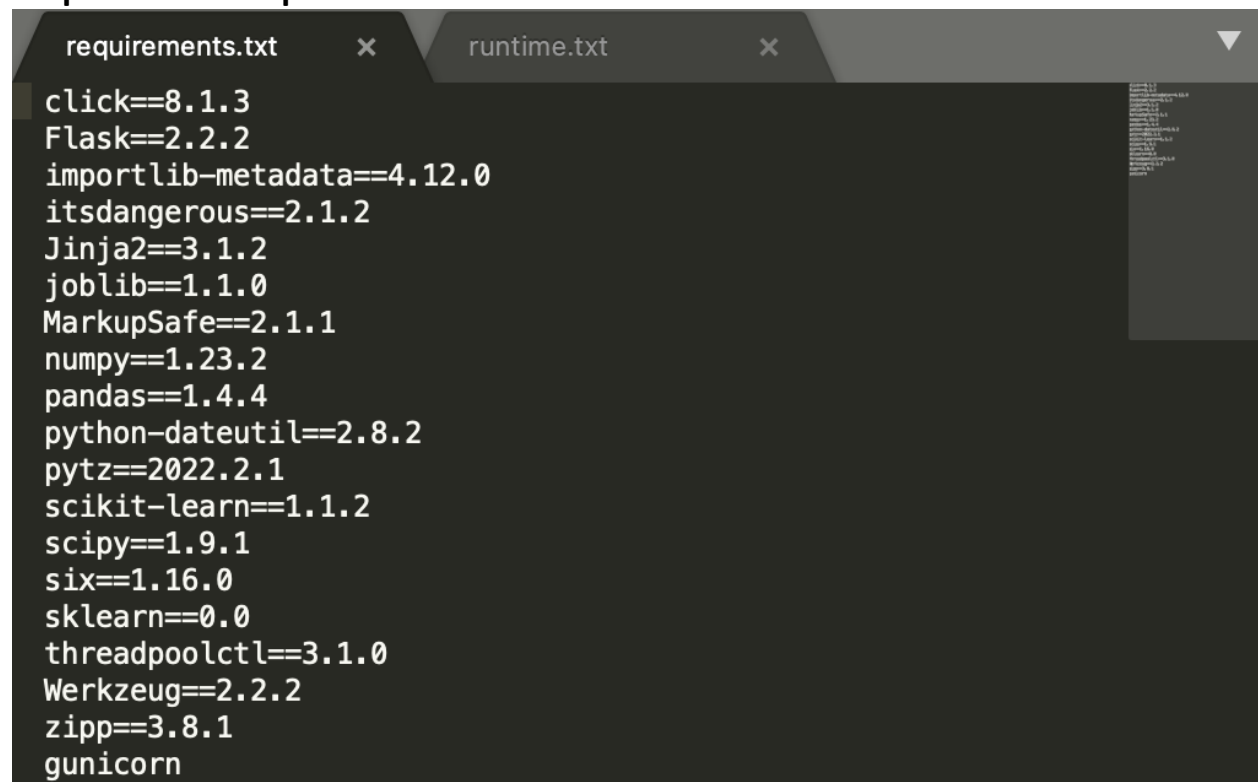
Step 3: Write Flask app (app.py)

```
model.py 5  app.py 2 x  index.html
Users > benjaminprudhomme > Downloads > Flask-Deployment > app.py > ...
1  import numpy as np
2  from flask import Flask, request, render_template
3  import pickle
4
5  app = Flask(__name__)
6  model = pickle.load(open('model.pkl', 'rb'))
7
8  @app.route('/')
9  def home():
10     return render_template('index.html')
11
12  @app.route('/predict',methods=['POST'])
13  def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17
18     float_features = [float(x) for x in request.form.values()]
19     final_features = [np.array(float_features)]
20     prediction = model.predict(final_features)
21
22     class_num = round(prediction[0])
23     if class_num == 0:
24         output = "Iris-setosa"
25     elif class_num == 1:
26         output = "Iris-versicolor"
27     elif class_num == 2:
28         output = "Iris-virginica"
29     return render_template('index.html', prediction_text='Predicted species: {}'.format(output))
30
31  if __name__ == "__main__":
32     app.run(debug=True)
```

Step 4: Create Procfile

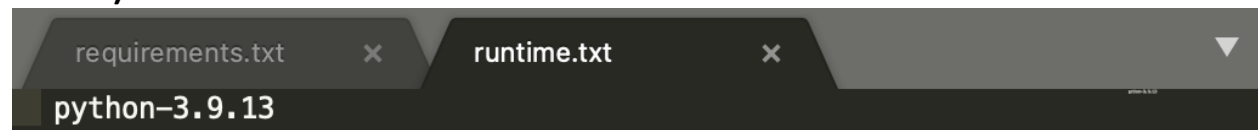
```
Procfile
web: gunicorn app:app
```

Step 5: Create requirements.txt

A screenshot of a code editor with two tabs: 'requirements.txt' and 'runtime.txt'. The 'requirements.txt' tab is active and contains a list of Python dependencies with version constraints. The 'runtime.txt' tab is also visible but empty. The code is as follows:

```
click==8.1.3
Flask==2.2.2
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
MarkupSafe==2.1.1
numpy==1.23.2
pandas==1.4.4
python-dateutil==2.8.2
pytz==2022.2.1
scikit-learn==1.1.2
scipy==1.9.1
six==1.16.0
sklearn==0.0
threadpoolctl==3.1.0
Werkzeug==2.2.2
zipp==3.8.1
unicorn
```

Step 6: Create runtime.txt to force Python 3.9.13 (the version that's on my local device)

A screenshot of a code editor with two tabs: 'requirements.txt' and 'runtime.txt'. The 'runtime.txt' tab is active and contains the text 'python-3.9.13'. The 'requirements.txt' tab is also visible but empty. The code is as follows:

```
python-3.9.13
```

Step 7: Upload to github (iris-classification-bprudhomme (multiple commits present due to debugging))

The screenshot shows a GitHub repository page for user boardPro43119, repository iris-classification-bprudhomme. The repository is public and has 1 branch (main) and 0 tags. The commit history shows 9 commits, with the latest commit being 'Remove 1 requirement (again)' 27 minutes ago. The file list includes static, templates, .DS_Store, Flask-Deployment.docx, Flask-Deployment.pdf, Procfile, README.md, app.py, model.pkl, model.py, price.csv, requirements.txt, and runtime.txt. The right sidebar shows the repository's About section, which is currently empty, and the Releases and Packages sections, which also show no published content. The Environments section shows one active environment named 'iris-classification-bprudhomme'.

File	Commit Message	Time Ago
static	Redone in py39 environment	9 hours ago
templates	Restore original file structure	13 hours ago
.DS_Store	Add gunicorn requirement	30 minutes ago
Flask-Deployment.docx	Initial commit	yesterday
Flask-Deployment.pdf	Initial commit	yesterday
Procfile	Initial commit	yesterday
README.md	Restore original file structure	13 hours ago
app.py	Initial commit	yesterday
model.pkl	Redone in py39 environment	9 hours ago
model.py	Initial commit	yesterday
price.csv	Restore original file structure	13 hours ago
requirements.txt	Remove 1 requirement (again)	27 minutes ago
runtime.txt	Add runtime.txt	9 hours ago

Step 8: Create Heroku account (not pictured)

Step 9: Create Heroku app (same name as github repo)

The screenshot shows the Heroku dashboard for a personal account. A welcome message states: 'Welcome to Heroku. Now that your account has been set up, here's how to get started.' Below this, there is a notification banner stating: 'Starting November 28th, 2022, free Heroku Dynos, free Heroku Postgres, and free Heroku Data for Redis* will no longer be available. If you have apps using any of these resources, you must upgrade to paid plans by this date to ensure your apps continue to run and to retain your data. For students, we will announce a new program by the end of September. Learn more'. A search bar is present with the text 'Filter apps and pipelines'. The list of apps shows one app named 'iris-classification-bprudhomme' with a status of 'Python · heroku-22 · United States'.

Step 10: Connect to, deploy w/ github

Deployment method



 Heroku Git
Use Heroku CLI

 GitHub
Connected

 Container Registry
Use Heroku CLI

App connected to GitHub


Code diffs, manual and auto deploys are available for this app.

Connected to  [boardPro43119/iris-classification-bprudhomme](#) by  [boardPro43119](#) Disconnect...

Releases in the [activity feed](#) link to GitHub to view commit diffs

Automatic deploys


Enables a chosen branch to be automatically deployed to this app.

 You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).

Choose a branch to deploy

 main

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys


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

 main Deploy Branch

Receive code from GitHub

✓

Build main 4f89b64a

✓


Release phase

✓

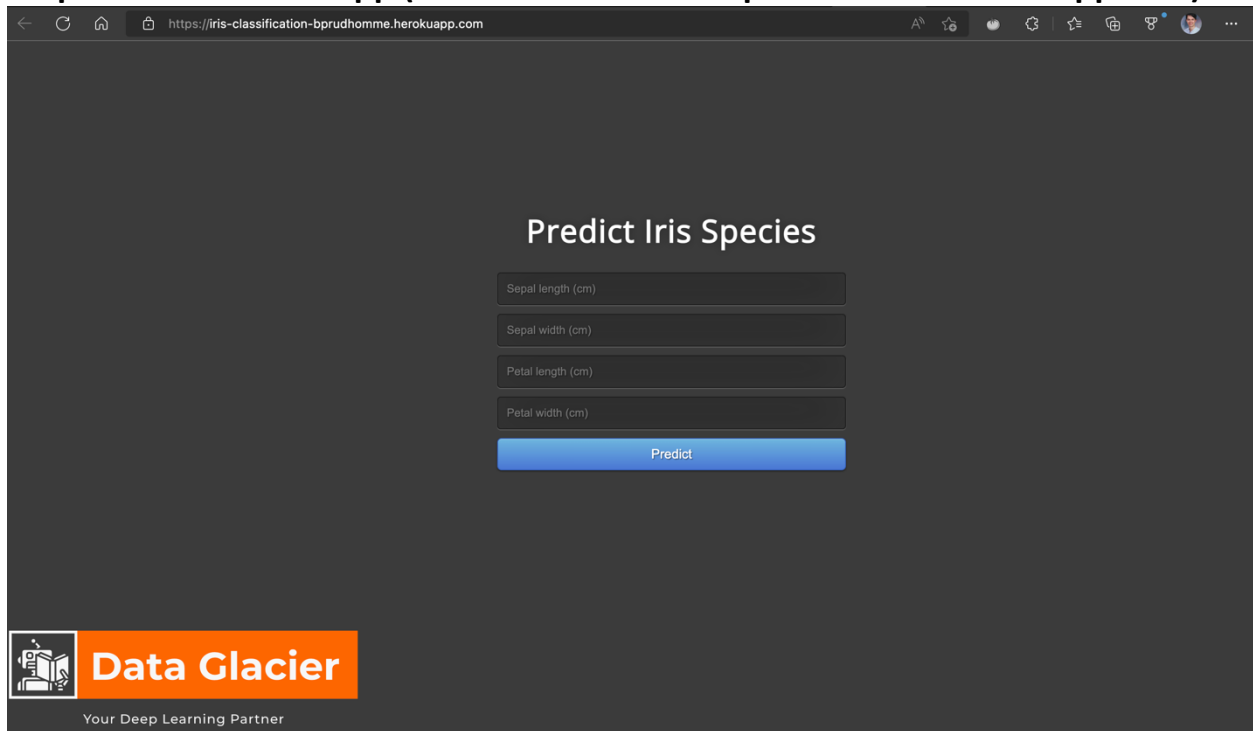
Deploy to Heroku

✓

Your app was successfully deployed.

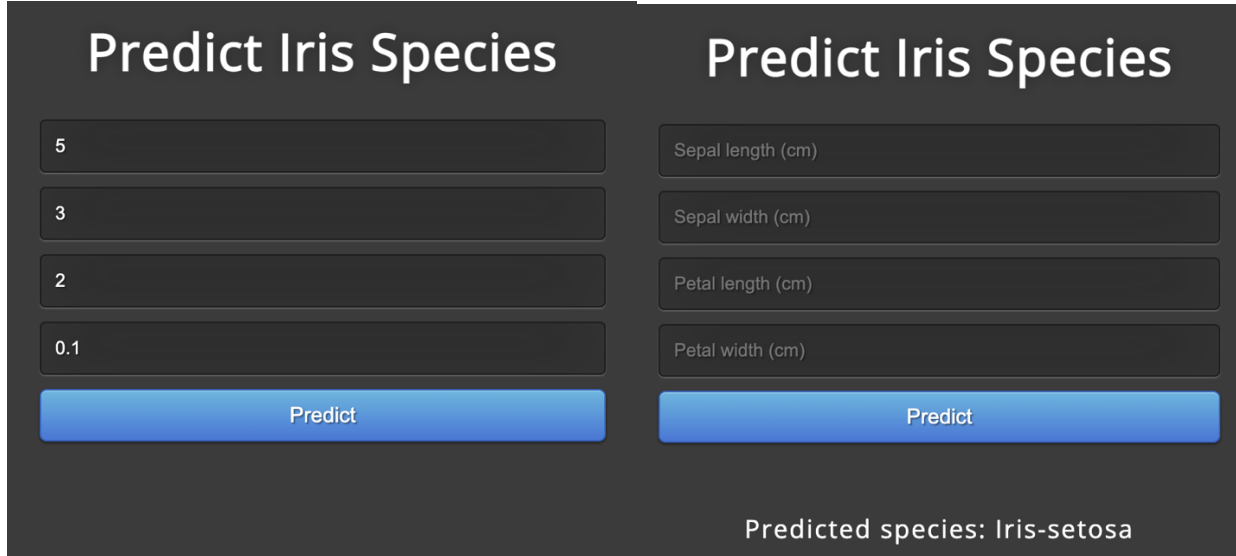
 View

Step 11: Launch web app (at iris-classification-bprudhomme.herokuapp.com)



The screenshot shows a web browser window with the URL `https://iris-classification-bprudhomme.herokuapp.com`. The page has a dark gray background and features a central form titled "Predict Iris Species". The form contains four input fields for "Sepal length (cm)", "Sepal width (cm)", "Petal length (cm)", and "Petal width (cm)", each with a light gray border. Below these fields is a blue "Predict" button. In the bottom left corner, there is a logo for "Data Glacier" with the tagline "Your Deep Learning Partner".

Step 12: Test model



This block displays two side-by-side screenshots of the "Predict Iris Species" web app. The left screenshot shows the input fields with the values 5, 3, 2, and 0.1 entered, and the "Predict" button highlighted. The right screenshot shows the same interface but with the input fields empty and the "Predict" button highlighted. Below the right screenshot, the text "Predicted species: Iris-setosa" is displayed.