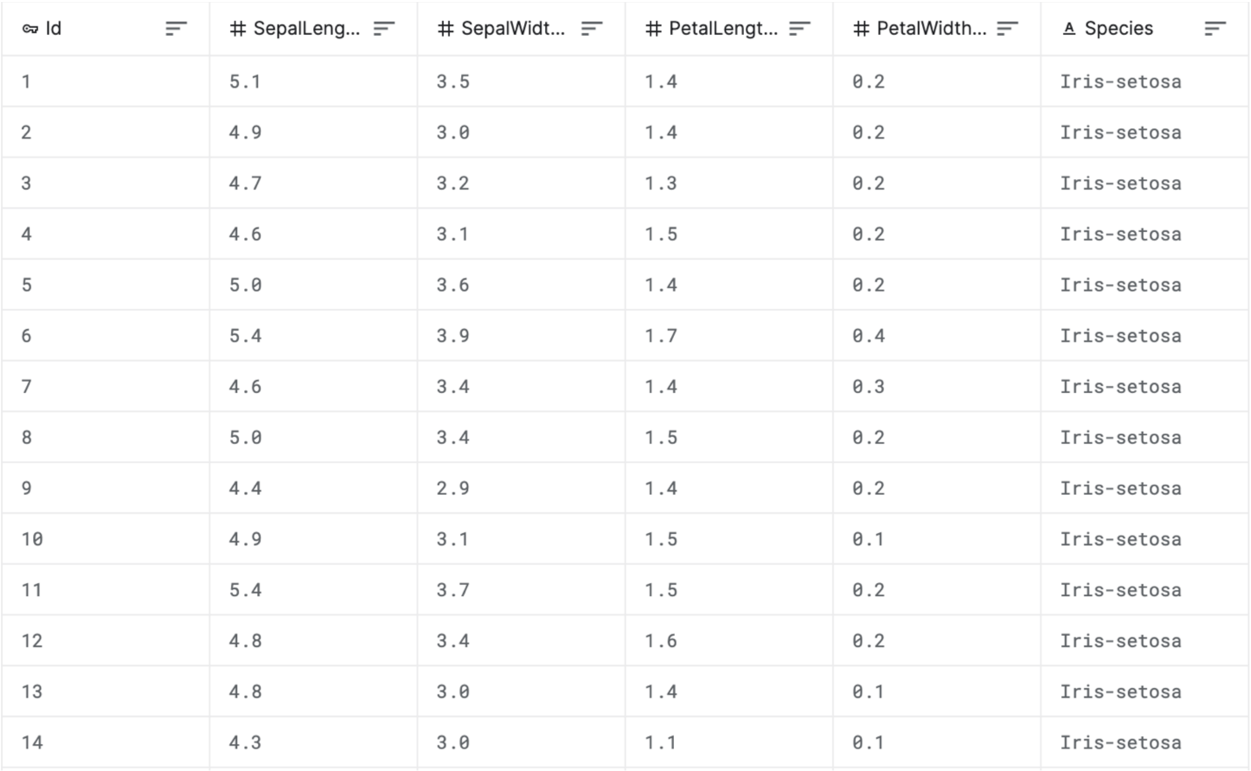
Name: Benjamin Prud’homme
Batch: LISUM12
Date: September 4, 2022
Submitted to: Data Glacier

Deployment on Heroku

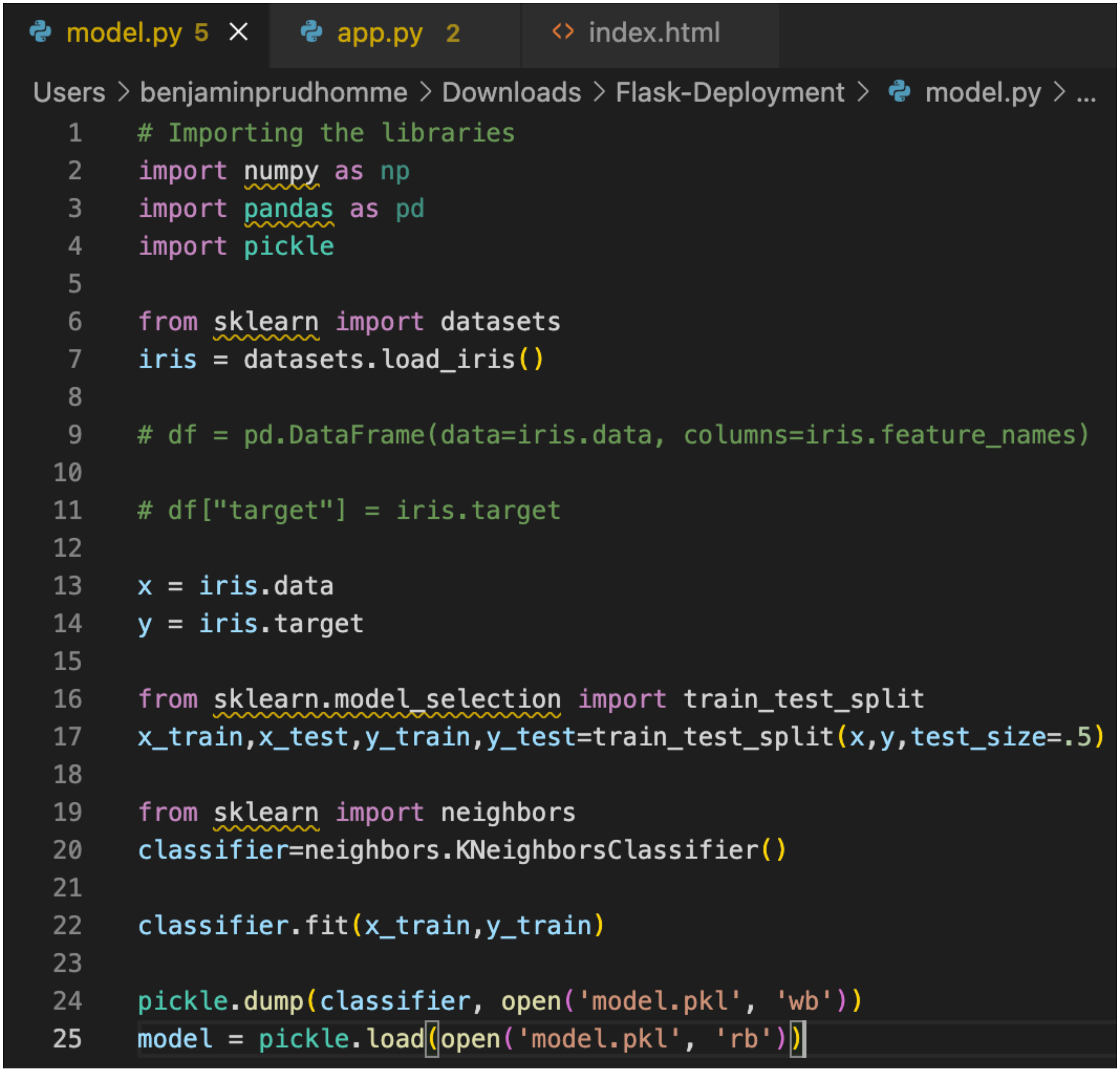
Step 1: Download Iris dataset



Id	# SepalLeng...	# SepalWidt...	# PetalLengt...	# 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

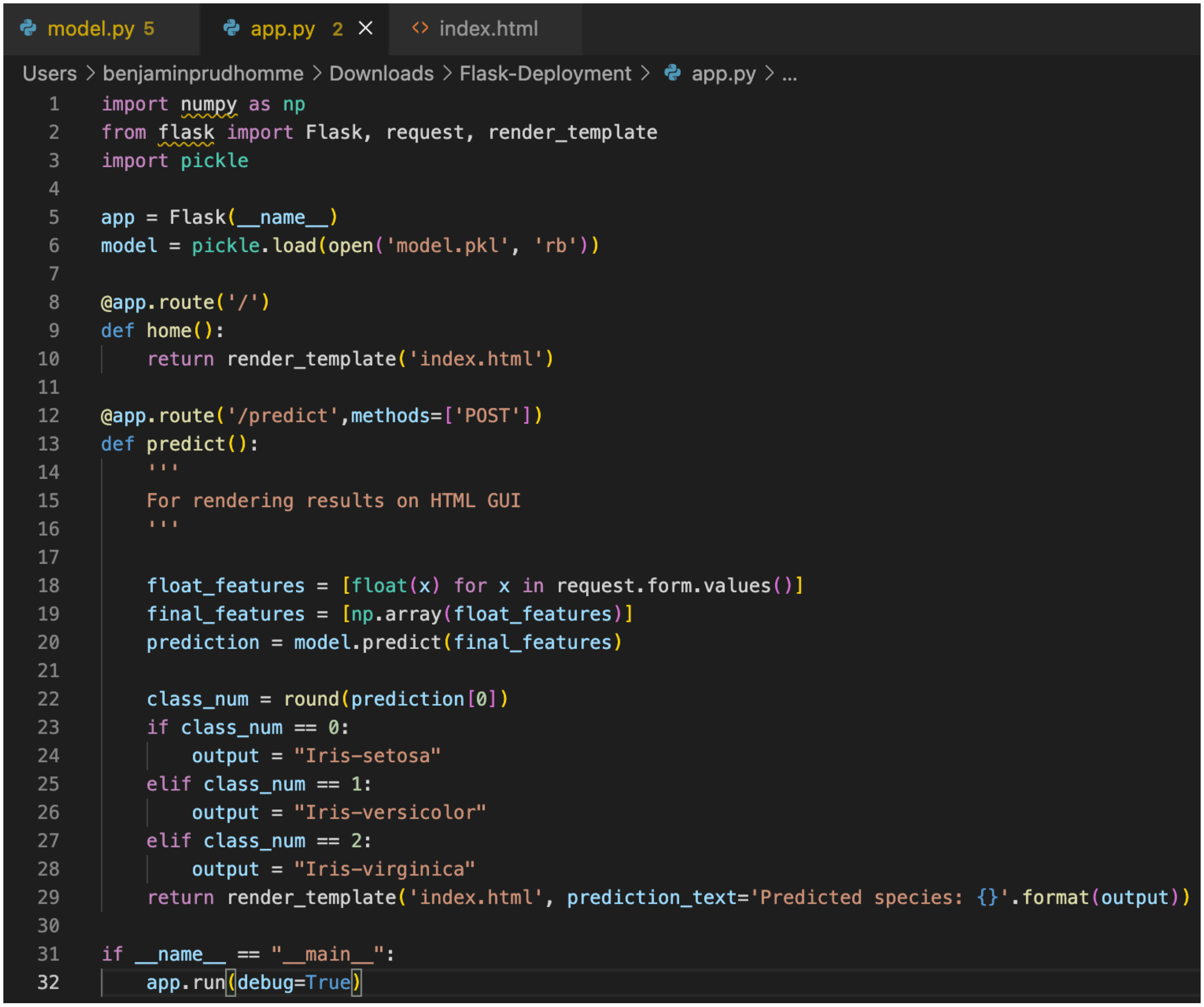
- Features: SepalLength, SepalWidth, PetalLength, PetalWidth
    - All floats, in inches
- To predict: Species (Iris-setosa, Iris-versicolor, Iris-virginica)

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


model.py 5 ×
app.py 2
index.html
Users > benjaminprudhomme > Downloads > Flask-Deployment > model.py > ...

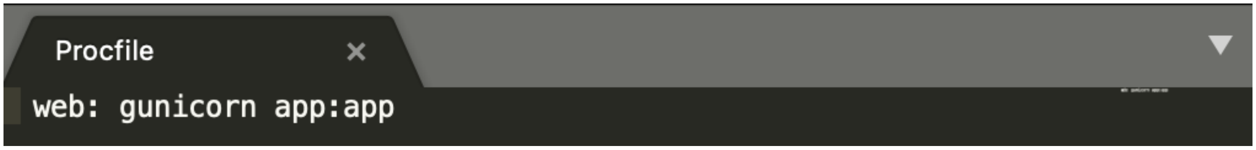
```python
# Importing the libraries
import numpy as np
import pandas as pd
import pickle

from sklearn import datasets
iris = datasets.load_iris()

# df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

# df["target"] = iris.target

x = iris.data
y = iris.target

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.5)

from sklearn import neighbors
classifier=neighbors.KNeighborsClassifier()

classifier.fit(x_train,y_train)

pickle.dump(classifier, open('model.pkl', 'wb'))
model = pickle.load(open('model.pkl', 'rb'))
```

Step 3: Write Flask app (app.py)


model.py 5
app.py 2
index.html
Users > benjaminprudhomme > Downloads > Flask-Deployment > app.py > ...

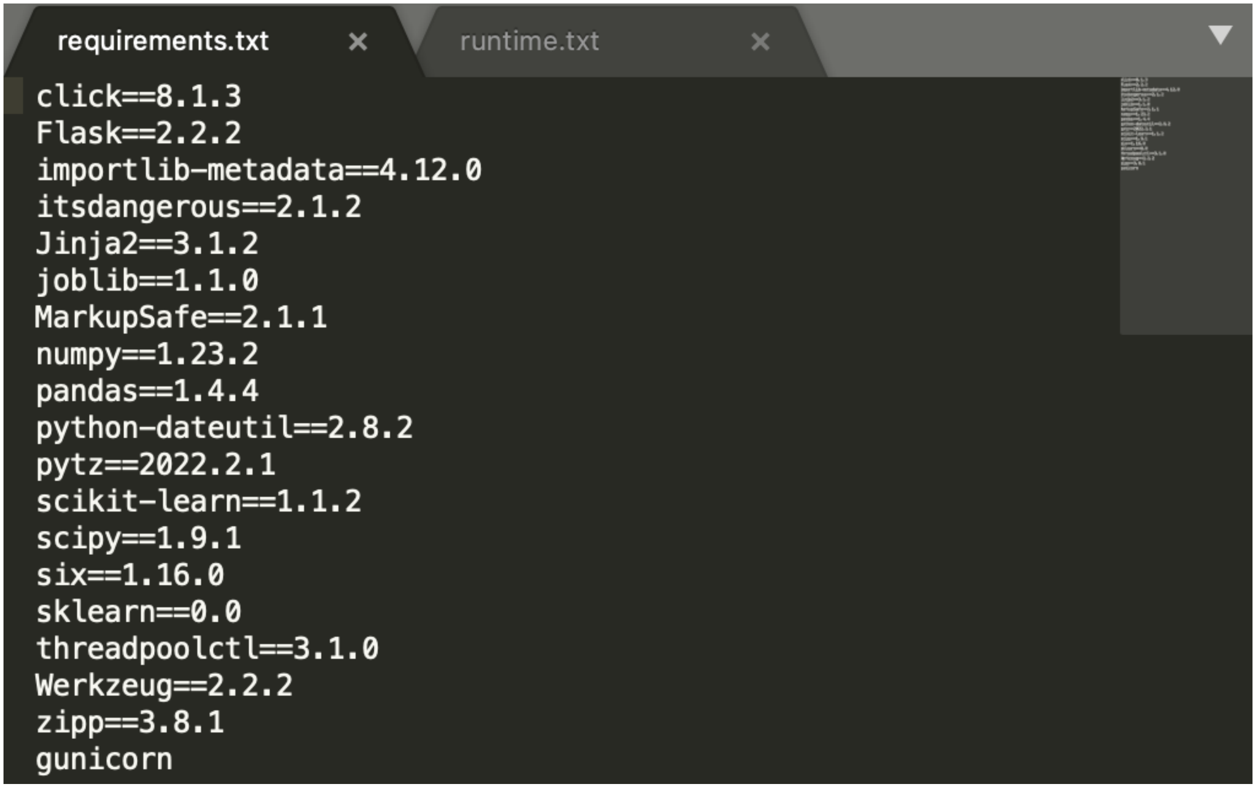
```python
import numpy as np
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''

    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    prediction = model.predict(final_features)

    class_num = round(prediction[0])
    if class_num == 0:
        output = "Iris-setosa"
    elif class_num == 1:
        output = "Iris-versicolor"
    elif class_num == 2:
        output = "Iris-virginica"
    return render_template('index.html', prediction_text='Predicted species: {}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)
```

Step 4: Create Procfile



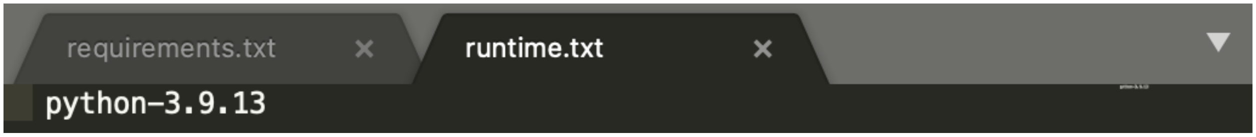
```
web: gunicorn app:app
```

Step 5: Create requirements.txt



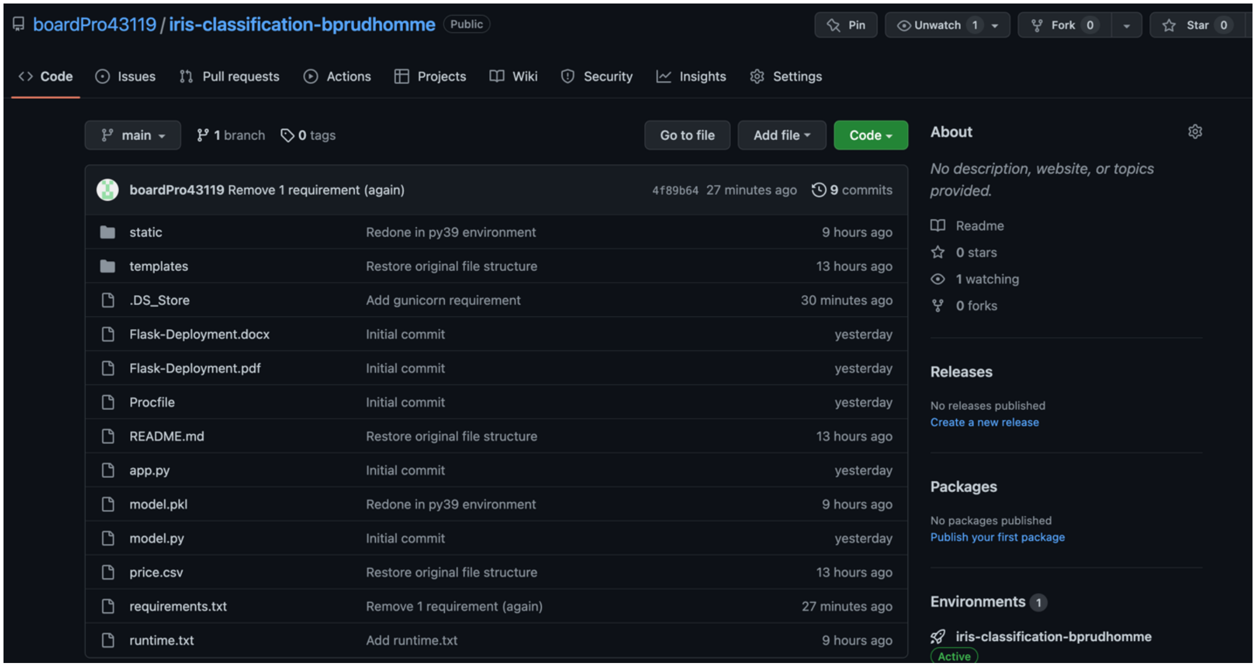
```
click==8.1.3
Flask==2.2.2
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
MarkupSafe==2.1.1
numpy==1.23.2
pandas==1.4.4
python-dateutil==2.8.2
pytz==2022.2.1
scikit-learn==1.1.2
scipy==1.9.1
six==1.16.0
sklearn==0.0
threadpoolctl==3.1.0
Werkzeug==2.2.2
zipp==3.8.1
gunicorn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

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



```
python-3.9.13
```

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



boardPro43119 Remove 1 requirement (again) 4f89b64 27 minutes ago 9 commits

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

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

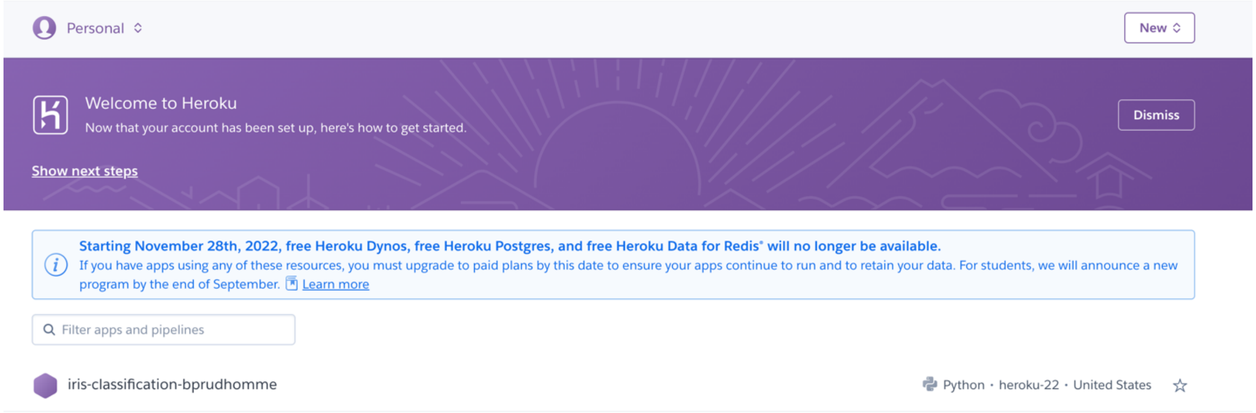
No packages published
Publish your first package

Environments 1

iris-classification-bprudhomme Active

Step 8: Create Heroku account (not pictured)

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



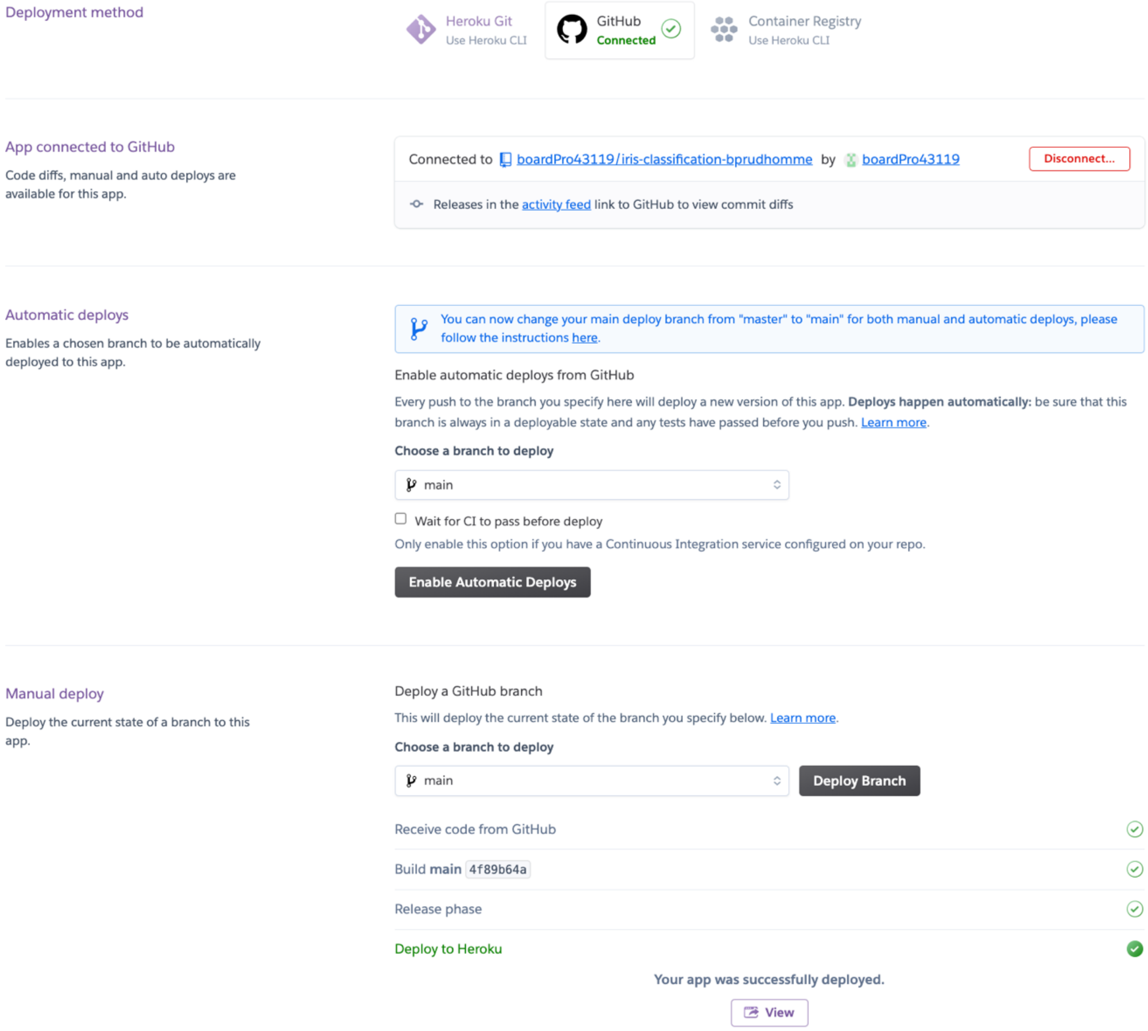
Welcome to Heroku

Now that your account has been set up, here's how to get started.

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

iris-classification-bprudhomme Python · heroku-22 · United States

Step 10: Connect to, deploy w/ github



Deployment method

App connected to GitHub

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

Connected to boardPro43119/iris-classification-bprudhomme by boardPro43119

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.

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

Receive code from GitHub ✓

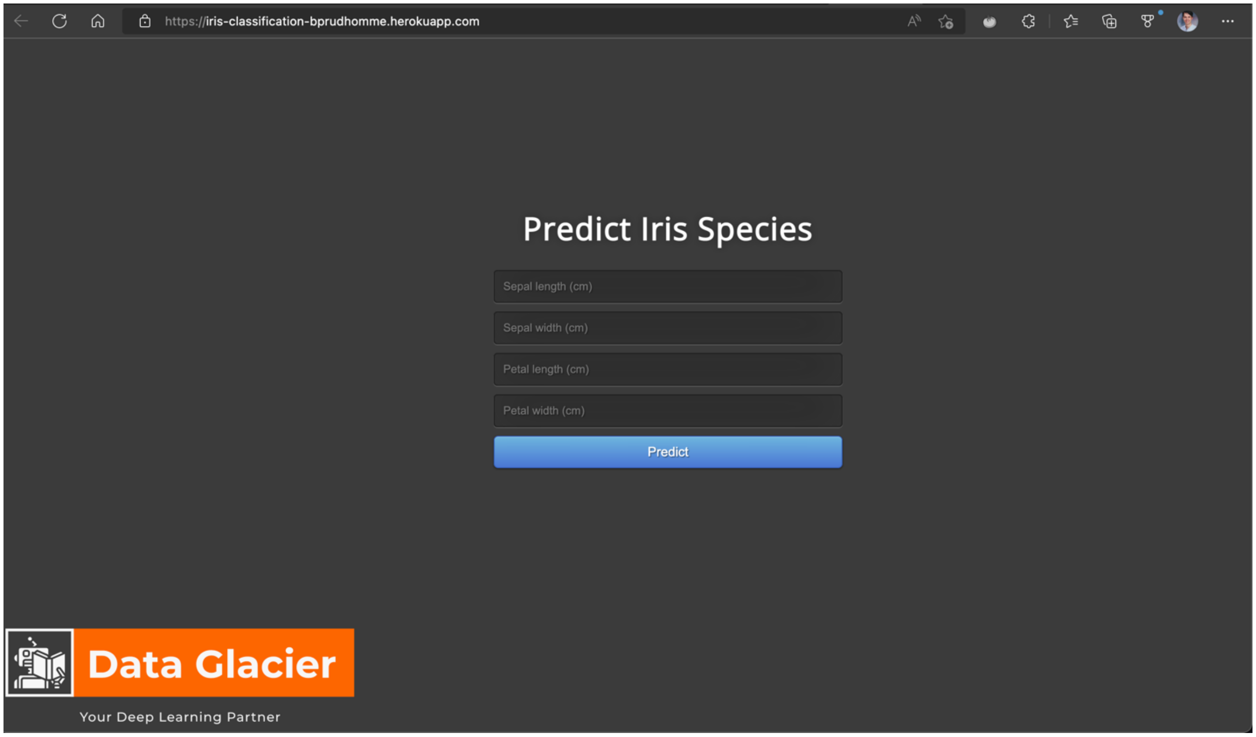
Build **main** `4f89b64a` ✓

Release phase ✓

Deploy to Heroku ✓

Your app was successfully deployed.

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



https://iris-classification-bprudhomme.herokuapp.com

Predict Iris Species

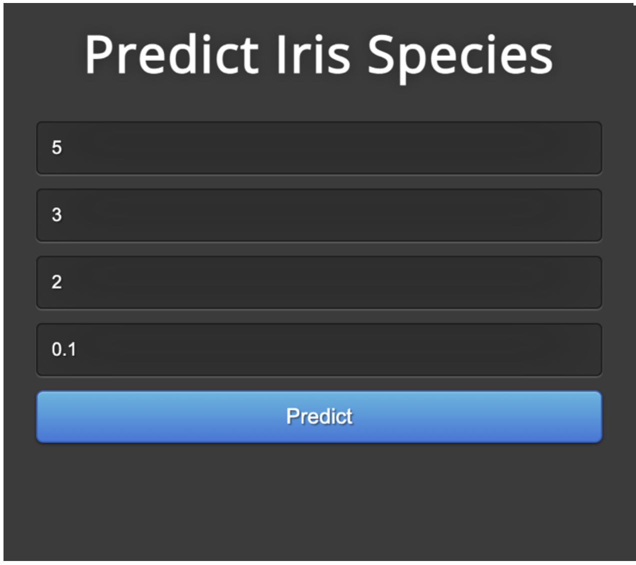
Sepal length (cm)

Sepal width (cm)

Petal length (cm)

Petal width (cm)

Step 12: Test model



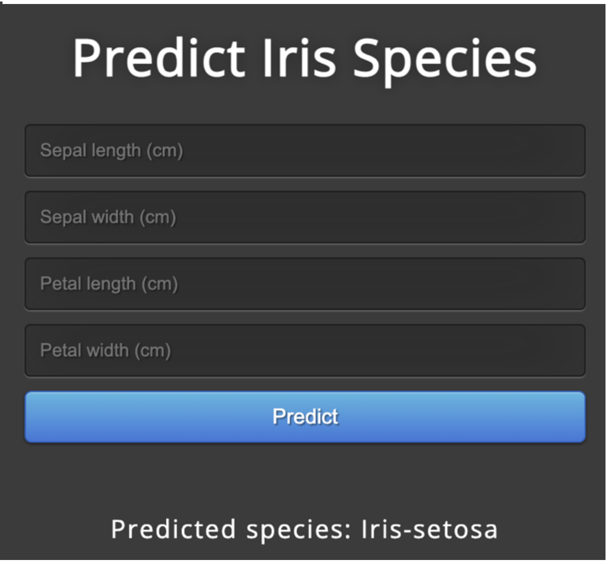
Predict Iris Species

5

3

2

0.1



Predict Iris Species

Sepal length (cm)

Sepal width (cm)

Petal length (cm)

Petal width (cm)

Predicted species: Iris-setosa