Kristina Kaliagina
Submission date: 04.08.2022
Submitted to: Data Glacier

Heroku deployment of ML App

Link to final app: https://deployment-by-heroku.herokuapp.com/

Steps of deployment:

1. Dataset

The data was taken from the library of *sklearn.datasets*, which are called *'iris'* and contain information about these flowers:

- petal length;
- petal width;
- sepal length;
- sepal width;
- class of irises.

2. Build and save the model

A LogisticRegression model from the *sklearn* package was used to predict the class of irises, saved in pickle format - *model.pkl* (filename - *build.py* in folder *"model"*).

model > build.py > ...

```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

import joblib

# prepare the train and test data
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y)

# train a model
model = LogisticRegression(max_iter=5000)
model.fit(X_train, y_train)

# save the model
joblib.dump(model, 'iris_model.pkl')

```

3. Creating Web App using Flask (filename - *app.py*)

app.py > predict

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

app = Flask(__name__, template_folder='templates')

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

@app.route('/predict', methods=["POST"])
def predict():
    petal_length = request.form['petal_length']
    sepal_length = request.form['sepal_length']
    petal_width = request.form['petal_width']
    sepal_width = request.form['sepal_width']

    sample_data = [sepal_length, sepal_width, petal_length, petal_width]
    clean_data = [float(i) for i in sample_data]

    ex1 = np.array(clean_data).reshape(1,-1)
    model=joblib.load('model/iris_model.pkl')
    result_prediction = model.predict(ex1)

    if result_prediction == 0:
        predicted_class = 'Iris-setosa'
    elif result_prediction == 1:
        predicted_class = 'Iris-versicolor'
    else:
        predicted_class = 'Iris-virginica'


    return render_template('web.html', prediction_text='Type of Iris: {}'.format(predicted_class))

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

4. Adding several files

“Procfile”:

Procfile

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

“requirements.txt”:

requirements.txt

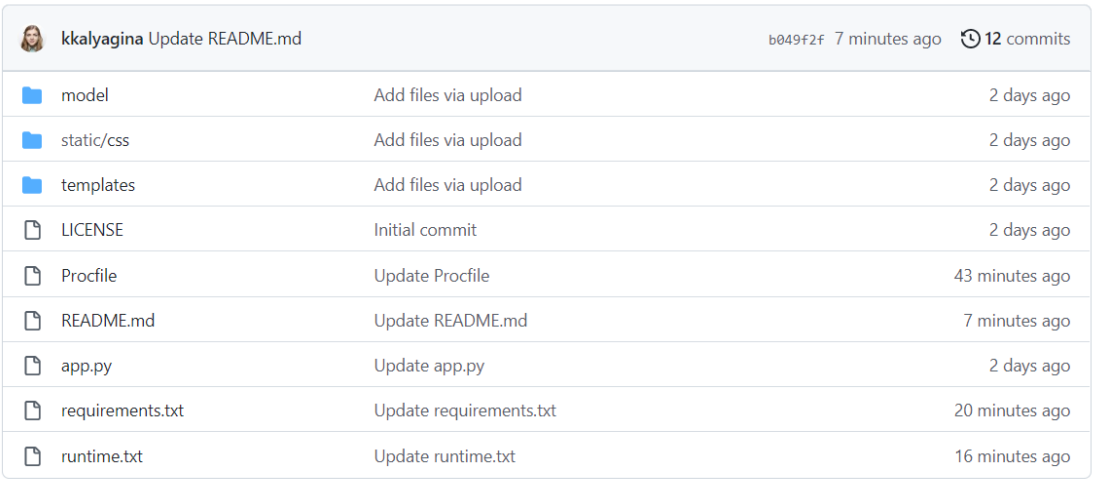
```
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.9.2
scipy>=0.15.1
scikit-learn>=0.18
matplotlib>=1.4.3
pandas>=0.19
```

"runtime.txt":

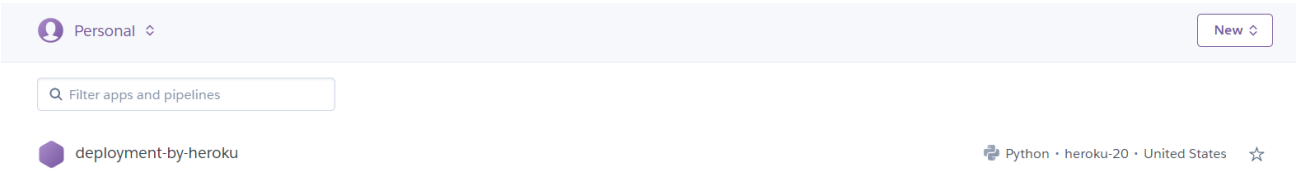
runtime.txt

```
python-3.8.5
```

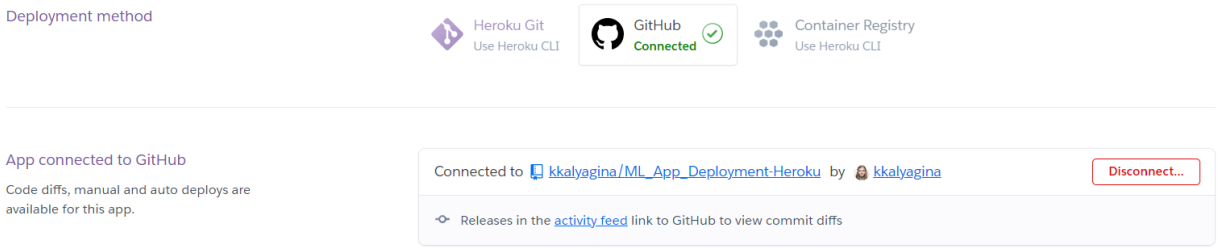
5. Commiting code in online repo "*ML_App_Deployment-Heroku*"



6. Account and app *"deployment-by-heroku"* creation in Heroku



7. Linking of online repo to Heroku



8. Deployment of ML Model

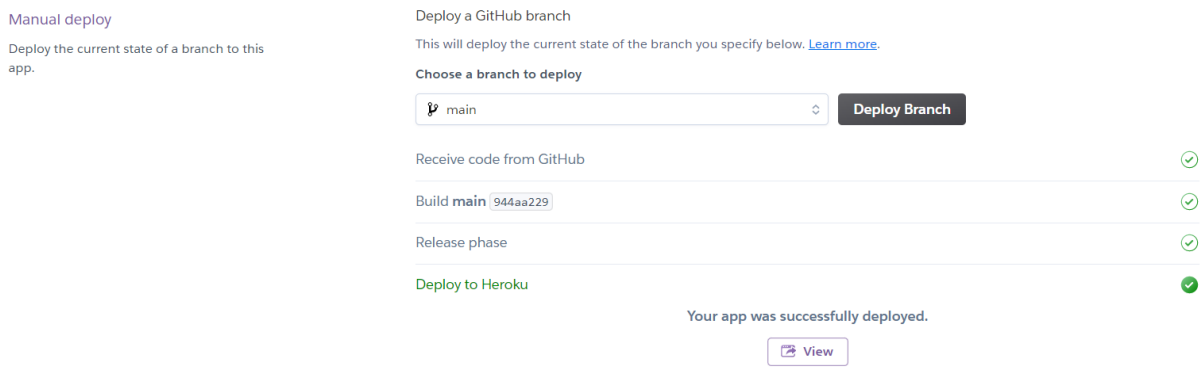
Build Log:

```
-----> Building on the Heroku-20 stack
-----> Using buildpack: heroku/python
-----> Python app detected
-----> Using Python version specified in runtime.txt
!     Python has released a security update! Please consider upgrading to python-3.8.13
      Learn More: https://devcenter.heroku.com/articles/python-runtimes
-----> Python version has changed from python-3.10.6 to python-3.8.5, clearing cache
-----> Requirements file has been changed, clearing cached dependencies
-----> Installing python-3.8.5
-----> Installing pip 22.1.2, setuptools 60.10.0 and wheel 0.37.1
-----> Installing SQLite3
-----> Installing requirements with pip
      Collecting Flask==1.1.1
        Downloading Flask-1.1.1-py2.py3-none-any.whl (94 kB)
      Collecting gunicorn==19.9.0
        Downloading gunicorn-19.9.0-py2.py3-none-any.whl (112 kB)
      Collecting itsdangerous==1.1.0
        Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
      Collecting Jinja2==2.10.1
        Downloading Jinja2-2.10.1-py2.py3-none-any.whl (124 kB)
      Collecting MarkupSafe==1.1.1
        Downloading MarkupSafe-1.1.1-cp38-cp38-manylinux2010_x86_64.whl (32 kB)
      Collecting Werkzeug==0.15.5
        Downloading Werkzeug-0.15.5-py2.py3-none-any.whl (328 kB)
      Collecting numpy>=1.9.2
        Downloading numpy-1.23.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
      Collecting scipy>=0.15.1
        Downloading scipy-1.9.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (43.4 MB)
```

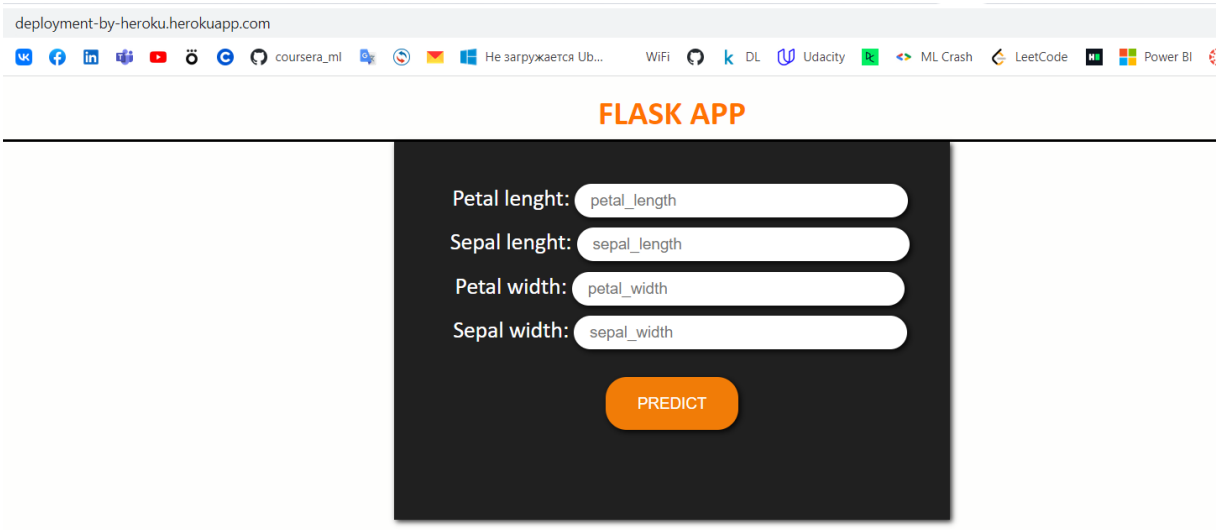
```
      Collecting scikit-learn>=0.18
        Downloading
scikit_learn-1.1.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (31.2
MB)
      Collecting matplotlib>=1.4.3
        Downloading
matplotlib-3.5.2-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.whl (11.3 MB)
      Collecting pandas>=0.19
        Downloading
pandas-1.4.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
      Collecting joblib>=0.16.0
        Downloading joblib-1.1.0-py2.py3-none-any.whl (306 kB)
      Collecting requests==2.25.1
        Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
      Collecting click>=5.1
        Downloading click-8.1.3-py3-none-any.whl (96 kB)
      Collecting chardet<5,>=3.0.2
        Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
      Collecting urllib3<1.27,>=1.21.1
        Downloading urllib3-1.26.11-py2.py3-none-any.whl (139 kB)
      Collecting certifi>=2017.4.17
        Downloading certifi-2022.6.15-py3-none-any.whl (160 kB)
      Collecting idna<3,>=2.5
        Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
      Collecting threadpoolctl>=2.0.0
        Downloading threadpoolctl-3.1.0-py3-none-any.whl (14 kB)
      Collecting python-dateutil>=2.7
        Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
      Collecting pyparsing>=2.2.1
        Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
      Collecting cycler>=0.10
        Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)
      Collecting kiwisolver>=1.0.1
        Downloading
kiwisolver-1.4.4-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.whl (1.2 MB)
      Collecting pillow>=6.2.0
        Downloading Pillow-9.2.0-cp38-cp38-manylinux_2_28_x86_64.whl (3.2 MB)
      Collecting packaging>=20.0
        Downloading packaging-21.3-py3-none-any.whl (40 kB)
      Collecting fonttools>=4.22.0
        Downloading fonttools-4.34.4-py3-none-any.whl (944 kB)
      Collecting pytz>=2020.1
        Downloading pytz-2022.1-py2.py3-none-any.whl (503 kB)
      Collecting six>=1.5
        Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
```

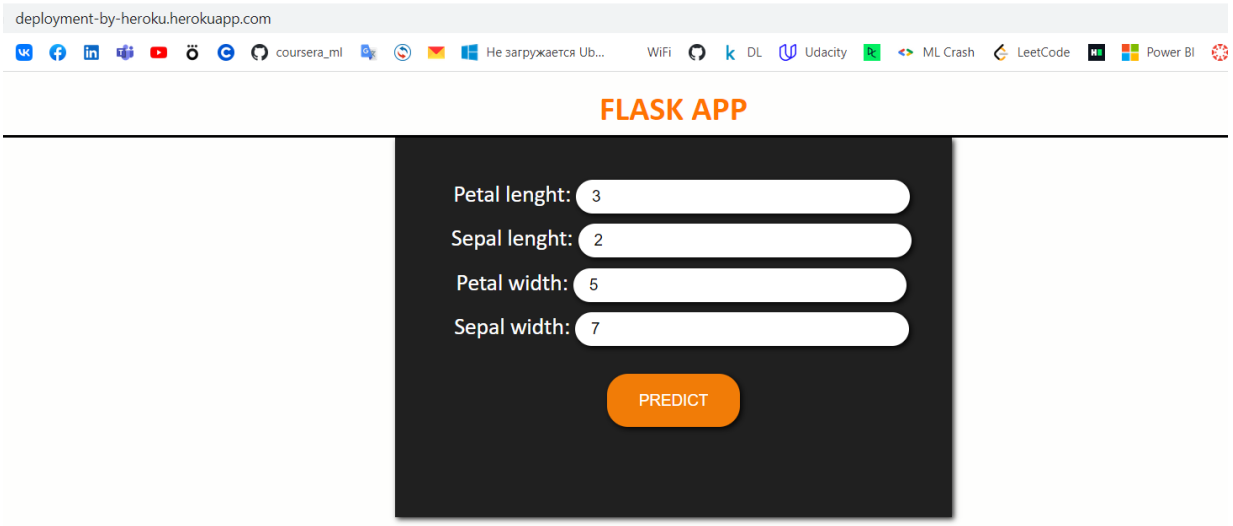
```
      Installing collected packages: pytz, Werkzeug, urllib3, threadpoolctl, six,
pyparsing, pillow, numpy, MarkupSafe, kiwisolver, joblib, itsdangerous, idna,
gunicorn, fonttools, cycler, click, chardet, certifi, scipy, requests,
python-dateutil, packaging, Jinja2, scikit-learn, pandas, matplotlib, Flask
      Successfully installed Flask-1.1.1 Jinja2-2.10.1 MarkupSafe-1.1.1
Werkzeug-0.15.5 certifi-2022.6.15 chardet-4.0.0 click-8.1.3 cycler-0.11.0
fonttools-4.34.4 gunicorn-19.9.0 idna-2.10 itsdangerous-1.1.0 joblib-1.1.0
kiwisolver-1.4.4 matplotlib-3.5.2 numpy-1.23.1 packaging-21.3 pandas-1.4.3
pillow-9.2.0 pyparsing-3.0.9 python-dateutil-2.8.2 pytz-2022.1 requests-2.25.1
scikit-learn-1.1.1 scipy-1.9.0 six-1.16.0 threadpoolctl-3.1.0 urllib3-1.26.11
-----> Discovering process types
      Procfile declares types -> web
-----> Compressing...
      Done: 185.1M
-----> Launching...
      Released v12
      https://deployment-by-heroku.herokuapp.com/ deployed to Heroku
This app is using the Heroku-20 stack, however a newer stack is available.
To upgrade to Heroku-22, see:
https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
```

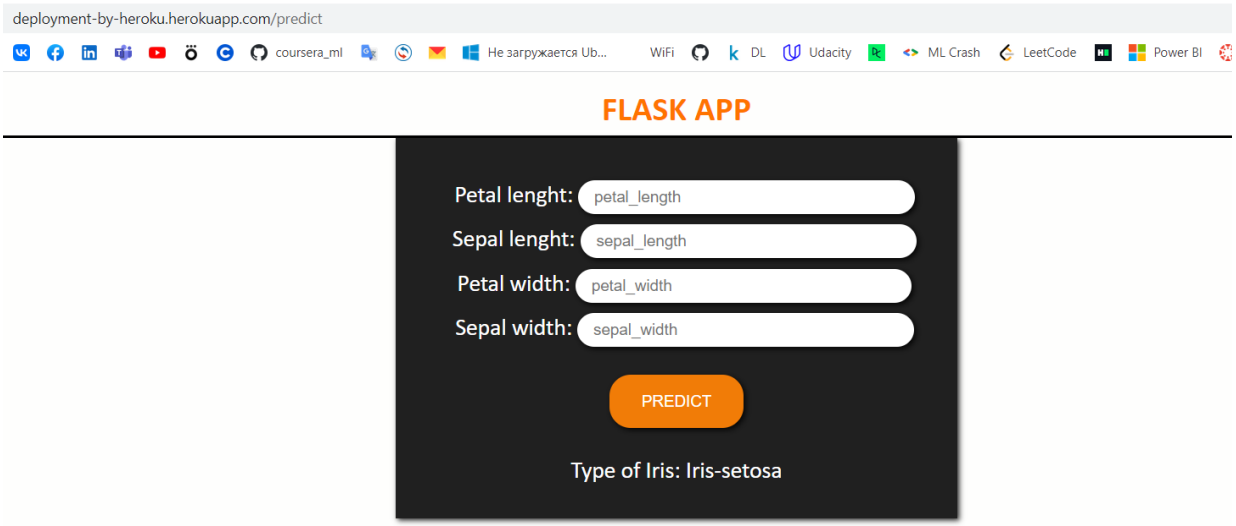
Deployment to Heroku:



9. Testing the Web App







Link to final app: https://deployment-by-heroku.herokuapp.com/