

Name: Leon Emmanuel ISHIMWE

Email: [ishimweleon@gmail.com](mailto:ishimweleon@gmail.com)

Batch code: lisum11

Submission date: 2 sept 2022

Submitted to: github

## Introduction

Using a toy dataset (Iris dataset) from sklearn, we will deploy a model using flask on our local machine. Our model has 4 feature values: "sepal length (cm)", "sepal width (cm)", "petal length (cm)" and "petal width (cm)". Our target value has 3 classes: 'setosa', 'versicolor' and 'virginica'.

Step 1: Launch jupyter and choose toy dataset as sklearn iris dataset which contains 3 targets: 'setosa', 'versicolor' and 'virginica'.

```
In [17]: import pandas as pd
         from sklearn.datasets import load_iris
         from sklearn.linear_model import LogisticRegression
         import pickle
```

```
In [28]: data = load_iris()
         X = data.data
         Y = data.target
         print(data.target_names)

['setosa' 'versicolor' 'virginica']
```

```
In [25]: lm = LogisticRegression(max_iter=1000)
         lm.fit(X, Y)
         pickle.dump(lm, open("model.pickle", "wb"))
```

```
In [26]: model = pickle.load(open("model.pickle", "rb"))
         model.predict([[1, 3, 1, 2]])
```

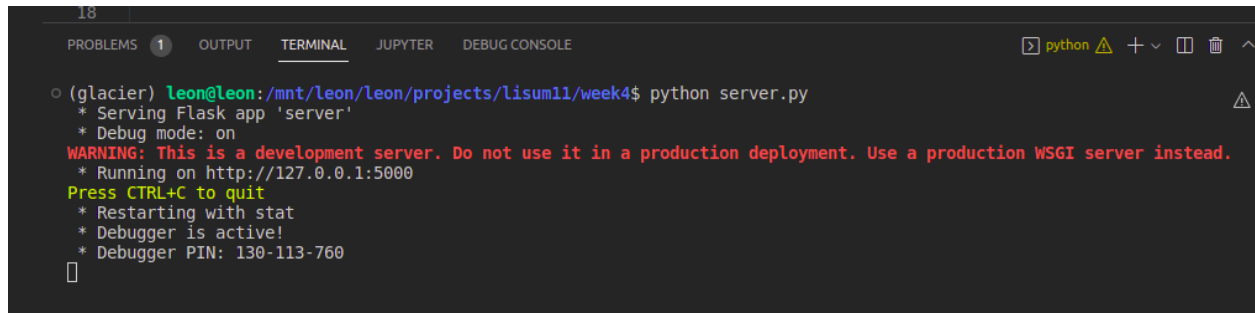
```
Out[26]: array([0])
```

```
In [ ]:
```

## Step 2: Open vs code and add flask codes

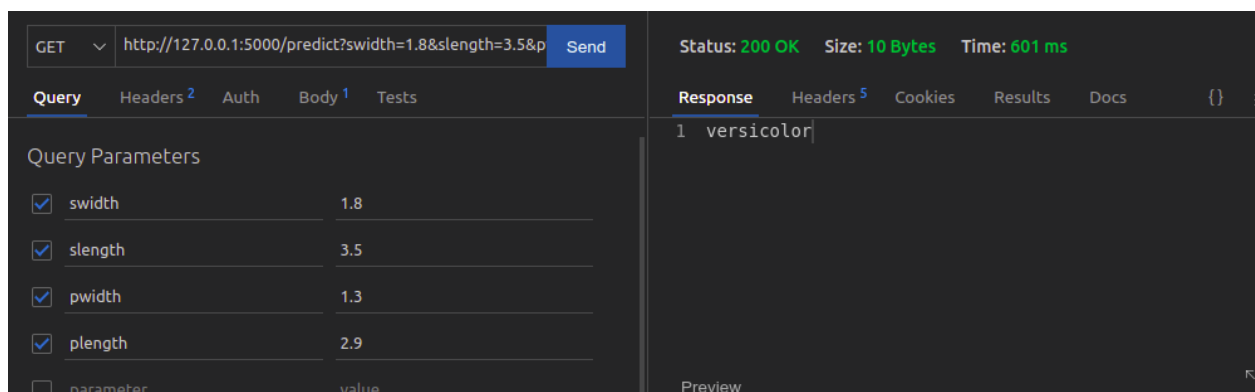
```
server.py 1, U ×  TC Release Notes  TC New Request
server.py > predict
1  from flask import Flask, request, jsonify
2  import pickle
3
4
5  targets = ["setosa", "versicolor", "virginica"]
6
7
8  app = Flask(__name__)
9
10 @app.route("/", methods = ["GET", "POST"])
11 def status():
12     return jsonify({"status": "online"})
13
14
15 @app.route("/predict", methods = ["GET", "POST"])
16 def predict():
17     model = pickle.load(open("model.pickle", "rb"))
18
19     slength = float(request.args.get("slength"))
20     swidth = float(request.args.get("swidth"))
21
22     plength = float(request.args.get("plength"))
23     pwidth = float(request.args.get("pwidth"))
24
25     res = model.predict([[slength, swidth, plength, pwidth]])
26     return targets[res[0]]
27
28 if __name__ == "__main__":
29     app.run(debug = True)
```

## Step 3: Run codes



```
18
PROBLEMS 1 OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
python
o (glacier) leon@leon:/mnt/leon/leon/projects/lisum11/week4$ python server.py
  * Serving Flask app 'server'
  * Debug mode: on
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on http://127.0.0.1:5000
  Press CTRL+C to quit
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 130-113-760
[]
```

## Step 4: Test codes using Thunderclient



GET  Send

Status: 200 OK Size: 10 Bytes Time: 601 ms

Query Headers 2 Auth Body 1 Tests

Response Headers 5 Cookies Results Docs {}

Query Parameters

parameter	value
<input checked="" type="checkbox"/> swidth	1.8
<input checked="" type="checkbox"/> slength	3.5
<input checked="" type="checkbox"/> pwidth	1.3
<input checked="" type="checkbox"/> plength	2.9
<input type="checkbox"/> parameter	value

1 versicolor

Preview

## Conclusion

A toy model was successfully deployed on my localhost in flask, and we tested it by providing 4 feature values, and we got “versicolor” as predicted class.