

Name: Benjamin Prud'homme

Batch: LISUM12

Date: August 27, 2022

Submitted to: Data Glacier

## Deployment on Flask

### 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'))
```

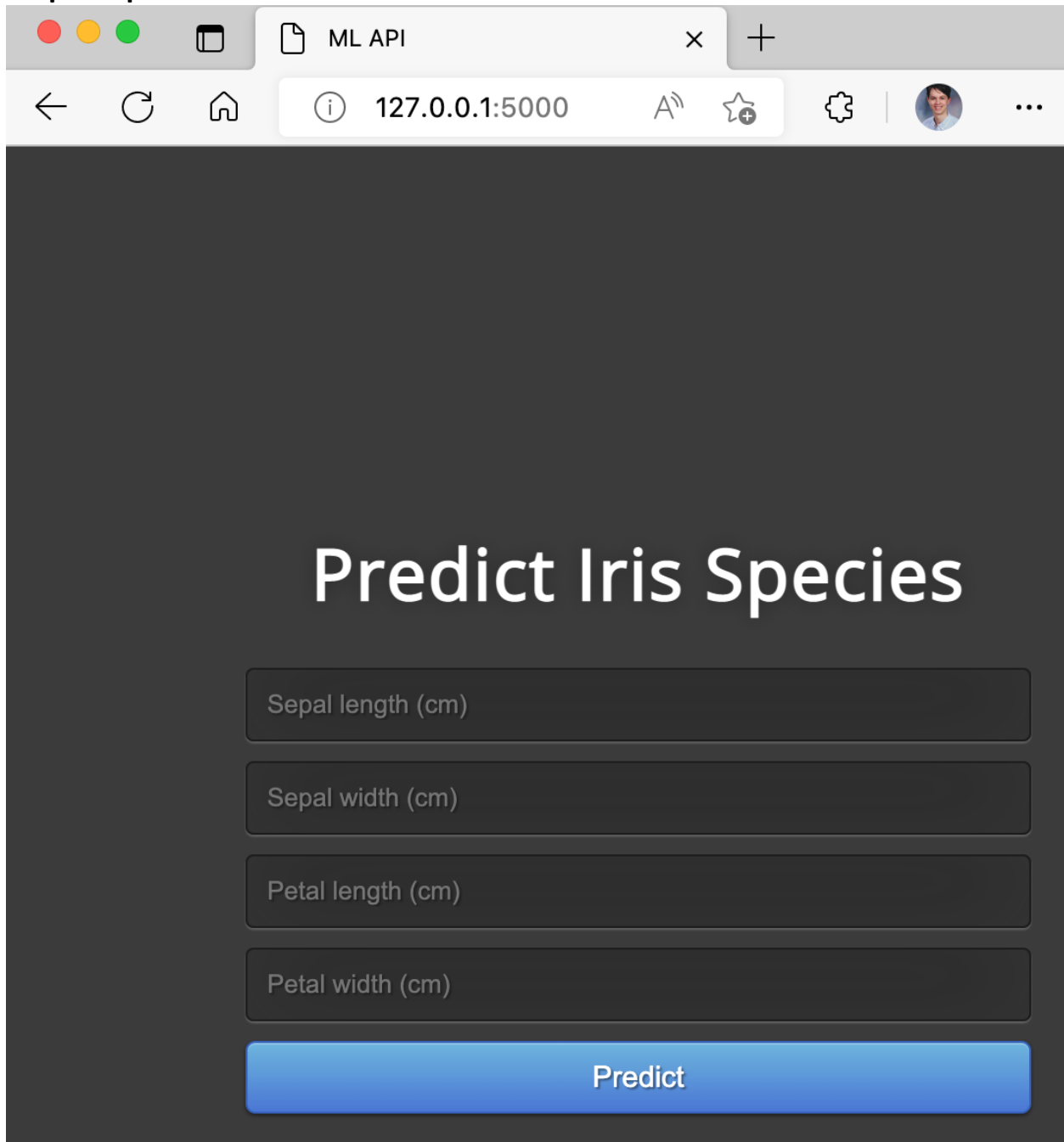
### Step 3: Write Flask app (app.py)

```
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: Run Flask app in terminal

```
[(base) benjaminprudhomme@MacBook-Air Flask-Deployment % set flask_app=app.py ]
[(base) benjaminprudhomme@MacBook-Air Flask-Deployment % flask run ]
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

### Step 5: Open in browser



The screenshot shows a web browser window with a single tab titled 'ML API'. The address bar displays '127.0.0.1:5000'. The main content area has a dark background and features the title 'Predict Iris Species' in large white text. Below the title are four text input fields, each with a light gray placeholder text: 'Sepal length (cm)', 'Sepal width (cm)', 'Petal length (cm)', and 'Petal width (cm)'. At the bottom of the form is a prominent blue button with the text 'Predict' in white.

# Predict Iris Species

Sepal length (cm)

Sepal width (cm)

Petal length (cm)

Petal width (cm)

Predict

## Step 6: Test

Predict Iris Species	Predict Iris Species
<input type="text" value="5"/>	<input type="text" value="Sepal length (cm)"/>
<input type="text" value="3"/>	<input type="text" value="Sepal width (cm)"/>
<input type="text" value="2"/>	<input type="text" value="Petal length (cm)"/>
<input type="text" value="0.1"/>	<input type="text" value="Petal width (cm)"/>
<input type="button" value="Predict"/>	<input type="button" value="Predict"/>
Predicted species: Iris-setosa	