

Name: Jeffrey Joseph
Batchcode: LISUM31
SubmissionDate: 28/03/2024
Task: Flask Deployment

Deployment on Flask

An interface on the web allows us to communicate with the machine-learning model that is used in this assignment.

- Building ML model
- Deployment using Flask

Introduction:

This document provides an overview of a Python codebase developed to deploy a machine learning model using the Flask framework. The machine learning model is trained on the Iris dataset, a popular dataset in the field of machine learning and data science. The Iris dataset is commonly used for classification tasks and consists of measurements of iris flowers' sepal and petal dimensions, along with their corresponding species.

The primary objective of this codebase is to train a machine learning model capable of predicting the species of iris flowers based on their sepal length, sepal width, petal length, and petal width. Once the model is trained, it is deployed as a web application using the Flask framework, allowing users to interactively input the iris flower's measurements and receive predictions on their species in real-time.

This project serves as an example of deploying machine learning models into production environments, enabling seamless integration of predictive analytics into web applications. By leveraging the Flask framework, users can access the predictive capabilities of the trained model through a user-friendly interface, facilitating practical applications of machine learning in various domains.

Snapshot of each step of deployment:

ML model:

```
main.py x iris.py x app.py x index.html x
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.ensemble import RandomForestClassifier
4 from sklearn.model_selection import train_test_split
5 import pickle
6
7 # Load the CSV file
8 df = pd.read_csv("/Users/jef/Downloads/loan predictions/Deployment-of-ML-model-using-Flask-main/Iris.csv")
9
10 # Display the first few rows of the dataframe
11 print(df.head(10))
12
13 # Select the independent and dependent variables
14 X = df[["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"]]
15 y = df["Species"]
16
17 # Split the dataset into train and test sets
18 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=40)
19
20 # Perform feature scaling
21 sc = StandardScaler()
22 X_train = sc.fit_transform(X_train)
23 X_test = sc.transform(X_test)
24
25 # Instantiate the random forest classifier model
26 classifier = RandomForestClassifier()
27
28 # Fit the model to the training data
29 classifier.fit(X_train, y_train)
30
31 # Save the trained model as a pickle file
32 pickle.dump(classifier, open("model.pkl", "wb"))
```

terminal Python Packages Python Console Services

ay 11:40 am) 24:1 LF UTF-8 4 spaces Python 3.10 (pythonProject5)

ML Model Result:

```
iris
/Users/jef/Desktop/PYTHON/pythonProject5/venv/bin/python /Users/jef/Desktop/PYTHON/pythonProject5/iris.py
Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
0 1 5.1 3.5 1.4 0.2 Iris-setosa
1 2 4.9 3.0 1.4 0.2 Iris-setosa
2 3 4.7 3.2 1.3 0.2 Iris-setosa
3 4 4.6 3.1 1.5 0.2 Iris-setosa
4 5 5.0 3.6 1.4 0.2 Iris-setosa
5 6 5.4 3.9 1.7 0.4 Iris-setosa
6 7 4.6 3.4 1.4 0.3 Iris-setosa
7 8 5.0 3.4 1.5 0.2 Iris-setosa
8 9 4.4 2.9 1.4 0.2 Iris-setosa
9 10 4.9 3.1 1.5 0.1 Iris-setosa

Process finished with exit code 0
```

Deploying Flask

```
main.py x iris.py x app.py x index.html x
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 # Create a Flask app
6 app = Flask(__name__)
7
8 # Load the model for making predictions
9 model = pickle.load(open("model.pkl", "rb"))
10
11
12 @app.route("/")
13 def home():
14     # Render the index.html template
15     return render_template("index.html")
16
17
18 @app.route("/predict", methods=["POST"])
19 def predict():
20     # Retrieve the input values from the form
21     float_features = [float(x) for x in request.form.values()]
22     features = [np.array(float_features)]
23
24     # Make predictions using the loaded model
25     prediction = model.predict(features)
26
27     # Render the index.html template with the prediction result
28     return render_template("index.html", prediction_text=f"The flower species is {prediction}")
29
30
31 if __name__ == "__main__":
32     # Run the Flask app
33     app.run(debug=True, port=5000)
34
```

Output:

```
app
/Users/jef/Desktop/PYTHON/pythonProject5/venv/bin/python /Users/jef/Desktop/PYTHON/pythonProject5/app.py
* Serving Flask app 'app'
* 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: 133-730-767
```

HTML (index.html)

```
<body>
  <div class="container">
    <h1>Flower Species Prediction</h1>
    <form action="/predict" method="post">
      <label for="sepal_length">Sepal Length (cm):</label>
      <input type="number" id="sepal_length" name="sepal_length" step="0.1" required><br>

      <label for="sepal_width">Sepal Width (cm):</label>
      <input type="number" id="sepal_width" name="sepal_width" step="0.1" required><br>

      <label for="petal_length">Petal Length (cm):</label>
      <input type="number" id="petal_length" name="petal_length" step="0.1" required><br>

      <label for="petal_width">Petal Width (cm):</label>
      <input type="number" id="petal_width" name="petal_width" step="0.1" required><br>

      <button type="submit">Predict Species</button>
    </form>
    {% if prediction_text %}
    <div class="prediction">
      <h2>{{ prediction_text }}</h2>
    </div>
    {% endif %}
  </div>
</body>
</html>
```

Browser Output:

Flower Species Prediction

Sepal Length (cm):

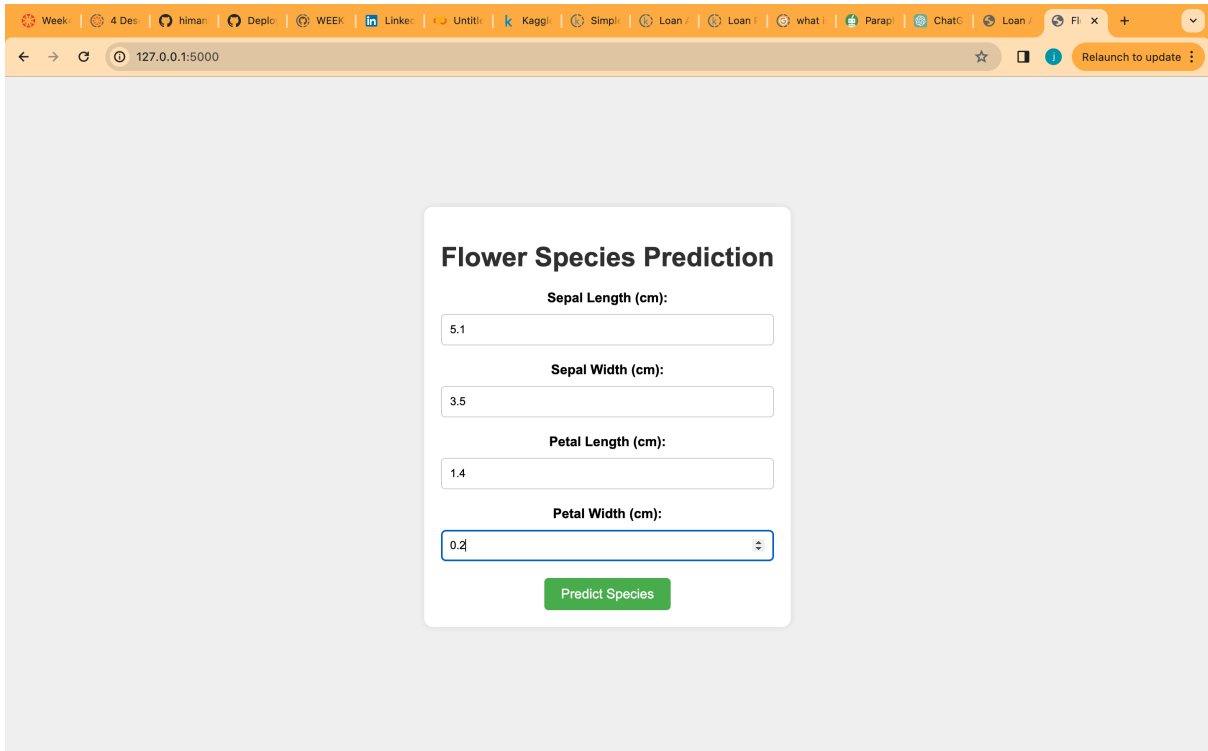
Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

Predict Species

Check the Model:



Flower Species Prediction

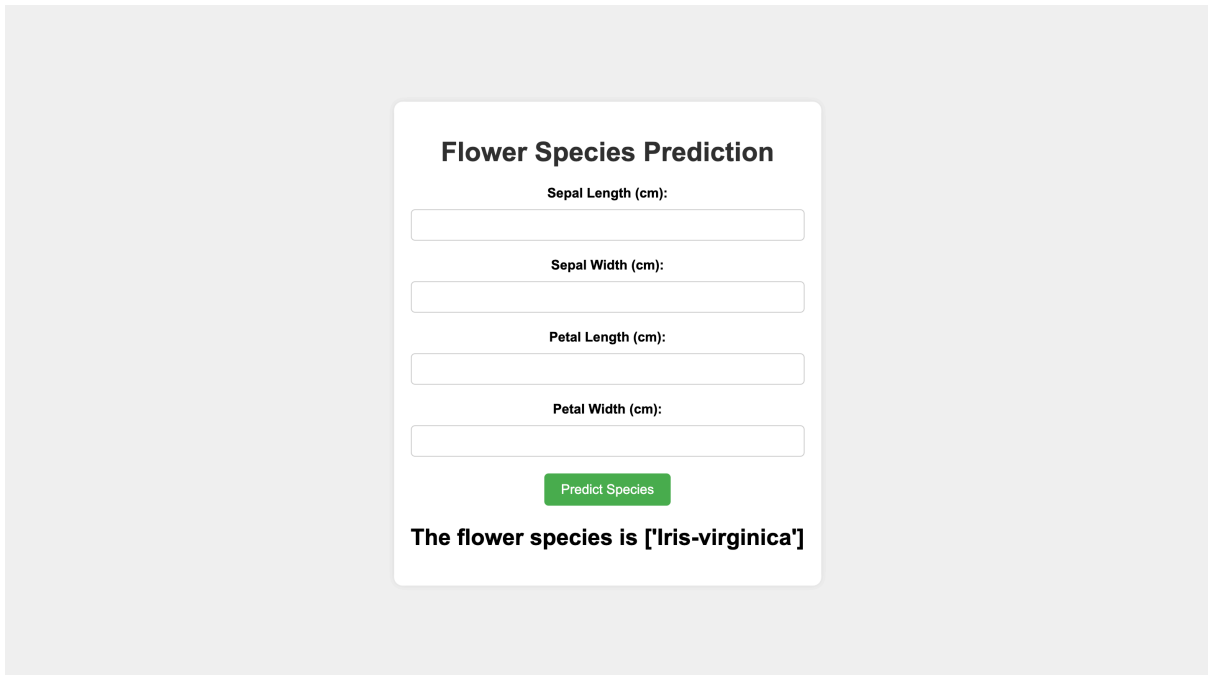
Sepal Length (cm):

Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

Prediction:



Flower Species Prediction

Sepal Length (cm):

Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

The flower species is ['Iris-virginica']