

# Homework Document

1. **Name:** Krishna Ratna Deepika Haripuram
2. **Batch code:** LISUM26
3. **Submission date:** 10-25-2023
4. **Submitted to:** Data Glacier

## Snapshots of Each Step of Deployment

### Writing Flask Application

```
Command Prompt - flask run
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\12267>cd desktop
C:\Users\12267\Desktop>cd flask_app
C:\Users\12267\Desktop\flask_app>flask run
 * Debug mode: off
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
127.0.0.1 - - [25/Oct/2023 09:15:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Oct/2023 09:15:53] "GET /favicon.ico HTTP/1.1" 404 -

C:\Users\12267\Desktop\flask_app>flask run
 * Debug mode: off
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
127.0.0.1 - - [25/Oct/2023 09:22:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Oct/2023 09:22:53] "GET /static/style.css HTTP/1.1" 404 -
[2023-10-25 09:24:05,890] ERROR in app: Exception on /predict [POST]
Traceback (most recent call last):
  File "C:\Users\12267\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 1455, in wsgi_app
    response = self.full_dispatch_request()
               ^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\12267\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 869, in full_dispatch_request
    rv = self.handle_user_exception(e)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
Command Prompt - flask run
  File "C:\Users\12267\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 869, in full_dispatch_request
    rv = self.handle_user_exception(e)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\12267\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 867, in full_dispatch_request
    rv = self.dispatch_request()
         ^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\12267\AppData\Local\Programs\Python\Python311\Lib\site-packages\flask\app.py", line 852, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\12267\Desktop\flask_app\app.py", line 18, in predict
    sepal_length = float(request.form['sepal_length'])
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: could not convert string to float: '5.1 cm'
127.0.0.1 - - [25/Oct/2023 09:24:05] "POST /predict HTTP/1.1" 500 -

C:\Users\12267\Desktop\flask_app>flask run
 * Debug mode: off
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
127.0.0.1 - - [25/Oct/2023 09:27:35] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Oct/2023 09:27:35] "GET /static/style.css HTTP/1.1" 404 -
127.0.0.1 - - [25/Oct/2023 09:28:52] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Oct/2023 09:28:52] "GET /static/style.css HTTP/1.1" 404 -
127.0.0.1 - - [25/Oct/2023 09:29:56] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Oct/2023 09:29:56] "GET /static/style.css HTTP/1.1" 404 -
```

### Showing Folders Containing Information

flask_app >					
	Name	Date modified	Type	Size	
	__pycache__	2023-10-25 9:27 AM	File folder		
	static	2023-10-25 9:21 AM	File folder		
	templates	2023-10-24 6:07 PM	File folder		
	app.py	2023-10-25 9:26 AM	PY File	2 KB	
	iris_model.joblib	2023-10-24 5:54 PM	JOBLIB File	1 KB	

## Showing app.py Code

```

import matplotlib.pyplot as plt
index.html
index.html
app.py
static.css
app.py

File Edit View

from flask import Flask, render_template, request
import numpy as np
import joblib
import re

app = Flask(__name__)

# Load the trained model
model = joblib.load('iris_model.joblib')

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

@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        # Get input values from the form
        sepal_length = float(re.sub('[^0-9.]+', '', request.form['sepal_length']))
        sepal_width = float(re.sub('[^0-9.]+', '', request.form['sepal_width']))
        petal_length = float(re.sub('[^0-9.]+', '', request.form['petal_length']))
        petal_width = float(re.sub('[^0-9.]+', '', request.form['petal_width']))

        # Create a NumPy array with the input values
        data = np.array([[sepal_length, sepal_width, petal_length, petal_width]])

        # Make the prediction
        prediction = model.predict(data)

        # Map prediction to Iris species names
        species_names = ['Setosa', 'Versicolor', 'Virginica']
        predicted_species = species_names[prediction[0]]

        return render_template('index.html', prediction_text=f'The predicted Iris species is: {predicted_species}')

    return render_template('index.html')

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

```

## Showing index.html Code

```
import matplotlib.pyplot as plt
index.html
index.html
app.py

File Edit View

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Iris Species Predictor</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <h1>Iris Species Predictor</h1>
  <form method="post" action="/predict">
    <label for="sepal_length">Sepal Length:</label>
    <input type="text" id="sepal_length" name="sepal_length" required>
    <br>
    <label for="sepal_width">Sepal Width:</label>
    <input type="text" id="sepal_width" name="sepal_width" required>
    <br>
    <label for="petal_length">Petal Length:</label>
    <input type="text" id="petal_length" name="petal_length" required>
    <br>
    <label for="petal_width">Petal Width:</label>
    <input type="text" id="petal_width" name="petal_width" required>
    <br>
    <button type="submit">Predict</button>
  </form>
  {% if prediction_text %}
  <h2>{{ prediction_text }}</h2>
  {% endif %}
</body>
</html>
```

## Showing CSS Code

```
import matplotlib.pyplot as plt
index.html
index.html
app.py
static.css
app.py
app.py

File Edit View

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

h1 {
  color: #333;
}

form {
  max-width: 400px;
  margin: 0 auto;
  padding: 20px;
  background-color: #fff;
  border: 1px solid #e1e1e1;
  border-radius: 5px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

label {
  font-size: 18px;
  margin-bottom: 10px;
  display: block;
}

input[type="text"] {
  width: calc(100% - 22px);
  padding: 10px;
  margin-bottom: 20px;
  border: 1px solid #e1e1e1;
  border-radius: 5px;
}

button {
  width: 100%;
  padding: 10px;
  background-color: #5cb85c;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #4cae4c;
}
```

## Opened the Link in Browser

# Iris Species Predictor

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:



## Testing the Model



### Iris Species Predictor

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

The predicted Iris species is: Versicolor



### Iris Species Predictor

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

The predicted Iris species is: Setosa