



Data Science Intern at Data Glacier

Week 4: Deployment on Flask

Name: Mahyar Arani

Batch code: LISUM19

Submission date: March 28, 2023

Submitted to: Data Glacier

Introduction:

In this project assignment, we will showcase how to deploy a trained machine learning model using Flask on the Heroku cloud app services. The model we will be using is a Random Forest Regressor, which was trained to predict the shipping cost estimate of sculptures based on features such as the price of sculpture, artist reputation, base shipping price, weight, and dimensions.

Deploying a machine learning model on a cloud platform like Heroku allows for easy accessibility and use of the model by others. The workflow for deploying the model involves creating a Flask web application that takes in user inputs and generates predictions using the trained model. Figure 1 below provides an overview of the workflow and the different components involved in the deployment process.

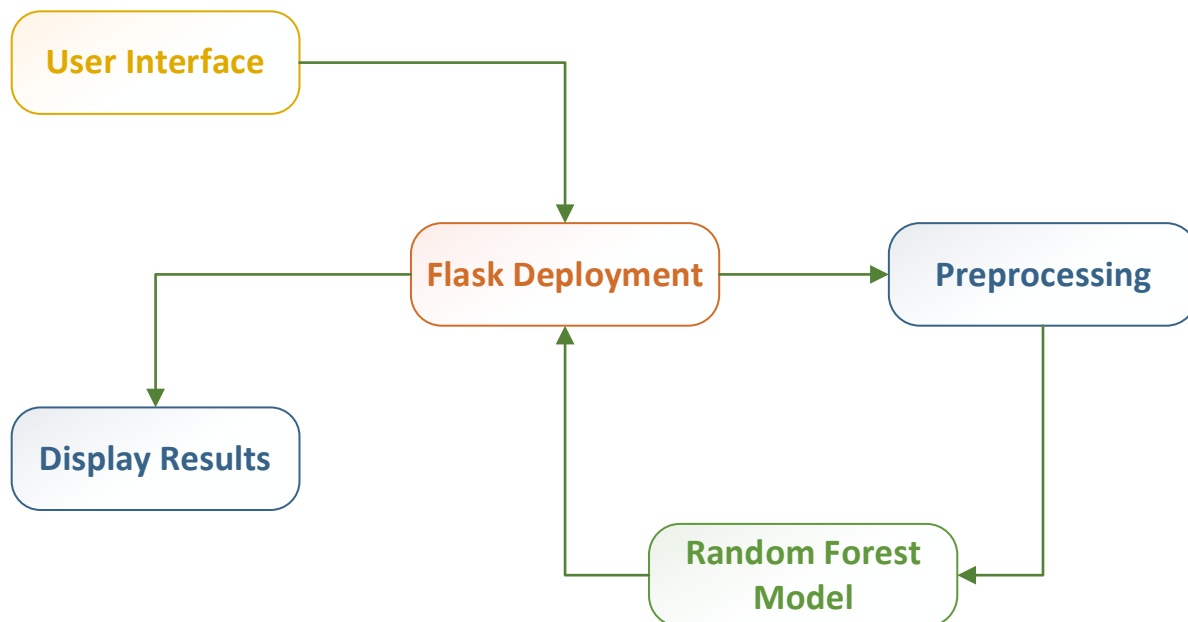
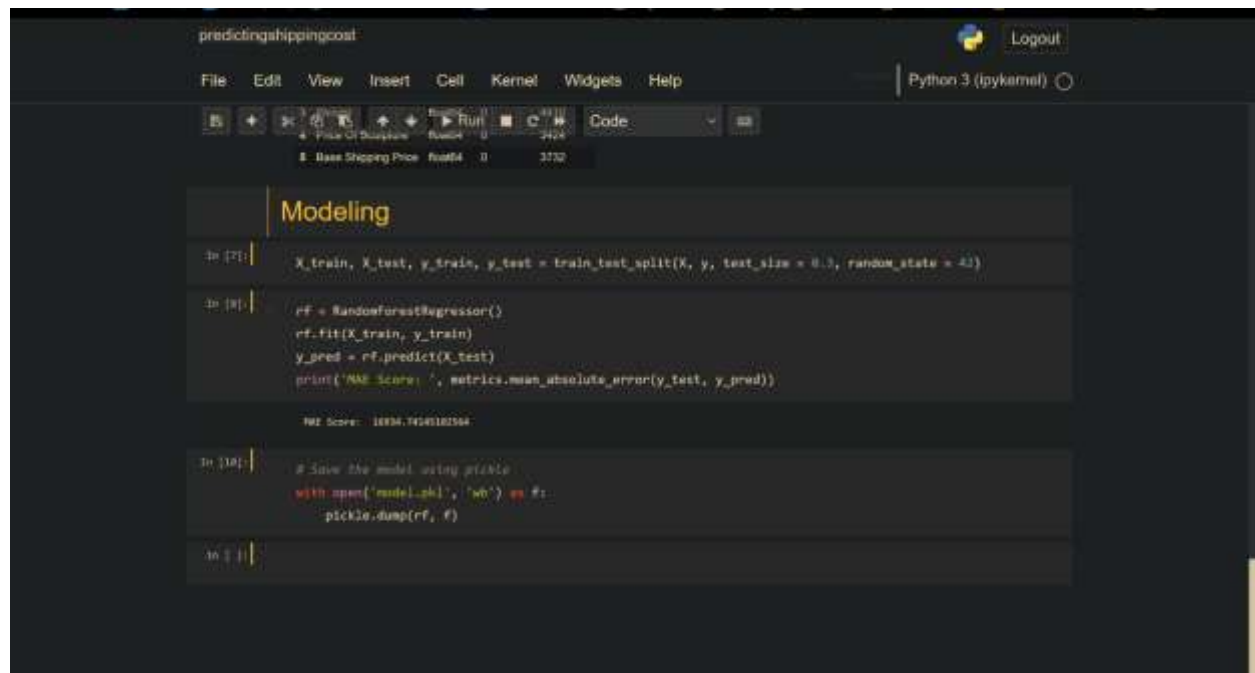


Figure 1: Deployment Workflow

The snapshots of the deployment are as follows:



The screenshot shows a Jupyter Notebook interface with the title 'predictingshippingcost'. The top bar includes a 'Logout' button and the text 'Python 3 (ipykernel)'. The notebook has a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with various icons for cell operations. The notebook content is divided into sections, with the current section titled 'Modeling'. It contains four code cells:

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

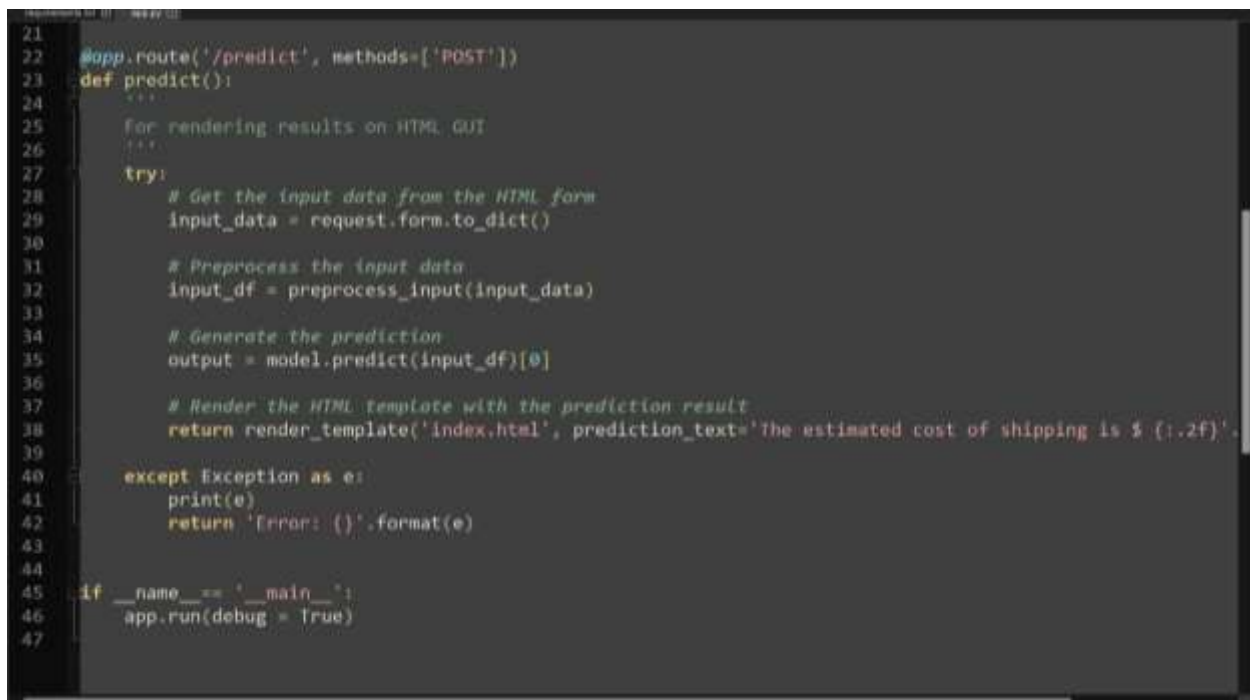
```
In [8]: rf = RandomForestRegressor()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
print('MAE Score: ', metrics.mean_absolute_error(y_test, y_pred))
```

MAE Score: 16834.74545182564

```
In [10]: # Save the model using pickle
with open('model.pkl', 'wb') as f:
    pickle.dump(rf, f)
```

```
In [1]:
```

Figure 3: Training and Saving the Model Via Pickle



The screenshot shows a Python script for Flask deployment. The code is as follows:

```
21
22 @app.route('/predict', methods=['POST'])
23 def predict():
24     """
25     for rendering results on HTML GUI
26     """
27     try:
28         # Get the input data from the HTML form
29         input_data = request.form.to_dict()
30
31         # Preprocess the input data
32         input_df = preprocess_input(input_data)
33
34         # Generate the prediction
35         output = model.predict(input_df)[0]
36
37         # Render the HTML template with the prediction result
38         return render_template('index.html', prediction_text='The estimated cost of shipping is $ {:.2f}').
39
40     except Exception as e:
41         print(e)
42         return 'Error: {}'.format(e)
43
44
45 if __name__ == '__main__':
46     app.run(debug = True)
47
```

Figure 2: Flask Deployment

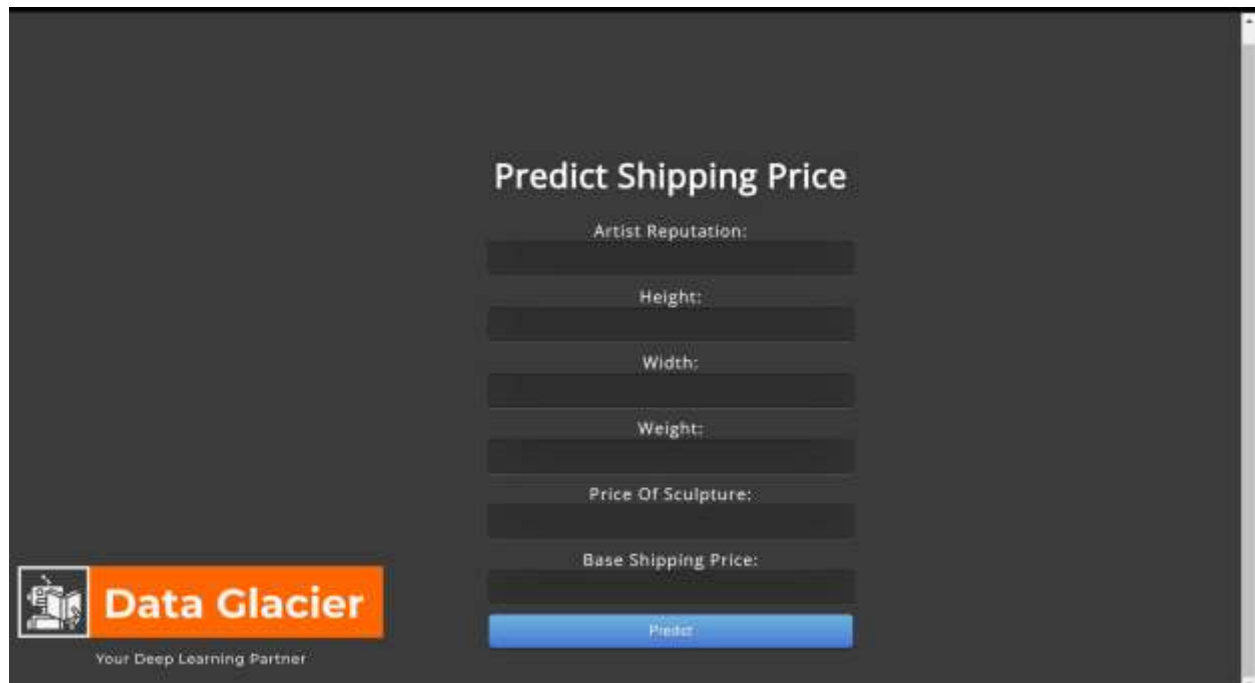
```
C:\Users\saahya\OneDrive\Career Improvement\Data Glacier Internship\Week 4 - Flask Deployment\AA Web Application using Flask\templates\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window Help
requirements.txt app.py index.html
16 <h1>Predict Shipping Price</h1>
17
18 <!-- Main Input For Receiving Query to our ML -->
19 <form action="{{ url_for('predict')}}" method="post">
20   <label for="Artist Reputation">Artist Reputation:</label>
21   <input type="number" name="Artist Reputation" step="0.01" required="required" /><br>
22
23   <label for="Height">Height:</label>
24   <input type="number" name="Height" required="required" /><br>
25
26   <label for="Width">Width:</label>
27   <input type="number" name="Width" required="required" /><br>
28
29   <label for="Weight">Weight:</label>
30   <input type="number" name="Weight" required="required" /><br>
31
32   <label for="Price Of Sculpture">Price Of Sculpture:</label>
33   <input type="number" name="Price Of Sculpture" step="0.01" required="required" /><br>
34
35   <label for="Base Shipping Price">Base Shipping Price:</label>
36   <input type="number" name="Base Shipping Price" step="0.01" required="required" /><br>
37
38   <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
39 </form>
40
41 <br>
42 <br>
43 {{ prediction text }}
44
Hyper Text Markup Language file length: 1804 - lines: 51 ln: 1 col: 1 pos: 1 utf-8 UTF-8 PD
```

Figure 3: Designing the Fields on HTML File

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saahya\OneDrive\Career Improvement\Data Glacier Internship\git-repos\Horoku-impl>Python 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: 887-856-549
```

Figure 4: Running the Local REST API



The image shows a web application interface titled "Predict Shipping Price". It features a dark gray background with white text. On the left, there is a logo for "Data Glacier" with the tagline "Your Deep Learning Partner". The main area contains seven input fields, each with a label above it: "Artist Reputation:", "Height:", "Width:", "Weight:", "Price Of Sculpture:", and "Base Shipping Price:". Below these fields is a blue button labeled "Predict".

Predict Shipping Price

Artist Reputation:

Height:

Width:

Weight:

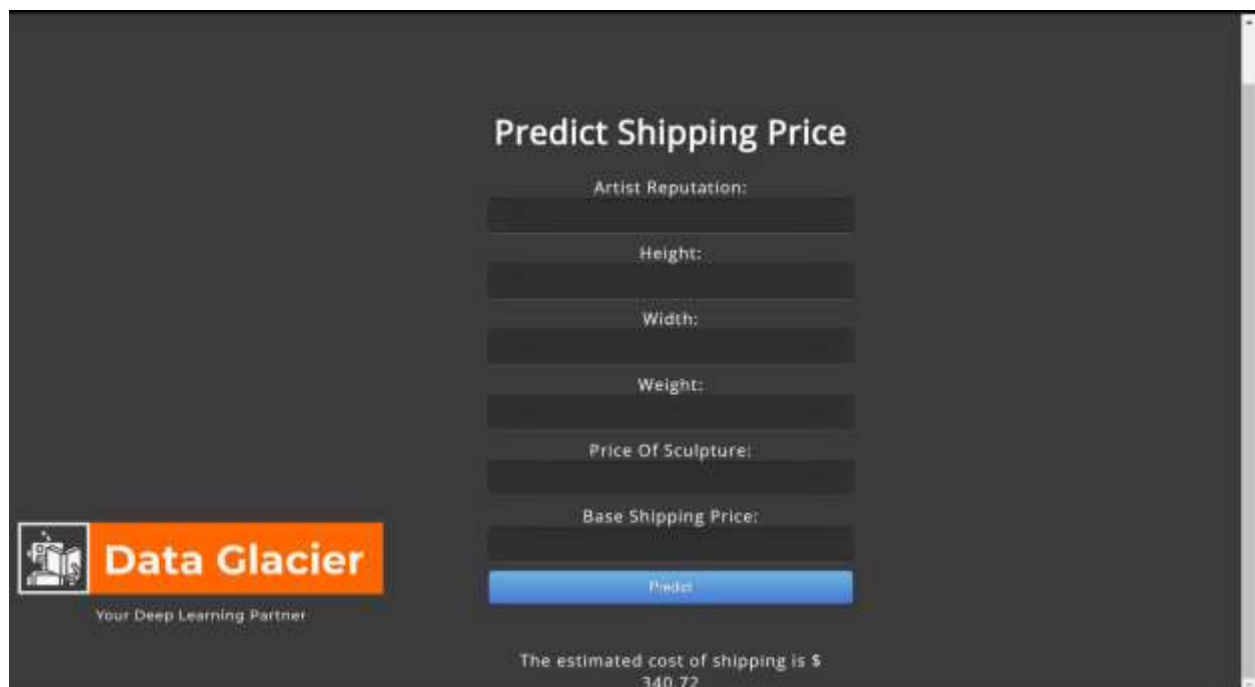
Price Of Sculpture:

Base Shipping Price:

Predict

Data Glacier
Your Deep Learning Partner

Figure 5: Model User Interface



The image shows the same web application interface as Figure 5, but with the "Predict" button highlighted in blue. Below the input fields, the text "The estimated cost of shipping is \$ 340.72" is displayed.

Predict Shipping Price

Artist Reputation:

Height:

Width:

Weight:

Price Of Sculpture:

Base Shipping Price:

Predict

Data Glacier
Your Deep Learning Partner

The estimated cost of shipping is \$ 340.72

Figure 6: The Result of a Sample Prediction

