



Data Glacier

Your Deep Learning Partner

Week #4 Deliverables

Name: Model Deployment on Flask

Report date: 02/28/25

Internship Batch: LISUM41

Version: 1.0

Data intake by: Sophonie Sidrac

Data intake reviewer: Sophonie Sidrac

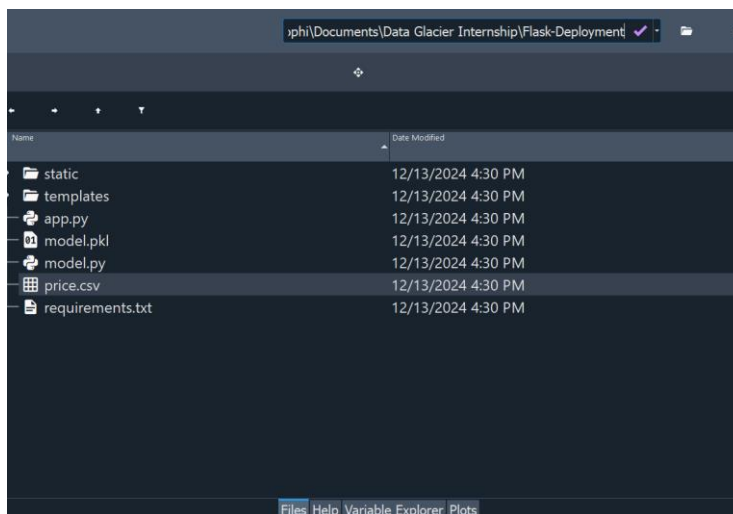
Data storage location: GitHub

Github Repo link:

<https://github.com/1Sophani/DataGlacier-Internship/tree/main/Week%204>

Steps for deployment:

1.



2.

```
untitled2.py* X model.py X style.css X app.py X index.html X
1  # -*- coding: utf-8 -*-
2  """
3  Created on Fri Dec 13 19:55:51 2024
4
5  @author: sophi
6  """
7
8  from flask import Flask
9
10 app = Flask(__name__)
11
12 @app.route('/') #http://www.google.com/
13 def home():
14     return "Hello, world!"
15
16 app.run(port=5000)
```

3.

```
untitled2.py* X model.py X style.css X app.py X index.html X
1  # Importing the libraries
2  import numpy as np
3  import pandas as pd
4  import pickle
5
6  dataset = pd.read_csv('price.csv')
7
8  dataset['bed_room'].fillna(0, inplace=True)
9
10 dataset['area'].fillna(dataset['area'].mean(), inplace=True)
11
12 X = dataset.iloc[:, :3]
13
14 #Converting words to integer values
15 def convert_to_int(word):
16     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
17                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
18     return word_dict[word]
19
20 X['bed_room'] = X['bed_room'].apply(lambda x : convert_to_int(x))
21
22 y = dataset.iloc[:, -1]
23
24 from sklearn.linear_model import LinearRegression
25 regressor = LinearRegression()
26
27 #Fitting model with trainig data
28 regressor.fit(X, y)
29
30 # Saving model to disk
31 pickle.dump(regressor, open('model.pkl','wb'))
32
33 # loading model to compare the results
34 model = pickle.load(open('model.pkl','rb'))
35 print(model.predict([[2, 2200, 5]]))
```

4.

```

1 @import url(https://fonts.googleapis.com/css?family=Open+Sans);
2 .btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; f
3 .btn:hover, .btn.active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
4 .btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -m
5 .btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-position:
6 .btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
7 .btn-primary.active { color: rgba(255, 255, 255, 0.75); }
8 .btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top, #6eb6de, #4a77
9 .btn-primary:hover, .btn-primary.active, .btn-primary.disabled, .btn-primary[di
10 .btn-block { width: 100%; display: block; }
11
12 * { -webkit-box-sizing: border-box; -moz-box-sizing: border-box; -ms-box-sizing: border-box; -o-box-siz
13
14 html { width: 100%; height: 100%; overflow: hidden; }
15
16 body {
17     width: 100%;
18     height: 100%;
19     font-family: 'Open Sans', sans-serif;
20     color: #fff;
21     font-size: 18px;
22     text-align: center;
23     letter-spacing: 1.2px;
24     background: #3B3B3B !important;
25     filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D', endColorstr='#09275
26 }
27
28 .login {
29     position: absolute;
30     top: 40%;
31     left: 50%;
32     margin: -150px 0 0 -150px;
33     width: 400px;
34     height: 400px;
35 }
36
37 .login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing: 1px; text-align: cente
38
39 input {
40     width: 100%;
41     margin-bottom: 10px;
42     background: rgba(0,0,0,0.3);
43     border: none;

```

5.

```

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     int_features = [int(x) for x in request.form.values()]
18     final_features = np.array(int_features)
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='House price should be {}'.format(output))
24
25 if __name__ == "__main__":
26     app.run(debug=True)

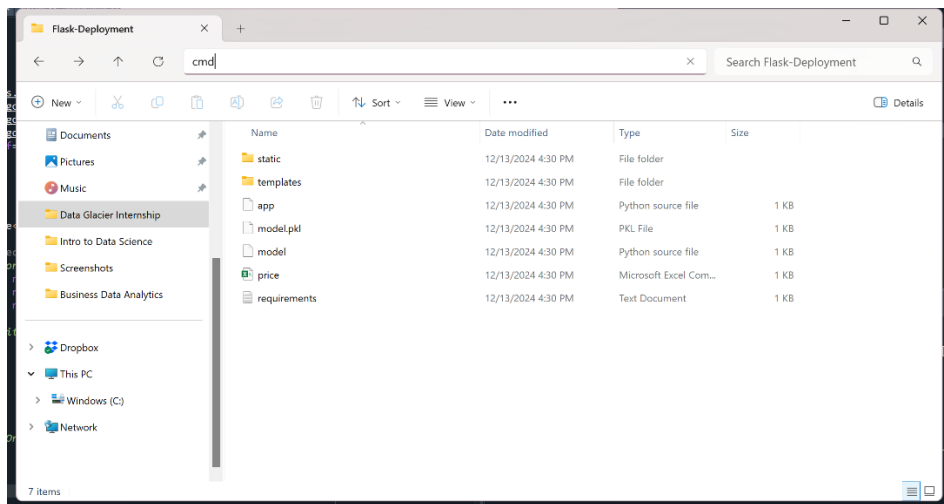
```

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>ML API</title>
6   <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css">
7   <link href="https://fonts.googleapis.com/css?family=Arimo" rel="stylesheet" type="text/css">
8   <link href="https://fonts.googleapis.com/css?family=Hind:300" rel="stylesheet" type="text/css">
9   <link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300" rel="stylesheet" type="text/css">
10  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15   <div class="Login">
16     <h1>Predict House Price</h1>
17
18     <!-- Main Input For Receiving Query to our ML -->
19     <form action="{{ url_for('predict') }}" method="post">
20       <input type="text" name="no_of_rooms" placeholder="Number of Rooms" required="" />
21       <input type="text" name="area" placeholder="Area (in square feet)" required="" />
22       <input type="text" name="house_age" placeholder="House Age" required="" />
23
24       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
25     </form>
26
27     <br>
28     <br>
29     {{ prediction_text }}
30
31   </div>
32   
33
34 </body>
35 </html>
36

```

6.



7.

```

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

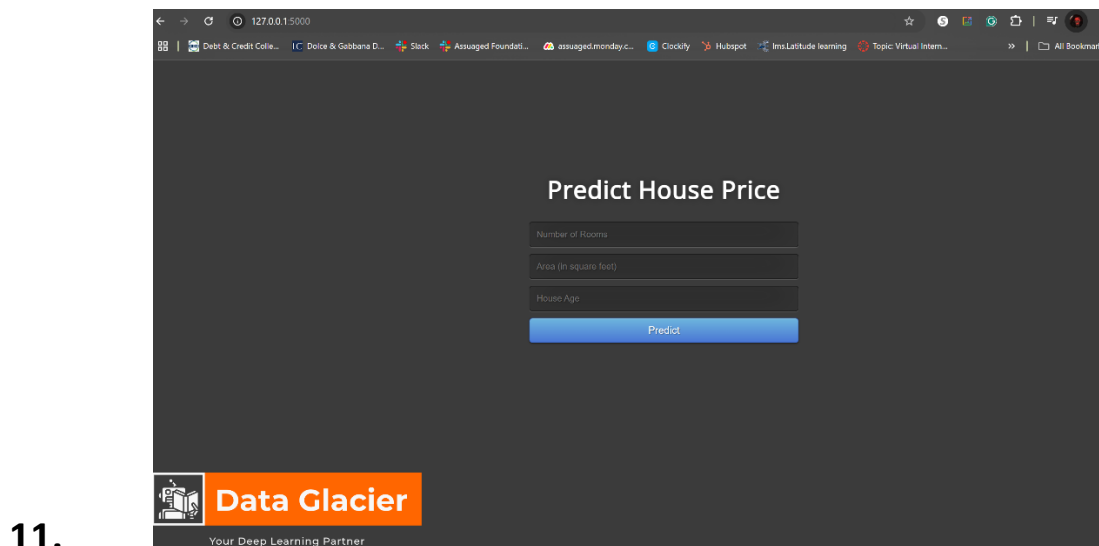
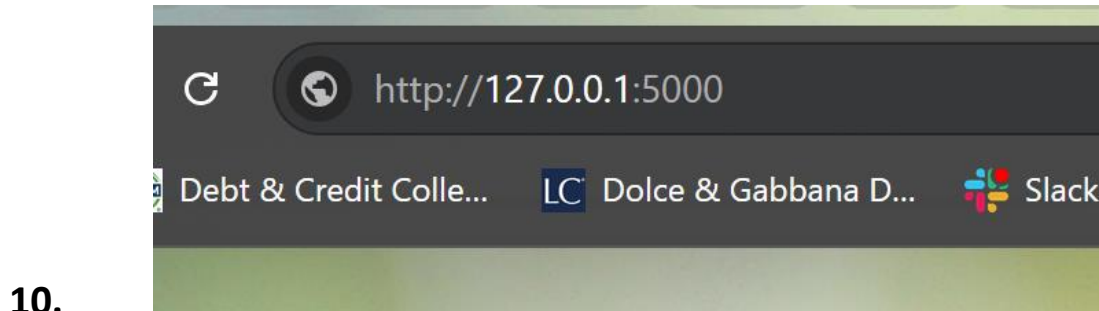
C:\Users\sophi\Documents\Data Glacier Internship\Flask-Deployment>python app.py
C:\Users\sophi\AppData\Local\Programs\Python\Python38\lib\site-packages\sklearn\base.py:348: InconsistentVersionWarning:
Trying to unpickle estimator LinearRegression from version 0.22.1 when using version 1.3.2. This might lead to breaking
code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* 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
C:\Users\sophi\AppData\Local\Programs\Python\Python38\lib\site-packages\sklearn\base.py:348: InconsistentVersionWarning:
Trying to unpickle estimator LinearRegression from version 0.22.1 when using version 1.3.2. This might lead to breaking
code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Debugger is active!
* Debugger PIN: 124-423-278

```

8.

9.

```
* Debug mode: on
WARNING: This is a development server. Do not
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
C:\Users\sophi\AppData\Local\Programs\Python
```



12.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/predict'. The browser's tab bar includes several open tabs: 'Debit & Credit Coffe...', 'Dolce & Gabbana D...', 'Slack', 'Assuaged Foundati...', 'assuaged.monday.c...', 'Clockify', 'Hubspot', 'Ins.Latitude learning', and 'Topic Virtual Intern...'. The main content area of the browser is a dark-themed web application titled 'Predict House Price'. It features three input fields labeled 'Number of Rooms', 'Area (in square feet)', and 'House Age'. Below these fields is a blue 'Predict' button. Underneath the button, the text 'House price should be \$ 95398.36' is displayed. In the bottom left corner of the application, there is a logo for 'Data Glacier' with the tagline 'Your Deep Learning Partner'.

Submitted by: Sophonie Sidrac
Submitted to: Data Glacier
Submission Date: 02/28/25