

## Week4: Deployment on Flask

Name: Madoka Fujii

Batch Code: LISUM23: 30

Submission Date: July 28, 2023

Submitted to: Data Glacier

### Step1: Select any toy data (simple data).

I found a toy data from Kaggle.

<https://www.kaggle.com/datasets/yasserh/housing-prices-dataset>

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

Originally, there are 13 variables.

```
# Separate the dependent variable and independent variables
Y = df['price']

X = df.drop(columns = {'price', 'mainroad', 'guestroom', 'airconditioning', 'prefarea', 'furnishingstatus'})

X = pd.get_dummies(X, drop_first = True)

X.head()
```

	area	bedrooms	bathrooms	stories	parking	basement_yes	hotwaterheating_yes
0	7420	4	2	3	2	0	0
1	8960	4	4	4	3	0	0
2	9960	3	2	2	2	1	0
3	7500	4	2	2	3	1	0
4	7420	4	1	2	2	1	0

But I dropped price and several variables from X. This is because price is dependent variable. And other several variables are dummy variables with Yes or No. In order to fit the flask deploying page, they were dropped. (That is not good idea to have so many variables in web to people fill out.)

I kept 7 independent variables to predict housing price.

## Step2: Save the model

```
import pickle
```

```
# Saving model to disk  
pickle.dump(gradient_reg, open('model.pkl', 'wb'))
```

```
# Loading model to compare the results  
model = pickle.load(open('model.pkl', 'rb'))  
print(model.predict([[8000, 3, 2, 3, 2, 1, 1]]))
```

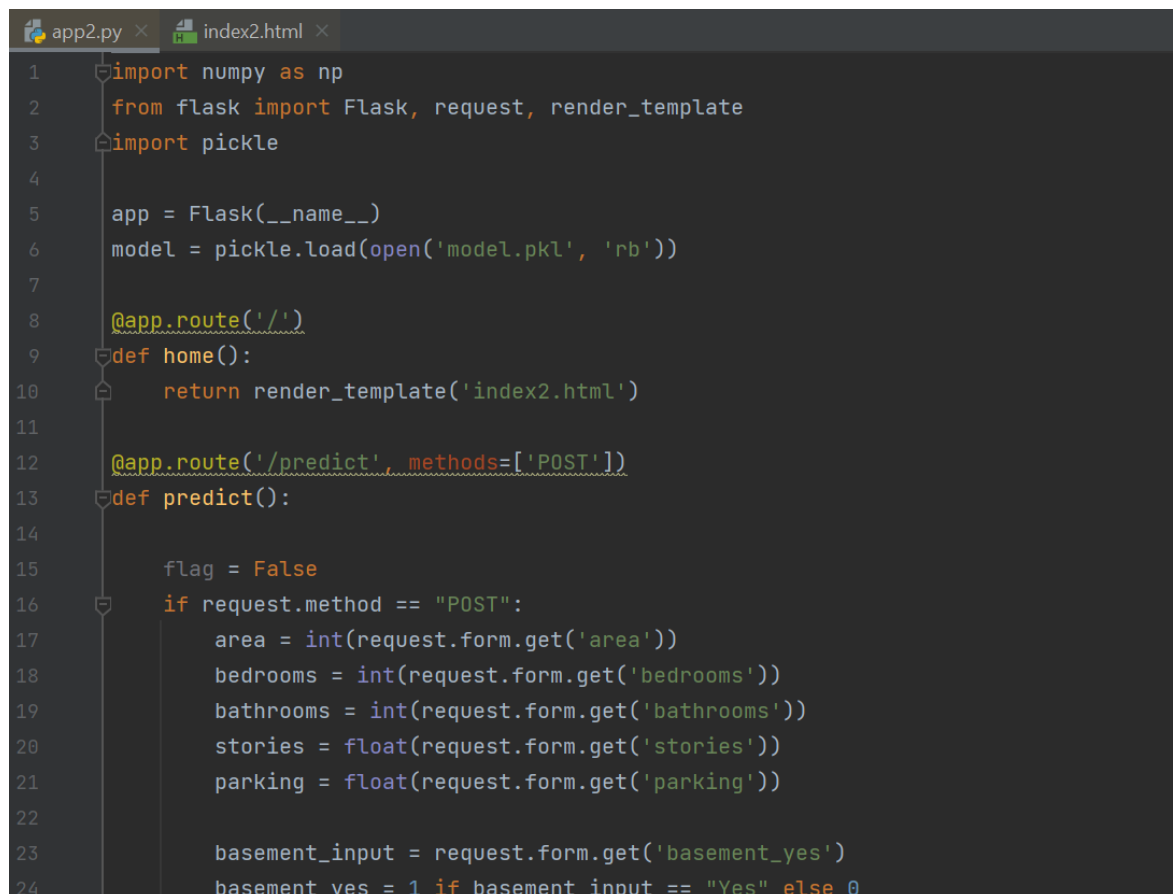
```
[8357433.19312691]
```

Using pickle, saved the model to deploy on Flask.

## Step3. Deploy the model on flask (web app)

In PyCharm, created app2.py file.

Categorical items are Yes or No choice. Set the code if Yes=1 else 0 so that program works.



```
app2.py x index2.html x  
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('index2.html')  
11  
12 @app.route('/predict', methods=['POST'])  
13 def predict():  
14  
15     flag = False  
16     if request.method == "POST":  
17         area = int(request.form.get('area'))  
18         bedrooms = int(request.form.get('bedrooms'))  
19         bathrooms = int(request.form.get('bathrooms'))  
20         stories = float(request.form.get('stories'))  
21         parking = float(request.form.get('parking'))  
22  
23         basement_input = request.form.get('basement_yes')  
24         basement ves = 1 if basement input == "Yes" else 0
```

```

25
26     hotwaterheating_input = request.form.get('hotwaterheating_yes')
27     hotwaterheating_yes = 1 if basement_input == "Yes" else 0
28
29     final_features = [area, bedrooms, bathrooms, stories, parking, basement_yes, hotwaterheating_yes]
30     features = [np.array(final_features)]
31
32     prediction = model.predict(features)
33
34     output = round(prediction[0], 2)
35
36     return render_template('index2.html', flag=True, prediction_text=f'Housing price should be ${output}.')
37
38 if __name__ == "__main__":
39     app.run(debug=True)

```

Created index2.html file.

```

20
21     <label for="bedrooms">Bedrooms:</label>
22     <input type="text" name="bedrooms" id="bedrooms" placeholder="eg:5" required="required" />
23
24     <label for="bathrooms">Bathrooms:</label>
25     <input type="text" name="bathrooms" id="bathrooms" placeholder="eg:3" required="required" />
26
27     <label for="stories">Stories:</label>
28     <input type="text" name="stories" id="stories" placeholder="eg:2" required="required" />
29
30     <label for="parking">Parking:</label>
31     <input type="text" name="parking" id="parking" placeholder="eg:1" required="required" />
32
33     <label for="basement_yes">Basement:</label>
34     <select name="basement_yes" id="basement_yes">
35         <option value="Yes">Yes</option>
36         <option value="No">No</option>
37     </select>
38
39     <label for="hotwaterheating_yes">HotwaterHeating:</label>
40     <select name="hotwaterheating_yes" id="hotwaterheating_yes">
41         <option value="Yes">Yes</option>
42         <option value="No">No</option>
43     </select>

```

Under the templates folder, saved the index2.html file.

Madoka - Personal > Desktop > Data Glacier > Week4\_2 > templates

Name	Status
index2.html	✓

Run app2.py in Command prompt

```

PS C:\Users\Madoka\OneDrive\Desktop\Data Glacier\Week4_2> python app2.py
* Serving Flask app 'app2'
* 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 watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 957-994-852

```

Open the link in new browser.

## Housing Price

Area:

Bedrooms:

Bathrooms:

Stories:

Parking:

Basement:  HotwaterHeating:

Fill in the format.

## Housing Price

Area:

Bedrooms:

Bathrooms:

Stories:

Parking:

Basement:  HotwaterHeating:

Press the predict button. Then, the prediction result appears.

## Housing Price

Area:

Bedrooms:

Bathrooms:

Stories:

Parking:

Basement:  HotwaterHeating:

Housing price should be \$6925008.87.