

Deployment on Flask

- Name : Fawzi El Khatib
- Batch code: LISP01
- Submitted to: Data Glacier Team
- Date: 22-March-2021

1. Dump the model

- The snapshot below the code for dumping (serialization code-pickling) our model from Jupyter Notebook
- I will use a Simple Linear Regression model which predict “house prices” based on their ‘Size’
- Our model is called ‘reg’ and the *size* feature is in feet square

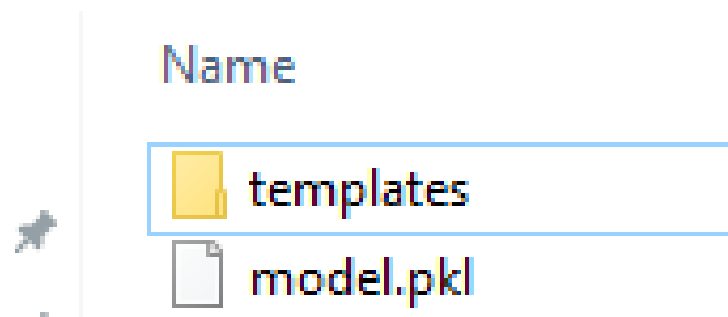
```
import pickle
```

```
with open('model.pkl','wb') as f:  
    pickle.dump(reg,f)
```

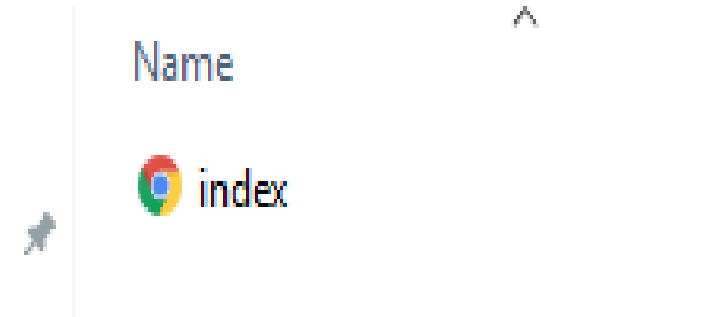
2. Create index.html file

- Create folder called “templates” in the same directory which contains our pickled model
- Create *index.html* file in templates folder that is compatible with the inputs of our model

Flask Deployment



Flask Deployment > templates



Index.html code

```
1 <!DOCTYPE html>
2 <html >
3 <body>
4 <div class="login">
5     <h1>House Price Prediction</h1>
6     <form action="{{ url_for('predict')}}" method="POST">
7         <input type="text" name="size" required="required" />
8
9         <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
10    </form>
11
12    <br>
13    <br>
14    {{ prediction_text }}
15 </div>
16
17 </body>
18 </html>
```

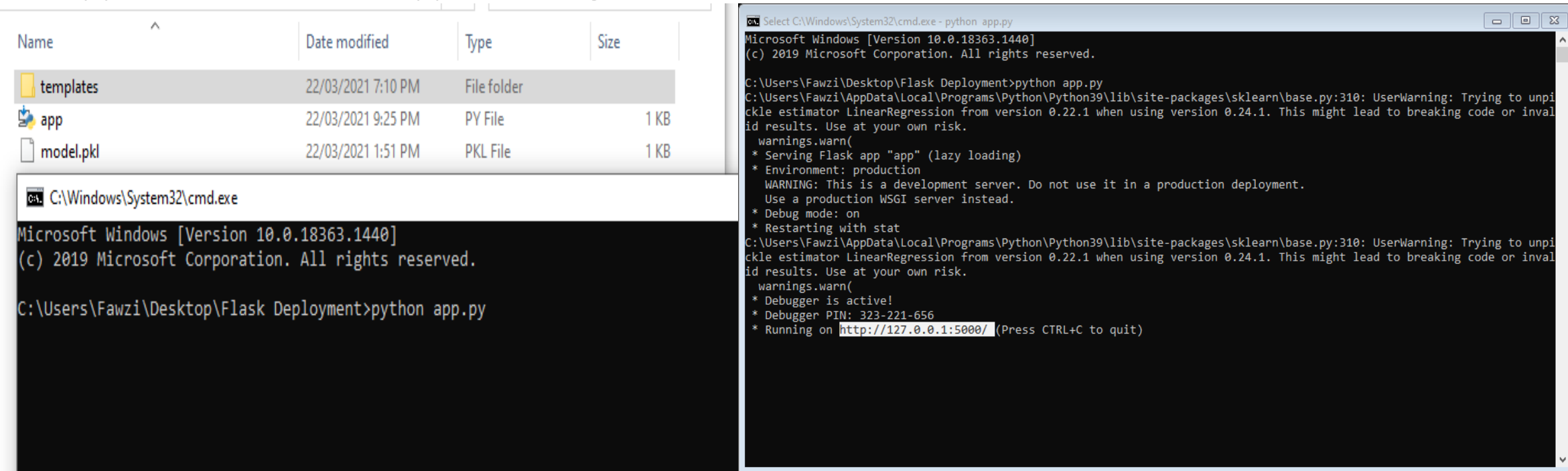
3. Create Flask app

- Create our flask object as shown in the code below
- unpickling the model (*pickle.load*), rendering the html code (*render_template*)
- save the file in the folder which contains templates folder and model.pkl

```
1 import pandas as pd
2 import numpy as np
3 import flask
4 import pickle
5 from flask import Flask, render_template, request
6 app=Flask(__name__)
7 model= pickle.load(open('model.pkl','rb'))
8 @app.route('/')
9 def home():
10     return flask.render_template('index.html')
11 @app.route('/predict',methods = ['POST'])
12 def predict():
13     int_features=[int(x) for x in request.form.values()]
14     final_features= [np.array(int_features)]
15     prediction= model.predict(final_features)
16
17     output= round(prediction[0],2)
18     return render_template('index.html', prediction_text= 'House Price {}'.format(output))
19 if __name__ == '__main__':
20     app.run(port= 5000, debug=True)
```

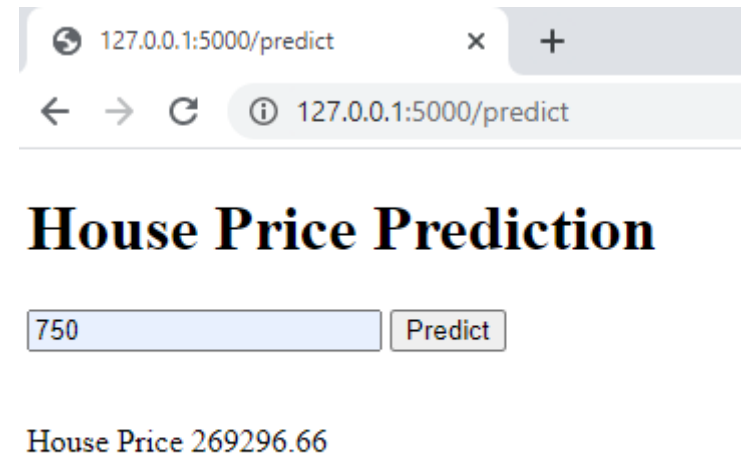
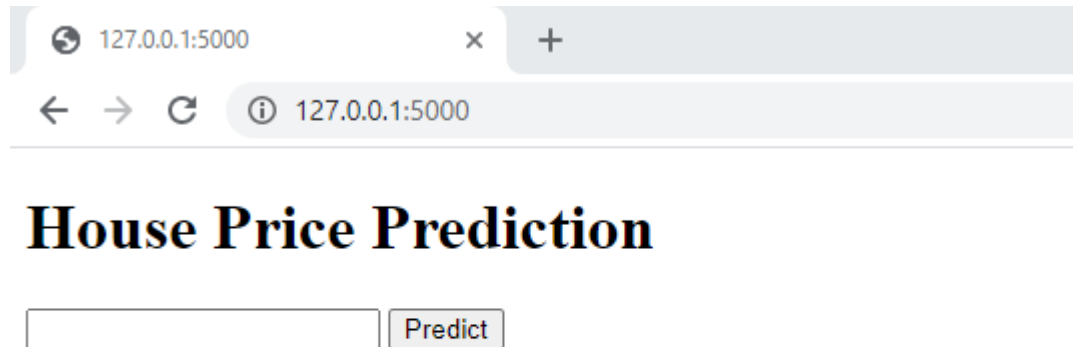
4. Open cmd and Run the app

- Go to the folder contains the files and open cmd and run the python code
- Copy the link where the app is running



5. Clone the link we got in chrome

- We get our web application on our local server as shown below. It requires the *size* feature.
- Enter the size and click predict it will show the prediction of our ML model



Ex 750 sq.ft, price predicted: 269296.66 \$

Finally, these are the steps needed to deploy our model on flask (web app) and use it to predict house prices.

We can follow the same logic to deploy any model we want on flask by writing the correspondent *html* and *flask app* codes