

Week 4 - Deployment on Flask

Name: Yuhong Chen

Batch Code: LISUM22

Submission Date: June 28th, 2023

Submitted to: Data Glacier-LISUM22

- Installation of Flask

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
zhajideMacBook-Pro:~ yuhongchen$ pip install flask
Collecting flask
  Downloading Flask-2.3.2-py3-none-any.whl (96 kB)
    _____ 96.9/96.9 kB 725.4 kB/s eta 0:00:00
Collecting Werkzeug>=2.3.3 (from flask)
  Downloading Werkzeug-2.3.6-py3-none-any.whl (242 kB)
    _____ 242.5/242.5 kB 958.3 kB/s eta 0:00:00
Collecting Jinja2>=3.1.2 (from flask)
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    _____ 133.1/133.1 kB 1.1 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
    _____ 96.6/96.6 kB 1.1 MB/s eta 0:00:00
Collecting blinker>=1.6.2 (from flask)
  Downloading blinker-1.6.2-py3-none-any.whl (13 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-2.1.3-cp311-cp311-macosx_10_9_x86_64.whl (13 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-2.3.6 blinker-1.6.2 click-8.1.3 flask-2.3.2 itsdangerous-2.1.2
zhajideMacBook-Pro:~ yuhongchen$ flask --version
Python 3.11.0
Flask 2.3.2
Werkzeug 2.3.6
zhajideMacBook-Pro:~ yuhongchen$
```

- Download toy dataset from Kaggle, I downloaded one Amazon toy data

	A1					
	A	B	C	D	E	F
1	manufacturer	price	stock	reviews	answers	
2	Hornby	3.42	5	15	1	
3	FunkyBuys	16.99		2	1	
4	ccf	9.99	2	17	2	
5	Hornby	39.99		1	2	
6	Hornby	32.19		3	2	
7	Generic	6.99		2	1	
8	Hornby	24.99		2	1	
9	Hornby	69.93	3	36	7	
10	Hornby	235.58	4	1	1	
11	Chuggington		1	8	1	
12	Hornby	27.49	6	1	1	
13	Kato (USA)	273.6		1	1	
14	Bachmann	9.6	2	1	1	
15	Hornby	119.5	2	3	1	
16	Kato		18	1	1	
17	Kato		12	1	1	
18	Power Trains		2	2	1	
19	Chuggington		1	2	1	
20	Kato	17.08	26	1	1	
21						
22						

- Now we have app.py

```

app.py
Welcome  app.py  model.py
Users > yuhongchen > Desktop > work > summer intern > week4 > app.py > ...
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)

```

- And model.py

```
Users > yuhongchen > Desktop > work > summer intern > week4 > model.py > ...
1  # Importing the libraries
2  import numpy as np
3  import pandas as pd
4  import pickle
5  import sklearn
6  print(sklearn.__version__)
7
8  dataset = pd.read_csv('amazontoy.csv')
9
10 dataset['manufacturer'].fillna(0, inplace=True)
11
12 dataset['reviews'].fillna(dataset['reviews'].mean(), inplace=True)
13
14 X = dataset.iloc[:, :3]
15
16 #Converting words to integer values
17 '''def convert_to_int(word):
18     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
19                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
20     return word_dict[word]'''
21
22 #X['manufacturer'] = X['manufacturer'].apply(lambda x : convert_to_int(x))
23
24 y = dataset.iloc[:, -1]
25
26 from sklearn.linear_model import LinearRegression
27 regressor = LinearRegression()
28
29 #Fitting model with training data
30 regressor.fit(X, y)
31
32 # Saving model to disk
33 pickle.dump(regressor, open('model.pkl','wb'))
34
35 # Loading model to compare the results
36 model = pickle.load(open('model.pkl','rb'))
37 print(model.predict([[2, 2200, 5]]))
```

- Now run app.py using bash

```
zhajideMacBook-Pro:week4v yuhongchen$ python3 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: 116-112-706
```

- Now use the link open 127.0.0.1:5000

