

Week4: Deployment on Flask

Name: Jinwen Li

Batch code: LISUM11: 30

Submission date: 7/26/2022

Submitted to: Data Science Track

Snapshots:

	A	B	C	D	E	F
1	No	square_feet	price			
2	1	150	6450			
3	2	200	7450			
4	3	250	8450			
5	4	300	9450			
6	5	350	11450			
7	6	400	15450			
8	7	600	18450			
9						
10						
11						
12						
13						

```
TrainingModel.py      app.py      page.html
1 import pandas as pd
2 from sklearn.linear_model import LinearRegression
3 import pickle
4
5 # 从csv文件中读取数据, 分别为: X列表和对应的Y列表
6 def get_data(file_name):
7     # 1. 用pandas读取csv
8     data = pd.read_csv(file_name)
9
10    # 2. 构造X列表和Y列表
11    X_parameter = []
12    Y_parameter = []
13    for single_square_feet,single_price_value in zip(data['square_feet'],data['price']):
14        X_parameter.append([float(single_square_feet)])
15        Y_parameter.append(float(single_price_value))
16
17    return X_parameter,Y_parameter
18
19 # 线性回归分析模型训练、保存
20 def linear_model(X_parameter, Y_parameter):
21     #训练模型
22     regr = LinearRegression()
23     regr.fit(X_parameter, Y_parameter)
24     #保存模型
25     pickle.dump(regr, open('model.pkl','wb'))
26
27 if __name__ == '__main__':
28     # 1. 读取数据
29     X, Y = get_data('./house_price.csv')
30
31     # 2. 训练、保存模型
32     linear_model(X, Y)
33
34     print("finish model")
```

```
TrainingModel.py  app.py  page.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8   <div align="center">
9     <h2>House Price Prediction</h2>
10    <br>
11    <form action="{{ url_for('predict')}}" method="post">
12      <b>square feet</b><input type="text" name="housesize" required />
13      <br>
14      <button type="submit">Predict house price</button>
15    </form>
16    <br>
17    {{ prediction_display_area }}
18  </div>
19
20 </body>
21 </html>
22
```

```
TrainingModel.py  app.py  page.html
1 import numpy as np
2 from flask import Flask, request, jsonify, 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('page.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     features_list = [float(x) for x in request.form.values()]
15     features = np.array(features_list).reshape(1,-1)
16     predict_outcome_list = model.predict(features)
17     predict_outcome = round(predict_outcome_list[0],2)
18
19     return render_template('page.html',prediction_display_area='Predicted Price is {}'.format(predict_outcome))
20
21 if __name__ == "__main__":
22     app.run(port=5000,debug = True)
23
```

```
houseprice_flask — python • python app.py — 80x24
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with watchdog (fsevents)
* Debugger is active!
* Debugger PIN: 118-240-813
127.0.0.1 - - [26/Jul/2022 20:40:35] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Jul/2022 20:41:49] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Jul/2022 20:42:07] "POST /predict HTTP/1.1" 200 -
^C
(base) jinwen@jinwendeMacBook-Air houseprice_flask % python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with watchdog (fsevents)
* Debugger is active!
* Debugger PIN: 118-240-813
```



House Price Prediction

square feet

House Price Prediction

square feet

Predicted Price is : 30548.4