

Flask Deployment Tutorial

1. Imported a toy dataset named "Salary_Data.csv" which includes 30 observations of employee years of experience and salary
2. Created "model.py" script to train and test simple linear regression of salary against years of experience with dataset:

```
1 # simple linear regression of salary against years of experience
2
3 # import libraries
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import pandas as pd
7 import pickle
8 import requests
9 import json
10
11 # import salary dataset
12 dataset = pd.read_csv('Salary_Data.csv')
13 X = dataset.iloc[:, :-1].values
14 y = dataset.iloc[:, 1].values
15
16 # split dataset into train and test set
17 from sklearn.model_selection import train_test_split
18 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
19
20 # fit simple linear regression to train set
21 from sklearn.linear_model import LinearRegression
22 regressor = LinearRegression()
23 regressor.fit(X_train, y_train)
24
25 # predict test set
26 y_pred = regressor.predict(X_test)
27
28 # save model with pickle
29 pickle.dump(regressor, open('model.pkl', 'wb'))
30
31 # load model to print prediction result for 5 years of experience
32 model = pickle.load( open('model.pkl', 'rb'))
33 print(model.predict([[5]]))
```

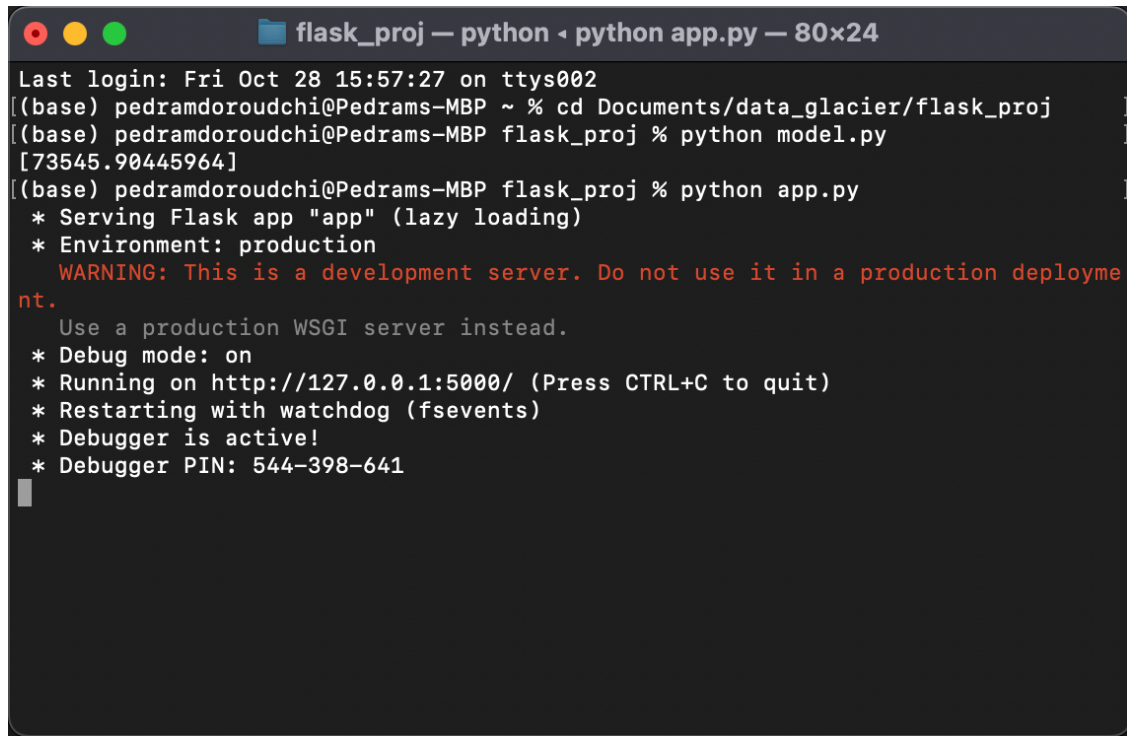
3. Created “index.html” script to format web application template:

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

4. Created “app.py” script which contains flask code and runs the web app:

```
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 # create flask app
6 app = Flask(__name__)
7
8 # load pickle model
9 model = pickle.load(open('model.pkl', 'rb'))
10
11 # route to app home page, designed in "index.html" template
12 @app.route('/')
13 def home():
14     return render_template('index.html')
15
16 # render results on HTML GUI
17 @app.route('/predict', methods=['POST'])
18 def predict():
19     int_features = [int(x) for x in request.form.values()]
20     final_features = [np.array(int_features)]
21     prediction = model.predict(final_features)
22
23     output = round(prediction[0], 2)
24
25     return render_template('index.html', prediction_text='Predicted salary is {}'.format(output))
26
27 if __name__ == "__main__":
28     app.run(debug=True)
```

5. Opened computer terminal and ran “model.py” / “app.py”



```
flask_proj — python ◀ python app.py — 80x24
Last login: Fri Oct 28 15:57:27 on ttys002
[(base) pedramdoroudchi@Pedrams-MBP ~ % cd Documents/data_glacier/flask_proj ]
[(base) pedramdoroudchi@Pedrams-MBP flask_proj % python model.py ]
[73545.90445964]
[(base) pedramdoroudchi@Pedrams-MBP flask_proj % 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: 544-398-641
```

6. Copied and pasted HTTP address into web browser to access web app and verified predicted salary given 5 years of experience:

