

# Week 4: Deployment on Flask

Name: Zyad Al-Azazi

Batch Code: LISP01

Submission Date: 31/03/2021

Submitted to: Data Glacier

## Creating the Model:

For more details refer to the Jupyter notebook on the repository: [zyadalazazi/healthcare-cost-prediction \(github.com\)](https://github.com/zyadalazazi/healthcare-cost-prediction).

### 1) Feature engineering, training and testing the model.

```
In [24]: x_train, x_test, y_train, y_test = train_test_split(df_health, charges, test_size = 0.2, random_state = 100)

x_train = sm.add_constant(x_train)
model = sm.OLS(y_train, x_train)
results = model.fit()

pd.DataFrame({'coef': results.params, 'p-value': round(results.pvalues, 9)})

Out[24]:
```

	coef	p-value
const	-11307.293746	0.000000
age	254.023906	0.000000
sex	13.432983	0.971037
bmi	308.710555	0.000000
children	474.010957	0.032057
smoker	23470.307706	0.000000
region	-325.005527	0.027095

```
In [25]: x_train = x_train.drop('sex', axis = 1)
x_train = x_train.drop('region', axis = 1)

x_test = x_test.drop('sex', axis = 1)
x_test = x_test.drop('region', axis = 1)

final_model = make_pipeline(StandardScaler(), LinearRegression())
final_model.fit(x_train, y_train)
final_model.score(x_test, y_test)
```

### 2) Using pickle to dump and load the model for the purpose of testing.

```
In [27]: import pickle

In [28]: with open('model_pickle', 'wb') as f:
pickle.dump(final_model, f)

In [29]: with open('model_pickle', 'rb') as f:
m = pickle.load(f)

In [33]: m.predict(pd.DataFrame([[29, 19.5, 0, 1]]))

Out[33]: array([[22998.69308379]])
```

## Creating the Python Application and GUI:

- 1) Creating the .py application using Brackets text editor.

```
1 import numpy as np
2 import pandas as pd
3 from flask import Flask, request, render_template
4 import pickle
5
6 app = Flask(__name__)
7 model = pickle.load(open('model_pickle', 'rb'))
8
9 @app.route('/')
10 def home():
11     return render_template('webpage.html')
12
13 @app.route('/predict', methods = ['POST'])
14 def predict():
15
16     age_value = request.form['age']
17     bmi_value = request.form['bmi']
18     children_number = request.form['children']
19     smoker_value = request.form['smoker']
20     query_df = pd.DataFrame([[age_value, bmi_value, children_number, smoker_value]])
21
22     prediction = model.predict(query_df)
23     output = round(prediction[0], 2)
24
25     return render_template('webpage.html', prediction_text = 'Predicted Charge: {} USD'.format(output))
26
27
28 if __name__ == '__main__':
29     app.run(port = 9566)
```

- 2) Creating the .html file for GUI.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Predicting Healthcare Charges</title>
5 </head>
6 <body>
7 <center>
8 <h3><b><center>Predicting Healthcare Charges</center></b></h3>
9 <h3><b><center>Put your information below</center></b></h3>
10
11 <p><font color = "black">Please input information related to age, bmi, number of children and smoking status.</font></p>
12
13 <div class = "Prediction">
14 <center>
15 <form action = "{% url_for('predict') %}", method="POST">
16 <b> Age: <input type = "text" name = "age" placeholder = "Enter Age" required = "required" /><br><br>
17 BMI: <input type = "text" name = "bmi" placeholder = "Enter BMI" required = "required" /><br><br>
18 Children Number: <input type = "text" name = "children" placeholder = "Enter Number of Children" required = "required" /><br><br>
19 Do you smoke? <input type = "text" name = "smoker" placeholder = "Smoker (0 for no, 1 for yes)" required = "required" /><br><br>
20
21 <button type = "submit", class = "btn btn-primary btn-block btn-large">Predict</button>
22 </form>
23 </center>
24 <h3><font color = "white">{{prediction_text}}</font></h3>
25 </div>
26 </center>
27 <body style="background-color:#060555;">
28
29 </body>
30 </html>
```

## Running Flask and Local Testing of the Application:

### 1) Running Flask.

```
C:\Users\MEPI\Documents\Flask>env\Scripts\activate

(env) C:\Users\MEPI\Documents\Flask>FLASK_APP=Deployed_Model.py
'FLASK_APP' is not recognized as an internal or external command,
operable program or batch file.

(env) C:\Users\MEPI\Documents\Flask>set FLASK_APP=Deployed_Model.py

(env) C:\Users\MEPI\Documents\Flask>flask run
* Serving Flask app "Deployed_Model.py"
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [28/Mar/2021 16:28:41] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [28/Mar/2021 16:28:56] "[37mPOST /predict HTTP/1.1[0m" 200 -
```

### 2) Result from the app.

**Predicting Healthcare Charges**

**Put your information below**

Please input information related to age, bmi, number of children and smoking status.

Age:

BMI:

Children Number:

Do you smoke?

**Predicted Charge: 21970.84 USD**

## Deployment on Heroku:

- 1) Linking Heroku to the GitHub repository containing the files of the application.

The screenshot shows the Heroku dashboard for a user named 'zvdalazazi'. The 'Deployment method' section is active, showing that the app is connected to GitHub. The 'Automatic deploys' section is also visible, indicating that the app is configured for automatic deployments from GitHub.

Deployment method

Heroku Git (via Heroku CLI) | GitHub (Connected) | Container Registry (via Heroku CLI)

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [zvdalazazi/healthcare-cost-prediction](#) by [zvdalazazi](#) [Disconnect...](#)

Releases in the [activity feed](#) link to GitHub to view commit diffs

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. Deploys happen automatically; be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#)

Choose a branch to deploy.

- 2) Choosing the “Manual Deploy” option.

The screenshot shows the Heroku dashboard for a user named 'zvdalazazi'. The 'Manual deploy' section is active, showing the 'Deploy a GitHub branch' option. The 'Deploy Branch' button is visible, and the 'Deploy to Heroku' status is confirmed.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#)

Choose a branch to deploy

[Deploy Branch](#)

Receive code from GitHub

Build master / [assemble](#)

Release phase

Deploy to Heroku

Your app was successfully deployed.

[View](#)

### 3) Checking the result of deployment.



The screenshot shows a web browser window with the URL <https://healthcare-cost-prediction-2.herokuapp.com>. The page has a dark green background and contains the following text and form elements:

- Predicting Healthcare Charges**
- Put your information below**
- Please input information related to age, bmi, number of children and smoking status.**
- Age:**
- BMI:**
- Children Number:**
- Do you smoke?**
- Predict** button