

Week5: Cloud API Deployment

Name: Ilyas Nayle

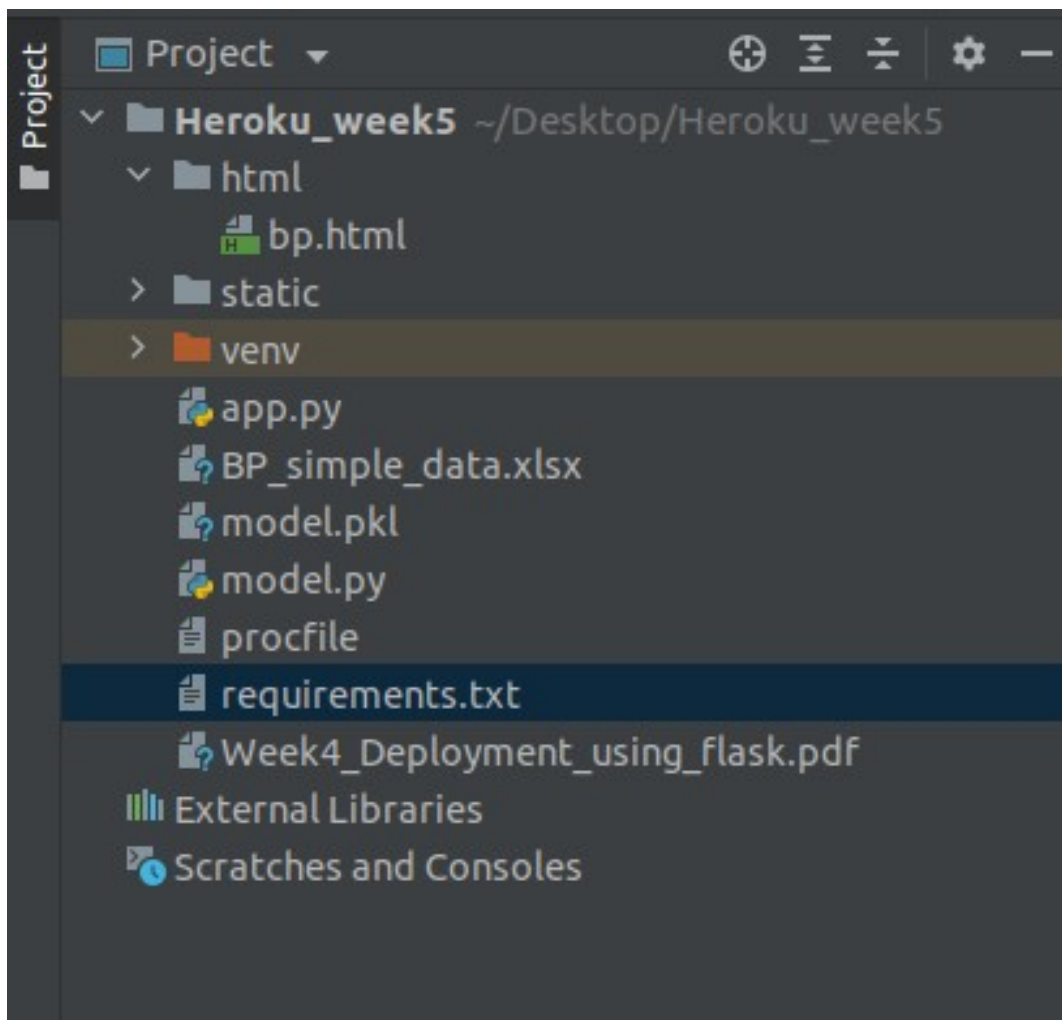
Batch Code: LISUM03

Submission Date: 19/09/2021

Submitted to: Data Glacier

For the Week5- Cloud API Deployment the following steps were applied.

Directory:



Step 1:

- Found a sample Blood pressure Data
- Converted the .xlsx file format to .pkl using python code.
- Saved the file in model.pkl in the directory

```
app.py × model.py × bp.html × style.css ×
1 import pandas as pan
2 import pickle
3 from sklearn.linear_model import LinearRegression
4
5 data = pan.read_excel('BP_simple_data.xlsx')
6
7 x1 = data.iloc[:, 1:]
8 x2 = data.iloc[:, 0:1]
9
10 reg = LinearRegression()
11 reg.fit(x1, x2)
12
13 prediction = reg.predict(x1)
14 pickle.dump(reg, open('model.pkl', 'wb'))
15
```

Step 2:

- Creating of the HTML (bp.html) and CSS (style.css) file

```
bp.html
app.py x model.py x bp.html x style.css x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5
6 <title>Predict BP</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=Arino' 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
12 <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
13
14 </head>
15
16 <body>
17 <div>
18 <h1 style="color: rgb(72,111,148); text-align: center;" >PREDICT YOUR BLOOD PRESSURE</h1>
19
20 <h2 class="text" >What is the Blood Pressure?</h2>
21
22 <p class="para">
23 The blood pressure is the pressure of the blood within the arteries. It is produced primarily by the contraction of the heart muscle.
24 Its measurement is recorded by two numbers. The first (systolic pressure) is measured after the heart contracts and is highest.</p></div>
25 <hr>
26 <div class="input">
27 <!-- Main Input For Receiving Query to our ML -->
28 <form action="{{ url_for('predict') }}" method="post">
29 <p style="color: #ffffff"> Please Enter your information to predict your Blood Pressure.</p>
30 <p style="color: #ffffff"> * required</p>
31 <p><input type="text" name="age" placeholder="* Age" required="required" /></p>
32 <p><input type="text" name="age" placeholder="* Age" required="required" /></p>
33 <p><input type="text" name="weight" placeholder="* Weight" required="required" /></p>
34 <p><input type="text" name="bsa" placeholder="* Body Surface Area( sq m)" required="required" /></p>
35 <p><input type="text" name="hypertension" placeholder="* Duration of Hypertension" required="required" /></p>
36 <p><input type="text" name="pulse" placeholder="* Basal Pulse(b/m)" required="required" /></p>
37 <p><input type="text" name="stress" placeholder="* Stress" required="required" /></p>
38
39 <p>
40 <div>
41 <button class="button button1" type="submit" type = "submit">Predict My Blood Pressure </button>
42 <b style="color: #8ab1c4;">{{ prediction_text }} </b>
43 </div>
44
45 </p>
46
47 </form>
48 </div>
49
50
51
52 <br>
53 <hr>
54 <footer>
55 <br>
56 <p style="text-align:center; color: #88ab12"><b> &copy; Ilyas Nayle 13/09/2021</b><br>
57
58 </footer>
59 </body>
60 </html>
```

```
app.py × model.py × bp.html × style.css ×
1 body, html {
2     height: 100%;
3     line-height: 1.8;
4 }
5
6 body {
7     background-image: url("new.jpg");
8     background-repeat: no-repeat;
9     background-size: cover;
10    background-attachment: fixed;
11 }
12
13 input[type=text] {
14     width: 20%;
15     padding: 10px 15px;
16     margin: 8px 0;
17     box-sizing: border-box;
18     border: none;
19     background-color: #052f3f;
20     color: #e5dddd;
21     position: center;
22 }
23
24 .button {
25     background-color: #7b491c; /* Green */
26     border: none;
27     color: white;
28     padding: 16px 32px;
29     text-align: center;
30     text-decoration: none;
31     display: inline-block;
32     font-size: 16px;
33 }
```

```
app.py × model.py × bp.html × style.css ×
37
38
39 .button1 {
40     background-color: #b91a3a;
41     color: #172540;
42     border: none;
43 }
44
45
46 .button1:hover {
47     background-color: #082c36;
48     color: #d6152a;
49 }
50
51 .text{
52     border: 1px solid #15c6f9;
53     padding: 4px;
54     border-style: dotted;
55     border-top-left-radius: 5px;
56     border-top-right-radius: 5px;
57     text-align:center ;
58     text-transform: capitalize;
59     color: #ffffff;
60 }
61
62 .para{
63     text-indent: 20px;
64     color: #15c6f9;
65     letter-spacing: 2px;
66 }
```

Step 3:

- Creating the Web Application app.py

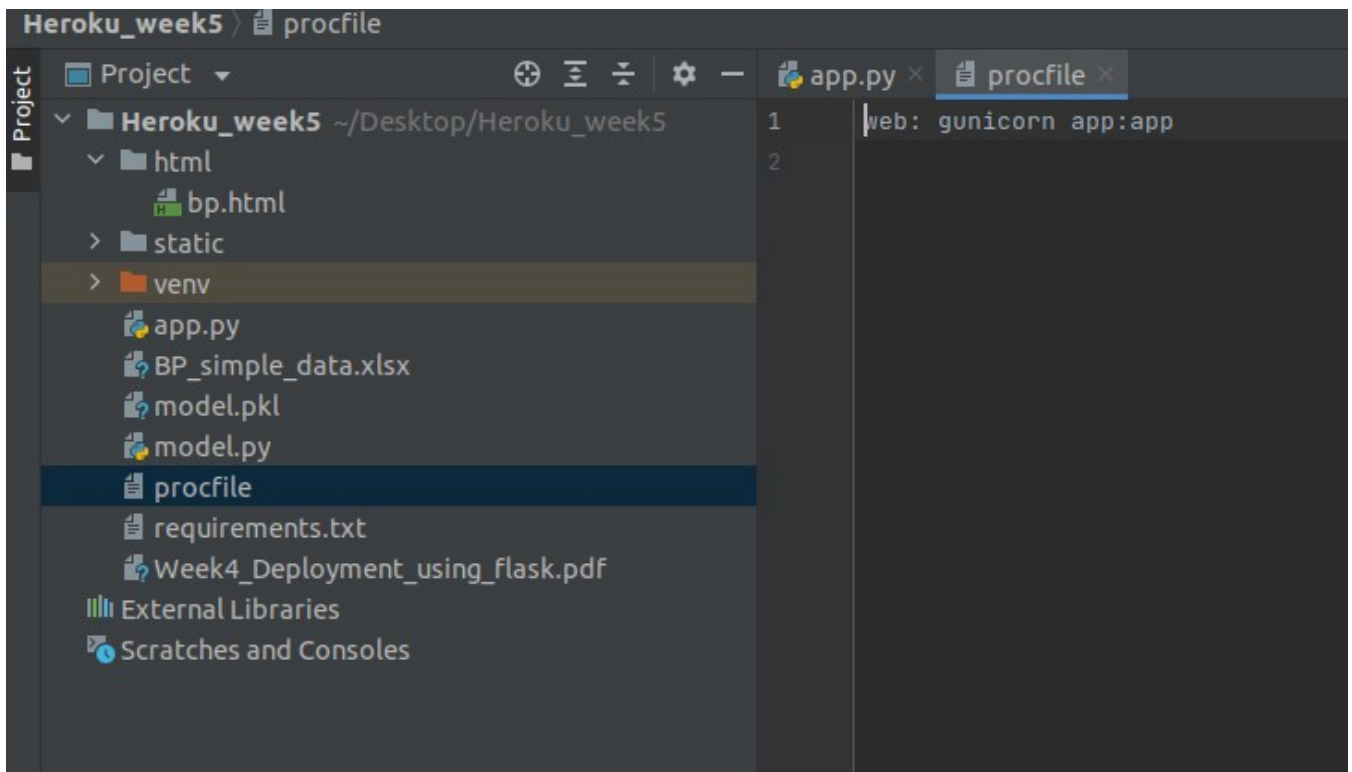
```
app.py × model.py × bp.html × style.css ×
3 import numpy as np
4 from flask import Flask, request, render_template
5 import pickle
6
7 app = Flask(__name__, template_folder='html')
8 model = pickle.load(open('model.pkl', 'rb'))
9
10
11 @app.route('/')
12 def home():
13     return render_template('bp.html')
14
15
16
17 @app.route('/predicted', methods=['POST'])
18 def predict():
19     """
20     For rendering results on HTML GUI
21     """
22     int_features = [int(float(x)) for x in request.form.values()]
23     final_features = [np.array(int_features)]
24     prediction = model.predict(final_features)
25
26     output = np.round(prediction[0], 2)
27
28     return render_template('bp.html', prediction_text='Your Blood Pressure is {} mm Hg. Be Healthy.'.format(output))
29
30
31
32 if __name__ == "__main__":
33     app.run(debug=True)
```

Step 4: Testing and Deployment of the model using command prompt to check if our app runs properly

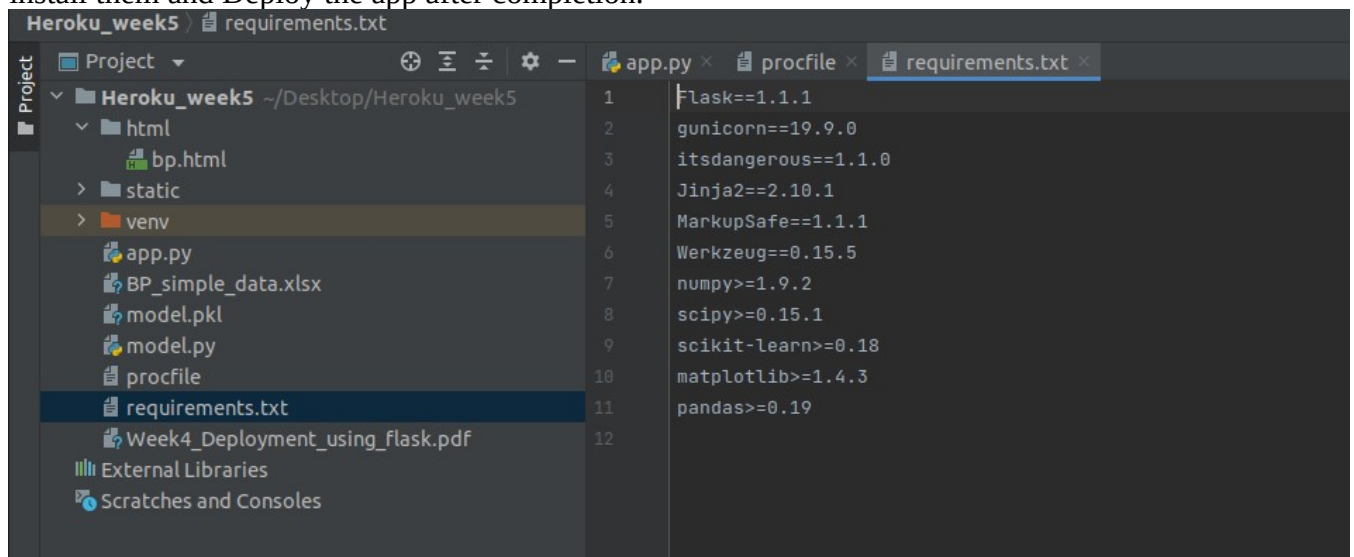
- We open the command prompt
- Navigate to the location of the folder
- Run the file as: python3 app.py

```
Terminal: Local × + ▾
(venv) coder_me_ilyas@ilyasnayle:~/PycharmProjects/Week4_flask_development$ python3 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 stat
* Debugger is active!
* Debugger PIN: 483-454-315
```

Step 5: Creating the Procfile(Without giving it any extension) which tells the heroku about the process beforehand.



Step 6: Creating the requirements.txt file which is for letting Heroku to set the application libraries and install them and Deploy the app after completion.



Step 7: Adding and pushing all the files in GitHub repository(in my case I created a new repository)

The screenshot shows the GitHub repository page for 'FutureCoderme/week5_heroku'. The repository is public and has 1 branch (main) and 0 tags. The file list includes: html, static/css, BP_simple_data.xlsx, LICENSE, Procfile, README.md, Week4_Deployment_using_flask.pdf, app.py, model.pkl, model.py, and requirements.txt. The README.md file is selected, showing the text 'week5_heroku'. The right sidebar contains sections for About, Releases, Packages, Environments (heroku-week5-app is active), and Languages (HTML 52.6%, CSS 24.0%, Python 23.4%).

Step 8: Creating Heroku account and linking GitHub with the Heroku account.

The screenshot shows the Heroku dashboard for the 'heroku-week5-app'. The top navigation bar includes 'Personal', 'heroku-week5-app', and 'More'. The main content area has tabs for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. The 'Deploy' tab is selected, showing options to add the app to a pipeline or to a stage in a pipeline. The 'Deployment method' section shows 'Heroku Git' (Use Heroku CLI), 'GitHub' (Connected), and 'Container Registry' (Use Heroku CLI). The 'App connected to GitHub' section shows the app is connected to 'FutureCoderme/week5_heroku' by 'FutureCoderme', with a 'Disconnect...' button.

Step 9: Deploying the launch.

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

main

Deploy Branch

Receive code from GitHub



Build main 0d38910e



Release phase



Deploy to Heroku



Your app was successfully deployed.

[View](#)

After the successful deployment we will get the link for launching our app and the result is as below.

