

Week 4: Deployment on Flask

Name	Runa Veigas
Batch Code	LISUM01
Submission Date	04 July 2021
Submitted To	Data Glacier Team

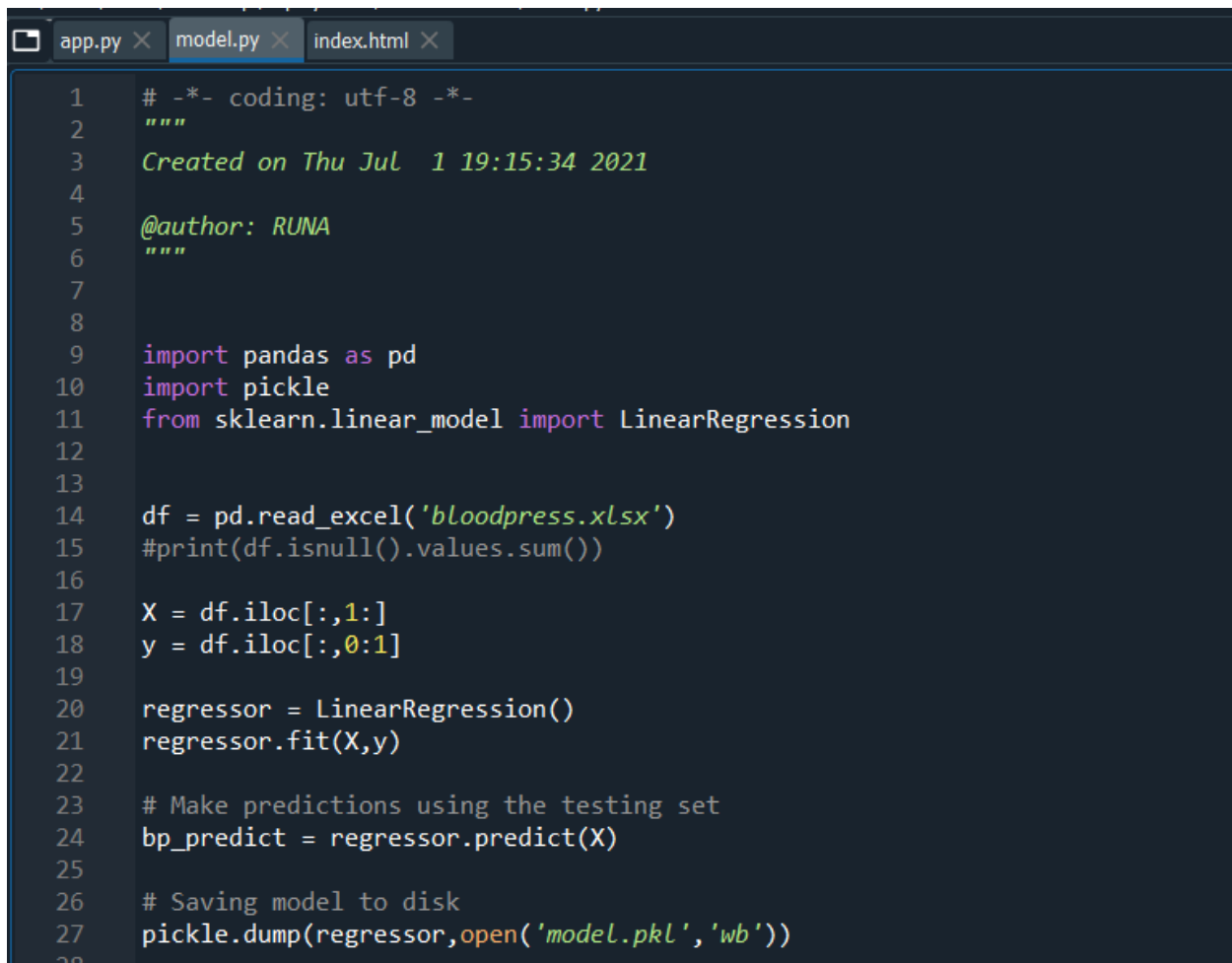
Directory Structure

Name	Date Modified
static	02-07-2021 22:54
css	02-07-2021 22:57
style9.css	03-07-2021 00:42
templates	03-07-2021 11:12
index.html	03-07-2021 11:09
app.py	03-07-2021 00:18
bloodpress.xlsx	02-07-2021 19:53
model.pkl	02-07-2021 22:33
model.py	04-07-2021 13:00

Steps Involved in the Deployment

Step 1:

1. Train the Blood Pressure Predictor model
2. Use the pickle operation to serialize the machine learning algorithm (Linear Regression) and dumping the serialized format to the file **model.pkl** in a binary write format.
3. Save the file in **model.py**



```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Jul  1 19:15:34 2021
4
5  @author: RUNA
6  """
7
8
9  import pandas as pd
10 import pickle
11 from sklearn.linear_model import LinearRegression
12
13
14 df = pd.read_excel('bloodpress.xlsx')
15 #print(df.isnull().values.sum())
16
17 X = df.iloc[:,1:]
18 y = df.iloc[:,0:1]
19
20 regressor = LinearRegression()
21 regressor.fit(X,y)
22
23 # Make predictions using the testing set
24 bp_predict = regressor.predict(X)
25
26 # Saving model to disk
27 pickle.dump(regressor,open('model.pkl','wb'))
28
```

Step 2: Creating the HTML file and saving in **index.html** and **CSS** (for styling) file.

```
app.py x model.py x index.html x
1 <!DOCTYPE html>
2 <html>
3 <!-- From https://codepen.io/frtytyler/pen/EGdtg-->
4 <head>
5   <meta charset="UTF-8">
6   <title>Blood Pressure Prediction 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
12
13  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
14  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
15  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
16
17  <link rel="stylesheet" href="{{ url_for('static', filename = 'css/style9.css') }}">
18 </head>
19
20 <body>
21   <div class="w3-container w3-light-grey">
22     <h1 class="w3-center",style="padding:128px 16px">PREDICT YOUR BLOOD PRESSURE</h1>
23
24     <!-- Main Input For Receiving Query to our ML -->
25     <form action="{{ url_for('predict') }}" method="post">
26       <p><input class="w3-input w3-border" type="text" name="age" placeholder="Age(years)" required="required" /></p>
27       <p><input class="w3-input w3-border" type="text" name="weight" placeholder="Weight(kg)" required="required" /></p>
28       <p><input class="w3-input w3-border" type="text" name="bsa" placeholder="Body Surface Area(sq m)" required="required" /></p>
29       <p><input class="w3-input w3-border" type="text" name="hypertension" placeholder="Duration of Hypertension(years)" required="required" /></p>
30       <p><input class="w3-input w3-border" type="text" name="pulse" placeholder="Basal Pulse(beats per minute)" required="required" /></p>
31       <p><input class="w3-input w3-border" type="text" name="stress" placeholder="Stress Index" required="required" /></p>
32
33       <p>
34         <div class="center">
35           <button type="submit" class="w3-button w3-black" type="submit">
36             <i class="fa fa-heart"></i>
37             Predict Blood Pressure
38           </button>
39         </div>
40       </p>
41
42     </form>
43
44   </div>
45   <br>
46   <br>
47   {{ prediction_text }}
48
49 </div>
50
```

```
app.py x model.py x index.html x style9.css x
1 body, html {
2   height: 100%;
3   line-height: 1.8;
4 }
5
6 /* Full height image header */
7 .bgimg-1 {
8   background-position: center;
9   background-size: cover;
10  background-image: url("bp.jpg");
11  min-height: 100%;
12 }
13
14 .w3-bar .w3-button {
15   padding: 16px;
16 }
17
18 .center {
19   display: flex;
20   justify-content: center;
21   align-items: center;
22   height: 200px;
23 }
```

Step 3: Creating the web application file.

```
app.py x model.py x index.html x style9.css x
1  #-*- coding: utf-8 -*-
2  """
3  Created on Fri Jul  2 10:49:56 2021
4
5  @author: RUNA
6  """
7
8  import numpy as np
9  from flask import Flask, request, render_template
10 import pickle
11
12
13 app = Flask(__name__)
14 model = pickle.load(open('model.pkl', 'rb'))
15
16 @app.route('/')
17 def home():
18     return render_template('index.html')
19
20 @app.route('/predict',methods=['POST'])
21 def predict():
22     '''
23     For rendering results on HTML GUI
24     '''
25     int_features = [int(float(x)) for x in request.form.values()]
26     final_features = [np.array(int_features)]
27     prediction = model.predict(final_features)
28
29     output = np.round(prediction[0], 2)
30
31     return render_template('index.html', prediction_text='The predicted Blood Pressure is {} mm Hg'.format(output))
32
33
34 if __name__ == "__main__":
35     app.run(debug=True)
```

1. Creating the object of class **Flask**.
2. Whenever user visits that URL **'/'**, **index()** method would be called automatically, and the **index()** method returns our main HTML page called **index.html**.
The **flask.render_template()** looks for the this **index.html** file in the **templates** folder and dynamically generates/renders a HTML page for the end user
3. **@app.route ('/predict')** maps the **predict()** method with the **/predict** URL , this takes the input given by the user, does all the pre-processing and generates the final feature vector, runs the model on it and gets the final prediction.

Step 4: Deployment of the model

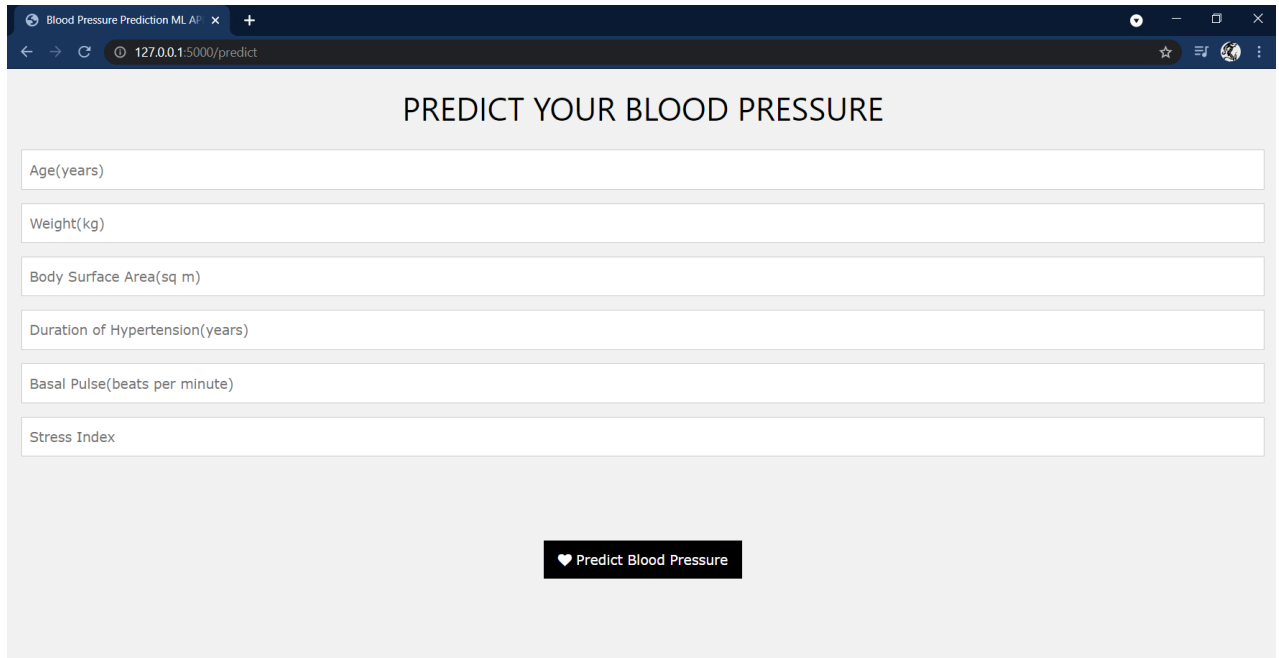
1. Open the Command Prompt.
2. Navigate to the location where the application file is present.
3. Run the file as: **python app.py**

```
(base) C:\Users\RUNA\Internship\Deployment\Blood Pressure>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
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 270-629-930
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

<http://127.0.0.1:5000/> this is where the web application is running

4. Paste this URL in the web browser and the resulting web application will be displayed.

Step 5: Final Result



Blood Pressure Prediction ML API

127.0.0.1:5000/predict

PREDICT YOUR BLOOD PRESSURE

Age(years)

Weight(kg)

Body Surface Area(sq m)

Duration of Hypertension(years)

Basal Pulse(beats per minute)

Stress Index

♥ Predict Blood Pressure

The predicted Blood Pressure is [135.04] mm Hg

*****Thank You*****