

Model Deployment on Flask

Name: Robin Masawi

Batch Code: LISP01

Submission Date: 23-March-2021

Submitted To: Data Glacier

1. Built a simple regression model then saved the model by serializing it using pickle.

```
model.py > ...
1  import pandas as pd
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  from sklearn.linear_model import LinearRegression
6  from sklearn import datasets
7  data = datasets.load_diabetes()
8
9  df = pd.DataFrame(data.data)
10 df.columns = data.feature_names
11 df['target'] = data.target
12
13 x = df[["age", "sex", "bmi", "bp"]]
14 y = df[["target"]]
15
16 regressor = LinearRegression()
17 regressor.fit(x, y)
18
19 import pickle
20 pickle.dump(regressor, open('model.pkl', 'wb'))
```

2. Create an index.html file to allow user to enter details (age, sex, bmi, bp) and displays the predicted diabetes progression.

```
index.html
templates > index.html > ...
1  <!DOCTYPE html>
2
3  <html >
4
5  <head>
6
7      <meta charset="UTF-8">
8
9      <title>Diabetes Linear Regression Model</title>
10
11      <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
12
13      <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
14
15      <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
16
17      <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
18
19      <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
20
21
22
23  </head>
24
25
26
```

```
27 <body>
28
29     <div class="login">
30
31         <h1>Quantitative Measure of Diabetes Progression One Year After Baseline</h1>
32
33
34
35         <!-- Main Input For Receiving Query to our ML -->
36
37         <form action="{{ url_for('predict')}}" method="post">
38
39             <input type="text" name="age" placeholder="Age" required="required" />
40             <input type="text" name="sex" placeholder="Sex" required="required" />
41             <input type="text" name="bmi" placeholder="BMI" required="required" />
42             <input type="text" name="bp" placeholder="BP" required="required" />
43
44             <button type="submit" class="btn btn-primary btn-block btn-large">Predict measure of diabetes progression</button>
45
46         </form>
47
48
49
50         <br>
51
52         <br>
53
54         {{ prediction_text }}
55
56
57
58     </div>
59
60 </body>
61
62 </html>
```

3. Made an API which receives details of the diabetes patients through GUI and computes the predicted diabetes progression on the model.

```
app.py x
app.py > ...
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
11     return render_template('index.html')
12
13
14 @app.route('/predict', methods=['POST'])
15 def predict():
16
17     ...
18
19     For rendering results on HTML GUI
20
21     ...
22
23     float_features = [float(x) for x in request.form.values()]
24     final_features = [np.array(float_features)]
25     prediction = model.predict(final_features)
26
27     output = prediction[0]
28
29     return render_template('index.html', prediction_text='Diabetes progression one year after baseline is {}'.format(output))
```

```
30
31
32
33 @app.route('/predict_api', methods=['POST'])
34 def predict_api():
35
36     ...
37
38     For direct API calls through request
39
40     ...
41
42     data = request.get_json(force=True)
43     prediction = model.predict([np.array(list(data.values()))])
44
45     output = prediction[0]
46
47     return jsonify(output)
48
49
50
51 if __name__ == "__main__":
52
53     app.run(debug=True)
```

4. Used css to make the interactive web interface.

```
static > css > # style.css.css > ...
1  @import url(https://fonts.googleapis.com/css?family=Open+Sans);
2
3
4  html { width: 100%; height:100%; overflow:hidden; }
5
6
7  body {
8
9      width: 100%;
10
11     height:100%;
12
13     font-family: 'Helvetica';
14
15     background: □ #000;
16
17     color: ■ #fff;
18
19     font-size: 24px;
20
21     text-align:center;
22
23     letter-spacing:1.4px;
24
25
26 }
27
28
```

```
29  .login {
30
31     position: absolute;
32
33     top: 40%;
34
35     left: 50%;
36
37     margin: -150px 0 0 -150px;
38
39     width:400px;
40
41     height:400px;
42
43 }
44
45
```

```
44
45
46 .login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }
47
48
49 input {
50
51     width: 100%;
52
53     margin-bottom: 10px;
54
55     background: rgba(0,0,0,0.3);
56
57     border: none;
58
59     outline: none;
60
61     padding: 10px;
62
63     font-size: 13px;
64
65     color: #fff;
66
67     text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
68
69     border: 1px solid rgba(0,0,0,0.3);
70
71     border-radius: 4px;
72
73     box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
74
75     -webkit-transition: box-shadow .5s ease;
76
77     -moz-transition: box-shadow .5s ease;
78
79     -o-transition: box-shadow .5s ease;
80
81     -ms-transition: box-shadow .5s ease;
82
83     transition: box-shadow .5s ease;
84
85 }
```