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 > ...

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression
from sklearn import datasets
data = datasets.load_diabetes()

df = pd.DataFrame(data.data)
df.columns = data.feature_names
df['target'] = data.target

x = df[["age","sex","bmi","bp"]]
y = df[["target"]]

regressor = LinearRegression()
regressor.fit(x, y)

import pickle
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 > ...

```
<!DOCTYPE html>

<html >

<head>

  <meta charset="UTF-8">

  <title>Diabetes Linear Regression Model</title>

  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">



</head>


<body>

<div class="login">

  <h1>Quantitative Measure of Diabetes Progression One Year After Baseline</h1>



    <!-- Main Input For Receiving Query to our ML -->

  <form action="{{ url_for('predict')}}"method="post">

      <input type="text" name="age" placeholder="Age" required="required" />
      <input type="text" name="sex" placeholder="Sex" required="required" />
      <input type="text" name="bmi" placeholder="BMI" required="required" />
      <input type="text" name="bp" placeholder="BP" required="required" />

      <button type="submit" class="btn btn-primary btn-block btn-large">Predict measure of diabetes progression</button>

  </form>



  <br>

  <br>

  {{ prediction_text }}



 </div>

</body>

</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

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():

    return render_template('index.html')


@app.route('/predict',methods=['POST'])
def predict():

    '''

    For rendering results on HTML GUI

    '''

    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    prediction = model.predict(final_features)

    output = prediction[0]

    return render_template('index.html', prediction_text='Diabetes progression one year after baseline is {}'.format(output))
```

```python


@app.route('/predict_api',methods=['POST'])
def predict_api():

    '''

    For direct API calls trought request

    '''

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]

    return jsonify(output)


if __name__ == "__main__":

    app.run(debug=True)
```

4. Used css to make the interactive web interface.

static > css > # style.css > ...

```css
@import url(https://fonts.googleapis.com/css?family=Open+Sans);


html { width: 100%; height:100%; overflow:hidden; }


body {
    width: 100%;
    height:100%;
    font-family: 'Helvetica';
    background: #000;
    color: #fff;
    font-size: 24px;
    text-align:center;
    letter-spacing:1.4px;


}
```

```css
.login {
    position: absolute;
    top: 40%;
    left: 50%;
    margin: -150px 0 0 -150px;
    width:400px;
    height:400px;
}
```

```
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  }
```

5. Ran the application using the app.py file.