Week 4: Deployment on Flask

Muhammed Shehab

LICAN01

21-3-2021

Data Glacier

1- “Model deployment – Salary dataset” using Linear Regression (Splitting the dataset into the Training set and Test set).

Splitting the dataset into the Training set and Test set

In [4]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
```

Training the Simple Linear Regression model on the Training set

In [5]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[5]:
```
LinearRegression()
```

Predicting the Test set results

In [7]:
```
y_pred = regressor.predict(X_test)
```

In [12]:
```
pickle.dump(regressor, open('model.pkl','wb'))
```

In [14]:
```
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[2]]))
```

```
[45508.07713028]
```

2. Creating an API that returns data to its users.

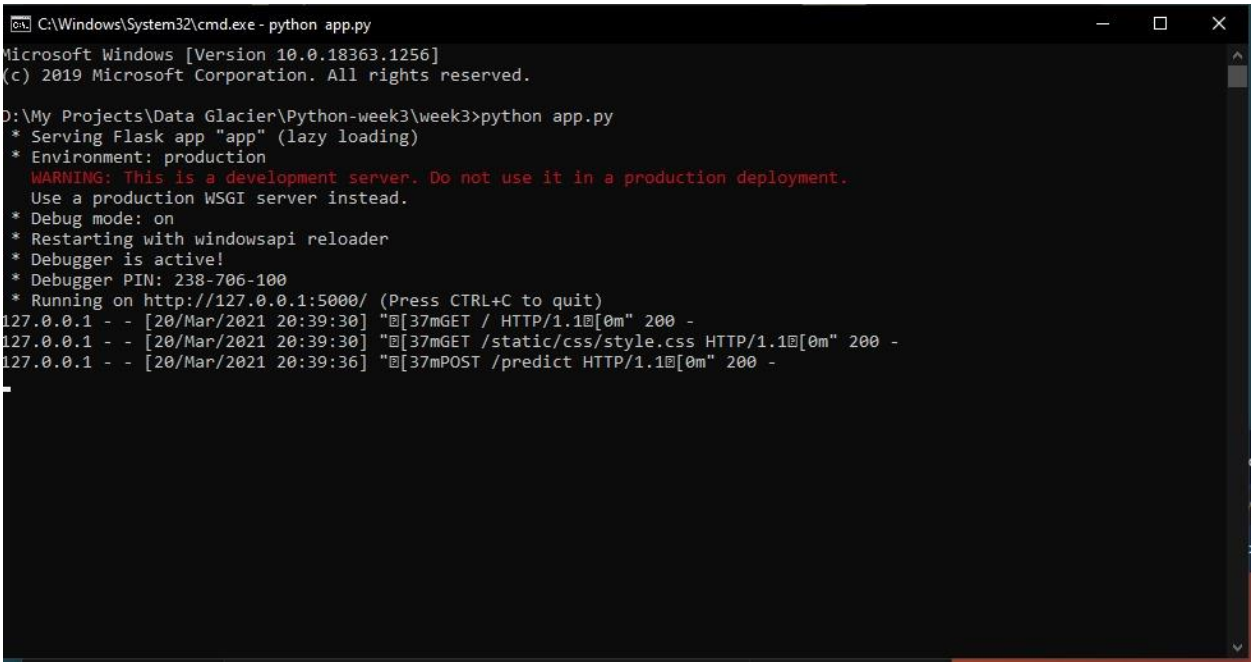
```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

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
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))

@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)
```

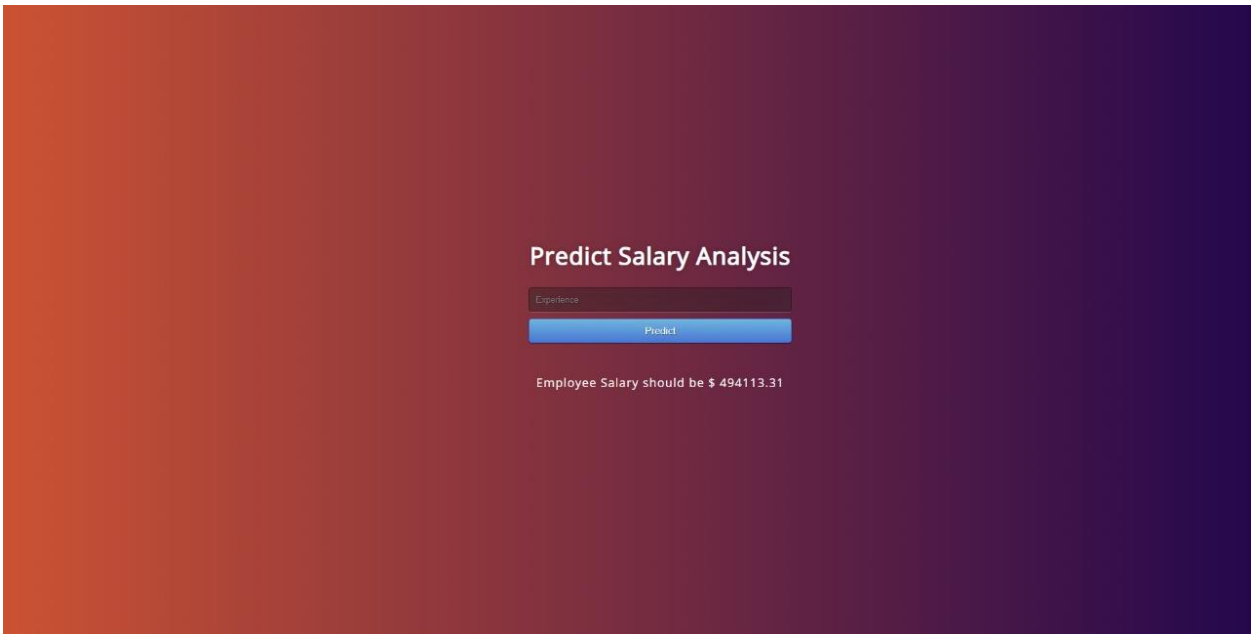
3. Creating HTML & CSS Files (GUI)

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <title>ML API FLAST API</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>Predict Salary Analysis</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="experience" placeholder="Experience" required="required" />
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

```
* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }

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

body {
        width: 100%;
        height:100%;
        font-family: 'Open Sans', sans-serif;
        color: #fff;
        font-size: 18px;
        text-align:center;
        letter-spacing:1.2px;
        background: #23074d;  /* fallback for old browsers */
        background: -webkit-linear-gradient(to right, #cc5333, #23074d);  /* Chrome 10-25, Safari 5.1-6 */
        background: linear-gradient(to right, #cc5333, #23074d); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */



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

.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }

input {
        width: 100%;
        margin-bottom: 10px;
        background: rgba(0,0,0,0.3);
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #fff;
```

4- Output


C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.
D:\My Projects\Data Glacier\Python-week3\week3>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: 238-706-100
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [20/Mar/2021 20:39:30] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [20/Mar/2021 20:39:30] "[37mGET /static/css/style.css HTTP/1.1[0m" 200 -
127.0.0.1 - - [20/Mar/2021 20:39:36] "[37mPOST /predict HTTP/1.1[0m" 200 -


Predict Salary Analysis
Experience
Predict
Employee Salary should be $ 494113.31