

Name: Ruizhe Zhang

Batch code: LISUM10: 30 May - 30 Aug 2022

Submission date: 06/14/2022

Submitted to: Canvas (GitHub URL)

Step 1: Data Choosing

In order to understand how to use Flask better, I select the dataset showing NBA players' points, rebounds, assists and salary (shown in the files as "NBA.csv"), trying to predict NBA player's salary based on his performance:

	A	B	C	D	E	F	G	H
1	pts	rbd	ast	salary				
2	17.6	5	6.7	41.4				
3	27.2	7.9	7	41.4				
4	34.3	6.6	7.5	41.3				
5	25.3	7.8	10.2	39.2				
6	21.5	5.7	3.9	35.5				
7	14.4	3.2	4.4	34.5				
8	19.6	6.9	3.2	34.4				
9	19.9	6.7	6	34.4				
10	20.4	3.9	4.8	34.4				
11	27.1	7.1	4.9	34.4				
242	6	2.3	0.9	2.3				
243	6.9	2	2.5	2.3				
244	8.3	2.2	1.4	2.3				
245	10.1	2.2	0.8	2.3				
246	11.2	2	1.7	2.3				
247	3.6	2.3	0.5	2.2				
248	8.7	4.9	1.2	2.2				
249	8.8	2.1	2.4	2.1				
250	9.3	1.9	1.9	2.1				
251	3.7	2.6	0.4	2				
252	6.7	1.9	1.2	2				
253	7.6	4.5	1	2				
254								

Step 2: Saving the model using model.py (code shown below)

```
1  # Importing the libraries
2  import numpy as np
3  import pandas as pd
4  import pickle
5
6  dataset = pd.read_csv('NBA.csv')
7
8  X = dataset.iloc[:, :3]
9
10 y = dataset.iloc[:, -1]
11
12 from sklearn.linear_model import LinearRegression
13 regressor = LinearRegression()
14
15 #Fitting model with trainig data
16 regressor.fit(X, y)
17
18 # Saving model to disk
19 pickle.dump(regressor, open('model.pkl','wb'))
20
21 # Loading model to compare the results
22 model = pickle.load(open('model.pkl','rb'))
23 print(model.predict([[36, 9, 12]]))
```

Step 3: Changing data type and prediction_text in app.py

```
1  import numpy as np
2  from flask import Flask, request,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     return render_template('index.html')
11
12 @app.route('/predict',methods=['POST'])
13 def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17     int_features = [float(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='The salary of the player should be $ {} million'.format(output))
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

Step 4: Making changes to the HTML template (index.html)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>ML API</title>
6   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7   <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9   <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15   <div class="Login">
16     <h1>Predict NBA Player Salary</h1>
17
18     <!-- Main Input For Receiving Query to our ML -->
19     <form action="{{ url_for('predict')}}" method="post">
20       <input type="text" name="no_of_rooms" placeholder="Points" required="required" />
21       <input type="text" name="area" placeholder="Rebounds" required="required" />
22       <input type="text" name="house_age" placeholder="Assists" required="required" />
23
24       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
25     </form>
26
27     <br>
28     <br>
29     {{ prediction_text }}
30
31   </div>
32   
33
34 </body>
35 </html>
36
```

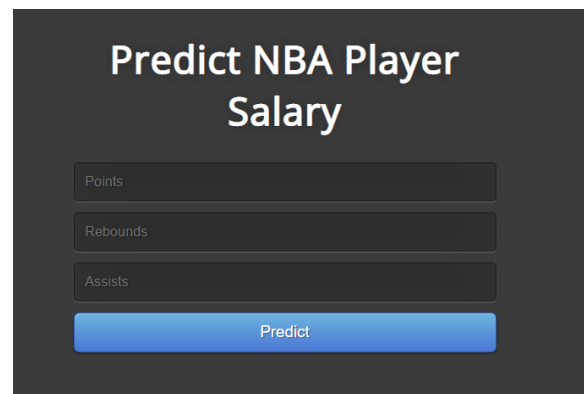
(Only the title and placeholders were changed in order to prevent unexpected errors)

Step 5: Deploying the model on flask

```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [版本 10.0.22000.739]
(c) Microsoft Corporation。保留所有权利。

D:\Flask-Deployment>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
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 772-791-613
```

Step 6: Opening the running app and type in random data to predict:



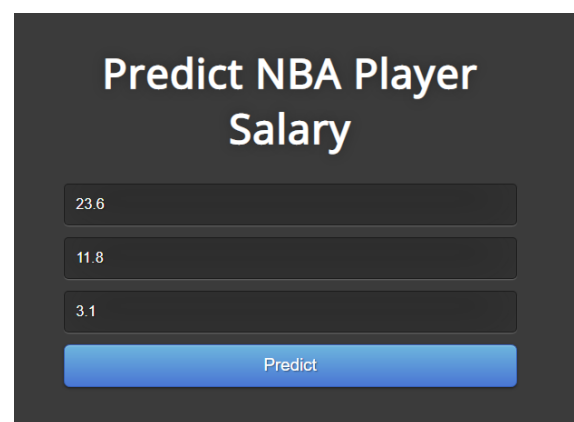
Predict NBA Player Salary

Points

Rebounds

Assists

Predict



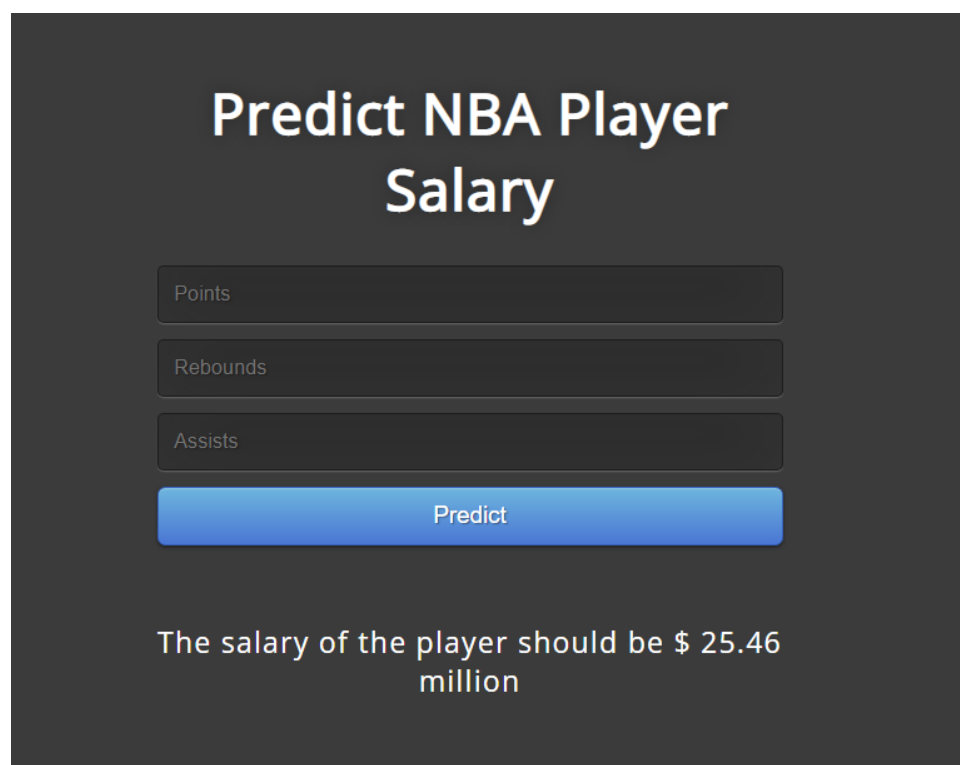
Predict NBA Player Salary

23.6

11.8

3.1

Predict



Predict NBA Player Salary

Points

Rebounds

Assists

Predict

The salary of the player should be \$ 25.46 million