

Name: Ruizhe Zhang

Batch code: LISUM10: 30 May - 30 Aug 2022

Submission date: 06/30/2022

Submitted to: Canvas (GitHub URL)

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								

Part 1: API

Model:

```
temp.py × app.py × model.py ×
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'))
```

App:

```
temp.py × app.py × model.py ×
1 from flask import Flask, request, jsonify
2 import pickle
3 import pandas as pd
4
5 app = Flask(__name__)
6
7 @app.route('/', methods=['GET', 'POST'])
8 def home():
9     if request.method == 'GET':
10
11         data = "Hello World"
12         return jsonify({'data':data})
13
14 @app.route('/predict/')
15 def predict():
16     model = pickle.load(open('model.pkl', 'rb'))
17     pts = request.args.get('pts')
18     rbd = request.args.get('rbd')
19     ast = request.args.get('ast')
20
21     test_df = pd.DataFrame({'pts':[pts], 'rbd':[rbd], 'ast':[ast]})
22
23     pred_price = model.predict(test_df)
24     return jsonify({'The salary of this player should be':str(pred_price)})
25
26 if __name__ == "__main__":
27     app.run(debug=True)
```

Deployment 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。

C:\Users\Ruizhe Zhang\Desktop\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
```

Predicting using endpoints:

http://127.0.0.1:5000/

Save

GET

http://127.0.0.1:5000/

Send

Params

Auth

Headers (8)

Body

Pre-req.

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION		Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

200 OK

13 ms

192 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

"data": "Hello World"

http://127.0.0.1:5000/predict?pts=23.5&rbd=10&ast=5

Save

GET http://127.0.0.1:5000/predict?pts=23.5&rbd=10&ast=5 Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	pts	23.5			
<input checked="" type="checkbox"/>	rbd	10			
<input checked="" type="checkbox"/>	ast	5			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 515 ms 217 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2 "Salary (in million dollars)": "[26.14203389]"
3
```

Part 2: Heroku

Model:

```
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]]))
```

App:

```
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)
```

Deployment on Heroku:

App name

heroku-demo-app22



heroku-demo-app22 is available

Choose a region

 United States



Add to pipeline...

Create app

Deployment method



Heroku Git
Use Heroku CLI



GitHub
Connected



Container Registry
Use Heroku CLI

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [richardzzhang/Heroku-demo](#) by [richardzzhang](#)

Disconnect...

Releases in the [activity feed](#) link to GitHub to view commit diffs

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

main

Deploy Branch

Receive code from GitHub



Build main c398d57d



Release phase



Deploy to Heroku



Your app was successfully deployed.

View

Test:

Predict NBA Player Salary

Predict

The salary of the player should be \$ 27.23 million