

## Performance report

### HTTP API Specification

Endpoint: /recommend/ (e.g. /recommend/u00000) with get,post method

รับค่า param

```
@app.route("/recommend/<user_id>", methods=['GET','POST'])
def recommend(user_id):
    # รับ param body json
    params = request.get_json()
    latitude = float(params.get('latitude'))
    longitude = float(params.get('longitude'))
    max_dis = int(params.get('max_dis',5000))
    size = int(params.get('size', 20))
    sort_dist = int(params.get('sort_dist', 1))
```

Response: in JSON format

```
{
  "restaurants": [
    {
      "difference": 23.66758402660264,
      "displacement": 420.98658171798274,
      "latitude": "13.629531575765236",
      "longitude": "100.75870051163373",
      "restaurant_id": "r1737"
    },
  ],
}
```

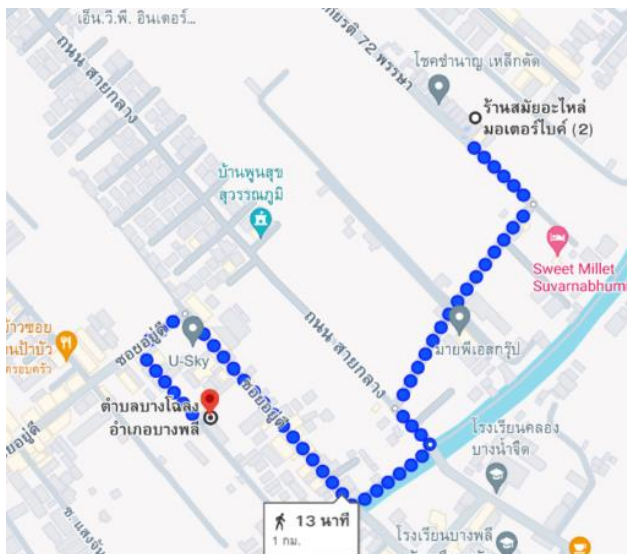
- restaurant\_id : Restaurant\_ID
- difference: Euclidean distance returned from the model
- displacement: Geodesic or great circle displacement using h3
- #latitude,longitude ค่าของ restaurant เก็บไว้เพื่อนำไปลองคำนวณกับ param ที่ได้รับว่าตรงตาม h3 จริงหรือป่าว (สามารถเอาออกได้)

## great circle displacement test

ทดลอง ให้ user ใส่ {

```
"latitude" :13.632325928068196,  
"longitude" :100.76132906024361,  
"max_dis" : 500000,  
"size" : 1,  
"sort_dist" : 1  
}  
  
จะ return json  
"restaurants": [  
  
  {  
  
    "difference": 23.66758402660264,  
    "displacement": 420.98658171798274,  
    "latitude": "13.629531575765236",  
    "longitude": "100.75870051163373",  
    "restaurant_id": "r1737"  
  
  },
```

ลองทดสอบคำนวณจาก website ค่า displacement = 420.99 ตรงตาม h3 ที่ใช้



### Coordinates — first point

Hemisphere (N/S) [N ▾](#)

Latitude 13.632325928068196 [° ▾](#)

Hemisphere (E/W) [W ▾](#)

Longitude 100.76132906024361 [° ▾](#)

### Coordinates — second point

Hemisphere (N/S) [N ▾](#)

Latitude 13.629531575765236 [° ▾](#)

Hemisphere (E/W) [W ▾](#)

Longitude 100.75870051163373 [° ▾](#)

### Great circle distance

Desired distance unit: [m ▾](#)

Great circle distance (sphere) 420.99 m

Great circle distance (ellipsoid) 420.09 m

## Test with parameter

ผมได้ทดลองผ่าน user.small เนื่องจากมีปัญหาเกี่ยวกับตัว docker(ผมเขียนอธิบายไว้ใน readme.md ครับ)  
นอกจากนั้นผมได้ใช้ requestfix.parquet(เปลี่ยนแค่ตัว userid ข้อมูลที่เหลือเหมือนเดิม) แล้วทดลอง test method get,post

- ผมแนบไฟล์ userfull เอาไว้สามารถ import เข้า database แล้วเปลี่ยน requestfix.parquet เป็น request.parquet ได้เลยครับ

```
servertest.py / ...
import pandas as pd
import requests
import time

df = pd.read_parquet('requestfix.parquet')

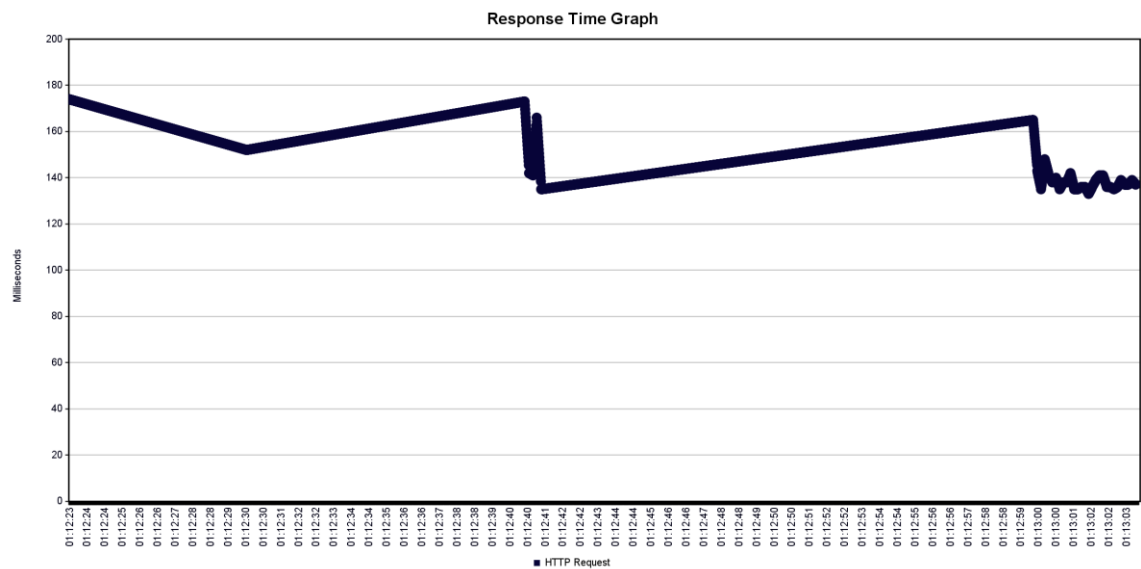
# เก็บค่า ms
response_times = []

# Start timer
start_time = time.time()

for index, row in df.iterrows():
    user_id = row['user_id']
```

หลังจากนั้นสามารถ run code เพื่อทดสอบได้เลยครับ

```
User ID: u00008
Status code: 200
Response: {'restaurants': [{'difference': 24.8340067200266, 'restaurant_id': 'r7317'}, {'difference': 24.665229, 'restaurant_id': 'r7986'}, {'difference': 24.8919739100037, 'displacement': 9680.8503504, 'displacement': 9680.8503504}]}
This userID u00009 has no restaurant near
User ID: u00009
Status code: 200
Response: {'restaurants': []}
requests per second: 6.535672433468119
Average Response Time: 150.99263366336638 ms
```



timeStamp	elapsed	label	responseC	responseM	threadName	dataType	success	failureMes	bytes	sentBytes	grpThread	allThreads	URL	Latency
1.71E+12	165	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	165
1.71E+12	143	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	143
1.71E+12	135	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	135
1.71E+12	148	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	148
1.71E+12	141	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	141
1.71E+12	138	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	138
1.71E+12	140	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	140
1.71E+12	135	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	135
1.71E+12	138	HTTP Requ	200	OK	Thread Gr	text	TRUE		961	318	1	1	http://127.	137

POST

http://127.0.0.1:5000/recommend/u00000

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

7

8

{

"latitude": 13.632325928868196,

"longitude": 100.76132986624361,

"max\_dis": 500000,

"size": 5,

"sort\_dist": 1

}

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 143 ms

Size: 961 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

{

"restaurants": [

{

"difference": 23.66758482668264,

"displacement": 420.98658171798274,

"latitude": 13.629531575765236,

"longitude": 100.758790951163373,

"restaurant\_id": "i1737"

},

]

}

## Jeerakit Seetha

หลังจากการทดลอง Test แล้ว

- Request จะอยู่ที่ 6 Per sec
- AVG MS อยู่ที่ 150

Conclusion:

ผมรู้สึกว่าการนี้เป็นงานที่ทำหยาบมากครับเนื่องจากผมไม่มีความรู้เกี่ยวกับ Docker, Docker-compose เลยจึงต้องเอาเวลาส่วนมากไปนั่งทำความเข้าใจ รวมถึงแก้ปัญหาเรื่อง hardware ผมเลยเน้นไปที่การเชื่อมต่อกับ database sever และตัว app.py ที่ทำให้รับค่าและเอามาประมวลผลได้

ในส่วนของการแก้ไข model และปรับปรุงการรับค่า ผมเลยไม่ได้แก้ไขเพิ่มเติมจากที่มี ทำให้การรับส่งข้อมูลช้าครับ ผมมองว่าถ้ามีเวลาเพิ่มเติมและมี hardware ที่ดีกว่านี้ ผมจะสามารถเข้าไปเช็ค model รวมถึง ลองเปลี่ยน database server เป็นตัวอื่น เพื่อเพิ่มประสิทธิภาพครับ

