

Лабораторна робота 2. Розробка додатків на Python і Flask

Команда 13

Сидоренко Альбіна Олександрівна

Гаєва Поліна Денисівна

Шудрик Андрій Олександрович

Файл скрипту і зображення : https://github.com/PolinaHaieva/P_2_13.git

1.

Клонування репозиторію та перехід до директорії проекту

1_folder_structure.png

```
theia@theiadocker-secacppol:/home/project$ git clone https://github.com/ibm-developer-skills-network/oaqjp-final-project-emb-ai.git
Cloning into 'oaqjp-final-project-emb-ai'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 16 (delta 3), reused 0 (delta 0), pack-reused 9
Receiving objects: 100% (16/16), 8.21 KiB | 8.21 MiB/s, done.
Resolving deltas: 100% (3/3), done.
theia@theiadocker-secacppol:/home/project$ ls
oaqjp-final-project-emb-ai
theia@theiadocker-secacppol:/home/project$ cd oaqjp-final-project-emb-ai
theia@theiadocker-secacppol:/home/project/oaqjp-final-project-emb-ai$ ls
LICENSE  README.md  static  templates
theia@theiadocker-secacppol:/home/project/oaqjp-final-project-emb-ai$
```

2.

Використання Embeddable Watson AI для емоційного аналізу тексту

2a_emotion_detection.png

```
1 import requests
2
3 def emotion_detector(text_to_analyze):
4     url = 'https://sn-watson-emotion.labs.skills.network/v1/watson.runtime.nlp.v1/NlpService/EmotionPredict'
5     headers = {"grpc-metadata-mm-model-id": "emotion_aggregated-workflow_lang_en_stock"}
6     input_json = { "raw_document": { "text": text_to_analyze } }
7     response = requests.post(url, json = input_json, headers=headers)
8     return response.text
9
```

Тестування функції

2b_application_creation.png

```
theia@theiadocker-secacppol:/home/project/oaqjp-final-project-emb-ai$ python3.11
Python 3.11.9 (main, Apr 6 2024, 17:59:24) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from emotion_detection import emotion_detector
>>> print(emotion_detector("I love this new technology"))
...
...
KeyboardInterrupt
>>> print(emotion_detector("I love this new technology"))
{"emotionPredictions":[{"emotion":{"anger":0.0132405795, "disgust":0.0020517302, "fear":0.009090992, "joy":0.9699522, "sadness":0.054984167}, "target":"","emotionMentions":[{"span":{"begin":0, "end":26, "text":"I love this new technology"}, "emotion":{"anger":0.0132405795, "disgust":0.0020517302, "fear":0.009090992, "joy":0.9699522, "sadness":0.054984167}}]}], "producerId":{"name":"Ensemble Aggregated Emotion Workflow", "version":"0.0.1"}}
>>>
```

3.

Модифікація відповіді для користувача

3a_output_formatting.png

```
1 import requests
2 import json
3
4 def emotion_detector(text_to_analyze):
5     url = 'https://sn-watson-emotion.labs.skills.network/v1/watson.runtime.nlp.v1/NlpService/EmotionPredict'
6     headers = {"grpc-metadata-mm-model-id": "emotion_aggregated-workflow_lang_en_stock"}
7     input_json = { "raw_document": { "text": text_to_analyze } }
8
9     response = requests.post(url, json=input_json, headers=headers)
10
11     response_data = json.loads(response.text)
12     emotions = response_data['emotionPredictions'][0]['emotion']
13     dominant_emotion = max(emotions, key=emotions.get)
14
15     result = {
16         'anger': emotions['anger'],
17         'disgust': emotions['disgust'],
18         'fear': emotions['fear'],
19         'joy': emotions['joy'],
20         'sadness': emotions['sadness'],
21         'dominant_emotion': dominant_emotion
22     }
23     return result
```

Тестування функції

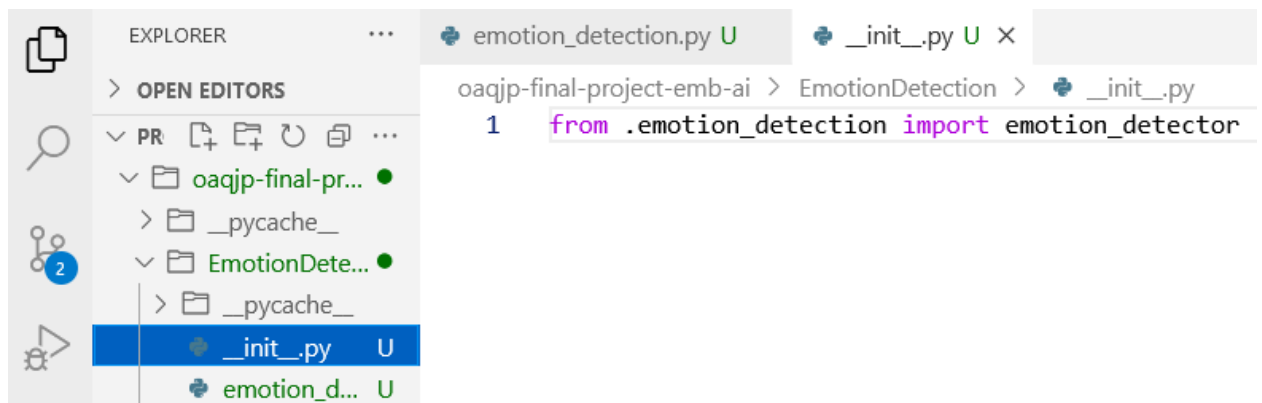
3b_formatted_output_test.png

```
theia@theiadocker-secacppol:/home/project/oaqjp-final-project-emb-ai$ python3.11
Python 3.11.9 (main, Apr  6 2024, 17:59:24) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from emotion_detecction import emotion_detector
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'emotion_detecction'
>>> from emotion_detection import emotion_detector
>>> print(emotion_detector("I am so happy I am doing this"))
{'anger': 0.0043079085, 'disgust': 0.00041127237, 'fear': 0.0037504788, 'joy': 0.9918804, 'sadness': 0.014091322,
'dominant_emotion': 'joy'}
```

4.

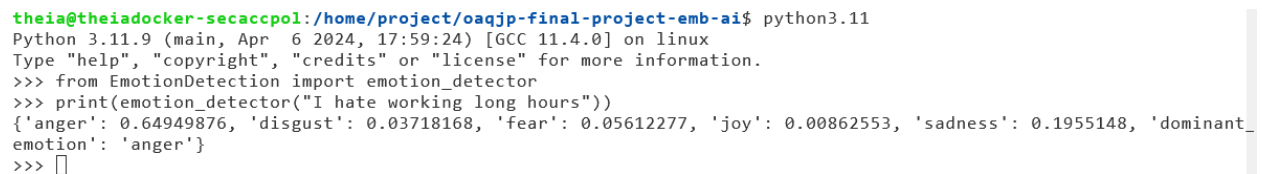
Створення пакету

4a_packaging.png



Тестування пакету

4b_packaging_test.png



5.

Юніт-тести

5a_unit_testing.png

```
1  from EmotionDetection.emotion_detection import emotion_detector
2  import unittest
3
4  class TestEmotionDetection(unittest.TestCase):
5      def test_emotion_detection(self):
6          test_cases = [
7              ("I am glad this happened", 'joy'),
8              ("I am really mad about this", 'anger'),
9              ("I feel disgusted just hearing about this", 'disgust'),
10             ("I am so sad about this", 'sadness'),
11             ("I am really afraid that this will happen", 'fear')
12         ]
13
14         for text, expected_emotion in test_cases:
15             with self.subTest(text=text):
16                 result = emotion_detector(text)
17                 self.assertEqual(result['dominant_emotion'], expected_emotion)
18
19 if __name__ == '__main__':
20     unittest.main()
```

Перевірка юніт-тестами

5b_unit_testing_result.png

```
theia@theiadocker-secaccp01:/home/project/oaqjp-final-project-emb-ai$ python3.11 test_emotion_detection.py
.
-----
Ran 1 test in 0.567s

OK
theia@theiadocker-secaccp01:/home/project/oaqjp-final-project-emb-ai$
```

6.

Деплой застоснки через Flask

6a_server.png

```
1 from flask import Flask, render_template, request
2 from EmotionDetection import emotion_detector
3
4 app = Flask(__name__)
5
6 @app.route('/emotionDetector', methods=['GET'])
7 def detect_emotion():
8     """
9     This endpoint receives a text input via query parameters,
10    processes it using the emotion_detector function, and returns the detected emotions.
11    """
12    text_to_analyze = request.args.get('textToAnalyze')
13    response = emotion_detector(text_to_analyze)
14
15    if not text_to_analyze or response.get('dominant_emotion') is None:
16        return "Invalid text! Please provide a valid input."
17
18    emotions = ", ".join([f'{emotion}': {score} for emotion, score in response.items() if emotion != 'dominant_emotion'])
19    dominant_emotion = response['dominant_emotion']
20
21    return f"For the given statement, the system detected the following emotions: {emotions}. The dominant emotion is {dominant_emotion}."
22
23 @app.route('/')
24 def index():
25     """
26     This endpoint renders the main HTML page.
27     """
28     return render_template('index.html')
29
30 if __name__ == "__main__":
31     app.run(host='0.0.0.0', port=5000)
```

Запуск додатку і перевірка роботи

6b_deployment_test.png

< > ↻ <https://secaccpol-5000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/>

Please enter the text to be analyzed

I think I am having fun

Run Sentiment Analysis

Result of Emotion Detection

For the given statement, the system detected the following emotions: 'anger': 0.029103195, 'disgust': 0.0067921067, 'fear': 0.027528232, 'joy': 0.876574, 'sadness': 0.06151191. The dominant emotion is joy.

7.

Обробка виключних випадків в функції

7a_error_handling_function.png

```
4 def emotion_detector(text_to_analyze):
5     url = 'https://sn-watson-emotion.labs.skills.network/v1/watson.runtime.nlp.v1/NlpService/EmotionPredict'
6     headers = {'grpc-metadata-mm-model-id': 'emotion_aggregated-workflow_lang_en_stock'}
7     input_json = { "raw_document": { "text": text_to_analyze } }
8
9     response = requests.post(url, json = input_json, headers=headers)
10
11     if response.status_code == 400:
12         return {
13             'anger': None,
14             'disgust': None,
15             'fear': None,
16             'joy': None,
17             'sadness': None,
18             'dominant_emotion': None,
19         }
20
21
22     response_data = json.loads(response.text)
23     emotions = response_data['emotionPredictions'][0]['emotion']
24     dominant_emotion = max(emotions, key=emotions.get)
25
26     result = {
27         'anger': emotions['anger'],
28         'disgust': emotions['disgust'],
29         'fear': emotions['fear'],
30         'joy': emotions['joy'],
31         'sadness': emotions['sadness'],
32         'dominant_emotion': dominant_emotion
33     }
34
35     return result
```

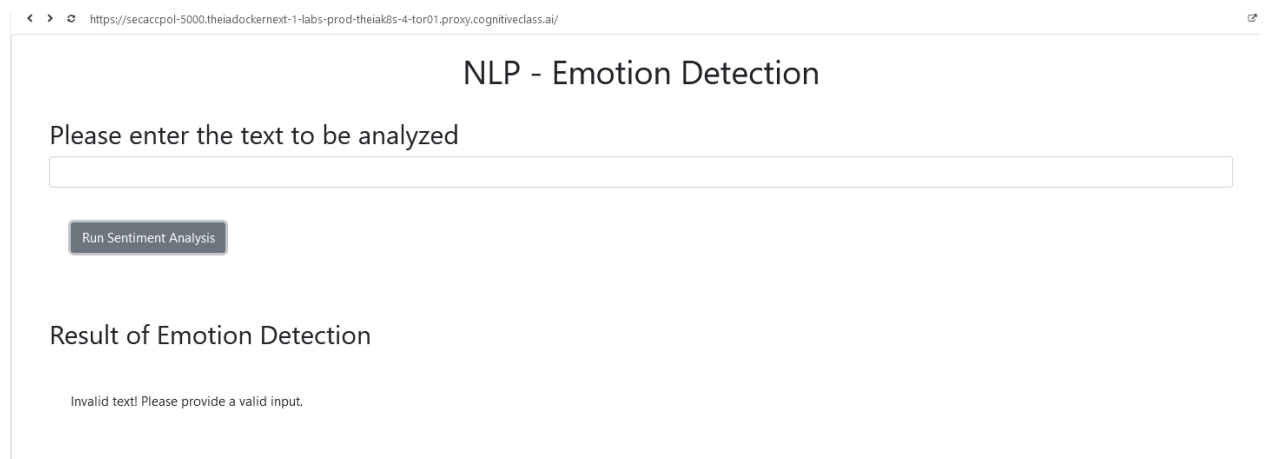
Обробка виключних випадків в сервері

7b_error_handling_server.png

```
1 from flask import Flask, render_template, request
2 from EmotionDetection import emotion_detector
3
4 app = Flask(__name__)
5
6 @app.route('/emotionDetector', methods=['GET'])
7 def detect_emotion():
8     """
9     This endpoint receives a text input via query parameters,
10     processes it using the emotion_detector function, and returns the detected emotions.
11     """
12     text_to_analyze = request.args.get('textToAnalyze')
13
14     if not text_to_analyze:
15         return "Invalid text! Please provide a valid input."
16
17     response = emotion_detector(text_to_analyze)
18     dominant_emotion = response.get('dominant_emotion')
19
20     if dominant_emotion is None:
21         return "Invalid text! Please provide a valid input."
22
23     emotions = ", ".join([f'{emotion}: {score}' for emotion, score in response.items() if emotion != 'dominant_emotion'])
24     result = (f'For the given statement, the system detected the following emotions: {emotions}. '
25             f'The dominant emotion is {dominant_emotion}.')
26
27     return result
28
29 @app.route('/')
30 def index():
31     """
32     This endpoint renders the main HTML page.
33     """
34     return render_template('index.html')
35
36 if __name__ == "__main__":
37     app.run(host='0.0.0.0', port=5000)
38
39
40
```

Перевірка обробки виключень(пустого вводу)

7c_error_handling_interface.png



8.

Стилістична модифікація коду

8a_server_modified.png

```
-
6  from flask import Flask, render_template, request
7  from EmotionDetection import emotion_detector
8
9  app = Flask(__name__)
10
11 @app.route('/emotionDetector', methods=['GET'])
12 def detect_emotion():
13     """
14     Endpoint to detect emotions from a given text.
15
16     Receives a text input via query parameters, processes it using the
17     emotion_detector function, and returns the detected emotions.
18
19     Returns:
20     | str: A formatted string containing the emotions and their scores,
21     | or an error message if the input is invalid.
22     """
23     text_to_analyze = request.args.get('textToAnalyze')
24
25     if not text_to_analyze:
26         return "Invalid text! Please provide a valid input."
27
28     response = emotion_detector(text_to_analyze)
29     dominant_emotion = response.get('dominant_emotion')
30
31     if dominant_emotion is None:
32         return "Invalid text! Please provide a valid input."
33
34     emotions = [
35         f"{emotion}: {score}"
36         for emotion, score in response.items()
37         if emotion != 'dominant_emotion'
38     ]
39     result = {
40         f"For the given statement, the system detected the following emotions: "
41         f"{', '.join(emotions)}. The dominant emotion is {dominant_emotion}."
42     }
43
44     return result
45
46 @app.route('/')
47 def index():
48     """
49     Renders the main HTML page.
50
51     Returns:
52     | str: The rendered HTML content for the index page.
53     """
54     return render_template('index.html')
55
56 if __name__ == "__main__":
57     app.run(host='0.0.0.0', port=5000)
```

Перевірка через pylint

8b_static_code_analysis.png

```
theia@theiadocker-secacpol: /home/project/oaqjp-final-project-emb-ai X
theia@theiadocker-secacpol:/home/project$ ls
oaqjp-final-project-emb-ai
theia@theiadocker-secacpol:/home/project$ cd oaqjp-final-project-emb-ai
theia@theiadocker-secacpol:/home/project/oaqjp-final-project-emb-ai$ pylint server.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
theia@theiadocker-secacpol:/home/project/oaqjp-final-project-emb-ai$ █
```

Висновок: в ході цієї роботи ми створили застосунок із використанням фреймворку Flask.

А також:

Обробляли HTTP/HTTPS запити, рендерили шаблони, деплоїли, обробляли виключні ситуації.

Отримали досвід роботи із бібліотеками json, request, Embeddable Watson AI

Навчились робити кращим стиль коду, використовуючи PyLint