|  |
| --- |
| **파이썬프로그래밍 및 실습**  **Emotional**  **therapist**  **진척 보고서 #2** |

|  |  |
| --- | --- |
|  | 제출일자:12월 10일  제출자명:김세희  제출자학번:233973 |

**1. Introduction**

**1) Background (**

In these times of grayness, many people live without understanding their own emotions and feelings. If they continue without realizing their emotions, it could lead to significant issues for themselves in the future. To address this, I believe an emotional management program is necessary

**2) Project goal**

Analyzing the user's diary to rank emotions Aim to extract the first and second highest emotions and suggest solutions

**3) Differences from existing programs**

There is a difference from existing programs when it comes to ranking and recommending solutions to users rather than abstractly picking emotions by looking at the diary

**2. Function plan**

**1) Retrieve User Diary Functionality**

- Description: After receiving diary entries from the user for each day of the week (Monday through Sunday), the "load\_diary\_from\_file" function is used to retrieve the diary text. Subsequently, the "read\_diary\_word" function is employed to separate the text into words and return them as a list.

**2)** **사용자 일기 주간 감정 그래프를 시각화 및 파일로 출력하는 기능**

- Description: The user's weekly sentiment score is visualized through a graph and output to the program. You can also save the results by printing out a graph as an image file (png).

**3) Providing solutions through weekly user sentiment analysis**

- Description: Ranking the user's sentiment and returning it to the dictionary to provide a solution based on the overall sentiment score.

**4) User diary writing feature[added]**

- Description: Ability to write and save a diary within the program instead of importing files

**5) Playing music**

- Description: Plays a song so that the user can relax while using the program.

**3. Progress**

**1) Feature Implementation**

**(1) Ability to import user diary**

Input: Enter the file name of the user diary

Output: If the user file is not found, use the -> "Error: {filename} could not be found. Exit the program." Output

Description: The 'diaries' list gives the user a diary from Monday to Sunday via loops. In addition, after receiving the diary, it returns the diary text through the 'load\_diary\_from\_file' function. The 'read\_diary\_word' function then returns the words to each list.

Applied Lessons Learned

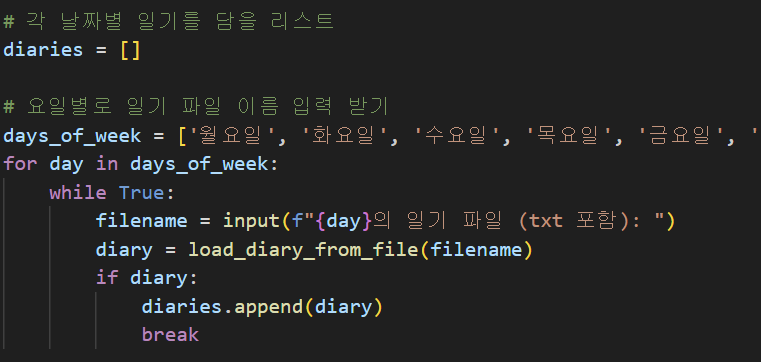
- functions: 'load\_diary\_from\_file' , 'read\_diary\_word'

- try, except: executes the code if the file is read via try, and exits the program with except if the file fails to read.

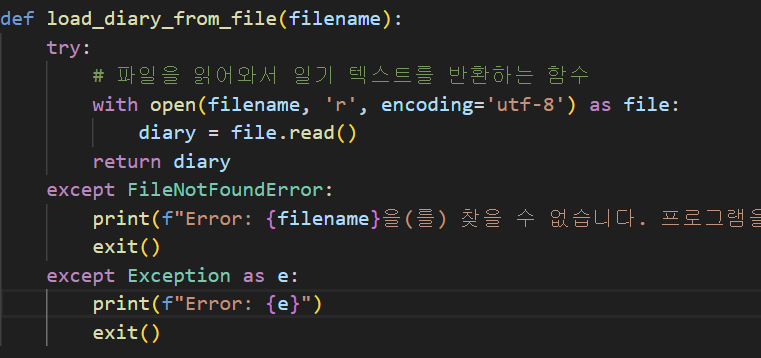
- split: Split to break up the text of the diary into words and return it to a list.

- Code Screenshot

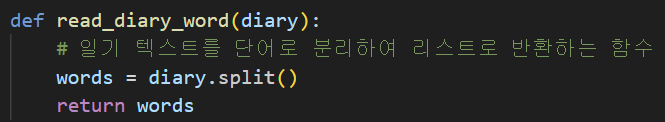
Code to receive diary entry from the user



A function that reads a file and returns it as text



Function to return diary text as a list



**(2) Ability to visualize user diary weekly sentiment graphs and output them to a file**

Type: Full Weekly Sentiment Analysis List 'weekly\_emotions'

output: graph image file

Description: Visualize the user's weekly sentiment analysis list through the 'plot\_weekly\_sentiment\_scores' function and save the graph as an image file by calling 'save\_graph'.

Applied Lessons Learned

* Functions: 'plot\_weekly\_sentiment\_scores', 'save\_gragh'

- Dictionary, List: List weekly\_emotion dictionary by extracting only the key value

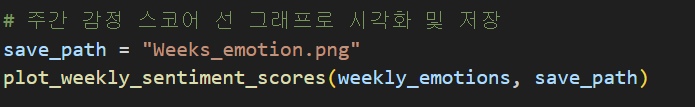
- Graph: Titile: Weekly sentiment change, X: Day of the week, Y: Score

- PLT: Outputs the resulting graph via pit to the program screen.

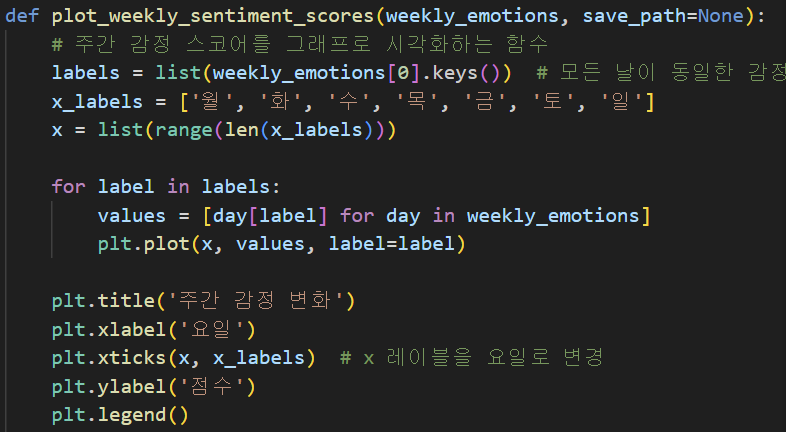
- Try, except: Outputs an error if the image file is not saved successfully.

Code Screenshot

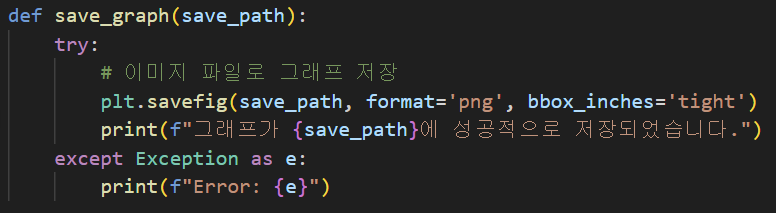
Function call code



A function that visualizes the weekly sentiment score as a graph



Function to save an image file



**(3) Providing solutions through weekly user sentiment analysis**

Input: A list of diaries received from the user

Output: Provide a solution to negative emotions

Description: The diary list received from the user is ranked by analyzing the emotions through machine learning. Provide a solution if negative emotions are high.

applied lessons

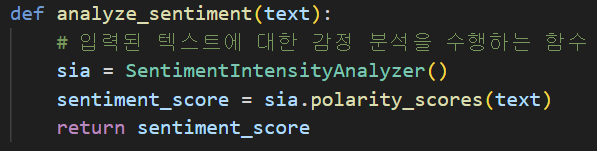
* Loop: Perform a full weekly sentiment analysis with a for in loop.
* functions: ‘analyze\_sentiment’ , ‘rank\_emotions’, ‘provide\_solution’,

‘provide\_negative\_solution’

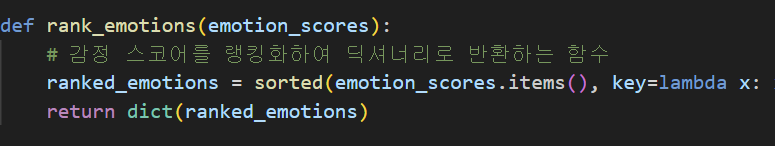
* Dictionary
* Conditionals: Provide a negative solution if the negative emotion is above a certain threshold
* Sorted: Rank the sentiment score through the sorted function.
* Machine Learning: Conduct sentiment analysis machine learning through NLTK.

Code Screenshot

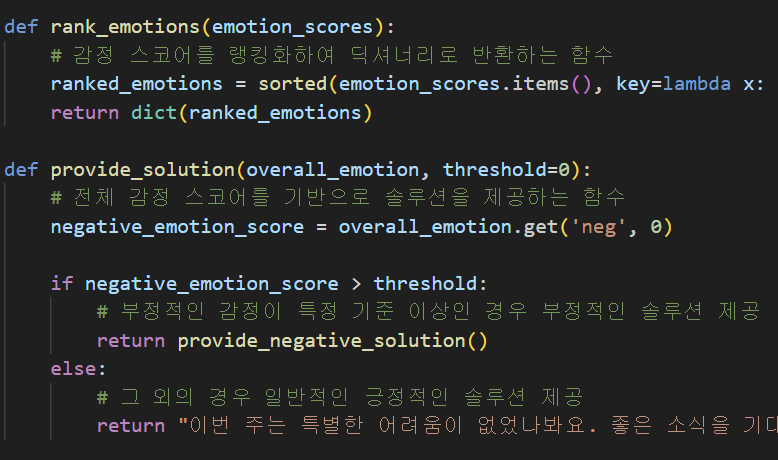
Functions that perform sentiment analysis on input text



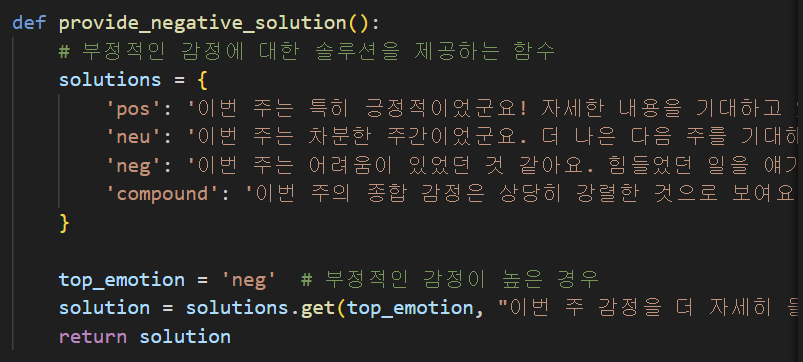
A function that ranks the sentiment score and returns it to the dictionary



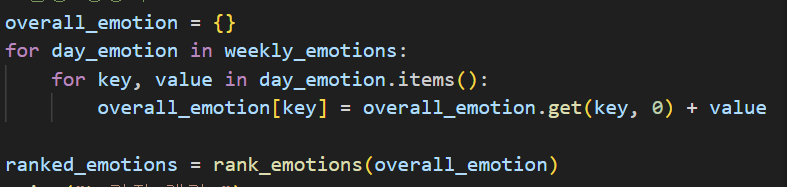
A function that provides a solution based on the overall sentiment scorer



Functions that provide solutions to negative emotions



Calling the function after sentiment ranking

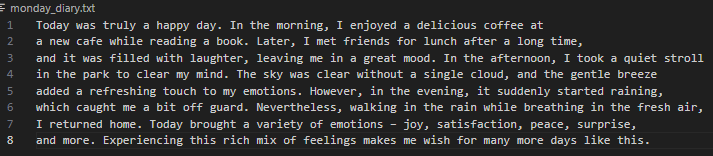


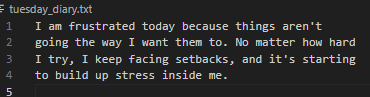
**2) Test Results**

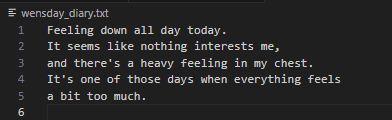
**(1) Ability to import user diary**

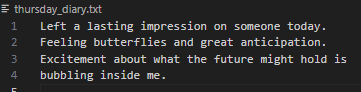
- Description: As an example of a user diary, the result is 7 inputs and the output to a dictionary appraisal list.

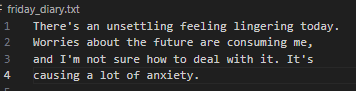
7 user diary examples

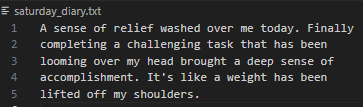
****

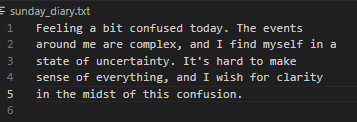




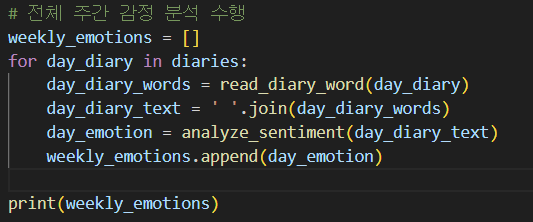


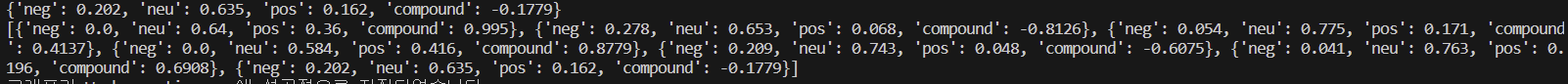






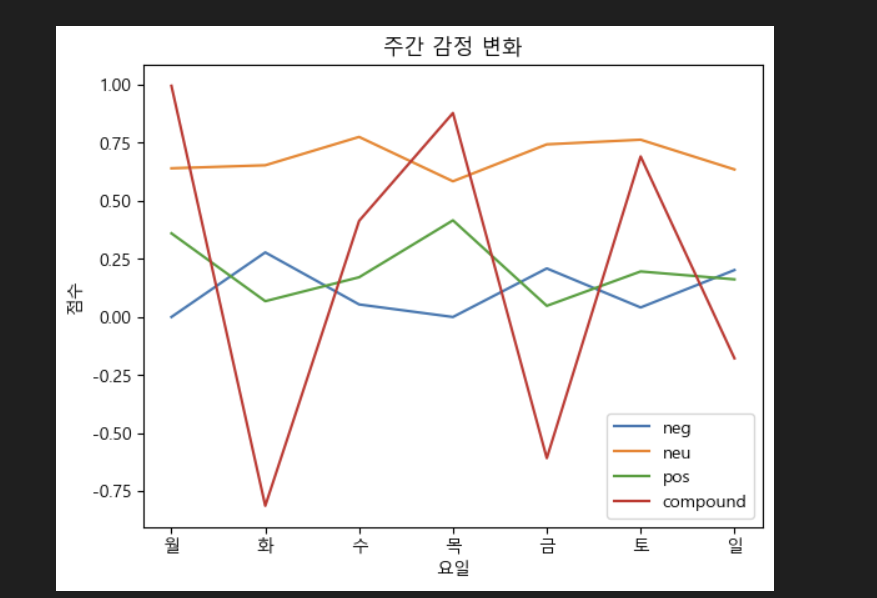
weekly\_emotions Output list result values



****

**(2)** **Ability to visualize user diary weekly sentiment graphs and output them to a file**

Files saved as images

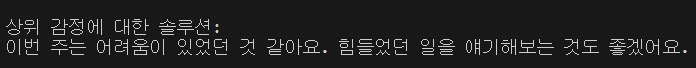


**(3)** **Providing solutions through weekly user sentiment analysis**

Sentiment Ranking



Top Sentiment Solutions



**4. Changes compared to the plan**

**- None**

**5. Project Timeline**

Submit a report

* November 3 Proposal completed
* November 30 Progress Report #1 submitted
* December 10 Progress Report #2 submitted

User diary import function

- Functional completion on December 10

Providing solutions through weekly user sentiment analysis

- Completed on December 10

- Add and refine solutions by December 2

User diary function

- Scheduled to be produced from December 13th ~ December 21st

Playing music

- Scheduled to be produced from December 18th ~ December 19th

something

- Function modularization until December 21

- Organize and simplify programs