Python Programming and Practice

Gift Recommending System on MBTI Types

Progress Report : 1

Date : 2023. 11. 24

Name : Minchae choi

ID : 223479

1. Introduction

1) Background

Recently, the MBTI personality type test has gained popularity and is being utilized in various fields. Additionally, there is a growing trend in the use of online gift-giving features.

2) Project goal

By recommending gifts based on MBTI personality types, the program offers convenience to its users, fulfills practicality for gift recipients, and additionally brings advertising benefits to the suppliers.

3) Differences from existing programs

The current gift recommendation system relies solely on preferences based on age and gender, which may not always align with the actual preferences of individuals. To address this limitation, a system has been developed to recommend gifts based on MBTI personality types. Furthermore, users can input their MBTI and preferred gifts to gather data, making the system more tailored to individual needs and preferences.

2. Functional Requirement

1) Gift recommendations based on MBTI types & category

- At first, the user inputs the gift category. Depending on the entered category, extract the most frequent item from the preferred gifts data files (4 files of each MBTI position E/I, S/N, F/T, P/J). This item becomes the recommended gift.

2) Enter favorite gift and MBTI type

- When a user inputs their preferred gift and their own MBTI into the program, it is saved in the preferred gift data list. This helps increase the amount of data, thus improving the program's performance.

3) Providing a gift card message

- Based on the category of the gift, the program randomly generates a gift card message. This function provides convenience to the user.

3. Progress

1) Functional Requirements

(1) Frame of the program

- Output a list of available functions, and user input the desired function to select.

- I applied Condition, Loop and Function on this code.

```
def opt():
    while True:
        print("choose option")
        print("1. Get gift recommendation")
        print("2. Enter the desired gift")
        # add 3rd function "getting a gift card message for random"
        print("3. Get a gift card message")
        print("4. Quit")
        option = int(input("enter the number of your option --> "))

        if option == 1:
            output_gift()
        elif option == 2:
            input_gift()
        elif option == 3:
            output_msg()
        elif option == 4:
            break
        else:
            print("Wrong input. Please input 1~4.")
```

2) Test Result

(1) Frame of the program

- If users choose options 1 to 3, the corresponding functions will be executed. Choosing option 4 is designed to exit the program. If numbers other than 1 to 4 is pressed, it will prompt for input again.

```
choose option
1. Get gift recommendation
2. Enter the desired gift
3. Get a gift card message
4. Quit
enter the number of your option --> 1
choose option
1. Get gift recommendation
2. Enter the desired gift
3. Get a gift card message
4. Quit
enter the number of your option --> 5
Wrong input. Please input 1~4.
choose option
1. Get gift recommendation
2. Enter the desired gift
3. Get a gift card message
4. Quit
enter the number of your option --> 4
```

4. Changes in Comparison to the Plan

1) Categorization of data files

- I originally planned to have simple, unstructured data files without categorization.

- Categorize gift lists in data files.

- By categorizing, it becomes possible to narrow down the scope and make somewhat personalized gift recommendations based on preferences.

2) Add Function 3 (Providing a gift card message)

- I originally planned only 2 functions.

- Add function 3. Providing a gift card message.

- By recommending gift card messages, it provides convenience to the user.

5. Schedule

- 진행 상황 표기

업무	11/3	11/10	11/17	11/24	12/1	12/8	12/15	12/22
Proposal	finish							
data & Frame		finish						
function 1				progressing				
function 2					---------->			
Function 3						---------->		
Code inspection							---------->	
Final report							---------->	