파일 이름은 진척보고서 1

Python programming and practice

Food Recipe Recommendations Based on User Preferences or Ingredients

Progress Report : 1

Date : 2023/11/26
Name : Seo Jun
ID :230750

1. Introduction (16 pt)

1) Background (14 pt)

Recipe recommendation systems make it easy for users to find information about cooking, save time compared to internet searches, and provide practical cooking ideas. Interest in cooking and food-related activities is on the rise, leading to an increased demand for recipe recommendation services.

2) Project goal

Goal is to create a program that recommends recipes based on customer preferences and available ingredients.

3) Differences from existing programs

There is an app called '만개의 레시피' that provides recipes for free and includes links to purchase cooking ingredients. Users can also upload their own recipes. However, my program differentiates itself by recommending recipes based on user preferences and available ingredients.

2. Functional Requirement

1) Collecting Information from Users

- Feature to Input Preferred Foods or Ingredients.

(1) Collecting Food and Ingredient Information by Creating an Arbitrary Text File

- Ability to add, delete, modify, and view ingredients in the text file.

2) Comparing Recipe Files with User Profiles

- Function to calculate similarity for each file

(1) Preprocessing each sentence, creating a token set to calculate similarity, and storing similarity scores in a dictionary.

- Sorting the similarity dictionary in descending order and ranking recipes based on similarity.

3) Extracting Recipes with High Similarity

- Implemented as a function to extract the top 3 recipes with high similarity and present them to the user.

(1) If the user is not satisfied, extract the next 3 in the ranked list.

- Access the next index in the list sorted by similarity after excluding the top 3.

3. Progress

1) Function Implementation

(1) Collecting Information from Users

-Input/Output

add_ingredient(ingredient_list, ingredient)

Input: ingredient_list - a list containing existing ingredients

ingredient - the name of the ingredient to be added

Output: None

display_ingredient(ingredient_list)

Input: ingredient_list - a list containing the current ingredients

Output: None

save_ingredients_to_file(ingredient_list, filename)

Input: ingredient_list - a list containing ingredients to be saved

filename - the name of the file to save the ingredients

Output: None

load_ingredients_from_file(filename)

Input: filename - the name of the file to be read

Output: a list containing the loaded ingredients

-Description

IngredientModule class:

A class containing functions for handling ingredient lists.

add_ingredient(ingredient_list, ingredient) function:

Adds a new ingredient to the list of ingredients.

display_ingredient(ingredient_list) function:

Displays the current list of ingredients.

save_ingredients_to_file(ingredient_list, filename) function:

Saves the list of ingredients to a file.

load_ingredients_from_file(filename) function:

Reads the list of ingredients from a file.

-Applied Concepts (e.g., loops, conditional statements, functions, classes, modules)

Modules, classes, functions, input/output, exception handling, loops, conditional statements, file input/output (File I/O).

-Code Snapshot

```python
class ingredient_module:
    #재료 추가하는 함수 생성
    def add_ingredient(ingredient_list, ingredient):
        ingredient_list.append(ingredient)
    #재료 보여주는 함수 생성
    def display_ingredient(ingredient_list):
        if not ingredient_list:
            print("재료 목록이 비어 있습니다.")
        else:
            print("재료 목록:")
            for i in range(len(ingredient_list)):
                ingredient = ingredient_list[i]
                print(f"{i+1}. {ingredient}")
    #재료를 파일에 저장하는 함수 생성
    def save_ingredients_to_file(ingredient_list, filename):
        write_fp = open(filename, 'w', encoding="utf8")
        for ingredient in ingredient_list:
            write_fp.write(ingredient+"\n")
        write_fp.close()
        print("재료가 파일에 저장되었습니다.")
```

```python
#재료를 읽어오는 함수
def load_ingredients_from_file(filename):
    try:
        ingredient_list = []
        read_fp = open(filename,'r', encoding="utf8")
        lines = read_fp.readlines()
        for line in lines:
            ingredient_list.append(line.strip())
        read_fp.close()
        return ingredient_list
    except FileNotFoundError:
        print("파일을 찾을 수 없습니다.")
```

```python
from ingredient_module import add_ingredient, display_ingredient, save_ingredients_to_file, load_ingredients_from_file

try:
    file_name ="ingredient.txt"
    ingredient_list = load_ingredients_from_file(file_name)
    while True:
        print("\n레시피 추천 프로그램")
        print("1. 재료 추가")
        print("2. 재료 목록 표시")
        print("3. 종료")
        choice = input("선택: ")
        if choice == '1':
            ingredient = input("재료를 입력하세요: ")
            add_ingredient(ingredient_list, ingredient)
            print("재료가 추가되었습니다.")
        elif choice == '2':
            display_ingredient(ingredient_list)
        elif choice == '3':
            save_ingredients_to_file(ingredient_list, file_name)
            print("프로그램을 종료합니다.")
            break
        else:
            print("잘못된 선택입니다. 다시 시도하세요.")
except FileNotFoundError:
    print("파일을 찾을 수 없습니다.")
```

2) Test Result

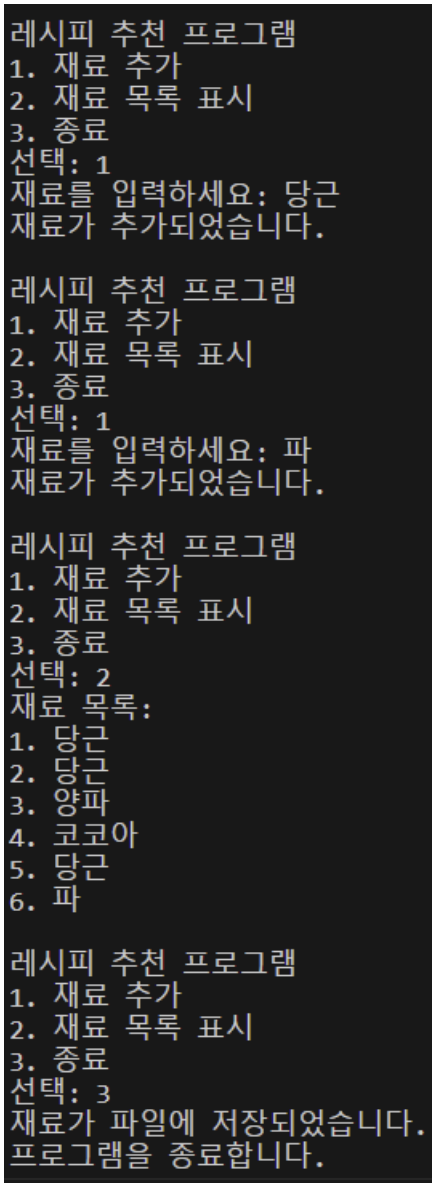
(1) Collecting Information from Users

- Description

I tested inputting, saving to a file, and program termination in the file.

-Test Results Screenshots

The test revealed an issue where the module was not being called due to a conflict between the class and the name. After renaming the class, I successfully called the module.



4. Changes in Comparison to the Plan

Title of the Change

-Before

Output remains the same even if the user knows the recipe.

-After

If the user knows the recipe, prompt for input and output a

different recipe.

-Reason

Initially, individual cooking knowledge was not considered. However, based on feedback that chef's recipes are too common, this modification is made to address the issue.

5. Schedule

업무		11/3	11/10	11/30	12/5	12/15
Create a proposal		Completion				
Collecting Information from Users	Collecting Food and Ingredient Information by Creating an Arbitrary Text File		---ongoing --->			
Comparing Recipe Files with User Profiles	Preprocessing each sentence, creating a token set to calculate similarity, and storing similarity scores in a dictionary.			---ongoing --->		

Extracting Recipes with High Similarity	**If the user is not satisfied, extract the next 3 in the ranked list.**			**---------->**
TEST				**------->**