Food data generator

Description This simulation tool aims to generate synthetic data representing the interactions between a group of users and a food recommendation system during a parameterized number of days. The library also has an interface for Jupyter notebooks that facilitates its use and the variation of parameters to obtain different results.

Table of contents:

- Requirements
- Parameters
- Code Organization
- Documentation
- Execution
- Develop

Requirements

The required python version is **Python version**: 3.9. And the following libraries are required to execute the generation script:

- numpy
- pandas
- faker
- openai
- json
- graphviz
- ipywidgets

Parameters

To run the simulation several parameters can be setter trough the GUI in the main notebook or directly in the default_inputs.py file. The parameters that can be set are described bellow:

- **Age**: This parameter sets the probability distribution for different age ranges. This probability define the user age range distribution. The values in this parameter should sum up 1.
- **Gender**: This parameter define the distribution for users between the clinical genders male (M) and female (F). The values in this parameter should sum up 1.
- **BMI**: This parameter defines the probability distributions for users within the 4 Body Mass Index (BMI) conditions (underweight, healthy, overweight, obesity). The values in this parameter should sum up 1.
- **Allergies**: This parameter defines the probability distribution to assign users under different allergy conditions. The values in this parameter should sum up 1.
- **Food restrictions**: This parameter defines the probability distribution to assign users under different cultural restrictions (e.g., vegan, vegetarian, halal, kosher). The values in this parameter should sum up 1.

- **Meal probabilities**: This parameter determine the percentages of users that could take a meal or not (e.g., breakfast with 0.8 probability will be present in the 80% of the users). The values in this parameter are independent and each one can vary between 0 and 1.
- **Flexible probability**: This parameter defines the user's soft-preferences (e.g., a person could have a preference for vegan food but also can take vegetarian food also).
- **BMI transition probability**: This probabilities define the transition probability between BMI states, this probabilities define if the user change or maintain they current BMI.

Code Organization

Code is organized in the following setup:

- The source code is stored in the **src**. In the src folder the modules are grouped according their function.
- The documentation is

Documentation

Documentation has been automatic generated with Sphinx from inline python documentation in files and is stored in **docs** folder.

Execution

To execute the notebook is required a python environment that can be created with spec-file.txt.example using the following command in a OS-x machine:

```
conda create --name <env> --file <this file>
```

Where `<env>` should be replaced by the name of the environment and `<this file>` should be replaced by the spec-file.txt.example. Once the environment has been created successfully it should be activated with the following command:

```
conda activate <env>
```

where `<env>` should be replaced by the environment name. With the environment activated please execute the following installation commands:

```
sudo apt-get install graphviz
pip install graphviz
pip install faker
```

Additionally to the method above mentioned the environment can be create with different tools like venv the only requirement is to ensure that the Python version used is 3.9, once the environment is created you

must install the libraries defined in the requirements.txt.example file using the following command with the environment enabled:

```
pip install -r requirements.txt.example
```

Once the environment has been fully configured and activated please run the notebook server executing in the terminal with the environment activated the following command:

```
jupyter notebook
```

Open the browser in the address indicated in the terminal and open the notebook main_generator_notebook.ipynb in the src folder.

Develop

To contribute in the development please make a fork or create a new branch of this repository, develop your changes and test it. Once your changes has been tested and documented ask for a pull request that should be approved by repository administrators.