

## Enhancements to expl-types.py module in explanation\_food\_recsys\_2:

To execute the explanation module, it is necessary to start the two servers using the following commands:

- `python food_rs_webservice.py`
- `python web_expl.py`

Once the servers are running, you can send requests in the following format:

```
python foodWebApp/cmd_expl.py --type 1 --style 1 --mood good --stress no --depression no --bmi over --activity high --goal gain --sleep good --restr low-nickel --imgurl1 https%3A%2F%2Fwww.giallozafferano.it%2Fimages%2Fricette%2F209%2F20923%2Ffoto_hd%2Fhd650x433_wm.jpg --imgurl2 https%3A%2F%2Fwww.giallozafferano.it%2Fimages%2Fricette%2F157%2F15788%2Ffoto_hd%2Fhd650x433_wm.jpg --difficulty 5 --user_time 40 --user_cost 4 --user_ingredients spinach --user_age U50 --season autumn --sex m
```

```
C:\Users\gianf\Desktop\Uni\SIITA\explanations_food_recsys_2>python foodWebApp/cmd_expl.py --type 1 --style 1 --mood good --stress no --depression no --bmi over --activity high --goal gain --sleep good --restr low-nickel --imgurl1 https%3A%2F%2Fwww.giallozafferano.it%2Fimages%2Fricette%2F209%2F20923%2Ffoto_hd%2Fhd650x433_wm.jpg --imgurl2 https%3A%2F%2Fwww.giallozafferano.it%2Fimages%2Fricette%2F157%2F15788%2Ffoto_hd%2Fhd650x433_wm.jpg --difficulty 5 --user_time 40 --user_cost 4 --user_ingredients spinach --user_age U50 --season autumn --sex m
{"explanations": {"foodGoals_two": "Ricotta doughnut with honey and pine nuts has 387.0 Kcal. It is a good choice, since you are aiming to gain weight, this recipe represents 14.07% of your daily intake. Many users with similar goals and activity levels have enjoyed this recipe. Lime vermicelli and pink pepper has 730.0 Kcal. It is a good choice, since you are aiming to gain weight. The average daily calorie intake for a man with your goal and type of activity is 2750 Kcal and this recipe represents 26.55% of your daily intake. Many users with similar goals and activity levels have enjoyed this recipe. Anyway, between the two, Lime vermicelli and pink pepper seems more suitable for you since it has more calories."}}
```

Several improvements have been made to the explanation module, including:

- **Minor grammatical enhancements**, such as removing "any" and "all" in the restrictions explanation and eliminating repetitions about calories.
- Improved explicit comparison in comparative explanations, providing users with a clearer preference between recipes, as illustrated in the example above. In the previous version of the project, the users were left with just a list of facts, but **no real suggestions were given**, like in this example:

```
C:\Users\gianf\Desktop\Uni\SIITA\explanations_food_recsys_2>python foodWebApp/cmd_expl.py --type 5 --style 1 --mood good --stress no --depression no --bmi normal --activity high --goal gain --sleep good --imgurl1 https://www.giallozafferano.it/images/ricette/7/759/foto_hd/hd650x433_wm.jpg --imgurl2 https://www.giallozafferano.it/images/ricette/4/443/foto_hd/hd650x433_wm.jpg --sex m
{"explanations": {"FoodFeatureHealthRisk_two": "Yeastdough braid has a greater amount of proteins (9.6 gr) than Pumpkin cream (3.9 gr). Yeastdough braid has a smaller amount of fat (8.8 gr) than Pumpkin cream (15.6 gr). Yeastdough braid has a smaller amount of fibers (1.5 gr) than Pumpkin cream (2.0 gr). Yeastdough braid has a greater amount of sugars (17.7 gr) than Pumpkin cream (6.4 gr). Yeastdough braid has a smaller amount of sodium (0.177 gr) than Pumpkin cream (1.314 gr). Yeastdough braid has a greater amount of carbohydrates (55.8 gr) than Pumpkin cream (16.2 gr). Yeastdough braid has a greater amount of saturated fat (4.47 gr) than Pumpkin cream (2.4 gr). Yeastdough braid's quantity of carbohydrates (55.8 g) is higher than Pumpkin cream's (16.2 g); a high intake of carbohydrates could increase the risk of heart diseases. Yeastdough braid's quantity of sugars (17.7 g) is higher than Pumpkin cream's (6.4 g); a high intake of sugars could increase the risk of tooth decay. Furthermore, Yeastdough braid has more calories (340.0 Kcal) than Pumpkin cream (221.0 Kcal)."} }
```

- **Removed explanations based on dopamine content** in ingredients, considering the sensitivity of the topic, especially when providing dietary advice to individuals dealing with depression.
- Implemented a different approach to the foodGoals explanation method. Now, the daily caloric intake is calculated based on the user's gender and goal ('lose', 'gain', or 'maintain'). The base intake is then adjusted with a caloric surplus based on the user's activity level. Each recipe is evaluated based on the calculated intake and the goal, both in single and comparative explanation modes. **(Previously the daily caloric intake was calculated just by the formula that a meal is about 40% of the daily intake)**

- Introduced a new explanation method: foodMacros.

This method analyzes the standard deviation of recipe macronutrients relative to an ideal ratio. The deviation is calculated based on the ideal macronutrient distribution adjusted for sex and activity:

<code>ideal_macros["carbs fat proteins"] / 100</code>	converts the percentage value of the ideal carbs distribution to a decimal.
<code>total_recipe_macros</code>	sum of the actual values of carbs, fats, and proteins in the recipe.
<code>total_recipe_macros * (ideal_macros["carbs fat proteins"] / 100)</code>	calculates the amount of carbs that would be present in the recipe if it perfectly matched the ideal distribution(%).
<code>recipe_macros["carbs fat proteins"] - (total_recipe_macros * (ideal_macros["carbs fat proteins"] / 100))</code>	This is the deviation formula: The intuition is to subtract the expected amount of carbs from the actual amount of carbs in the recipe.

The result is the deviation, representing how much the actual carbs content deviates from what is expected based on the ideal distribution. This new method has also been incorporated in the smartExplanation function.

## Telegram Bot Re-implementation - module FoodRecommendationSys.py

The modification of the expl-types module aimed to enhance the Telegram bot [https://t.me/Food\\_Food\\_Food\\_bot](https://t.me/Food_Food_Food_bot), which is the subject of Francesco Lopedota's thesis. Unfortunately, due to the deactivation of the associated DialogFlow account handling bot intents, the original bot could not be recovered.

Inherited the source code of the bot, including the Recommender Sys, the explanation system (old expl-types), and the Python code connecting the bot and DialogFlow (module FoodRecommenderSys.py), a functional Telegram bot was re-implemented at [https://t.me/food\\_recsys\\_bot](https://t.me/food_recsys_bot).

The re-implementation involved several changes, such as:

- User interaction translated entirely into English.
- Update of project requirements.  
The project initially **relied on the python-telegram-bot v13.4 library**, which was deprecated and required an upgrade to version 20.7.
- With the new version, the main function logic changed, **removing the dispatcher** and introducing the Application component that encapsulates essential elements for bot execution, such as updater, dispatcher etc.
- The new version and the component application led to the adoption of the async-await paradigm for handling responses asynchronously.  
All functions within the FoodRecommendationSys.py module have been modified to adhere to the asynchronous paradigm. Consequently, response management now occurs asynchronously, awaiting the resolution of message updates (chat-bot side) before concluding.
- The entire project is now executed locally.  
**Previously deployed on Heroku**, the project (as for DialogFlow agent) became inaccessible.  
Therefore, the entire project is now run locally using three commands:
  - `python food_rs_webservice.py` # To start the recommendation system server
  - `python web_expl.py` # To start the explanation server
  - `python FoodRecommenderSys.py` # To start the Telegram bot
- Created a new agent on a new DialogFlow account.  
Using Google Cloud, a service account was created to generate credentials for the Python connection between DialogFlow and the bot. **Old project credentials were removed for security reasons (they were pushed to GitHub repo)**, and new ones were implemented but added to the .gitignore file in order to keep them private.
- The new agent maintains basic chitchat capabilities and welcoming features but categorizes intents into five specific areas:

<u>Intent category</u>	<u>Description</u>	<u>Triggering messages</u>
Introduction	In addition to the default welcome intent, another intent has been created to provide general information about the bot and direct the user to the <code>/create</code> command to either create their profile or, in case they wish to modify it, guide them to the <code>/modify</code> command.	<ul style="list-style-type: none"><li>• hi</li><li>• hello</li><li>• who are you</li><li>• what do you do</li><li>• etc...</li></ul>
Suggestion	The suggestion intent initializes the recommendation procedure after the user creates a user profile. A request, composed of parameters that form the user profile, is sent to the recommendation server. The server processes the request and returns a recipe, which is then presented to the user via a message.	<ul style="list-style-type: none"><li>• Suggest something to eat</li><li>• I want a <code>*ingredient*</code>-based recipe</li><li>• Can you give me a recipe with <code>*ingredient*</code>?</li></ul>
Change Suggestion	The user has the option to make two changes to the recipe. If they do not like the proposed recipe or do not have suitable ingredients, they can simply ask the bot to provide another one, and the bot will respond with the second (or third) best recommendation for them.	<ul style="list-style-type: none"><li>• Another one</li><li>• i don't like this, do you have another one?</li><li>• can you change it?</li><li>• give me another one</li></ul>

	After the third recommendation, the bot will not provide further suggestions unless the user modifies their profile parameters.	
Explanation	<p>Upon receiving a recommendation, the user can inquire about the reasons behind the recommendation and its intrinsic characteristics.</p> <p>This will send a request to the web_expl server that returns the text of the explanation as message to the user.</p>	<ul style="list-style-type: none"> <li>• what are its nutritional values?</li> <li>• how popular is it?</li> <li>• what are its characteristics?</li> <li>• why did you suggest this?</li> <li>• is it easy to cook?</li> <li>• is it in line with my goals?</li> <li>• does it have some healthy benefits?</li> <li>• what are its health risks?</li> <li>• how much does it cost?</li> <li>• is it adequate for my age?</li> <li>• is it ok for my restrictions?</li> <li>• is it good for my lifestyle?</li> <li>• do I have enough time?</li> <li>• are its macros good?</li> <li>• is it sustainable?</li> <li>• is it seasonal?</li> </ul>
Comparison	<p>The users are also given the possibility to compare two different recipes, in order to get the best one for their preferences. This is only possible whenever the user has requested more than one recommendation (max 3).</p> <p>The bot, currently, only takes into account the latest two recommendations to make the comparisons (so if the user has requested 3 suggestions the comparisons will be between the second and the third one).</p> <p>This will send a request to the web_expl server that returns the text of the comparison as message to the user.</p>	<ul style="list-style-type: none"> <li>• compare them according to their nutritional values</li> <li>• compare them according to their popularity/rating</li> <li>• compare them according to their characteristics</li> <li>• compare them according to the skill required to cook them</li> <li>• compare them according to my goals</li> <li>• compare them according to their health benefits</li> <li>• compare them according to their health risks</li> <li>• compare them according to their costs</li> <li>• compare them according to my age</li> <li>• compare them according to my restrictions</li> <li>• compare them according to my lifestyle</li> <li>• compare them according to cooking time</li> <li>• compare them according to their macros</li> <li>• compare them according to their sustainability</li> <li>• compare them according to their seasonability</li> </ul>

### Intents list

• Change suggestion 1	• Explanation age
• Change suggestion 2	• Explanation check
• Comparison age	• Explanation cost
• Comparison cost	• Explanation goals
• Comparison goal	• Explanation health-benefit
• Comparison health-benefit	• Explanation health-risk
• Comparison health-risk	• Explanation lifestyle
• Comparison lifestyle	• Explanation macros
• Comparison macros	• Explanation meal
• Comparison meal checks	• Explanation popularity
• Comparison popularity	• Explanation restriction
• Comparison restriction	• Explanation seasonability
• Comparison seasonability	• Explanation skill
• Comparison skill	• Explanation smart
• Comparison smart	• Explanation sustainability
• Comparison sustainability	• Explanation time
• Comparison time	• Introduction ^
• Comparison two meals	• ↳ Introduction - create profile
• Default Fallback Intent	• ↳ Introduction - modify profile
• Default Welcome Intent	• Specific suggestion
	• Stop suggestions
	• Suggestion

- Original User-Profiling Modification:

The original User Profiling process has been streamlined, specifically **reducing the initial set of questions from a total of 22 to approximately 10** (the number varies based on the user's gender; if male, there will be no inquiries about pregnancy).

Responses to the first four questions are obtained through the user's open-ended input, where they need to type in the exact values they wish to input:

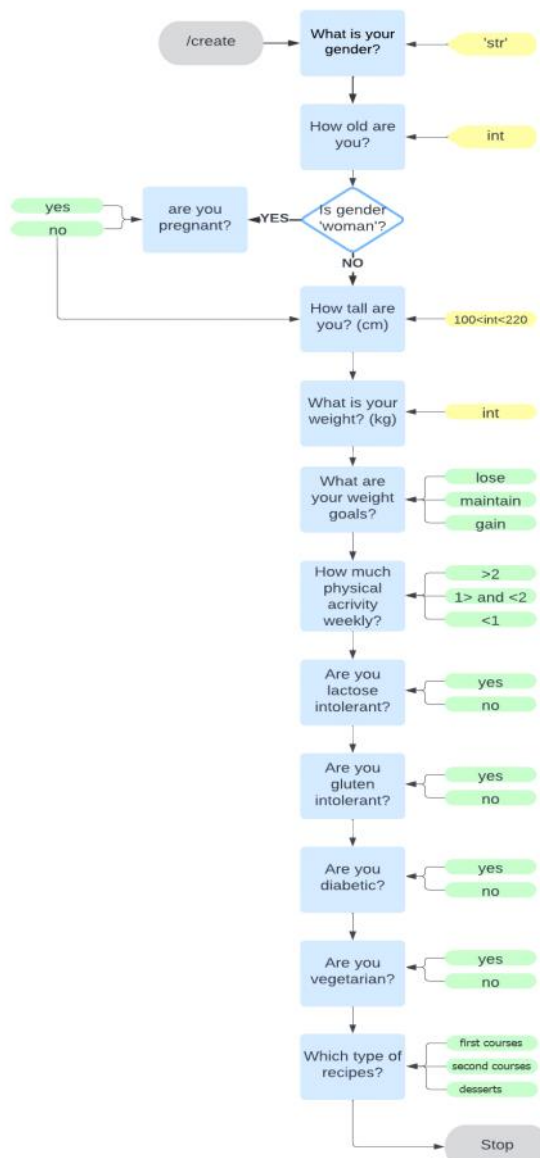
- What is your gender?
- How old are you?
  - *Only if female or unspecified: Are you pregnant? (response via buttons: yes/no)*
- How tall are you? (cm)
- What is your weight? (kg)

Subsequently, all remaining questions are answered through buttons:

- What are your goals regarding your weight? (lose/maintain/gain)
- How much physical activity do you practice weekly? (a lot, just enough, not so much)
- Are you LACTOSE intolerant? (yes/no)
- Are you GLUTEN intolerant? (yes/no)
- Are you DIABETIC? (yes/no)
- Are you vegetarian? (yes/no)
- Which type of recipes do you want me to suggest? (first courses/second courses/desserts)

The aim is to gather only the most fundamental information for the recommendation, minimizing user inconvenience. Values not set by the user are defaulted to their mean value.

This choice is communicated to the user, along with the **/modify** command, allowing them to view all profile values and modify them.



- **Addition of User-Profiling Modification:**

The previous project lacked a section for modifying user data. Therefore, the `/modify` command was implemented and managed through three functions and a specific conversational handler to give users the opportunity to modify their preferences:

- The `/modify` command triggers the `modify_profile` function, displaying the user's profile and asking if they want to modify a specific parameter, returning the next state of the conversational handler.
- In this new state, the conversational handler launches the `choose_attribute` function to let the user choose the parameter to modify. Depending on the user's choice, the bot will display buttons related to the values they can choose for that specific attribute.
- In the final state, the conversational handler launches the `change_attribute_value` function to effectively modify the chosen attribute's value with the user-inputted value and displays a confirmation message to the user.

- **Addition of User-specific Request:**

In the previous project there was no way for the user to express a preference for the ingredients of the recipe to suggest.

Now they can ask the bot for a suggestion based on an ingredient or type of dish they provide.

The bot will attempt to satisfy the request by searching through the top 15 suggestions (ranked).

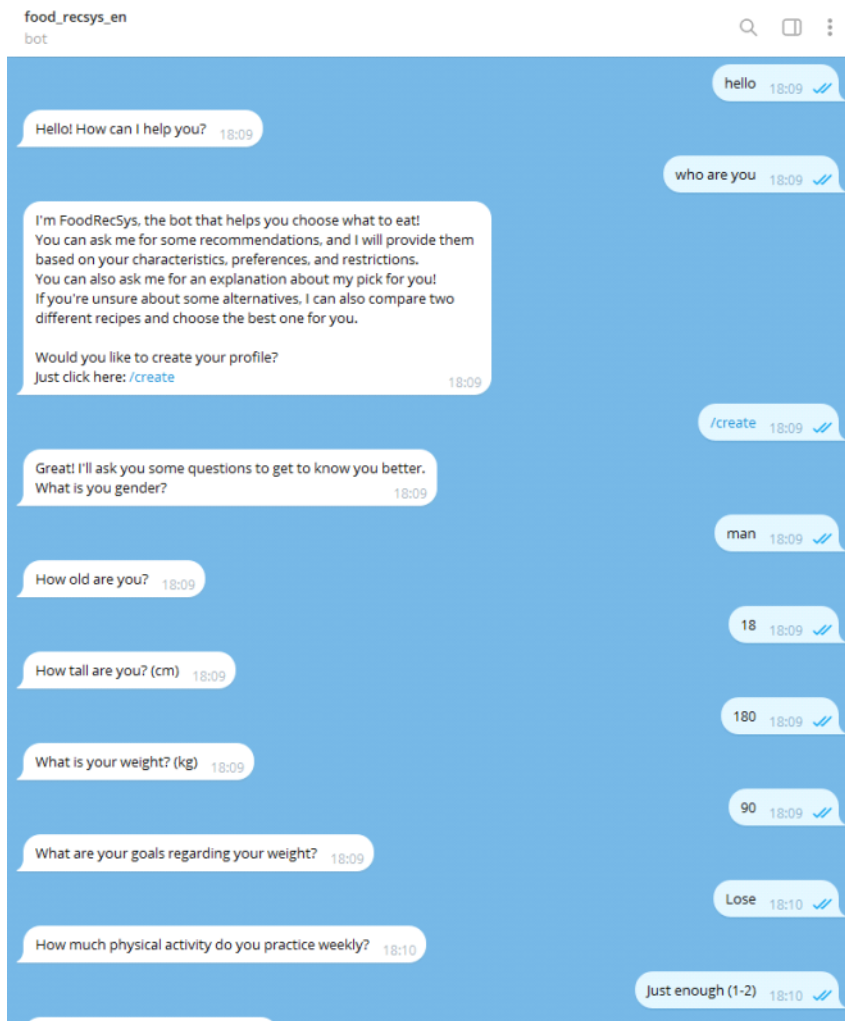
The request is first interpreted by identifying the dialogflow entity representing the ingredient/type, the ingredient/type is then isolated and transformed into lowercase and singular form through the library `inflect`.

The same goes for the ingredients of every recipe checked.

If the bot finds a match (a recipe containing the specified ingredient either in the title or in the ingredients), it will return it as a suggestion.

Otherwise, it will inform the user that based on the parameters chosen, there is no suitable recipe to suggest.

### Chat instance:



### USER PROFILING



How much physical activity do you practice weekly? 18:10

Just enough (1-2) 18:10 ✓

Are you LACTOSE intolerant? 18:10

No 18:10 ✓

Are you GLUTEN intolerant? 18:10

No 18:10 ✓

Are you DIABETIC? 18:10

No 18:10 ✓

Are you vegetarian? 18:10

Yes 18:10 ✓

Ok, we are done! Which type of recipes do you want me to suggest? 18:10

Second courses 18:10 ✓

Thank you for your time!  
I've assumed some other values for you (averaging our users), if you want to check your profile out click here: [/modify](#).  
Or just ask me something to eat! 18:10

/modify 18:10 ✓

Ok! This is your current profile.  
1 means TRUE.  
0 means FALSE.  
X/5 it's a scale where 1 is lowest and 5 highest.

- Category: second courses
- Low Nickel: 0
- Vegetarian: 1
- Lactose Free: 0
- Gluten Free: 0
- Diabetic: 0
- Pregnant: 0
- User Skill: 3/5
- User Lifestyle: 3/5
- Goal: -1
- User Budget: 3/5
- User Time: 0
- Weight: over
- Age: U20
- Sex: m
- Activity: normal
- Stress: 0
- Sleep: good

What would you like to modify?  
Please type in the attribute you want to modify or 'none' if you changed your mind: 18:10

lactose free 18:10 ✓

Your choices are: I am lactose intolerant or I am NOT lactose intolerant. Please select one: 18:10

I am lactose intolerant 18:10 ✓

You successfully changed attribute 'Lactose-Free' from: 0 -> 1.  
Now you can ask me to suggest something again. 18:10

suggest me something to eat 18:10 ✓

Recipe: Omelet with tomatoes and basil  
URL: <https://ricette.giallozafferano.it/Frittata-con-pomodori-e-basilico.html>

You can ask me something about this recipe, such as:  
its nutritional facts, its cost, its popularity...and much more!  
Or we could focus on another recipe if you don't like it.  
Just try me.

Ricette di cucina - Le Ricette di GialloZafferan...  
Frittata con pomodori e basilico  
La frittata con pomodori e basilico è una ricetta dai sapori semplici e facile da realizzare, un perfetto salvacena estivo che piacerà a tutti!

18:10

how much does it cost? 18:12 ✓

Omelet with tomatoes and basil has a **low** cost level, in line with how much you intend to spend (medium cost level). 18:12

is it sustainable? 18:12 ✓

## PROFILE MODIFICATION

## SUGGESTION 1 AND EXPLANATIONS


Omelet with tomatoes and basil has a **low** cost level, in line with how much you intend to spend (medium cost level). 18:12

is it sustainable? 18:12 ✓

Omelet with tomatoes and basil contains egg and basil, which have low carbon/water footprint sustainability. 18:12

suggest me another one 18:12 ✓

Ricetta: Baked zucchini omelette  
URL: <https://ricette.giallozafferano.it/Frittata-di-zucchine-al-forno.html>

Ricette di cucina - Le Ricette di GialloZafferan...  
**Frittata di zucchine al forno**  
La frittata di zucchine al forno con fiori di zucca è un secondo piatto gustoso e salutare, adatto anche da servire come antipasto o per un buffet. 

Here you go.  
Now you can also ask me to compare them according to some criteria such as:  
- nutritional values;  
- cooking time;  
- health risks;  
- popularity;  
- seasonability;  
And much more, just try me! 18:12

compare them according to their macros 18:13 ✓

Omelet with tomatoes and basil has the following macronutrient distribution: 2.3g of carbs, 16.3g of fats, and 14.5g of proteins. The ideal macronutrient distribution for a man with your type of activity is 53% carbs, 15% fats, 32% proteins.

This recipe may not be the best choice as it deviates significantly from the ideal macronutrient ratio. It is **low** in carbs. It is **high** in fats. Consider adjusting the portion size or exploring other recipes. The deviations from ideal values are: carbs = -15.24, fats= 11.34, proteins = 3.91 (anything in -4.4 is good).

Baked zucchini omelette has the following macronutrient distribution: 3.2g of carbs, 18.5g of fats, and 22.5g of proteins. The ideal macronutrient distribution for a man with your type of activity is 53% carbs, 15% fats, 32% proteins.

This recipe may not be the best choice as it deviates significantly from the ideal macronutrient ratio. It is **low** in carbs. It is **high** in fats. It is **high** in proteins. Consider adjusting the portion size or exploring other recipes. The deviations from ideal values are: carbs = -20.23, fats= 11.87, proteins = 8.36 (anything in -4.4 is good).


Between the two, Omelet with tomatoes and basil seems **less unbalanced** in the macro-nutrients composition. 18:13




compare them according to their rating 18:13 ✓

Omelet with tomatoes and basil (54 ratings, average: 4.3/5) is more popular than Baked zucchini omelette (45 ratings, average: 4.3/5) in the community. 18:13

i want a potato-based recipe 18:14 ✓

Recipe: Potato omelette  
URL: <https://ricette.giallozafferano.it/Frittata-di-patate.html>

Ricette di cucina - Le Ricette di GialloZafferan...  
**Frittata di patate**  
La frittata con le patate è buonissima da gustare calda o fredda, un ricco secondo piatto veloce e semplice da preparare! 

Menu Write a message...   

## SUGGESTION 2

## COMPARISONS SUGGESTIONS 1-2

## SPECIFIC USER-REQUEST