**What to Eat Today – Healthy Diet Recommendation System**

by

谢欣言 (1930026136)

徐贺 (1930026139)

A Final Year Project Thesis

submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Science (Honours)

in

Data Science

at

BNU-HKBU

UNITED INTERNATIONAL COLLEGE

November, 2022

# ABSTRACT

What to eat today? Should I eat healthy for keep fit or indulge in some delicious junk food? As there are so many foods to choose today, everyone may ask these questions during lunch or dinner time. For different kind of people, such as who are working out or with chronic illnesses, their daily nutritional needs and intakes that require attention are different.

Our main objection is to develop a website for everyone to find their diet plan according to their personal physical condition. The website will track the users’ daily nutritional intake or sports activities to give the most appropriate recommendations. In addition, we not just want users to find what’s “healthy” to eat. If users are not satisfied with the automatic recommendations, they can customize them: change the recommendation method, and find what other users with similar flavor are eating.

Besides, the website can provide other services such as querying food databases to visualize the nutrient contents of different foods, recording users’ dietary history to help them make decisions and plans.

To implement our project, Elasticsearch is used as the core of search engine and recommendation for nutritional standards, and Collaborative Filtering, which is a big data analytical method, is used to find similar users and provide flavor-based recommendations. In addition, we used Python + Django architecture for website frontend.

This thesis will present our approach to the above problem and our plan to achieve it, as well as the methodology we will use and the expected results.

Contents

[ABSTRACT 2](#_Toc119880211)

[1. Introduction 5](#_Toc119880212)

[2. The Dataset 6](#_Toc119880213)

[2.1 Our Goal and Source 6](#_Toc119880214)

[2.2 Visualization 7](#_Toc119880215)

[3. Nutritional Background Information 9](#_Toc119880216)

[3.1 BMR (Basal metabolic rate) 9](#_Toc119880217)

[3.2 TDEE (Total Daily Energy Expenditure) 10](#_Toc119880218)

[3.3 Selection and Proportion of the 8 Nutrient Indicators 11](#_Toc119880219)

[4. Eat Healthy: Search by nutritional standards using Elasticsearch 14](#_Toc119880220)

[4.1 Motivations and Goals 14](#_Toc119880221)

[4.2 Elasticsearch (ES) 14](#_Toc119880222)

[4.2.1 Faster Speed Relative to MySQL 14](#_Toc119880223)

[4.2.2 Distributed Storage of Data (indexes) 14](#_Toc119880224)

[4.2.3 Support for Complex Query Methods and Custom Scripts 15](#_Toc119880225)

[4.3 Indexing 15](#_Toc119880226)

[4.4 Scripted Scoring 17](#_Toc119880227)

[5. Eat What You Like: Find Similar Users by Collaborative Filtering 22](#_Toc119880228)

[5.1 Motivations and Goals 22](#_Toc119880229)

[5.2 Conception 22](#_Toc119880230)

[5.3 Strategy 22](#_Toc119880231)

[5.4 Methodology 24](#_Toc119880232)

[5.5 Implementation in Our System 25](#_Toc119880233)

[5.6 Conclusion 26](#_Toc119880234)

[6. Architecture and Development Process of Web Interface 28](#_Toc119880235)

[6.1 Website Architecture 28](#_Toc119880236)

[6.2 Website Frontend: 29](#_Toc119880237)

[6.2.1 Index.html 29](#_Toc119880238)

[6.2.2 Search.html 30](#_Toc119880239)

[6.2.3 User.html 31](#_Toc119880240)

[6.2.4 Data.html 31](#_Toc119880241)

[6.3 Website Backend and Basic Functions: 32](#_Toc119880242)

[6.3.1 User Registration and Physical Condition Recording & Updating 32](#_Toc119880243)

[6.3.2 Recording and Visualization of User Previous Eating History 33](#_Toc119880244)

[6.3.3 Automatic Healthy Food Recommendation 33](#_Toc119880245)

[6.3.4 Querying of Food Nutrient Content Database 35](#_Toc119880246)

[6.3.5 Other Special Features 35](#_Toc119880247)

[7. Working Life-cycle of the System 37](#_Toc119880248)

[8. Conclusion 38](#_Toc119880249)

[References 40](#_Toc119880250)

[Appendices 42](#_Toc119880251)

# Introduction

Various types of people have different daily nutritional requirements and intakes that need to be considered, such as those who exercise regularly or have chronic conditions.

The creation of a website where everyone can obtain a diet plan tailored to their unique physical situation is our main argument. To provide the best advice, the website will monitor users' daily dietary consumption and exercise routines. Additionally, we don't just want users to discover “healthy” food. Users have the option to adjust the recommendation algorithm and look up what other users who have similar tastes are consuming if they are dissatisfied with the automatic choices.

In the implementation system, Elasticsearch serves as the foundation of the search engine and nutritional standards recommendations, while Collaborative Filtering, a big data analytical technique, is utilized to identify similar users and deliver flavor-based suggestions. In addition, the front end of the website was built using Python and Django.

The following parts will introduce the methodology we used in this process, the intended functions of the website and the life-cycle of how a user will use it as well as what they can get from it.

# The Dataset

## 2.1 Our Goal and Source

We strongly hope that on our platform, logged users can obtain as much food information as possible, which not only includes the different types of vegetables, meat, etc., but also requires specific cooking methods. Moreover, this is also based on our philosophy, to make an ideal recipe as directly as possible for the recommended choice. During our search for the dataset, our UIC’s food science technology (FST) students were also consulted about whether various practices of food directly or indirectly affect the nutritional value of the food. For example, vitamin C will be destroyed at high temperature, and we will add appropriate condiments, such as salt and oil, when cooking. These factors lead to completely different effects when eaten raw or cooked. Based on this idea, we found out a website in HKSAR called **“Centre for Food Safety”** <https://www.cfs.gov.hk/>. The food data on this website is much different than what we see on Kaggle.com or some other dataset website platforms. Ordinary food data sets usually lack nutritional data, such as saturated fat and trans-fat. Although these similar nutrients may be extremely easy to ignore in our lives, as a rigorous graduation FYP thesis design, we want these data to be reflected in us above the final result. At the same time, quite a lot of practices have also been entered into this data set. For example, Chinese cabbage is prepared without adding salt or with salt added. The nutritional data of each must be different, which is reflected in this data set.

We obtained the data sets of several food types we need through data crawling, as well as the processing of some blank data. Since the reason for some blanks on the website is lack of experimental proof and missing, so our processing of blank data is to return to 0. These are our result of categorizing them in eight types:

|  |
| --- |
| Aquatic animals other than fish |
| Meat and meat products |
| Eggs and egg products |
| Cereals and cereal products |
| Milk and milk products |
| Poultry and poultry products |
| Vegetables and vegetable products |
| Fruits and fruit products |
| Fish and fish products |

Table 1 Eight Types

(In our subsequent dataset using in our project, we again reduce the categories to six, which contained and named as cereal, fruit, meat, vegetable, milk, egg.)

## 2.2 Visualization

In our datasets of these foods, we used the information of 12 kinds of food elements including energy, protein, fat, carbohydrates, etc. as the usage data of our project. Among the 12 nutritional elements in our datasets, we selected eight of the main nutritional elements as our judgment on whether to give the recommendation to users. The reason we made this choice is that these eight elements are relatively dominant based on our reference to the nutritional value contributions of other nutritional knowledge literature and related recipes. For the remaining four (Sugar, Trans fat, Saturated fat, Cholesterol), we use them for display purposes, that is, it is very convenient for users to view them if they have special needs.

Here we want to show the data of food energy in our datasets:

Table 2 Meat Energy

In the meat food, it can be seen that the general calorie is relatively high, and the kcal distributed in the range of 0-500 is very large. This is one of the reasons why many people who lose weight need to eat less meat. Of course, proper cooking methods may also affect calories. For example, it can be seen in the picture that some meats have lower calories.

Table 3 Fruit Energy

In fruit, it can be easily seen from the figure that the calories are far less than that of meat, and there is more kcal distributed between 0-150. Interestingly it is worth mentioning that among fruits, durian has a very high calorie content. As you can see in the picture, the outlier is the calorie content of durian. Therefore, people who lose weight must eat less or avoid eating durian.

Table 4 Vegetable Energy

We can find that the calorie distribution of vegetables is relatively scattered. After looking at the data set, we found that the practice of vegetables largely determines this. Raw vegetables are usually low in calories. But after cooking, they increase a lot.

# Nutritional Background Information

For the nutritional standards calculation in Elasticsearch in the following part, some basic knowledges are needed:

## 3.1 BMR (Basal metabolic rate)

Basal metabolic rate (BMR) is the rate of energy expenditure per unit time by endothermic animals at rest. It is reported in energy units per unit time ranging from watt (joule/second) to ml or joule per hour per kg body mass . Proper measurement requires a strict set of criteria to be met. These criteria include being in a physically and psychologically undisturbed state and being in a thermally neutral environment while in the post-absorptive state (i.e., not actively digesting food). In some theories of the physiology, there are strong evidences that confirm the BMR varies from person to person. A study of 150 adults representative of the Scottish population reported basal metabolic rates ranging from 1,027 kcal (4,300 kJ) per day to as high as 2,499 kcal (10,460 kJ); mean BMR was 1,500 calories (6,300 kJ) per day. Statistically, the researchers calculated that 62.3% of this variation could be explained by differences in fat-free mass.

Other factors explaining the difference included fat mass (6.7%), age (1.7%), and experimental error, including within-subject variation (2%). The remaining variation (26.7%) was unexplained. This remaining difference can be explained neither by sex nor by different tissue sizes in high-energy organs such as the brain. As well as in biochemistry, approximately 70% of the total human energy expenditure is due to fundamental vital processes taking place in the body's organs (see table). About 20% of a person's energy expenditure comes from physical activity and another 10% comes from thermogenesis or food digestion (postprandial thermogenesis).

At the same time, what needs to be added is that the basal metabolic rate is the energy consumed by the most basic metabolism of the human body. Simply put, the energy consumed by the human body even if lying still and doing nothing all day.

Regarding our calculation formula for BMR, because there are actually many formulas that can calculate this value, such as the original Harris–Benedict equation, the Mifflin St Jeor equation and the Katch–McArdle formula (resting daily energy expenditure).

Consequently, we chose the **Mifflin St Jeor equation** to calculate BMR:

* + - s is +5 for males and −161 for females,
    - P is total heat production at complete rest,
    - m is mass (kg),
    - h is height (cm),
    - a is age (years).

## 3.2 TDEE (Total Daily Energy Expenditure)

BMR is only the most basic indicator of metabolic consumption, besides, we also need to calculate the Total Daily Energy Expenditure (TDEE). Since the BMR represents how many calories our body burns when at rest, it is necessary to adjust the numbers upwards to account for the calories you burn during the day.

In our calculation, we multiply the BMR with a coefficient named physical activity level (PAL). To explain it, PAL is a coefficient that reflect the degree of your body activities in a week. More exercises and sports mean higher PAL, and vice versa. And the format for PAL:

We set four levels of physical activity level to calculate TDEE

1. Sedentary (Little or no exercise), corresponding to 1.2
2. Lightly active (Light exercise/sports 3-5 days a week), corresponding to 1.375
3. Moderately active (Moderate exercise/sports 3-5 days a week), corresponding to 1.55
4. Very active (Hard exercise/sports 6-7 days a week), corresponding to 1.72
5. Extra active (Hard exercise/sports 6-7 days a week, plus physical job), corresponding to 1.9.

In our research, BMR is a basic index obtained from human data, and TTDE is a very important index of human body weight loss or weight gain, which is calculated on the basis of BMR. BMR is significant for us to refer to people of different weights, heights, ages and genders. When we want to lose our weight in the near future or a certain period, we need to calculate the BMR from our current body data, and then multiply it by a CAL coefficient, and we can obtain the TDEE. When the overall calorie intake of food in a day is less than the TDEE value, and by the way, we should also have an appropriate amount of exercise. Ultimately, we should lose weight after a certain period. This is the scientific and healthy principle of weight loss, and it is also completely possible for those who want to gain weight. The main testimony is that we must strictly follow the BMR and TDEE level, and calculate our intake in details.

## 3.3 Selection and Proportion of the 8 Nutrient Indicators

Before we officially launch our project, it is necessary to define the nutrients and their suitable content range. In our daily life, dietary structure is the most direct and crucial factor for human growth and health. A long-term regular and reasonable diet with sufficient nutrients in the diet can maintain and promote human health, improve the body's immunity, and resist various diseases. Diet is also a variable factor, influenced by environment, knowledge, economy, culture, etc. According to the “The Chinese dietary guideline (2022)” from the website <http://dg.cnsoc.org/>, the ratio of intaking nutrients as follow:

|  |  |
| --- | --- |
| Salt | Less than 5g |
| Oil | 25~30g |
| Milk & Milk product | 300~500g |
| Meat | 120~200g |
| Vegetable | 300~500g |
| Fruit | 200~350g |
| Cereal | 200~300g |
| Water | 1.5~1.7L |

Table 5 Ratio of Nutrients

It is also worth mentioning that we have read the information and found that the energy sources of food we ingest in three meals a day are three categories: carbohydrates, fats and proteins. Their proportions and the heat generated per unit (gram) are as follows:

Table 6 Proportion

1. Calories produced by Carbohydrates = 4 kcal (4 kcal)/gram
2. Calories produced by protein = 4 kcal (4 kcal)/g
3. Calories produced by fat = 9 kcal (9 kcal)/g

At the same time, our project wants to consider the health of users, so we hope that the foods we recommend are based on health theorems. Therefore, among the many nutrients, the six most basic nutrients for the human body: carbohydrates, fats, proteins, water, vitamins, inorganic salts. Since water is not food, we do not recommend water. Then we chose sodium and calcium as the main representatives of inorganic salts (the content of other elements is less), and vitamin C as the main representative of vitamins. We also want to emphasize that dietary fiber intake is one of the most important factors in weight control. Consequently, in our project, as just mentioned in the introduction to the data set, we finally chose **energy (calory)**, **carbohydrates**, **protein**, **fat**, **dietary** **fiber**, **sodium**, **vitamin** **C**, **calcium**, these eight nutrients as our recommended basis of foods.

In that case, we need to get **the total quantity** of each nutrient to be consumed per day:

1. For **energy (calory), carbohydrates, protein**: the quantity can be obtained from the above graph;
2. For other **trace elements** (**dietary** **fiber**, **sodium**, **vitamin** **C**, **calcium**), because their amount should be consumed each day is basically the same for people with different physical conditions, we find the table in “The Chinese dietary guideline (2022)” as below:

|  |  |
| --- | --- |
| Dietary fiber | 25g |
| Calcium | 800mg |
| Sodium | 1500mg |
| Vitamin C | 100mg |

Table 7 Trace Elements

Also, **the proportion** of each type of food to each nutrient can be inferenced from the above quantities (the proportion is basically equal to the ratio of quantity):

1. For **energy (calory), carbohydrates, protein, sodium**, and **calcium:** theproportion is the distribution of their quantities;
2. For **dietary** **fiber:** as we mentioned in the dataset part, dietary fiberonly exists in **fruits, vegetables** and **cereals**, so the proportion is the distribution of only these 3 categories;
3. For **Vitamin C**: because **VC** cannot exist in high temperature, they can only be got from **fruits** (eaten raw), so the proportion is only taken by **fruits**.

In addition, there several fundamental rules that we all should obey are as follow. We should cherish food, prepare meals according to needs, and advocate sharing meals without wasting them. At the same time, it is necessary to choose fresh and hygienic food and appropriate cooking methods. The process of food preparation is divided according to the raw and cooked food, and the second heating of cooked food should be thoroughly heated. We must also learn to read food labels and choose food reasonably. It is very important to come home to eat more often, to enjoy food and family affection. We should inherit the fine culture and develop a new style of food civilization.

# Eat Healthy: Search by nutritional standards using Elasticsearch

## Motivations and Goals

As one of the main functions of this system, we need to find recipes for the users based on their body information in the food database that best meet their nutritional needs.

Since our website records users' height, weight, age, gender, activity level and dietary goals during registration, we can set nutritional standards by the previously mentioned nutritional calculation measurements (BMR, TDEE, fixed values of micronutrients, etc.) and then use Elasticsearch for a custom scoring script to search for ingredients that are closest to the standards.

## Elasticsearch (ES)

Elasticsearch is a search engine core based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. According to the DB-Engines rating, Elasticsearch is the most widely used enterprise search engine now in the world. It is used in our system because it has the following advantages and features:

### 4.2.1 Faster Speed Relative to MySQL

Elasticsearch processes all the input data and creates “index” to store them, and then put them in memory, thus improving the search efficiency. In this regard, ES is better than MySQL's B+ tree structure. MySQL needs to put the database on the disk, so each reading command needs to read from disk first and then find the corresponding data node, while ES can find the approximate location of the target document directly in the memory, maximizing speed of each search command always in 1ms.

### 4.2.2 Distributed Storage of Data (indexes)

ES is inherently a distributed architecture, which ensure data reliability and availability.

**Nodes**

ES is clustered by multiple nodes, which are essentially a Java process. Each node holds information about the state of the cluster, including all node information, all indexes and related Mapping to Setting information and routing information for the sharding.

**Shards**

Sharding is the cornerstone of ES distributed storage and the underlying basic read and write unit. The purpose of sharding is to partition huge indexes and spread data around the cluster. The shard is divided into a master shard, which handles write requests and stores data, and a replica shard, which only stores data and is a copy of the master shard, with documents stored on a specific master and replica shard.

### 4.2.3 Support for Complex Query Methods and Custom Scripts

ES is obviously better than MySQL when it comes to combination queries. It supports complex combination queries such as aggregation operations (MySQL has to build indexes beforehand to do this), and can accomplish such complex operations since each field is indexed by default and can be combined with each other when querying.

Besides, ES support custom scoring script as searching method to replace default BM25 as well as TFIDF, which is extremely important for the nutritional standard calculation in our system to generate recommendations.

## Indexing

In Elasticsearch, data are stored as indexes, which can be indexed in a structured or unstructured form. in our system, we firstly transferred the six previously preprocessed datasets into **json** formats, which is one of the structured data types.

Here we also put the **max** and **min** value of each record (calculated in the data preprocessing step) into the index, therefore normalization of standards can be conducted in the following scripted scoring.

Besides, **normalized** values of each type of nutrition are also put into the index, to conduct normalization in advance beyond the following scripted scoring, thus improving the **efficiency** of the algorithm.



Table 8 JSON

Then we index those 6 json datasets as 6 indexes for different types of food. While indexing, we adopt the “bulk” methods, which allow us to accelerate the speed and efficiency of indexing for large amount of data:

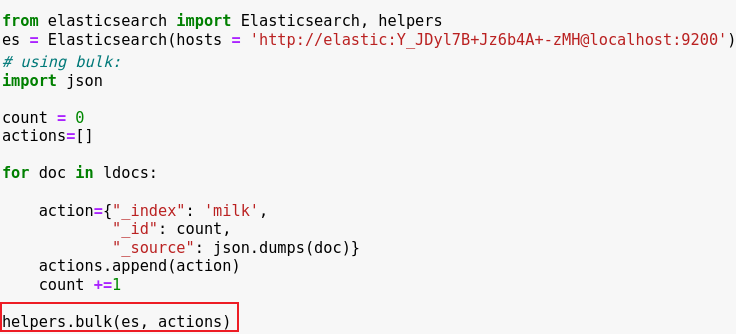


Table 9 bulk

The indices can be checked on the web interface of Elasticsearch:

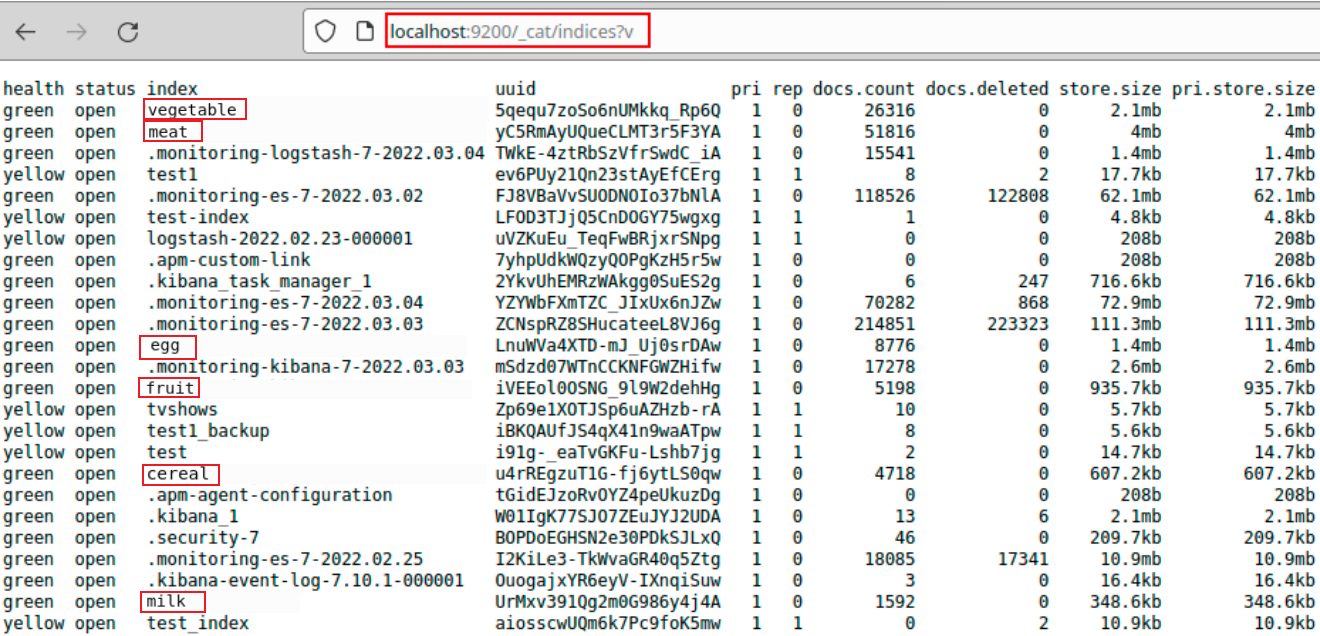


Table 10 indexes

## Scripted Scoring

By default, Elasticsearch uses the BM25 algorithm as a search method to perform a TFIDF-like text search based on word frequency, so we can conduct search like following:

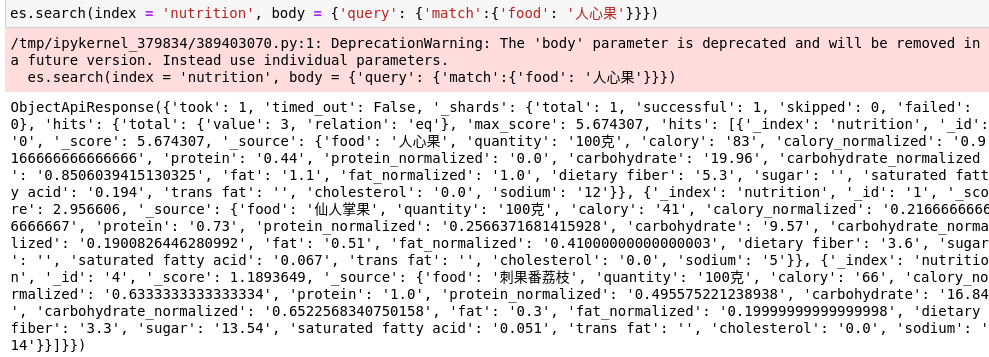


Table 11 search results

In addition to that, ES allows us to customize the search algorithm, like following:

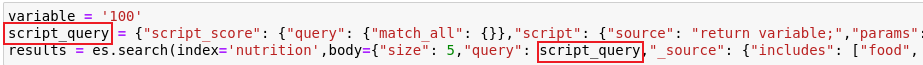


Table 12 script

In that case, we transformed the previously described nutritional standards into the following algorithm (Recommendation Scoring Function):

1. Firstly, based on user information (**weight**, **height**, **age** and **gender**), calculate the **BMR** (Basal Metabolic Rate) of each user:
2. script\_query = {"script\_score": {"query": {"match\_all": {}},"script": {"source": '''**double** BMR = 9.99 \* params.weight + 6.25 \* params.height - 4.92 \* params.age + 166 \* params.gender - 161 + params.bias;
3. Based on the **active level**, which means user's daily exercise level (**Tends to be static: 1.2, low activity level: 1.375, normal activity level: 1.55, high activity level: 1.72, intense activity level: 1.9**), we can calculate the **TDEE** (Total Daily Energy Expenditure), which indicates the **standard** of daily **calory** intake (as well as **normalization** for that value):
4. **double** TDEE = BMR \* params.active\_level;
5. **double** TDEE\_normalized = (TDEE - doc['calory\_min'].value) / (doc['calory\_max'].value - doc['calory\_min'].value) \* ratio1;

Besides, **ratio1** is the parameter used to proportionally divide the total daily nutrient content of all ingredients into individual food groups (based on the proportion we talked before, according to the respective **quantity** share), which is defined as following:

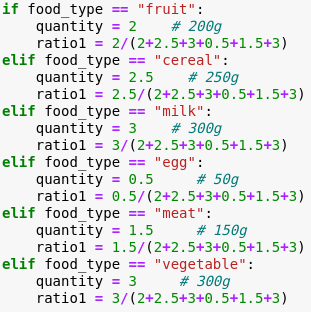


Table 13 ratio1

1. As we mentioned in the nutritional calculation part, **TDEE**, which indicates to the daily calory standard, can be divided into 3 sources: **protein**, **fat** and **carbohydrate**. The proportion is **0.15, 0.25** and **0.65** respectively (Daily **calory** intake: **15%** from **protein**, **25%** from **fat**, and **65%** from **carbohydrate**). Based on that, we can calculate the standard for the 3 types separately (as well as **normalization** for that values):
2. **double** protein standard = TDEE \* 0.15;
3. **double** protein\_standard\_normalized = (protein\_standard - doc['protein\_min'].value) / (doc['protein\_max'].value - doc['protein\_min'].value) \* ratio1;
4. **double** fat\_standard = TDEE \* 0.25;
5. **double** fat\_standard\_normalized = (fat\_standard - doc['fat\_min'].value) / (doc['fat\_max'].value - doc['fat\_min'].value) \* ratio1;
6. **double** carbohydrate\_standard = TDEE \* 0.65;
7. **double** carbohydrate\_standard\_normalized = (carbohydrate\_standard - doc['carbohydrate\_min'].value) / (doc['carbohydrate\_max'].value - doc['carbohydrate\_min'].value) \* ratio1;
8. After generating the standards, we can calculate the difference (**absolute value**) between them (**normalized**) and the amount of each type of nutrients from each food (only for **calory**, **protein**, **carbohydrate**, **fat**, **Na**, and **Ca**):
9. **double** calory\_res = TDEE\_normalized - params.quantity \* doc['calory\_normalized'].value;
10. **double** protein\_res = protein\_standard\_normalized - params.quantity \* doc['protein\_normalized'].value;
11. **double** fat\_res = fat\_standard\_normalized - params.quantity \* doc['fat\_normalized'].value;
12. **double** carbohydrate\_res = carbohydrate\_standard\_normalized - params.quantity \* doc['carbohydrate\_normalized'].value;
13. **double** Na\_res = doc['Na\_standard\_normalized'].value \* ratio1 - params.quantity \* doc['sodium\_normalized'].value;
14. **double** Ca\_res = doc['Ca\_standard\_normalized'].value \* ratio1 - params.quantity \* doc['calcium\_normalized'].value;

The **quantity** is the amount of each type of food that a person is supposed to intake each day, which is defined as following (the theory of defining quantity is talked in the nutrition calculation part):

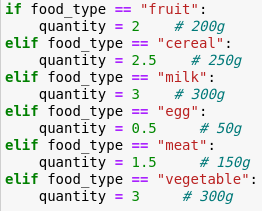


Table 14 ratio2

1. For **dietary fiber**, because it only exists in **fruits, vegetables** and **cereals**, we added two parameters: **ratio2\_1** and **ratio2\_2** for it to calculate the standard and residual value exclusively:
2. **double** dietary\_fiber\_standard\_normalized = doc['dietary\_fiber\_standard\_normalized'].value \* params.ratio2\_1;
3. **double** dietary\_fiber\_res = (dietary\_fiber\_standard\_normalized - params.quantity \* doc['dietary\_fiber\_normalized'].value) \* params.ratio2\_2;

The **ratio2\_1** here indicates the proportion of the amount of dietary fiber shared between the three types of food (other types of food are with value **1**), and **ratio2\_2** indicates whether to use the standard of dietary fiber as a criterion to limit a certain food, where 1 for only **fruits, vegetables** and **cereals**, and **0** for the others:

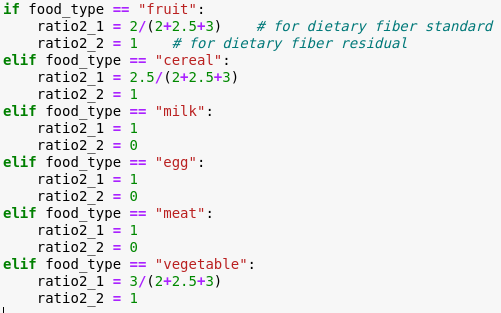


Table 15 ratio2

1. For **Vitamin C**, because it only exists in **fruits** (because it would be broken in high temperatures), we use **ratio3** to calculate its residual:
2. **double** VC\_res = (doc['VC\_standard\_normalized'].value - params.quantity \* doc['Vitamin\_C\_normalized'].value) \* params.ratio3;

The **ratio3** indicates whether to use the standard of Vitamin C as a criterion to limit a certain food, where only the value of fruit is set to **1**:

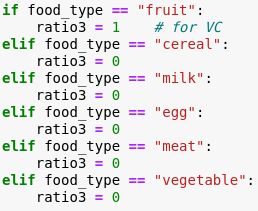


Table 16 ratio3

1. The next step is to transfer those residuals to their **absolute value**, because we want the exact nutrient values of each food are **floating up and down** of the standard value while being as close to the it as possible (some nutrients value can exceed or leave below the standard to meet different needs of users).
2. **if** (calory\_res <= 0){calory\_res = (-1) \* calory\_res}
3. **if** (protein\_res <= 0){protein\_res = (-1) \* protein\_res}
4. **if** (fat\_res <= 0){fat\_res = (-1) \* fat\_res}
5. **if** (carbohydrate\_res <= 0){carbohydrate\_res = (-1) \* carbohydrate\_res}
6. **if** (Na\_res <= 0){Na\_res = (-1) \* Na\_res}
7. **if** (Ca\_res <= 0){Ca\_res = (-1) \* Ca\_res}
8. **if** (dietary\_fiber\_res <= 0){dietary\_fiber\_res = (-1) \* dietary\_fiber\_res}
9. **if** (VC\_res <= 0){VC\_res = (-1) \* VC\_res}
10. Finally, we need to score each food to select the **best foods** with the **highest scoring values**.

As the score of each food, the return value needs to show the **sum of the residuals**, and the larger the sum, the score should be smaller; the smaller the sum, the score should be larger (larger sum means that the nutritional value of the food is farther from the standard, so it is inversely proportional to the score).

In that case, we use the **inverse of the sum of residuals** (+1 to avoid denominator of 0) as return value:

This scoring formula can efficiently give larger differences lower scores, and the codes are following:

1. **return** 1/(calory\_res + protein\_res + fat\_res + carbohydrate\_res + Na\_res + Ca\_res + dietary\_fiber\_res + VC\_res + 1)''',

The outer parameters input to this script are:

1. "params": {"weight": weight, "height": height, "age": age, "gender": gender, "active\_level": active\_level, "quantity": quantity, "bias": bias, "ratio1": ratio1, "ratio2\_1": ratio2\_1, "ratio2\_2": ratio2\_2, "ratio3": ratio3}}}}

# Eat What You Like: Find Similar Users by Collaborative Filtering

## Motivations and Goals

Because if only use nutritional standards to search for foods provided in the dietary meal for users, their flavor or preference cannot be met.

In that case, we chose to use **Collaborative Filtering (CF)** to find the users with similar flavor to the current user.

CF reads the **“ratings”: users’ eating history**, which is the **frequency** of each food a particular user has eaten in the previous week (7 days). Because the foods that users tend to eat regularly are often **what they like to eat** or what they "should" eat (because of fitness or disease requirements, which we also interpret as what they "want" to eat).

Therefore, this step can try to let users not only eat “healthy”, but also eat what they “like” to eat.

## 5.2 Conception

In recommender systems, collaborative filtering (CF) is a method that is frequently employed. By examining the resemblance ("synergy") between users or items, this technology predicts the content that the user may be interested in and suggests this content to the user. The similarity in this case might be either demographic (gender, age, location of residence, etc.) or browsing content-related (for instance, they have shown interest in content connected to Chinese food) or it can be provided by individuals through a specific mechanism. the reaction to something (for example, some teaching sites let users rate the instructor). For instance, users A and B are both Beijing-based women in their 20s and 30s who have shown interest in information about clothing and cosmetics. In this case, collaborative filtering may consider that A and B are highly similar. Therefore, it is possible to recommend to B the content that A pays attention to and B does not pay attention to, and vice versa.

## 5.3 Strategy

Using similar statistical methods to obtain adjacent users with similar hobbies or interests, it is called user-based collaborative filtering. Method steps:

1. Collect user information: Collect information that can represent user interests. The general website system uses the method of scoring or giving evaluation, which is called "active scoring". The other is "passive scoring", which is based on the user's behavior pattern and the system completes the evaluation instead of the user, without the user directly scoring or inputting evaluation data. E-commerce websites have their advantages in obtaining passive scoring data, and the records of products purchased by users are quite useful data.
2. Nearest neighbor search (NNS): The starting point of user-based collaborative filtering is another group of users with the same interests and hobbies as the user, which is to calculate the similarity between two users. For example: Find n users who have similar interests with A, and use their ratings on M as A's rating prediction on M. Generally, different algorithms are selected according to different data. Currently, the most commonly used similarity algorithms are Pearson Correlation Coefficient, Cosine-based Similarity, and Adjusted Cosine Similarity.
3. Generate recommendation results: With the nearest neighbor set, it is possible to predict the interest of the target user and generate recommendation results. Different forms of recommendation are made according to different recommendation purposes. The more common recommendation results include Top-N recommendation and relationship recommendation. Top-N recommendations are generated for individual users and produce different results for each person. For example, by counting the nearest neighbor users of user A, select the ones with high frequency and do not exist in the rating items of user A, as Recommendation results; relational recommendation is to mine association rules from the records of the nearest neighbor users.

The similarity judgment in the recommendation algorithm actually has such a potential assumption: if two objects are very similar, that is, the distance is very close, then the two objects can easily produce the same action. If two news articles are very similar, then they are easy to be clicked and read by the same person, and if two users are very similar, then it is easy for them to click on the same news. This intuitive assumption works well most of the time.

Our project aims to find the relationship between each user. We would put hundreds of user records as the history dataset in each type of food. In these dietary records, we record the frequency of intake of a certain food by the user in a period of one week as a unit. It should be noted here that in general recommendation systems based on user selections, user ratings are usually used. We replaced "rating" with "eating frequency" in our User-based collaborative filtering.

## 5.4 Methodology

Based on the collaborative filtering recommendation algorithm of users, we can divide the algorithm into two steps. The first step is to find a user set with similar interests to the target user, and the other step is to find the items that the user likes in this set and the target user has not heard of and recommend to the target user.

1. Find a collection of users with similar interests

Seek for users who have similar rating behavior to the active user (the user whom the prediction is for). To find user groups with similar interests, we must first define indicators to measure the similarity between users. This is also like the K-nearest neighbor algorithm we mentioned, or the shadow of the clustering algorithm. We introduce cosine similarity here, and use the two to calculate the similarity between users. Assuming that A represents the frequency set of food eaten by user a every week, and B is the frequency set of food eaten by user b every week, then the similarity between users is calculated by cosine similarity as follows:

1. Recommending items

Make a prediction for the current user using the feedback from the users who shared your interests in step 1 by using their result of calculating similarity of frequency.

Give a simple example to show how to calculate the similarity between users u and v, assuming that the relevant food (encoding as 1-7) that user u and user v like and the frequency of eating in a week are as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| User/Food | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Jerry | 4 | 0 | 5 | 3 | 5 | 0 | 0 |
| Tom | 0 | 4 | 0 | 4 | 0 | 5 | 0 |
| Lily | 2 | 0 | 2 | 0 | 1 | 0 | 0 |

Table 17 Sample Frequency

Our goal is to find other customers who are most similar to Jerry. Then, through the calculation of the law of cosines and the above formula, we can get:

From this we can infer that Lily is more similar than Tom while comparing with Jerry. In other words, the behaviors of Jerry and Lily are similar, and they can be attributed to similar users, so as to make better recommendations.

## 5.5 Implementation in Our System

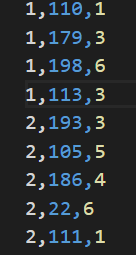
In our project, the specific implementation of the recommendation system is through the user's eating frequency and the types of various foods. The format of the foods and users data we input to the code are as follows (.csv file):



Table 18 csv

And in our code, our main part has two functions code name “getCost” and “getNearestNeighbor”, which instead of calculating the cosine similarity number and list them by the ascending order.

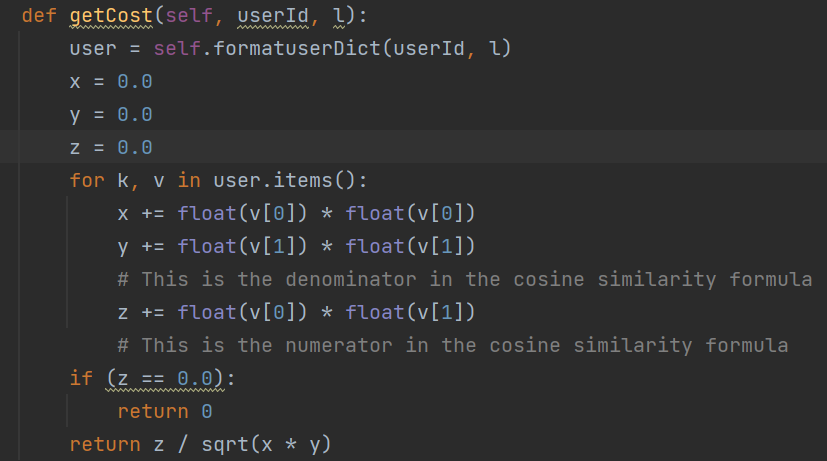


Table 19 Code1



Table 20 Code 2

Because in the recommendation process, we calculate the distance between the target user and other users through the cosine similarity theorem mentioned and explained above, and then arrange them in order from small to large. The results of the final recommended fruit to the targeted user:

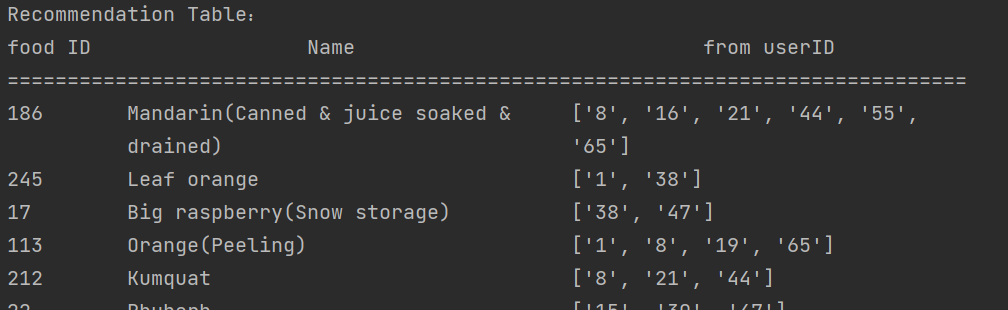


Table 21 Results

## 5.6 Conclusion

User Based CF (UBCF) is one of the most popular personalized recommendation technologies today. The basic idea of ​​this algorithm is to find users with high similarity and recommend items of interest to target users. The entire recommendation process can be divided into the following three points: first, use the existing user rating records to calculate the similarity between users; then, sort the obtained similarity results in descending order, and sort the similarity results As the nearest neighbor of the target user, the user generates the target user's neighbor matrix, and uses the existing score records of the target user's neighbor matrix to predict the score of the target user on the unrated items; finally, select the top ranked items in the predicted score as recommendations The results are fed back to the user. Compared with other algorithms, UBCF has the following advantages:

1. It has the ability to recommend information. Since users cannot get the content of recommended information in advance, it can mine the potential needs of users (that is, interest preferences that exist implicitly but have not yet been discovered by themselves) And make corresponding recommendations;
2. It can exclude analysis information that is difficult for machines to recognize, such as pictures, music, etc.;
3. Drawing on the experience and experiments of others, it improves the accuracy and integrity of content analysis, and can filter some concepts that are difficult to describe (e.g., information quality, aesthetics).

# Architecture and Development Process of Web Interface

## Website Architecture

Since basic functions of our system are implemented in **Python**, we adopted **Django** architecture for the construction of our website, which uses the **MTV** framework model: **Model M**, **View V** and **Template T**.

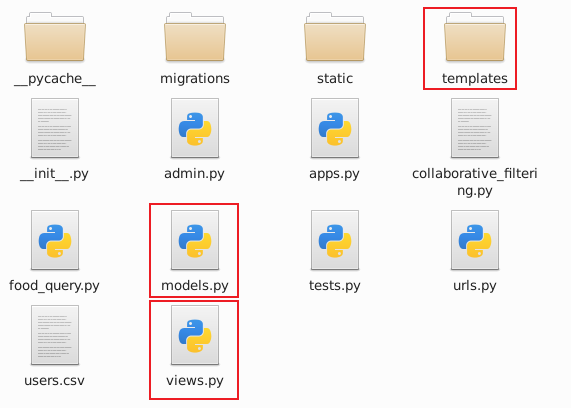


Table 22 MTV

Briefly speaking, **Templates** contain our **HTML files** for each web page, which would send **request** (submit forms) to the backend:

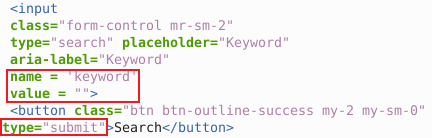


Table 23 request

Or get the **responds** from backend:



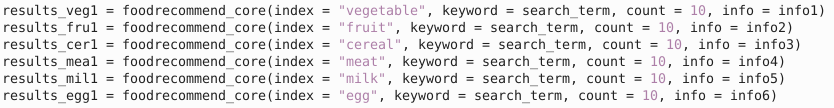
Table 24 responds

**view.py** includes the functions to **respond the requests** of each web page,



Table 25 respond

And thus call the function to conduct **Elasticsearch searching** (recommendation):



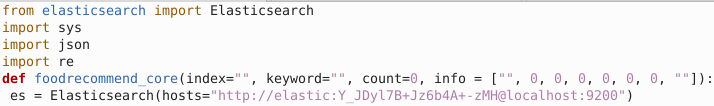
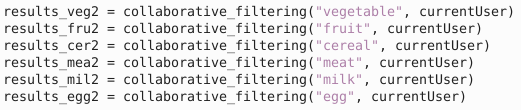




Table 26 ES search

Or call the function of **collaborative filtering** to give recommendations based on users’ preference:



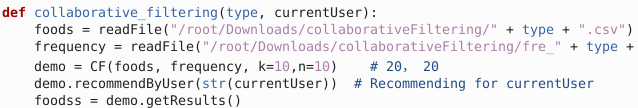


Table 27 CF search

And finally, render these recommendations as responds to the frontend (HTML):

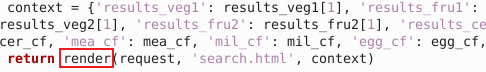


Table 28 render

## Website Frontend:

Our website has 4 pages in total: **index.html** for home page, **search.html** for recommendations, **user.html** for user information and history recording in real time and **data.html** for visualizing the foods & nutrients database.

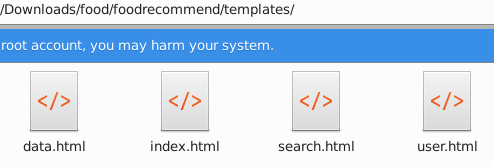


Table 29 HTML

### 6.2.1 Index.html

A **home page** with logo and search bar, that allow users to start their journey with our recommendation system.

The **search bar** with the **check boxes** below allow user to get recommendations either based on the nutrition standards (**Elasticsearch**) or preference (**Collaborative Filtering**).

Under the search bar, user need to **login** or **register** their account by input user name.

At the top of this page, **links** to other pages are provided.

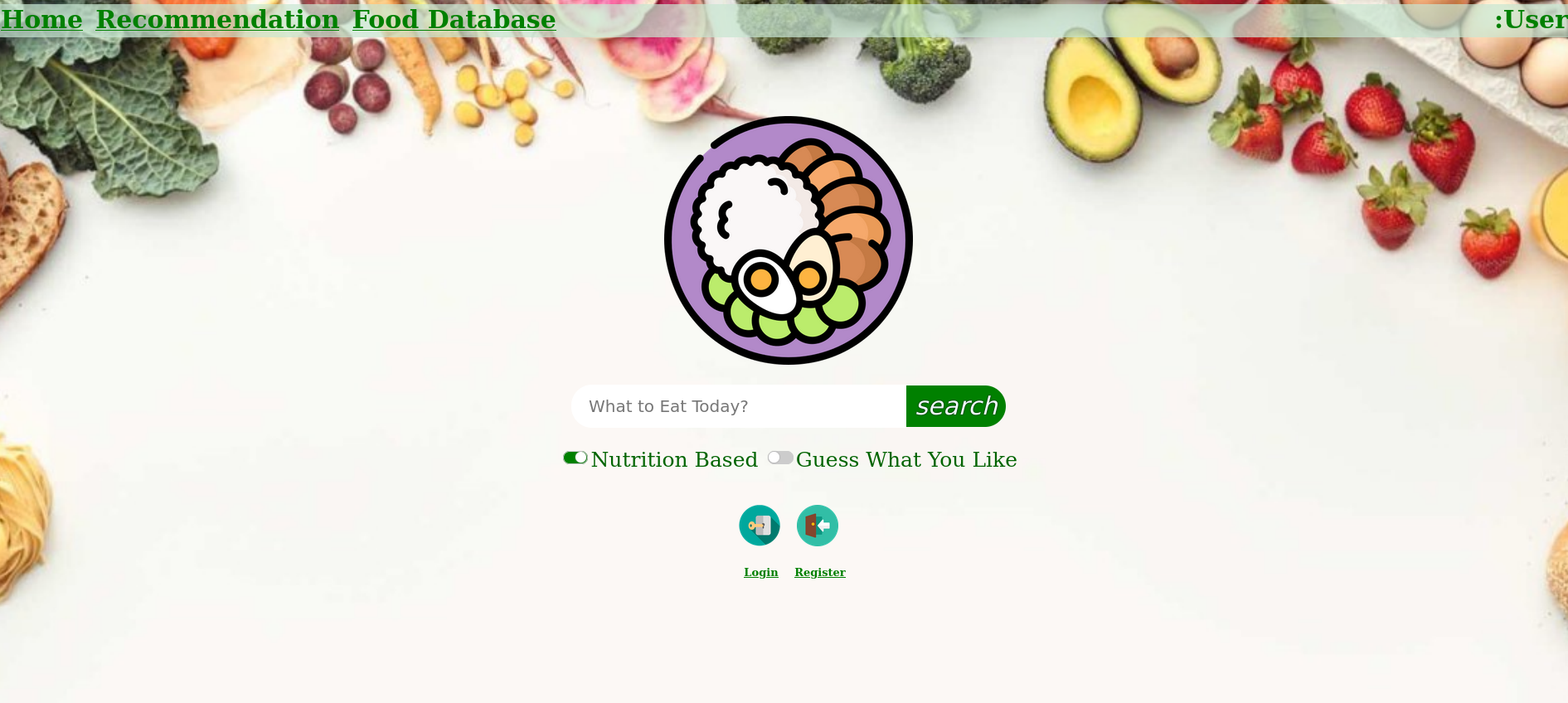


Table 30 index

### 6.2.2 Search.html

A **main page** that users can get their **recommendations** or customize them.

The recommendations of **ES** would be shown automatically, users can choose one or more foods they want to eat from each category, or change the recommending method of one category to **“Guess What You Like”**, which changed to **CF** recommendations based on preference.

Finally, their **recommended recipes** (combination of all 6 categories of foods) will be given in the order of calory value.

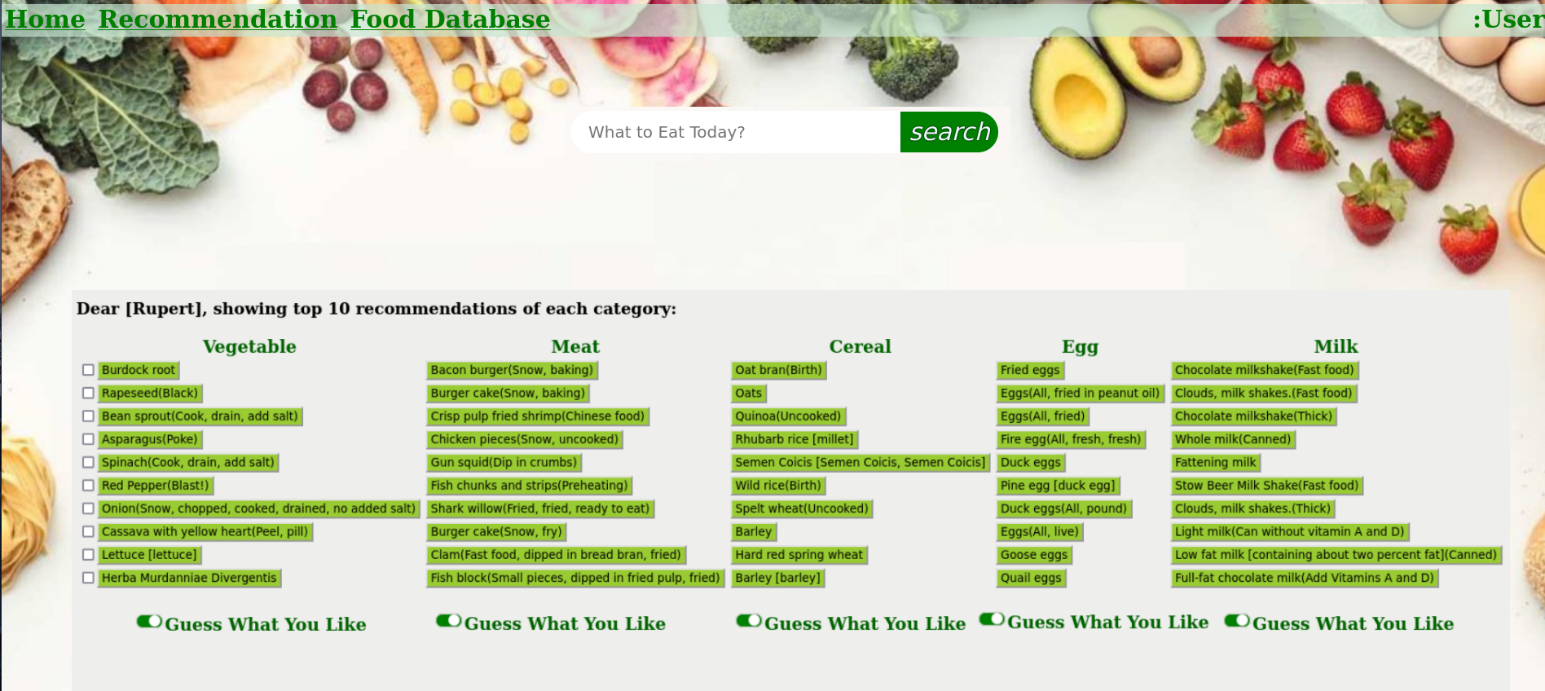


Table 31 search

### 6.2.3 User.html

The **user information page** that users would be allowed to: register as a new user (input their info), update their physical info, visualize their eating history.

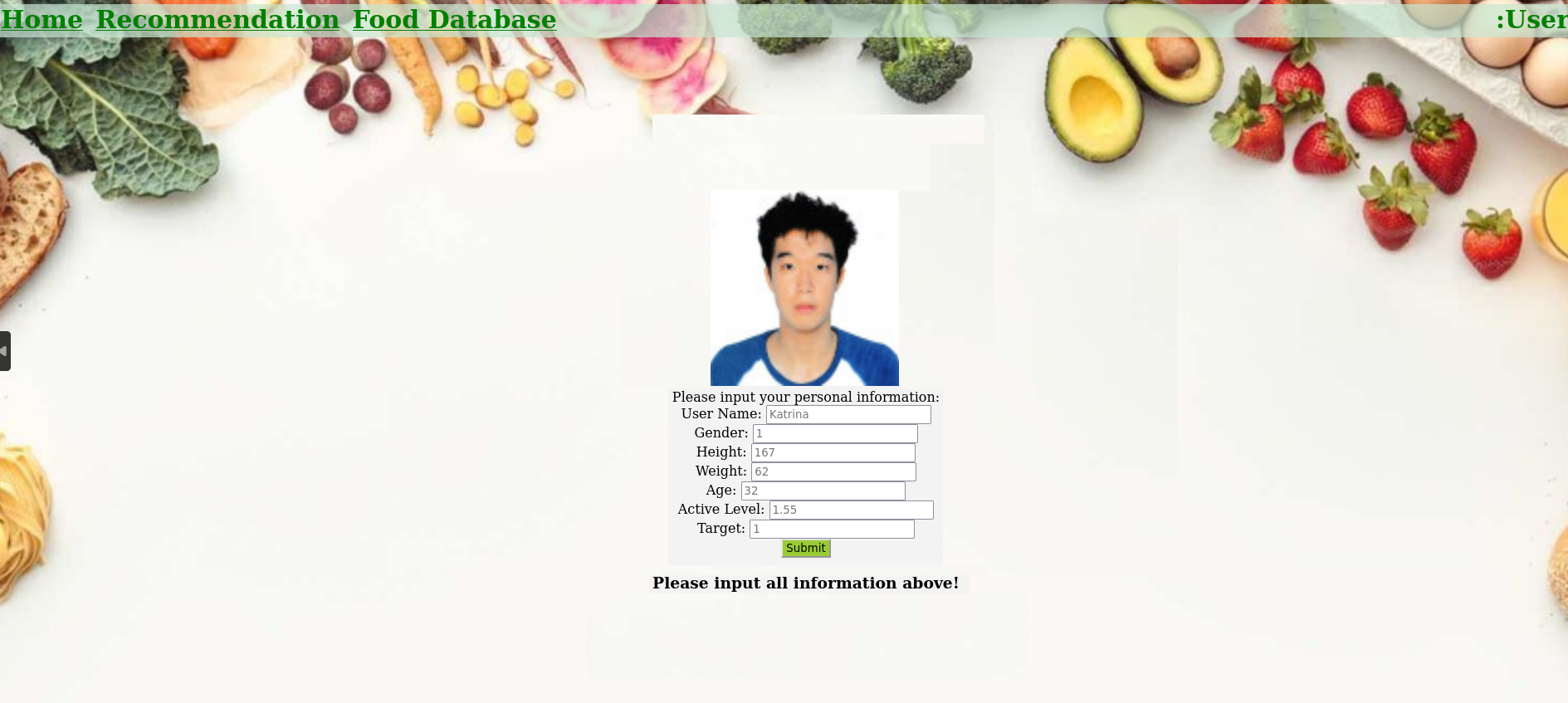


Table 32 user

### 6.2.4 Data.html

The **database search and visualize page** for users to search for any foods and check the nutrients of them.

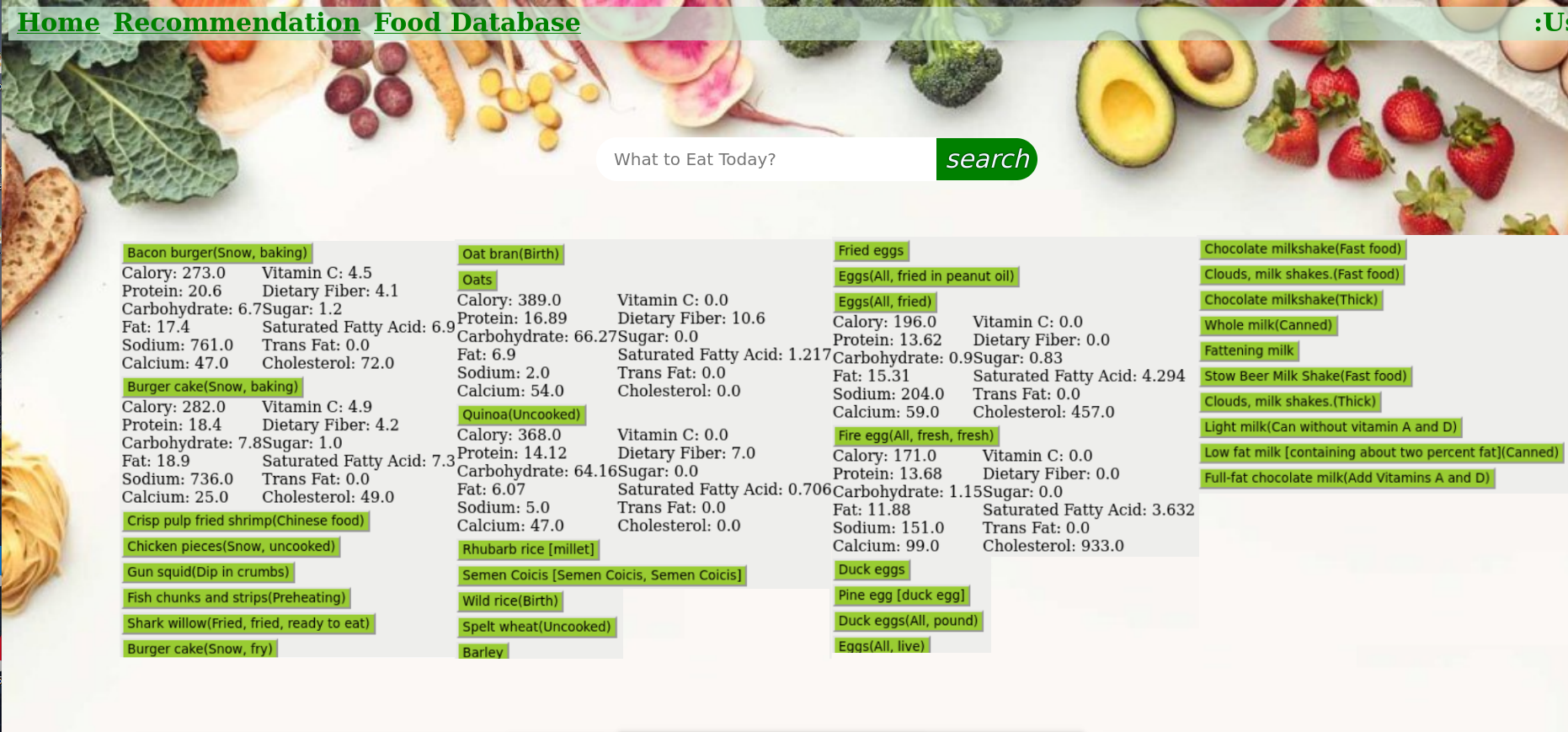


Table 33 data

## Website Backend and Basic Functions:

In our recommendation system, the following functions are implemented in the backend for users:

### 6.3.1 User Registration and Physical Condition Recording & Updating

When a user trying to use the system, they need to either login or register first.

Login process would require the user to input their username, and check with the name in the database.

For new users, when they register this website, they need to input their personal physical conditions (**username, gender, age, weight, height, active level** and **fitness target**), which can be used to calculate their nutritional standards.

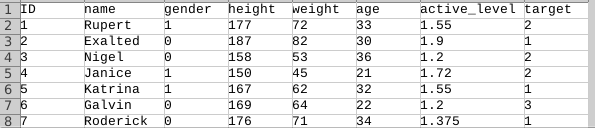


Table 34 users

The login status of current user will be passed to each page the user is staying at now using **“session”**, so the system can provide information for them:

1. **from** django.shortcuts **import** render
2. **def** index(request):
3. context=request.session.get('currentUser')
4. **return** render(request,'user.html',context)

Besides, each user can change their physical conditions or fitness target on the **“User”** page if the “**session**” has status “**update**”:

1. **if** (update == 1):   # update status
2. **for** id **in** df['ID']:
3. **if** currentUser == id:
4. # ID is not allowed to be updated
5. **if** request.GET.get('name'):
6. df.loc[int(id)-1,'name'] = request.GET['name']
7. **if** request.GET.get('gender'):
8. df.loc[int(id)-1,'gender'] = request.GET['gender']
9. **if** request.GET.get('height'):
10. df.loc[int(id)-1,'height'] = request.GET['height']
11. **if** request.GET.get('weight'):
12. df.loc[int(id)-1,'weight'] = request.GET['weight']
13. **if** request.GET.get('age'):
14. df.loc[int(id)-1,'age'] = request.GET['age']
15. **if** request.GET.get('active\_level'):
16. df.loc[int(id)-1,'active\_level'] = request.GET['active\_level']
17. **if** request.GET.get('target'):
18. df.loc[int(id)-1,'target'] = request.GET['target']
19. df.to\_csv('/root/Downloads/food/foodrecommend/users.csv')
20. update = 0 # exit the update status

### 6.3.2 Recording and Visualization of User Previous Eating History

In each day they use this website, they need to record what they eat, so the system can record their eating habits and previous foods. (If the user is **newly registered** but they want to use the recommendation system, he/she can choose to give recommendations just based on his/her personal conditions or input previous days’ diets).

These history data can help the system analyze the users’ behaviors, which are the foods he/she ate in the previous days. These foods will be used in **Collaborative Filtering** to generate the **similar users** which have **similar flavors**.

Besides, the users can see their previous foods by visualization with **pie chart** to see the amount of each nutrient they take in each day.

### 6.3.3 Automatic Healthy Food Recommendation

**Healthy: Based on Nutritional Standards (Elasticsearch)**

While a user login and move to the **“search”** page, the system will search in the database using **Elasticsearch** automatically for each category of food, and get the most appropriate foods (**top 10**) for recommendations.

Users’ physical conditions, active level and fitness target will be used to calculate and adjust the standard of each category of food, thus search and score each food using ES.

The results will be processed on the backend, and rendered to the frontend (the function is implemented in **food\_query.py** separately):

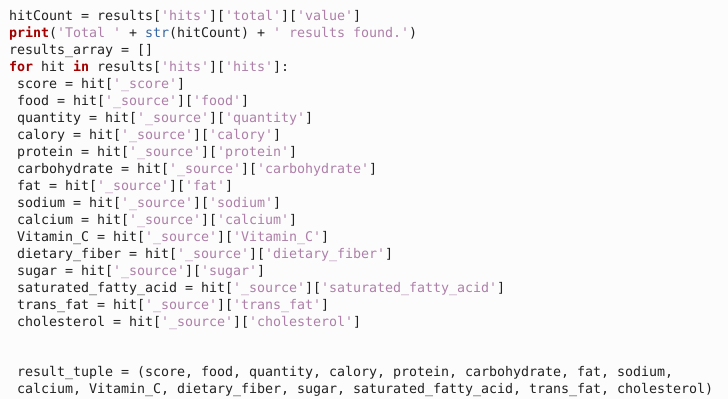


Table 35 food\_query

**Customize Recommendations**

Users don’t need to adopt all the recommendations provided for them, instead, they just need to choose (tick) some of them that they like to eat. These one or more foods in each category would be combined together as many integral recipes, and be scored and sorted finally.

**Like: Based on Preference of Similar Users (Collaborative Filtering)**

If a user is totally not satisfied with the automatic recommendations, or he/she has no idea what to eat today but just want the system to decide for them, we will change the method to **Collaborative Filtering** to recommend something they may like to eat.

Because we have recorded each user’s eating history, we can generate the frequency of each food each user has eaten within one week (7 days), which can strongly indicate each user’s eating preference (flavor), we will recommend the food that the users similar to the current user would like to eat.

The range of recommendation is still inside the foods that are “**healthy**” for the user, which means they are from the results of **Elasticsearch** (the function is implemented in **collaborative\_filtering.py** separately):



Table 36 CF

Particularly, for those users with **no or insufficient dietary history**, we still automatically recommend the most appropriate foods based on **nutritional standards.**

**Sorting of Recipe Combinations Using Elasticsearch**

Finally, the combination of 6 categories of foods will be combined together, and indexed into **Elasticsearch** again.

ES will calculate their **total distance from the sum of standards**, which will give the **healthiest** recipe.

In addition, the recipes can also be sorted by the **quantity of calory** for users to make decision.

### 6.3.4 Querying of Food Nutrient Content Database

At the same time, all the **foods** can be checked in the database to get the **detailed nutritional information**. Users can search for whatever foods in any categories freely, thus have a better understanding of the recommendations.

### 6.3.5 Other Special Features

Our website will also have some other special features:

**Cooking Method & Allergic Information**

While giving recommendations, they are not just combinations of foods, but list the nutrition structure, and for each food, we know that different cooking method (fried or steaming) or different spices (sauces) can lead to different nutrition structures, so we also give the details of “how” to eat the foods. These cooking methods are appended to each food, because different cooking methods will lead to different nutrient amount:

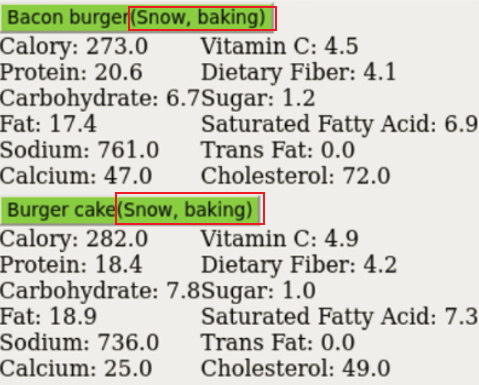


Table 37 cooking methods

Besides, some allergic information such as beans are also very important and be provided.

**Separation of Different Kind of Users**

We will separate different kinds of users: for users not registered, they cannot see the previous eating history, and cannot get an accurate recommendation, whereas registered users can use all of the functions above.

# Working Life-cycle of the System

As we separated the users into different kinds, the life-cycle of this system while a user is using will be like following:

Figure 5: Working Life-cycle of the Website

Table 38 Working Lifecycle

Registered users

Input their personal conditions and other information

(When daily log in) Record diet and sports activity level

Get automatic recommendation using Elasticsearch

Satisfied?

Choose what they want to eat from each food category

Yes

Customize the recommendation by changing to Collaborative Filtering for what they may like

Register?

New users

Visitors

No

Pre-define the expected dietary control intensity or goals

No

Yes

Visualize the eating history in pie chart in the “User” page

Get the recipe/combinations in the order of calory or distances form standards

Search for nutrition composition in the database

choose to get recommendations just based on his/her personal conditions or input previous days’ diets

# Conclusion

This Healthy Food Recommendation System is a project which combines the big data analysis & information retrieval techniques and knowledges from food technology.

The innovative methodology is that we adopted both nutrition-based Elasticsearch scripted scoring from the perspective of food science as well as recommendations from other users with similar flavors from the perspective of data analysis, which can allow users both **“eat healthy and eat what they like”**.

Starting from BMR, it can calculate the user’s TDEE based on their physical conditions, active level and fitness target. Then the nutritional standards of calory-relative nutrients will be formulated. Combined with other fixed level of trace elements, the healthiest foods will be recommended to users.

While users want to customize their recipe, the system can also find the similar users from the history, and recommend foods that the user may like to eat.

We think this project is very realistic and modern people are really in need of this system, because of the following reasons:

People in modern society often suffer from health or obesity problems, but due to the fast-paced life, it is often difficult to achieve a healthy body in exchange for intense fitness. Therefore, inspired by this, we want to use this platform to help people become healthier by controlling their diet and developing a diet that only suits them.

At the same time, the unbearable fitness meals commonly prepared for the general public on the Internet that are different from their own tastes are often a major factor preventing adherence to a controlled diet. To solve this problem, we again try to recommend suitable healthy meals within a healthy range and according to the user's taste as much as possible, thus promoting a long-lasting healthy body.

However, there are still many aspects of our system that deserve improvement:

Our dataset comes from the Hong Kong government, and in order to be applicable to people in all regions of the world, the dataset should be prepared separately for people with different eating habits in different regions;

Our website becomes slower and less efficient in operation when faced with processing a large amount of historical data from users, which needs to continue to be optimized;

Other features, such as an ordering service that provides users with recommended result menus, health knowledge and guidelines, have not been implemented in time due to time constraints.

We will keep working on those problems to make a better system.

In summary, my partner and me believe that it is a very innovative and practical study because of the special functions and ideas we try to realize, and we want to do our best to provide a platform where users can easily get their diet plan which is not only healthy but also “enjoyable”, to solve the “vital” problem for everyone: what to eat today?

# References

1. Johnstone AM, Murison SD, Duncan JS, Rance KA, Speakman JR, Koh YO (2005). "Factors influencing variation in basal metabolic rate include fat-free mass, fat mass, age, and circulating thyroxine but not sex, circulating leptin, or triiodothyronine". American Journal of Clinical Nutrition. 82 (5): 941–948. doi:10.1093/ajcn/82.5.941. PMID 16280423.
2. Fang Qi. (2018). Research on personalized healthy diet recommendation method (Master's thesis, Chang'an University).
3. Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35
4. McArdle, William D. (1986). Exercise Physiology (2nd ed.). Philadelphia: Lea & Febigier.[page needed]
5. “The Chinese dietary guideline (2022)” <http://dg.cnsoc.org/>
6. International Standard ISO 31-4: Quantities and units, Part 4: Heat. Annex B (informative): Other units given for information, especially regarding the conversion factor. International Organization for Standardization, 1992.
7. Berg J, Tymoczko JL, Stryer L. Biochemistry 5th. San Francisco, CA: W.H. Freeman. 2002: 603. ISBN 0716746840.
8. Hao, S.N., & Zhao, L.J.. (2017). An ElasticSearch-based recommender system architecture. Computer Knowledge and Technology, 36.
9. Divya, M. S., & Goyal, S. K. (2013). ElasticSearch: An advanced and quick search technique to handle voluminous data. Compusoft, 2(6), 171.
10. Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009.
11. Zhou Hua, & Yuan Zhi. (2017). Research on Cosine Similarity Weighted Slope One Collaborative Filtering Algorithm. Computer Science and Application, 7, 1036.
12. He, Jinchao, Luo, Fang, Yuan, Zhicai, & Huang, Huizhong. (2019). Application of collaborative filtering and particle swarm algorithm in dietary recommendation. Computer Applications and Software, 36(8), 36-40.
13. Li, J. B., Lin, S. Y., Hsu, Y. H., & Huang, Y. C. (2020). An empirical study of alternating least squares collaborative filtering recommendation for Movielens on Apache Hadoop and Spark. International Journal of Grid and Utility Computing, 11(5), 674-682.
14. Huang Dawei. (2020). Food and beverage recommendation system based on big data framework (Master's thesis, South China University of Technology).\
15. Meng P. (2016). Research on personalized diet recommendation service based on eating behavior (Master's thesis, Shaanxi Normal University).
16. Divya, M. S., & Goyal, S. K. (2013). ElasticSearch: An advanced and quick search technique to handle voluminous data. Compusoft, 2(6), 171.

# Appendices

[1] Food Nutrients Dataset: <https://www.cfs.gov.hk/sc_chi/nutrient/search1.php>

[2] Dietary Factors from 1989 & 2002 WHO:

|  |  |  |
| --- | --- | --- |
| Dietary factor | 1989 WHO Study Group  recommendations | 2002 Joint WHO/FAO Expert Consultation recommendations |
| Total fat | 15-30% | 15-30% |
| Saturated fatty acids (SFAs) | 0-10% | <10% |
| Polyunsaturated fatty acids (PUFAs) | 3-7% | 6-10% |
| n-6 PUFAs |  | 5-8% |
| n-3 PUFAs |  | 1-2% |
| Trans fatty acids |  | < 1% |
| Monounsaturated fatty acids (MUFAs) |  | By difference |
| Total carbohydrate | 55-75% | 55-75% |
| Free sugars | 0-10% | < 10% |
| Complex carbohydrate | 50-70% | No recommendation |
| Protein | 10-15% | 10-15% |
| Cholesterol | 0-300 mg/day | < 300 mg/day |
| Sodium chloride (Sodium) | < 6g/day | <= 5 g/day (< 2 g/day) |
| Fruits and vegetable | >=400 g/day | >=400 g/day |
| Pulses, nuts and seeds | >= 30 g/day (as part of the 400g of fruit and vegetables) |  |
| Total dietary fiber | 27-40 g/day | From foods |
| Non-starch polysaccharide (NSP) | 16-24 g/day | From foods |