MASTER OF TECHNOLOGY INTELLIGENT SYSTEMS

2020

PATTERN RECOGNITION SYSTEMS

FINAL PROJECT REPORT

Your Dietician

Team members:

Zhan Sheng
Wu Jingxuan
Li Kaitong
Lin Danmeng

Content

1. Executive Summary

With more attention drawn to the importance of diet in modern society, having a healthy diet has been a preference to people from all over the world. However, in many cases, people have no idea what kind of food they are supposed to take every day to keep themselves healthy. In order to enable users to have a clear understanding of their own physical condition and provide them with proper dietary advice based on their physical conditions and needs, this project Your Dietician was carried out.

Body mass index (BMI) is a value derived from the mass (weight) and height of a person.The BMI is a convenient rule of thumb used to broadly categorize a person as underweight, normal weight, overweight, or obese based on tissue mass (muscle, fat, and bone) and height..

In the whole process, four main functions have been designed for our project:

(1) Calculation of body BMI: users can choose directly input height and weight data or upload their facial photos to the website. BMI, which is a very important parameter to measure body fitness, will be shown.

(2) Calorie Bank: users can enter what food they have in their meal here, and the system will tell you the total number of calories. If the intake calories exceed the basal metabolic calories, the system will recommend relevant sports and exercise to consume them.

(3) Food image recognition: users can upload the images of food they have taken. Then they will be provided with the predicted name and corresponding calorie value.

(4) Recipe recommendation: recommend daily recipes for users according to their selections.

It is believed that the first three functions described above can help users understand their own physical condition and whether their current eating habits are appropriate. Then, recipes are recommended for users based on their needs. The completion of this project can solve the puzzles of most dieters. Similarly, it can also let housewives know whether the nutrition of three meals a day is insufficient.

Four of our team members also use the system to provide reference for our daily diet. In the process of implementing this project, we found that the biggest challenge was how to collect and analyze the information widely, so as to

provide the most appropriate suggestions for users.

Therefore, we decided to create our own dataset. In addition to using public datasets like food image and recipe dataset, the data was collected by using crawlers to obtain relevant data (human face) . After doing data filtering, the dataset was ready to realize the functions. Then, algorithms and models were used to learn in the pattern recognition course to train and debug different functions.

In the process of project implementation, many difficulties were encountered and some interesting ideas had been forced to give up. Although we are so tired, we still have gained a lot. Limited by the time and our personal ability, the project still has a lot of improving space. Maybe in the future after we explore more on the road of machine learning, we will back to fill in some gaps in this project.

2. Project Objective

This project covers three main objectives, which are to let the users:

1. Be aware of his own BMI. Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women. It is better for people knowing their physical situation before giving recipes suggestions.

2. Be aware of their calorie intake based on what they have eaten. Through the calorie value given by either food image recognition or manual food selection, users can get to know whether they have consumed too much or whether having this kind of food is a right choice.

3. Obtain recommended recipes based on their needs and selections.

3. Background & Introduction

The diet must conform to the characteristics of individual growth and development and physiological conditions, containing all kinds of nutrients needed by the human body. Why? Because the food can: 1) fully meet the needs of the body 2) maintain normal physiological functions 3) promote growth and development and health. The USDA guidelines recommend a large daily increase in the amount of cereals, breads, vegetables, fruits, and dairy

products and a significant reduction in the amount of meat and dairy products required.

In any corner of the world, diet must be one of the most important issues because it will have a significant impact on their daily activities. All people want to be healthy, but people don't know how to eat so that they can be healthy. For example: knowing little about their own physical condition, not knowing whether the current diet is lack of nutrition or not, not knowing whether the food in front of you is excessive in calories or not and what kind of food is fitness one, etc.

Considering that it is ridiculous to blindly recommend recipes to users just based on their physical condition and current eating food. So the recipe recommendation is done after understanding the user's needs (like building muscles and losing weight, they will contribute to completely different recipes).

Therefore, the system covers four functions to satisfy the objective, which is shown in figure 1 below.

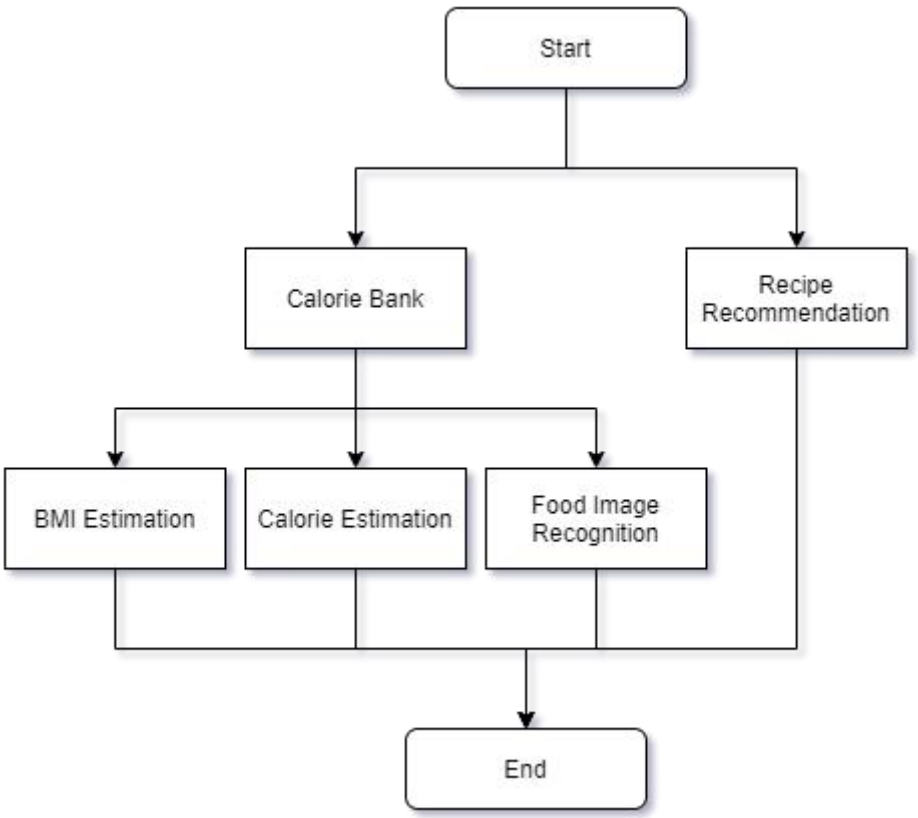

Figure 1: Main Process of the project

3.1. Calorie Bank

There are 3 functions inside the Calorie Bank section.

3.1.1. Calculation of body BMI

In this part, users can choose directly to upload the weight and height data or upload a facial picture to the website. If upload the data, the
the website will automatically give you the BMI result, which can make users understand more of their current body condition. If the users feel a little bit ashamed to input this private data, they can just choose one of their photos and upload to the system. The website can show the BMI, weight, height data prediction.

3.1.2. Calorie Estimation

In this section, users can enter their basic body information like age and weight, then obtain the calories they need according to the basic metabolic formula. Basic metabolic calories are divided by age, and calculated by applying different formulas based on weight.

In the interface, users can select different types and amounts of food they eat in a day, then obtain the total calories they intake through all the eaten food in a day. By comparing the number of calories intake and consumed by users, users will obtain recommendations for a reasonable diet to fill up the basal metabolic calories, or exercise recommendations to burn the excess calories.

3.1.3. Food Image Recognition

In this section, users can upload an image of food they have taken from the local disk. This system will do the prediction of what food users have uploaded. Then an estimated calorie value will be shown.

3.2. Recipe Recommendation

In this section, when users click the Recipe Recommendation link on the webpage, one message box will pop up. Users can choose what kinds of recipes they want on the selection menu based on their special needs. After clicking “OK” on the message box, it will show a selection report. 10 recipe recommendations will automatically show up on the webpage at the final stage.

4. Methodology

In this 4 different Python files were compiled to import into HTML pages. This section will be explained in detail in terms of functions.

4.1. Calculation of body BMI:

This function can work through 2 different ways, one is to upload a facial photo to the web page, one is to input the weight and height data directly.

4.1.1. Evaluating your BMI, weight and height data only with facial pictures

In the beginning, it is necessary to obtain the training and testing data set, which is collected by ourselves. Facial photos of celebrities from the Internet are downloaded and used for forming the evaluation model based on publicly available BMI, weight and height data, which formed our training and testing data set. Then, use facenet pretrained architecture to extract face embedding. The next process comes to the selection of different models among Linear Regression, Random Forest and Kernel Ridge Regression. Finally, compare the different model scores and accuracy, export the best one. The processing model is shown in figure 2 below.

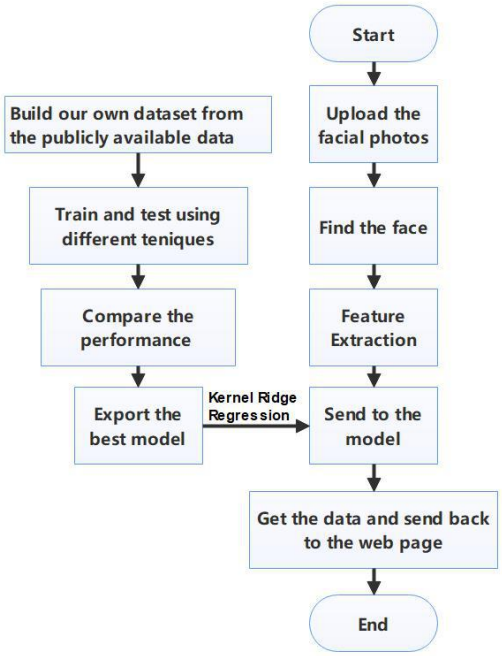

Figure 2: The processing model for body evaluation.

4.1.2. Obtaining the physical information with weight and height.

Using the SVM model to train the dataset (500 Person-Gender-Height-Weight-Body Mass Index data). So that this system have been able to divide the data point with height and weight into different cluster labels ('Malnourished', 'Underweight', 'Healthy Weight', 'Slightly Overweight', 'Overweight', 'Obese'). Now, it is possible to tell the users his physical condition according to the input data. Figure 3 below shows the working process of these functions.

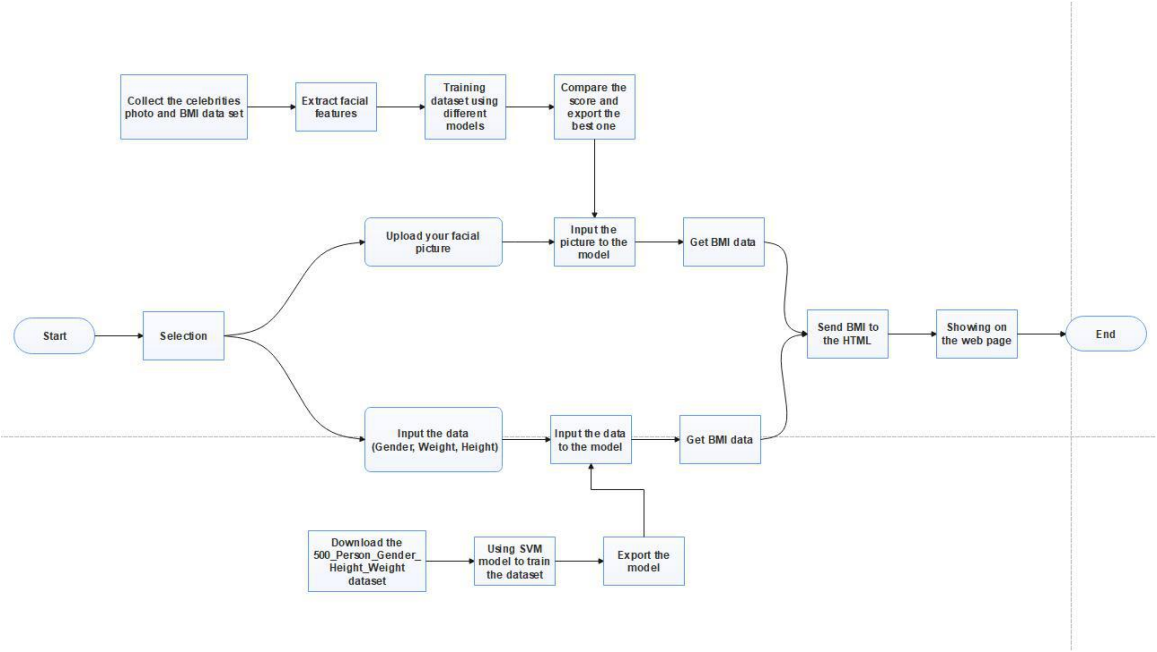

Figure 3: The working process of obtaining the physical information with weight and height.

4.2. Calorie Estimation

The calorie bank allows users to choose from 101 kinds of food, and see their calories. Get started by entering your age and weight to obtain the daily calories needed, then select the food to eat to calculate the total calories intake, finally view how to consume the energy.

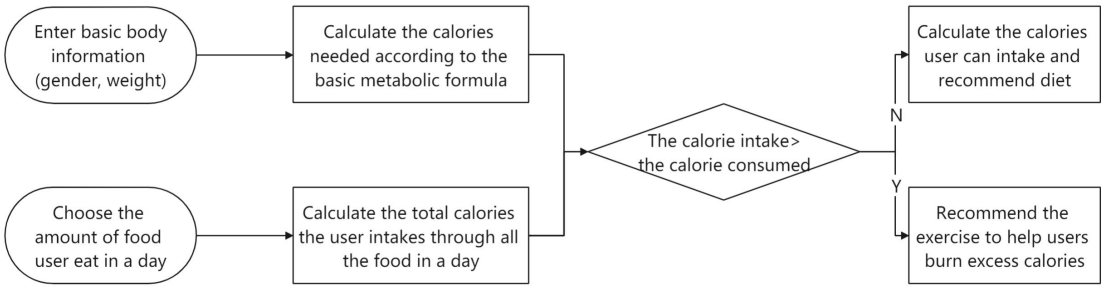

Figure 4: The working process of calorie estimation function

As shown in figure 4, the first part of input is the user inputs their own basic body information, age and weight, then the system calculates the calories needed according to the basic metabolic formula.

For male:
Age: 18-30 years old
basic calorie consumption (kcal)=[weight (kg)*0.063+2.896]*240
Age: 31-60 years old
basic calorie consumption (kcal)=[weight (kg)*0.048+3.653]*240
Age: over 60 years old
basic calorie consumption (kcal)=[weight (kg)*0.040+2.459]*240

For female:
Age: 18-30 years old
basic calorie consumption (kcal)=[weight (kg)*0.062+2.036]*240
Age: 31-60 years old
basic calorie consumption (kcal)=[weight (kg)*0.034+3.538]*240
Age: over 60 years old
basic calorie consumption (kcal)=[weight (kg)*0.038+2.755]*240

The second part of input is the user chooses the type and amount of food they intake in a day, then the system calculates the total calories the user intakes through all the food in a day.

By comparing the relationship between calorie intake and consumption, when the user's calorie intake is less than the consumption, the system will calculate the calories the user can still intake in a day and recommend diet, otherwise the system will recommend some exercises and sport to help users burn excess calories.

4.3. Food Image Recognition

The food image recognition section allows users to upload the image of food they have taken and then check the predicted food name and calorie value.

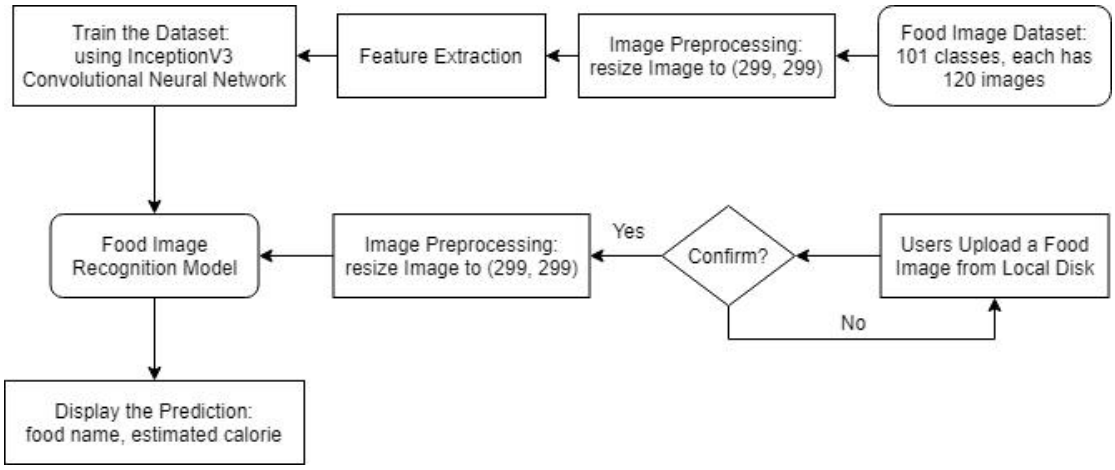

Figure 5: The working flow of food image recognition function

Figure 5 illustrates the working flow of this function. For model training, *food_image_dataset*, which includes 101 classes of food and each of them involves 120 food images, was created referring to the public food image dataset *food-101*. Before training the dataset, image preprocessing was done, resizing each image to the size of 299*299. InceptionV3 convolutional neural network was applied to train the image dataset, with 0.2 as the rate of validation split.

Users are allowed to upload a food image from their local disk. They can re-upload the image if they choose the wrong one. Then click the Confirm button to run the food image recognition model. Output generated from the model as the predicted name of this kind of food would be displayed on the user interface, along with its corresponding calorie value.

4.4. Recipe Recommendation

The recipe recommendation section includes data training and selection presentation. It uses Python as the backend programming language.

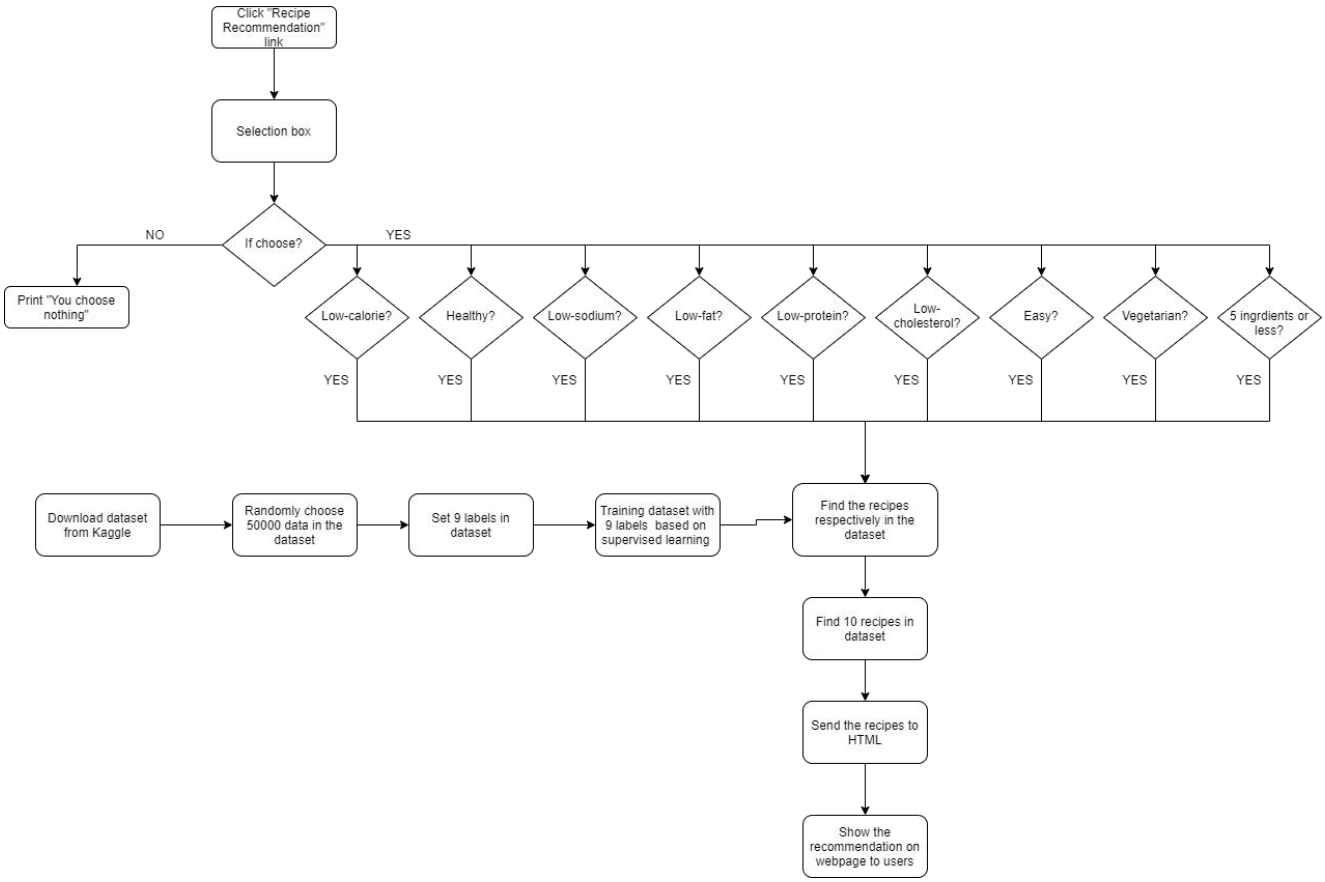

Figure 6: The working flow of recipe recommendation function

Figure 6 illustrates the flow chart of the recipe recommendation working process. In the data training, the data was randomly selected 50000 from Kaggle Food.com Recipes and Interaction dataset [4]. After obtaining the dataset, the “nutrition” data was divided into several parts according to their specific nutrition names, such as calories, sugars, sodium, etc. Supervised learning methods including K-Nearest Neighbors (KNN), Naive Bayes (NB), Neural Network (NN) and Logistic Regression (LR) were used to train the data with labels. The labels were set as “healthy”, “low-fat”, “low-calorie”, “low-sodium”, “low-protein”, “low-cholesterol”, “easy”, “5-ingredient-or-less”, “vegetarian”.

In the selection presentation section, the labels were set as options in the selection box, users select what kinds of recipes they want to get on the selection box. After selection, the result will be sent back to python to check which label has been chosen and search the corresponding recipe in the dataset. 10 recommended recipes result will show up on the webpage after obtaining the corresponding recipes from the recipe dataset.

5. Result & Analysis

5.1. Calculation of body BMI

The Calculation of body BMI contains two functions: the first one is evaluating the BMI, weight and height data only with facial pictures, and the second one is obtaining the physical information with weight and height.

5.1.1. Evaluating the BMI, weight and height data only with facial pictures

After face-features extraction using the library face_recognition, the dataset has been trained and tested applying 4 pattern recognition techniques (Linear Regression, Ridge Linear Regression, Random Forest, Kernel Ridge Regression). And considering about the mean squared and the variance score, Kernel Ridge Regression model performs the best. The process of the comparison between models and part of facial photos collected by ourselves are shown in figure 7 and 8 respectively below.

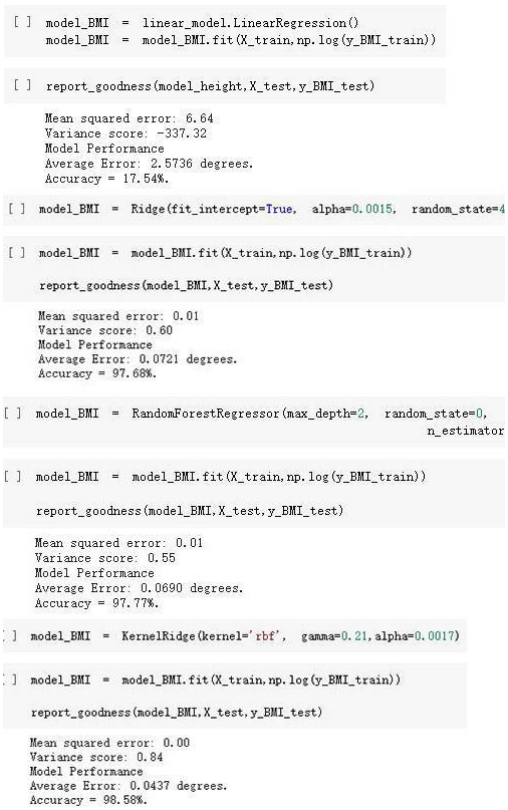

Figure 7: The comparison code between models

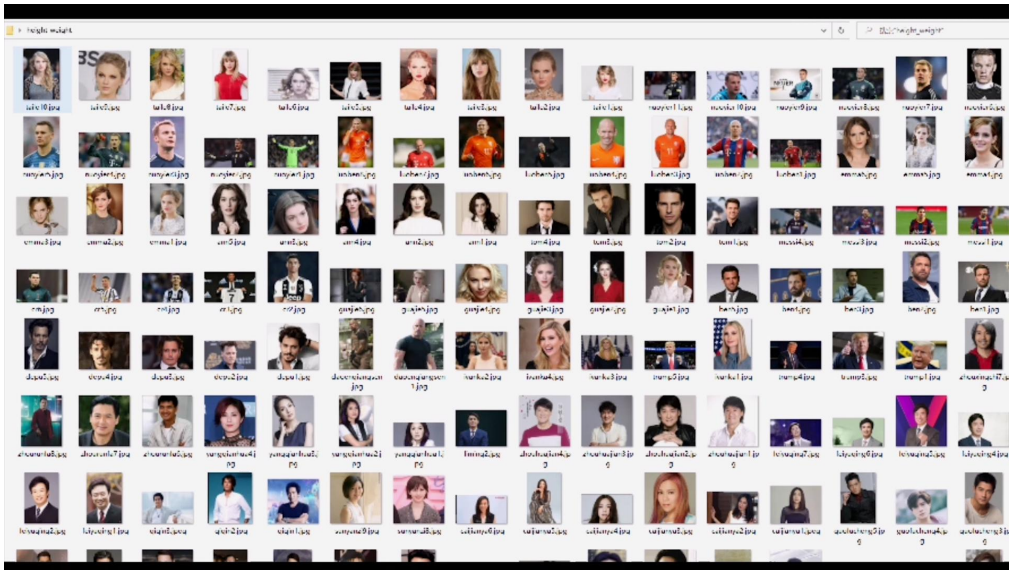

Figure 8: Part of facial photos

5.1.2. Obtaining the physical information with weight and height.

According to the dataset 500-person-BMI, SVM was used to train and export our model. The figure 9 below shows the classification report.

```
In [30]: print(classification_report(y_test, prediction))
```

	precision	recall	f1-score	support
Healthy Weight	0.73	0.80	0.76	10
Malnourished	0.00	0.00	0.00	3
Obese	0.88	0.97	0.92	36
Overweight	0.96	0.78	0.86	32
Slightly Overweight	0.78	0.93	0.85	15
Underweight	0.60	0.75	0.67	4
accuracy			0.85	100
macro avg	0.66	0.71	0.68	100
weighted avg	0.84	0.85	0.84	100

Figure 9: The classification report of SVM model

5.2. Calorie Estimation

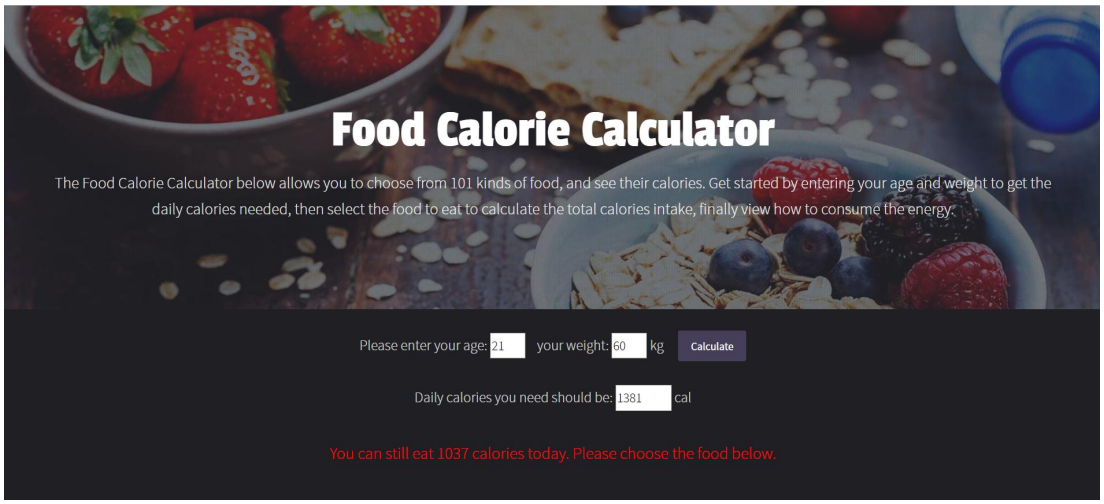

Figure 10: The food calorie calculator web page

As shown in figure 10, this section calculates the user's daily basal metabolic rate by entering the user's age and weight and applying the previous formula, and presents it on the website.

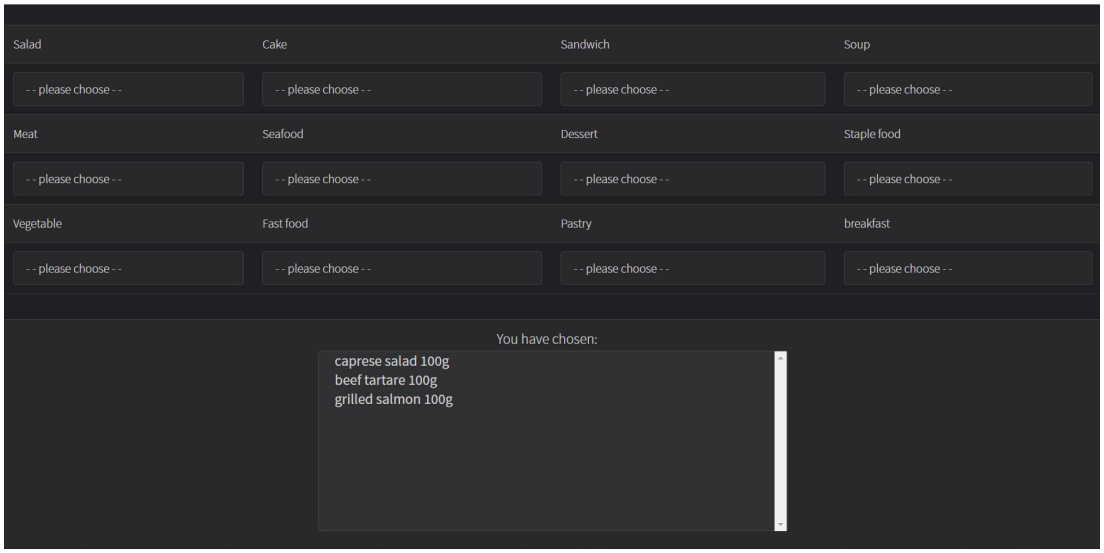

Figure 11: The food options and presentation

As shown in figure 11, this section is for users to choose the food they eat in a day according to different types and amounts, and the chosen options are presented in the below chart. The food library is composed of 101 kinds of foods. These foods are divided into 6 categories according to the types of ingredients, which are convenient for users to find and add.

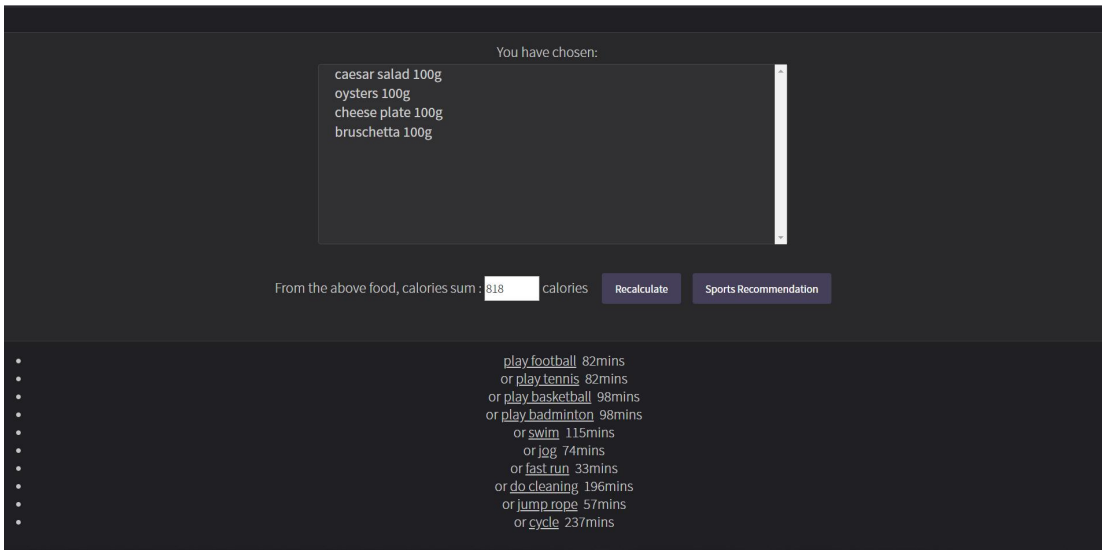

Figure 12: The sport recommendation presentation

As shown in figure 12, after the user has input all the information above, the system will query the calories per unit weight of the food and collect it in the database corresponding to each food. When the user selects a specific food and weight, the system automatically sums up the calories of the selected food. If the total intake of calories exceeds the basal metabolic rate, the system will list several sports and exercising time to help consuming excess calories.

5.3. Food Image Recognition

The dataset applied consists of 12,120 images in total. The proportion of validation split is 0.2. So the training dataset consists of 9696 images and the validation one involves 2424 images. InceptionV3 convolutional neural network was used to train the dataset.

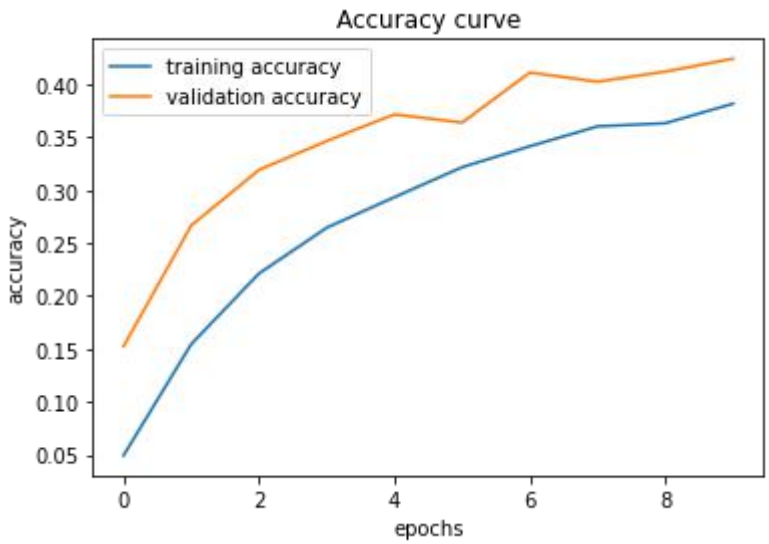

Figure 13: The Accuracy Curve for training and validation

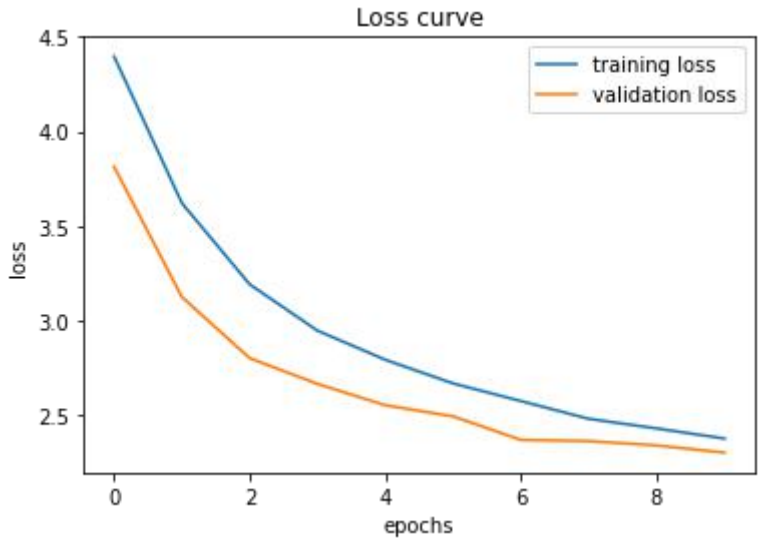

Figure 14: The Loss Curve for training and validation

Figure 13 and figure 14 are the Accuracy Curve and the Loss Curve of this model. It is quite obvious on the curves that the model is becoming more accurate and hasless loss with the number of epochs rising. Although the highest accuracy among all these is only about 0.5, the actual prediction is quite satisfactory through experiments. To better improve the image recognition, a larger dataset is needed in this model training, as there are only 120 images for one class of food in the current dataset.

5.4. Recipe Recommendation

Data has been trained in 4 main methods based on supervised learning, such as K-Nearest Neighbors (KNN), Naive Bayes (NB), Neural Network (NN) and Logistic Regression (LR). For each method, the data was set as a training set and a testing set separately. The ratio of the two datasets is 5. The labels were set in 9 different aspects according to the nutrition of each recipe, including calories, total fat, sugars, sodium, protein, stated fat and total carbohydrate. Each label was trained and tested separately.

According to the accuracy score of the model and the classification report, the NB, NN and LR models perform better in prediction, and the predictions for all labels are almost 100% accurate. If the KNN model is used, it is difficult to find the most suitable number of neighbors within the specified range.

```
max_test_score = max(test_scores)
test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_score]
print('Max test score {} % and k = {}'.format(max_test_score*100,list(map(lambda x: x+1, test_scores_ind))))
```

```
Max test score 78.25 % and k = [13]
```

Figure 15: The testing accuracy based on KNN model of Low Protein label

```
predictions = mlp.predict(X_test)

from sklearn import metrics
print("Accuracy", metrics.accuracy_score(y_test, predictions))
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))
```

```
Accuracy 0.9998
[[7513    0]
 [   2 2485]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7513
           1       1.00      1.00      1.00      2487

    accuracy                           1.00     10000
   macro avg       1.00      1.00      1.00     10000
weighted avg       1.00      1.00      1.00     10000
```

```
[ ] print("Accuracy on training set: {:.3f}".format(mlp.score(X_train, y_train)))
    print("Accuracy on test set: {:.3f}".format(mlp.score(X_test, y_test)))
```

```
Accuracy on training set: 1.000
Accuracy on test set: 1.000
```

Figure 16: The training and testing accuracy based on NN model of Low Protein label

Figure 15 and 16 are examples of the accuracy result based on the same Low Protein label. The first figure shows the accuracy of the testing result which is not as good as what was testing based on the NN model. In the NN model, both the training set and testing set are almost 100% accurate.

6. Limitations & Future Works

Due to the shortage of time, some models in our project were not trained so completely to face all situations.

(1) The BMI calculation by facial photos does not behave so well when uploading a child photo, because one of our libraries' face_recognition is only suitable for the adults in its official introduction.

(2) The food image recognition can not recognize some unknown food because of the limitation of GPU.

(3) BMI was the only body index to measure a person’s body condition in this

project. More factors need to be considered in order to understand users' physical condition better.

If we had a longer time frame to work on this project, we would have worked upon the following points of improvement:

1) Dataset. Since the time frame was short, some of the training and testing dataset might have been a little small. We will choose and collect more dataset to obtain a better result in the next iteration.

2) Website GUI: Actually, this is the first time for us to design a website, so next time more attention will be paid on how to make the users feel and look better.

3) More details: In the next version, more factors will be added to measure users' physical conditions. Furthermore, the users will obtain more kinds of information from the Calories Bank and the Food Recognition part.

7. Conclusion

The whole team spent a great time on this project. From this period, there is no doubt that we got the chance to exercise and put the knowledge of machine learning we have learnt into practice. In this project, we applied various techniques we learned from the Pattern Recognition courses, such as SVM, CNN, KNN, Naive Bayes and Logistic Regression. Although we had no similar work experience before, every one of this team was able to independently design functions and eventually integrate them into one web version system. It was absolutely a memorable experience. We learned a lot from each other in machine learning. Overall, we spent more than ten days on the project. Maybe in the future, when it is time for us to head into the workplace, this experience is still worth mentioning and remembering.

References

[1] 廖恩红,李会芳,王华,庞雄文.基于卷积神经网络的食品图像识别[J].华南师范大学学报(自然科学版),2019,51(04):113-119.

[2] Meng Chen, Xiaoyi Jia, Elizabeth Gorbonos, et al. Eating healthier: Exploring nutrition information for healthier recipe recommendation. 2019.

[3] de Ridder Denise,Kroese Floor,Evers Catharine,Adriaanse Marieke,Gillebaart Marleen. Healthy diet: Health impact, prevalence, correlates, and interventions.[J]. Psychology & health,2017,32(8).

[4] Shuyang Li, Bodhisattwa Prasad Majumder, Kaggle, Food.com Recipes and Interactions[online], URL: https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions

[5] Bidani, Shiv,Priya, R. Padma,Vijayarajan, V.,Prasath, V.B. Surya. Automatic body mass index detection using correlation of face visual cues[J]. Technology and Health Care,2019,Pre-press(Pre-press).