MASTER OF TECHNOLOGY PROJECT REPORT

A Restaurant and Food Recommendation System

TEAM MEMBERS

A0231486A: YI CHEN

A0231374L: HUANG JIAHAO

A0231482L: SHAO ZHERUI

MASTER OF TECHNOLOGY

Contents

1 Business Information

1.1 Business Background

Nowadays, people have a higher and higher pursuit of quality of food and drink. Especially, more and more people have a personal and special preference for diet choices. Facing a large number of restaurants, there is a strong potential demand to recommend them to the customers accurately and efficiently, which could help customers to enjoy the diets and avoid wasting on what they do not like.

When the customers select a recommended restaurant, they continue to order what is very probably delicious for them. In the meantime, food balanced nutrition is also getting more and more attention. Therefore, our system will recommend reasonable and healthy food based on customers' information, such as gender, fitness times and personal objectives.

With obesity on the rise, many people are trying to eat healthily or meet personal weight goals. Therefore, food recommendation has great potential. By collecting user information, selecting and recommending well-matched nutritious diets based on user characteristics, it can help dieters who want to control their daily energy intake to plan their diets very well. It saves users time, learning costs and improves the efficiency of their diet planning.

1.2 Business Case

Our system focuses on 2 parts of the customers eating. Firstly, recommend the restaurants for the customers based on their rating records, which aims to provide the restaurants most likely to satisfy them and meeting their diet preference. Besides,

Our Good Meal aims to provide a better and healthier eating experience by quickly and accurately planning next meal for users.

GoodMeal uses the personal health information entered by users to set different nutritional goals, change the recommended rules, and select a variety of food categories from the database to recommend to users.

For example, users of different ages have different requirements for each nutrient, and their energy and protein requirements can vary significantly depending on their personal exercise habits. In addition, for people who want to lose fat and weight, the intake of carbohydrates and fats will be significantly reduced.

After the completion of the formulation of nutrition goals, it is necessary to set the rules of screening, using the food nutrition composition in the database as constraints, after planning, o get the combination of specific food, and to show to users.

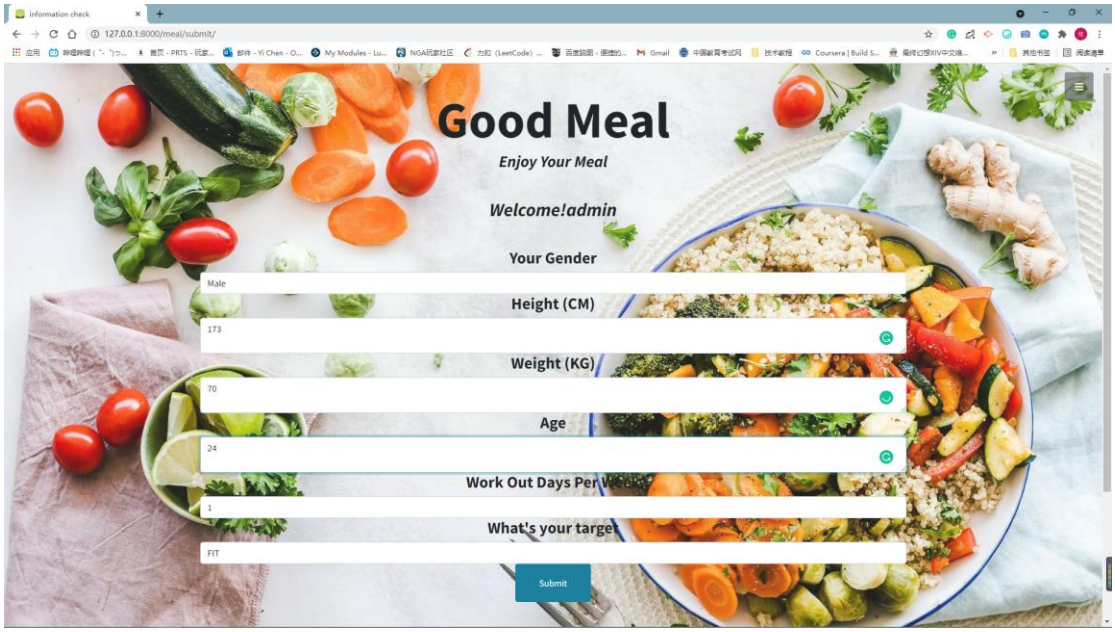


Figure 1 Food recommendation collecting data

1.3 Financial Evaluation

1.3.1 Cost

The cost of our system includes several aspects below.

1. Collecting data. The demand of data of restaurants ratings and order ratings is high and increasing for accurate recommendation. However, to some extent these data are commercial secrets costing much.
2. Update software. Especially, the GUI of our APP is simple at present, which will be re-designed, implemented and updated in the future.
3. Rent of server. The user information and order records will be saved using Ali cloud, each recommendation is calculated and done by combination of remote servers.
4. Development environment construction. There may need to pay extra money to set up the development environment, use front-end frameworks or static materials during development can also cost.

1.3.2 Profit

In the early stage, our system is free to use, promoting the system and collecting more extensive data for subsequent modifications of our models.

Then after some improvements have been done, such as better APP and computing device, we will charge users for some a specific number of uses, provide subscription service and charge corresponding advertising fees.

1.4 Product Plan

Currently, our system is able to recommend appropriate restarants based on cutomers' order history and recommend food based on customers' input information, respectively.

In fact, we wanted to perform the 2 related results together, which means recommend the food in the restaurant recomended. It failed due to there are not suitable data we can collect. Specifically, customer-restaurant-order data usually do not contain the actual food components of each dish, so it is impossible to evaluate the nutritional balance. Therefore, we are willing to collect new data to achieve original plan by spending more energy and money.

Besides, we decide to develop and improve the GUI iteratively in the future.

1.5 Market Research

At present, there are many softwares provide similar recommendation, such as Google Search, Google Map and HungryGoWhere, etc, which is not personalized enough. Compared with them, the business mode of our recommendation system is more thorough and more targeted.

The characteristic is our disadvantage as well, our application scope and function is small, therefore, we think it may be necessray to cooperate with other APP in the early stage. At the same time, we must understand customers requirement and extend our functions.

We used SWOT analysis to better understand the competition in the market.

Figure 2 SWOT analysis

Overall, compared to other competitors. GoodMeal focuses on recommendation and pays more attention to accurate recommendation. The goal is to create richer and more personalized precise recommendations for users.
In contrast, the project still has a lot to be desired in its early stages. For example, the technical aspects are not fully mature and the results of recommendations are not completely satisfactory. However, in the subsequent updates, deficiencies will be gradually improved and competitive advantages will be expanded.

2 System Design

2.1 User Case Design

At current stage, our initial system includes 2 main user cases.

For Customers, they could register after admin check and store, then they can login the system. They can get the recommendation result and sorting scores of restaurants, and the recommendation and analysis result of food as well.

For admin, he can check and store the users' information in database, and manage the food, restaurant and order records.

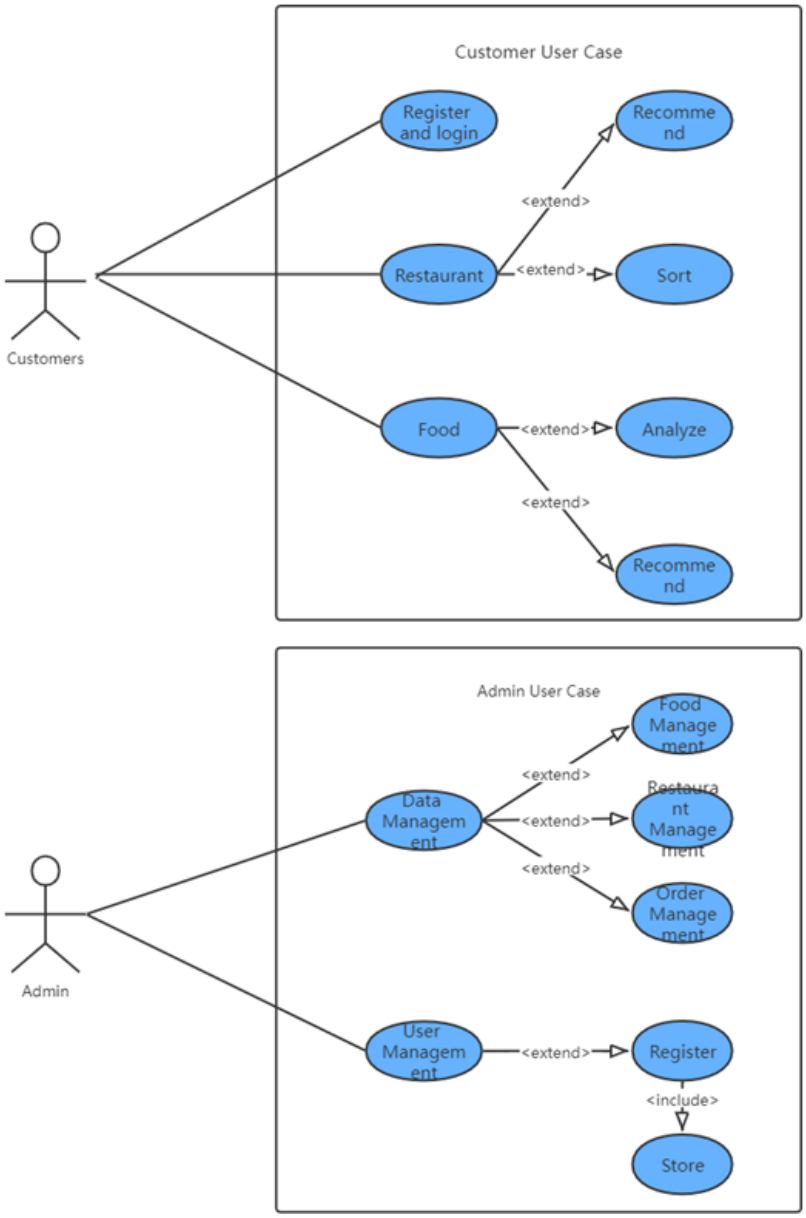


Figure 3. User Case

2.2 Key User Cases

Table 1. Recommend Restaurant

Use Case Name	Recommend Restaurant
Summary	Recommend some restaurants based on users' rating history.
Basic Flow	1. Query whether it is a new user.

	2. If so, recommend current restaurants with high rating. 3. Else, calculate the similarity with other users. 4. Predict the user's ratings for all restaurants . 5. The system shows the results.
Business Rules:	System recommends based current data.

Table 2. Recommend Food

Use Case Name	Recommend Food
Summary	Recommend foods based on users' personal health information.
Basic Flow	1. Check whether the user is logged in. 2. The Users enter personal health data. 3. The system will Generate nutritional requirements according to rules. 4. The system will call the ortools to get the result of Constrained optimization question. 5. The system shows result .
Business Rules	System recommends based on user data and database.

2.3 Structure Design

Python Django is used to develop and execute our system. It connects the database and Knowledge Engine in back-end, and shows the information in front-end.

The Django framework of Python used in the system design adopts the classic MVT mode, as shown in Figure, where M represents the model, corresponding to the data access layer, which performs all access and verification operations related to the data table; T stands for template, corresponding to presentation layer, which is the page that

users can operate between; V represents the view, that is, the business logic layer, which contains the logic related to accessing the model and accessing the appropriate templates.

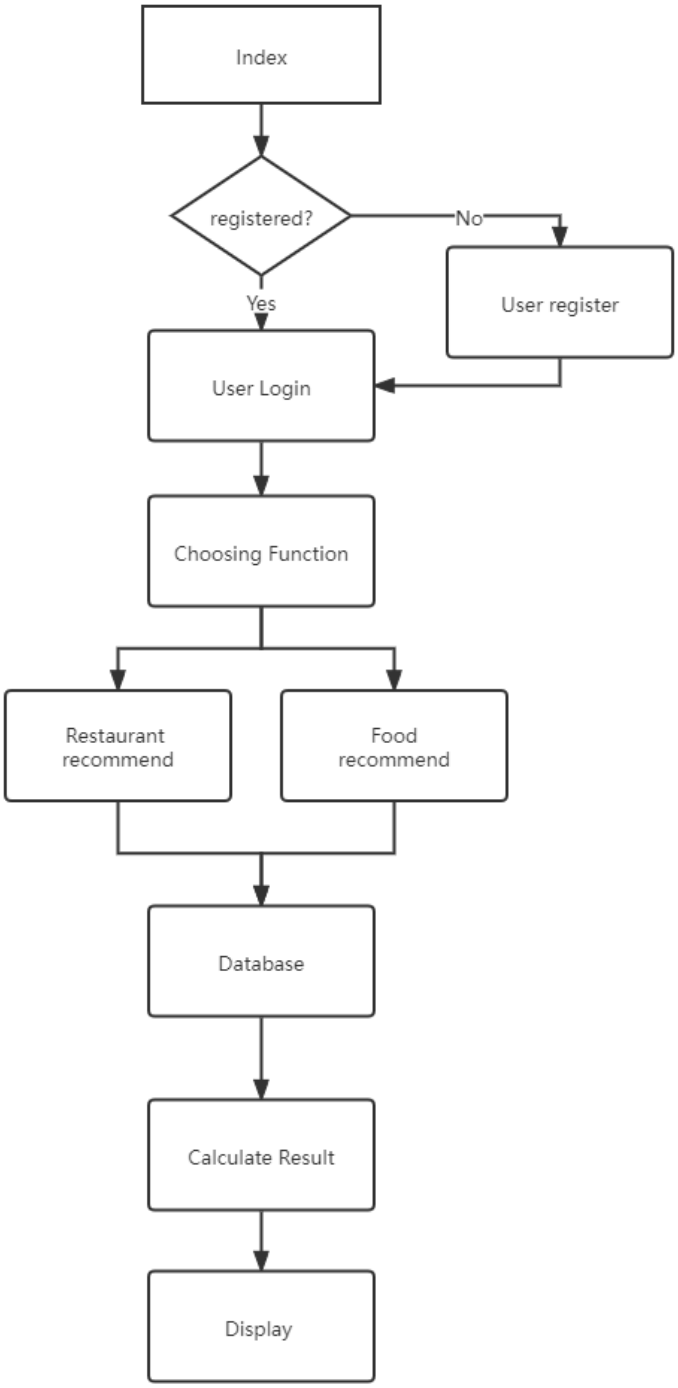


Figure 4. Structure Design

And the whole system strcture is showed as figure4.

3 Module Design

3.1 Restaurant Recommendation

3.1.1 Data Introduction

For the restaurant data, there are 2 parts, representing different relationships.

One includes the records of customers, restaurants and other information. The total number of its features are 73, only 10 are selected for our system after searching related knowledge on the Internet. The other is the record of orders of customers, the gross number of features are 26, finally 6 of them are chosen for our system after considering and discussing.

Table 2. Transaction Records

Column/Feature	**Meaning**
customer_id	Each customer's unique identification
gender	Customer's gender
location_type	Customer's location in current record
id	Each vendor's unique identification in reality
vendor_category_en	Vendor's cuisine category
delivery_charge	Delivery charge in current record
serving_distance	Serving distance from vendor to customer
prepration_time	Preparation time for food
vendor_rating	Vendor's mean rating
vendor_tag_name	Vendor's primary tag information

Table 3. Order Records

Column/Feature	Meaning
customer_id	Each customer’s unique identification
grand_total	Total cost of the order
vendor_rating	Order’s specific rating in reality
delivery_time	Delivery time of the order
vendor_id	Each vendor’s unique identification
LOCATION_TYPE	Customer’s location in current order

However, we need to cooperate with other APPs to get the practical data in reality, which would be more complicated.

3.1.2 Data Processing

Analyze the above data, we rechon that customer and restaurant can be the ID together. And for the design of this part, the fundamental features here are the restaurant ratings and customers' mean rating for each restaurant, which constructs the values of subsequent rating matrix.

Thus, the 2 datasets are merged based on identical ID defined above. Then the new data merged are saved as initial data for restaurant recommendation, which have 14 features totally.

3.1.3 Initialization for Collaborative Filtering

3.1.3.1 Rating Matrix

For our current stage, the basic ratings are used. These records are grouped by customer and restaurant, leading to all the records of a restaurant with the same

customer are collected together and we get the customers' mean rating for each restaurant.

Then we displayed the matrix and found it was a sparse matrix, there were a large number of nan values. In order to facilitate subsequent calculations, it is acceptable to fill them with rating 0.

3.1.3.2 Restaurants Mean Rating

For the new customers, they have no records in our system. We adopt a simple idea to solve the cold-start problem here, just calculate the mean ratings of our current data for each restaurant, sort and save the result in class.

3.1.3.3 Similarity Matrix

We have implemented 2 types of calculation methods of similarity, respectively. The first one is cosine similarity, the other is pearson coefficient similarity. Both are generated based on the Rating Matrix.

It is relatively easy and convenient to calculate, save and apply the cosine similarity. But if the difference of data is huge, the cosine similarity would be not accurate. For the other choice, Pearson Correlation Coefficient is good at handling the diversified data and eliminating the negative impact of data bias, but it costs much more time and memory. Comprehensively, cosine similarity is our default choice and Root Mean Squared Error is the loss function.

3.1.4 Recommend Using Collaborative Filtering

For the existing customers, their ratings to all restaurants are extracted to be further processed. Through setting the number of neighbours, the top n similar customers are

selected for current customer to predict the ratings.Then for the restaurants they have bought and rated, the ratings would be kept if it is allowed to recommend the repeated vendors, otherwise the ratings would be assigned 0, which means a pretty low priority. Finally, get all the vendors sorted with ratings in descending order.

For the brand-new customers who are not recorded in our data, the strategy adopted is to get all the restaurants sorted with ratings in descending order as well.

The last operation for both typles is to get related tag information from the original data, and match the tag and corresponding restaurant id, and return the predicted rating.

3.1.5 Application for Collaborative Filtering

We try our best to reuse the existing class and function in this part. After the initialization of user-based collaborative filtering, the input should contain customer id, recommendation number and neighbor number. And the output calculated and returned includes the restaurant id, restaurant's mean rating, restaurant's tag and predicted rating of current customer.

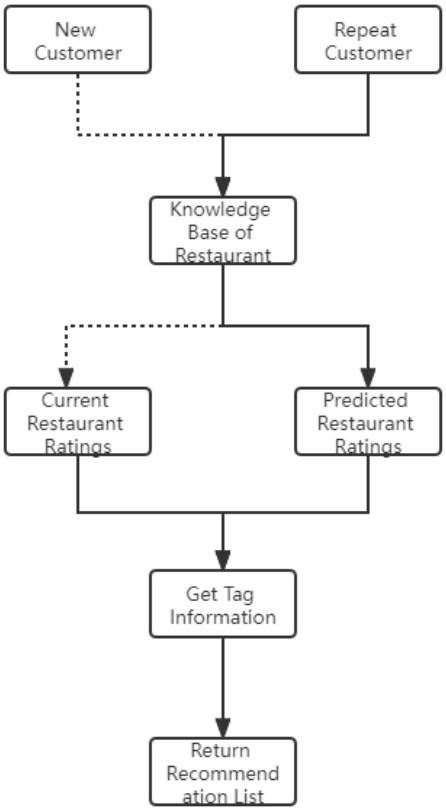


Figure 5. Recommendation Mode of Application

3.2 Food Recommendation

3.2.1 Data Introduction

We selected a few features from a huge food dataset OpenFoodFact, which got more than 40 columns. And reduce the number of food to 2k .
We also collect User data as the input of food recommend, those tables are showed below:

Table4. FoodDatabase

Column/Feature	**Meaning**
Food Name	Each food's name
Food Group	Food's Group, like fast food, egg product, vegetable product …
Food Sub Group	Sub group to make food divided more accurate
Per Serving Household Measure	Weight of food per meal or container size
Cuisine	The style of cooking
Sugars (g)	Nutrient content per serving
Sodium (mg)	Nutrient content per serving
Carbohydrates (kcal)	Nutrient content per serving
Fats (kcal)	Nutrient content per serving
Protein (kcal)	Nutrient content per serving
Total Calories (kcal)	Nutrient content per serving

Table5. UserData

Column/Feature	**Meaning**
Gender	User's Gender(Male or Female)
Height	User's Height (CM)
Weight	User's Weight(KG)
Age	User's Age (years)
Work_days	The number of days that user workout
Muslce	Whether User's target is to gain muslce

3.2.2 Data Processing

We found that the values of some features in the food data were missing, so we cleaned the data and found that many foods have the same name but different portions, and they can be combined into small portions.

And for the user's data, we coded for them to convert text into number.

3.2.3 Initialization for Constrained optimization

3.2.3.1 Knowledge Representation

In the food recommend part we try to extract rules from specific specialized nutrition books, but find it too hard. So we simplify the task by referencing some known nutrition rules.
For example:
Calculate basal metabolic rate (BMR), which is the total number of calories your body needs to perform basic, life-sustaining functions. These basic functions include circulation, respiration, cell production, nutrient processing, protein synthesis, and ion transport.
It uses gender, height, weight, age as input:

Male: BMR = (66.47 + (13.7 * weight) + (5.0 * height) - (6.8 * age))
Female: BMR = (655.1 + (9.6 * weight) + (1.8 * height) - (4.7 * age))

On top of that, we introduced activity intensity to better calculate the amount of energy required each day.
It's work_day to represent the physical activity level:
When work_day is less than 1 days per week, then the level will be low.
When work_day is less than 3 days but more than 1 day per week, then the level will be normal.
When work_day is less than 5 but more than 3days per week, then the level will be high.
When work_day is more than 5 days per week, then the level will be very high.

Table6. Level and Energy

No	Level	Daily energy required
1	low	BMR*1.2
2	normal	BMR*1.375
3	high	BMR*1.55
4	very high	BMR*1.725

3.2.3.2 Google Ortools

When we have the knowledge of nutrition, and can calculate the daily energy requirements and the amount of various nutrients. We can turn the question into a constrained optimization question, which is o the basis of meeting energy requirements, the existing food database is used to select foods and keep other nutrients within a reasonable range. Optimization is based on these constraints.

So we import google ortools library to solve the question.As it describe in the Google website, OR-Tools is an open source software suite for optimization, tuned for tackling the world's toughest problems in vehicle routing, flows, integer and linear programming, and constraint programming.

After modeling your problem in the programming language of your choice, you can use any of a half dozen solvers to solve it: commercial solvers such as Gurobi or CPLEX, or open-source solvers such as SCIP, GLPK, or Google's GLOP and award-winning CP-SAT.

3.2.3.3 Application of Constrained optimization

Most optimization solvers work the same way: define variables, define constraints, define objective functions, call solutions, and see the results.
We followed the steps by:

1. Defining each food as a variable and set their Coefficient
2. Using nutrients table to define constraints
3. Define objective functions set to minimization
4. Call solver to create solutions
5. Output solutions value

And one last thing is to display the result.

3.3 Input and output

Table 7. Input and Output

No.	input	output
1.Login	Account & Password	Login status
2.Restaurant	Login status (User id)	Restaurant list
3.Food	User information	Different type foods

4 System Development & Implementation in tools

The development environment of our system is Windows 10, the main tools are PyCharm and Colab, the framework is Django and the programming language is Python. For the process of training the models, it is recommended to equip the Nvidia GPU, using CUDA to speed up the training and testing of the models. Sometimes, remote training is another choice if only the integrated GPU is available, and Colab does a good job. The specific information is shown below.

Table 8: Develop environment.

Name	Version
Micosoft Windows	10
JetBrains PyCharm	2019.3.1
Google Colaboratory	
PyTorch	1.6.0
Django	3.2

5 Future Improvements

We tried our best to develop the system in limited time. However, there are still a lot of things we could do to pursue a better performance.

1. Collect more reasonable data. The initial restaurant data of our system do not include the specific food information, leading to we cannot connect the restaurant and food recommendation together. Currently, we just do them separately.
2. Update the GUI. The GUI of our system is simple right now, lacking of the interactivity with actual users, which needs more interesting designs.
3. Modify the recommendation mode. A basic user-based collaborative filtering is applied for restaurants at preset, however, we are still considering add weights of current customers' records based on the optimized management of database. Furthermore, the item-based idea may be also used here.
4. The food recommendation part is quite simple right now， require more knowledge to modify the rule and using more constrains to make more flexible recommendations. Change the solution to output meals for 1 week is a good direction for this part.

Reference

[1] Google Ortools Documentation, 2021, https://developers.google.com/optimization

[2] Django Documentation, 2021 https://docs.djangoproject.com/en/3.2/

[3] openfoodfacts-python , OpenFoodFact, 2021 https://github.com/openfoodfacts/openfoodfacts-

[4] jiajiewang0326, KG_Based_Recommendation_with_GNNs, 2019 https://github.com/jiajiewang0326/KG_Based_Recommendation_with_GNNs

[5] Bootstrap Documentation, 2021 https://getbootstrap.com/docs/5.1/getting-started/introduction/

[6] Asadoughi, mealplanner, 2015, https://github.com/asadoughi/mealplanner

[7] Arimai, meal-planner, 2017 https://github.com/asadoughi/mealplanner

APPENDIX OF REPORT A

Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal:
17/9/2021

Project Title:
A Restaurant and Food Recommendation System

Group ID (As Enrolled in LumiNUS Class Groups):
Group 11

Group Members (Name , Student ID):

No	Name	Student ID
1	Yi Chen	A0231486A
2	Huang Jiahao	A0231374L
3	Shao Zherui	A0231482L

Sponsor/Client: *(Company Name, Address and Contact Name, Email, if any)*
Academic Self Sponsored Project

Background/Aims/Objectives:
Our system tries to recommend reasonable restaurant, fulfilling customers' taste preference. In the meantime, our system will recommend healthy food based on customers' information, such as gender, fitness times and personal objectives.

Project Descriptions:

Data set:

1. Our Restaurant data comes from Restaurant Recommendation Challenge in Kaggle mainly. The Data Scientists collected a large amount of expenses records, including customers information, vendors information and orders information, etc.
2. Our food data comes from Open Food Fact ,which is a public data source from github. The data collected huge amount of food, including food's group, ingredients, allergens, nutrition facts and all the tidbits of information.

Software Resource Considered:

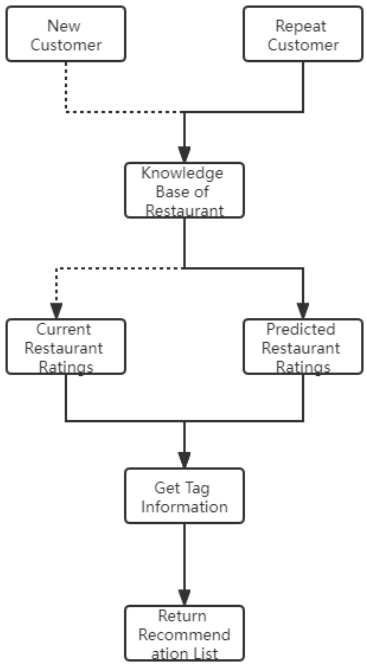
· Numpy – Statistical and Scientific Computing
· Pandas – Open source data analysis and manipulation tool
· scikit-learn – Simple and efficient tools for predictive data analysis

· Cv2 – Computer Vision library
· Dlib – Predictor and detector library
· Imutils – Utility library for object landmarks
· Matplotlib – Graphing library
· Pytorch – An optimized tensor library for deep learning

Used skill:

Knowledge Base, user-based Collaborative Filtering, Constrained optimization

Dialog flow:



APPENDIX OF REPORT B

Mapped System Functionalities against knowledge, techniques and skills of modular courses: MR, RS, CGS

Business Rules

We use knowledge-based inference engine to calculate the BMR, and Daily energy requirement based user data and other parameters.

Informed search :

We use ortools to set constraints, and perform informed search optimization to recommend the most appropriate foods.

Knowledge discovery & data mining techniques

For the restaurant recommendation, we build a user-based Collaborative Filtering as knowledge discovery engine to reason. And calculate the Rating Matrix, Cosine similarity and Pearson Coefficient Similarity to mine the features and knowlegde from the data.

APPENDIX OF REPORT C

Individual Reports

Your Name:	HUANG JIAHAO (A0231374L)

1. personal contribution to group project

As a member of the team, I participated in many parts of the project. The main task is to build the Collaborative Filtering for Restaurant Recommendation.

1. I designed part of the system structure and all the user cases.
2. I preprocced, cleaned and select the restaurant data.
3. I implemented Restaurant Mean Rating, Rating Matrix, Cosine Similarity and Pearson Coefficient Similarity.
4. I set the neibour number, determined a cold-start way, and developed Collaborative Filtering successfully.
5. I reconstructed and provided an interface for application, assisted YI CHEN and SHAO ZHERUI to run the part in Front-End.
6. I designed the categlog/format of report.

2. what learnt is most useful for you

I have learnt the importance and difficulty of combining theory with practice.

It is hard to extract and build Knowledge Base from dataset, and the data preprocessing is extremely flexible depending on practical situation. And I have more understanding of the recommendation system, especially the Collaborative Filtering algorithm. Its performance hinges on the data seriously.

I have enriched my technical knowledge skills, increased a lot of experience of Reasoning System and Knowledge based reasoning techniques.
Beisdes, I learnt kind of how to use Python's Django framework from YI CHEN.

3. how you can apply the knowledge and skills in other situations or your workplaces

I realize I have forgot many knowledge taught by professors, because I do use them rarely. And terribly, I have a few problems when I want to practice using some Knowledge Base techniques.

Therefore, I need to focus on review and try to practice in suitable senarios as much as possible.

Individual Reports

Your Name:	YI CHEN (A0231486A)

1. personal contribution to group project

The main task of mine is to build the Food Recommendation and the system framework based on Django.

1. I designed the system structure and use MVC parttern to develop.
2. I cleaned and select the food data from Open Food Fact.
3. I implemented business rules and Constrained optimization in food recommend.
4. I designed and developed the front-end of the system.
5. I completed the system design with teammates.
6. Deploying the system, and conducted testing and maintenance work

2. what learnt is most useful for you

I tried to combine the knowledge I learned with the actual development, which caused some difficulties, but also gave me a lot of inspiration, and the communication and cooperation with the team members also solved many of my doubts.
It is hard to build business rules and do the recommend, and by reading materials, I understand the field more and get ideas how to implement.
I also learn to choose suitable data source and how to do the data processing work.
I spent a lot of time beautifying the interface and debugging the business logic in the front-end design, which improved my development ability.
I also learned some knowledge of CF and matrix factorization algorithms from Huang Jia Hao.

3. how you can apply the knowledge and skills in other situations or your workplaces

By completing this project development, I have gained a deeper understanding of Reasoning System and experience in how to apply this knowledge.
In the future, when I need to deal with similar recommendation system tasks or other tasks in the workplace, I will be much more experienced, and more efficiency to solve similar questions during development.
In addition, in my daily life, I am also willing to try to implement some interesting systems (interest-based programming) through recommendation system or constraint optimization method to help me deal with some special tasks and achieve satisfactory results.

Individual Reports

Your Name:	SHAO ZHERUI (A0231482L)

1. personal contribution to group project

I participated in many parts of the project with other teammates. My main contribution is to design and implement the food recommendation function with Yi Chen.

1. I designed the system structure with other.
2. I discussed and designed the business model with others.
3. I implemented part of the domain knowledge functions and Google Ortools constraint optimization in food recommendation module.
4. I assisted teammates to implemented and tested the front end system.
5. I wrote a part of the report.

2. what learnt is most useful for you

This project is a good lesson that thought me how to apply the theories learnt in the course to build a intelligent reasoning system. At first, I had a better understanding about the system design and system structure of intelligent reasoning system. Also learnt how to represent the domain knowledge and how to use it to extract information (features) from user data. I learnt how to use constraint programming tools to do search to find a best solution.

I also learnt CF and matrix factorization algorithms from Huang Jia Hao, and I learnt some knowledge about Django framework from Yi Chen.

3. how you can apply the knowledge and skills in other situations or your workplaces

This project provides me with experience of the entire procedure of developing an recommendation application on intelligent reasoning system, including the business model design, system design, data collection and processing, back end function implementation and front end implementation. This experience would help me to lead an intelligent reasoning project in my future work. Also, the knowledge about all these algorithms implemented in this project and the programming skills in reasoning system would help me to be a good algorithm engineer and a good programmer.