# Restaurant Recommendation

---

*A restaurant recommendation website based on clustering, decision trees, K-nearest neighbors and rule-based methods.*


Team member:

Chen Gu (cg736)
Jiaqi Gu (jg2385)
Ruiming Kou (rk656)

Table of Contents



# 1.Introduction

We implemented a restaurant recommendation system based on the background, preferences and needs of customers. We mainly use data mining methods, like K-means clustering, decision trees, to train our data and predict. We will then do a cross validation to assess and tune our classifier. Finally, we built a website GUI for user to register and search the restaurant. During registration, we will collect a customer's background information and apply decision tree model to this customer to categorize this user. With the categorical information, we can recommend the most rated restaurants by this certain group of users. Users can also set some conditions to filter the recommendation result. The rest of the report is organized as follow: Section 2 introduces our training data. We will discuss the recommendation algorithms in depth in section 3. In section 4, we will illustrate the workflow of our GUI. Finally, conclusions and contributions will be specified in section 5.

# 2.Training Data

## 2.1 Data source

The data that we use is from UCI machine-learning-databases (https://archive.ics.uci.edu/ml/machine-learning- databases/00232/). This dataset contains 9 tables with more than 900 restaurants recorded, along with their detailed information (more than 20 features) such as location, dress_code, smoking area, has_parking, cuisine etc. There are also user data (more than 20 features) like their budget, dressing_preference etc. More detailed data description can be found in the README file.

## 2.2 Preprocessing:

- **Clean up**

Since he original dataset is not complete and may contain unuseful information regarding clustering and classification, so the first step we need to do is to clean up the data and extract useful information. There is some missing value in the original dataset. For continuous variable, we calculate the mean to replace the missing value. For discrete data, we use the mode to replace the missing value.

- **Transformation**

We transform the data to integers such that each class and each attribute is represented as a positive integer. Such transformation is extremely useful when we want to do Principle Component Analysis. Another benefit is that given the information about a user, we can manipulate the data easily.

- **Feature Extraction**

Since there are many features about a user and a restaurant and it is hard to determine which feature is more important when it comes to clustering or classification. So instead of select representative features manually, we do Principal Component Analysis on user and restaurant data to extract the best 7 features for future clustering and classification.

# 3.Recommendation Algorithms

## 3.1 Clustering

Based on the preprocessed data, we use K-means to cluster the user into 5 categories.

## 3.2 Decision Tree

Then we can use these labels to model a decision tree classifier. Below is the decision tree.

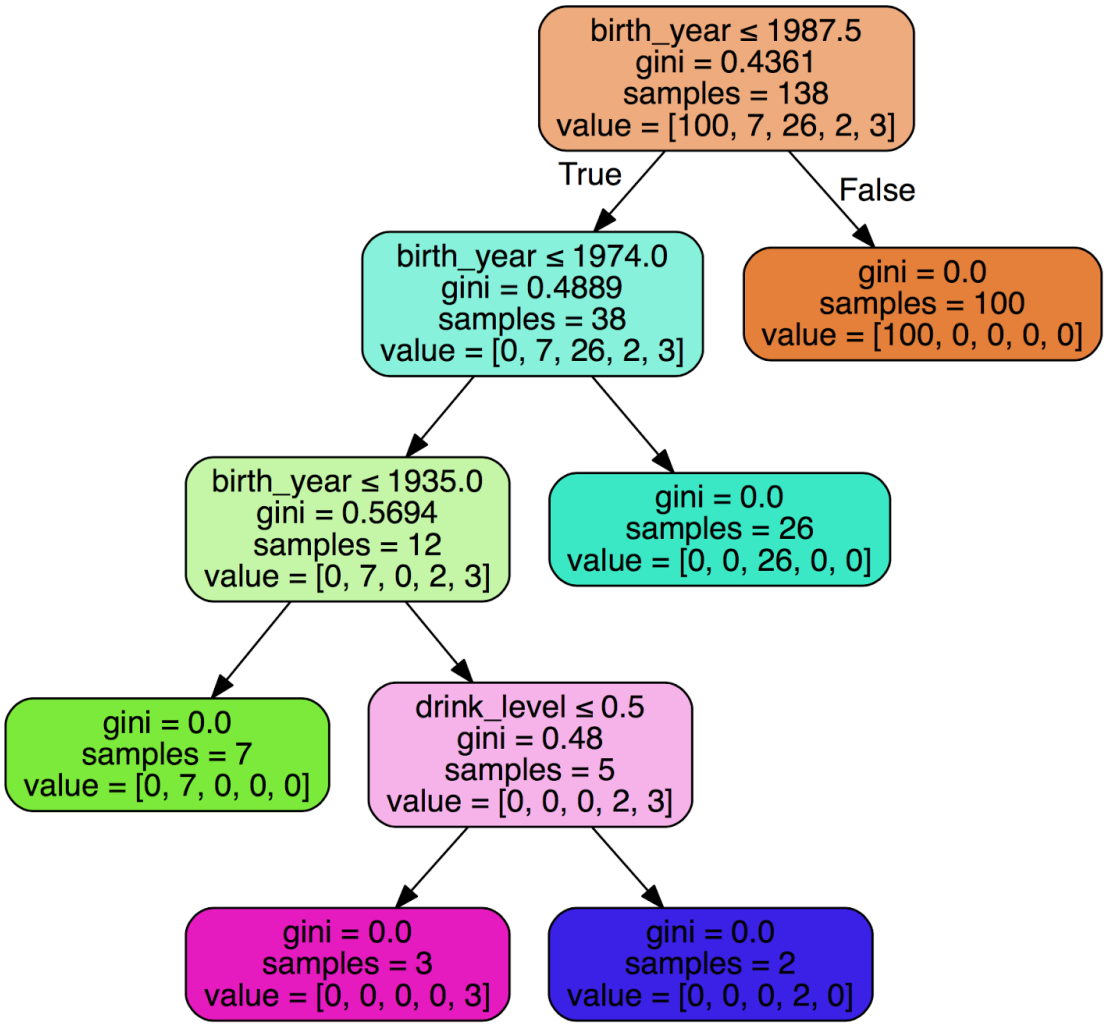


Figure 1: User Decision Tree

## 3.3 Classify

For new users, we will collect some basic information during registration. We classify new users into one of the 5 categories using the decision tree model trained in the last step. After we predict their labels, we find the most rated restaurant in this certain group of people. We sort the restaurant by their ratings and recommend the top restaurant to users.

## 3.4 Filtering by rule-based method

We may return multiple recommendations. Users can customize the result by adding filter conditions to achieve fine-grained recommendations. For example, if the user decides to drive a car to his destination, we prefer to recommend some restaurants that have parking lots. With the help of the combination of data mining and rule based method, the system could provide restaurants that meet the exact requirement from our users.

## 3.5 Further Recommendation

Based on user's selection, we find 3-nearest neighbor of the user-selected restaurant for user to compare (work like related item in Amazon).

# 4.GUI Workflow

A new user need to sign up first. After filling in the basic information and questionnaire, the system will create an account for the user and automatically log in the restaurant store. The information collected from the questionnaire will be used for future clustering and classification.

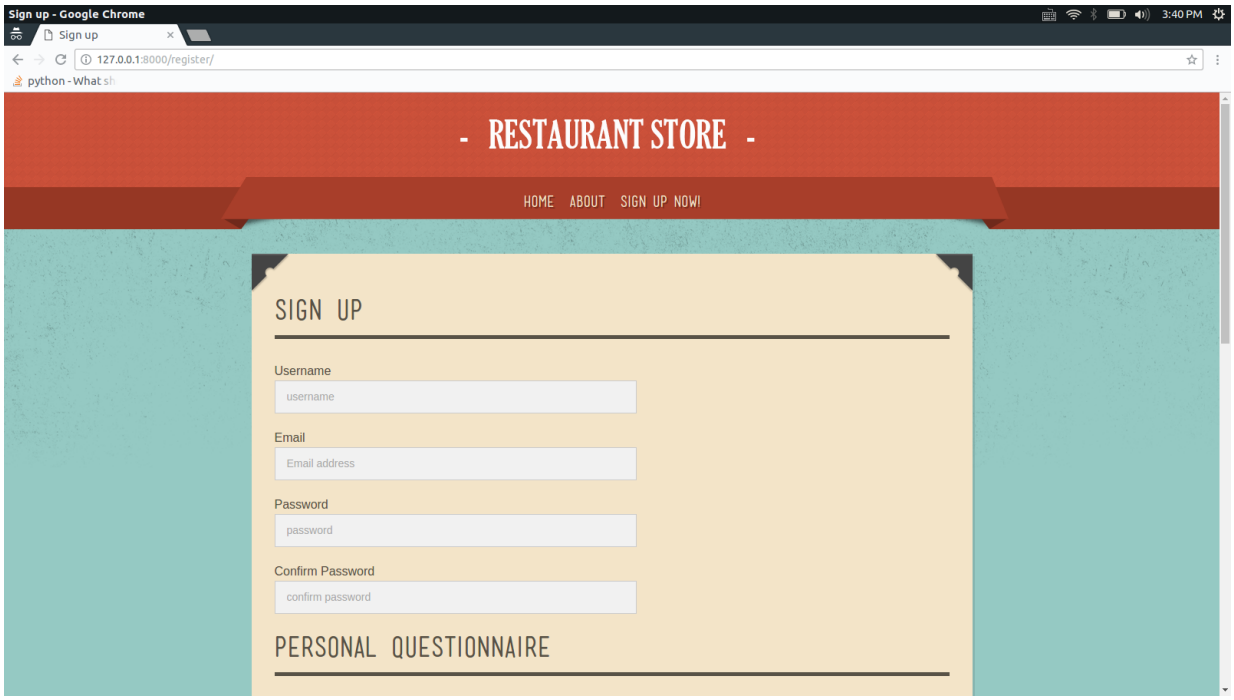


Figure 2a: Sign up page

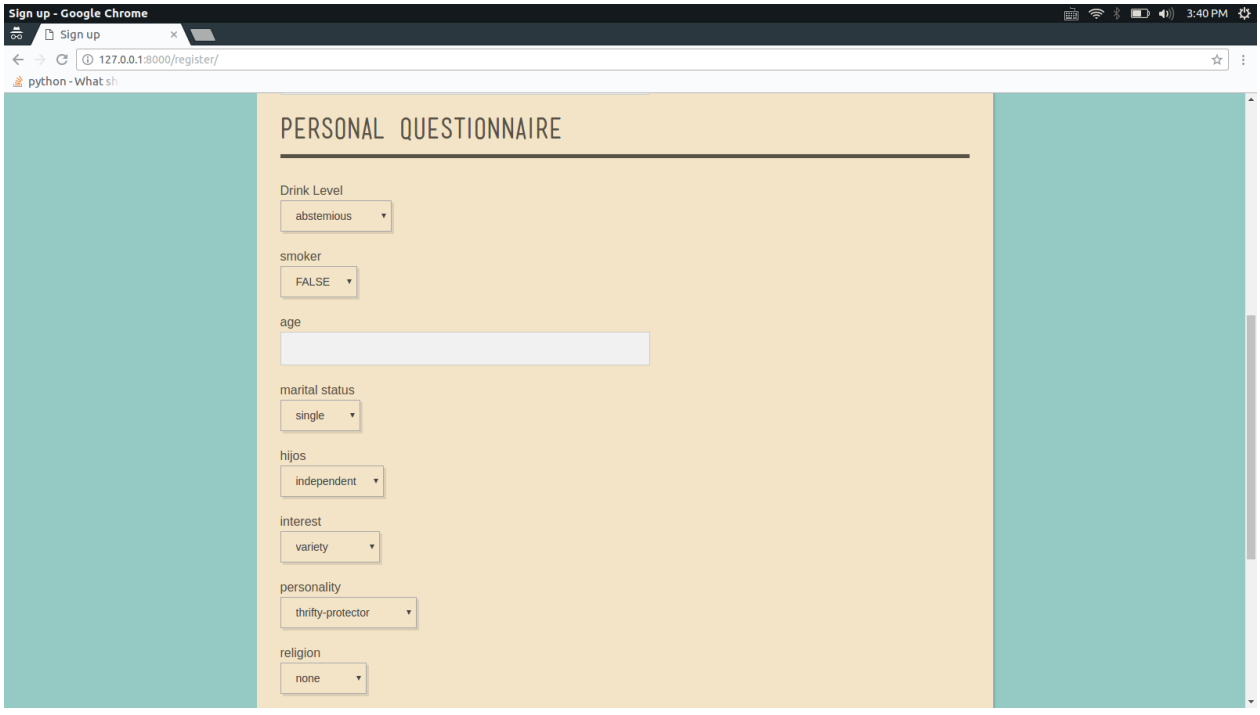


Figure 2b: Sign up page

After the user account successfully created, the user can log in Restaurant Store using his username and password by hitting the ‘Log in now’ link on the home page.

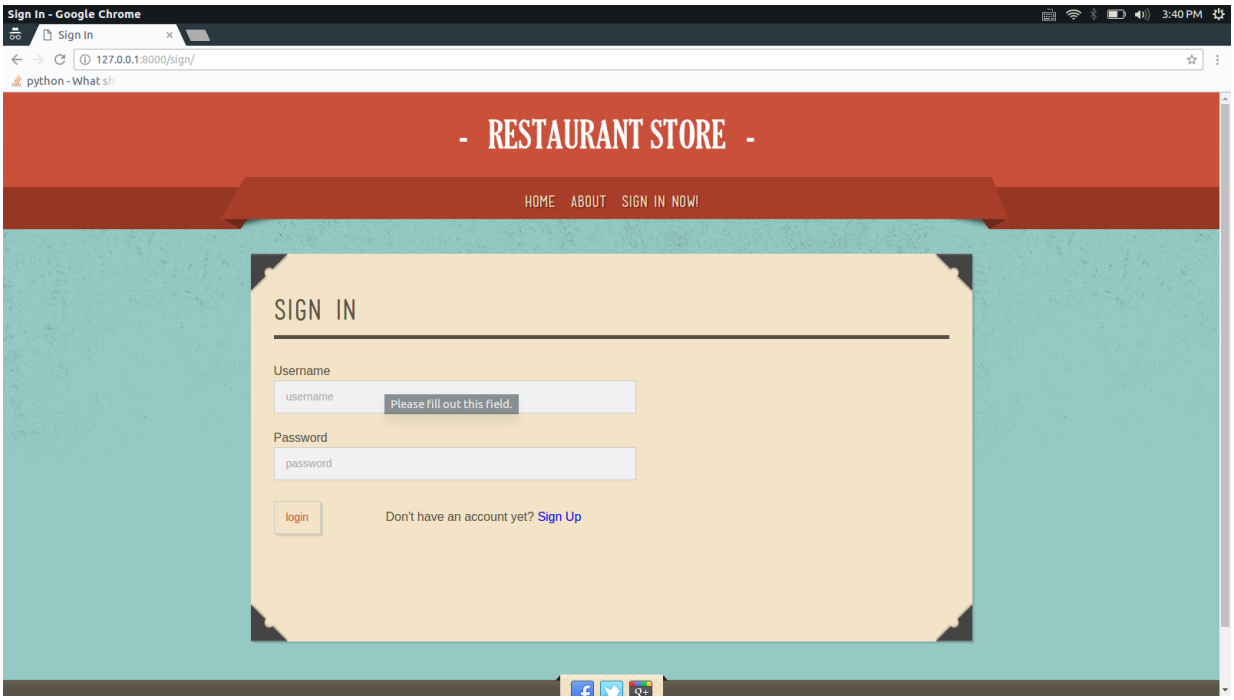


Figure 3: Sign in page

When a user successfully log in, a window will pop up, in this window, the user should select his particular requirement of his planning meal. Restaurant Store will filter the result from classification and clustering process based on user’s selection in this window.

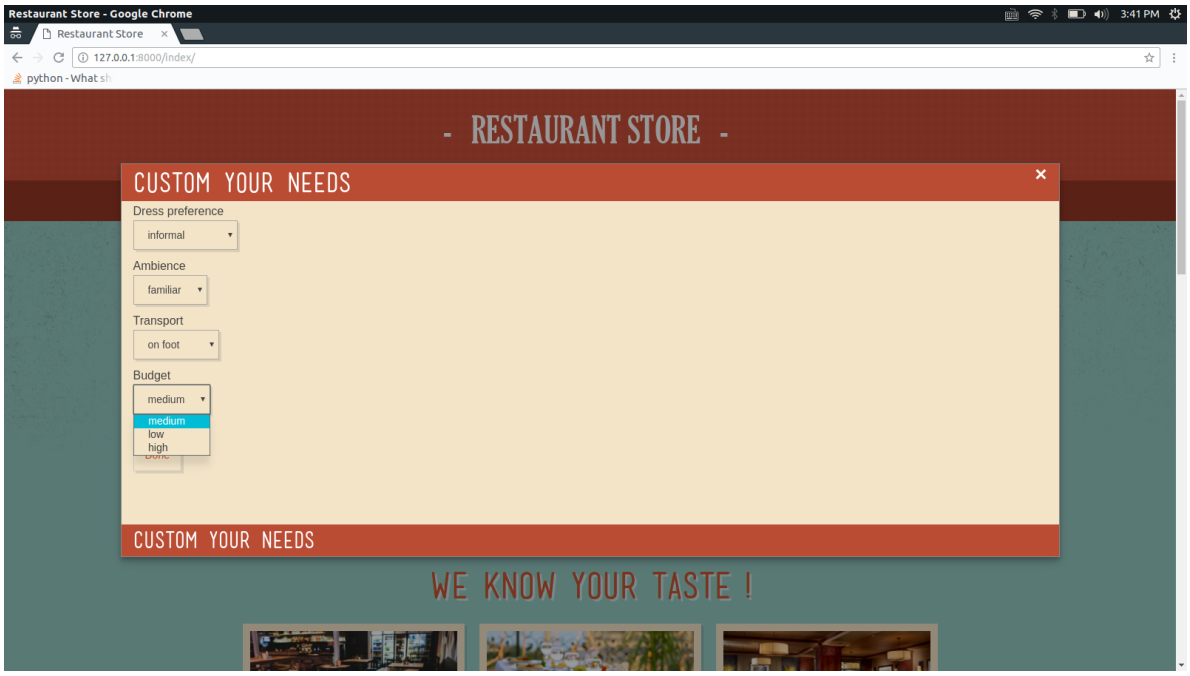


Figure 4: Fine-grained selection

After doing that, a list of restaurants will be displayed in front of the user which is based on his profile and his selection in the popped window. In this page, we can see the picture and price of those recommended restaurants. Every time a user want to change his requirement, he can hit the button ‘recommend’ in the header to show the ‘custom your needs’ window.

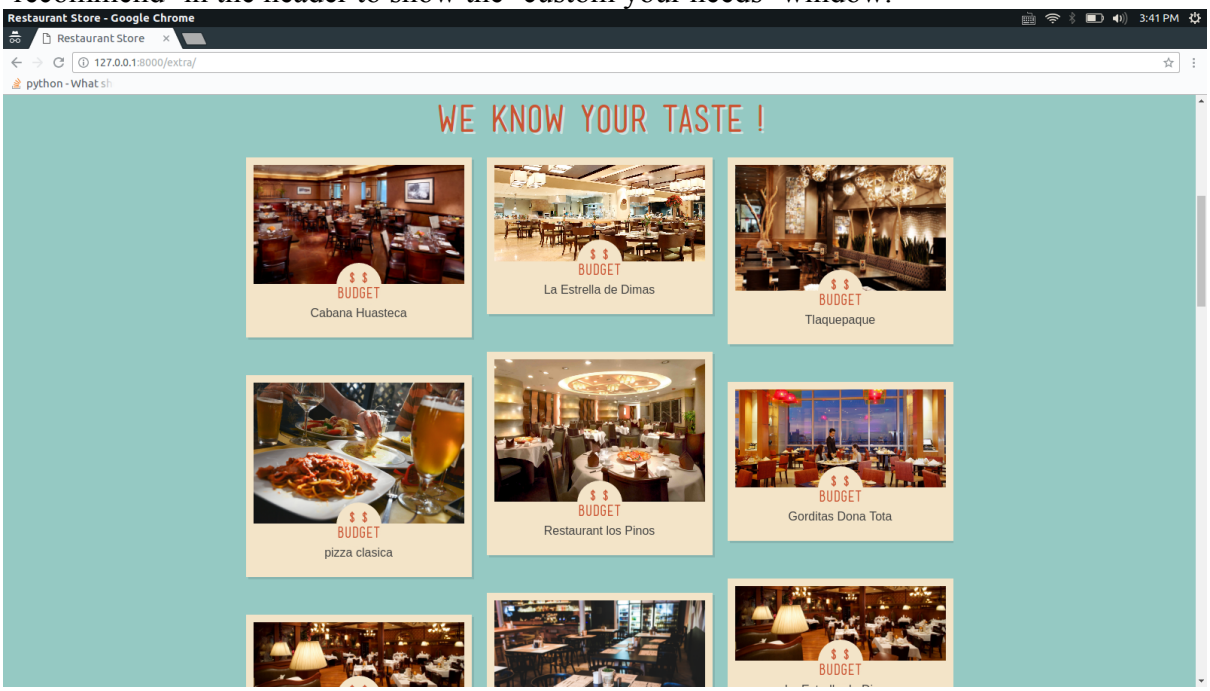


Figure 5: Recommendation page

If the user prefer to get the detail information of a particular restaurant, he could hit the picture (or name) of this restaurant and the detailed restaurant description page will be shown.

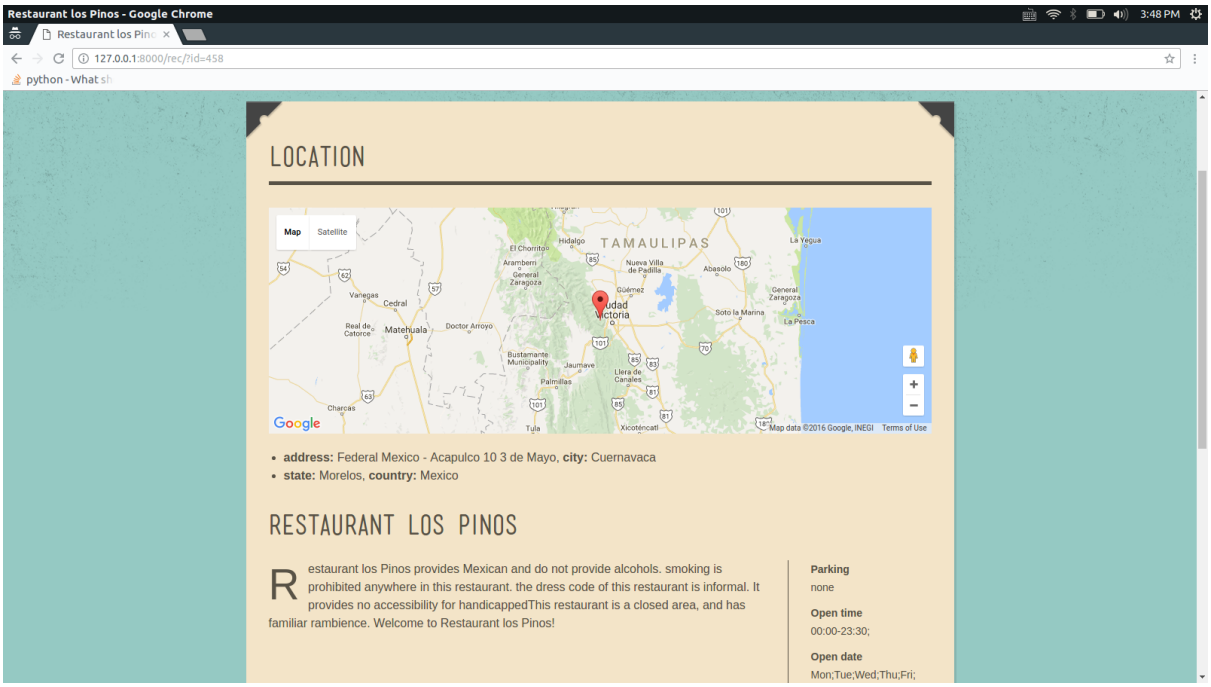


Figure 6: Restaurant map page

In this page, the user could get the basic information about this restaurant such as its geo-location, address, whether it will provide smoking area, what is the drinking level of this restaurant and so on. At the bottom of this page. We could get the related restaurants which have similar features of the above one. We get the related restaurant using kNN method in data mining

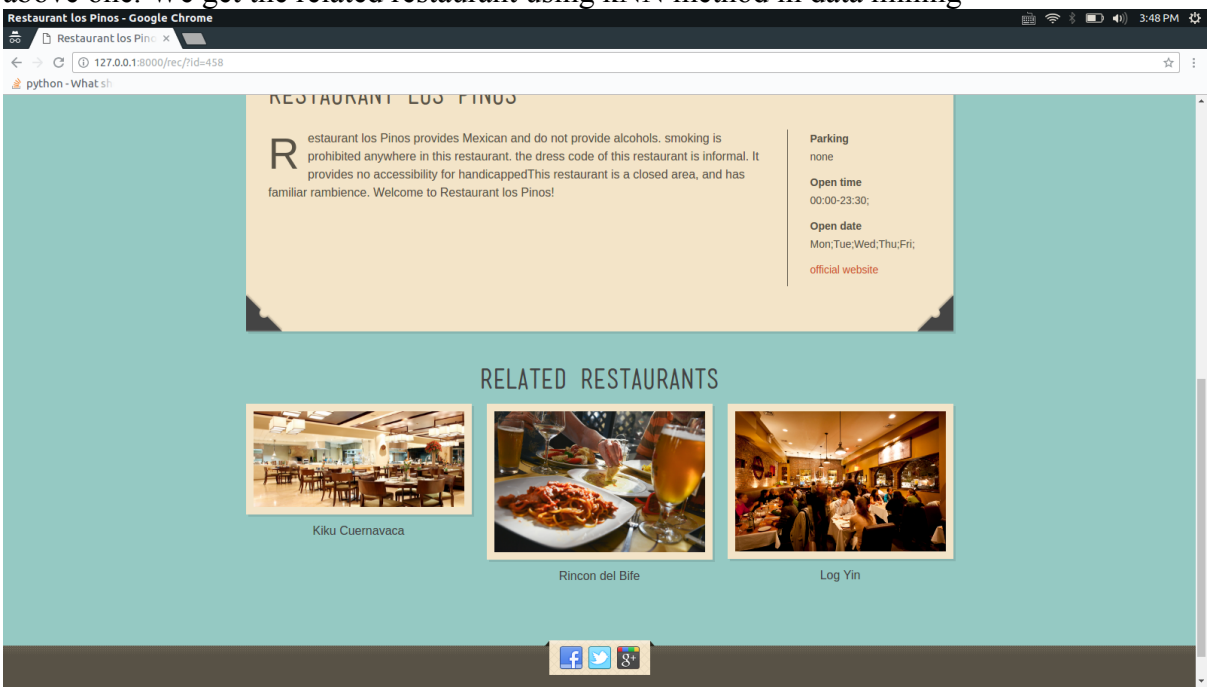


Figure 7: Related restaurant recommendation

A user can check his basic information and his answer of the questionnaire in the ‘info’ page.

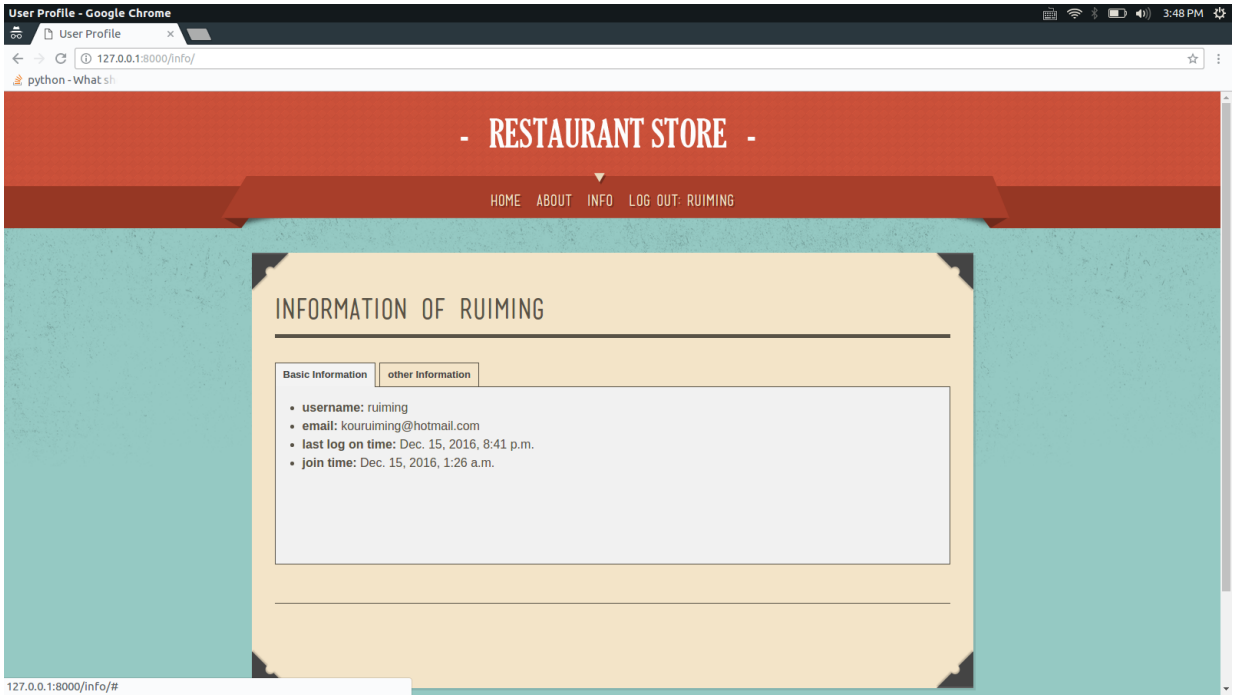


Figure 8a: User information

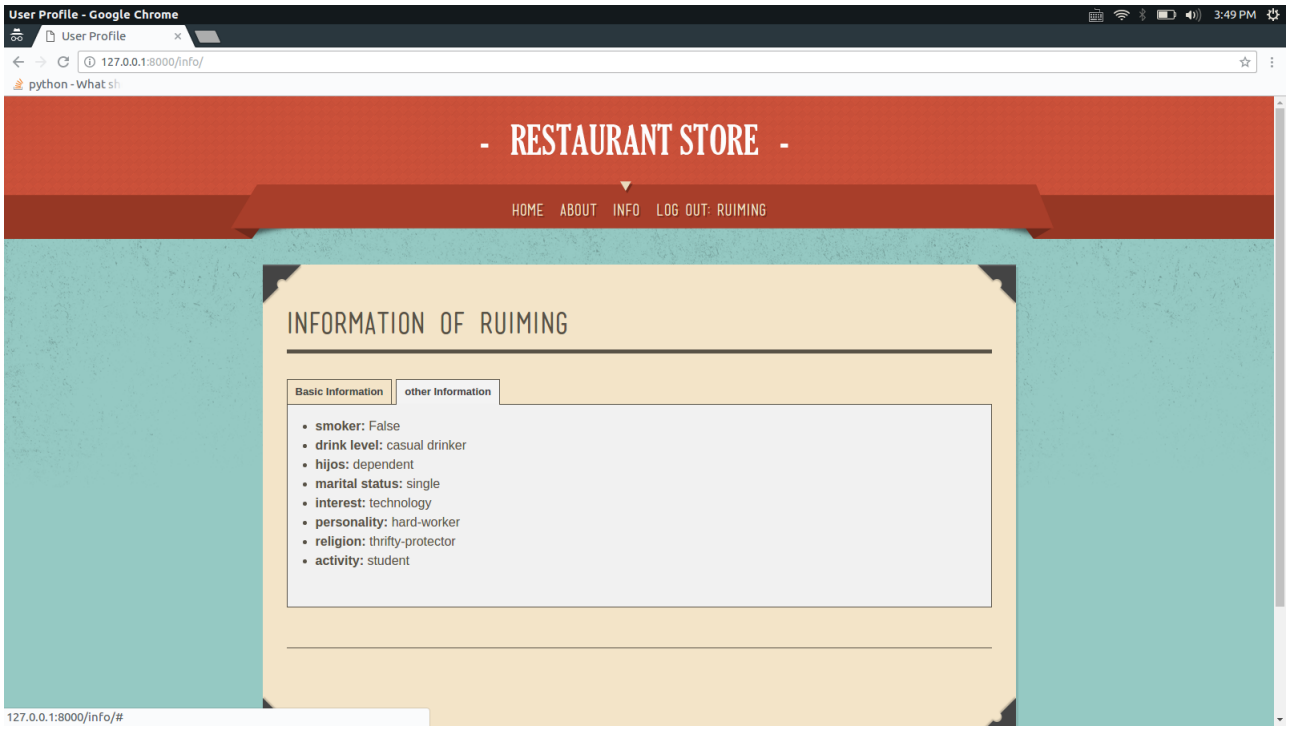


Figure 8a: User information

We post the brief description of our overall project in the 'about' page.

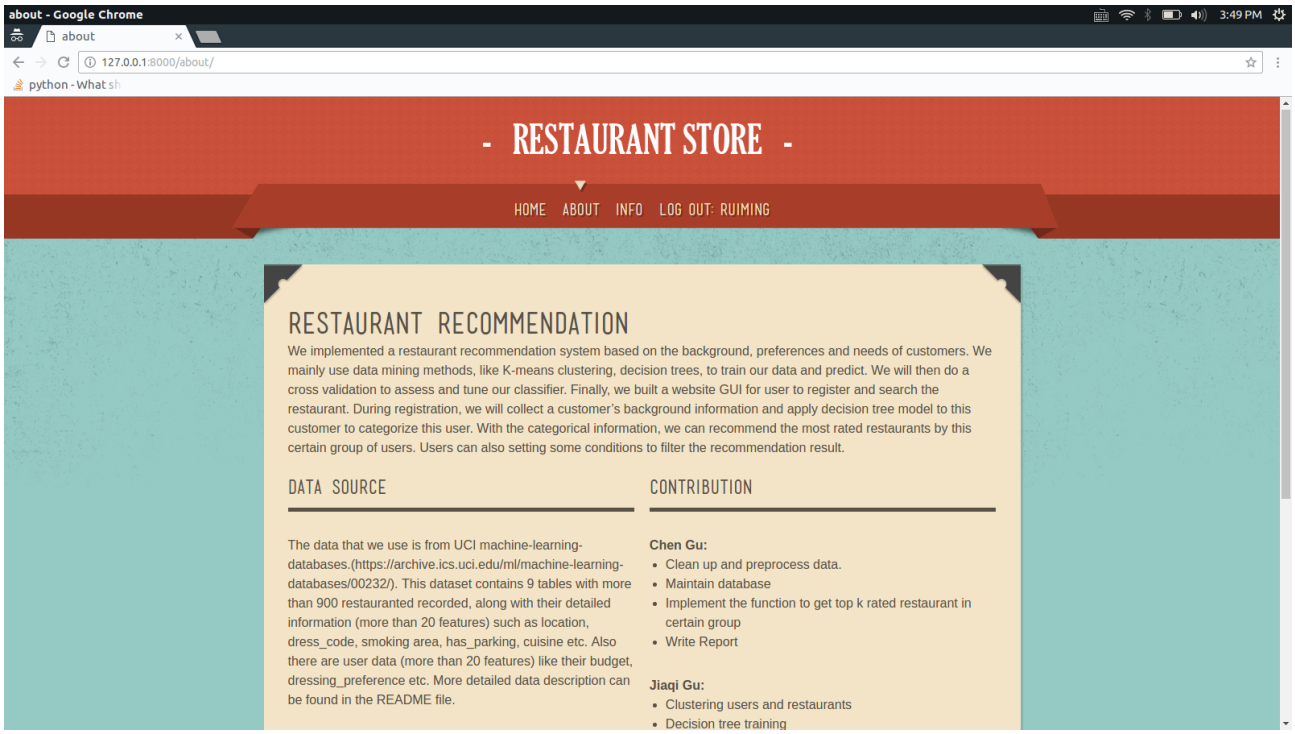


Figure 9: Project information

When a user made up his mind, he could hit the 'log out {username}' link to log out.

# 5.Usage

## Prerequisite

1. pandas
2. scipy
3. sklearn
4. pydotplus
5. django
6. sqlite3

**Install:** All above library can be installed by pip. (e.g. pip3 install pandas)

## Run the server locally

In the root directory, run following command:
python3 mange.py runserver [ip:port] (e.g. python3 mange.py runserver 0.0.0.0:8000)

**Note:** ip:port is optional, the default value of which is 127.0.0.1:8000

## Deployment

We deployed our website on Google Cloud, so it can be accessed by the following URL:
http://104.197.202.136:8000/index/

**Default User:**
**Username:** ruiming
**Password:** 1234567

# 6.Conclusion and Contribution

Chen Gu:

1. Clean up and preprocess data
2. Maintain database
3. Implement the function to get top k rated restaurant in certain group
4. Write Report

Jiaqi Gu:

1. Clustering
2. Decision tree training
3. Classification

Ruiming Kou:

1. Implement the overall system using python
2. Design the rule based method in this system.
3. Implement the kNN method to get related restaurants