Meal planning for the new Millennium

1 Introduction
Deciding the recipes for each day is an important problem in daily life and experts have studied diet problem since 1940s. And when it comes to this problem, people often consider several elements, such as the nutrients requirements, cooking time each recipe costs and expense of recipes. And our project will help to solve meal plan problem.

2 Diet Problem Statement
2.1 Frame of diet problem
In order to find most suitable 5 recipes from a recipe pool, our group built an optimization problem with coefficient of personal preference ratings, based on constraints of nutrients, time and other requirements, while using recipe information, like ingredients, from the recipe website.
The objective of this optimization program is to maximize the taste preference when satisfy all 3 constraints: nutrition, cost and time for two roommates' 5 meals in 5 days. The first step is to scrape data from recipe website, the next step is to acquire preference ratings based on SVD method, then scrape price data of ingredients to calculate recipes allowance. Eventually, after placing upper and lower bound, all constraints are formulated to solve optimize program.

2.2 Method
2.2.1 Singular Value Decomposition (SVD)
SVD is a matrix factorization technique and is usually help to decrease dimension of matrix from K to N, where N < K. But for meal problem, SVD are used to find ratings for different receipes, only consider the factorization keeping same dimension. Similar in recommadation system, the original matrix as ustility matrix shows customers opinion for items. Using SVD method, utility matrix could be decomposed into two singluer matrixs U and V, and a diagonal matrix S, as stated in below equation[1]:

$$X = USV^T$$

X denotes utility matrix, U indicates relationship between users and lantent factors, V represents relationship between items and lantent factors, S represent the strength of latent factors.
In meal problem, the utility matrix is sparse and most ratings aren't known, we use SVD to decrease MSE, to decompose original matrix two singular matrixs then to help predict personal ratings.

3 Methodology
3.1 Model Construction
3.1.1 Mixed Integer Programming
Considering the objectives under constraints condition, here we we construct the MIP[2]:

$$\text{Max} \quad \sum_{k=1}^{K} r_k \cdot x_k - \alpha \cdot W$$

$$\text{s.t.} \quad \sum_{k=1}^{K} x_k = 5 \,, \qquad \text{5 meals per week}$$

$$\text{s.t} \quad \sum_{k=1}^{K} \sum_{i=1}^{5} (z_{k,i} + y_{k,i}) - x_k = 0$$

$$\text{s.t} \quad \sum_{i=1}^{5} (z_{k,i} + y_{k,i}) - x_k = 0 \,, k \in 1, \ldots K$$

$$\text{s.t} \quad \sum_{k=1}^{K} x_k = 5 \,, k \in 1, \ldots K$$

$$\text{s.t} \quad \sum_{k=1}^{K} (z_{k,i} + y_{k,i}) \leq 1 \,, i \in 1, \ldots 5 \qquad \text{one day one recipe}$$

$$\text{s.t} \quad \sum_{i=1}^{5} (z_{k,i} + y_{k,i}) \leq 1 \,, k \in 1, \ldots K \qquad \text{no repeat meal}$$

$$\text{s.t} \quad \sum_{i=1}^{5} \sum_{k=1}^{K} t_k \cdot (y_{k,i} - z_{k,i}) \leq W \,, \qquad \text{time difference}$$

$$\text{s.t} \quad \sum_{i=1}^{5} \sum_{k=1}^{K} t_k \cdot (-y_{k,i} + z_{k,i}) \leq W$$

$$\text{s.t} \quad \sum_{k=1}^{K} t_k \cdot y_{k,i} \leq t_i \,, i \in 1, \ldots 5 \qquad \text{cooking time constraint}$$

$$\text{s.t} \quad \sum_{k=1}^{K} t_k \cdot z_{k,i} \leq t_i \,, i \in 1, \ldots 5$$

$$\text{s.t} \quad \sum_{k=1}^{K} p2_k \cdot x_k \leq B_k \,, \qquad \text{expense constraint}$$

$$\text{s.t} \quad \sum_{k=1}^{K} A_k \cdot x_k \leq N_u \,, \qquad \text{nutrients constraints}$$

$$\text{s.t} \quad \sum_{k=1}^{K} A_k \cdot x_k \geq N_l$$

where: k represents ordinal number of recipes, i represent different day. The meaning of variables and parameters are shown in below two parts.

3.1.2 Define the variables

First of all, x_k is a binary variable, represents the status of chosen or not chosen, and the index k shows the number of recipes. And r_k indicates the rating of each recipe. Then, $y_{k,i}$ and $z_{k,i}$ are also binary variables, the index k shows the number of recipes and i represents days. And y is corresponded to Shuting and z is corresponded to Qian, represent Shuting or Qian will cook k recipe in i^{th} day. W indicates the penalty, related to imbalanced cooking time. And α is the weight of penalty. The indexes used here have same meaning as indexes used to defining parameters[2].

3.1.3 Define the parameters

Considering the constraints on time and budget, we define T_y, T_z and B as the corresponding parameters. The value of each recipe's cooking time and expense is shown as t_k and p_{2k}, and k indicates recipes.

Due to various type of nutrients and their minimum and maximum requirements, we define a lower bound and a upper bound.

$N_l = (c1_l, c2_l, f_l, p_l)^T$

$N_u = (c1_u, c2_u, c3_u, f_u, p_u, s1_u, s2_u)^T$

$A_k = (c1_k, c2_k, c3_k, f_k, p_k, s1_k, s2_k)^T$

c1, c2, c3, f, p, s1, s2 represent calorie, carbohydrate, cholesterol, fat, protein, saturated Fat and sodium, respectively. A_k represents all nutrients values for all k recipes[2].

For particular value of parameters, we use the Daily Nutritional Goals for Age-Sex Groups' recommendation[3] as our criterion of setting the suitable value for upper bound and lower bound. So based on Dietary Reference Intakes and Dietary, we finally choose Female 19-30 group as our objective. And we adjust the requirement of calories based on our BMI. For other nutrients, according to relationships between other nutrients and calories, like the percentage of each nutrient take up with calorie, value of requirement could be acquired.

3.2 Data processing

3.2.1 Scrape data

In this project, recipe data are scraped from epicures website[4] and below shows the detail information about data we get.

Data information:

1. Recipe Title

   At this stage we scrape the recipes based on our preference by looking at the recipes that are from gallery we like and exclude some recipes with the title of 'slow-cooker', 'spice', 'Lamb' and etc.
2. Rating
3. Nutrition Information

   we scrape nutrients information provided, for instance, calories, carbohydrates and fat.
4. Time

   At this stage, we choose using active time rather than total time. Because we think prepare process could be accomplished the day before. The prepare process shouldn't take up the cooking time. And here we unify the units as minutes.
5. Ingredient

   We also scrape ingredients in order to calculate the cost of each recipes for our model. We exclude the ingredients which unit of measure is tablespoons(tablespoon, teaspoons, teaspoon, tsp., Tsp, Tbsp.), because the amount of those ingredients are small and eliminating those ingredients will has little influence on price calculation.

Therefore, we get the main ingredient of each recipe. Then we create a table describe the amount of ingredients need for each recipe and also get a column of ingredient which are used to match prices.

6. servings
   Since the servings of each recipes are different, we need to be tailed the recipes suit for two people. This will be used to calculate our cost of recipe later.
7. price of ingredients
   At this point, we scrape price data from Amazon fresh, and use ingredients' prices to calculate expense of recipes later.

Below is the matrix of recipes and relevant information we get.

Figure 1 The data matrix for MnM Problem

```
df = pd.read_json('recipes_data.json')
df
```

	activetime	calories	carbohydrates	cholesterol	fat	fiber	monounsaturatedFat	personal_rating	polyunsaturatedFat	protein	rating	saturateFat	servings	sodium	time	title	total_ingredients
0	30.0	1083.0	73.0	313.0	64.0	3.0	24.0	[[dylan_moench , 2], [janmurray from Orlando, ...	5.0	52.0	4.0	32.0	4.0	1145.0	[30, minutes]	None	[8 oz. coarsely grated Fontina cheese (about 2...
1	35.0	NaN	NaN	NaN	NaN	NaN	NaN	[[Fatketocat , 5], [car3yc0x from Stockton, CA...	NaN	NaN	4.5	NaN	NaN	NaN	[35, minutes]	Cauliflower "Mac 'n' Cheese" Casserole	[8 cups coarsely chopped cauliflower (about 2 ...
2	10.0	365.0	38.0	36.0	19.0	3.0	5.0	[[janet_callej from MD , 5], [ddeuddeg from Bu...	1.0	11.0	5.0	11.0	2.0	275.0	[10, minutes]	Cinnamon White Hot Chocolate	[4 (3") cinnamon sticks, 2 1/2 cups whole milk...
3	30.0	553.0	55.0	130.0	27.0	7.0	12.0	[[oliveoil62 from Bellingham, WA , 4], [busser...	4.0	23.0	4.5	9.0	4.0	689.0	[30, minutes]	Pasta Carbonara with Cabbage and Mushrooms	[8 ounces shiitake mushrooms, stems removed, s...

3.2.2 Clean data

Because the optimization problem has constraints on time, nutrients and cost, the lack of related information, such as nutrients information or time, for some recipes will leads to difficulty on formulating constraints. Therefore, we decide to delete rows contain NA value. Meanwhile, we only consider the recipes with at least ten reviews so that the accuracy of personal ratings will be increased.

3.2.3 Coefficient calculation

(1) Calculation of personal ratings

Here we use SVD method to predict personal ratings based on previous user's ratings. Because the matrix we get now is a sparse matrix, in order to have less NA's we combine the people who are from the same state as one typical user. And due to this operation, more recipes are rated by typical users.

The first step is to define standard to combine several personal as a typical user. We assume that people come from same state may have similar preference, so we set the region of user as our criteria. The next step is to manipulate the ratings to get the final matrix, which will be used to get personal ratings in SVD process. To achieve this goal, we use the information of location, rating and recipe title. The state abbreviation of each user is recorded and if customers' location doesn't match 54 states, label those users with "other". Then those data are grouped by state abbreviation and the mean ratings are calculated as the typical user's ratings for each recipe. Eventually, we get a table whose columns contain information of recipe title and state

abbreviation and values of ratings. A matrix is gotten by pivoting the table. After this, we insert a row to rate some of the recipes we don't like and get the final matrix.
By using the final matrix as input file, we estimate our personal ratings for all recipes through SVD and this list of value is used as the coefficient of objective function later.
(2) Calculation of recipes' expense
In order to get the constraint parameter for our model, we need to calculate the cost of each recipe. Things can sometimes sound very simple but can be fiendishly difficult when the details are considered. In particular, in this case, it is difficult to calculate exactly price for each recipe. For example, consider the following:

1. The units of measurements for each ingredient in recipes is different.
2. The cost will be varied in terms of the quality of ingredient and where people buy.

After considering various situations will be met in the process, we decide to exclude the ingredients like salt and any other seasoning since the units of measures are vague and amount are small. And the amount of ingredients used, and its corresponding recipe title are stored in a matrix. Combined with matrix contains information of price of recipes, we can calculate total price of recipes. Because each recipe may have different servings, so we divide total price by servings then multiple 2 as two servings for two roommates.

4 Result and discussion

After getting exact latest values from Internet and building the Mixed Integer Program, we applied Cplex solver to find the solution of diet problem. The final result is recipe 2, 13, 16, 19 and 33 are chosen, and Shuting is suggested to cook recipe 2 and 19 on Friday and Monday, respectively. Meanwhile, Qian is suggested to cook recipe 13, 16 and 33 on Thursday, Wednesday and Tuesday.

According to solution, we find the difference between total estimate cooking time for Shuting and that for Qian is small, which reveals great equality and proves this result make sense. But to contrast with 25, the value of objective function is quite small. For this problem, one reason is the low socre of of personal ratings. Since only few people will grade most of recipes we chosen and even we set region as our constraints to combine several customers into one typical customer, the matrix used to do SVD is still sparse, which we think may place influences on final ratings. Before estimating our own ratings, we add some ratings in that matrix and grade very low for recipes we don't like, which have high grades among other person. Those factors lead to the low grades of personal ratings so that the result of objective function would be small. In conclude, five meals chosen all can satisfy our requirement of nutritions, time and budget.

Figure 2 The Final optimization result for MnM problem

```
objective: 10.220
  x_2=1
  x_13=1
  x_16=1
  x_19=1
  x_33=1
  y_2_4=1
  y_19_0=1
  z_13_3=1
  z_16_2=1
  z_33_1=1
  w=3.000
```

Reference

[1] Huang, Steeve. “Introduction to Recommender System. Part 1 (Collaborative Filtering, Singular Value Decomposition).” *Hacker Noon*, Hacker Noon, 28 Jan. 2018, hackernoon.com/introduction-to-recommender-system-part-1-collaborative-filtering-singular-value-decomposition-44c9659c5e75.

[2] Xu, Jia Jun. “MnM_instance.” *Microsoft Azure Notebooks - Online Jupyter Notebooks*, 2019, notebooks.azure.com/jiajunxu/projects/MnMinstance/html/MnM_MIP.ipynb.

[3] Institute of Medicine. Dietary Reference Intakes for Calcium and Vitamin D. Washington (DC): The National Academies Press; 2010. https://health.gov/dietaryguidelines/2015/guidelines/appendix-7/

[4] Xu, Jia Jun. “The Notebook Used to Generate the Data for Meal Planning for the New Millennium (MnM) Problem.” *Google*, Google, 2019, colab.research.google.com/drive/1UIoAKLnOkpBGtBrDGy3XxmbyZz7sJ_Qa#scrollTo=9VDhj16Ob4OZ.