



31005

Advanced Data Analytics

Spring 2016

Assignment 2
Practical Data Mining Project

Table of Contents

1. Introduction	- 1 -
2. Data Exploration.....	- 2 -
3. Modelling	- 2 -
4. Initial Findings	- 4 -
4.1. Duration of cooking.....	- 4 -
4.2. Correlation between energy, sugar, protein and fat..	- 7 -
4.3. The nutritional values correlation between lamb, pork, fish, beef and chicken.....	- 7 -
4.4. The highest nutritional values of all ingredients	- 10 -
5. Further Findings of GA Results	- 10 -
6. Difficulties Encountered	- 15 -
7. Evaluation.....	- 16 -
7.1 Decision Tree	- 16 -
7.2 Random Forest	- 20 -
7.3 Recommendation	- 22 -

1. Introduction

This report provides an analysis and evaluation of the current problem of the public community having difficulty in finding nutritional information in the foods that they need for their specific diets which is provided by UNSW nutritionists. They believe that the public community are having difficulty in finding nutritional information in the foods that they need and certain individuals need a specific requirement for their diet due to specific diets, cultural backgrounds and improvement of health foods.

Another factored problem is the amount of time that the individuals would spend trying to find the correct diet requirements and not all individuals have the luxury of time to be searching for foods that would benefit them, which may conclude them to end their search and not improve their diets.

The difficulty in this occurs when the individuals are searching for their needs on the internet and online web databases, due to the large amount of information that is provided on the internet, it is difficult to determine whether this information is accurate and can be relied on.

Using different methods of data analytics, we discover that due to the large amount of data on the internet, it is deemed difficult to determine what information may be correct and reliable. The main objective is to identify the best rated meal with a certain nutritional requirement which will meet the needs of the individuals. The main method of analysis we will be using to determine this is generated through an algorithm called " Genetic algorithm".

The ingredient database was provided through research from the USDA nutrient database, while the recipes were obtained through various different types of healthy online food recipe sites.

Our main target audience include healthy individuals, patients suffering from different diseases and athletes. The main objective that UNSW nutritionists aim to focus on is to identify the best rated meal with a certain nutritional requirement which will meet the needs of the individuals.

The process of identifying the best rated recipe is done by inputting the required parameters of diet requirements, a set of number is inputted for each parameter and through the program, it will generate a set of ingredients provided for a meal and will then be processed through several generations to find the best matching result. From the result, it will craft a meal from the database and identify that as the best rated meal. This process is shown

through our "Findings" section with more detail and display output of the program.

2. Data Exploration

Our database is developed through various highly rated recipe websites that provide nutritional and healthy food guides to improve lifestyles. The main website that we have used to gather information on our food ingredients however, is called USDA National Nutrient Database, after thorough search for online databases, this website database seem to be the most suitable and reliable database to use for our project. We gather our recipe database by just mainly inputting the ingredients and method from the websites we have researched suitable (nutritional facts optional). USDA provides a large amount of different parameters for nutritional value, but the only values we recorded will only be Energy, Protein, Fat total, Fat saturated, Carbohydrates total, Carbohydrates sugar and Sodium for this project as a guide from the client of what they needed.

Because of the large amount of ingredients in total that were needed within this database, most of these ingredients relied on different quantity or portion for their recipe value. Our ingredient database was generalized in a way that its nutritional information was only set for one specific value, For example, Garlic could vary in portion where recipes may not need the whole garlic and just a clove of the garlic, our ingredient database would have generalized its food nutritional value to one single clove of garlic instead of the whole garlic's value. As we finally gather all our information for the database, we ended up with 4 sets of information to work with being Recipe, Ingredient's, Recipes ingredient and the Method.

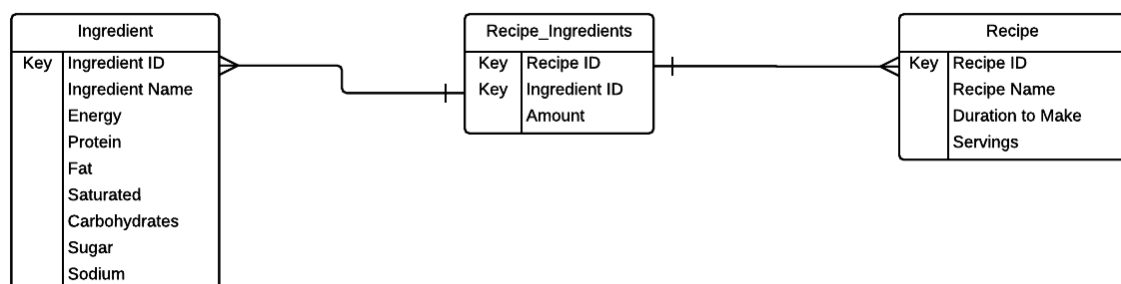


Figure 1. Entity Relationship Diagram - Database

3. Modelling

The problem in this report is solved using the concept of Genetic Algorithm.

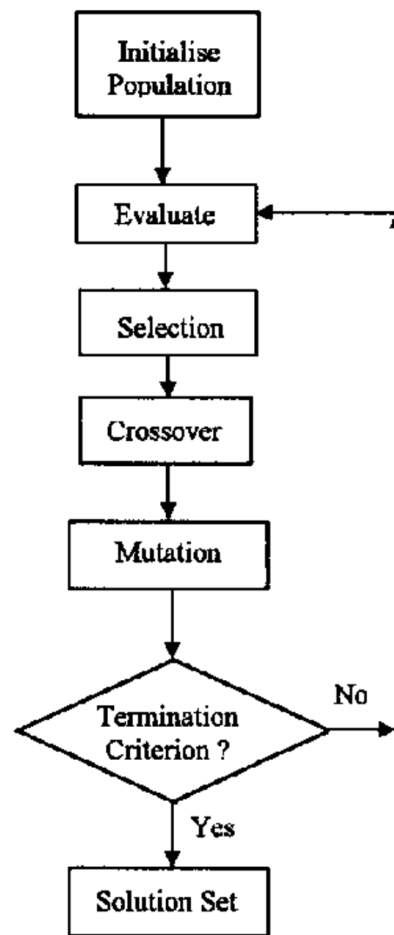


Figure 2. Genetic algorithm process.

Genetic algorithm is an optimization algorithm which is based on the idea of evolution of natural selection and genetics. This is achieved by using the repetition of its properties of selection, crossover, and mutation process to develop a solution towards a given requirement. The idea of genetic works as it will produce groups of random solutions, and each of these solutions will be evaluated based on the relevance to the problem and how they match those values. Which defines the selection operator for the algorithm being that it will select the best solution paired against another solution. If one were better than the other, that one would be selected instead and the other one will be removed. Crossover operator works as a offspring process, where two solutions will be selected and crossover of the two solutions will occur. Then an offspring solution will be processed as a result of combining both of the parents to make a better solution. Finally, the Mutation operator works as a random generator where it will select that solution and randomly make small changes or parts of the solution, this will be evaluated in the same way as the rest to see the reliability and how well it matches. These processes are then to be repeated until the best solution.

In this report, we applied the selection and mutation feature of the Genetic Algorithm. To do this, we used Python programming language to store, process, and analyse our data. The user will first enter the amount of each nutrient values, the number of generations and the number of ingredients each solution will produce.

The program will then randomly select the number of ingredients and evaluate how the solution match the target, which is the data that the user input. The evaluation includes summing up each nutrient values from the selected ingredients, measuring the error margin between the summed up value and the target value of each nutrient value and summing all the error margins.

The program will then begin the mutation process by randomly select one ingredients and change it to a random ingredient from our pool of ingredients. The current solution will then be evaluated again, calculating the total error margin. This error margin will then be compared to the previous error margin. If the current solution has less error margin, then the change will be kept as is. If the current solution has more error margin, then the ingredient change will be reverted back.

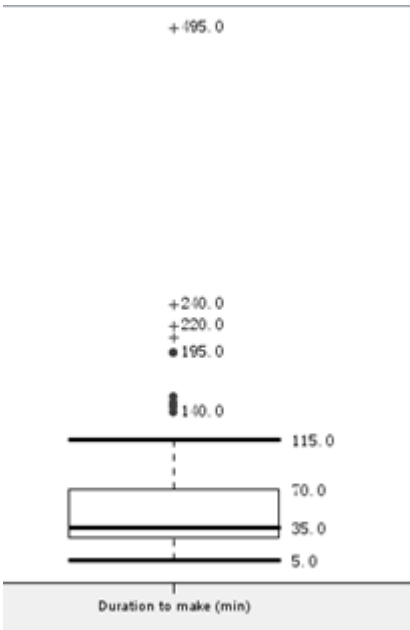
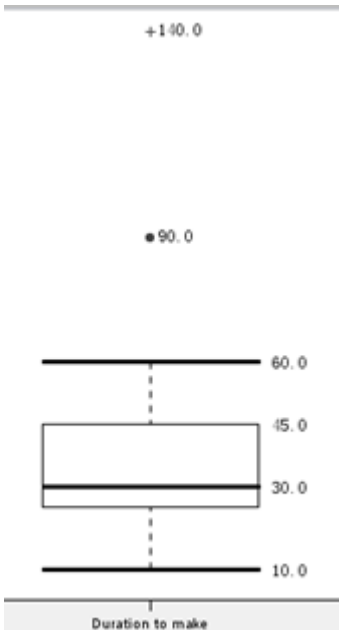
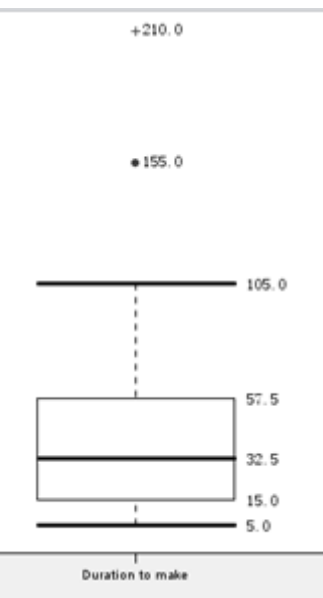
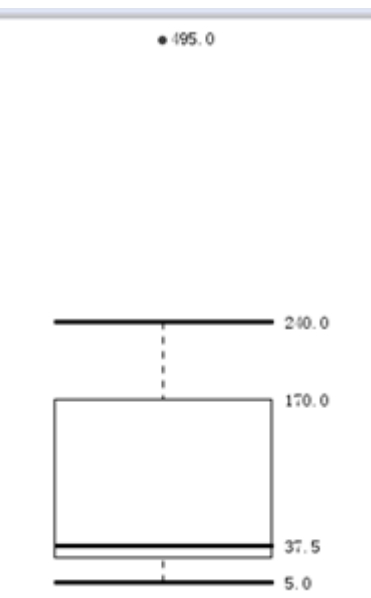
This process is repeated multiple times depending on the number of generations that the user inputted initially. In the end of the iterations, the resulting ingredients will have nutrient values which are a few amounts away from the target values.

4. Initial Findings

In this part, we are going to analysis different areas of data before doing genetic algorithm, so that we can fully understand our database and find any useful information for further processing. Initial finding includes duration for making each kind of food, correlation between sugar, protein, fat and energy values. It also involves the comparison of different nutrition values between different food groups such as chicken, beef, pork or fish.

4.1. Duration of cooking

According to the ingredients in each recipe, our group simply divide these recipes into five parts depending on the meat inside the recipes, which are chicken, lamb, pork, beef and fish. Then we analysis the duration for making these recipes that contain any of those five meats by using box plot in KNIME. After that, we got the minimum, first quartile, median, third quartile, maximum and outliers. (See figure 3)

<p>total recipes: min: 5mins Q1: 25mins median: 35mins Q3: 70mins max: 115 mins outlier: 140-240mins</p>	<p>chicken recipes: min: 10mins Q1: 25mins median: 30mins Q3: 45mins max: 60mins outlier: 90 and 140 mins</p>
 <p>Box plot showing the distribution of 'Duration to make (min)' for total recipes. The median is 35.0, Q1 is 25.0, and Q3 is 70.0. Whiskers extend from 5.0 to 115.0. Outliers are marked at 140.0, 150.0, 195.0, 220.0, and 240.0.</p>	 <p>Box plot showing the distribution of 'Duration to make' for chicken recipes. The median is 30.0, Q1 is 25.0, and Q3 is 45.0. Whiskers extend from 10.0 to 60.0. Outliers are marked at 90.0 and 140.0.</p>
<p>lamb recipes: min: 5mins Q1: 15mins median: 32.5mins Q3: 57.5mins max: 105 mins outlier: 155 and 210 mins</p>	<p>pork recipes: min: 5mins Q1: 27.5mins median: 37.5mins Q3: 170 mins max: 240mins outlier: 495 mins</p>
 <p>Box plot showing the distribution of 'Duration to make' for lamb recipes. The median is 32.5, Q1 is 15.0, and Q3 is 57.5. Whiskers extend from 5.0 to 105.0. Outliers are marked at 155.0 and 210.0.</p>	 <p>Box plot showing the distribution of 'Duration to make' for pork recipes. The median is 37.5, Q1 is 27.5, and Q3 is 170.0. Whiskers extend from 5.0 to 240.0. An outlier is marked at 495.0.</p>
<p>beef recipes: min: 15mins Q1: 30mins median: 40mins Q3: 90mins max: 150mins outlier: 195 mins</p>	<p>fish recipes: min: 20mins Q1: 27mins median: 35mins Q3: 55mins max: 85 mins outlier: 110 mins</p>

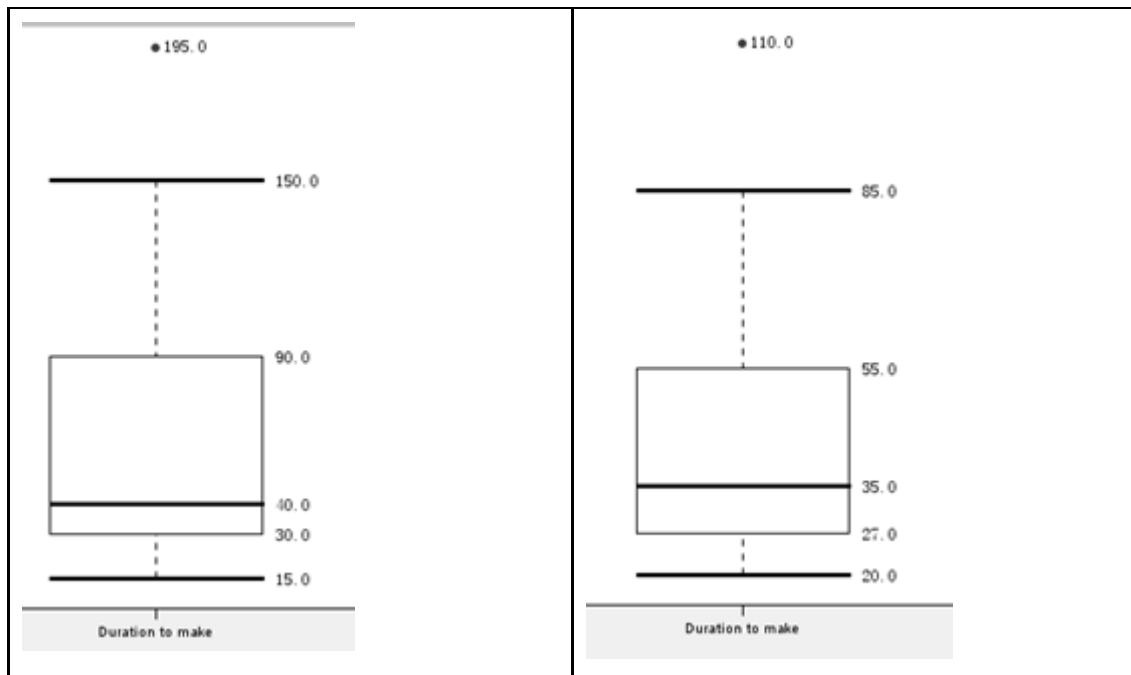


Figure 3. Duration to make (mins) for each kind of recipe

Then our group compare the duration of making each kind of meal. As a result, in figure 4, we can see the median time of making each kind of meal is similar. However, lamb recipes with less time for cooking before median. From the third quartile and maximum whisker, we can see chicken recipes have the less cooking time. The second less one is fish. Then there are lamb, beef, and pork.

In conclusion, the analyzing of duration for cooking each kind of food is for our clients to decide if they have enough time to cooking the ingredients they get.

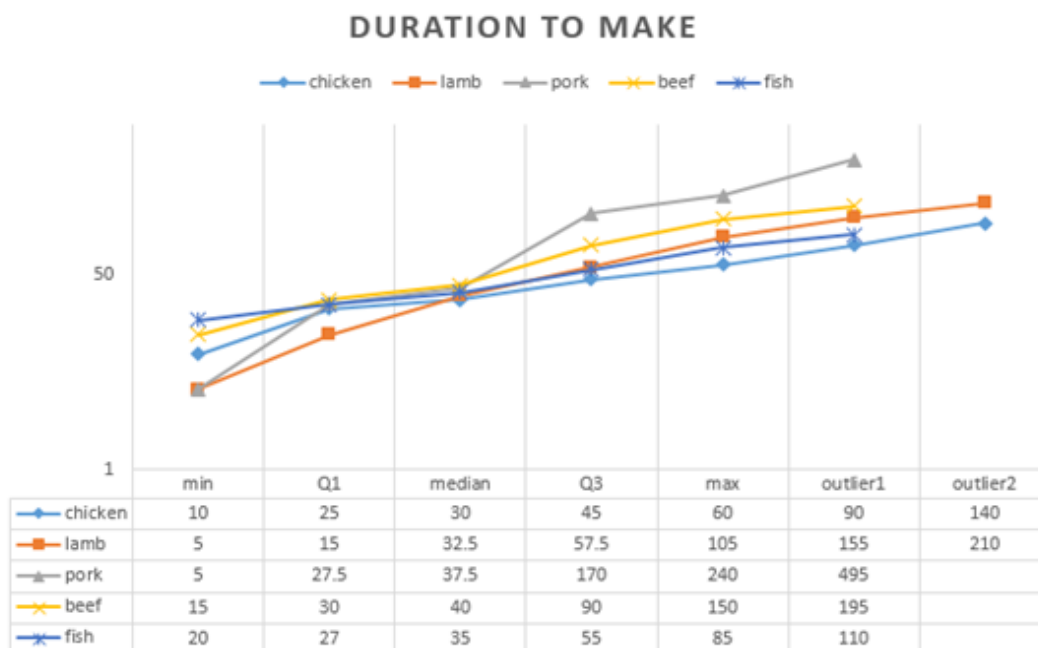


Figure 4. Duration to make (mins) for each kind of food groups

4.2. Correlation between energy, sugar, protein and fat

Our companies consider that customers may doubt the relations between sugar, protein, fat and energy. As a result, we analyse the relation between energy and sugar in Figure 5. With the very small slope which is 0.0014, we can determine that the more sugar in ingredient, more energy they have. But the relation is not strong for the line is very flat. In figure 6, we analyse the correlation between energy and protein. And in figure 7, we analyse the correlation between energy and fat. The slopes are 0.093 for protein and 0.055 for fat. The amount of slopes for those two is much bigger than the sugar one. As a result, we conclude that protein and fat have more influence on energy than sugar.

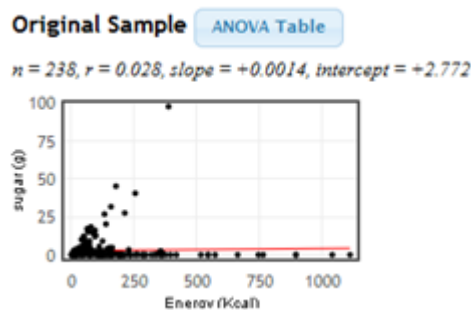


Figure 5. Correlation between Energy (Kcal) and sugar (g)

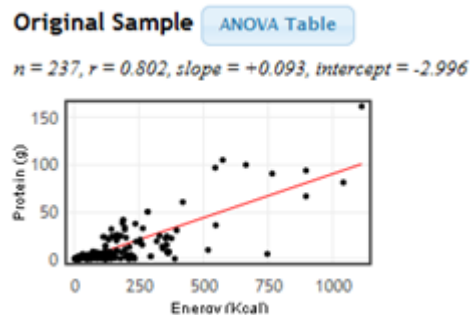


Figure 6. Correlation between energy (Kcal) and protein (g)

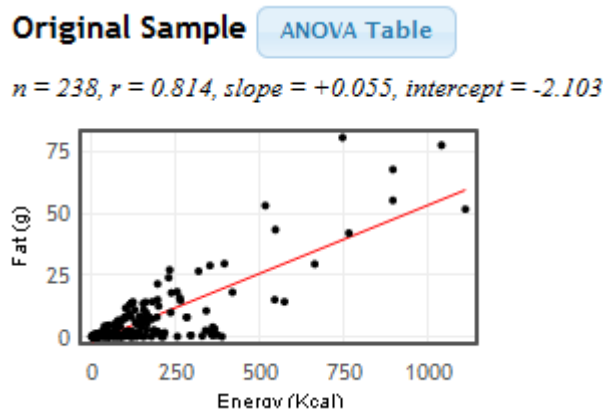


Figure 7. Correlation between energy (Kcal) and fat (g)

4.3. The nutritional values correlation between lamb, pork, fish, beef and chicken

Figure 8 shows the average energy of different groups of food between lamb, pork, fish, beef and chicken. From the data given, fish recipes have the highest average energy and pork recipes have the lowest average energy.

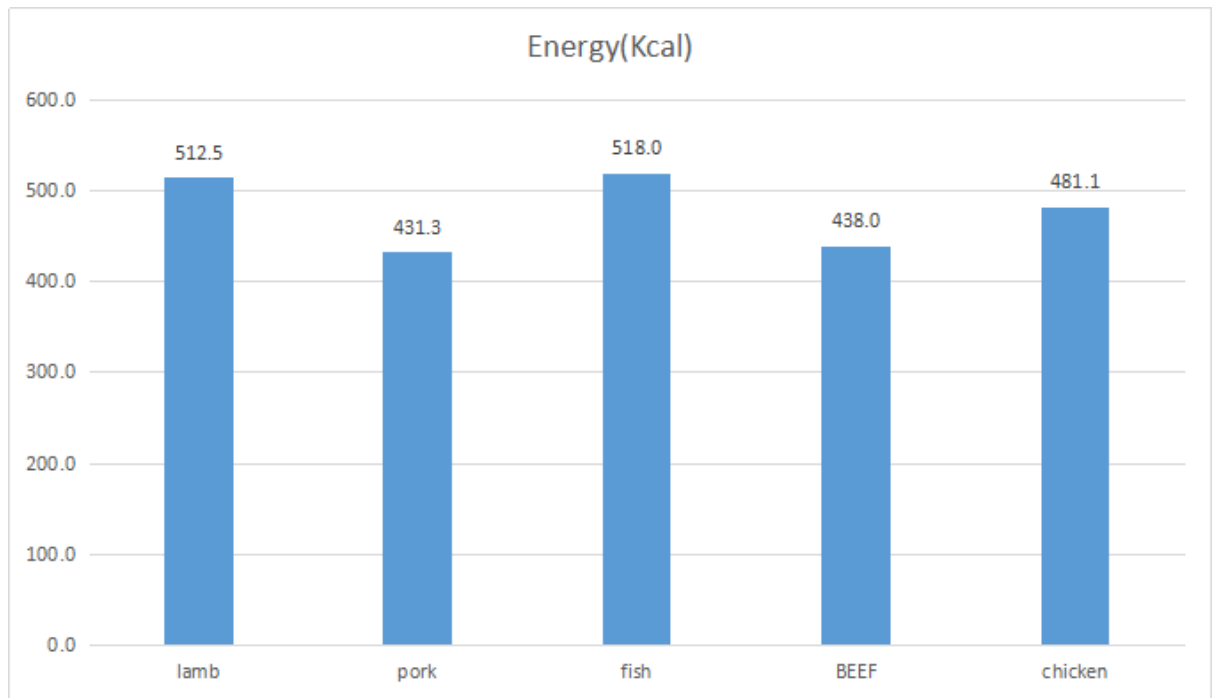


Figure 8. The average energy correlation (Kcal)

Figure 9 shows the average fat of different groups of food between lamb, pork, fish, beef and chicken. From the data given, lamb and pork recipes have the highest average fat and beef recipes have the lowest average fat. So if those people who want to lose weight can choose beef recipes.

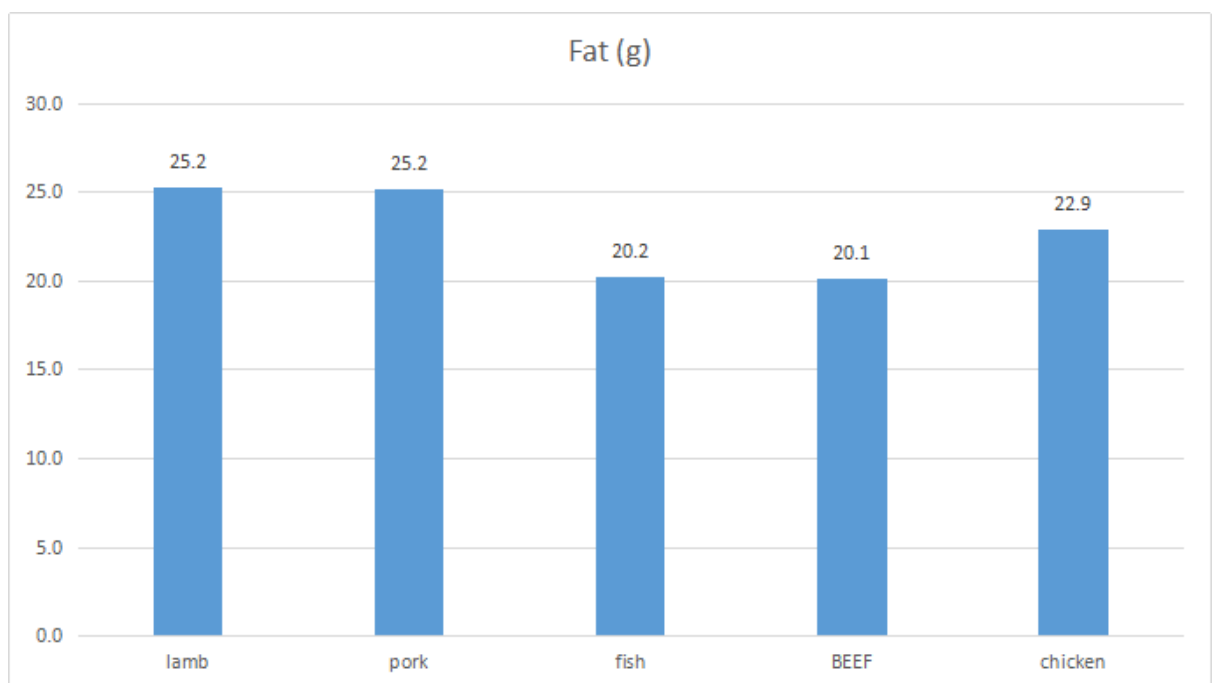


Figure 9. Average fat of each kind of food groups

This bar chart of figure 10 shows the average sugar of different groups of food between lamb, pork, fish, beef and chicken. From the data given, pork recipes

have the highest average sugar and chicken recipes have the lowest average sugar. So the diabetic can choose chicken recipes.

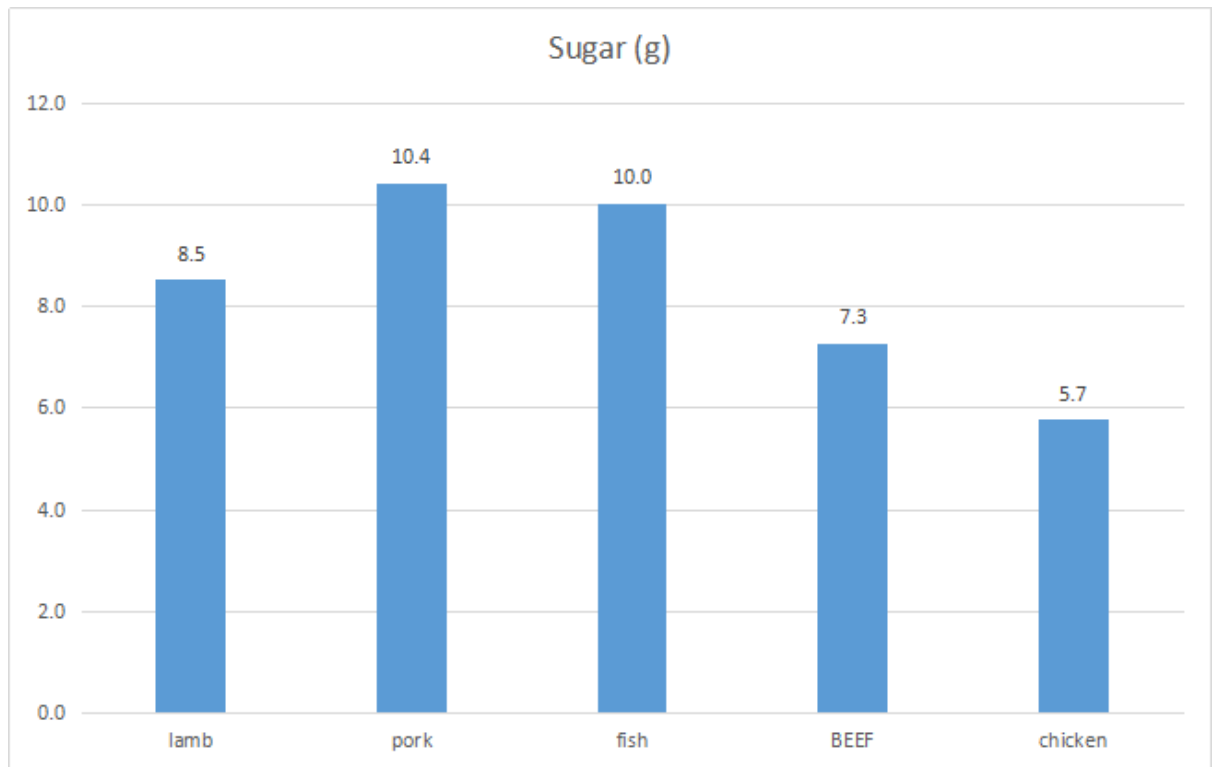


Figure 10. Average sugar of each kind of food groups

The bar chart from graph ## shows the average protein of different groups of food between lamb, pork, fish, beef and chicken. From the data given, fish recipes have the highest average protein and beef recipes have the lowest average protein.

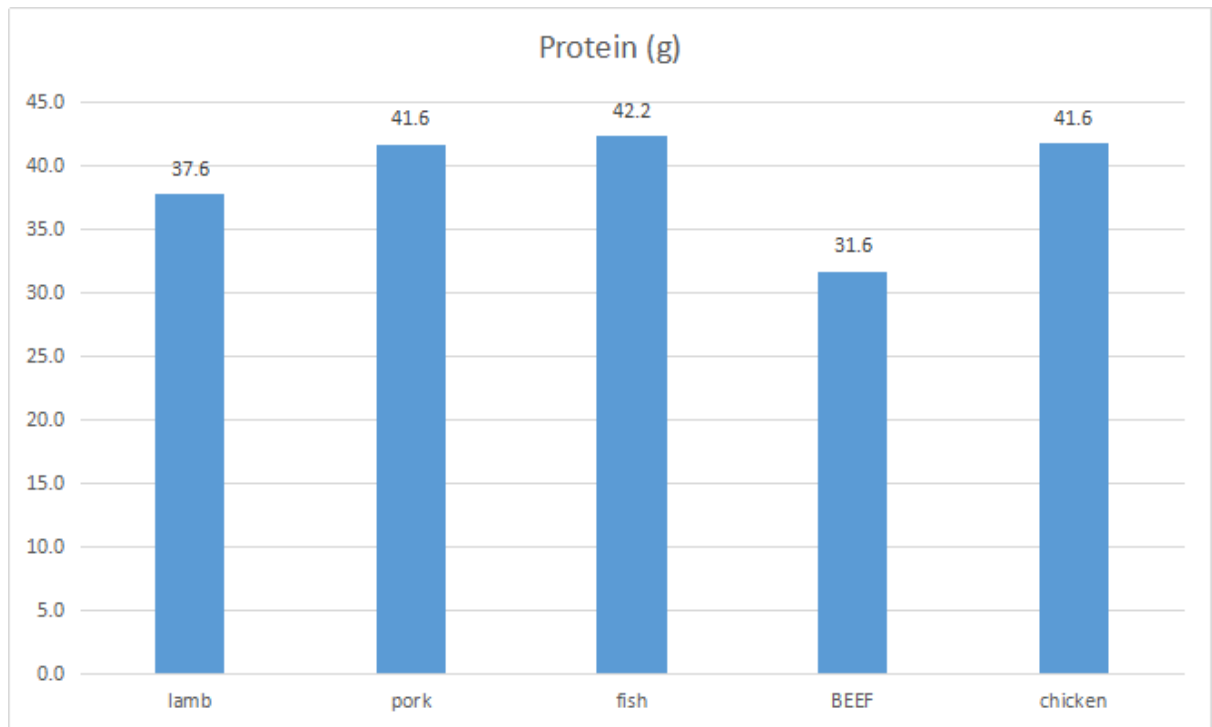


Figure 11. Average protein of each kind of food groups

4.4. The highest nutritional values of all ingredients

This table shows the highest nutritional values of all ingredients including energy, protein, fat, saturated, carbohydrates, sugar and sodium. The beef has the highest energy and protein of all recipes ingredients. Pork has the highest fat and lamb leg steaks have the highest saturated. Star anise has the most carbohydrate and sugar and teriyaki sauce has the most sodium.

Energy (Kcal)	Beef	1114
Protein (g)	Beef	162.3
Fat (g)	Pork	80.5
Saturated (g)	Lamb leg steaks	33.702
Carbohydrates (g)	Star anise	99.77
sugar (g)	Star anise	97.81
sodium (mg)	Teriyaki sauce	11039

5. Further Findings of GA Results

The genetic algorithm format is structured by the representation of [a,b,c,d,e,f,g,h,i]. The respective order represents Energy, Protein, Fat, Saturates, Carbohydrates, Sugar, Sodium, Number of ingredients and Amount of Generations.

We used [1500,50,14,2.5,9,1.5,450,10,200] as the input value, which is a nutritional value set defined for men that want to lose weight. As default, the number of ingredients selected will be 10 and the amount of generations is 200 for the meal. The system will select an initial starting sample of 10 ingredients from the database randomly, five parts will be shown as a result of selecting the best rated meal.

Figure 12 shows the result of the first generation, we can see that 10 ingredients have been generated and randomly picked by the program. It will generate the ingredients ID and the name, alongside the parameters that we initially stated from A-G. At the end of this result, there is a difference generated which is 4941.825, this describes the difference of our target value requirement and our current value being that difference. The larger the number means that the target is further whereas we are aiming to lessen that number to a minimal.

```
Original population:
[143.0, 'Milk', 43.0, 3.48, 0.97, 0.604, 4.97, 0.0, 52.0]
[189.0, 'pork tenderloin', 575.0, 105.2, 14.11, 4.816, 0.0, 0.0, 229.0]
[16.0, 'Beetroot', 70.0, 6.4, 0.73, 0.088, 13.57, 0.0, 14.0]
[245.0, 'Tamarind paste', 180.0, 24.6, 7.0, 4.41, 3.5, 0.59, 1789.0]
[114.0, 'Hake fillets', 189.0, 41.14, 1.55, 0.303, 0.0, 0.0, 125.0]
[111.0, 'Salad', 143.0, 2.68, 8.2, 1.429, 11.17, 0.0, 529.0]
[197.0, 'Red chilli', 30.0, 1.4, 0.33, 0.032, 6.61, 3.98, 7.0]
[257.0, 'turmeric', 29.0, 0.91, 0.31, 0.173, 6.31, 0.3, 3.0]
[217.0, 'Sea salt', 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2325.0]
[171.0, 'Peanut', 161.0, 7.31, 13.96, 1.78, 4.57, 1.34, 5.0]
Original difference:
4941.825
-----
```

Figure 12. The result of first time generation

However, after 200 times generations, we determine the best 10 ingredients. The nutrition value of those 10 ingredients is the best match to our target values (See Figure 13). Same as the first time generation, we calculate the difference between target value and our final results value. This time, we get the difference is 248.896, which is a huge decrease from the first time.

```
After population:
[263.0, 'white peppercorn', 21.0, 0.74, 0.15, 0.044, 4.87, 0.0, 0.0]
[189.0, 'pork tenderloin', 575.0, 105.2, 14.11, 4.816, 0.0, 0.0, 229.0]
[259.0, 'Vegetable stock', 208.0, 0.0, 0.0, 22.59, 0.0, 0.0, 0.0]
[158.0, 'Onion', 64.0, 1.76, 0.16, 0.067, 14.94, 6.78, 6.0]
[157.0, 'Olive oil', 119.0, 0.0, 13.5, 1.864, 0.0, 0.0, 0.0]
[96.0, 'Filo pastry', 57.0, 1.35, 1.14, 0.279, 9.99, 0.03, 92.0]
[249.0, 'thyme', 101.0, 5.56, 1.68, 0.467, 24.45, 0.0, 9.0]
[62.0, 'Cider Vinegar', 21.0, 0.0, 0.0, 0.0, 0.93, 0.4, 5.0]
[126.0, 'Lamb chops', 163.0, 22.98, 7.91, 3.179, 0.0, 0.0, 89.0]
[171.0, 'Peanut', 161.0, 7.31, 13.96, 1.78, 4.57, 1.34, 5.0]
After difference:
248.896
```

Figure 13. The results after 200 times generations

After we get the best 10 ingredients, we display the nutritional values from those 10 ingredients, which is [1490,144.9,52.61,35.086,59.75,8.55,435.0]. From Figure 14, we can see our result is very similar to the original target value we were aiming for.

```
-----  
Target: [1500, 50, 14, 2.5, 9, 1.5, 450]  
Total nutrient values:  
[1490.0, 144.9, 52.61000000000001, 35.086, 59.75, 8.55, 435.0]  
-----
```

Figure 14. Compare target value and genetic algorithm result

Because our database for ingredients and recipes are linked together, we can determine the most suitable recipe according to the ingredients. In this case, the suggested recipe is 'honey mustard chicken pot with parsnips'. Figure 15 shows the information of this recipe. There are eight ingredients for this meal. The ingredients in this recipe with asterisk in the front is the same ingredients after generation. As a result, there are 4 similar ingredients in this recipe which is also in the best matched ingredients. The result has a similarity of 4 and a percentage of 50 from the eight ingredients within suggested meal.

```
Suggested meal:  
[94.0, 'Honey mustard chicken pot with parsnips', 40.0, 4.0]  
Ingredients:  
* [157.0, 'Olive oil', 119.0, 0.0, 13.5, 1.864, 0.0, 0.0, 0.0]  
[51.0, 'Chicken thigh', 396.0, 30.28, 29.51, 8.077, 0.31, 0.0, 155.0]  
* [158.0, 'Onion', 64.0, 1.76, 0.16, 0.067, 14.94, 6.78, 6.0]  
[169.0, 'Parsnip', 75.0, 1.2, 0.3, 0.05, 17.99, 4.8, 10.0]  
* [259.0, 'Vegetable stock', 208.0, 0.0, 0.0, 22.59, 0.0, 0.0, 0.0]  
[151.0, 'Mustard', 3.0, 0.19, 0.17, 0.011, 0.29, 0.05, 55.0]  
[120.0, 'Honey', 64.0, 0.06, 0.0, 0.0, 17.3, 17.25, 1.0]  
* [249.0, 'thyme', 101.0, 5.56, 1.68, 0.467, 24.45, 0.0, 9.0]  
With similarity: 4  
Percentage: 50.0
```

Figure 15. Ingredients and nutrition information for suggested meal

Figure 16 presents the trend of difference. The X-axis represents the number of generation, while the Y-axis represents the difference between the result of the current generation and the target value. From this diagram, we can see there is a steep decrease in difference from the first few generations. From the 25th to the 150th generations, the difference are going down very slowly. After 150 generations, the difference does not decrease anymore, showing a straight line in the figure at the difference value of 250. The reason why the difference is going down slowly and the absence of decrease in later generations is because the algorithm cannot find any better ingredient to be

swapped where it will decrease the difference. However, from this diagram, we know the genetic algorithm is working in this project.

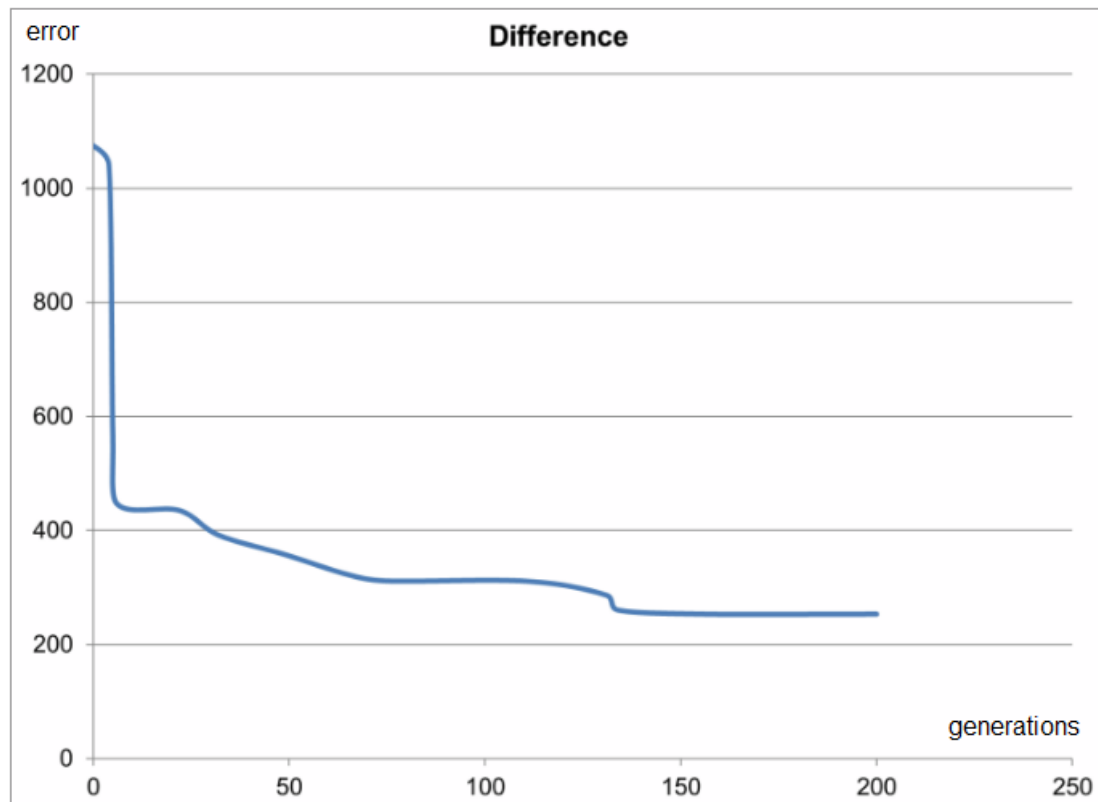


Figure 16. Difference of generations diagram

In addition, we analyse how does each parameter change during generations. Using the previous set the target value [1500,50,14,2.5,9,1.5,450,10,200], which is the nutrition value for a man who wants to lose weight. We then record values of each parameter from every single generation. As a result, we transfer those data into line charts, to show the distinct difference of each different generations for these clients to see. (See Figure 17)

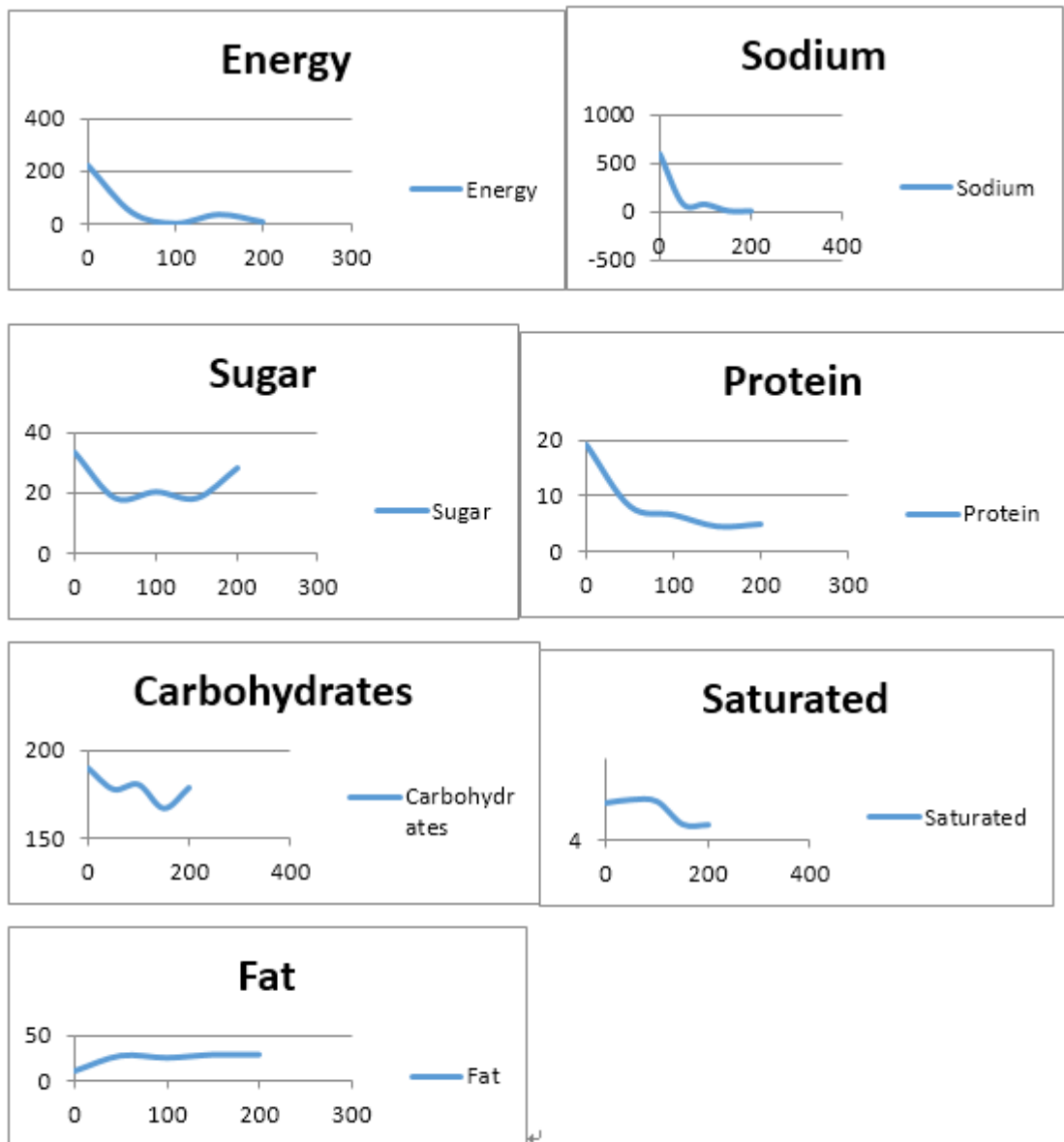


Figure 17. Difference changes of each parameters during 200 times generations

For these line charts, x-axis represents the amount of generations and Y-axis represents the difference between target value and the value after generation.

For these line charts, x-axis represents the amount of generations and Y-axis represents the difference between target value and the value after generation.

The ideal result that we are looking for in line chart should be declining and eventually reaching zero after 200th generation for being the best match. The value of difference that equals to zero means that there is no difference between the nutrition values of the ingredients we get and the target value we set. The value of difference in energy, sodium, protein and saturated line charts are going down during generations. However, the value of difference in fat, carbohydrates and sugar line charts are going up and down during

generations. This is because when the process of genetic algorithm happens, we keep changing ingredients randomly. So some ingredients with low energy but high sugar, may lead to the energy decreasing and increasing of the sugar because the benefits of the target difference outcome is much better by doing so.

In conclusion, it is obviously found that we can find out the best ingredients and get the recipe which contains those ingredients through genetic algorithm. Also, this program can be used by people with different diet requirements by simply changing the input values and number of ingredients. Moreover, in our mid-project update, we noticed that our recipe data has a field for the duration of cooking. If we can implement them into our program, that will be helpful for people who have limited time for cooking. We believe the result will be better if we have enough big database.

6. Difficulties Encountered

Numerous Ingredients

There are numerous ingredients in a recipe, which some of them are just the same ingredient but divide into different parts. Including all these separated ingredients in the database is difficult, although they increase the details on each recipe giving customers more options, these similar ingredients make the database hard to manipulate. To solve this problem, we have to eliminate the separated ingredients and combine them back together, so we don't have to deal with duplicate ingredients.

Rearrange data format

At the beginning of the project, the format of the data we collected were not in the correct format we wanted. For example, the data from our database are declare as string not number and there are no connections between data. Therefore, when we try to input those data into the program we use, we have to rearrange the format of the data into the array we want, so that the result will be more clear and understandable for these clients.

Search for food database

There is numerous food website with recipes and methods in the internet, but due to the fact they are profit making organizations, they don't share their database for free. In this case we don't have fund to purchase the access to their database so what we have to do is generate our own database. We have to collect recipes with ingredients and methods from numerous website and match each ingredient with nutrition values.

Sample Size

As mention before the database is generated by ourselves, we got smaller data size compared to those profit making organization. For example, there are limitations in the types of meat or the amount of ingredients in the database, which might lead to limited choice for the customers and affect the result. We solve this by adding as many recipe as we can to retain the accuracy of the result.

7. Evaluation

In order to see if genetic algorithm is the best classifier for our choice, we use two different models to compare with it. In this section, we are trying to find out the relationship between energy, fat, sugars and the nutrition grade. The following pages will give a simple compare between these three classifications.

7.1 Decision Tree

Decision tree is a predictive model of machine learning methods, it represents a mapping between object attributes and values of an object. Therefore, we chose it as the first algorithms for comparing. In our project, it is clear too see that the node error is 0.70213 and the average class error is 55%.

```

Summary of the Decision Tree model for Classification (built using 'rpart'):

n=188 (406 observations deleted due to missingness)

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 188 132 d (0.15 0.12 0.15 0.3 0.27)
  2) energy_100g< 1557 99 70 a (0.29 0.21 0.2 0.2 0.091)
    4) sugars_100g< 19.6 78 49 a (0.37 0.27 0.17 0.13 0.064)
      8) fat_100g< 1.825 31 10 a (0.68 0.19 0.032 0.032 0.065) *
      9) fat_100g>=1.825 47 32 b (0.17 0.32 0.26 0.19 0.064)
        18) fat_100g< 9.25 30 19 b (0.27 0.37 0.27 0.1 0)
          36) fat_100g>=6.785 8 5 a (0.38 0.38 0 0.25 0) *
          37) fat_100g< 6.785 22 14 b (0.23 0.36 0.36 0.045 0)
            74) energy_100g< 430 9 5 b (0.33 0.44 0.11 0.11 0) *
            75) energy_100g>=430 13 6 c (0.15 0.31 0.54 0 0) *
              19) fat_100g>=9.25 17 11 d (0 0.24 0.24 0.35 0.18) *
        5) sugars_100g>=19.6 21 11 d (0 0 0.33 0.48 0.19) *
    3) energy_100g>=1557 89 47 e (0 0.022 0.1 0.4 0.47)
      6) sugars_100g< 31.05 53 28 d (0 0.038 0.17 0.47 0.32)
        12) fat_100g< 30.3 23 8 d (0 0.043 0.22 0.65 0.087) *
        13) fat_100g>=30.3 30 15 e (0 0.033 0.13 0.33 0.5) *
          7) sugars_100g>=31.05 36 11 e (0 0 0 0.31 0.69)
            14) fat_100g< 5.46 9 1 d (0 0 0 0.89 0.11) *
            15) fat_100g>=5.46 27 3 e (0 0 0 0.11 0.89) *

Classification tree:
rpart(formula = nutrition_grade_fr ~ ., data = crs$dataset[crs$train,
  c(crs$input, crs$target)], method = "class", parms = list(split = "information"),
  control = rpart.control(usesurrogate = 0, maxsurrogate = 0))

Variables actually used in tree construction:
[1] energy_100g fat_100g sugars_100g

Root node error: 132/188 = 0.70213

n=188 (406 observations deleted due to missingness)

      CP nsplit rel error  xerror    xstd
1 0.113636      0  1.00000 1.05303 0.045599
2 0.075758      1  0.88636 0.93939 0.049221
3 0.060606      2  0.81061 0.90152 0.050066
4 0.053030      3  0.75000 0.85606 0.050865
5 0.037879      5  0.64394 0.84091 0.051080
6 0.015152      6  0.60606 0.75000 0.051863
7 0.011364      7  0.59091 0.75000 0.051863
8 0.010000      9  0.56818 0.71970 0.051934

Time taken: 0.01 secs

Rattle timestamp: 2016-10-11 01:22:51 iGakki
=====

```

Figure 18. Summary of Decision Tree model

Tree as rules:

```
Rule number: 12 [nutrition_grade_fr=d cover=23 (12%) prob=15.00]
  energy_100g>=1557
  sugars_100g< 31.05
  fat_100g< 30.3

Rule number: 13 [nutrition_grade_fr=e cover=30 (16%) prob=10.00]
  energy_100g>=1557
  sugars_100g< 31.05
  fat_100g>=30.3

Rule number: 5 [nutrition_grade_fr=d cover=21 (11%) prob=10.00]
  energy_100g< 1557
  sugars_100g>=19.6

Rule number: 14 [nutrition_grade_fr=d cover=9 (5%) prob=8.00]
  energy_100g>=1557
  sugars_100g>=31.05
  fat_100g< 5.46

Rule number: 19 [nutrition_grade_fr=d cover=17 (9%) prob=6.00]
  energy_100g< 1557
  sugars_100g< 19.6
  fat_100g>=1.825
  fat_100g>=9.25

Rule number: 15 [nutrition_grade_fr=e cover=27 (14%) prob=3.00]
  energy_100g>=1557
  sugars_100g>=31.05
  fat_100g>=5.46

Rule number: 36 [nutrition_grade_fr=a cover=8 (4%) prob=2.00]
  energy_100g< 1557
  sugars_100g< 19.6
  fat_100g>=1.825
  fat_100g< 9.25
  fat_100g>=6.785

Rule number: 74 [nutrition_grade_fr=b cover=9 (5%) prob=1.00]
  energy_100g< 1557
  sugars_100g< 19.6
  fat_100g>=1.825
  fat_100g< 9.25
  fat_100g< 6.785
  energy_100g< 430

Rule number: 8 [nutrition_grade_fr=a cover=31 (16%) prob=1.00]
  energy_100g< 1557
  sugars_100g< 19.6
  fat_100g< 1.825

Rule number: 75 [nutrition_grade_fr=c cover=13 (7%) prob=0.00]
  energy_100g< 1557
  sugars_100g< 19.6
  fat_100g>=1.825
  fat_100g< 9.25
  fat_100g< 6.785
  energy_100g>=430
```

[1] 1 13 14 2 15 17 16 12 3 5 18 11 19 6 7 9 8 4 10

Rattle timestamp: 2016-10-11 01:23:21 iGakki

Figure 19. Decision Tree model rules

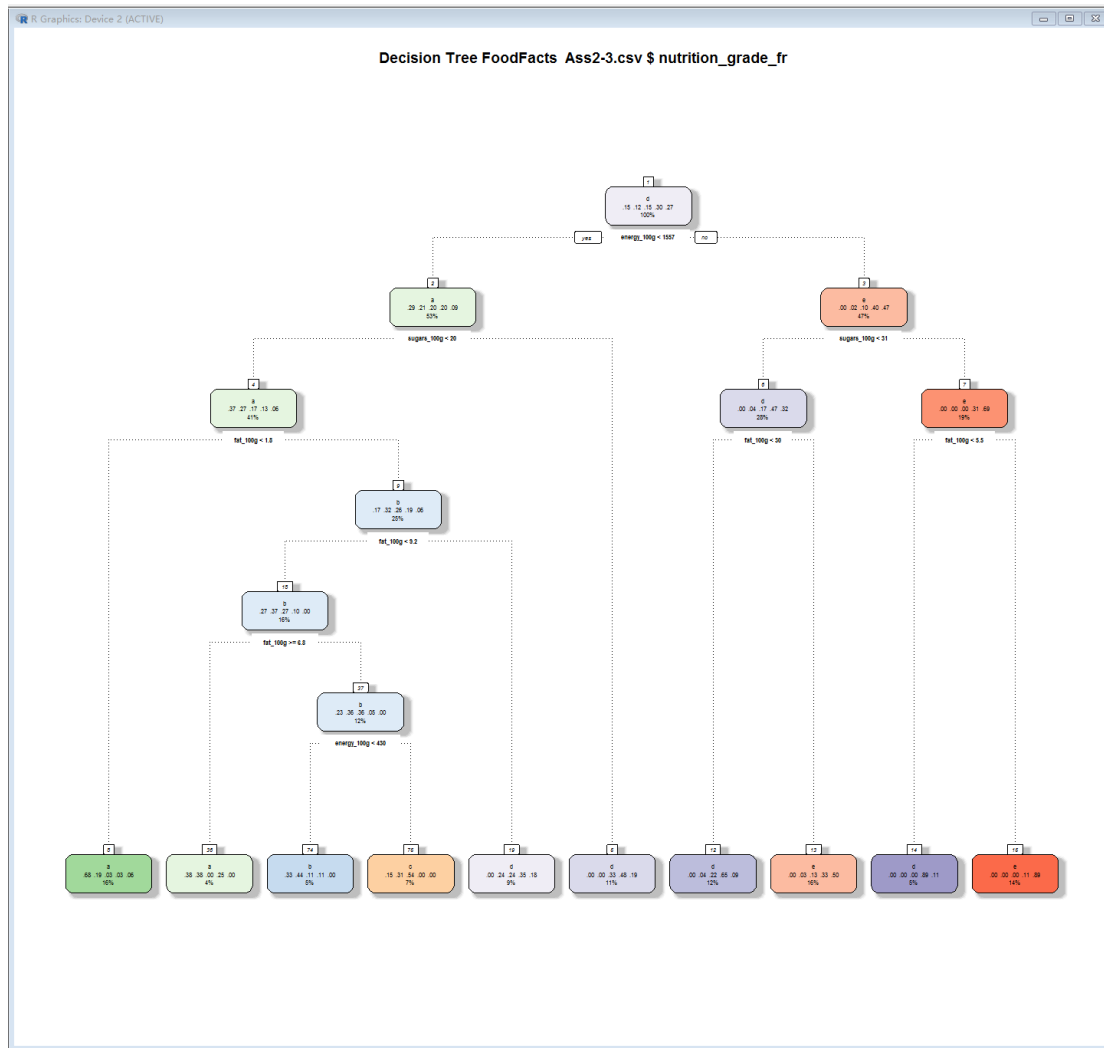


Figure 20. Decision Tree View

Error matrix for the Decision Tree model on FoodFacts Ass2-3.csv [validate] (counts):

Actual	Predicted				
	a	b	c	d	e
a	2	0	1	1	0
b	5	0	0	4	0
c	3	0	2	2	1
d	1	0	0	16	6
e	0	0	0	2	7
<NA>	4	0	0	70	0

Error matrix for the Decision Tree model on FoodFacts Ass2-3.csv [validate] (proportions):

Actual	Predicted					Error
	a	b	c	d	e	
a	0.04	0	0.02	0.02	0.00	0.50
b	0.09	0	0.00	0.08	0.00	1.00
c	0.06	0	0.04	0.04	0.02	0.75
d	0.02	0	0.00	0.30	0.11	0.30
e	0.00	0	0.00	0.04	0.13	0.22

Overall error: 49%, Averaged class error: 55%

Rattle timestamp: 2016-10-11 01:22:11 iGakki

Figure 21. Decision Tree error matrix

7.2 Random Forest

The second algorithm is random forest, which is a classifier comprising a plurality of decision tree. It generates the category of output through running several individual tree output categories, hence, it could be a method produce high accuracy classifier for a variety of materials. In our case, the result for average class error is 42%.

```

Summary of the Random Forest Model
=====

Number of observations used to build the model: 594
Missing value imputation is active.

Call:
randomForest(formula = nutrition_grade_fr ~ .,
              data = crs$dataset[crs$sample, c(crs$input, crs$target)],
              ntree = 500, mtry = 1, importance = TRUE, replace = FALSE, na.action = randomForest::na.roughfix)

              Type of random forest: classification
              Number of trees: 500
No. of variables tried at each split: 1

              OOB estimate of error rate: 18.35%
Confusion matrix:
  a b c d e class.error
a 15 6 2 6 0 0.48275862
b 9 9 0 19 1 0.76315789
c 3 5 2 17 2 0.93103448
d 3 3 4 426 11 0.04697987
e 0 0 3 15 33 0.35294118

Variable Importance
=====
              a b c d e MeanDecreaseAccuracy
fat_100g 19.92 12.37 7.18 28.39 28.18 38.43
sugars_100g 29.70 6.52 1.29 29.09 52.36 41.43
energy_100g 3.14 12.77 0.96 24.52 26.44 32.01
              MeanDecreaseGini
fat_100g 43.81
sugars_100g 44.25
energy_100g 42.84

Time taken: 0.16 secs

Rattle timestamp: 2016-10-11 01:24:39 iGakki
=====

```

Figure 22. Summary of Random Forest model

```

Error matrix for the Random Forest model on FoodFacts Ass2-3.csv [validate] (counts):

      Predicted
Actual a b c d e
a 2 1 0 1 0
b 1 3 0 3 0
c 0 2 3 3 0
d 0 0 1 19 3
e 0 0 0 2 7

Error matrix for the Random Forest model on FoodFacts Ass2-3.csv [validate] (proportions):

      Predicted
Actual a b c d e Error
a 0.04 0.02 0.00 0.02 0.00 0.50
b 0.02 0.06 0.00 0.06 0.00 0.57
c 0.00 0.04 0.06 0.06 0.00 0.62
d 0.00 0.00 0.02 0.37 0.06 0.17
e 0.00 0.00 0.00 0.04 0.14 0.22

Overall error: 33%, Averaged class error: 42%

Rattle timestamp: 2016-10-11 01:22:11 iGakki
=====

```

Figure 23. Random Forest error matrix

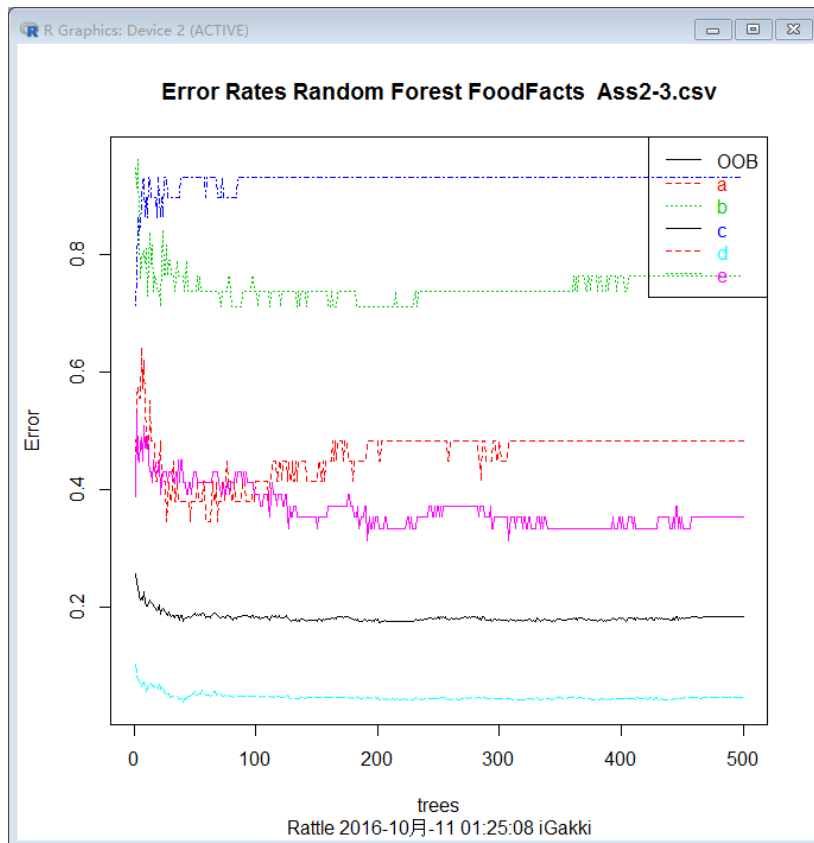


Figure 23. Random Forest error rates

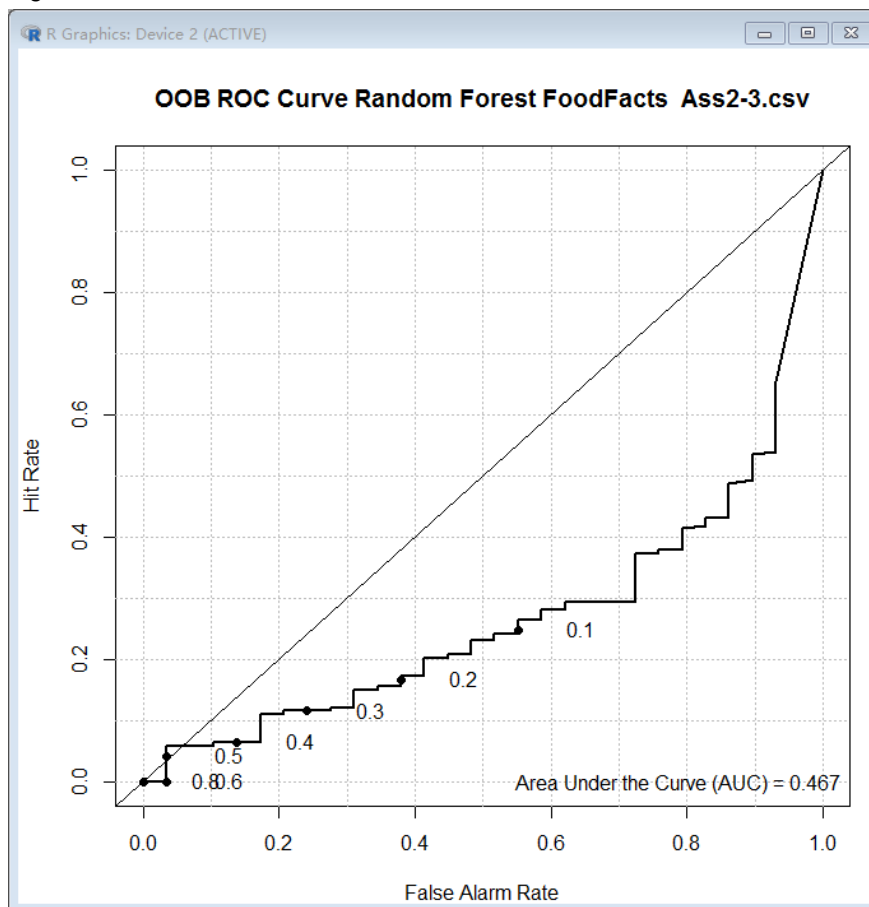


Figure 24. OOB ROC Curve of Random Forest model

7.3 Recommendation

Here, we use **Pseudo R^2** to find out which model is better than the rest.

When examine the pseudo r square, the higher value indicating better model fit.

Therefore, through the three image present under, we can summary the result as following table:

Model	Pseudo R^2 Value
Decision Tree	0.7124
Random Forest	0.7584
Genetic Algorithm	0.7734

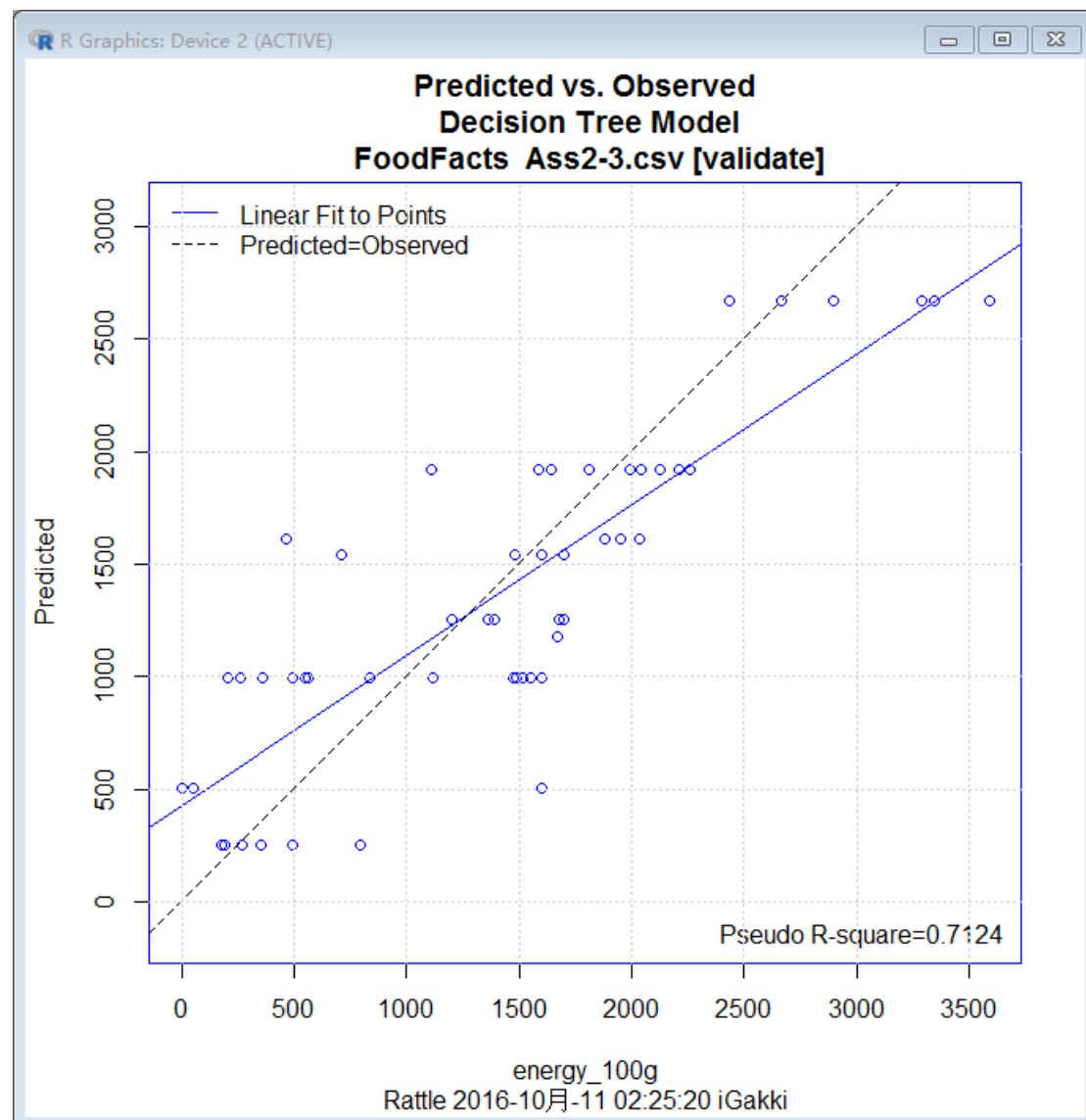


Figure 25. Predicted V.S. Observed Decision Tree model

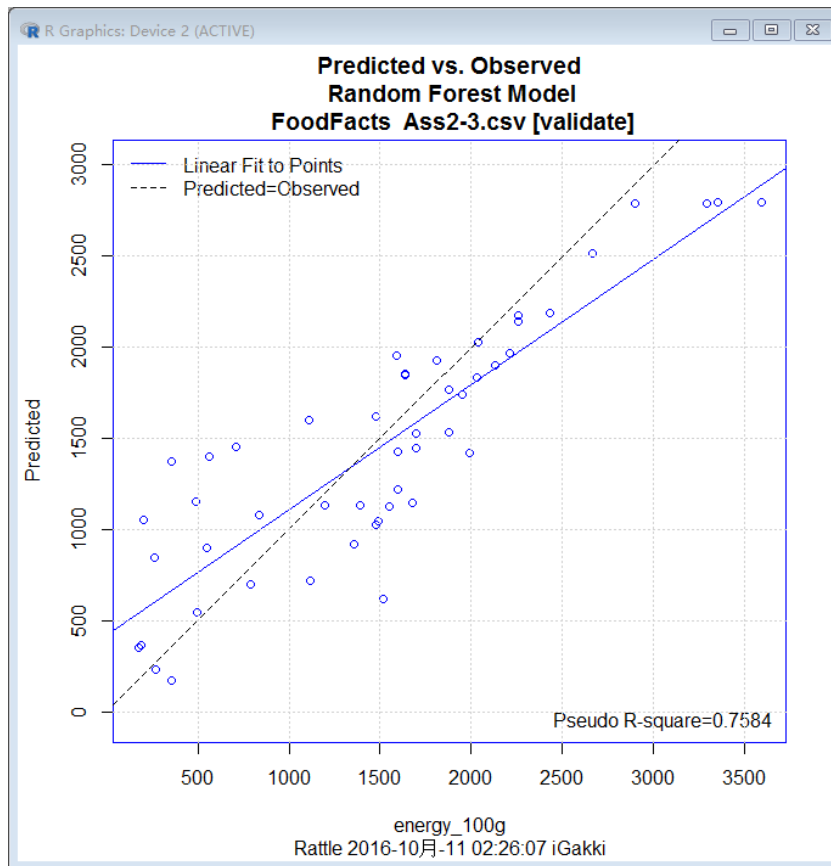


Figure 26. Predicted V.S. Observed Random Forest model

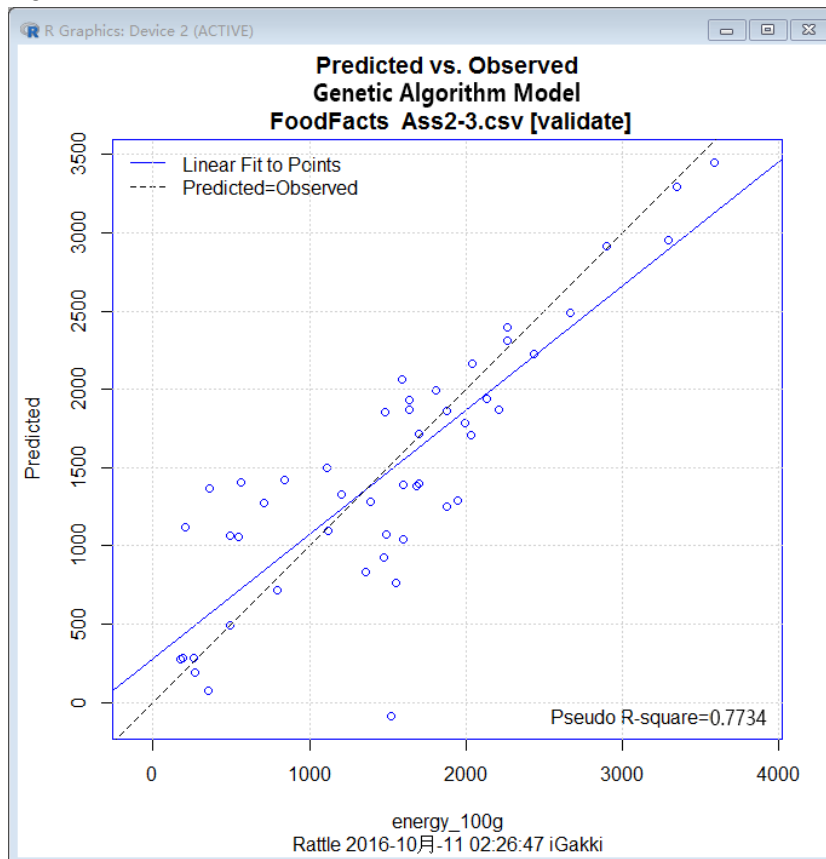


Figure 27. Predicted V.S. Genetic Algorithm model

As a result, we could say GA is slightly better than the other algorithms, however, there are also disadvantages of genetic algorithm. One major issue is that it takes a lot time to decent sized population and generations before we get a good result. In our project, we only take a small part out of the whole dataset, therefore, the consumption of training time is still reasonable. But, when dealing with a huge simulation, it might take days for the results.

Overall, in this project, the pseudo r square from genetic algorithm is slightly higher than the other two classifications, therefore, we suggest to use the Genetic Algorithm in this case.

The following link contains our short presentation for our project:

<https://www.youtube.com/watch?v=u7jUx6e5ll0&feature=youtu.be>