Министерство науки и образования Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Лабораторная работа №1**

Дисциплина: «Системы искусственного интеллекта»

«Генетические алгоритмы»

Вариант 1

Выполнил студент  
группы ИВТАCбд-41  
Абросимов К. С.

Проверил:  
преподаватель кафедры ВТ  
Хайруллин И. Д.

Ульяновск, 2024

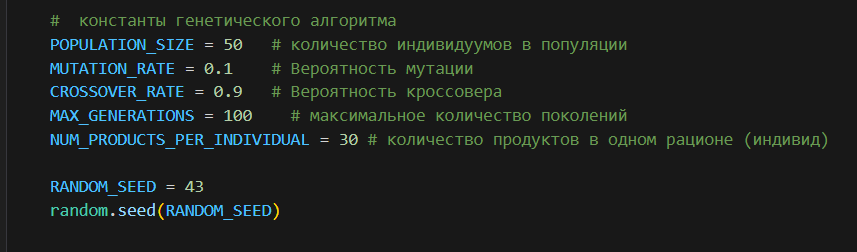
**Постановка задачи**

1. Необходимо разработать программу на языке python, реализующую генетический алгоритм по предложенному вариантом заданию.
2. Провести эксперименты по разным способам скрещивания (не менее 3-х), разным способам мутирования (не менее трех). Результат отобразить в виде графиков
3. Моделирование данных производить на основе максимально правдоподобных данных. Т.е. если рассматривается задача, в которой есть калорийность продуктов, то должны использоваться данные о реальных продуктах с реальной калорийностью.

**Ход работы**

Вариант содержит такую постановку задачи: Дано N наименований продуктов, для каждого из которых известно m характеристик. Необходимо получить самый дешевый рацион из k наименований, удовлетворяющий заданным медицинским нормам для каждой из m характеристик.

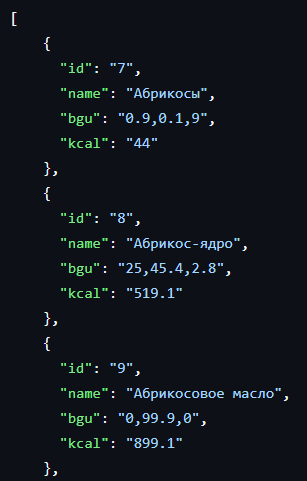
1) Объявление констант



2) Импортирование реальных данных продуктов

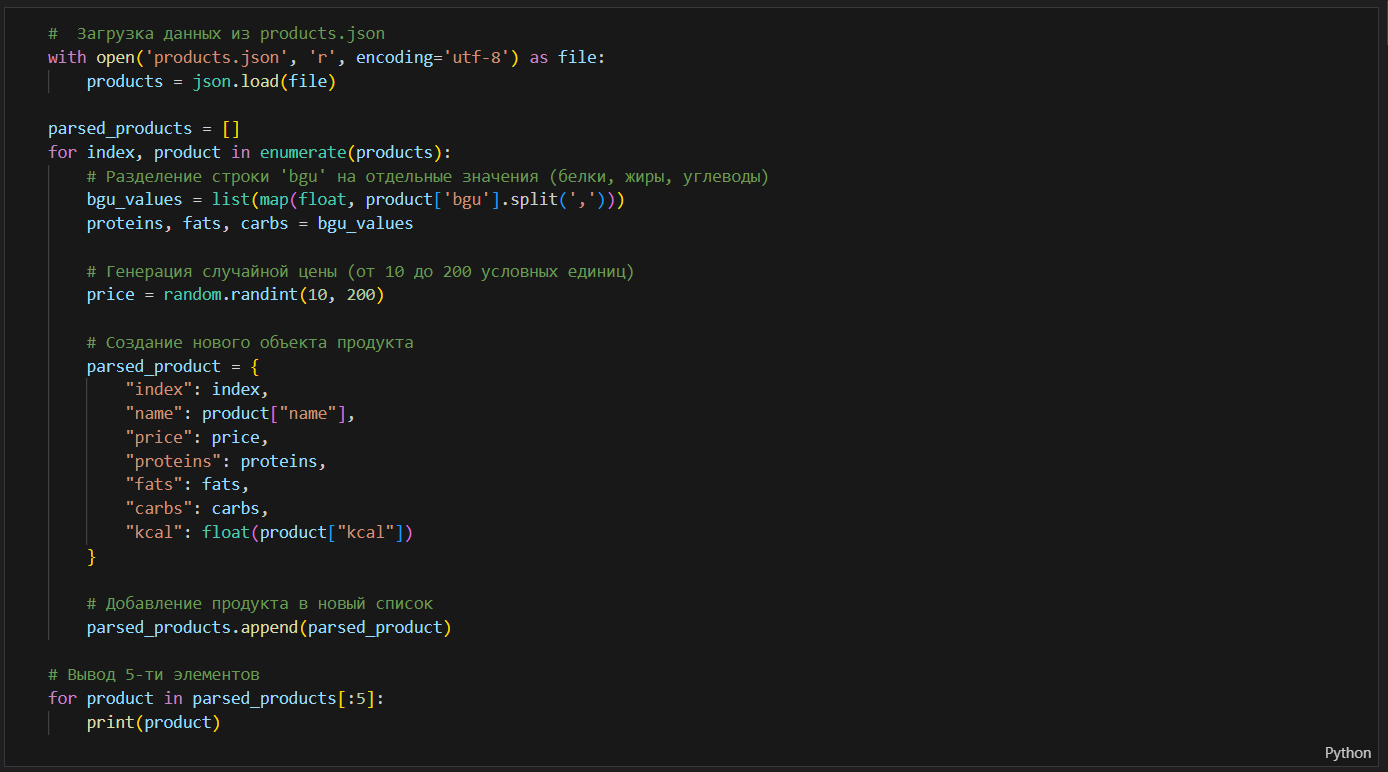
Данные продуктов были взяты с github: <https://github.com/goodwin74/prod_rus/blob/main/products.json>

База продуктов 2021 (фрукты, овощи, ягоды и тд) на русском языке с данными о белках, жирах, углеводов и калорийности (килокалорий).



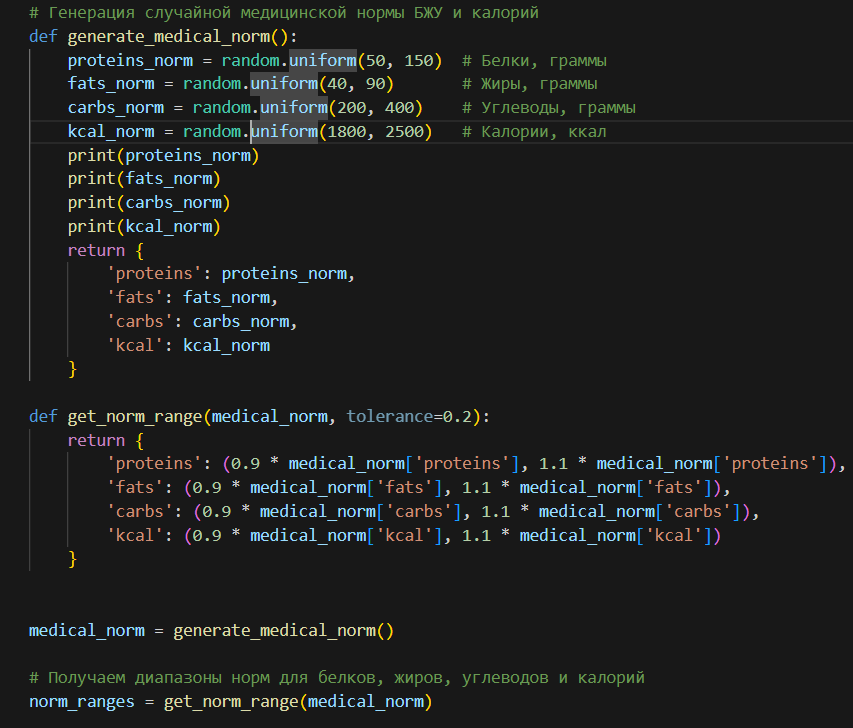
Каждый продукт содержит id (в рамках этой работы оно нам не нужно), наименование, содержание белков, жиров и углеводов и килакалории.

Но нам нужно также указать цену каждого продукта. Задание цены будет генерироваться **случайно**.



3) Генерация случайно медицинской нормы.

В условии задачи необходимо, чтобы рацион продуктов не заступал границ медицинских норм. Эти промежутки также будут генерироваться **случайно**.



4) Постройка тепловой матрицы для первичного анализа продуктов.

Матрица покажет, какие продукты сразу же будут наименее приспособлены в качестве выбора для рациона.

Самый лучший исход: Находится рацион с минимальной стоимостью, каждый из продуктов которой соответствует медицинской норме.

Самый худший исход: Находится рацион с максимальной стоимостью, каждый из продуктов которой сильно превысил медицинскую норму.

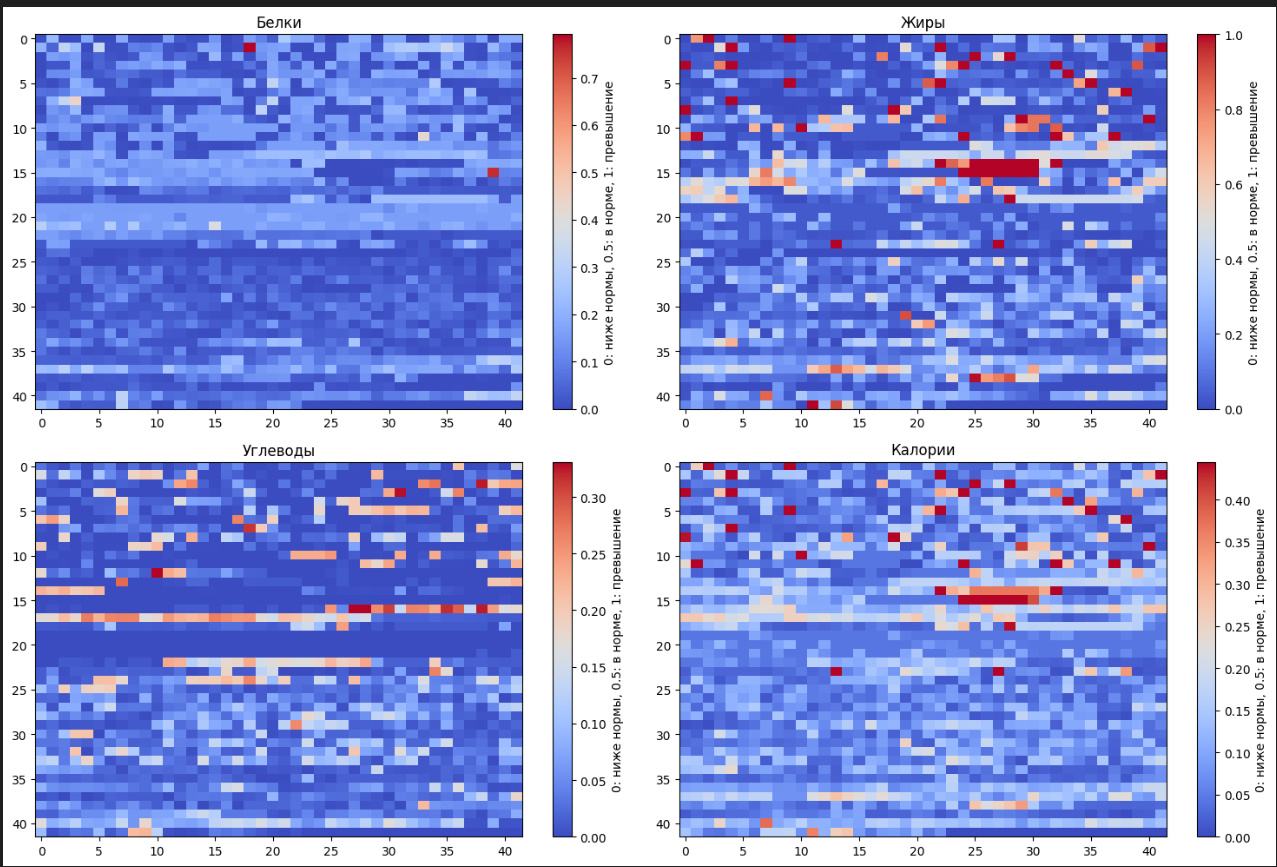


Рис. 1 «Тепловая матрица»

5) 1-й этап генетического алгоритма – создание популяции.

Для начала генетического алгоритма необходимо создать изначальную популяцию. Каждый индивид – это рацион продуктов. Продукты выбираются случайно из общего массива продуктов, и не повторяются во всей популяции.



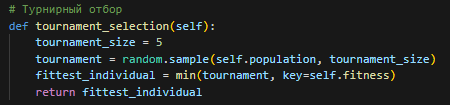
6) 2-й этап генетического алгоритма – проверка условия.

Каждый индивид проходит через функцию приспособленности (в данной работе – штрафов). Рациону назначаются штрафы за каждое отклонение медицинской нормы каждого продукта, в конце выходит значение из общей стоимости всех продуктов + штрафы.

Чем выше значение – тем менее приспоблен индивид.

7) 3-й этап – эволюция популяции

Эволюция состоит из трёх этапов: отбор, кроссовер и мутация.

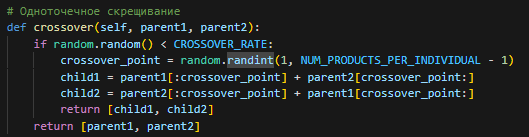


Использовался турнирный отбор: случайно отбираются несколько индивидуумов, после чего выбираются победители, которые будут иметь право оставить «потомство» на следующие поколения.

Скрещивание происходит между двумя отобранными родителями для получения лучшего потомка.

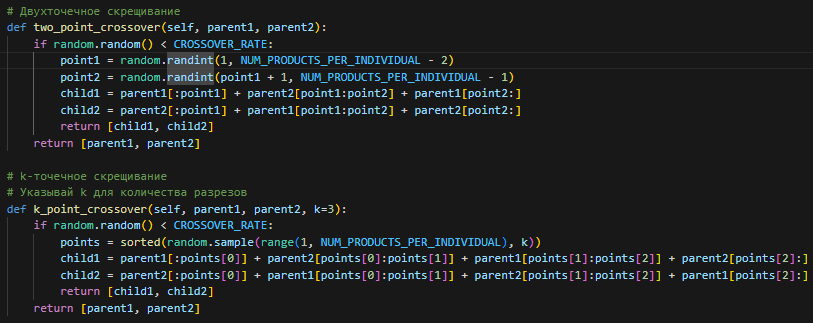
* Одноточечное скрещивание

Общее количество хромосом родителя делятся наполовину, после чего производится обмен половинок каждого родителя:



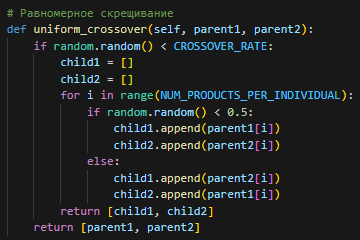
* Двухточечное, k-точечное скрещивание

Общее количество хромосом родителя делятся два раза (k раз), после чего происходит обмен:



* Равномерное скрещивание

Последовательно и попарно выбираются хромосомы родителей. В зависимости от результата генератора случайных чисел, родители обмениваются хромосомами:

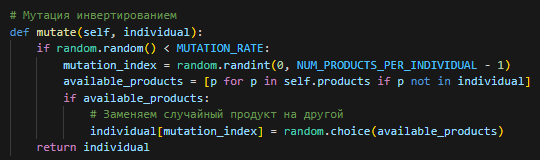


Последний этап эволюции – мутации. Этот механизм позволяет расширять область поиска решения задачи и сохранять разнообразие популяции.

Реализованы три вида мутации:

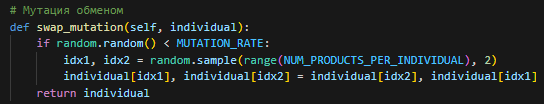
* Мутация инвертирования

Случайная хромосома потомка меняется на другую случайную хромосому. В данном случае нынешний продукт заменяется на случайный из общего массива продуктов.



* Мутация обменом

Гены у потомка случайно меняются местами:



* Мутация обращением

Выбирается некоторая последовательность генов внутри потомка, после чего инвертируется:

def reverse\_mutation(self, individual):

        if random.random() < MUTATION\_RATE:

            start, end = sorted(random.sample(range(NUM\_PRODUCTS\_PER\_INDIVIDUAL), 2))

            individual[start:end] = individual[start:end][::-1]

        return individual

После этого получается новая популяция. Цикл будет продолжаться до тех пор, пока количество поколений не превысит установленное значение.

Отобразим полученные данные и графики:

Поколение 0: Макс. штраф = 12463.107173837852, Средний штраф = 41424.20801533128

Поколение 1: Макс. штраф = 9180.577173837859, Средний штраф = 28296.76491669904

Поколение 2: Макс. штраф = 9180.577173837857, Средний штраф = 19513.29433319587

Поколение 3: Макс. штраф = 7513.107173837856, Средний штраф = 11581.718597983008

Поколение 4: Макс. штраф = 4987.298119939707, Средний штраф = 9262.883002417955

Поколение 5: Макс. штраф = 4987.298119939707, Средний штраф = 7515.888611681931

Поколение 6: Макс. штраф = 4987.298119939707, Средний штраф = 6625.523781980266

Поколение 7: Макс. штраф = 4921.898119939708, Средний штраф = 5484.660088772961

Поколение 8: Макс. штраф = 4816.398119939708, Средний штраф = 4983.960317888556

Поколение 9: Макс. штраф = 4816.398119939708, Средний штраф = 4956.070317888557

Поколение 10: Макс. штраф = 4380.898119939707, Средний штраф = 4917.445898966519

Поколение 11: Макс. штраф = 4380.898119939707, Средний штраф = 4712.65883393361

Поколение 12: Макс. штраф = 4057.898119939707, Средний штраф = 4477.630307970092

Поколение 13: Макс. штраф = 4057.898119939707, Средний штраф = 4245.2830935498405

Поколение 14: Макс. штраф = 3914.7289634840217, Средний штраф = 4246.205946338042

Поколение 15: Макс. штраф = 3565.3981199397076, Средний штраф = 4164.53596936238

Поколение 16: Макс. штраф = 3565.3981199397076, Средний штраф = 4056.667748389192

Поколение 17: Макс. штраф = 3565.3981199397067, Средний штраф = 3837.118554480449

Поколение 18: Макс. штраф = 3490.7289634840217, Средний штраф = 3605.501522390343

Поколение 19: Макс. штраф = 3314.159758548468, Средний штраф = 3546.0044563878355

Поколение 20: Макс. штраф = 3314.159758548468, Средний штраф = 3460.29327285385

Поколение 21: Макс. штраф = 3314.159758548468, Средний штраф = 3360.250520029133

Поколение 22: Макс. штраф = 3314.159758548468, Средний штраф = 3314.159758548467

Поколение 23: Макс. штраф = 2698, Средний штраф = 3301.816563377498

Поколение 24: Макс. штраф = 2698, Средний штраф = 3240.120587522651

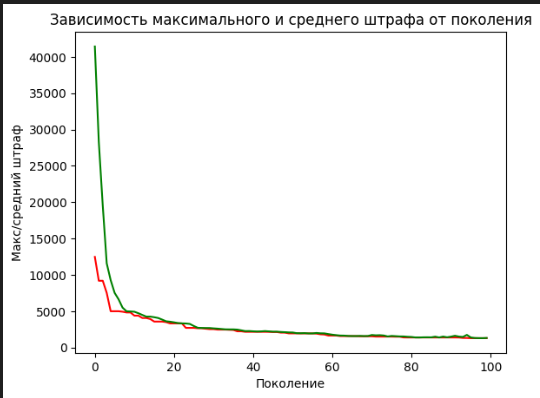
...

Поколение 96: Макс. штраф = 1281, Средний штраф = 1303.76

Поколение 97: Макс. штраф = 1281, Средний штраф = 1285.6

Поколение 98: Макс. штраф = 1281, Средний штраф = 1281.0

Поколение 99: Макс. штраф = 1281, Средний штраф = 1311.549007212825



Лучший представитель:

Продукт: Казецит, Цена: 41, Белки: 3.0, Жиры: 5.0, Углеводы: 4.8, Ккал: 75.1

Продукт: Котлеты по-гречески, Цена: 62, Белки: 6.4, Жиры: 6.7, Углеводы: 24.6, Ккал: 178.1

Продукт: Мушмула, Цена: 49, Белки: 0.0, Жиры: 0.0, Углеводы: 14.0, Ккал: 52.5

Продукт: Казецит, Цена: 41, Белки: 3.0, Жиры: 5.0, Углеводы: 4.8, Ккал: 75.1

Продукт: Перец красный сладкий, Цена: 21, Белки: 1.3, Жиры: 0.0, Углеводы: 5.7, Ккал: 26.6

Продукт: Котлеты по-гречески, Цена: 62, Белки: 6.4, Жиры: 6.7, Углеводы: 24.6, Ккал: 178.1

Продукт: Фасоль струч. консерв., Цена: 15, Белки: 1.2, Жиры: 0.1, Углеводы: 2.4, Ккал: 16.0

Продукт: Мушмула, Цена: 49, Белки: 0.0, Жиры: 0.0, Углеводы: 14.0, Ккал: 52.5

Продукт: Фасоль струч. консерв., Цена: 15, Белки: 1.2, Жиры: 0.1, Углеводы: 2.4, Ккал: 16.0

Продукт: Яблоки печеные, Цена: 197, Белки: 0.5, Жиры: 0.4, Углеводы: 43.6, Ккал: 169.1

Продукт: Фасоль струч. консерв., Цена: 15, Белки: 1.2, Жиры: 0.1, Углеводы: 2.4, Ккал: 16.0

Продукт: Перец красный сладкий, Цена: 21, Белки: 1.3, Жиры: 0.0, Углеводы: 5.7, Ккал: 26.6

Продукт: Треска соленая, Цена: 16, Белки: 23.1, Жиры: 0.6, Углеводы: 0.0, Ккал: 98.0

Продукт: Перец красный сладкий, Цена: 21, Белки: 1.3, Жиры: 0.0, Углеводы: 5.7, Ккал: 26.6

Продукт: Крапива ранняя, Цена: 61, Белки: 1.5, Жиры: 0.0, Углеводы: 5.0, Ккал: 24.8

Продукт: Перец красный сладкий, Цена: 21, Белки: 1.3, Жиры: 0.0, Углеводы: 5.7, Ккал: 26.6

Продукт: Перец красный сладкий, Цена: 21, Белки: 1.3, Жиры: 0.0, Углеводы: 5.7, Ккал: 26.6

Продукт: Язык морской, Цена: 57, Белки: 10.3, Жиры: 5.2, Углеводы: 0.0, Ккал: 88.0

Продукт: Котлеты по-гречески, Цена: 62, Белки: 6.4, Жиры: 6.7, Углеводы: 24.6, Ккал: 178.1

Продукт: Лисички, Цена: 76, Белки: 1.5, Жиры: 1.0, Углеводы: 1.0, Ккал: 19.0

Продукт: Треска соленая, Цена: 16, Белки: 23.1, Жиры: 0.6, Углеводы: 0.0, Ккал: 98.0

Продукт: Фасоль струч. консерв., Цена: 15, Белки: 1.2, Жиры: 0.1, Углеводы: 2.4, Ккал: 16.0

Продукт: Котлеты по-гречески, Цена: 62, Белки: 6.4, Жиры: 6.7, Углеводы: 24.6, Ккал: 178.1

Продукт: Котлеты по-гречески, Цена: 62, Белки: 6.4, Жиры: 6.7, Углеводы: 24.6, Ккал: 178.1

...

Продукт: Казецит, Цена: 41, Белки: 3.0, Жиры: 5.0, Углеводы: 4.8, Ккал: 75.1

Продукт: Котлеты по-гречески, Цена: 62, Белки: 6.4, Жиры: 6.7, Углеводы: 24.6, Ккал: 178.1

Продукт: Мушмула, Цена: 49, Белки: 0.0, Жиры: 0.0, Углеводы: 14.0, Ккал: 52.5

Продукт: Перец красный сладкий, Цена: 21, Белки: 1.3, Жиры: 0.0, Углеводы: 5.7, Ккал: 26.6

На графике видно, что с каждым поколением штрафы рационов приобретают всё меньшее и меньшее значение.

**Вывод**

В процессе выполнения лабораторной работы был разработан генетический алгоритм для определения наилучшего рациона с наименьшей стоимости. Был произведён анализ входных данных, вследствие чего был разработан необходимый алгоритм для эволюции популяции индивидуумов (рационов). Было освоено создание начальной популяции, изучены методы отбора, скрещивания и мутаций.